



Державний університет
телекомунікацій
Навчально-науковий інститут
інформаційних технологій
Кафедра Інженерії
програмного забезпечення



Всеукраїнська
науково-технічна конференція

Застосування програмного забезпечення в інфокомунікаційних технологіях

Збірник матеріалів
20 квітня
2023 року



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**Всеукраїнська науково-технічна конференція
ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
В ІНФОКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЯХ**

20 квітня 2023 року

ЗБІРНИК ТЕЗ

м. Київ

Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологіях». Збірник тез. – К.: ДУТ, 2023.

Даний збірник містить тези учасників конференції, представлених на Всеукраїнській науково-технічній конференції «Застосування програмного забезпечення в інфокомунікаційних технологіях», яка проходила 20 квітня 2023 р. на кафедрі Інженерія програмного забезпечення Навчально-наукового інституту інформаційних технологій Державного університету телекомунікацій, м. Київ.

Робоча мова конференції – українська та англійська.

У збірнику представлені тези доповідей Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інфокомунікаційних технологіях». Розглянуті сучасні проблеми розвитку науки і техніки та визначено шляхи їх вирішення.

Вчений секретар конференції

Тертична Юлія – Державний університет

телекомунікацій

моб.тел.+38(098)071-37-87

e-mail: infocomipz@ukr.net

ОРГАНІЗАТОРИ КОНФЕРЕНЦІЇ
Державний університет телекомунікацій
Навчально-науковий інститут інформаційних технологій
Кафедра Інженерії програмного забезпечення

ПРОГРАМНИЙ КОМІТЕТ

Бондарчук А.П. – д.т.н., професор, Державний університет телекомунікацій

Бичков О.С. – д.т.н., професор, Київський національний університет імені Тараса Шевченка

Іларіонов О.Є. – к.т.н., доцент, Київський національний університет імені Тараса Шевченка

Журавський Ю.В. – д.т.н., старший науковий співробітник, Житомирський військовий інститут ім. С. П. Корольова

Негоденко О.В. – к.т.н., доцент, Державний університет телекомунікацій

Скубак О. М. – к.т.н., доцент, Національний університет біоресурсів і природокористування України

Жебка В.В. – д.т.н., доцент, Державний університет телекомунікацій

Шевченко С.М. – к.т.н., доцент, Київський університет імені Бориса Грінченка

Шербина І.С. – к.т.н., доцент, Державний університет телекомунікацій

Садовенко В.С. – к.ф.-м.н., доцент, Державний університет телекомунікацій

НАПРЯМ 1. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕЛЕКОМУНІКАЦІЙ

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ ФІНАНСАМИ ROCKETPENNY МОВОЮ PYTHON

Диндар А.В., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

Питання контролю фінансів в наші часи є одним з найбільш актуальних. Незалежно від того, який рівень доходу у людини і звідки він отримується, витрати та контроль за ними є важливим завданням для кожного. Такі інструменти як RocketPenny допомагають людям краще керувати своїми фінансами та досягати фінансової незалежності.

RocketPenny це зручний і легкий у використанні мобільний додаток для керування фінансами, що дозволяє користувачам відстежувати свої витрати і прибутки, розподіляти кошти на різні категорії та редагувати їх, встановлювати бюджет на деякий період і контролювати його дотримання. Крім того, користувачі матимуть доступ до кругової діаграми, яка показує які витрати йдуть на яку категорію, що дозволяє легко визначати, де можна зменшити витрати, або повністю переглянути підхід до певної категорії витрат.

Для розробки додатку будуть використовуватися існуючі інформаційні технології та системи, такі як:

- Python - це основна мова програмування, яка буде використовуватися для розробки мобільного додатку для керування фінансами. Python є дуже потужним та простим в освоєнні інструментом програмування.
- Kivy - це фреймворк для розробки мобільних додатків мовою Python. За його допомогою буде створюватися інтерфейс користувача та графічні ефекти.
- SQLite - це легка база даних, яка буде використовуватись для зберігання даних користувача та нашого додатку.
- Pandas - це бібліотека для аналізу даних, яка допоможе при обробці фінансових даних користувача.
- Matplotlib - це бібліотека для візуалізації даних, яка допоможе у створенні графіків та діаграм, які в свою чергу допоможуть користувачеві краще зрозуміти своє фінансове становище.
- PyInstaller - це інструмент для збирання Python-додатків в виконуваний файли, що можна використовувати на мобільних пристроях. Зокрема, він знадобиться щоб запустити виконуваний файл програми на смартфоні з ОС Андроїд.

Література:

1. Habr [Електронний ресурс] – <https://habr.com>
2. PY4E [Електронний ресурс] – <https://www.py4e.com/book>

ВИЗНАЧЕННЯ ВИМОГ ДО ТЕЛЕГРАМ-БОТУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ ЗІ СТУДЕНТАМИ

Анісімов Б. А., Золотухіна О. А.
Державного університету телекомунікацій, м. Київ

Постановка задачі.

В наш час взаємодія між студентами та викладачами переважно проходить онлайн. Одним із найпопулярніших засобів взаємодії є месенджер Telegram, у якому учасники навчального процесу найчастіше контактують один з одним. Чат-бот дозволить спростити та автоматизувати цей процес за рахунок надання студентам та викладачам набір інструментів для організації спільної роботи.

Задачею є розробка чат-боту у месенджері Telegram який може стати доповненням до вже існуючих засобів онлайн навчання та організації інших видів роботи зі студентами.

Мета дослідження.

Визначення ключових можливостей та вимог до засобів реалізації і архітектури чат-боту для організації роботи зі студентами.

Результати дослідження.

Для розробки чат-боту на платформі .NET використовується фреймворк Telegram.Bot. Telegram.Bot є найпопулярнішим клієнтом .NET для Telegram Bot API.

Чат-бот має архітектуру “клієнт - сервер”. З боку користувача клієнтом є месенджер, через який іде взаємодія з ботом. Взаємодія з чат-ботом проходить за допомогою кнопок, тексту та команд у діалоговому вікні. Усі варіанти вибору у діалоговому вікні прописані заздалегідь.

Для роботи з ботом потрібен пристрій з можливістю підтримки месенджеру Telegram та підключення до мережі Інтернет.

Висновки та перспективи.

Створений продукт має полегшити організацію роботи зі студентами та доповнити існуючі засоби онлайн навчання, що може позитивно вплинути на успішність студентів у навчанні.

Обрані технології (.NET та Telegram.Bot) надають можливість реалізувати чат-бот з урахуванням усіх вимог, а обрана платформа для розгортання (месенджер Telegram) дозволяє охопити найбільшу кількість користувачів серед учасників навчального процесу за рахунок його популярності та наявності на усіх поширених пристроях.

Також, месенджер Telegram дає можливість створити чат-боту user friendly інтерфейс, який не викликає труднощів у використанні більшістю користувачів, що зводить витрати часу на освоєння бота та користування ним до мінімуму.

Література:

1. A guide to Telegram.Bot library - [Електронний ресурс] - Режим доступу: <https://telegrambots.github.io/book/>

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЩОДО АНАЛІЗУ КОМУНАЛЬНИХ ВИТРАТ З ВИКОРИСТАННЯМ UML ДІАГРАМИ СТАНІВ

Висоцький В.В., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Контроль за витрачанням ресурсів актуальним буде в будь який час. Аналіз витрачання ресурсів- запорука успішної економії.

Можна використовувати UML діаграми станів для моделювання меню системи з аналізу комунальних витрат.UML це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування.

За допомогою UML діаграм легше писати код(Рис.1).

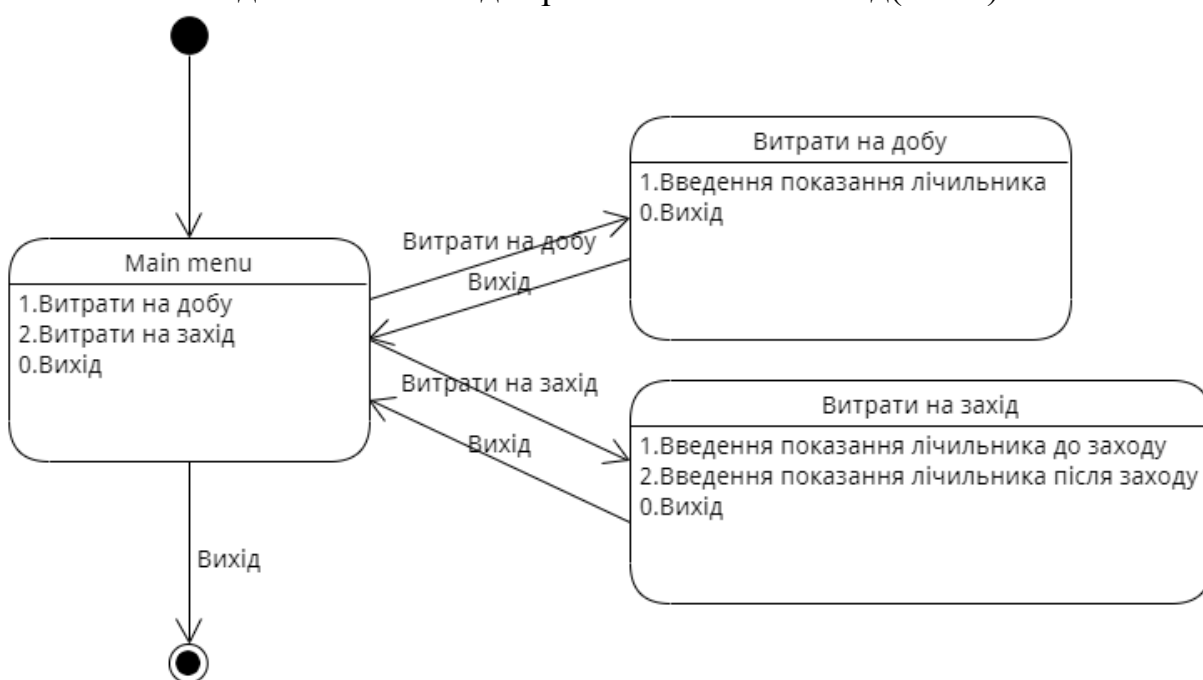


Рисунок 1-Діаграма станів (перша частина)

На даній діаграмі зображений функціонал аналізу комунальних витрат. Спочатку користувач потрапляє у головне меню програми «Main menu» і може обрати події «Витрати на добу», «Витрати на захід», «Вихід». Обравши подію «Витрати на добу» користувач потрапляє у меню «Витрати на добу» і може у події «Введення показання лічильника» ввести данні з показника для розрахунку пропонуємих витрат. (Див рис.2). Користувач отримує перелік даних із своїми витратами наприклад води за певний період і пропонуємі витрати на цей же період.

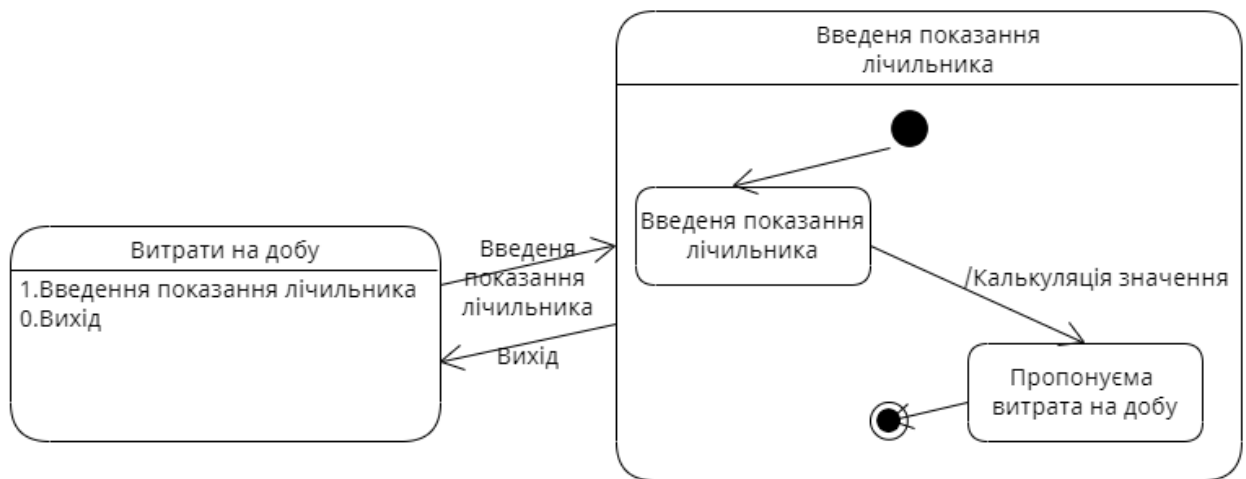


Рисунок 2-Діаграма станів(друга частина)

Діаграма станів зручна для моделювання меню аналізу комунальних витрат.

Література:

1. Опис діаграм :[Електронний ресурс]. – Режим доступу: <https://www.uml-diagrams.org/>

СХЕМОТЕХНІКА: БІОЧІП

Кужентський О.Г., Лемешко А.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Нанотехнологія – високотехнологічна галузь, спрямована на вивчення і роботу з атомами і молекулами. Однією з сфер, в якій використовується нанобіотехнологія, є створення біочіпів.

Біочіп – це узагальнююча назва технології, яка передбачає використання невеликого за розмірами твердого носія, на якому щільно нанесена велика кількість біологічних молекул для різноманітних біохімічних тестів.

Понад 90% сучасних біочіпів розробляються з використанням ДНК. Тому саме ДНК – чіп технології мають велике значення для сучасної науки.

Мета дослідження. Дослідити що таке Біочіп, а також навіщо він потрібний.

Результати дослідження. ДНК – чіпи або DNA – microarrays представляють собою організоване розміщення молекул ДНК на спеціальному носії. Частіше за все це пластинка зі скла, оскільки скло має низьку внутрішню флуоресценцію, воно є однорідним та інертним. Іноді використовують й інші матеріали, наприклад кремній, пластик.

Мікроскопічний розмір біочіпів дозволяє розмішувати на невеликій площі величезну кількість різних молекул ДНК і зчитувати з цієї площі інформацію

за допомогою флуоресцентного мікроскопа або спеціального лазерного пристрою для читання.

Виділяють 3 основних види ДНК – чіпів:

1. Синтезовані *in situ* олігонуклеотидні ДНК – чіпи (генні чіпи);
2. Нанесені на скло олігонуклеотидні мікрочіпи;
3. Нанесені на скло ДНК – чіпи.

Використання ДНК – чіпів лежить в основі багатьох методів, які дозволяють виділяти, детектувати, ідентифікувати та кількісно аналізувати експресію десятків тисяч генів. ДНК – чіпи використовуються для встановлення спорідненості, визначення генетично модифікованих організмів, ранньої діагностики онкологічних захворювань.

Біочіпи широко використовують в *in vitro*-діагностиці. В основі їх механізму дії є молекулярне розпізнавання молекул, які взаємодіють із біополімерами, нанесеними на чіп. Так, Б. використовують для діагностики туберкульозу, кліщових інфекцій, лейкемії, СНІДу, онкозахворювань, а також для експрес-діагностики вірусу грипу та стафілококової інфекції, для раннього виявлення хвороби Альцгеймера, яка відрізняється від хвороби Паркінсона. Б. ідентифікують за лічені хвилини сполуки різної хімічної природи: від білків, вітамінів, вірусів

В основі проведення аналізу на ДНК – чіпах лежить гібридизація до мішеней на поверхні ДНК – чіпа мічених флуоресцентними барвниками молекул ДНК (проб). Проби отримують за допомогою зворотньої транскрипції та полімеразної ланцюгової реакції виділеної з клітин мРНК.

Після проведення гібридизації є необхідною детекція результатів. Вона полягає у визначенні тих комірок чіпа, де відбулась гібридизація іммобілізованих на поверхні чіпа ДНК і ДНК аналізованої послідовності.

Оскільки кожна комірка чіпа містить ДНК мішень однієї певної послідовності, то її гібридизація з аналізованою ДНК однозначно вказує на наявність у складі аналізованої послідовності ДНК комплементарної послідовності ДНК мішені, розташованої в цій комірці.

Фактично, визначення комірок чіпа, в яких відбулася гібридизація іммобілізованих в ній ДНК проб з аналізованою ДНК послідовністю полягає у визначенні наявності в комірці аналізованої ДНК.

Для визначення аналізованої ДНК вона повинна бути міченою. Відмінність методів, що використовуються для детекції результатів, полягає в типі використовуваної мітки.

Для детекції інтенсивності флуоресценції кожної плями використовують апаратуру.

Висновки та перспективи. Наявність ДНК – чіпів дозволяє дуже швидко і ефективно здійснювати безліч процедур аналізу індивідуальних організмів, що дуже важливо для різних практичних заходів і завдань. На сьогоднішній день ДНК – чіпи можна використовувати для детекції однонуклеотидного поліморфізму, генотипуванні, секвенуванні мутантних геномів, вивченні екзон-інтронної будови генів, аналізі диференційної генної експресії.

Література:

1. Домар Н.А. БІОЧІП (БІОЛОГІЧНИЙ МІКРОЧІП) [Електронний ресурс] / Домар Н.А. // Фармацевтична енциклопедія – Режим доступу до ресурсу: <https://www.pharmencyclopedia.com.ua/article/6166/biochip-biologichnij-mikrochip>.
2. Майданюк Т.В. ДНК – чіп технології та їх застосування [Електронний ресурс] / Майданюк Т.В. // Національний технічний університет України «Київський політехнічний інститут», Україна – Режим доступу до ресурсу: http://www.rusnauka.com/25_SSN_2009/Biologia/51730.doc.htm.

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ ПРАВИЛ ЕТИКЕТУ МОВОЮ С#

Король Р.Є., Аверічев І.М.

Державного університету телекомунікацій, м. Київ

Розробка мобільного додатку, що дозволяє ефективно вивчати правила етикету та збільшувати культуру спілкування за допомогою інтерактивних завдань та вправ, реалізованого на платформі С#, є актуальною та перспективною задачею, яка допоможе користувачам підвищити рівень етикету та культури спілкування в сучасному цифровому світі. Розробка такого додатку передбачає проведення комплексного дослідження, включаючи аналіз наявних рішень у цій галузі, визначення потреб та очікувань цільової аудиторії, проектування та розробку додатку з використанням сучасних технологій розробки програмного забезпечення та баз даних.

Для досягнення поставленої мети, необхідно провести аналіз наявних додатків для вивчення етикету та вивчити актуальні правила етикету в сучасному світі. На основі цього аналізу, необхідно розробити дизайн та функціональність додатку, що відповідає потребам користувачів та надає зручний та доступний інтерфейс для вивчення правил етикету.

Для розробки додатку будуть використані сучасні технології програмування на платформі С#, такі як Windows Forms, ASP.NET, Xamarin – фреймворки для проектування інтерактивних вебзастосунків, які працюють на платформі .NET, що дозволяють розробляти кросплатформні нативні програми для мобільних операційних систем iOS та Android [1]. Додаток буде містити інтерактивні завдання та вправи, що дозволяють користувачам активно вивчати та закріплювати набуті знання, а також можливість збереження власних нотаток у ході прогресу навчання. У процесі розробки додатку буде враховано особливості сприйняття матеріалу користувачами різного рівня підготовки та вікових категорій, що дозволить забезпечити максимальну ефективність навчання.

Результатом роботи є створення ефективного мобільного додатку, що дозволяє вивчати правила етикету та збільшувати культуру спілкування за

допомогою інтерактивних завдань та вправ, реалізованого на платформі C#. Розробка такого додатку допоможе підвищити рівень етикету та культури спілкування серед користувачів та сприятиме розвитку та популяризації мови програмування C#. У результаті розробки додатку очікується підвищення рівня культури спілкування та відповідальності у суспільстві, що може позитивно вплинути на розвиток бізнесу, політики, культури та інших сфер життя. Також розробка мобільного додатку може мати значний практичний внесок в освіту та розвиток інформаційно-комунікаційних технологій.

Додаток буде корисним для різних категорій користувачів, включаючи студентів, бізнесменів, викладачів, та будь-яких осіб, які хочуть поліпшити свою етикет-культуру та навички спілкування. Розроблений додаток буде допомагати користувачам відповідати правильно на будь-які ситуації, що виникають у процесі спілкування, а також дозволить збільшити рівень професійного успіху та підвищити якість комунікації у різних сферах діяльності. Крім того, розробка мобільного додатку для вивчення правил етикету мовою C# має важливе значення для підвищення рівня культури спілкування та відповідальності у суспільстві. Користувачі зможуть отримати певний набір знань та навичок, які дозволять їм вести себе адекватно та професійно у різних ситуаціях.

Особливість додатку полягатиме у тому, що він буде використовувати інноваційний підхід до вивчення етикету, забезпечуючи користувачам не тільки теоретичні знання, а й практичний досвід застосування набутих знань у реальних життєвих ситуаціях. Таким чином, додаток забезпечить користувачів комплексним навчальним досвідом, що дозволить підвищити ефективність та результативність навчання.

Література:

1. Сергій Бондаренко, Що таке ASP.NET? Принцип функціонування та моделі розробки. [Електронний ресурс] - Режим доступу: <https://highload.today/uk/shho-take-asp-net-printsip-funktsionuvannya-ta-modeli-rozrobki/>

ВИМОГИ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА ПРИ РОЗРОБЦІ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ

Думенко І.О., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Інтерфейс користувача — важлива річ в кожному програмному забезпеченні. В процесі створення застосунку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання опрацьовано вимоги до інтерфейсу користувача, які необхідно враховувати:

1. Легкість використання: інтерфейс повинен бути простим і зрозумілим для користувача з будь-яким рівнем технічної грамотності. Важливо, щоб користувач міг з легкістю знайти потрібні функції та операції.
2. Надійність та стабільність: застосунок повинен працювати без збоїв та перерв у роботі, щоб не заважати користувачеві виконувати завдання.
3. Адаптивність: інтерфейс повинен бути адаптований до різних розмірів екранів та пристроїв, щоб користувач міг працювати з застосунком на будь-якому пристрої, який він використовує.
4. Зручність навігації: користувач повинен з легкістю знайти необхідні функції та виконати потрібні операції. Для цього важливо створити логічну структуру меню та кнопок.
5. Функціональність: застосунок повинен мати всі необхідні функції та можливості, які потрібні користувачеві для підготовки до зовнішнього незалежного оцінювання. Важливо також забезпечити можливість зберігання та перегляду результатів тестування.
6. Інформативність: інтерфейс повинен надавати користувачеві достатньо інформації про те, як використовувати застосунок та його функції, щоб користувач міг працювати з ним без додаткової допомоги.

Література:

1. Інтерфейс користувача: [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Інтерфейс_користувача

АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЇ ENTITY FRAMEWORK CORE ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ВІДВІДУВАЧІВ ВИСТАВКИ

Михайленко В.Д., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Об'єктно орієнтоване програмування використовує об'єкти, класи та інші концепції для написання коду, в той час як реляційна модель баз даних використовує таблиці і стовпці та реляції для організації даних. Це абсолютно різні парадигми проблеми взаємодії між якими вирішує технологія Entity Framework Core

Мета дослідження. Метою цього дослідження є розгляд застосування технології Entity Framework Core для взаємодіяти з базами даних, використовуючи об'єктно-орієнтований підхід та аналіз його можливостей, таких як виконання різних операцій з базою даних: створення, читання, оновлення та видалення записів, робота з транзакціями, виконання складних запитів та оптимізація продуктивності.

Результат дослідження. Entity Framework Core дозволяє описувати модель бази даних як класи в C#, що дозволяє розробникам працювати з даними в

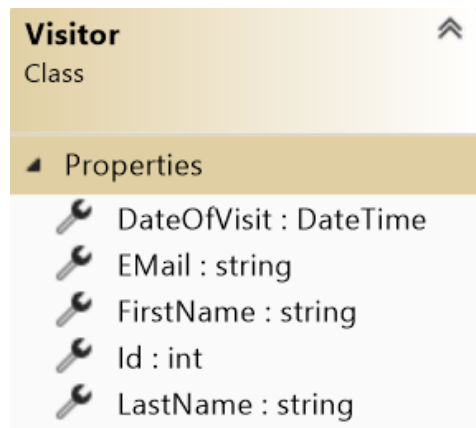
об'єктно-орієнтованому стилі, замість того, щоб писати SQL-запити вручну. EF Core автоматично генерує SQL-запити на основі взаємодії з об'єктами, EF Core надає набір класів для роботи базами даних в програмах на платформі .NET що включає в себе класи для налаштування з'єднання з базою даних, опису моделі даних, виконання SQL-запитів, відстеження змін, а також класи для виконувати CRUD-операції (додавання, зчитування, оновлення та видалення) даних в базі даних.

Також ця технологія підтримує різні типи баз даних, включаючи реляційні (такі як Microsoft SQL Server, MySQL, PostgreSQL), що робить його гнучким рішенням для взаємодії з різними системами управління базами даних.

Опис таблиць баз даних виконується за допомогою класів, які називаються класами сутностей, які представляють таблиці бази даних. Кожен клас сутності відповідає одній таблиці бази даних, а властивості цього класу відповідають стовпцям таблиці

Порівняння класів сутностей моделі C# з таблицею в базі даних :

Мій клас Visitor має 5 властивостей



Після опису моделі бази даних і конфігурації її ми отримали таке відтворення таблиці в базі даних

	Id	FirstName	LastName	EMail	DateOfVisit
1	1	FName1	LName1	Email1@gmail.com	2019-12-28
2	2	FName2	LName2	Email2@gmail.com	2019-12-29
3	3	FName3	LName3	Email3@gmail.com	2019-12-31
4	4	FName1	LName1	Email1@gmail.com	2019-12-28

Для конфігурації таблиці я використовував підхід Fluent API, і ось приклад налаштування моєї бази даних:

```
builder
    .ToTable("Visitors");

builder
    .HasKey(x => x.Id);

builder
    .Property(x => x.FirstName)
    .HasColumnName("FirstName")
    .HasMaxLength(50)
    .IsRequired();

builder
    .Property(x => x.LastName)
    .HasColumnName("LastName")
```

```

        .HasMaxLength(50)
        .IsRequired();

builder
    .Property(x => x.Email)
    .HasColumnName("EMail")
    .HasMaxLength(100)
    .IsRequired();

builder
    .Property(x => x.DateOfVisit)
    .HasColumnName("DateOfVisit")
    .HasColumnType("date");

```

І ось результат генерації SQL- запита

```

USE [EfCoreBasicDb]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Visitors](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [FirstName] [nvarchar](50) NOT NULL,
    [LastName] [nvarchar](50) NOT NULL,
    [EMail] [nvarchar](100) NOT NULL,
    [DateOfVisit] [date] NOT NULL,
    CONSTRAINT [PK_Visitors] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

```

Висновок. Використання технології Entity Framework Core для розробки додатків з використанням баз даних, спрощує розробку моделі даних - дозволяє використовувати класи .NET для визначення моделі даних, що спрощує розробку та підтримку моделі даних, так як є більш зручним та зрозумілим підходом, а також дозволяє автоматично оновлювати базу даних під час збереження змін без потреби вручну виконувати SQL-запити. Тобто має великі переваги, які сприяють комфортній та швидкій розробці програмного забезпечення для обліку відвідувачів виставки

Література:

1. Офіційна документація EF Core
<https://docs.microsoft.com/en-us/ef/core/>
2. Відео курс від навчального центру CyberBionic Systematics
<https://itvdn.com/ua/video/ef-core-basics>

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ВІДВІДУВАЧІВ ВИСТАВКИ З ВИКОРИСТАННЯМ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ

Михайленко В.Д., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Розробка програмного забезпечення для обліку відвідувачів виставки може бути складним процесом, який включає такі етапи як аналіз вимог, проектування, реалізацію, тестування та впровадження. Один з ефективних методів проектування програмного забезпечення є використання діаграм прецедентів UML

Мета дослідження. Використання діаграм прецедентів допомагає ідентифікувати користувачів системи, які взаємодіють з системою обліку відвідувачів виставки, та описати її функціональність через прецеденти - сценарії взаємодії між акторами та системою. Це допомагає зрозуміти вимоги до системи та її функціональність.

Результат дослідження. Результатом дослідження стала готова Діаграма прецедентів, яка стала основою для розробки функціональності системи обліку відвідувачів виставки, виявленням можливої взаємодією з програмою.

Діаграма прецедентів була використана в якості вказівок для розробки коду програмного забезпечення: реалізація функцій, взаємодій між акторами та обробку вхідних даних.

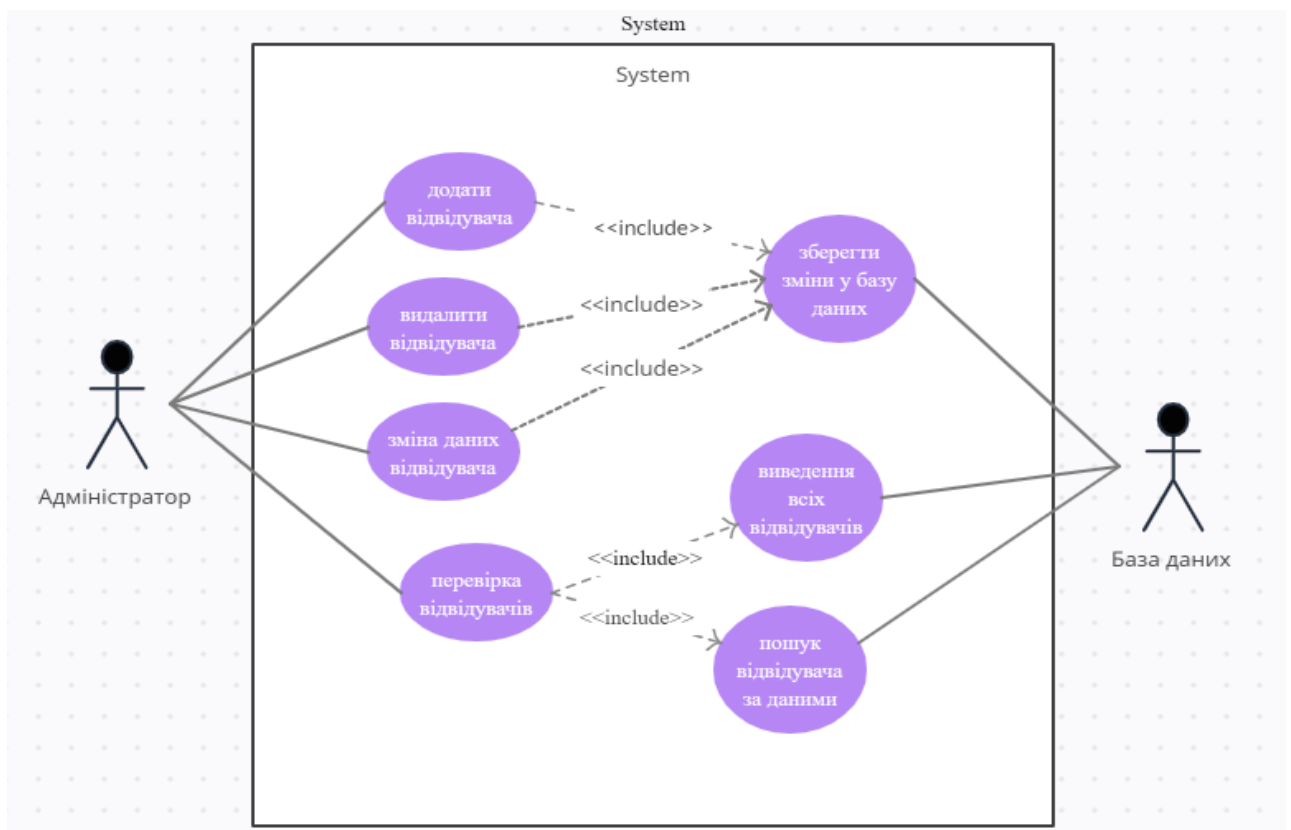


Рисунок 1 – Діаграма прецедентів пз для обліку відвідувачів виставки

Опис прецедентів:

1. Додати відвідувача – адміністратор виставки вписує дані нового відвідувача і зберігає зміни у базу даних.
2. Видалити відвідувача - адміністратор виставки видаляє відвідувача іта його дані і зберігає зміни у базу даних.
3. Зміна даних відвідувача - адміністратор виставки змінює дані відвідувача і зберігає зміни у базу даних.
4. Перевірка відвідувачі - адміністратор виставки перевіряє дані відвідувачі, що включає в себе або виведення всіх відвідувачів, або пошук відвідувачів за конкретними даними.

Висновок. Використання UML діаграм прецедентів є ефективним методом при розробці програмного забезпечення для обліку відвідувачів виставки. Вони допомагають визначити вимоги до системи, проектувати її структуру та взаємодії між різними компонентами.

Література:

1. SQL Server technical documentation : [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
2. Офіційна документація EF Core : [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/ef/core/>
3. UML - Basic Notations : [Електронний ресурс]. – Режим доступу: https://www.tutorialspoint.com/uml/uml_basic_notations.htm

ВИКОРИСТАННЯ HTML, CSS, JAVASCRIPT ПРИ РОЗРОБЦІ FRONTEND-ЧАСТИНИ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ

Думенко І.О., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

HTML, CSS та JavaScript використовуються для створення зовнішнього інтерфейсу користувача веб-застосунку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання.

HTML використовується для створення структури веб-сторінки та додавання на них текстового та мультимедійного контенту, такого як зображення, відео та звук. Наприклад, для створення форм введення даних, списків та інших компонентів інтерфейсу користувача.

CSS використовується для стилізації елементів веб-сторінки та задання їх вигляду та розміщення на екрані. Наприклад, використовується для зміни кольору тла, шрифтів, рамок та іншого візуального оформлення.

JavaScript використовується для додавання динаміки на веб-сторінці та забезпечення інтерактивності. Наприклад, використовується для валідації введених

даних, анімації елементів інтерфейсу, обробки подій користувача та забезпечення іншої функціональності.

Для забезпечення легкості використання сайту з підготовки до зовнішнього незалежного оцінювання використовується така структура сайту:

header	
navigation	
section	aside
article	
footer	

Рисунок 1 — Структура сайту

HTML має семантичні елементи, які забезпечують цю структуру:
<header>, <nav>, <section>, <article>, <aside>, <footer>.

Для забезпечення адаптованості до різних розмірів екранів використовується адаптивна верстка. Основою її є медіа запити CSS @media.

Для забезпечення зберігання та перегляду результатів тестування в базі даних MySQL використовується модуль JavaScript mysql. Він дозволяє мати доступ до бази даних MySQL та здійснювати маніпуляцію ними.

Література:

1. Node.js MySQL: [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/nodejs/nodejs_mysql.asp
 2. HTML Layout Elements and Techniques : [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/html/html_layout.asp
- Media CSS значення : [Електронний ресурс]. – Режим доступу: <https://css.in.ua/css/value/media>

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЩОДО АНАЛІЗУ КОМУНАЛЬНИХ ВИТРАТ З ВИКОРИСТАННЯМ UML ДІАГРАМИ СТАНІВ

Висоцький В.В., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Контроль за витрачанням ресурсів актуальним буде в будь який час. Аналіз витрачання ресурсів- запорука успішної економії.

За допомогою UML діаграм легше писати код(Рис.1).

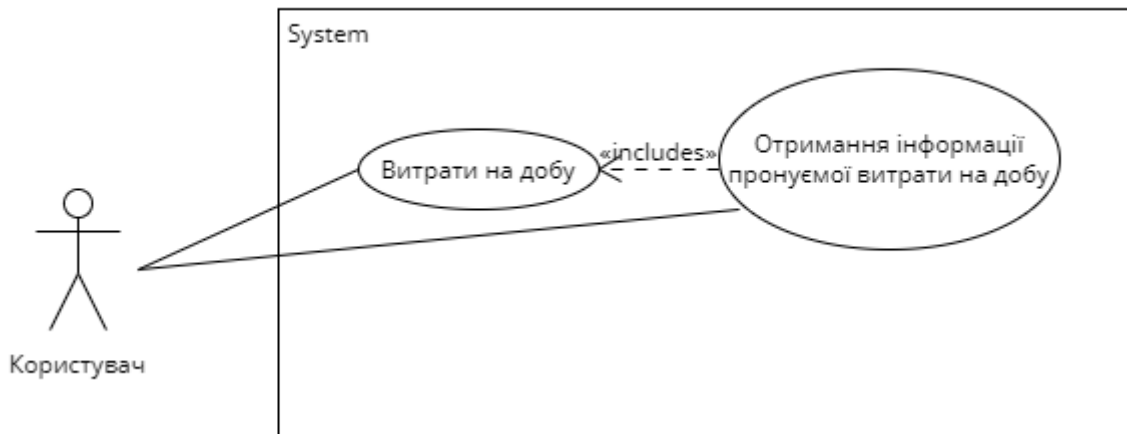


Рисунок 1-Діаграма функціоналу системи (перша частина)

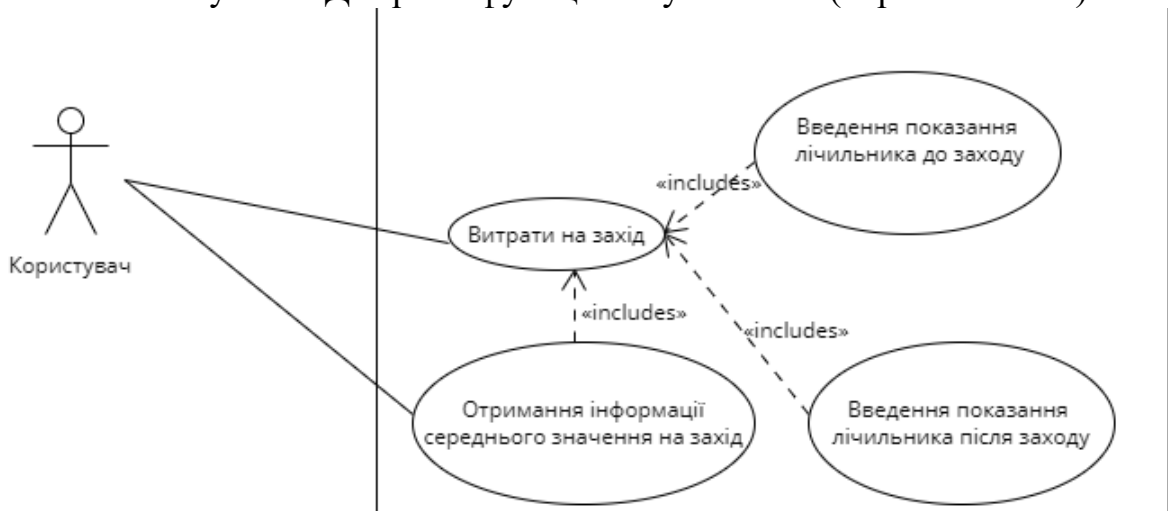


Рисунок 2-Діаграма станів(друга частина)



Рисунок 3-Діаграма станів(друга частина)

На діаграмі функціоналу зображений функціонал аналізу комунальних витрат. Користувач може звернутися до методу «Витрати на добу» який включає у себе метод «отримання інформації проноуємої витрати на добу». Це надасть інформацію користувачу про його добові комунальні витрати.

Також користувач може дізнатися скільки йому потрібно витрат до певного заходу через метод «витрати на захід» який включає у себе три метода «введення показання лічильника до заходу», «введення показання лічильника після заходу», «Отримання інформації середнього значення на захід».

І ще є метод який дозволяє планувати витрати на певний період «Планування витрат на добу», «Вибір заходів що обумовлено пропонуємою витратою на добу».

Діаграма функціоналу зручна для моделювання меню аналізу комунальних витрат.

Література:

1. Опис діаграм :[Електронний ресурс]. – Режим доступу: <https://www.uml-diagrams.org/>

АНАЛІЗ СУЧАСНИХ ІНСТРУМЕНТІВ ТА МЕТОДІВ СТВОРЕННЯ ВЕБСАЙТІВ

Малюк В.В., Яскевич В.О.

Державного університету телекомунікацій, м. Київ

Постановка задачі

Завдання полягає у дослідженні використання різноманітних методів створення вебсайтів. Дослідити кожний метод, знайти всі переваги та недоліки кожного способу, щоб виявити, що актуальніше і зручніше використовувати в наш час.

Мета дослідження

Основна мета дослідження – це зрозуміти, яку саме стратегію створення сайту обирати в залежності від власних навичок, складності самого вебсайту та фінансової доступності.

Результати дослідження

Вебсайти можна створювати різними способами, якщо у вас є певні навички програмування, то ви можете створити сайт самостійно, в залежності від його складності та ваших вмінь. Якщо ви хочете зробити односторінковий Landing page, то вам цілком буде достатньо знати основи HTML (мова розмітки сторінки), CSS (мова стилів сторінки) і JS (для надання сторінці живого вигляду). Але, якщо ви маєте на меті створити, наприклад, інтернет магазин, то вам вже знадобиться ширший стек технологій у вашому запасі, наприклад PHP щоб створити реєстрацію на сайті, MySQL для створення бази даних для товарів чи коментарів, зображень і так далі. Тому, якщо ви хочете створити щось відносно просте, ви можете самостійно розібратися з основами створення веб сайтів і таким чином уже самостійно розробити даний ресурс. Але, якщо ви все ж таки плануєте відкривати особистий інтернет магазин, то

нажаль, маючи навіть високий рівень навичок у програмуванні, ви витратите дуже багато часу для його створення.

Другий спосіб створення вебсайтів – це онлайн конструктори. На сьогоднішній день існують дуже багато конструкторів, які надають можливість безкоштовно створювати доволі таки непоганий вебсайт навіть людині без досвіду та навичок програмування. Зазвичай, користуючись безкоштовними тарифами, ви зможете створювати прості односторінкові додатки без особливого функціоналу. Але, якщо ви хочете додати до свого вебсайту більш цікавий і зручний функціонал, ви можете користуватись платним тарифом. Є можливість обрати будь-який шаблон і просто заповнити його контентом, тексти зображення і т.д. Але нажаль у таких конструкторів зазвичай мінусів більше ніж плюсів. Насамперед – це однотипність, ви можете знайти в інтернеті десятки візуально майже однакових вебсайтів, через те, що не всі конструктори дають можливість змінювати шаблони, а якщо така можливість є, то зміни в шаблоні можуть бути не суттєві. Також при створенні такого вебсайту, конструктор генерує HTML код, який в свою чергу через свою насиченість різними непотрібними стилями займає від 1 до 10 Мб, в порівнянні з простим HTML, який буде займати близько 100 Кб, очевидно, що це сильно вплине на швидкість завантаження сторінки. Також можуть виникнути проблеми, якщо ви вирішите просувати свій вебсайт, так як до таких сайтів, пошукові системи відносяться з недовірою. Ну, і звісно ж, гроші, звичайні шаблони зазвичай середньої якості, якщо вам потрібно щось більш унікальне, то такі шаблони коштують грошей.

Третій спосіб являє собою розробку сайту через CMS (Content Management System). В готовій CMS системі можна використовувати вже готовий шаблон, переробивши його під свої власні потреби. Головна перевага це простота і доступність таких систем в інтернеті. Тут не потрібні якісь особливі навички програмування і до того ж даний вид сайтів буде динамічно зберігати всі оновлення, також регулярно з'являються оновлення, що покращують роботу системи. Головний недолік – це необхідність вивчення структури обраної системи, а також можуть виникнути труднощі з платними версіями деяких систем. В інтернеті є величезна кількість інформації, яка може бути як корисною так і шкідливою, нажаль, деякі люди створюють контент по тематиці використання CMS, але самі далекі від розуміння принципів роботи такої системи, і можуть навчати людей як вносити правки в код, тим самим при оновленні системи є досить великий шанс того що ви зламаєте свій вебсайт.

Висновки та перспективи

Підсумовуючи все вище сказане, можна зробити наступні висновки – вибір методу створення вебсайтів напряду залежить від його складності та його типу. Якщо ви плануєте розробити доволі простий односторінковий сайт, і не бажаєте вивчати щось для його створення, вам ідеально підійдуть сайти конструктори. Якщо ви хочете все ж таки додати до свого сайту певну гнучкість та динамічність, ви можете використовувати CMS системи, використавши уже готові шаблони переробивши їх під себе. Але, якщо все ж

таки ваш сайт буде мати певний функціонал і архітектуру, то вам краще розробити такий сайт з нуля самостійно, якщо дозволяють навички та час, або знайшовши додаткових розробників. Індустрія веб розробки збільшується у своїх масштабах кожен день, і це не дивно, оскільки сьогодні у світі активно проходить оцифрування. Кожна людина, яка займається підприємницькою діяльністю, планує зробити або вже зробила свій власний вебсайт для розширення бізнесу. Саме в інтернеті можна розмістити свій вебсайт, яким будуть користуватися велика кількість користувачів за умови правильності налаштування. Очевидно, що сьогодні мати свій сайт це вигідно для бізнесу. Саме тому люди і шукають для себе можливість розробити свій власний вебсайт. Займатися розробкою вебсайтів стало також вигідним, оскільки де є попит там є і пропозиція.

Література:

1. [Електронний ресурс] <https://www.avahost.ua/ua/sposobi-stvorennya-sajtiv/> Вебсайт Avahost.ua
2. [Електронний ресурс] <https://zcollage.com.ua/perevahy-ta-nedoliky-onlayn-konstruktoriv-dlia-stvorennia-sajtiv/>
3. [Електронний ресурс] <https://www.dali.te.ua/dali/964-analiz-perevah-ta-nedolikiv-cms.html>

РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ СТВОРЕННЯ ІНДИВІДУАЛЬНОГО ПЛАНУ ХАРЧУВАННЯ ТА ТРЕНУВАНЬ ЛЮДИНИ

Малюк В.В., Яскевич В.О.

Державного університету телекомунікацій, м. Київ

Постановка задачі

Завданням проєкту є створення вебсайту де люди матимуть змогу створити свій власний план спортивної дієти для нарощування маси або навпаки схуднення, і також створити план тренувань для схуднення або нарощування м'язів.

Мета дослідження

В Україні кожна четверта особа має зайву вагу і кожна п'ята – страждає на ожиріння. Записатися у спортзал недостатньо, потрібно також знати яким навантаженням піддавати своє тіло, щоб не заподіяти шкоди своєму здоров'ю. Але, якщо тренування допомагають позбутися зайвого жиру, то спортивна дієта слугує ліками для організму. Саме тому мною була взята ініціатива створення вебсайту, де люди зможуть для себе підібрати план тренувань і раціон спортивної дієти.

Результати дослідження

В результаті було розроблено вебсайт, на якому людина зможе ознайомитися з інформацією, яка пов'язана із здоровим способом життя, переглянути цікаві тематичні відео, почитати цікаві факти про тренування та

правильне харчування. Також будь-яка людина зможе у своєму особистому кабінеті розробити свою власну стратегію тренувань і план дієтичного харчування для того, щоб привести своє тіло і здоров'я в порядок, або покращити свою наявну фізичну форму.

Висновки та перспективи

В наш час дійсно потрібно слідкувати за своїм здоров'ям та тілом. Нажаль проблема ожиріння присутня, як в Україні, так і у всьому світі. Проблема нікуди не зникне, але є можливість знизити негативну статистику проблем з ожирінням і зайвою вагою. Маючи уже готовий план тренувань і спортивної дієти, людина може значно підвищити результативність своїх тренувань. Спочатку планується повністю безкоштовний доступ до функцій створення планів тренувань та харчування, а в майбутньому можлива інтеграція реклами або невеликої плати за користування ресурсом. В будь-якому випадку, при правильному просуванні сайту, можна добитися отримання монетизації тобто отримання прибутку від користування сайтом.

Література:

1. [Електронний ресурс] <https://www.bsmu.edu.ua/blog/ozhyrinnya-aktualna-problema-dlya-rozvynutyh-krayin/>
2. [Електронний ресурс] <https://nutrium.com/blog/how-to-create-meal-plans-your-athletes-will-want-to-follow/>

ЗАБЕЗПЕЧЕННЯ ВІДПОВІДНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО СТАНДАРТІВ БЕЗПЕКИ, ЩО СПРИЯЄ ПІДВИЩЕННЮ НАДІЙНОСТІ ТА ДОВГОВІЧНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кужентський О.Г., Лемешко А.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Задача полягає в забезпеченні відповідності програмного забезпечення до стандартів безпеки, з метою збільшення надійності та довговічності програми. Це може включати аналіз потенційних загроз безпеці, підвищення рівня захисту від хакерських атак та виконання інших заходів безпеки, таких як криптографічні алгоритми, контроль доступу і захист від вразливостей програмного забезпечення. Забезпечення відповідності до стандартів безпеки може допомогти запобігти збоїв і відновити роботу програми у разі виникнення проблем. Результатом таких заходів повинне бути стабільне та безпечне програмне забезпечення, яке виконує свої функції відповідно до вимог користувачів.

Мета дослідження. Метою дослідження є вивчення методів забезпечення відповідності програмного забезпечення до стандартів безпеки з метою збільшення надійності та довговічності програмного забезпечення.

Результати дослідження. Забезпечення відповідності програмного забезпечення до стандартів безпеки є важливим кроком у забезпеченні високої якості програмного забезпечення. Це дозволяє запобігти виникненню вразливостей та інших проблем безпеки, що можуть призвести до серйозних наслідків для користувачів програмного забезпечення.

Стандарти безпеки, такі як ISO 27001, OWASP Top 10 та CERT C, містять вимоги щодо безпеки програмного забезпечення, які допомагають розробникам забезпечити високий рівень безпеки своїх продуктів. Це може збільшити вартість та тривалість проекту розробки програмного забезпечення, проте забезпечення відповідності до стандартів безпеки варто розглядати як інвестицію у якість та надійність програмного забезпечення.

Висновки та перспективи. Забезпечення відповідності програмного забезпечення до стандартів безпеки - це важлива складова якості та надійності програмного забезпечення. Дотримання стандартів безпеки дозволяє зменшити ризики вразливостей та інших проблем безпеки, що можуть привести до негативних наслідків для користувачів програмного забезпечення. Хоча забезпечення відповідності до стандартів безпеки може збільшити вартість та тривалість проекту розробки програмного забезпечення, це може зекономити значні кошти та час в майбутньому, оскільки програмне забезпечення буде міцнішим, стійким та менш вразливим до атак. Тому, забезпечення відповідності програмного забезпечення до стандартів безпеки - це важлива практика для забезпечення високої якості та надійності програмного забезпечення.

Література:

1. Невказаний. Класифікація моделей якості [Електронний ресурс] / Невказаний – Режим доступу до ресурсу:
https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/page6.html.
2. Заплотинський Б.А. ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ
<http://dspace.onua.edu.ua/bitstream/handle/11300/11111/%D0%9E%D0%86%D0%91%20%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9.pdf?sequence=1&isAllowed=y> / Б. А. Заплотинський

ВИМОГИ ДО BACKEND- ЧАСТИНИ ПРИ РОЗРОБЦІ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ

Дубей Р.Р., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

При розробці backend-частини застосунку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання вимагається використання певних технологій та забезпечення деякої функціональності.

1. База даних: Потрібно мати базу даних для зберігання інформації про курси, учасників курсів, вчителів, результати оцінювання та інше. Вимагається використання баз даних, які забезпечують швидкий доступ та збереження даних з належною безпекою.
2. Аутентифікація та авторизація: Для забезпечення безпеки та обмеження доступу до певних ресурсів потрібно використовувати механізми аутентифікації та авторизації. Наприклад, можна використовувати технології, такі як JWT або OAuth.
3. API: Backend-частина повинна мати API, який надає доступ до даних та функціональності застосунку. Наприклад, API може надавати можливість створення та редагування курсів, реєстрацію учасників, отримання результатів оцінювання та інше.
4. Сервер: Backend-частина повинна бути розгорнута на стабільному та швидкому сервері, який забезпечує постійний доступ до застосунку. Важливо використовувати серверні технології, які забезпечують високу продуктивність та масштабованість застосунку.
5. Безпека: Backend-частина повинна бути забезпечена від потенційних загроз безпеці, таких як атаки на базу даних, злам ресурсів та інше. Важливо використовувати методи шифрування та інші заходи безпеки для захисту від зловмисників.

Література:

1. Front end та back end : [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Front_end_%D1%82%D0%B0_back_end

ВИКОРИСТАННЯ .NET CORE ПРИ РОЗРОБЦІ BACKEND-ЧАСТИНИ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ

Дубей Р.Р., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

.NET Core - це крос-платформний фреймворк для розробки застосунків, що дозволяє розробникам створювати високопродуктивні та масштабовані застосунки. Використання *.NET Core* при розробці backend-частини застосунку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання дозволяє забезпечити швидку та надійну роботу застосунку.

Основні переваги використання *.NET Core* включають:

1. Крос-платформність - фреймворк підтримує розробку застосунків для *Windows*, *Linux* та *macOS*.
2. Висока продуктивність - застосунки, розроблені з використанням *.NET Core*, зазвичай працюють швидше та забезпечують кращу продуктивність порівняно з іншими фреймворками.
3. Масштабованість - *.NET Core* дозволяє розробникам розробляти масштабовані застосунки, що можуть обслуговувати велику кількість користувачів.
4. Безпека - *.NET Core* має вбудовану систему безпеки та захисту від атак.
5. Легкість у використанні - *.NET Core* має простий та зрозумілий інтерфейс, що дозволяє розробникам швидко розпочати роботу з ним.

Таким чином, використання *.NET Core* при розробці backend-частини застосунку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання дозволяє забезпечити ефективну та надійну роботу застосунку, що відповідає вимогам сучасних стандартів розробки програмного забезпечення.

Література:

1. ASP.NET Core : [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/ASP.NET_Core

АНАЛІЗ ТА РОЗРОБКА ВЕБ-РЕСУРСІВ З ПІДТРИМКОЮ ІНКЛЮЗИВНОГО ІНТЕРФЕЙСУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ З ВИКОРИСТАННЯМ СПЕЦІАЛІЗОВАНИХ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ

Юхимець Б.С.

Державного університету телекомунікацій, м. Київ

Існує багато типів інвалідності, для кожного типу є певні вимоги з Веб-доступності. Розглянемо деякі з них. Незрячі користувачі використовують програму зчитування з екрана, вона озвучує все, що знаходиться на екрані. Приклад з сайтом: програма зчитування з екрана спочатку прочитає назву веб-переглядача та заголовок веб-сторінки. Це заголовок, який відображається в рядку заголовка веб-переглядача або на вкладках у веб-переглядачі, а також коли сторінка додається до закладок. Потім її програма зчитування з екрана зчитує сторінку зліва направо, яка, залежно від вмісту сторінки, може прочитувати такий вміст, як логотип і слоган, можливо, деяку вступну інформацію, поле пошуку, область навігації, як-от меню, і потім вміст головної сторінки.

Увесь вміст має бути представлений у вигляді тексту або через текстовий еквівалент наприклад, альтернативний текст для зображень або інших нетекстових об'єктів.

Програми зчитування з екрана не можуть безпосередньо читати нетекстовий вміст наприклад, зображення, але вони можуть читати альтернативний текст, який ви надаєте.

Інформація не повинна передаватись лише візуальними атрибутами наприклад, кольором, просторовим розташуванням, товщиною тексту, виділенням фону.

Не вся візуальна інформація доступна для програм зчитування з екрана. Навіть візуальні атрибути, які доступні для деяких програм зчитування з екрана, як-от колір тексту, зазвичай не повідомляються за замовчуванням.

функції мають бути доступні лише за допомогою клавіатури.

Незважаючи на те, що більшість незрячих користувачів фізично можуть використовувати мишу або трекпад, це не приносить їм особливої користі, оскільки вони не бачать, де знаходиться вказівник миші. Їм ефективніше орієнтуватися за допомогою клавіатури.

Контент повинен використовувати розмітку з хорошою структурою та семантикою заголовки, орієнтири, таблиці, списки.

Користувачі програми зчитування з екрана часто відкривають списки заголовків, орієнтирів та інших семантичних елементів, щоб допомогти їм зрозуміти, що на сторінці. Вони також можуть переходити за цими елементами наприклад, переходити безпосередньо до орієнтира основного вмісту або до певного заголовка.

Усі користувацькі елементи керування наприклад, кнопки розгортання/згортання, регулятор гучності медіаплеєра, діалогові вікна

повинні мати правильну назву/мітку, роль (або з HTML, або з ARIA. Люди зі слабким зором бачать, але їх зір ще досить далекий від ідеального навіть з потужними коригувальними лінзами. вони не бачать достатньо добре, щоб читати більшість друкованого тексту, якщо вони не збільшують його. Слабкий зір – це не одиничний стан. Це широка категорія, яка охоплює багато різних станів із різним ступенем погіршення. Веб-доступність для людей зі слабким зором.

Читання дрібного тексту є надзвичайно складним завданням для користувача зі слабким зором. їм може знадобитися використовувати функцію збільшення екрана, як-от ZoomText або MAGic, яка одночасно показує невелику частину екрана. Коли екран збільшується, він збільшує частину екрана та відображає її з високою роздільною здатністю.

ZoomText і MAGic має велику кількість функцій, які допомагають оптимізувати візуальний досвід для людей із слабким зором. Якщо на веб-сторінці вимкнено масштабування користувачі зі слабким зором, які використовують екранні лупи для читання вмісту, можуть не бачити належним чином інформацію на веб-сторінці.

Увесь текст має відповідати вимогам щодо контрастності фону перевірте за допомогою розширення для веб-переглядача доступності DevTools Deque або подібного інструменту.

Деякі користувачі зі слабким зором можуть бачити низький контраст. Таким чином, текст, рамки та інші елементи можуть мати однакові або подібні відтінки яскравості. Текстові елементи, яскравість яких надто близька до фонових кольорів, може бути надзвичайно важкою для читання цим користувачам.

Посилання, кнопки та елементи керування повинні мати видимий :focusстан і повинні мати видимий :hoverстан . Інтерфейс користувача має забезпечувати чітке візуальне розмежування між вмістом, текстом і елементами керування, кнопками, посиланнями. користувачі, які можуть бачити вміст із низьким контрастом, можуть мати труднощі з визначенням того, чи діють елементи керування на веб-сторінці, оскільки ці елементи можуть змішуватися з навколишнім текстом і кольорами фону. Не використовуйте лише колір для передачі інформації

Оскільки кольори можуть виглядати схожими один на одного для дальтоніків, дуже важливо, щоб колір не використовувався як єдиний спосіб передачі чи розрізнення інформації.

Проблема глухоти полягає в доступності аудіо у форматі лише аудіо або аудіо в поєднанні з відео, як це часто буває. Для попередньо записаних лише аудіофайлів найкращим рішенням є розшифровка аудіо. Стенограма має включати всю розповідь, діалог та інші важливі звуки, як-от оплески, тріски, музика. Усний переклад відео мовою жестів може бути дуже корисним.

Доступність для людей з різними обмеженими можливостями. Доступність для людей з кількома обмеженими можливостями залежить від конкретних потреб людини. Якщо необхідно, комунікацію слід підтримувати за допомогою доповнювальної та альтернативної комунікації, такої як мова

жестів, матеріальні символи або комп'ютерне мовлення. Людям із кількома вадами також може бути корисно використовувати допоміжні технології, щоб допомогти подолати функціональні обмеження та підтримати спілкування. Оскільки важко зрозуміти, що може знадобитися людині з кількома вадами, надання підтримки як допоміжних, так і альтернативних комунікаційних і допоміжних технологій, а також дотримання стандартів доступності є ключовими для того, щоб зробити фізичне та технологічне середовище більш доступним.

Література:

1. Ендрю Кіркпатрік, Майкл Р. Беркс Web Accessibility, 2006 — 354 с.
2. Онищенко, В. В., Бондарчук А. П. "Програмна інженерія: проблеми та перспективи." Зв'язок 1 (2015): 10-13.
3. Introduction to Web Accessibility – W3C. URL: <https://www.w3.org/WAI/fundamentals/accessibility-intro/> (дата звернення: 10.04.2021).

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ПРОЦЕСУ НАДАННЯ ПЕРШОЇ ДОПОМОГИ

Спіцин А.Я., Трінтіна Н.А.

Державний університет телекомунікацій, м. Київ

Програмне забезпечення для першої допомоги у реальному часі буде здатне надати необхідні інструкції для надання допомоги потерпілому в аварійній ситуації. Основна робота програми полягатиме у відтворенні аудіо-інструкцій для користувача. Всі інструкції будуть начитані штучним інтелектом, щоб забезпечити максимальну нейтральність у голосі.

Інтерфейс програми буде максимально мінімалістичним. На екрані будуть знаходитися великі кнопки з варіантами стану потерпілого. Увесь шрифт також буде доволі великим та легко читатися, щоб користувачі могли легко вибрати потрібну опцію.

При відкриванні додатку, програма одразу почне читати вступний текст, де пропонуватиметься оглянути потерпілого за процедурою та обрати, що саме з ним не так. Після обрання стану потерпілого, програма почне зачитувати відповідні до випадку інструкції, змінюючи кнопки на варіанти наступного вибору.

Для реалізації цього додатку була обрана мова програмування Python. Причиною такого рішення є те, що Python є дуже популярною мовою програмування серед розробників. Вона має велику кількість бібліотек та фреймворків, які полегшують розробку програмного забезпечення. Крім того, Python має простий синтаксис та легко читається, що полегшує розробку та розуміння коду.

Зважаючи на технічні обмеження, перша версія програмного забезпечення буде доступна лише для мобільних пристроїв на базі операційної системи Android. Крім того, важливо зазначити, що додаток буде розроблений лише для української мови, щоб забезпечити максимальний комфорт та зрозумілість користувачам української спільноти.

У світі, де технології швидко розвиваються, ця програма може стати незамінною для людей, які опинилися в небезпечній ситуації та потребують невідкладної допомоги. Програмне забезпечення для першої допомоги у реальному часі може допомогти простим людям надати необхідну допомогу, покращуючи тим самим становище поранених та хворих.

Звичайно, важливо зазначити, що інструкції, які будуть використовуватися в додатку, будуть взяті з надійного та авторитетного джерела - Червоного Хреста. Це забезпечить максимальну достовірність інформації та допоможе користувачам надавати першу допомогу на високому рівні. Додаток стане надійним помічником в будь-яких непередбачуваних ситуаціях та дозволить швидко та ефективно діяти в екстремальних умовах.

Література:

1. Red Cross. Comprehensive Guide for First Aid & CPR – Режим доступу: https://www.redcross.ca/crc/documents/comprehensive_guide_for_firstaidcpr_en.pdf
2. Сайт Appstirr [Електронний ресурс] – Режим доступу: <https://www.appstirr.com/blog/how-to-convert-an-android-app-to-ios>

ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ТА ПІДТРИМУВАНOSTІ СЕРВЕРНОГО ДОДАТКУ ЗА ДОПОМОГОЮ БАГАТОРІВНЕВОЇ АРХІТЕКТУРИ

Решетнік Н.О., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

У сучасному світі, коли додатки виконують роль ключового елемента бізнесу, його надійність, стійкість, підтримуваність та потенціал до розширення є невід'ємною частиною його успішної роботи. За останні десять років підходи мінялись та розроблялись все більш новітні та ефективні підходи, у цій тезі ми розглянемо ключові принципи одної із базових та більш розповсюджених архітектур додатків – багаторівневу архітектуру або N-Tier Architecture. Також розглянемо її переваги та недоліки, реальні приклади успішної реалізації та різні варіанти реалізацій.

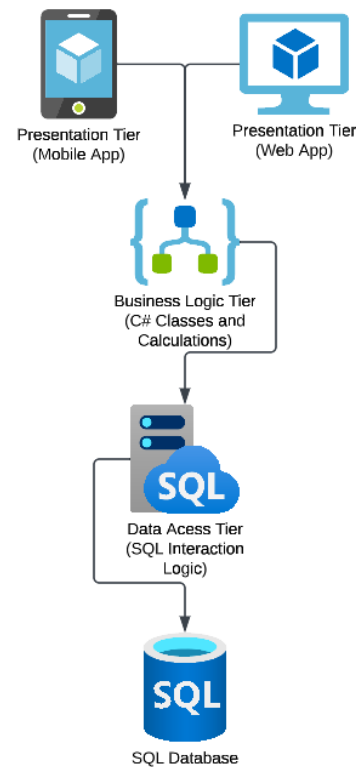
У розробці програмних додатків – багаторівнева архітектура, це клієнт-серверна архітектура в якій використовується підхід розділення відповідальності окремих частин додатків, які з'єднані між собою послідовно, та кожен компонент знає там має доступ тільки тих що поруч по ієрархії. Найбільш популярна варіація багатошарової архітектури це трьохрівнева архітектура (Рис. 1), де додаток поділяється на три основні частини:

1. Рівень представлення (Presentation Layer) – це самий перший та верхній рівень який представляє собою інтерфейс користувача. Його основна задача – відображення даних, та взаємодія з користувачем через графічний інтерфейс додатку. Цим рівнем може виступати веб-додаток, додаток для ПК чи мобільних пристроїв
2. Рівень бізнес-логіки (Business Logic Layer) – це сервісний рівень який в собі містить всю бізнес логіку додатку. Саме в цьому рівні відбуваються всі необхідні потрібні фільтрації, обчислення, та дії які були зініційовані клієнтом або самим додатком.
3. Рівень доступу до даних (Data Access Layer) – цей рівень відповідає за доступ та взаємодію з постійними сховищами даних. Саме він зберігає в собі логіку запитів до баз даних, SQL код, або абстракції над базою даних та сутностей що в ній зберігаються.

Ці рівні є базовими але не являють собою обмеження в їх кількість, що означає те, що в залежності від потреб додатку, кількість шарів може збільшуватись або зменшуватись.

При застосуванні даного підходу є багато плюсів, наприклад:

1. **Безпека:** оскільки клієнт може отримати доступ до рівня даних лише через логічний рівень, це зменшує точку входу та захищає багато небезпечних системних функцій.
2. **Розподілена розробка:** Також завдяки цьому підходу паралельна розробка додатку з іншими розробниками стає більш зручною, через меншу кількість конфліктів при змінах.
3. **Модульність та ізольованість:** за допомогою модульності додатку, ми можемо відокремлювати різні частини додатку в різні рівні, що позбавляє нас змішування логіки відображення, та доступу к даним. Кожен рівень має свою відповідальність, ми в цьому впевненні, і тому на рівні абстракцій легше розуміти структуру проекту.



**Рисунок 1 -
Діаграма додатку з 3-х
шаровою
архітектурою**

4. **Підтримка:** Розподілена структура архітектури сприяє зручності підтримки та масштабованості системи. Кожен є окремим елементом проекту, що зменшує ризики помилок в додатку, та полегшує зміни технічного стеку.

Але все має і свої мінуси, в даному випадку це:

1. **Часом надлишкова абстракція:** при використанні цієї архітектури, можуть виникати випадки, коли логіка переходів між шарами є надлишковою, наприклад коли клієнт просить повернути просто всіх користувачів. В даному випадку нам не потрібна ніяка фільтрація, чи бізнес логіка, але запит все ще має пройти від найвищого до найнижчого рівня, навіть якщо на кожному рівні використовується тільки виклик наступного
2. **Низька швидкість роботи:** з минулого пункту випливає що через те що часто є випадки коли дані просто переходять від шару до шару без логіки, що в свою чергу потребує зайвих ресурсів
3. **Дуплікація коду:** часом для підтримування абстракцій на всіх рівнях може виникнути ситуація в потребі дублювання коду, та не потрібних операцій (наприклад Маппінгу). Але цей пункт можливо легко вирішити загальним рівнем для всіх інших рівнів, який буде в собі утримувати всю загальну логіку (Domain Tier)

Література:

1. Advantages and Disadvantages of Three-Tier Architecture in DBMS - Режим доступу до ресурсу: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-three-tier-architecture-in-dbms/>
2. What is N-Tier Architecture? How It Works, Examples, Tutorials, and More - Режим доступу до ресурсу: <https://stackify.com/n-tier-architecture/>

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБМІНУ НЕПОТРІБНИМИ ПОДАРУНКАМИ

Конох В.В., Золотухіна О.А.

Державного університету телекомунікацій, м. Київ

Обмін непотрібними подарунками стає все більш поширеною практикою, що стала можливою завдяки розвитку програмного забезпечення. Однак, в Україні подібного сервісу не існує, тому розглядаються світові аналоги. Аналіз програмного забезпечення для обміну непотрібними подарунками може допомогти виявити переваги та недоліки наявних рішень і визначити напрямки подальшого розвитку цієї галузі.

Одним із найпопулярніших засобів для обміну непотрібними подарунками є онлайн-платформа Giftster [1]. Giftster забезпечує зручну і просту систему обміну подарунками між друзями і родичами. Платформа

надає можливість створювати список бажань та побажань, співпрацювати зі своїм родинним колом і ділитися інформацією про те, що бажаєте отримати в подарунок. Також Giftster забезпечує функцію анонімного обміну подарунками, що робить процес більш інтригуючим.

Іншим варіантом є програма Elfster [2], яка також дозволяє створювати список бажань та долучати своїх друзів і родичів до обміну подарунками. Elfster дозволяє використовувати необов'язкову опцію анонімного обміну, що дозволяє зберігати інтригу і не розкривати, хто кого отримав у подарунок. Програма також надає можливість відстежувати статус подарунку, щоб переконатися, що всі учасники отримали свій подарунок.

Також серед популярних рішень для обміну непотрібними подарунками є програма Secret Santa Organizer [3]. Ця програма дозволяє створювати обмежений список бажань і обмінюватися подарунками між друзями, колегами або членами родини. Secret Santa Organizer забезпечує анонімний обмін подарунками, що робить процес більш забавним та інтригуючим. Крім того, програма дозволяє керувати списком бажань та обмінюватися повідомленнями з іншими учасниками.

Крім цього, існує багато інших рішень для обміну непотрібними подарунками, таких як сайти та додатки для мобільних пристроїв, що дозволяють створювати списки бажань та обмінюватися подарунками. Незалежно від обраного рішення, обмін подарунками може стати цікавим та забавним досвідом для всіх учасників.

У результаті аналізу було виявлено, що багато програм мають спільні функціональні можливості, такі як можливість створення списків бажань, розсилка запрошень до обміну, анонімність обміну і т.д. Однак, деякі програми мають інші додаткові функції, такі як можливість створення груп для обміну і використання групових чатів, можливість обміну подарунками між країнами і т.д.

Література:

1. Giftster. (2022). [Електронний ресурс]. – Режим доступу: <https://www.giftster.com/>
2. Elfster. (2022). [Електронний ресурс]. – Режим доступу: <https://www.elfster.com/>
3. Secret Santa Organizer. (2022). [Електронний ресурс]. – Режим доступу: <https://www.secretsantaorganizer.com/>
4. Krunic, M. (2020). 15 Best Gift Exchange Apps and Websites. [Електронний ресурс]. – Режим доступу: <https://www.techjunkie.com/gift-exchange-apps-websites/>

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОНТРОЛЮ ЗДОРОВ'Я ДОМАШНІХ ТВАРИН МОВОЮ C#

Смирнова К.Б., Негоденко О.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі

Наразі домашні улюбленці стали невід'ємною частиною сім'ї, тому питання здоров'я та правильного раціону стало більш актуальним. Такого типу додатку дають змогу записувати все, що торкається життя тварини та контролювати це.

Мета дослідження

Метою даного дослідження є розробка програмного забезпечення для контролю здоров'я домашніх тварин, яке зможе відслідковувати медичні записи, графіки візитів до ветеринара, інформацію про препарати та інші важливі дані, що допоможуть власникам тварин забезпечити тварин.

Результати дослідження

Під час дослідження було розроблено програмне забезпечення на мові C#[1], із використанням технології Xamarin [2]. Додаток дозволяє власникам тварин вести детальний медичний журнал із допомогою бази даних SQLite [3] та власних алгоритмів. Було розроблено гнучке та швидке рішення автоматизації ведення картотеки медичних даних котрі дозволяють поліпшувати медичний стан тварин, ці дані мають інформацію про відвідування ветеринарів, їх рекомендації та назначення, історію хвороб та стан здоров'я тварин. Також про прийом ліків, їжі та іншого. Ця програма допомагає власникам тварин отримати повну картину здоров'я своєї домашньої тварини та дозволяє вчасно реагувати на будь-які зміни в ньому. А також налаштовувати застосунок за для профілактики лікування здоров'я хатніх улюбленців.

Висновки та перспективи

Це програмне забезпечення може у перспективі автоматизувати процеси замовлення медичних послуг, замовлення їжі, ліків, інших речей для тварин із допомогою сервісів автоматизованого пошуку та замовлення найліпших варіантів необхідних речей у найпоширеніших зоомагазинах та ветклініках по усій країні та навіть більше, що збереже час власників тварин та зробить процес більш ефективним та різнобічним у виборі тих чи інших джерел продуктів та ліків.

Література:

1. C# documentation [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/>
2. Xamarin documentation [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/xamarin/>
3. SQLite documentation [Електронний ресурс] – Режим доступу: <https://www.sqlite.org/docs.html>

МІКРОСЕРВІСНА АРХІТЕКТУРА ПРИ РОЗРОБЦІ ТЕЛЕГРАМ-БОТІВ

Скидан П.В., Негоденко О.В.

Державного університету телекомунікацій, м. Київ

Мета дослідження

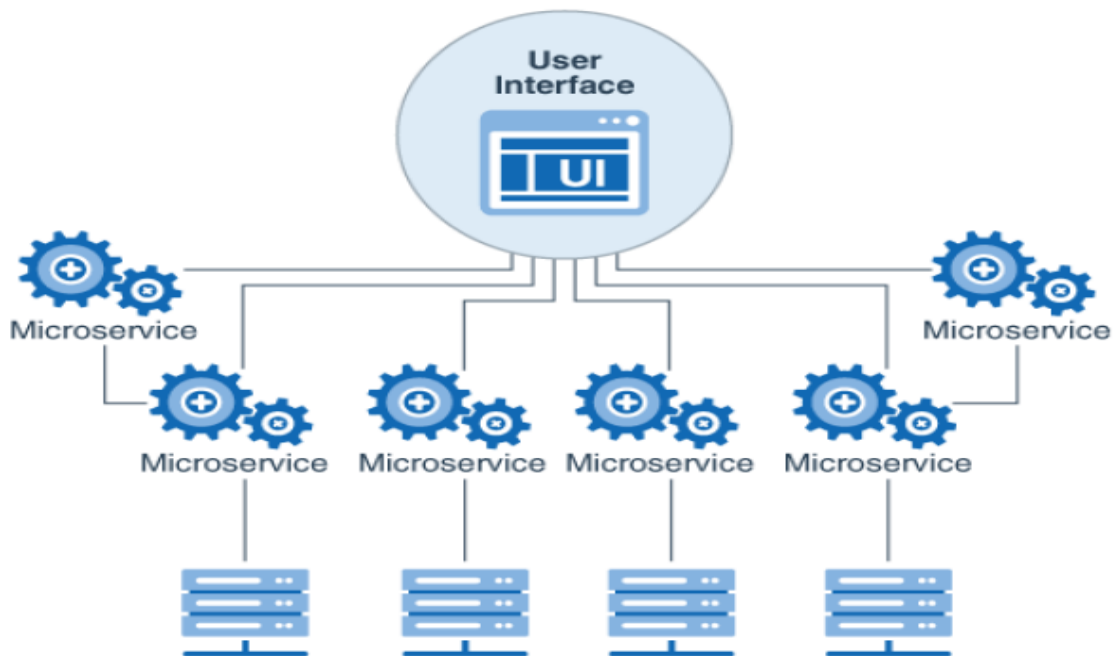
Головна мета застосування мікросервісної архітектури, це навчитися відокремлювати компоненти проєкту один від одного задля забезпечення гнучкого налаштування, розширення та вдосконалення.

Результати дослідження

Спроектвана мікросервісна архітектура, яка лежатиме в основі телеграм-боту та забезпечить зручність в написанні та налаштуванні.

Висновки та перспективи

Microservice Architecture



Мал.1 Схематичне зображення МСА

Далі наведені головні переваги такої архітектури:

1. Додавання або видалення сервісів можливе без впливу на інші сервіси, що дозволяє легко масштабувати додаток.
2. Гнучкість: легко реалізувати нові функції або змінювати існуючі без необхідності зміни всього додатку.
3. Швидкість розгортання: розгортання нових функцій можливе без необхідності розгортання всього додатку, що прискорює процес розгортання.
4. Легкість супроводу: кожен сервіс можна супроводжувати окремо, що полегшує роботу з ним.

Перехід на таку архітектуру може призвести до появи нових проблем. Але вони будуть тимчасовими, але потім таке рішення дозволить швидко та легко розробляти новий функціонал.

Література:

1. Мікросервісна архітектура: плюси та мінуси [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://blog.iteducenter.ua/articles/microservices-architecture-advantages-and-disadvantages/>.
2. Монолитная vs Микросервисная архитектура [Електронний ресурс] – Режим доступу до ресурсу: <https://proglib.io/p/monolitnaya-vs-mikroservisnaya-arhitektura-2019-09-16>.

ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ТА ЗАБЕЗПЕЧЕННЯ БІЛЬШ ДОВГОГО ЖИТТЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кужентський О.Г., Лемешко А.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Задача полягає в тому, щоб розібрати методи для підвищення продуктивності та забезпечення більш довгого життя програмного забезпечення. Для досягнення подовження життя програмного забезпечення необхідно виконувати регулярні огляди коду, використовувати кращі практики програмування, тестувати програмне забезпечення на різних рівнях та використовувати автоматичні тести.

Мета дослідження. Метою дослідження є виявлення методів для підвищення продуктивності та забезпечення більш довгого життя програмного забезпечення.

Результати дослідження. Підвищення продуктивності та забезпечення більш довгого життя програмного забезпечення - це важливі завдання для будь-якої компанії, яка займається розробкою програмного забезпечення. Для досягнення цих цілей необхідно використовувати кращі практики програмування, такі як SOLID, проводити регулярні огляди коду, тестувати програмне забезпечення на різних рівнях та використовувати автоматичні тести. Також необхідно використовувати системи контролю версій та управління конфігурацією, такі як Git, SVN та JIRA, щоб забезпечити ефективне управління конфігурацією та підтримку програмного забезпечення. Використання цих стратегій та методів дозволить знизити кількість помилок, зменшити час на розробку та підтримку програмного забезпечення, покращити якість коду та забезпечити більш довге життя програмного забезпечення.

Висновки та перспективи. Отже, підвищення продуктивності та забезпечення більш довгого життя програмного забезпечення є ключовими завданнями для будь-якої компанії, що займається розробкою програмного забезпечення. Застосування кращих практик програмування, таких як SOLID, тестування на різних рівнях та використання систем контролю версій та управління конфігурацією допоможуть досягти цих цілей. Правильне застосування цих стратегій та методів дозволить забезпечити покращення якості програмного забезпечення, зниження витрат на його розробку та підтримку, а також збільшення терміну його життя.

Література:

1. ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧАСТИНА 1. ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ [Електронний ресурс] – Режим доступу до ресурсу:
https://ela.kpi.ua/bitstream/123456789/50623/1/TRPZ_Ch1_ZhTsPZ.pdf.
2. Основи проектування та розробки інформаційних систем [Електронний ресурс] – Режим доступу до ресурсу:
https://learn.ztu.edu.ua/pluginfile.php/23193/mod_resource/content/1/PM-Lectures.pdf.

ВИВЧЕННЯ МЕТОДІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЇХ ВИКОРИСТАННЯ В ПРОЦЕСІ РОЗРОБКИ ПЗ

Іванов О. П.

Державного університету телекомунікацій, м. Київ

Швидкий розвиток технологій та постійне зростання вимог до якості програмного забезпечення ставлять перед розробниками ПЗ складні завдання забезпечення надійності та безпеки продукту. Один із ключових етапів процесу розробки ПЗ - це тестування, яке дозволяє виявити помилки та дефекти, забезпечити відповідність функціональних та технічних вимог, а також підвищити якість та ефективність продукту. У зв'язку з цим, поставлено задачу вивчити та використовувати ефективні методи тестування ПЗ в процесі його розробки.

Головною метою дослідження є вивчення та аналіз ефективних методів тестування програмного забезпечення та їх використання в процесі розробки ПЗ для забезпечення високої якості та надійності продукту. В рамках дослідження будуть досліджені різні методи тестування, їх переваги та

недоліки, а також можливості їх використання в залежності від типу програмного забезпечення.

Під час дослідження було вивчено та аналізовано різні методи тестування програмного забезпечення, зокрема, функціональне, навантажувальне, регресійне, модульне та інші. Було проведено детальний аналіз їх переваг та недоліків, визначено сферу застосування та можливості їх використання в різних видах ПЗ. Було виявлено, що кожен метод має свої особливості, але в комбінації з використанням декількох методів тестування можна досягти більш високої якості ПЗ та знизити ризик виявлення помилок після випуску продукту на ринок.

Також було досліджено інструменти тестування, які можуть бути використані для автоматизації процесу тестування та зменшення часу, необхідного для проведення тестів. Було встановлено, що використання автоматизованих інструментів дозволяє не тільки збільшити ефективність тестування, але й підвищити точність результатів тестування та скоротити час на виявлення та виправлення помилок.

На основі дослідження можна зробити висновок, що ефективне тестування програмного забезпечення є необхідною умовою забезпечення високої якості та надійності продукту. Для досягнення цієї мети рекомендується використовувати різні методи тестування та автоматизовані інструменти для зменшення часу, необхідного для проведення тестів та підвищення точності результатів. Також важливо проводити тестування на ранніх етапах розробки ПЗ для зменшення витрат на виявлення та виправлення помилок у подальшому.

У майбутньому можливо дослідити більш детально конкретні методи тестування та їх вплив на якість та надійність ПЗ, а також розглянути можливість використання штучного інтелекту та машинного навчання для автоматизації процесу тестування та забезпечення високої якості продукту.

Література:

1. https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення
2. <https://sqa.lviv.ua/yaki-ye-typy-testuvannya>

РОЗРОБКА ІТ-РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ГОДУВАННЯ ТВАРИН НА ОСНОВІ СИСТЕМИ BARF

Місіна Н.В., Яскевич В.О.

Державного університету телекомунікацій, м. Київ

Дослідження доцільності проекту для оптимізації процесу годування тварин на основі системи BARF з метою забезпечення користувачам простого та зручного інтерфейсу, можливості розрахунку доз годівлі та відстеження статистики годування, а також автоматизації процесу підготовки раціону.

Мета цього дослідження - полягає в розробці та оптимізації процесу

годування тварин на основі системи BARF з використанням функціональних рішень для оптимізації процесу. Таким чином, метою цього дослідження є покращення процесу годування тварин та забезпечення оптимальних умов їх утримання з використанням сучасних технологій.

Система годування BARF є корисною для домашніх тварин, оскільки базується на природному харчуванні. Вона забезпечує тваринам всі необхідні живильні речовини, підвищує імунітет та зменшує ризик розвитку алергій та хвороб шкіри. Система BARF також дозволяє власникам контролювати раціон своєї тварини та уникати додавання штучних консервантів та барвників. Враховуючи ці переваги, BARF може бути ефективною альтернативою традиційному комерційному корму для домашніх тварин. В дослідженні, опублікованому в журналі "Journal of Animal Physiology and Animal Nutrition", встановлено, що тварини, які отримують раціон, більш схожий на той, що мали їх предки у дикій природі, мають кращу загальну імунну систему та здоров'я.

Під час дослідження було розроблено IT-рішення для оптимізації процесу годування тварин на основі системи BARF. Дослідження виявило, що ручне планування та годування тварин неефективне, оскільки може призводити до незбалансованого харчування та здоров'я тварин. Розроблене рішення дозволяє автоматизувати процес годування тварин з використанням системи BARF, що забезпечує оптимальний раціон для тварин.

Архітектура програмного продукту складається з декількох модулів. Першим модулем є модуль внесення даних, який дозволяє вносити інформацію про тварин та їх потреби в харчуванні з отриманням порад. Другим модулем є модуль планування, який автоматично створює графік годування на основі інформації про тварин та їх потреби в харчуванні. Третім модулем є модуль годування, який дозволяє здійснювати годування тварин за створеним графіком.

Ролі користувачів такої системи можуть бути різними: власник тварини, який додає свого улюбленця в систему та налаштовує дієту, ветеринар, який може переглядати історію харчування тварини та вносити корективи у дієту. Система може мати наступні користувацькі історії:

- Власник тварини може додати тварину в систему та налаштувати дієту.
- Ветеринар може переглянути історію харчування тварини та внести корективи у дієту.

Архітектура програмного забезпечення може бути побудована на основі моделі клієнт-сервер, де клієнтська частина буде веб-додатком, розробленим з використанням фреймворку React, а серверна частина - Node.js з фреймворком Express, що забезпечить роботу та зв'язок з базою даних MongoDB.

Сценарії використання розробленого IT-рішення включають імпорт даних про тварин, створення графіка годування та автоматичне годування тварин. Крім того, система дозволяє здійснювати моніторинг успішності плану харчування відповідно до ваги та стану улюбленця, а також налаштувати параметри

системи відповідно до потреб користувачів.

Дослідження показало, що розроблене ІТ-рішення для оптимізації процесу годування тварин на основі системи BARF є ефективним та практичним. Оскільки система годування BARF потребує дотримання деяких особливостей та врахування індивідуальних потреб тварин, програмне забезпечення може забезпечити власникам більш точний та ефективний підхід до годування їхніх улюбленців. Використання автоматизованої системи дозволяє зберегти час та зусилля, забезпечити оптимальний результат.

Література:

1. Raw feeding. [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Raw_feeding
2. BARF Diet: How to Make a Complete and Balanced Meal for Your Dog" by Adrienne Farricelli on PetHelpful [Електронний ресурс]. – Режим доступу: <https://pethelpful.com/dogs/Barf-Diet-How-to-Make-a-Complete-and-Balanced-Meal-for-Your-Dog>
3. "The Benefits and Risks of a Raw Diet for Your Dog". [Електронний ресурс]. – Режим доступу: <https://www.akc.org/expert-advice/nutrition/raw-diet-for-dogs/>
4. "A Veterinarian's Take on the BARF Diet" by Dr. Jennifer Coates on PetMD. [Електронний ресурс]. – Режим доступу: https://www.petmd.com/dog/nutrition/evr_multi_veterinarians-take-barf-diet

АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ТА ІНСТРУМЕНТІВ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІКТ

Студент Б.М., Негоденко О.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Завдання полягає у дослідженні та аналізі використання автоматизованого тестування програмного забезпечення в ІКТ-індустрії. Основна мета - визначити ефективність використання автоматизованого тестування ПЗ у порівнянні з ручним тестуванням, виявити переваги та недоліки використання автоматизованого тестування, а також розглянути оптимальні підходи до його використання для забезпечення якості ПЗ в ІКТ. Дослідження також має на меті дослідити сучасний стан технологій автоматизованого тестування, їх можливості та обмеження, а також визначити можливість використання інтелектуальних методів для автоматизованого тестування програмного забезпечення в ІКТ.

Мета дослідження. Аналіз та дослідження застосування автоматизованого тестування програмного забезпечення в галузі ІКТ та визначення переваг та недоліків цього методу тестування. Дослідження

має на меті також визначення найкращих практик використання автоматизованого тестування та інструментів для ефективного планування тестових сценаріїв.

Результати дослідження: На даний момент ІКТ використовується в практично в усіх галузях, тому тестування програмного забезпечення є одним із найважливіших етапів розробки програмного продукту. Автоматизація тестування дозволяє підвищити ефективність та точність тестування, а також зменшити витрати на тестування в порівнянні з ручним тестуванням.

Застосування автоматизованого тестування програмного забезпечення у галузі ІКТ є надзвичайно актуальною темою, оскільки дозволяє досягти якісного та швидкого тестування програмного продукту, що є ключовим фактором для його успішної розробки та належної роботи.

Автоматизоване тестування програмного забезпечення у галузі ІКТ є важливим етапом в розробці програмного продукту, що дозволяє підвищити якість продукту та зменшити витрати на тестування. Цей підхід може бути застосований для тестування різних типів програмного забезпечення, включаючи веб-сайти, мобільні додатки, Desktop програми та інші.

Для веб-сайтів, наприклад, автоматизоване тестування може перевіряти функціональність різних елементів, таких як форми для введення даних, кнопки, меню, посилання та інші. Тестування може також охоплювати різні браузері та пристрої, щоб переконатися, що веб-сайт працює належним чином в різних умовах.

Для мобільних додатків автоматизоване тестування може перевірити функціональність додатку, його взаємодію з різними компонентами пристрою, такими як камера, мікрофон, GPS та інші. Тестування може також охоплювати різні версії операційних систем та різні моделі пристроїв, щоб переконатися, що додаток працює належним чином на різних пристроях.

Для Desktop програм автоматизоване тестування може перевірити функціональність програми, її взаємодію з різними компонентами комп'ютера, такими як монітор, клавіатура, миша та інші. Тестування може також охоплювати різні операційні системи, щоб переконатися, що програма працює належним чином на різних платформах.

Фреймворки для автоматизованого тестування, такі як Selenium, Appium та TestComplete, дозволяють автоматизувати взаємодію з програмою та перевірити її роботу в різних умовах. Наприклад, Selenium дозволяє записувати тестові сценарії, щоб перевірити роботу веб-сайту в різних браузерах та на різних пристроях. Appium дозволяє автоматизувати тестування мобільних додатків на різних платформах, включаючи Android та iOS. TestComplete надає зручний графічний інтерфейс для створення скриптів тестування. Крім того, скрипти можна створювати на різних мовах програмування, таких як JavaScript, Python,

VBScript, або C++Script. Це дозволяє тестувальникам використовувати мову, яка є для них найбільш зручною та зрозумілою.

Для досягнення максимальної ефективності автоматизованого тестування необхідно використовувати не тільки тестові фреймворки, але й інші інструменти, такі як інтегровані середовища розробки (IDE), системи контролю версій, засоби автоматизованої збірки (CI/CD) та інші.

Однією з переваг автоматизованого тестування є можливість тестування програмного продукту в різних умовах та швидкість виконання тестів. Завдяки автоматизації тестування можна перевірити роботу програмного продукту при різних навантаженнях, що дозволяє виявляти помилки та усувати їх в найкоротші терміни.

Іншою перевагою автоматизованого тестування є можливість використання тестування на ранніх етапах розробки програмного продукту, що дозволяє виявляти помилки та усувати їх на початкових етапах розробки. Це знижує витрати на тестування та забезпечує високу якість програмного продукту.

Також важливою перевагою є можливість автоматизованого тестування є збільшення продуктивності процесу розробки програмного забезпечення. Це досягається завдяки швидкості виконання тестів та автоматизованому процесу їх виконання.

Отже, використання автоматизованого тестування є ключовим етапом в розробці програмного забезпечення у галузі ІКТ. Це дає можливість підвищити якість та швидкість розробки програмного продукту та зменшити витрати на тестування. Використання автоматизованого тестування є невід'ємною частиною будь-якого проекту з розробки програмного забезпечення та дозволяє підвищити ефективність та точність тестування.

Висновки та перспективи. Використання автоматизованого тестування програмного забезпечення у галузі ІКТ, є необхідним етапом розробки програмного продукту. Це дозволяє забезпечити високу якість продукту, зменшити витрати на тестування та підвищити швидкість реагування на зміни, що забезпечує конкурентоспроможність продукту на ринку.

Автоматизоване тестування має декілька переваг перед ручним тестуванням. Наприклад, воно дозволяє виконувати тести при різних навантаженнях, що допомагає виявляти помилки та усувати їх найшвидше. Крім того, автоматизоване тестування можна використовувати на ранніх етапах розробки програмного продукту, що дозволяє виявляти помилки та усувати їх на початкових етапах розробки. Це знижує витрати на тестування та забезпечує високу якість програмного продукту.

Також важливою перевагою автоматизованого тестування є збільшення продуктивності процесу розробки програмного забезпечення. Це досягається завдяки швидкості виконання тестів та автоматизованому процесу їх виконання.

Щоб досягти максимальної ефективності використання автоматизованого тестування, необхідно правильно вибрати інструменти тестування та розробляти тестові скрипти з урахуванням специфіки тестування в галузі ІКТ. Слід враховувати особливості тестування інтернет-застосунків, мобільних додатків та інших продуктів, щоб забезпечити високу якість тестування та підвищити ефективність процесу розробки програмного забезпечення.

Література:

1. Qiu, T., Zuo, L., & He, P. (2020). An approach to automated software testing based on ontology modeling. In 2020 IEEE International Conference on Knowledge Innovation and Invention (ICKII) (pp. 483-487). IEEE.
2. Xiong, Y., Chen, T. Y., & Towey, D. (2019). Software testing automation for big data: A systematic mapping study. *Journal of Systems and Software*, 157, 110392.
3. Yang, G., Ma, L., & Zheng, W. (2020). Automated software testing for artificial intelligence applications: A systematic mapping study. *Journal of Systems and Software*, 169, 110711.
4. Zapata, R., & Fraser, G. (2018). Automated software testing using open source testing tools. In 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 756-760). IEEE.

РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ДОПОМОГИ СЕРВІСНИМ ЦЕНТРАМ

Кутняк М.Ю., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

Сучасна електроніка досить часто виходить з ладу та купляти новий пристрій не завжди є можливість та сенс. Іноді причиною виходу з ладу є не дуже дорогий компонент який можна відремонтувати значно дешевше аніж купляти повністю новий ноутбук чи відеокарту. Тому сучасні сервісні центри є досить затребуваними. Але в неофіційних сервісних центрах або майстри які працюють на дому за часту не мають доступу до офіційних матеріалів щодо ремонту тієї чи іншої електроніки.

На сьогоднішній день існує проблема з пошуком інформації щодо ремонту відео карт та ноутбуків. Якщо ти не працюєш в авторизованому сервісному центрі, то тобі постійно треба шукати інформацію на безлічі інтернет форумах та сайтах. Наразі більшість компаній що випускає електроніку не викладає в загальний доступ інформацію щодо ремонтів їх продуктів, тому при ремонті якогось девайса необхідно шукати інформацію від іншого майстра який вже зіткнувся зі схожою проблемою та знайшов її рішення попутно перемірявши пів плати і на пошук цієї інформації не рідко

витрачається більше часу чим на діагностику та ремонт самого пристрою.

Тому виникає необхідність в розробці сайту де більшість всієї інформації була би в доступі. Завдяки такому веб-додатку час що витрачається на пошук біоса під вашу відео карту, чи результат замірів супротиву та вольтажу на робочій платі може скоротитися істотно.

Тож важливими критеріями завдання для розробки подібного ресурса повинні стати:

- Зручність пошуку інформації
- Швидкість пошуку інформації
- Доступність для будь-яких майстрів
- Велика кількість інформації

Адже інакше подібний ресурс не буде мати сенсу.

Література:

1. Інформаційний пошук . [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Інформаційний_пошук
2. Схеми на материнські плати Asus . [Електронний ресурс]. – Режим доступу:
https://sector.biz.ua/docs/schematic_diagrams_asus_motherboards/schematic_diagrams_asus_motherboards.phtml#.ZD_QHnZBYUk
3. Схеми на материнські плати Msi . [Електронний ресурс]. – Режим доступу:
https://sector.biz.ua/docs/schematic_diagrams_msi_motherboards/schematic_diagrams_msi_motherboards.phtml#.ZD_QiXZBYUk
4. Схеми на материнські плати Gigabyte . [Електронний ресурс]. – Режим доступу:
https://sector.biz.ua/docs/schematic_diagrams_gigabyte_motherboards/schematic_diagrams_gigabyte_motherboards.phtml#.ZD_Qq3ZBYUk

ФРЕЙМВОРКИ REACT ТА ANGULAR

Жовніренко Є.В., Яскевич В.О.

Державний університет телекомунікацій, м.Київ

Фронтенд-розробка стає все більш важливою в сучасній веб-розробці, оскільки існує велика кількість фреймворків і бібліотек які використовуються розробниками для розробки інтерфейсної частини програмного забезпечення.

Двома найпопулярнішими фреймворками для розробки інтерфейсів є React та Angular. Хоча обидва фреймворки служать однаковим цілям, вони мають явні відмінності з погляду філософії дизайну, функцій та продуктивності.

React - це бібліотека JavaScript, розроблена Facebook, яка фокусується на створенні багаторазово використовуваних компонентів інтерфейсу

користувача. Основною концепцією React є компонент, що являє собою невелику автономну одиницю коду, яка є частиною інтерфейсу користувача. React використовує віртуальну DOM (об'єктну модель документа) для ефективного оновлення інтерфейсу користувача, що призводить до більш швидкої роботи додатків. React також наголошує на декларативному стилі програмування, який дозволяє розробникам описувати те, що вони хочуть, а не те, як це зробити.

Angular, з іншого боку, є повноцінним фреймворком для створення веб-додатків. Розроблений Google Angular надає повний набір інструментів для створення складних веб-додатків. Angular використовує двосторонній підхід до прив'язування даних, що означає, що зміни моделі автоматично оновлюють подання і навпаки. Angular також надає такі функції, як використання залежностей, перевірка форми та інтернаціоналізація, що робить його потужним інструментом для створення складних додатків.

Отже, який фреймворк вибрати і коли? Відповідь багато в чому залежить від конкретних вимог вашого проекту та переваг вашої команди розробників. Ось деякі фактори, які слід враховувати:

- Складність проекту: якщо ви створюєте простий додаток з кількома компонентами, React може бути найкращим вибором. Однак для складніших додатків з великою кількістю рухомих частин комплексний набір інструментів Angular може виявитися більш підходящим;
- Продуктивність: React використовує віртуальний DOM і наголошує на продуктивності, що робить його відмінним вибором для додатків, що вимагають високої продуктивності. Двостороння прив'язка даних Angular іноді може призвести до проблем із продуктивністю, особливо у великих додатках;
- Зосередженість React на невеликих повторно використовуваних компонентах та використання декларативного стилю програмування роблять його легшим у вивченні, ніж Angular. Проте набір інструментів які має Angular може бути приголомшливим для розробників-початківців;
- Підтримка спільноти: і React, і Angular мають великі та активні спільноти з великою кількістю доступних ресурсів та підтримки. Проте спільнота React зазвичай вважається яскравішою та активнішою;
- Якщо вам потрібна краща інтеграція з іншими технологіями, такими як Redux або React Router, то краще вибрати React. У разі використання технологій Google, наприклад Google Cloud Platform або Jasmine, Angular є кращим варіантом. Обидва фреймворки підтримують мови програмування JavaScript та TypeScript.

React, і Angular є потужними фреймворками для розробки інтерфейсу, які мають свої сильні і слабкі сторони. Вибір між ними залежить від конкретних вимог вашого проекту, переваг вашої команди розробників і

технологій, які ви вже використовуєте. Враховуючи ці фактори та зважуючи плюси та мінуси кожного фреймворку, ви можете прийняти зважене рішення про те, який фреймворк вибрати для свого проекту.

Література:

1. Developer Mozilla [Електронний ресурс]. – Режим доступу : https://developer.mozilla.org/en-US/docs/Web/Guide/Introduction_to_Web_development
2. Eleken [Електронний ресурс]. – Режим доступу : <https://www.eleken.co/blog-posts/ui-ux-designer-vs-front-end-developer>
3. Developers Google [Електронний ресурс]. – Режим доступу : <https://developers.google.com/web/updates/2018/08/using-workers-to-create-a-multi-threaded-web>

РОЗРОБКА WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ МОВОЮ C# НА ПЛАТФОРМІ ASP.NET CORE ТА BLAZOR

Чернявський Ж.А., Гаманюк І.М.
Державний університет телекомунікацій, м.Київ

Постановка задачі

Метою даного проекту є розробка web-додатку для автоматизації складського обліку на основі мови програмування C# та технологій ASP.NET Core та Blazor.

Мета дослідження

Метою дослідження є розробка web-додатку для складського обліку мовою C# на платформі ASP.NET Core та Blazor, який дозволить автоматизувати процеси складського обліку, спростити ведення документації, збереження та обробку даних про запаси товарів на складі та оптимізувати роботу складського працівника.

Результати дослідження

Розроблений web-додаток для складського обліку мовою C# на платформі ASP.NET Core та Blazor, який має наступні функціональні можливості:

- додавання/редагування/видалення товарів на складі;
- створення замовлень;
- формування звітів та статистики.
- Розроблена база даних для додатку, яка забезпечує зберігання та обробку інформації про запаси товарів на складі.
- Проведене тестування додатку та виправлення помилок.
- Забезпечена безпека даних та збереження конфіденційності інформації про запаси товарів на складі.
- Проведена експертиза розробленого додатку та оцінено його ефективність в роботі зі складськими запасами.
- Отримані рекомендації щодо вдосконалення додатку та можливостей його застосування в роботі зі складським обліком.

Висновки та перспективи

Отже, на основі розробленого web-додатку для складського обліку мовою C# на платформі ASP.NET Core та Blazor можна зробити висновок, що він є ефективним інструментом для автоматизації процесу ведення обліку запасів на складі. База даних, створена для додатку, забезпечує зручний та швидкий доступ до інформації про запаси товарів на складі. Простий та зрозумілий інтерфейс додатку дозволяє користувачам легко використовувати його без спеціальних навичок. Таким чином, в результаті дослідження можна стверджувати, що розроблений web-додаток успішно виконує поставлені перед ним завдання, забезпечуючи зручний та ефективний процес ведення обліку запасів на складі. Щодо перспектив, можна зазначити, що додаток можна подальшим розвитком розширити функціоналом та додати нові можливості, що зробить його ще більш корисним та вигідним для підприємств, що займаються складською діяльністю.

Література:

- .NET | Build. Test. Deploy. *Microsoft*. URL: <https://dotnet.microsoft.com/en-us/> (date of access: 18.04.2023).
- ASP.NET Core Blazor. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/ru-ru/aspnet/core/blazor/?view=aspnetcore-7.0> (date of access: 18.04.2023).

РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ХАРЧУВАННЯ ДОМАШНІХ УЛЮБЛЕНЦІВ ПО СИСТЕМІ BARF

Місіна Н.В., Яскевич В.О.

Державний університет телекомунікацій, м.Київ

Створення веб-сервісу для планування харчування домашніх тварин на основі системи BARF має бути дуже доцільним і корисним кроком для тварин і їх господарів. Система BARF (Biologically Appropriate Raw Food) базується на природному харчуванні тварин і передбачає складання раціону з сировини натурального походження.

Веб-сервіс для планування харчування на основі системи BARF буде допомагати господарям визначати оптимальний склад раціону для своїх тварин на основі їх віку, статі, розміру та інших факторів. Крім того, такий сервіс забезпечить доступ до інформації про продукти та їх склад, поради щодо зберігання та приготування їжі, а також функції відстеження раціону та його належного виконання.

У результаті, веб-сервіс для планування харчування домашніх тварин на основі системи BARF може мати значний потенціал для поліпшення якості життя тварин та задоволення потреб господарів в контролі якості та складу

раціону. Продуманий та ретельно розроблений сервіс стане цінним інструментом у підтримці здоров'я та довголіття домашніх улюбленців, а також допоможе оптимізувати цей процес задля мінімізації помилок.

Існує кілька потенційних плюсів створення веб-сервісу для планування харчування домашніх тварин на основі системи BARF для сфери IT. По-перше, розробка такого сервісу може сприяти розвитку технологій інтернету речей, оскільки користувачі зможуть отримувати рекомендації щодо харчування своїх домашніх тварин.

По-друге, створення такого сервісу може забезпечити можливість збору та аналізу даних про харчування домашніх тварин на основі системи BARF, що дозволить розробникам покращувати та удосконалювати саму систему, а також розробляти нові продукти та послуги на основі цієї інформації.

Ролі та відповідні користувачькі історії для системи планування харчування домашніх тварин на основі системи BARF будуть наступними
Господар:

1. Хоче зареєструватися в системі та додати своїх тварин.
2. Хоче встановити індивідуальний план харчування для своїх тварин на основі системи BARF.
3. Хоче знайти інформацію про продукти та інгредієнти, які можна використовувати при плануванні харчування для своїх тварин.
4. Хоче знайти рецепти та поради від інших користувачів системи.

Ветеринар:

1. Хоче дізнатися більше про систему BARF та її застосування для домашніх тварин.
2. Хоче знайти інформацію про продукти та інгредієнти, які можуть бути корисними для харчування домашніх тварин на основі системи BARF.
3. Хоче дізнатися про індивідуальні плани харчування для тварин, які лікує він.

Згідно з користувачькими історіями вимоги до системи будуть складатися з :

1. Реєстрація користувача та можливість додавання своїх тварин.
2. Можливість створення індивідуального плану харчування на основі системи BARF для кожної тварини.
3. База даних продуктів та інгредієнтів, що можна використовувати при плануванні харчування.
4. Можливість пошуку рецептів та порад від інших користувачів системи.
5. Інформація про систему BARF та її застосування для домашніх тварин.

Система матиме архітектуру:

1. Клієнтська частина: веб-інтерфейс для користувачів, що дозволяє створювати облікові записи, вводити дані про своїх тварин та отримувати поради щодо планування харчування на основі системи BARF. Для розробки клієнтської частини будуть використані HTML, CSS, JavaScript та фреймворк React.
2. База даних: система зберігання та управління даними, яка містить інформацію про користувачів, їх тварин та плани харчування на основі системи BARF. Для реалізації бази даних може буде використана

реляційна база даних MySQL.

Дослідження показало, що розроблене ІТ-рішення для оптимізації процесу годування тварин на основі системи BARF є ефективним та практичним. Використання автоматизованої системи дозволяє зберегти час та зусилля, забезпечити оптимальний результат.

Література:

1. Raw feeding. [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Raw_feeding
2. BARF Diet: How to Make a Complete and Balanced Meal for Your Dog" by Adrienne Farricelli on PetHelpful [Електронний ресурс]. – Режим доступу: <https://pethelpful.com/dogs/Barf-Diet-How-to-Make-a-Complete-and-Balanced-Meal-for-Your-Dog>
3. "The Benefits and Risks of a Raw Diet for Your Dog". [Електронний ресурс]. – Режим доступу: <https://www.akc.org/expert-advice/nutrition/raw-diet-for-dogs/>

АКТУАЛЬНІСТЬ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ПОШУКУ ПОПУТНОГО ТРАНСПОРТУ ДЛЯ ПЕРЕДАЧІ ПОСИЛОК

Горбань А.М., Золотухіна О.А.

Державного університету телекомунікацій, м. Київ

Сучасний світ зі стрімким розвитком технологій, електронної комерції та зростанням міграції все більше залежить від ефективної логістики та способів доставки посилок.

Також військові дії в Україні призводять до серйозних перешкод у забезпеченні стабільної та ефективної системи доставки посилок. Тому попутний транспорт набуває актуальності, особливо для передачі гуманітарних вантажів.

Традиційні способи доставки стикаються з рядом проблем, таких як високі витрати, повільність та негативний вплив на навколишнє середовище. У зв'язку з цим, автоматизація процесу пошуку попутного транспорту для передачі посилок набуває особливої актуальності.

Результати досліджень:

Передача посилок попутним транспортом це краудсорсингова модель використання транспорту, з наступними перевагами:

- швидкість обробки запиту на доставку;
- низька вартість доставки для замовника;
- зниження вартості поїздки або доставки основного вантажу за рахунок використання вільного місця;
- зниження ризиків відсутності наявного транспорту;

– зменшений вплив на екологію за рахунок спільного та ефективного використання транспорту.

Завдання пошуку попутного транспорту може бути вирішено за допомогою різноманітних методів, серед яких методи, що базуються на використанні інформаційних технологій, такі як розміщення дописів у соціальних мережах або на спеціалізованих веб-сайтах оголошень (наприклад, OLX). Також існують спеціалізовані сервіси, призначені для пошуку і спільного використання попутного транспорту, де водії та відправники можуть розміщувати свої заявки, а сервіс автоматично їх сполучає. Кожен з цих інструментів пошуку попутного транспорту має певні переваги та недоліки, зокрема відсутність автоматизації або обмеженість учасників за територіальним принципом, типом транспорту та розмірами вантажів.

Актуальним є розробка сервісу, який автоматизує цей процес та забезпечує доступність для участі різних користувачів з різними потребами. Автоматизація процесу пошуку попутного транспорту для передачі посилок сприятиме поліпшенню пошуку, залученню більшої кількості учасників, підвищенню ефективності та надійності процесу завдяки рейтингам, відгукам та верифікації користувачів, а також забезпечить доступність пошуку через локалізацію та стандартизацію маршрутів та замовлень.

Вирішення проблеми за допомогою автоматизованого сервісу може бути розділено на такі основні частини:

- створення і публікація заявок від відправника та перевізника;
- підбір варіантів маршрутів
- вибір переможця (поєднання замовника і виконавця)
- інформаційний супровід та відслідковування доставки
- завершення доставки, передача та оцінка результату у системі рейтингу.

Цільовою аудиторією системи автоматизованого пошуку попутного транспорту можуть бути різні категорії користувачів, включаючи індивідуальних користувачів, малі підприємства та волонтерські групи.

Автоматизація пошуку попутного транспорту для передачі посилок актуальна в сучасному світі, оскільки вона вирішує проблеми традиційних методів доставки, забезпечує економічні, екологічні та соціальні переваги, і сприяє розвитку технологічних інновацій. Веб-додаток для пошуку попутного транспорту створить зручний та надійний інструмент, підвищить ефективність, оптимізує ресурси та сприяє співпраці між людьми. Інтеграція з ШІ та ІоТ може забезпечити ще більшу ефективність, безпеку та зручність для всіх учасників доставки.

Література:

1. Qi, Wei & Li, Lefei & Liu, Sheng & Shen, Max. (2018). Shared Mobility for Last-Mile Delivery: Design, Operational Prescriptions, and Environmental Impact. *Manufacturing & Service Operations Management*. 20. 10.1287/msom.2017.0683.

2. Bellos, Ioannis & Ferguson, Mark & Toktay, Beril. (2017). The Car Sharing Economy: Interaction of Business Model Choice and Product Line Design. *Manufacturing & Service Operations Management*. 19. 10.1287/msom.2016.0605.

АНАЛІЗ І ПОРІВНЯННЯ РІЗНИХ ПІДХОДІВ ДО ПРОЕКТУВАННЯ RESTFUL ВЕБ-СЕРВІСІВ

Кужентський О.Г., Лемешко А.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Провести аналіз і порівняння різних підходів до проектування RESTful веб-сервісів з метою визначення переваг та недоліків кожного з них.

Для виконання задачі потрібно дослідити літературні джерела та практичні приклади проектування RESTful веб-сервісів. Далі необхідно проаналізувати та порівняти різні підходи до проектування RESTful веб-сервісів за такими критеріями, як ефективність, розширюваність, безпека та зручність використання. На основі результатів порівняння потрібно зробити висновки щодо використання різних підходів до проектування RESTful веб-сервісів.

Мета дослідження. Порівняти різні підходи до проектування RESTful веб-сервісів з точки зору їх ефективності, розширюваності, безпеки та зручності використання. Результати дослідження будуть корисними для розробників веб-сервісів та студентів, які вивчають веб-розробку та цікавляться RESTful веб-сервісами.

Результати дослідження "Аналіз і порівняння різних підходів до проектування RESTful веб-сервісів" вказують на те, що кожен підхід має свої переваги та недоліки залежно від конкретної ситуації, але загалом можна виділити декілька важливих спільних особливостей.

Перевагами RESTful підходів є:

- Простота та зрозумілість структури ресурсів та їхніх методів.
- Спрощення взаємодії між клієнтами та серверами за рахунок використання стандартних HTTP-протоколів.
- Можливість швидкої розробки та розгортання веб-сервісів за рахунок використання стандартних засобів та бібліотек.

Однак, варто враховувати й недоліки підходів, такі як:

- Обмежена функціональність та можливості порівняно з іншими підходами, такими як SOAP.
- Недостатня різноманітність управління помилками та винятками.
- Неможливість передачі більш складних даних за один запит, що може призвести до погіршення продуктивності.

Під час дослідження було виявлено, що вибір підходу до проектування RESTful веб-сервісів повинен залежати від потреб конкретного проекту та специфіки його вимог. Дослідження також показало, що при використанні RESTful підходів можна досягти високої ефективності та зручності використання веб-сервісів.

Висновки та перспективи. Висновок з дослідження полягає в тому, що при проектуванні RESTful веб-сервісів потрібно враховувати специфіку проекту та вимоги до функціональності, а також використовувати обґрунтований вибір підходу на основі аналізу ситуації. При цьому, кожен підхід має свої переваги та недоліки.

Дослідження підходів до проектування RESTful веб-сервісів є актуальним, оскільки веб-сервіси стають все більш популярними і важливими для розробки сучасних додатків та систем.

Для майбутнього дослідження можна розглянути такі перспективи: дослідження підходів до забезпечення безпеки RESTful веб-сервісів, порівняння інструментів та бібліотек для розробки RESTful веб-сервісів, дослідження використання RESTful веб-сервісів для розробки мікросервісної архітектури та аналіз використання RESTful веб-сервісів у сфері Інтернету речей.

У цілому, дослідження дозволить розробникам та студентам, що вивчають веб-розробку, краще зрозуміти підходи до розробки веб-сервісів та обрати оптимальний підхід для свого проекту.

Література:

1. Аналіз і порівняння різних підходів до проектування RESTful веб-сервісів [Електронний ресурс] – Режим доступу до ресурсу: <http://edu.asu.in.ua/mod/book/tool/print/index.php?id=117>.
2. Аналіз і порівняння різних підходів до проектування RESTful веб-сервісів [Електронний ресурс] – Режим доступу до ресурсу: <https://highload.today/rest-api-soap/>.
3. Аналіз і порівняння різних підходів до проектування RESTful веб-сервісів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/restful/index.htm>.
4. Аналіз і порівняння різних підходів до проектування RESTful веб-сервісів [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ОСОБИСТИХ СПОРТИВНИХ ТРЕНУВАНЬ З ВИКОРИСТАННЯМ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ

Луценко І.В., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

Моніторинг особистих спортивних тренувань є важливим аспектом досягнення будь-яких цілей фізичного розвитку людини. Це надає можливість бути усвідомленим в особистій “фітнес подорожі”, що має наступні переваги:

- можливість спостерігати негативні явища (наприклад плато або перетренування, останнє загрожує індивіду погіршенням якості життя за межами спортивної активності і викликати депресію, хронічну втоми і тд.) і змінювати план тренувань, дієту, час і якість відпочинку;
- можливість бути реалістичним у своїх цілях;
- допомагає бути вмотивованим, бачачи свої минулі досягнення;

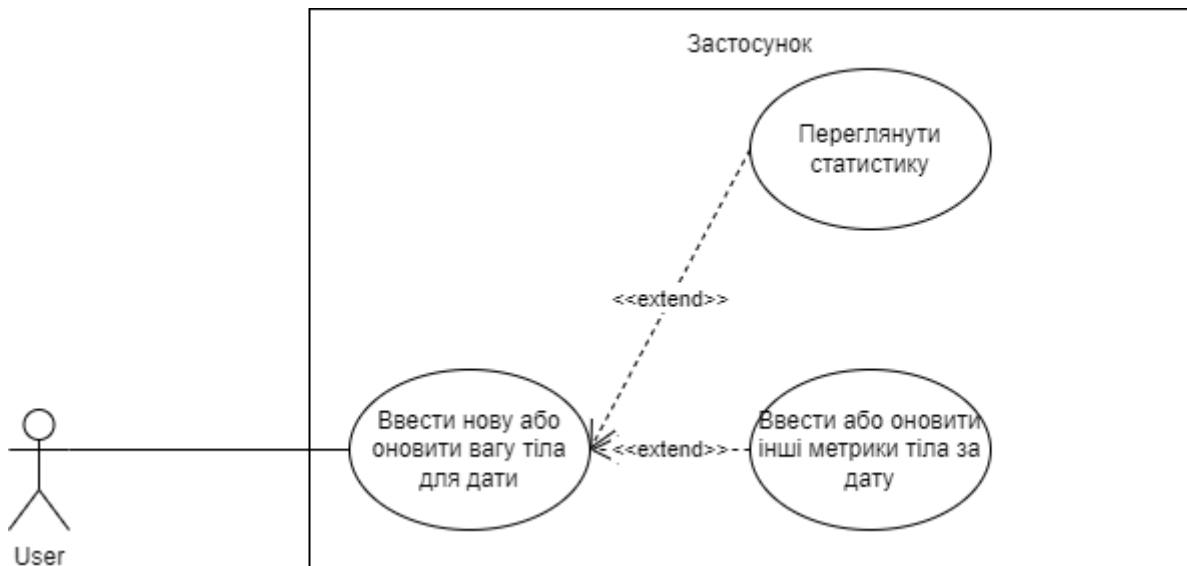


Рисунок 1 – UML діаграма прецедентів функціонала метрик

Одним із головним показником зміни тіла під час тренувань є зміна його ваги, тому користувач може записувати це і переглядати статистику. Також є можливість додавати інші найпопулярніші виміри тіла: відсоток маси жирової та м'язової маси. Ці метрики є необов'язковими через те, що пристрої для їх виміру не є такими розповсюдженими, на відміну від звичайних вагів.



Рисунок 2 – UML діаграма прецедентів функціонала вправ

У застосунку передбачаються список вбудованих вправ, які є найбільш розповсюдженими. Також можливо створювати свої вправи, редагувати і видаляти їх.

Видалення вправ передбачено методом soft delete для того, щоб вже існуючі тренування, які мають у собі цю вправу, не змінювалися.



Рисунок 3 – UML діаграма прецедентів функціонала тренувань

Тренування створюються із шаблонів тренувань. Це зроблено для того, щоб конкретне тренування не повинно було повністю копіювати його шаблон. Це дає можливість тренуванням бути гнучкими, що може стати корисним у багатьох випадках:

- немає можливості використати обладнання, але є можливість виконати аналогічну вправу;
- виконання інших вправ, які не використовують травмовані м'язи;
- можливість додавати або віднімати об'єм вправ в залежності від самопочуття, інших обставин;

Література:

1. Overtraining Syndrome. [Електронний ресурс]. – Режим доступу:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3435910/>

ФРЕЙМВОРКИ ASP.NET CORE та JAVA SPRING

Каліш В. В., Яскевич В.О.

Державний університет телекомунікацій, м. Київ

ASP.NET Core і Java Spring – дві популярні платформи, що використовуються у веб-розробці. Хоча обидва фреймворки створені з однією метою, є деякі відмінності, про які розробники повинні знати, вибираючи, який з них використовувати для свого проекту.

Однією з ключових відмінностей між ASP.NET Core і Java Spring є мова програмування. ASP.NET Core призначений для роботи з C# або F#, а Java Spring використовує Java. Це може бути основним фактором при виборі фреймворку, оскільки розробники можуть мати більше досвіду роботи з однією мовою, ніж з іншою. Обидві мови мають свої сильні та слабкі сторони, тому важливо враховувати, яка з них найбільше підходить для вимог проекту.

Ще одна різниця між двома фреймворками полягає в їхній архітектурі. ASP.NET Core слідує модульній архітектурі, яка дозволяє розробникам вибирати компоненти, необхідні для їх проекту. Це може бути перевагою з точки зору гнучкості, оскільки розробники можуть легко додавати або видаляти компоненти за необхідності. Java Spring, з іншого боку, дотримується архітектури, яка надає набір готових компонентів для використання розробниками. Це може полегшити початок роботи, але може бути менш гнучким у довгостроковій перспективі.

Коли доходить до продуктивності, ASP.NET Core зазвичай вважається швидше, ніж Java Spring. Частково це пов'язано з тим, що C# є мовою, що компілюється, що може призвести до більш високої продуктивності в порівнянні з Java, яка є інтерпретованою мовою. Однак різниця у продуктивності між двома фреймворками може бути незначною і при ухваленні рішення розробникам слід враховувати свої конкретні вимоги до проекту.

Що стосується екосистеми, ASP.NET Core має потужну екосистему інструментів та сервісів, що надаються Microsoft, включаючи Visual Studio та Azure. Це може спростити початок роботи з платформою та її інтеграцію з іншими інструментами та службами Microsoft. Java Spring, з іншого боку, має велику екосистему сторонніх бібліотек та інструментів, які можна використовувати для розширення інфраструктури. Це може надати розробникам велику гнучкість з погляду функціоналу, які вони можуть додати до свого проекту.

Щодо вимог до проекту, ASP.NET Core може краще підходити для

проектів малого та середнього розміру, а Java Spring може краще підходити для більших проектів корпоративного рівня. Однак обидві платформи можна використовувати для широкого кола проектів, і зрештою вибір між ними залежатиме від конкретних потреб проекту, а також навичок та досвіду команди розробників.

Ще одним важливим фактором, який слід враховувати, є підтримка спільноти. І ASP.NET Core, і Java Spring мають активні спільноти розробників, які роблять свій внесок у платформи і надають підтримку іншим розробникам. Розробники повинні враховувати розмір і якість цих спільнот під час прийняття рішення, оскільки це може вплинути на доступність ресурсів та підтримки під час роботи з фреймворком.

Насамкінець, і ASP.NET Core, і Java Spring є потужними платформами для створення веб-додатків, і вибір між ними залежатиме від безлічі фактів. Розробники повинні враховувати своє знайомство з мовами програмування, екосистемою доступних інструментів та сервісів, вимогами проекту, а також розміром та якістю підтримки спільноти під час вибору фреймворку для використання. Вибравши правильну структуру для проекту, розробники можуть створювати масштабовані, ефективні та дієві веб-програми, що відповідають потребам кінцевих користувачів.

Література:

1. ASP.NET documentation - [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0>
2. GitHub ASP.NET Core - [Електронний ресурс]. – Режим доступу : <https://github.com/dotnet/aspnetcore>
3. Java Spring documentation - [Електронний ресурс]. – Режим доступу : <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>
4. 10 Reasons to Use Spring Framework in Projects- [Електронний ресурс]. – Режим доступу : <https://www.geeksforgeeks.org/10-reasons-to-use-spring-framework-in-projects/>

ВИКОРИСТАННЯ SERVICE DISCOVERY ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ

Зайцев І. С., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

Service Discovery - це процес автоматичного виявлення та ідентифікації сервісів (пристроїв, застосунків, ресурсів).

При розробці застосунку який повинен масштабуватися виникає декілька

проблем:

- Автоматичного додавання нових вузлів при збільшенні навантаження без необхідності додаткового втручання. (Auto Scaling)
- Автоматичного видалення вузлів при проблемах з мережею або ресурсами віртуальної машини. (Fault tolerance)
- Рівномірного розподілення навантаження між усіма вузлами застосунку. (Load Balancing)

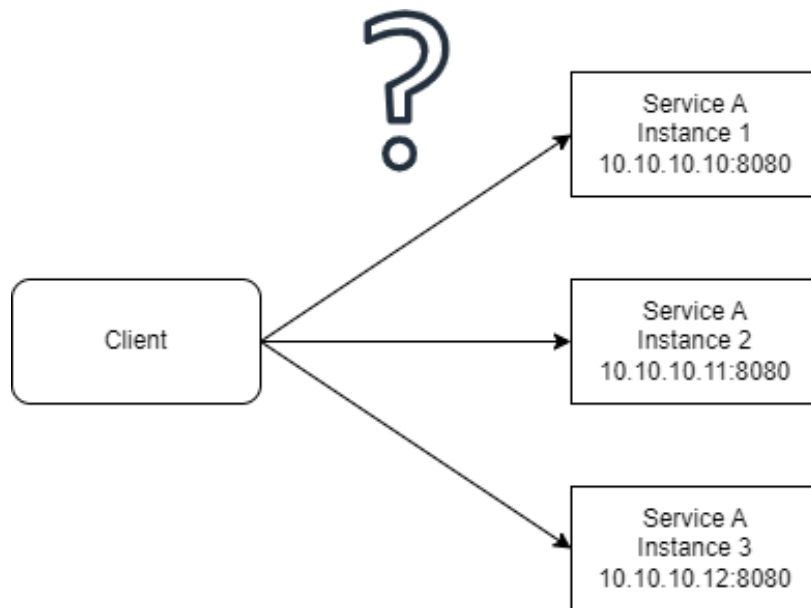


Рисунок 1 - Приклад проблеми балансування запитів

На стороні клієнта вести облік IP адресів проблематично:

- За звичай віртуальній машинізначається динамічний IP.
- Якщо один із вузлів буде недоступний то продуктивність системи починає деградувати.

Рішення

Кожен вузел застосунку реєструється в Service Registry та час від часу відправляє свій статус. Service Registry зберігає інформацію про всі зареєстровані вузли. Кожен вузел може отримати інформацію про сусідні зареєстровані вузли для можливості балансування запитів.

Види Service Discovery:

- На стороні клієнта (Client-Side)

Клієнтська сторона отримує всі зареєстровані вузли застосунку Service A та відповідальна за розподілення запитів.

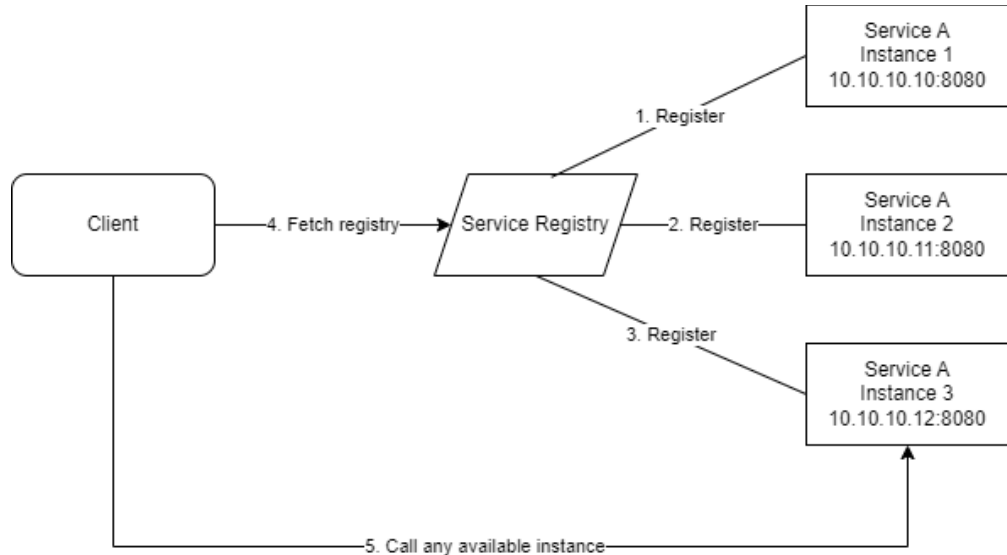


Рисунок 2 - Приклад балансування запитів на клієнтській стороні

Готові рішення:

Zookeeper Curator, Netflix Eureka, Consul

- На стороні сервера (Server-Side)

На серверній стороні створюється механізм який відповідальний за розподілення запитів, клієнтська сторона знає тільки про один сервіс, наприклад Gateway.

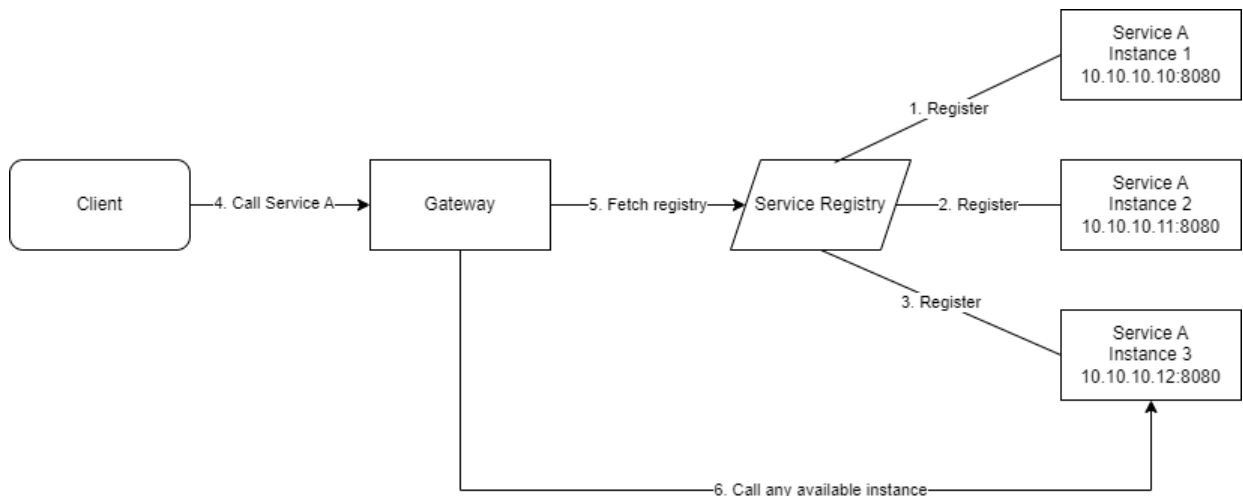


Рисунок 3 - Приклад балансування запитів на серверній стороні

Готові рішення:

DNS, NGINX, Kubernetes, Kafka

Можливі алгоритми балансування:

- Випадковим чином вибирається один із вузлів, наприклад Round Robin.
- В залежності від регіону.
- В залежності навантаження Service Registry може відсортувати вузли по пріоритетності та запропонувати клієнту який вузол краще використовувати.

Висновок

Враховуючи зазначене, для забезпечення реалізації горизонтального масштабування при створенні дистрибутивної системи виконання відкладених завдань було вибрано Zookeeper Service Discovery.

Література:

1) Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd ed. O'Reilly Media, 2021. 615 p.

ПЕРЕВАГИ БІБЛІОТЕКИ REACT ДЛЯ РОЗРОБКИ WEB-ДОДАТКІВ

Федоренко А.А., Трінтіна Н.А

Державний університет телекомунікацій, м. Київ

React – одна з найпопулярніших бібліотек для створення інтерфейсів користувача. Бібліотека має ряд переваг для створення зручних та інтерактивних web-додатків без шкоди для продуктивності SEO.

Одною з переваг бібліотеки є використання концепції віртуального DOM. Віртуальний DOM набагато швидше і ефективніше, оскільки він дозволяє мінімізувати кількість операцій над реальним DOM за рахунок порівняння та оптимального оновлення лише змінених елементів, забезпечуючи ефективну роботу з інтерфейсом користувача. Ця концепція дозволяє оновлювати інтерфейс користувача без необхідності перезавантажувати всю сторінку. Це робить додатки чутливими та зменшує час завантаження сторінки.

Ще одна перевага бібліотеки React є її екосистема. Поточна екосистема бібліотеки настільки обширна, що дозволяє створювати рішення для настільних комп'ютерів, мобільні додатки та створення статичних веб-сайтів.

React має багато розширень та доповнень, таких як Redux, React Router, Material-UI, які дозволяють покращити розробку web-додатків та забезпечують багатий функціонал.

Наступною перевагою є кодування за допомогою JSX. JSX – це синтаксис JavaScript XML. Він поєднує змінні JavaScript та HTML-код. Вирази JSX після компіляції стають звичайними викликами функцій JavaScript, що повертають об'єкти JavaScript. Це надає можливість використовувати JSX в середині виразів та циклів, присвоювати його змінним, приймати його як аргумент і повертати з функцій.

React використовує компонентний підхід до розробки інтерфейсів, дозволяючи розбити їх на окремі, повторно використовувані компоненти. Що сприяє структурованості коду та полегшує його тестування.

Отже, бібліотека React є популярним вибором для розробки web-додатків завдяки своїй продуктивності, легкості використання, гнучкості та зручності для SEO. Вона є потужним інструментом для розробки інтерактивних додатків з високою продуктивністю та зручним інтерфейсом користувача.

Література:

1. Вступ до JSX – React. *React – JavaScript-бібліотека для створення користувацьких інтерфейсів*. URL: <https://uk.reactjs.org/docs/introducing-jsx.html>
2. React for Web Development: 10 Reasons Why You Should to Use it. *Blog | TechMagic*. URL: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/>
3. React Virtual DOM Explained in Simple English - Programming with Mosh. *Programming with Mosh*. URL: <https://programmingwithmosh.com/react/react-virtual-dom-explained/>
4. Why React.js is Becoming Increasingly Popular - Chudovo. *Chudovo*. URL: <https://chudovo.com/why-react-js-is-becoming-increasingly-popular/>

РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ ОРГАНІЗАЦІЇ ОПИТУВАНЬ СТУДЕНТІВ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ REACT. JS

Шевчук Ю.О., Залива В.В.

Державного університету телекомунікацій, м. Київ

Інформатизація освіти стає головною умовою, яка впливає на подальший успішний розвиток як інтелектуального рівня студентів, так і потенціалу нації і суспільства в цілому.

Науковець В.Ю. Биков та інші дослідники констатують, що інформатизація освіти вимагає впровадження у вищу освіту інноваційних методів, засобів та форм професійної підготовки майбутніх фахівців нової формації, де основою даного процесу є впровадження та використання сучасних інформаційно-комунікаційних технологій (ІКТ), зокрема WEB-орієнтованих технологій [1 с. 111].

Застосування сучасних ІКТ та WEB-орієнтованих технологій має на

меті дозволити кожному студенту працювати у комфортному для нього темпі, виявляти свою активність, застосувати творчий підхід до виконання задач, значно покращувати свої знання, проявити вміння нестандартно вирішувати практичні завдання.

Основною метою для кожного викладача-професіонала стає - організувати особистісно-орієнтований підхід до навчання, що в свою чергу надасть змогу здійснити індивідуальний підхід до кожного студента через активізацію його особистої пізнавальної діяльності, використовуючи нові WEB-орієнтовані технології, дотримуючись основних дидактичних принципів: доступності, послідовності, диференційованого підходу, систематичності, науковості та ін.

При цьому викладач повинен так організувати самостійну роботу студентів, щоб забезпечити їм “доступність навчально-методичного забезпечення та довідкового матеріалу; наявність комп’ютерних класів; систему регулярного контролю якості виконаної самостійної роботи; консультаційну допомогу викладача” [2 с.128].

Саме від вміння викладача вдало застосовувати ці технології при викладанні дисциплін залежить рівень компетентності майбутніх фахівців. Для викладача прояв цього вміння є ознакою професіоналізму.

Під час організації процесу вивчення дисциплін одним із важливих аспектів є використання WEB-орієнтованих сервісів: компіляторів (GitHub, AWS Cloud), інтелект-карт, автоматизованих систем перевірки знань з програмування, таких хмарних додатків, як MS Office 365, Google-сайт, Forms, 4 Hangouts, Google Classroom у спільній роботі викладача зі студентами, команднопроектній роботі студентів та в їх комунікації між собою [3 с. 221].

Виходячи з поняття «інформаційні технології», а це є сукупність методів та програмно-технічних засобів, інтегрованих з метою ефективного опрацювання даних, дамо означення WEB-орієнтованих технологій. WEB-орієнтовані технології – це сукупність методів та програмно-технічних засобів, інтегрованих з метою ефективного опрацювання веб-ресурсів, які знаходяться у веб-просторі (локальному або глобальному, наприклад, мережі Інтернет).

На сьогоднішній день, використання веб-орієнтованих технологій у навчальному процесі є актуальним. Цей напрямок досліджується у багатьох публікаціях, як вітчизняних так і зарубіжних авторів. Так, у працях О.М. Спіріна та Т.А. Вакалюк доведено, що використання Web-орієнтованих технологій для навчання, є дуже корисним для проведення лекцій, лабораторних робіт, модульних контрольних робіт, заліків, практичних частин іспитів, тестування тощо [4 с.187].

Література:

1. Svitlana L. Proskura, Svitlana G. Lytvynova. Organization of independent studying of future bachelors in computer science within higher education institutions of Ukraine. ICTERI 2018: 14th International Conference on ICT in Education, Research, and Industrial Applications, Part II: 3d International Workshop on Professional Retraining and Life-Long Learning, using ICT:

Person-oriented Approach (3L-Person 2018) Kyiv. Ukraine. May 14-17, 2018. P. 348-358. URL: http://ceur-ws.org/Vol2104/paper_160.pdf

2. Lytvynova S.H. Cloud-oriented learning environment of secondary school Cloud Technologies in Education: Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017). Kryvyi Rih, Ukraine, April 28, 2017 / Edited by: Serhiy O. Semerikov, Mariya P. Shyshkina. – P. 7-12. – (CEUR Workshop Proceedings (CEURWS.org), Vol. 2168). URL : <http://ceur-ws.org/Vol-2168/paper2.pdf>

3. Биков В.Ю., Бондаренко О.М., Булахова Я.В. Особливості використання інформаційно-комунікаційних технологій у навчальному процесі. Загальна педагогіка та історія педагогіки. 2018.

4. Спірін О.С. Інформаційно-комунікаційні технології моніторингу впровадження результатів науково-дослідних робіт. Інформаційні технології і засоби навчання. 2013, Том 36, №4

СУЧАСНІ ЗАСОБИ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ В ІКТ

Волошин В.В., Негоденко О.В.

Державний університет телекомунікацій, м. Київ

Постановка задачі. Інформаційно-комунікаційні технології (ІКТ) стали необхідною складовою сучасного світу, що вимагає постійного розвитку та вдосконалення. Одним з найважливіших аспектів розробки ІКТ є тестування програмного забезпечення, яке вимагає значних зусиль та часу. У зв'язку з цим, доцільним є використання сучасних засобів автоматизації тестування програмного забезпечення, що дозволяють значно зменшити час та зусилля, витрачені на тестування.

Мета дослідження. Метою даного реферату є розгляд сучасних засобів автоматизації тестування програмного забезпечення в ІКТ, їх переваг та недоліків, а також визначення їх впливу на процес тестування ІКТ.

Результати дослідження: Сьогодні на ринку існує велика кількість засобів автоматизації тестування програмного забезпечення, що дозволяють знизити ризик та час, пов'язаний з тестуванням. Автоматизація тестування - це процес використання програмного забезпечення для виконання тестів, які раніше виконувались вручну. За допомогою автоматизації, тестування може бути більш ефективним та точним, що дозволяє розробникам зосередитися на інших завданнях.

Одним з найпопулярніших засобів, який широко використовується, є Selenium. Цей інструмент дозволяє автоматизувати тестування веб-додатків та інших веб-інтерфейсів. Крім того, для тестування функціональної стійкості веб-додатків та навантаження серверів використовується JMeter. Інші засоби автоматизації тестування включають Appium для мобільного тестування, SoapUI для тестування веб-сервісів, TestComplete для автоматизації тестування десктопних та мобільних додатків, а також Cucumber для автоматизації функціональних тестів.

Важливо зазначити, що автоматизація тестування дозволяє знизити ризик людської помилки та забезпечити високу точність тестування. Засоби автоматизації тестування можуть допомогти покращити ефективність тестування та знизити витрати на нього, що може бути особливо важливим для бізнесу. Крім того, засоби автоматизації тестування включають в себе широкий спектр можливостей, які дозволяють тестувати різні аспекти програмного забезпечення, включаючи функціональність, безпеку та продуктивність.

Для тестування безпеки програмного забезпечення, можна використовувати засоби, такі як OWASP ZAP, Burp Suite та Acunetix. Ці інструменти дозволяють виявляти вразливості в програмному забезпеченні та надавати звіти з рекомендаціями щодо їх виправлення.

Звичайно, не можна оминати недоліки автоматизованого тестування. Наприклад, низька стійкість до змін в програмному забезпеченні може призвести до неправильних результатів тестування. Крім того, процес автоматизованого тестування може бути досить складним та вимагати від розробників додаткових знань та навичок. Однак, з правильним підходом та належною підготовкою, автоматизація тестування може бути дуже ефективним інструментом для розробки програмного забезпечення.

В цілому, автоматизація тестування є однією з найважливіших складових розробки програмного забезпечення. З використанням засобів автоматизації тестування, можливо знизити ризики та затрати, пов'язані з тестуванням, та збільшити ефективність та точність тестування. Це дозволяє розробникам програмного забезпечення швидше та з більшою впевненістю випускати свої продукти на ринок. Крім того, з розвитком штучного інтелекту та машинного навчання, автоматизація тестування може стати ще більш ефективною та точною, дозволяючи розробникам програмного забезпечення зосередитися на більш складних завданнях та забезпечувати високу якість своїх продуктів..

Висновки та перспективи. Сучасні засоби автоматизації тестування є ефективним інструментом для забезпечення високої якості програмного забезпечення. Однак, для досягнення максимальної ефективності, необхідно обрати підходящий засіб, враховуючи особливості тестування та вимоги до програмного забезпечення. Крім того, важливо враховувати недоліки автоматизованого тестування та шукати шляхи їх подолання.

Література:

1. Selenium - <https://www.selenium.dev/>
2. JMeter - <https://jmeter.apache.org/>
3. Appium - <http://appium.io/>
4. SoapUI - <https://www.soapui.org/>

РОЗРОБКА НАВЧАЛЬНОГО МОДУЛЯ ЗА ТЕМОЮ "КОМБІНАТОРИКА" МОВОЮ C#

Черкасов К.А., Корецька В.О.

Комбінаторика, або комбінаторний аналіз - це розділ математики, що розглядає проблеми вибору, упорядкування та роботи в скінченій або дискретній величині. Однією з базових проблем є число можливих конфігурацій (наприклад графів, конструкцій, масивів тощо) заданого типу. Навіть коли правила визначення конфігурації, відносно прості, підрахунок інколи може становити значні труднощі. Математику, можливо, доведеться задовольнитися знаходженням приблизної відповіді.

Також є проблеми оптимізації. Наприклад, економічна функція f присвоює числове значення $f(x)$ будь-якій конфігурації x з певними заданими властивостями. У цьому випадку проблема полягає в тому, щоб обрати конфігурацію x_0 , яка мінімізує $f(x)$ або робить її $\epsilon = \text{minimal}$, тобто для будь-якого числа $\epsilon > 0$, $f(x_0) \leq f(x) + \epsilon$ для всіх конфігурацій x , із зазначеними властивостями.

C# (вимовляється як «Си Шарп») — сучасна, об'єктно-орієнтована типобезпечна мова програмування. C# дозволяє розробникам створювати багато видів безпечних та надійних програм, які працюють на платформі .NET. Мова C#, відповідно до ресурсу dou.ua, займає четверте місце по популярності комерційного використання в Україні з часткою 13,3%, поступаючись Python(13,4%), Java(14%) та JavaScript(19,1%). Головними перевагами мови C# вважаються універсальність - цю мову використовують для розробки найрізноманітніших додатків: веб-додатків, desktop-додатків, database-додатків, ігор тощо, багатоплатформність- можливість розробляти додатки для різних операційних систем та апаратних платформ, відносно висока швидкість виконання коду, автоматичний збір сміття (automatic garbage collection).

Головна задача додатка - це допомогти в першу чергу студентам ВУЗів у вивченні комбінаторики впродовж навчального процесу.

Зараз вже існують калькулятори для формул з комбінаторики, але вони лише приймають вхідні дані та виводять кінцеву відповідь, вони не пояснюють значення цих формул та не показують процес вирішення. Це може бути корисним для самоперевірки, але калькулятор не допоможе вивчити власне предмет.

Тож основна вимога до додатка - це зрозумілість. Користувач, котрий ніколи не чув про комбінаторику, має бути здатним повністю опанувати цей предмет без сторонніх джерел. Додаток має мати зрозумілий інтерфейс, зрозумілі теоретичні дані, перелік задач на різні теми та калькулятор, котрий буде детально описувати процес розв'язання задачі.

Література:

1. Combinatorics [Електронний ресурс]. – Режим доступу: <https://www.britannica.com/science/combinatorics>
2. Рейтинг мов програмування 2023 - [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/language-rating-2023/>
3. .NET documentation - [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/>
4. C# Introduction - [Електронний ресурс]. – Режим доступу: <https://www.knowledgehut.com/tutorials/csharp/csharp-introduction#:~:text=C%23%20is%20more%20type%20safe,programming%20language%20with%20multiple%20paradigm s.>

ВИДИ ТА МЕТОДИ ТЕСТУВАНЬ ПРОДУКТИВНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ

Бобков Р.А

Державного університету телекомунікацій, м. Київ

Тестування продуктивності – це комплекс типів тестування, метою якого є визначення працездатності, стабільності, споживання ресурсів та інших атрибутів якості додатка, в умовах різних сценаріїв використання та навантажень. Тестування продуктивності дозволяє виявити можливі вразливості та недоліки в системі, з метою запобігання їх негативному впливу на роботу програми. Також, тестування продуктивності може служити для перевірки та підтвердження інших атрибутів якості системи, таких як масштабованість, надійність та споживання ресурсів. [1]

Цілі тестувань можуть відрізнятися, в залежності від технологій, що використовуються додатком або його призначенням, однак вони можуть включати в себе наступне:

- Паралелізм / Пропускна спроможність;
- Час відповіді сервера;
- Час відображення;
- Вимоги до продуктивності. [2]

Послідовність етапів тестувань та їх особливості:

1. Аналіз системи та підбір вимог. Важливо, щоб перед проведенням тестування система була закінчена. Якщо процес розробки не завершено, отримані дані будуть відображати недійсні результати. Вимоги – це критерії, яким система відповідає з технічного погляду.

2. Підготовка стратегії. Стратегія розробляється на основі детального аналізу ПЗ та описує розширений підхід до тестування продуктивності.

3. Налаштування генератора навантаження. Для якісного проведення тестування продуктивності інструмент встановлюється на генератор

навантаження – віртуальну чи фізичну машину, розташовану максимально близько до сервера/ів програми.

4. Проведення моніторингу серверів та генератора навантаження. Для оцінки параметрів продуктивності ПЗ використовують такі утиліти як:

- Nmon;
- Performance monitor;
- Zabbix;
- Grafana;
- New Relic.

5. Підготовка тестових даних. Універсального підходу до такої підготовки немає. Часто для коректної роботи рішення потрібно створювати дані в обсязі, достатньому для розробки скриптів та проведення тестів.

Можна виділити 4 способи підготовки даних:

- Код;
- SQL-запити;
- API-запити;
- Інтерфейс.

Під час виконання скрипту користувачі записуються до БД, після чого робиться знімок файлової системи для подальшого використання отриманої інформації при запусках тестів.

6. Розробка навантажувальних скриптів. Сценарії тестування продуктивності створюються за допомогою вибраного інструменту. Процес розробки скриптів складається із трьох кроків:

- Вивчення сценаріїв;
- Створення тест-кейсів;
- Налаштування сценаріїв.

7. Попередні запуски тестів. Попередні запуски тестів потрібні для перевірки коректності роботи підготовлених скриптів навантаження та пошуку оптимальної моделі навантаження на систему.

При тестуванні продуктивності виконуються такі види перевірок:

- Стрес-тест;
- Навантажувальне тестування;
- Перевірка стабільності;
- Об'ємне тестування.

8. Аналіз результатів та підготовка звіту. За підсумками попередніх етапів створюється документ, який описує як результати проведених тестів, так і хід виконання кожного з них. [3]

Отже, тестування продуктивності є важливою частиною для правильного функціонування системи. Правильно проведене тестування та попередні розрахунки результатів можуть якісно допомогти в підтриманні функціонування, підвищення швидкості та комфорту використання комп'ютерними системами.

Література:

1. Різниця між тестуванням продуктивності, тестуванням навантаження та стрес-тестуванням - на прикладах [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.myservername.com/performance-testing-vs-load-testing-vs-stress-testing>
2. Тестування продуктивності [Електронний ресурс] – Режим доступу до ресурсу: https://www.wiki.uk-ua.nina.az/Тестування_продуктивності.html
3. Що таке тестування [Електронний ресурс] – Режим доступу до ресурсу: https://qalearning.com.ua/theory/about_qa/shpargalka-z-testuvannya/

РОЗРОБКА МЕТОДІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ RESTFUL ВЕБ-СЕРВІСІВ

Кужентський О.Г., Лемешко А.В.

Державного університету телекомунікацій, м.Київ

Постановка задачі. Задача полягає в розробці методів автоматизації тестування RESTful веб-сервісів. Методи повинні дозволяти проводити автоматизоване тестування, забезпечувати швидке та ефективне виконання тестів та легко розширювати тестовий набір. Це допоможе забезпечити високу якість та безперебійну роботу веб-сервісу.

Мета дослідження. Метою дослідження є розробка ефективних методів автоматизації тестування RESTful веб-сервісів з метою забезпечення високої якості тестування та безперебійної роботи веб-сервісів.

Результати дослідження. Результати дослідження включають огляд існуючих методів тестування RESTful веб-сервісів, розробку нових методів тестування, створення програмного забезпечення для автоматизації тестування, експериментальне дослідження розроблених методів та програмного забезпечення, а також розробку рекомендацій щодо використання розроблених методів тестування та їхньої оптимізації.

В результаті дослідження було виявлено переваги та недоліки існуючих методів тестування RESTful веб-сервісів. Наприклад, метод, що використовує засіб Postman, має недоліки в складності написання тестів та необхідності виконання тестів вручну.

Було розроблено нові методи тестування, які дозволяють ефективно тестувати RESTful веб-сервіси, зручно створювати й виконувати тести. Один з нових методів - створення тестових наборів, що дозволяє автоматично генерувати тести з різними параметрами, тим самим забезпечуючи покриття варіативності вхідних даних.

Для автоматизації тестування було створено програмне забезпечення, яке дозволяє автоматично створювати та виконувати тести. Програмне забезпечення було розроблено з використанням мови програмування Python та бібліотеки pytest.

Експериментальне дослідження розроблених методів та програмного забезпечення показало їхню ефективність та зручність використання. Було проведено тестування на реальних RESTful веб-сервісах, що дозволило оцінити якість тестування та виявити деякі проблеми в роботі сервісів.

Висновки та перспективи. В результаті проведеного дослідження було виявлено, що розроблені методи тестування та програмне забезпечення для їх

автоматизації є ефективними та зручними для використання при тестуванні RESTful веб-сервісів. Використання цих методів дозволяє забезпечити покриття різних варіантів вхідних даних та перевірити коректність поведінки сервісів в різних умовах.

Література:

1. Розробка методів автоматизації тестування RESTful веб-сервісів. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>.
2. Розробка методів автоматизації тестування RESTful веб-сервісів. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/restful-web-services/>.
3. Розробка методів автоматизації тестування RESTful веб-сервісів. [Електронний ресурс] – Режим доступу до ресурсу: <https://spring.io/guides/gs/rest-service/>.
4. Розробка методів автоматизації тестування RESTful веб-сервісів. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.

РОЗРОБКА МЕСЕНДЖЕРУ НА ANDROID

Виходцев М.М., Трінтіна Н.А.

Державний університет телекомунікацій, м. Київ

Створення месенджера це велика та комплексна задача. Перед розробником постає задача створити оптимальний продукт, щоб задовольнити потреби користувачів. Такий продукт має бути зручним у використанні, мати “real-time” функції, задовольняти питання безпеки та мати стабільне підключення.

Дуже важливою частиною додатка виступає сервер. Саме через нього відбуваються всі основні операції в месенджері: авторизація, збереження й читання інформації з бази даних, обмін повідомлень між користувачами в чатах тощо. Сервер має бути швидким, безперебійно працювати та завжди мати стабільне підключення з клієнтським додатком. Знаючи всі потреби, можна обрати платформу Firebase.

Firebase - це Backend-as-a-Service від Google (Рис. 1). Маючи великий стек технологій, такі як Authentication, Realtime Database, Cloud Storage тощо, Firebase можна вважати оптимальним рішенням, для розв’язання серверних задач. Додатковою перевагою є безпека даного рішення, яка забезпечується створенням правил на сервері, що керують доступом на читання й запис інформації з баз даних. Також варто зазначити, що весь трафік між сервером та клієнтським трафіком шифрується.

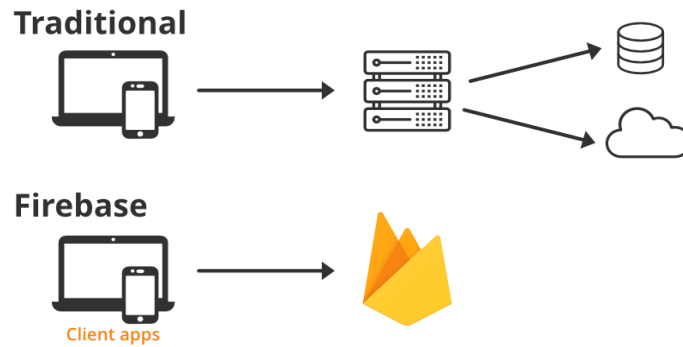


Рис. 1 – Firebase в порівнянні з традиційним сервером

Маючи Firebase як серверне рішення, залишилося розробити клієнтський додаток на Android. Мовою програмування виступає Java з використанням Android SDK – комплект для розробки програмного забезпечення, що містить повний набір інструментів для розробки. Додатково використовується Android Jetpack – набір бібліотек, інструментів та архітектурних рекомендацій, які допомагають швидко та легко створювати програми для Android. Для роботи з асинхронністю представлена бібліотека ReactiveX, яка являє собою функціональне-реактивне програмування. В розробці додатка використовується архітектурний патерн MVVM (Model-View-ViewModel) (Рис. 2)

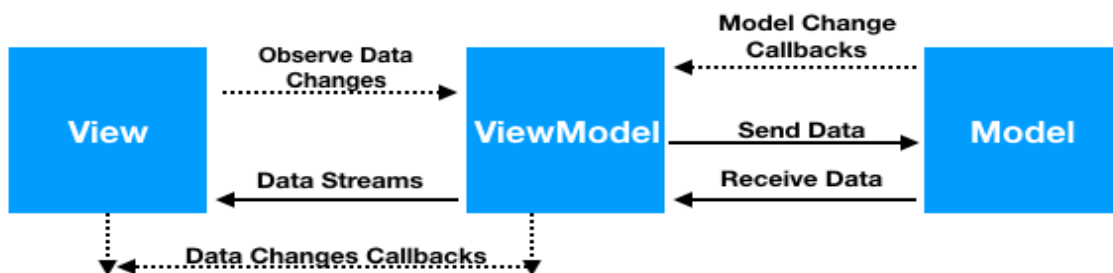


Рис. 2 – Приклад архітектурного патерну MVVM

З правильним підходом до розробки з використанням вищезазначених технологій, створення месенджеру стає набагато простішим, а кінцевий результат, легше підтримувати та доповнювати. Програмний продукт буде якіснішим та буде надавати кращий User Experience.

Література:

1. Android for developers. [Електронний ресурс]. – Режим доступу: <https://developer.android.com/>
2. Firebase docs. [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs>

ВИКОРИСТАННЯ ІКТ ДЛЯ ПОКРАЩЕННЯ ДІАГНОСТИКИ ТА ЛІКУВАННЯ ДІАБЕТУ

Петренко Е.Д., Трінтіна Н.А.

Державний університет телекомунікацій, м. Київ

За даними Всесвітньої організації охорони здоров'я (ВООЗ), у світі кількість людей, які живуть з діабетом, зросла від 108 мільйонів у 1980 році до 422 мільйонів у 2022 році. Це означає, що понад 8,5% дорослого населення світу мають діабет.

Діабет є одним з найпоширеніших захворювань у світі, і він ставить виклики перед системами охорони здоров'я і пацієнтами, які потребують постійного моніторингу рівня глюкози в крові і контролювання дієти.

За даними Міністерства охорони здоров'я України, станом на 2021 рік, більше 1,5 мільйонів українців страждають від діабету. Це означає, що понад 4% населення України мають цю хворобу. Крім того, згідно зі статистикою, кількість хворих на діабет українських дітей також зростає.

Поширеність діабету у світі та в Україні підкреслює необхідність розвитку інноваційних технологій, зокрема ІКТ, для покращення моніторингу та лікування цього захворювання.

ІКТ можуть бути використані для розробки інструментів, які дозволяють здійснювати моніторинг рівня глюкози в крові в режимі реального часу, управляти захворюванням і надавати дистанційну медичну допомогу людям з діабетом. Ця технологія дозволяє медичним працівникам дистанційно спостерігати за пацієнтами і надавати своєчасну підтримку в разі потреби.

Підключивши такі пристрої, як інсулінові помпи та монітори рівня глюкози в крові, до мережі, пацієнти та медичний персонал можуть отримати доступ до актуальної інформації про стан пацієнта та контролювати його стан з віддаленого місця.

Мета дослідження. Це дослідження має з'ясувати, як ІКТ можуть покращити лікування діабету, а також оцінити ефективність використання ІКТ для автоматичної інсулінотерапії та моніторингу рівня глюкози в крові.

Результати дослідження:

Інформаційно-комунікаційні технології (ІКТ) стали потужним інструментом в охороні здоров'я, який має потенціал для значного покращення лікування діабету. Однією з ключових переваг ІКТ є можливість збирати дані в режимі реального часу за допомогою датчиків. Зокрема, спеціалізовані пристрої можуть вимірювати рівень глюкози в крові та кількість введеного пацієнту інсуліну, що дозволяє лікарям постійно контролювати стан пацієнта та вчасно вживати заходів у разі потреби.

Потенціал ІКТ у лікуванні діабету може бути реалізований лише за умови розробки складного програмного забезпечення для збору та аналізу даних. За допомогою спеціалізованих програм, які забезпечують аналіз даних у режимі реального часу, лікарі можуть створювати індивідуальні плани лікування та коригувати їх відповідно до потреб пацієнта. Наявність онлайн-доступу до медичних даних за допомогою ІКТ також полегшує дистанційний моніторинг і контроль стану пацієнтів, що призводить до покращення результатів лікування.

ІКТ також можуть бути використані для полегшення автоматичної інсулінотерапії для діабетиків. Використовуючи спеціальні датчики і насоси, ІКТ-технології можуть контролювати рівень глюкози в крові і вводити інсулін за необхідності, допомагаючи пацієнтам підтримувати нормальний рівень глюкози в крові і знижуючи ризик ускладнень, пов'язаних з діабетом. Крім того, можливість зберігати та аналізувати дані про стан здоров'я пацієнтів за допомогою ІКТ-інструментів дозволяє лікарям надавати більш точні та індивідуалізовані плани лікування, а також здійснювати дистанційний моніторинг та контроль стану здоров'я пацієнтів.

Таким чином, використання ІКТ може суттєво підвищити ефективність і безпеку лікування діабету завдяки можливості збору та аналізу даних у режимі реального часу, підтримці пацієнтів у повсякденному житті та полегшенню дистанційного моніторингу. Хоча потенційні переваги ІКТ у лікуванні діабету є значними, важливо забезпечити високі стандарти безпеки та конфіденційності даних, перш ніж використовувати інструменти ІКТ у медичних цілях.

Висновки та перспективи. Роль ІКТ-технологій у лікуванні діабету важко переоцінити, оскільки вони мають величезний потенціал для покращення якості життя пацієнтів. Однак до їх впровадження необхідно підходити з обережністю та відповідальністю, щоб забезпечити їх ефективність та безпеку. Постійні дослідження і розробки в цій галузі можуть сприяти створенню нових технологій та інновацій, які можуть зробити революцію в лікуванні діабету та догляді за ним, що призведе до покращення стану здоров'я пацієнтів.

Література:

1. International Diabetes Federation. IDF Diabetes Atlas. 9th ed. Brussels, Belgium: International Diabetes Federation; 2019. Режим доступу: https://www.diabetesatlas.org/upload/resources/material/20200302_133351_IDFATLAS_9e-final-web.pdf
2. Ministry of Health of Ukraine. Diabetes. <https://moz.gov.ua/koronavirus-2019-ncov/diabet>
3. Gabbay RA, Lendel I, Saleem TM, Shaeffer G, Adelman AM, Maeser P. Використання хмарної системи управління діабетом для титрування інсуліну у пацієнтів з цукровим діабетом 2 типу. J Diabetes Sci Technol. 2019;10(3):707-713.
4. Lim S, Kang SM, Kim KM та ін. Багатофакторне втручання в лікуванні діабету з

використанням моніторингу в режимі реального часу та індивідуального зворотного зв'язку при діабеті 2 типу. Acta Diabetol. 2018;53(2):189-198.

РОЗРОБКА TELEGRAM-БОТА ДЛЯ ПОШУКУ КУЛІНАРНИХ РЕЦЕПТІВ МОВОЮ PYTHON

Яловик Д.В, Негоденко О.В.

Державний університет телекомунікації, м. Київ

Все більше користувачів у світі очікують від брендів негайної відповіді у будь-який час доби. Але більшість питань користувачів є типовими. Правильно налаштована програма, наприклад, Телеграм-бот, на них цілком може відповідати. Роботу не складно відповісти 100 користувачам одночасно, а менеджера доведеться чекати. Телеграм-бот створює відчуття, що бізнес завжди з тобою на зв'язку. У середньому Телеграм-ботам відповідає 30-40% користувачів, а добре налаштовані Телеграм-боти отримують відповіді у 90% випадків.

За даними статистики, 69% користувачів віддають перевагу спілкуванню з ботами, тому що вони можуть отримати відповіді зі зручною для себе швидкістю. Логіка робота контролюється за допомогою HTTPS запитів до нашого API для роботів.

Як створити телеграм-бота за допомогою Python:

Всі реалізації API зберігаються у класі під назвою **TeleBot**. Він пропонує багато способів прослуховування вхідних повідомлень, а також функції **send_message()**, **send_document()** та інші, щоб надсилати повідомлення.

Створіть новий файл **bot.py** та вставте наступний код:

```
import os
import telebot
BOT_TOKEN = os.environ.get('BOT_TOKEN')
bot = telebot.TeleBot(BOT_TOKEN)
```

У коді вище ми використовуємо бібліотеку **os**, щоб прочитати змінні середовища, збережені у нашій системі.

У минулому кроці ми експортували змінну середовища під назвою **BOT_TOKEN**. Значення **BOT_TOKEN** зчитується у змінній під назвою **BOT_TOKEN**. Крім того, ми використовуємо клас **TeleBot**, щоб створити екземпляр бота та передаємо йому **BOT_TOKEN**.

Потім потрібно зареєструвати обробники повідомлень. Ці обробники

повідомлень містять фільтри, які має пройти повідомлення. Якщо повідомлення проходить фільтр, викликається декорована функція, а вхідне повідомлення надається як аргумент.

Визначимо обробник повідомлень, який обробляє вхідні команди **/start** та **/hello**.

```
@bot.message_handler(commands=['start', 'hello'])
```

```
def send_welcome(message):
```

```
    bot.reply_to(message, "Howdy, how are you doing?")
```

Для функції, декорованої обробником повідомлень, прийнятне будь-яке ім'я, але вона може мати лише один параметр (повідомлення).

Додамо ще один обробник, який повертає усі вхідні повідомлення відправнику.

```
@bot.message_handler(func=lambda msg: True)
```

```
def echo_all(message):
```

```
    bot.reply_to(message, message.text)
```

Наведений вище код використовує вираз **lambda**, щоб протестувати повідомлення. Оскільки нам потрібно відправляти усі повідомлення назад, ми завжди повертаємо **True** від функції **lambda**.

Тепер ми маємо простий бот, який відповідає на команди **/start** та **/hello**, а також надсилає усі інші надіслані повідомлення відправнику. Додайте наступне до кінця файлу, щоб запустити бот:

```
bot.infinity_polling()
```

Література:

1. Документація телеграм боти: <https://tlgrm.ru/docs/bots>
2. Створення Телеграм-бота за допомогою Python:
<https://www.freecodecamp.org/ukrainian/news/yak-stvoryty-telehram-bota-za-dopomohoyu-python/>

АКТУАЛЬНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ПРОЦЕСУ НАДАННЯ ПЕРШОЇ ДОПОМОГИ

Спіцин А.Я., Трінтіна Н.А.

Державний університет телекомунікацій, м. Київ

У сучасному світі смартфони є одним з найпоширеніших засобів зв'язку та спілкування. Незалежно від того, чи використовуєте ви його для звичайних розмов або для ведення бізнесу, є безліч ситуацій, коли може знадобитися перша допомога. Статистика показує, що більшість людей не має достатніх знань про те, як надавати першу допомогу, що може призвести до непотрібної затримки в допомозі або навіть погіршення стану постраждалого.

Один зі способів розв'язання цієї проблеми полягає в розробці додатку, який надасть користувачам інструкції першої допомоги у режимі реального часу. Він може стати важливим допоміжним засобом в будь-якій ситуації, коли необхідна невідкладна медична допомога.

Щоб забезпечити максимальну ефективність, додаток буде озвучувати всі інструкції, щоб користувач міг сконцентруватися на наданні допомоги постраждалому. Проте, користувачеві все ще доведеться натискати на екран смартфона, щоб повідомити додаток про стан постраждалого для отримання наступних інструкцій.

На відміну від офіційного додатку Червоного Хреста для першої допомоги, розроблюваний додаток буде мати більш оперативний доступ до інструкцій. Це дасть можливість зекономити трохи часу та негайно перейти до допомоги потерпілому. Крім того представлення інформації в аудіо форматі звільнить користувача додатку необхідності постійно дивитися на екран, читаючи інструкції. Такі пункти вигідно відрізняють розроблюваний додаток.

Додаток стає особливо корисним у сучасних реаліях війни. У такий час значно зростає небезпека для громадян і крім того медична допомога потерпілому може бути недоступна, або надходити занадто довго. Але правильна та своєчасна перша допомога може виграти цей дорогоцінний час.

Загалом, розробка додатку, який би надавав інструкції першої допомоги у реальному часі, може стати важливим кроком у покращенні доступу до медичної допомоги в будь-яких умовах. Для цього необхідно внести значний вклад у розробку та вдосконалення додатку, а також в його популяризацію серед широкої громадськості, щоб кожен зміг навчитися надавати першу допомогу та зберегти життя в критичний момент.

Література:

3. Сайт ZME Science [Електронний ресурс] – Режим доступу:
<https://www.zmescience.com/science/many-uk-adults-not-confident-they-can-administer-first-aid-in-an-emergency/>
4. Сайт PubMed [Електронний ресурс] – Режим доступу:
<https://pubmed.ncbi.nlm.nih.gov/27149308/>

МОНІТОРИНГ АКТИВНОСТІ КОРИСТУВАЧІВ ВЕБ-РЕСУРСУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ AJAX

Волошко Д.С.

Державного університету телекомунікацій, м. Київ

В наш час Інтернет став невід'ємною частиною життя сучасної людини. Потужний розвиток інформаційного Web-середовища призвів до зміни вимог самих же Web-додатків. Цим зумовлена тенденція до створення так званих “багатих” Web-додатків, тобто додатків, інтерфейс яких надає ті самі

можливості які не відрізняються від можливостей у звичайному додатку.

Але при роботі багатьох програм, що підтримують протокол передачі даних, повністю прибрати затримку відповіді, пов'язану з передачею даних через Інтернет, практично неможливо.

Метою доповіді є аналіз переваг та недоліків технології Ajax – підходу, призначеного для зменшення негативного ефекту від затримки даних при передачі.

Ajax (асинхронний JavaScript + XML) – це фактично новий підхід, який був вперше представлений в статті Джесі Джеймса, щодо побудови користувацьких інтерфейсів Web-додатків, за яких Web-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні дані користувачу (тобто користувач може переглядати далі контент сайту, поки сервер все ще обробляє запит користувача, щоб потім йому довантажити потрібні дані). Дана технологія не самостійна, а скоріш за все – концепція використання суміжних технологій.

Ajax – підхід, який призначений для користувацьких інтерфейсів, комбінує в цілому кілька основних методів:

- використання DHTML служить для динамічної зміни змісту сторінки;
- використання XMLHttpRequest потрібне для звернення до сервера “на льоту”, без перезавантаження всієї сторінки повністю;
- динамічне підвантаження коду JavaScript;

Використання цих підходів дозволяє створювати набагато зручніші Web-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем.

Переваги даної технології:

- значна економія трафіку користувачів;
- зменшення навантаження на сервер;
- впливове прискорення реакції інтерфейсу;
- безмежні можливості для інтерактивної обробки.

Недоліками технології Ajax можна вважати:

- відсутність інтеграції з стандартними інструментами браузера;
- необхідність завжди контролювати включення JavaScript в браузері;
- ускладнення проекту.

На даний час спостереження за активністю користувачів мають значною мірою великий вплив для покращення роботи багатьох Web-ресурсів та для спрощення їх роботи з користувачем, тому Ajax-скрипти мають досить широку сферу застосування:

- створювати невеликі елементи управління, такі як додавання товарів в кошик в інтернет-магазині;
- виконувати непомітні для користувача дії, (наприклад, автозбереження редагованого тексту);
- динамічно довантажувати дані із сервера в міру необхідності (наприклад, в онлайн-чатах).

На даний час спостереження за активністю користувачів мають значною мірою великий вплив для покращення роботи багатьох Web-ресурсів та для

спрощення їх роботи з користувачем, тому Аїах-скрипти мають досить широку сферу застосування:

- створювати невеликі елементи управління, такі як додавання товарів в корзину в інтернет-магазині;
- виконувати непомітні для користувача дії, (наприклад, автозбереження редактованого тексту);
- динамічно довантажувати дані із сервера в міру необхідності (наприклад, в онлайн-чатах).

Література:

1. Яковлев А.А. Довжиков А.А. Моніторинг активності користувачів Web-ресурсу, 2014. – 96 с.
2. Інтернет-джерело – режим доступу: <http://webstudio2u.net/ua/programming/449-ajax.html>
3. Інтернет-джерело – режим доступу: <https://adaptivepath.org/ideas/ajax-new-approach-web-applications/>

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ DRAFTSIGHT

Степанова А.В

Державного університету телекомунікацій, м. Київ

Програма DraftSight дозволяє створювати, редагувати та переглядати будь-які 2D-креслення та наносити на них електронні позначки. Додаток має стандартний інтерфейс, тому перехід з будь-якої іншої системи не складе труднощів. DraftSight є пропрієтарною 2D САПР (системою автоматизованого проектування та автоматизованої розробки) для інженерів, архітекторів, дизайнерів, студентів та викладачів.

DWG-файли містять двійкові дані для проектування САД і цей формат використовується багатьма програмами САПР.

Формат DWG, розроблений компанією Autodesk і використовуваний у AutoCAD понад 35 років, є стандартом де-факто у сучасному проектуванні та часто використовується для перетворення файлів САД на універсальний формат, який можна прочитати в інших програмах САПР.

DraftSight має схожий інтерфейс і функціонал з класичним AutoCAD версій R14, 2000-2008, тому він відразу набув популярності серед інженерів, що працюють з 2D кресленнями.

DraftSight конкурує з більш ніж трьома десятками 2D або гібридних 2D/3D-систем на ринку. DraftSight для Windows була випущена в лютому 2011 року і була завантажена понад 1,8 мільйона разів (станом на грудень 2011 року).

Підтримує мови: англійську, спрощену китайську, традиційну китайську, чеську, французьку, німецьку, італійську, іспанську, японську, корейську, польську, португальську, турецьку та російську мову.

Переваги рішень DraftSight:

- Створення, редагування, перегляд та коментування будь-якого типу 2D- та 3D-файлів DWG.
- Гнучкі можливості вибору версії відповідно до бюджету та потреб бізнесу.
- Потужні можливості, які дають змогу скоротити час створення креслень, проектування та підготовки документації. Существенная экономия по сравнению с сопоставимыми решениями САД.
- Інтеграція з іншими рішеннями Dassault Systèmes, включаючи 3DEXPERIENCE Marketplace, GEOVIA, DELMIA, SOLIDWORKS, SOLIDWORKS PDM, 3D-друку, виробництва та інших завдань.
- Висока сумісність файлів DWG для поточних та застарілих проектів.
- Швидке освоєння завдяки звичному інтерфейсу користувача і командам.
- Налаштування та автоматизація за допомогою макросів та API, а також просте перенесення існуючих робочих процесів з інших програм.
- Просте розгортання, керування мережними ліцензіями та доступ до комплексної технічної підтримки для максимального підвищення продуктивності.

Отже, рішення призначене для окремих користувачів та компаній, яким потрібно САПР професійного класу з інструментами підвищення продуктивності та функціями налаштування за допомогою API-інтерфейсів. Ці API-інтерфейси особливо корисні для користувачів, які працюють із автоматизованою, налаштованою або інтегрованою САПР на базі DWG за допомогою стандартних операцій LISP або інших макросів. DraftSight Professional також включає стандартний галузевий контент, який легко додати до документа і який підтримує пакетний друк.

Література:

1. [електронний ресурс] — режим доступу: <https://ru.wikipedia.org/wiki/DraftSight>
2. [електронний ресурс] — режим доступу: <https://www.draftsight.com/ua/home-page>
3. [електронний ресурс] — режим доступу: <https://intersed.kiev.ua/draftsight>

ОБСЛУГОВУВАННЯ ПК ДЛЯ ЇХНЬОЇ БЕЗПОСЕРЕДНЬОЇ РОБОТИ

Меркулов М.І.

Державного університету телекомунікацій, м. Київ

Що таке взагалі обслуговування ПК? Це заходи які потрібні та спрямовані для більш стабільного та надійного функціоналу комп'ютера, а також частини його додатків та програмного забезпечення. З досвідом системного адміністратора можу сказати, що до обов'язків та підтримки безпосередньої роботи техніки входить дуже багато: сканери, принтери, гарнітура, телефони обслуговування мережі та найголовніше це ПК. Ця техніка вимагає постійного догляду та контролю поточного стану. Від обслуговуючої організації може знадобитися оцінити ресурс, що залишився, провести при необхідності апаратний або програмний апгрейд, а також підібрати відповідну заміну одиниці обладнання, що вийшла з ладу.

Також сама інфраструктура потребує постійної уваги, так як з розвитком технологій постійно з'являються нові технології, які роблять використання програм та техніки більш ефективною. Підвищуються вимоги щодо інформаційної безпеки. Нові методи зламування мереж змушують проводити регулярні оновлення антивірусного програмного забезпечення. Перехід на хмарні технології потребує підтримки в актуальному стані системи розмежування прав доступу.

Також після обслуговування, апгрейду та ремонту техніки, звісно треба проводити її тестування а стабільності роботи операційної системи та всіх пристроїв комп'ютера.

Почну з першого виду обслуговування. Прогнозне обслуговування. Він використовується за допомогою діагностичних інструментів. Це потрібно для того, щоб передбачити можливі збої та спробувати уникнути їх до їх появи.

Другий вид обслуговування – профілактичне обслуговування – це найпопулярніший вид обслуговування, який полягає в запобіганні можливих збоїв та покращенні функціонування системи.

Третій вид – це корекційне обслуговування, що застосовується, коли система прогнозування і профілактики не працювала належним чином або не змогли уникнути аварії.

Четвертий вид – еволюційне обслуговування. Цей тип технічного обслуговування призначений не для виправлення або запобігання можливих збоїв, а для розробки доступних обчислювальних ресурсів.

Отже, було розглянуто питання обслуговування ПК та види, які воно має; визначено види та їхні функції. Також необхідно зазначити, що не слід забувати про оновлення драйверів, вони також відіграють важливу роль у функціоналі комп'ютера.

Література:

1. <https://www.mobik.ua/ukr/catalog/remcom/product.html?id=2199>
2. <https://www.intersys.com.ua/uk/support/it-maintenance>
3. <https://nit-services.com/chto-takoe-obslyuzhivanie-kompyuterov#:~:text=Обслуговування%20комп%27ютерів%20–%20це%20комплекс,додатків%2C%20периферійного%20та%20мережевого%20обла,данья>

4. <https://nit-services.com/obsluzhivanie-kompyuterov-kak-osnova-ih-bespreboynoy-raboty>

DATABASE CONNECTION SOLUTIONS IN SYS ENGINEERING

Chupakhin N.S

State University of Telecommunications, Kyiv.

Reveal most usable types of database (DB) connection solutions and assess relevance of each one in system engineering. Compare application and *operating system* (OS) layer DB connections. Describe in summary key features of DB connections, difficulty of implementation, auto-recovery of DB Interactions.

Aim of my research is to get into alternative DB connection solutions. Estimate how spread each method can be used. Analyze pros and cons of both application layer solutions and operation layer connections. Make a conclusion about alternative methods to extract store and extract data in cloud.

A database (DB) is an organized structure designed for storing, modifying and processing interrelated information, usually of large volumes.

Any modern architecture contains a database for the implementation of one of the tasks: long-term data storage, storage of system configurations (static and dynamic variables, authorization data, etc.), temporary data storage.

Basic interaction between databases, regardless of their type, is performed using "uri".

URI - a set of information that identifies the instance of a specific database. Each database (mysql, mssql, psql, mongo, oracle, etc.) has its own unique syntax for writing uri.

However, connections of data layer is not enough to implement complex systems. For more complex tasks, database interfaces were created that moved first step away from the system layer.

Examples of such interfaces include ODBC, OLE, ADO.NET.

ODBC - primarily an API, the interaction with configured by the database driver developers. The interaction is independent of the database type and operating system, and is technically performed between the database drivers and the application layer.

OLE – (Object Linking and Embedding) is an API developed by Microsoft to access various types of data that are stored in a single form. It is a set of interfaces implemented using the Component Object Model (COM); it was designed as a further development to replace and succeed ODBC, extending the feature set to support a wider range of non-relational data sources, such as object-oriented databases or spreadsheets, and for which SQL is not required.

ADO.NET- used to access data through .NET environments, allows manipulation of a dataset, which is a downloaded copy of a DB. Manipulated then synchronized with the master copy of the DB; this allows processing of data pulled

across the Internet, which is then placed into the original source.

JDBC - concerns the use of Java. Running a Java program works through a locally installed API: the application calls the API, which calls the JDBC driver, which calls appropriate middleware, which accesses the DB. Most popular connection of data layer, that finally can be fully used for middle layer processes.

Middle layer - where transformation and translation of the data request take place. May pass the request to a system, or may read the data directly.

This layer is necessary to work with web servers. Obviously, must to know methods to execute a transaction to DB in web is:

CGI (Common Gateway Interface) – which is a collection of scripts written in a programming language. Queries on the database path are passed by a special interpreter, which can manage the frequency of sending queries if necessary to reduce the traffic density. 1 query equals 1 connection to the DB.

Second web to database method involves the use of an API that communicates with DLLs (dynamic link libraries), thereby significantly speeding up database transactions. But the main feature of this method is the multiplexing of requests during 1 DB connection, which CGI cannot do.

Investigating the usage and relevance of mentioned technologies I came to next conclusion:

1) The most spread technology to work with data layer – is ODBC (mainly Windows based systems) and JDBC (mainly Linux, Mac based systems). Using one of the technologies, you can be assured of a renewable connection and have a wide range of data handling functionality in the application layer. I'm not prepared to say that using these methods is difficult to implement, but it is more labour intensive to build an architecture using JDBC than using middle layer methods.

2) Middle layer features are much more attractive. Although, their use is picking up, the possibilities are vastly superior to using a data layer, both in terms of speed and variety of data handling.

Current methods are widely spread in Cloud. I suppose that DB connectivity in Cloud is going to progress with more complex API, which is easy-to-integrate in application layer code.

To sum up, I highlighted key features of each mentioned technology: ODBC - maximum interoperability; OLE – easy-to-embed, ADO.NET – disabled and scalable architecture; JDBC – persistent connection, CGI – easy-to-integrate, convenient in automate tasks; API based – highest performance.

Literature:

1. Internet source.
<https://stevevincent.info/>
2. Internet source.
<https://www.sciencedirect.com/topics/computer-science/database-connectivity>
3. *Six-Step Relational Database Design*, Fidel A. Captain 2013.
4. Internet source.
<https://learn.microsoft.com/ru-ru/sql/>

АКТУАЛЬНІСТЬ КОНТЕЙНЕРИЗАЦІЇ ДОДАТКІВ ЗА ДОПОМОГОЮ DOCKER

Федоренко М.А, Бурлака В.М.

Державного університету телекомунікацій, м. Київ

Програмістам та фахівцям айтї індустрії необхідно швидко доставляти або переміщувати код, іноді навіть з середовищем. З цим завданням чудово справляється Docker.

В основі роботи Docker лежить стандартизований спосіб виконання кода. Він схожий на віртуальну машину яка створює віртуальне представлення апаратного забезпечення сервера, контейнери створюють віртуальне представлення серверної операційної системи. Але контейнери і віртуальні машини це різні способи віртуалізації. Віртуальна машина реалізує її на рівні заліза, а Docker на рівні операційної системи.[1] Якщо ціль віртуальної машини повністю відтворити пристрій комп'ютера, то основна мета Docker - створити середовище для однієї програми. Віртуальне середовище контейнера запускається всередині операційної системи. Саме зручне що їй не потрібно віртуалізувати пристрій, тому що вона використовує його через операційну систему. Тому контейнери споживають менше ресурсів, швидше розгортаються, простіше масштабуються і менше важать. Після установки на кожний сервер представляється доступ до простих команд, необхідних для збірки, запуску або зупинки контейнерів.[2]

Він може бути розвернутий налюбій Linux-системі, також видають набір команд для управління контейнерами.

В установку Docker входять наступні компоненти:

- Docker daemon. Служба яка управляє об'єктами, сітями, сховище, образами і контейнерами.
- Docker image. Це незмінний образ, з якого розгортається контейнер.
- Docker host. Це операційна система на яку встановлюють Docker.
- Docker client. Консольний клієнт за допомогою команд якого взаємодіють з Docker daemon.
- Docker Registry. Репозиторій в якому зберігаються образи.
- Docker Compose. Інструмент для управління декількома контейнерами.[3]

Використовуючи Docker ми здатні швидше та ефективніше доставляти або переміщати код, також в цілому економимо кошти, оптимізуючи використання ресурсів. Завдяки цьому користувачі з великою надійністю отримують об'єкт який можна запускати налюбій платформі.

Література:

1. Що представляє собою докер?[Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Docker>
2. Як використовують докер в розробці.[Електронний ресурс] – Режим доступу до ресурсу: https://eternalhost.net/blog/razrabotka/chto-takoe-docker?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com
3. Архітектура та компоненти. [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/253877/>

PYTHON – НАЙКРАЩІЙ ВИБІР ДЛЯ СУЧАСНОЇ РОЗРОБКИ ПО

Бородін Н.В, Лемешко А.В.

Державного університету телекомунікацій, м. Київ

Python на даний момент є найпоширенішою скриптовою мовою. Python став основною і практично єдиною скриптовою мовою, витіснивши інші. Якщо говорити про професії – існує цілий комплекс професій, пов'язаних з цією мовою програмування. Python виявився дуже зручним для старту, тобто щоб швидко зробити Proof of concept, тому багато інших професій стали брати Пайтон в якості свого інструменту.

Python швидко піднімається на перший план серед найпопулярніших мов програмування у світі. Постійне зростання популярності визначає Python як одну з найбільш затребуваних мов програмування 2020 року. Як ви можете скористатися популярністю Python? Використовуючи популярну мову, ви маєте набагато більший шанс знайти рішення будь-якої проблеми, з якою ви можете зіткнутися. Насправді, якщо ваша проблема є достатньо поширеною, зараз, мабуть, є готове рішення в Python. Python має велику спільноту ентузіастів, які щодня прагнуть покращити мову, виправляючи помилки та відкриваючи нові можливості. І, як згадувалося раніше, він також користується сильною підтримкою найбільших світових корпорацій. Один з них – Google. Вони активно працюють над путівниками, навчальними посібниками та іншими ресурсами, щоб отримати максимальну користь від Python.

На даний час, Python є одним з найкращих рішень для розробки, так як його постійно оновлюють і роблять більш придатним до використання у різних задачах. Потенціал цієї мови досить великий, і професії в яких він використовується з кожним днем стає все більше.

Література:

- 1 Моя думка про Python - Foxminded [Електронний ресурс]. Режим доступу: <https://foxminded.ua/moja-dumka-pro-python/>
- 2 Що таке Python (Пітон): переваги програмування на Python

ТЕСТУВАННЯ ПРОДУКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АБО Ж ПЕРФОРМЕНС ТЕСТУВАННЯ

Хамко А.С

Державного університету телекомунікації, м. Київ

Тестування продуктивності (Performance Testing) – це процес тестування програмного забезпечення для визначення його продуктивності. Іншими словами, це процес дослідження швидкості реакції додатку на зовнішні взаємодії з ним з різною інтенсивністю та характеру навантаження, його стабільності, визначення кількості споживаних ресурсів та решти атрибутів якості. Даний тип тестування проводиться для визначення недоліків системи та запобіганню їх негативного впливу на подальшу експлуатацію продукту в умовах використання.

Не дивлячись на однозначність даного терміну, сам вид тестування має свої підвиди, такі як:

1) Навантажувальне тестування (Load Testing) – тип тестування продуктивності, що виконується для оцінки якості поведінки модуля/лів чи системи при поступовому підвищенні навантаження на неї, наприклад, підвищення кількості активних користувачів на сайті або кількості проведених транзакцій на визначений період часу, для визначення об'єму навантаження, яке може обробити компонент або система.

2) Тестування ємності (Capacity Testing) – тип тестування, за допомогою якого визначається кількість користувачів/транзакцій може бути оброблена компонентом/ми/системою, при цьому не втрачаючи встановлених вимог до якості продуктивності ПЗ.

3) Тестування масштабованості (Scalability Testing) – дослідження можливості програмного забезпечення збільшувати показники продуктивності у відповідності до кількості нових доступних програмному забезпеченню ресурсів.

4) Об'ємне тестування (Volume Testing) – дослідження показників продуктивності програмного забезпечення при обробці різних (і, як правило, великих) за типом об'ємів даних.

5) Стресс тестування (Stress Testing) – тип тестування, який виконується для охарактеризування поведінки модуля чи системи під граничним чи заграничним рівнем очікуваного чи встановленого рівня робочого навантаження, або під впливом зменшення робочих ресурсів, таких як кількість пам'яті чи серверів.

6) Тестування стабільності або робота на відмову (Stability/Reliability Testing) – тип тестування, що проводиться з метою переконання, що ПЗ працює як належне під навантаженням протягом тривалого періоду часу. Під час даного тестування фіксується кількість споживаної пам'яті додатком, для

визначення потенційних втрат. Також, таке тестування виявляє деградацію продуктивності, яка виражається у підвищенні часу відгуку ПЗ (Delay) на запити та у зниженні швидкості обробки інформації, після довготривалого використання ПЗ.

7) Конкурентне тестування (Concurrency Testing) – дослідницьке тестування, метою якого є визначення поведінки ПЗ при великій кількості різних запитів (до БД, пам'яті, каналів передачі даних, дискової підсистему і т.д.), коли додатку треба розподіляти ресурси між запитами. Також, інколи під цим типом тестування трактують перевірку багатопотокових додатків та коректність синхронізації дій, що відбуваються у різних потоках.

8) Шип тестування (Spike Testing) – «неочікуване» тестування, що заключається у різкому та, відповідно, неочікуваному збільшенню навантаження умовним збільшенням кількості користувачів на невеликий проміжок часу для визначення вразливості продуктивності, чи «впаде» додаток або продуктивність критично впаде, або ж додаток витримає різке збільшення навантаження.

Цілі тестування продуктивності. Основними цілями тестування продуктивності можуть бути різні варіанти відповідей, так як все може залежати від проекту, його вимог і т.д., але в декілька загальних критеріїв можна взяти:

Час відповіді сервера. Цей атрибут продуктивності вказує на кількість часу, що потребує один вузол додатку для обробки запиту, надісланого іншим вузлом додатку. Як приклад, HTTP запит 'GET', який виконується браузером для звертання до веб-серверу. Більша частина програм, що призначені для навантажувального тестування, працюють саме за цією схемою вимірювання. В деяких випадках більш раціональним може бути досягнення продуктивності часу відповіді сервера серед всіх вузлів додатку.

Паралелізм/Пропускна здатність. Якщо це додаток, в якому користувач постійно приймає активну участь, використовуючи його, то в такому випадку, паралелізм – це показник максимальної кількості користувачів, що використовують додаток. Сама ж поведінка користувачів при використанні ПЗ може мати значний вплив на здатність додатку паралельно оброблювати запити, зокрема, якщо така поведінка включає в себе постійний вхід та вихід із системи. Якщо з додатком не працює конкретний кінцевий користувач, то сюди підійде термін пропускна здатність – характеристика продуктивності ПЗ, що показує максимальну пропускную здатність, кількість оброблених транзакцій/запитів за одиницю часу.

Час відображення. Останнє та одне з найскладніших для програми для навантажувального тестування понять, так як в загальному випадку вони не використовують концепцію роботи з тим, що відбувається на окремих вузлах системи, обмежуючись тільки розпізнаванням періоду часу протягом якого немає мережевої активності. Для того, щоб заміряти час відображення, в загальному випадку потрібно включати функціональні тестові сценарії в тести продуктивності, але більшість програм для тестування продуктивності не включають в себе таку можливість.

Підсумовуючи, можна сказати, що процес тестування продуктивності є невід’ємною частиною процесу тестування будь-якого додатку, так як він опрацьовує можливі ризики, що можуть призвести до, як мінімум, великих втрат під час експлуатації, як максимум – абсолютна непригодність продукту і його обезцінення, що несе за собою фінансові втрати неймовірних масштабів. Сам вид тестування є достатньо об’ємним, включає в себе багато різних підходів, нюансів та інструментів для праці з продуктивністю продукту і, як на мою думку, має бути виконий на всіх проєктах в тій чи іншій мірі. Важливістю виду є вище згадані втрати фінансів за рахунок поганого юзер-експієнсу чи, умовно кажучи, неправильно оброблені або взагалі не оброблені дані (наприклад, транзакції), що знищує репутацію продукту або навіть компанії, або взагалі повна нерентабельність продукту.

Література:

1. Книга «Тестування програмного забезпечення. Базовий курс (3-тє видання)», автор Святослав Куліков
2. Стаття «Тестування продуктивності»,
https://uk.wikipedia.org/wiki/Тестування_продуктивності
3. Стаття «Performance Testing»,
<https://www.guru99.com/performance-testing.htm>

СИСТЕМА МОНІТОРИНГУ МЕРЕЖІ ZABBIX

Сараненко А.Д.

Державного університету телекомунікацій, м. Київ

Zabbix - програмне забезпечення з відкритим кодом, що призначено для виявлення несправностей в мережі та автоматичного виправлення помилок.

Програма розповсюджується безплатно і не має окремої версії для великих бізнес-користувачів, проте компанія пропонує допомогу в інтеграції системи в вже готову мережу. На відміну від простіших рішень Zabbix виявляє не тільки відсутність підключення між елементами мережі, але й за допомогою Zabbix agent може отримувати та обробляти телеметрію з пристроїв.

Наприклад за допомогою Zabbix agent можна слідкувати за кількістю вільного місця на HDD дисках сервера, або відстежувати скільки користувачів в цей момент використовують RDP підключення. Таким чином можна не тільки виявити проблему, але й дізнатись можливу причину зламання.

Використовуючи графічний інтерфейс Zabbix є можливість:

- Побудови карти мереж на якій буде відображатись всі несправні елементи мережі.
- Побудова графіків роботи мережі.

Проте виявлення проблеми це лише перший етап, адже після виявлення проблеми необхідно щоб спеціалісти отримали повідомлення про цю проблему.

В Zabbix система сповіщень пророблена дуже добре. Усі проблеми автоматично отримують рівень важливості (залежно від пресетів налаштувань, або в залежності від вподобань адміністратора).

Після виявлення помилки система надсилає повідомлення обраним користувачам через вказані канали зв'язку:

- Електронна пошта
- СМС повідомлення
- Webhook (наприклад Telegram).

Є можливість обирати які користувачі отримують повідомлення в залежності від їх графіків роботи або від мінімального рівня важливості задачі.

Таким чином спеціаліст отримує інформацію про проблему і може приступити до її вирішення, але не кожна проблема достатньо значуща, щоб витратити на неї дорогоцінний час спеціалістів. Тому в Zabbix є автоматизація розв'язання проблем. Для цього при виявленні проблеми Zabbix може виконати вказаний скрипт який, наприклад перезавантажить службу в якій виявлено проблему.

Таким чином Zabbix є не лише системою моніторингу, але й інструментом що на базовому рівні може виконувати певну автоматизацію для розв'язання типових проблем в мережі. Також через те що це програма з відкритим вихідним кодом є багато користувацьких розширень що додають нові можливості. Якщо, наприклад адміністратора не влаштовує як Zabbix графічно відображає дані про мережу можна під'єднати Grafana і більш гнучко підлаштувати елементи під свої потреби.

Одже Zabbix є потужним і достатньо гнучким інструментом, що може закрити більшість потреб в моніторингу мережі. На цей момент актуальною версією є 6.2, але вже є dev версія 6.4. З цього можна зробити висновок що програма активно оновлюється і впроваджує нові можливості.

Література:

- 1 Zabbix [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.zabbix.com>
- 2 Zabbix Manual [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.zabbix.com/documentation/current/en/manual>

НАПРЯМ 2. БАЗИ ДАНИХ ТА ЗНАНЬ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА ОБРОБКА ІНФОРМАЦІЇ

РОЗРОБКА WEB-ПЛАТФОРМИ ДЛЯ АГРЕГАЦІЇ ДАНИХ

Лебідь М.М., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

В сучасному світі ми маємо доступ до великої кількості інформації з різних джерел, таких як веб-сторінки, соціальні мережі, новинні сайти та багато інших. Агрегація даних є процесом збору даних з різних джерел та об'єднання їх в єдину базу даних, що дозволяє зберігати та аналізувати великі обсяги інформації.

Однак, ручне збирання даних з різних джерел може бути дуже складним та тривалим процесом. У таких випадках, web-платформа для агрегації даних може бути дуже корисною.

У темі розробки web-платформи для агрегації даних, що буде базуватися на базах даних та знань, інтелектуальному аналізі та обробці інформації. Основна мета цієї платформи - збирати дані з різних джерел та зберігати їх у єдиній базі даних, що дозволить ефективно аналізувати та використовувати їх.

Також, ця платформа буде включати інтелектуальний аналіз та обробку інформації. Наприклад використання аналітичних методів, такі як статистичний аналіз, класифікація та кластеризація, щоб отримати більше інформації про дані.

Одна з ключових переваг web-платформи - це зручний та легкий доступ до великої кількості даних з різних джерел в одному місці. Користувачі зможуть швидко та ефективно знайти необхідну інформацію та зберігати її для подальшого використання.

Загалом, цей проект зосереджений на розробці web-платформи, яка буде збирати, обробляти та аналізувати дані з різних джерел. Ми будемо використовувати різноманітні технології та методи для забезпечення якісної та ефективної роботи платформи. Користувачі зможуть отримувати доступ до великої кількості інформації та зручно зберігати та використовувати її для своїх потреб.

Література:

- 1.Data aggregation. [Електронний ресурс]. – Режим доступу:
<https://www.techtarget.com/searchdatamanagement/definition/data-aggregation>
- 2.What is Data Analytics. [Електронний ресурс]. – Режим доступу:
<https://careerfoundry.com/en/blog/data-analytics/what-is-data-analytics/>

АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЙ ІОТ ДЛЯ РЕАЛІЗАЦІЇ ПЛАТФОРМ З ПЛАНУВАННЯ ПОДОРОЖЕЙ

Юхта М.А., Бондарчук А.П.

Державного університету телекомунікацій, м. Київ

Подорожі завжди були важливою частиною нашого життя. В останні роки з'явилась можливість використання Інтернету речей (ІоТ) для поліпшення та спрощення нашого досвіду під час подорожей. У цьому

контексті, головною проблемою стає пошук цікавих місць, що можуть відповідати потребам та бажанням подорожуючих. Одним з можливих рішень цієї проблеми є використання ІоТ для пошуку цікавих місць.

Метою цього дослідження є вивчення ролі ІоТ у пошуку цікавих місць для подорожей та дослідження того, як ІоТ може допомогти підвищити якість досвіду подорожі для користувачів.

ІоТ може відігравати важливу роль у пошуку цікавих місць для подорожей. Споживачі можуть використовувати пристрої ІоТ, такі як смартфони та сенсори, для збору та аналізу інформації про різні місця.

Наприклад, ІоТ може забезпечувати інформацію про кількість людей, що відвідали певне місце, настрої та враження попередніх відвідувачів. Ця інформація може бути корисною для подорожуючих, які шукають місця зі сприятливою атмосферою та без зайвого натовпу людей.

Крім того, ІоТ може допомогти в розробці персоналізованих рекомендацій для користувачів, враховуючи їхні індивідуальні потреби та бажання. Наприклад, користувачі можуть отримувати рекомендації щодо місць, які відповідають їхнім індивідуальним інтересам, які були визначені за допомогою даних, зібраних з пристроїв ІоТ.

Також, ІоТ може допомогти у виявленні нових туристичних маршрутів та

цікавих місць, що можуть бути менш відомими для подорожуючих. Наприклад, за допомогою сенсорів, зібраних з пристроїв ІоТ, можна виявити певні місця з надзвичайною красою або цікавими історичними фактами, які досі не входять до туристичних маршрутів.

За допомогою ІоТ, подорожуючі можуть отримати більш

персоналізований та зручний досвід під час пошуку та вибору цікавих місць. Зібрана інформація та рекомендації можуть бути корисними для розробки нових туристичних маршрутів та покращення якості обслуговування у туризмі. Однак, важливо пам'ятати про захист особистої інформації користувачів та не допускати її неправомірного використання.

Отже, використання IoT для пошуку цікавих місць у туризмі може забезпечити значний вплив на задоволення туристів та збільшення прибутку туристичної галузі. Інтернет речей може допомогти в ідентифікації і прогнозуванні популярних туристичних місць, а також в забезпеченні персоналізованих рекомендацій та високоякісних послуг для туристів. Науковці та практики вже розробляють й імплементують різноманітні застосунки IoT в туризмі для покращення якості послуг та збільшення ефективності туристичної галузі.

Література:

1. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.
2. Zhang, X., Li, Y., & Zhang, J. (2019). Research on personalized recommendation system of tourist attractions based on big data. *Cluster Computing*, 22(1), 2317-2325.
3. Karpowicz, M., & Kukla, A. (2017). The internet of things in tourism industry. *Tourism Management Perspectives*, 23, 1-7.
4. Bellini, E., & Nesi, P. (2018). Internet of things in tourism: From research to practice. *Journal of Hospitality and Tourism Technology*, 9(4), 520-529.

РОСКЕТРЕННУ: ЯК МОБІЛЬНИЙ ДОДАТОК ДОПОМАГАЄ ЛЮДЯМ КЕРУВАТИ СВОЇМИ ФІНАНСАМИ

Диндар А.В., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

На сьогоднішній день все більше людей хочуть бути більш досвідченими та свідомими у своїх фінансах, тому приділяють увагу до особистих витрат, прибутків та заощаджень. В такій відповідальній справі важко утримати все в голові, тому попит на програмні продукти для фінансових задач стає частішим.

У сучасному світі багато людей мають проблеми з керуванням своїми фінансами. Це призводить до того, що вони можуть опинитися в ситуації, коли їм бракує коштів на будь-що. Також через невміння керувати своїми фінансами люди часто відмовляються від речей, які вони хочуть придбати, або обмежені в подорожах та своєму духовному розвитку.

Також, невміння керувати своїми фінансами створює суттєву перешкоду на шляху досягнення своїх матеріальних мрій. Іноді люди, можуть опинитися

у боргах і мати проблеми з їх погашенням, і в такому випадку про досягнення мрій мови йти не може.

Але що, якщо у людей буде додаток на смартфоні, який допоможе відстежувати і контролювати їх фінанси? Тоді вони зможуть легше досягнути мети щодо заощаджень і поліпшувати своє фінансове становище. RocketPenny - мобільний додаток, розроблений мовою програмування Python, який може стати надійним помічником в керуванні фінансами. Він дозволяє відстежувати витрати і прибутки, розподіляти кошти на різні категорії, встановлювати бюджет і контролювати його дотримання.

На відміну від додатків конкурентів на ринку RocketPenny відрізняється від них своєю унікальною функціональністю, що враховує досвід користувачів попередніх додатків, а також інноваційними можливостями, які роблять керування фінансами ще більш зручним та ефективним.

Таким чином, RocketPenny може бути ідеальним варіантом для тих, хто хоче навчитися керувати своїми фінансами і досягати фінансової стабільності.

Література:

1. Сайт PDFdrive [Електронний ресурс] – <https://www.pdfdrive.com/rich-dad-poor-dad-e136494023.html>.
2. Сайт PDFdrive [Електронний ресурс] – <https://www.pdfdrive.com/your-money-or-your-life-9-steps-to-transforming-your-relationship-with-money-and-achieving-financial-independence-revised-and-updated-for-the-21st-century-e159459889.html>.

ВИКОРИСТАННЯ ЗАСОБІВ МОВИ PYTHON ДЛЯ ВИРІШЕННЯ ЗАДАЧ ОБРОБКИ ТЕКСТУ

Лукашенко І.П., Золотухіна О.А.

Державного університету телекомунікацій, м.Київ

Python є однією з найпопулярніших мов програмування, що використовуються для розробки програмного забезпечення в різних галузях. Зокрема, мова Python має безліч бібліотек та модулів, які допомагають вирішувати різноманітні задачі, включаючи автоматизовану побудову тематичного словника.

Для створення програмного забезпечення для автоматизованої побудови тематичного словника можна використовувати такі бібліотеки, як NLTK (Natural Language Toolkit), SpaCy та Gensim. NLTK є однією з найбільш популярних бібліотек для обробки природних мов, яка надає функціонал для токенізації тексту, лематизації та побудови тематичних моделей. SpaCy є іншою потужною бібліотекою для обробки природних мов, яка забезпечує високу швидкість та точність в обробці тексту. Gensim є бібліотекою, яка спеціалізується на побудові тематичних моделей та

забезпечує ефективну реалізацію алгоритмів для знаходження схожих слів та побудови синонімів.

Одним з прикладів програмного забезпечення для автоматизованої побудови тематичного словника мовою Python є "Topic Modelling Toolkit" (tmtoolkit). Цей інструментарій використовує бібліотеку SpaCy та Gensim для побудови тематичної моделі та надає функціонал для візуалізації тем та слів, а також для аналізу тематичних моделей.

Бібліотека SpaCy дозволяє ефективно виконувати обробку тексту завдяки використанню унікальних алгоритмів обробки тексту. SpaCy використовує високооптимізований пайплайн обробки тексту, що дозволяє працювати з великими об'ємами даних швидко та ефективно. SpaCy надає можливість використовувати власні моделі машинного навчання, що дозволяє підлаштовувати інструмент для конкретних завдань та покращувати його точність.

Pipeline - це концепція інтеграції різних компонентів обробки тексту разом таким чином, що вихід одного компонента служить входом для наступного компонента. Spacy надає вбудовану функціональність пайплайнів, яку можна досить легко налаштувати. Пайплайн, за замовчуванням, складається з таких компонентів, як теггер, лемматизатор, парсер і засоби розпізнавання сутностей. При виклику nlp пайплайну створюється об'єкт Doc, який створюється шляхом токенизації тексту, потім цей документ обробляється в декілька етапів. По суті це набір токенів, тобто слово, розділовий знак, пробіл тощо.

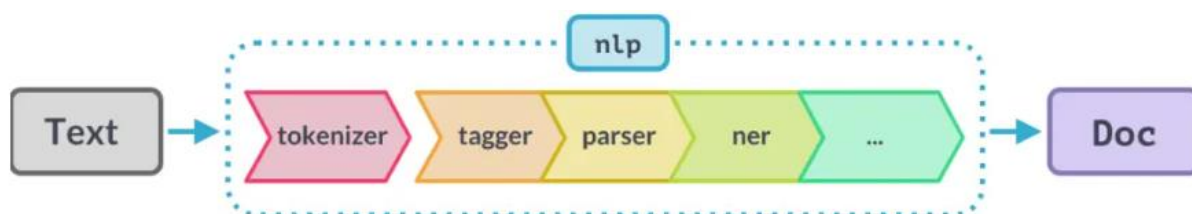


Рис. 1 Компоненти SpaCy NLP pipeline

Можливості пайплайну завжди залежать від компонентів, їх моделей і того, як вони були навчені. Наприклад, конвеєр для розпізнавання іменованих сутностей повинен включати навчений компонент розпізнавання іменованих сутностей зі статистичною моделлю та ваговими коефіцієнтами, які дозволяють робити прогнози міток сутностей.

Література:

1. spacy [Електронний ресурс] : [Веб-сайт] – Режим доступу <https://spacy.io/api/doc>
2. tmtoolkit [Електронний ресурс] : [Веб-сайт] – Режим доступу <https://tmtoolkit.readthedocs.io/en/latest/>

3. Top 7 Python NLP Libraries [And Their Applications in 2023] [Електронний ресурс] : [Веб-сайт] – Режим доступу <https://www.upgrad.com/blog/python-nlp-libraries-and-applications/>

ВИКОРИСТАННЯ JPA ТА HIBERNATE ДЛЯ СПРОЩЕННЯ РОБОТИ З БАЗАМИ ДАНИХ У ВЕБ ДОДАТКАХ

Левченко О.О., Трінтіна Н.А.

Державний університет телекомунікацій, м. Київ

На сьогоднішній день майже кожен веб додаток при розробці не обходиться без використання баз даних. Це стосується і великих і малих проектів. Саме для полегшення роботи з ними було розроблено JPA (Java Persistence API) та Hibernate. Вони є потужними інструментами для спрощення роботи з реляційними базами даних у веб додатках. За допомогою цих інструментів можна легко і швидко зберігати, зчитувати, оновлювати та видаляти дані з баз даних, не потребуючи написання складних SQL-запитів.

JPA є однією з специфікацій мови програмування Java для роботи з об'єктно-орієнтованими даними в базі даних. Виникла вона через велику популяризацію вільного ORM фреймворку Hibernate. Описує систему управління зберігання даних об'єктів в таблиці баз даних. Для цього використовує анотації Java для мапінгу об'єктів на таблиці баз даних. JPA надає уніфікований підхід для роботи з реляційними базами даних, що дозволяє розробникам використовувати один API незалежно від конкретної реалізації бази даних.

Hibernate найпопулярніша реалізація JPA. Вона є потужною технологією, що дозволяє розробникам швидко створювати та змінювати схему бази даних, зберігаючи при цьому об'єкти в пам'яті JVM (Java Virtual Machine). Hibernate забезпечує зв'язок об'єктів з таблицями баз даних, використовуючи різні стратегії мапінгу. Крім того, Hibernate підтримує різні бази даних, що дозволяє розробникам легко переносити додатки на іншу базу даних.

У підсумку, використання технологій JPA та Hibernate дозволяє розробникам зосередитись на логіці веб додатка, звільнити розробника від значних типових завдань із програмування взаємодії з базою даних, що спрощує роботу з реляційними базами даних. Написання коду стає більш зрозумілим та читабельним, що сприяє швидкому розвитку та підтримці

проектів надалі.

Література:

1. Spring Data JPA - Reference Documentation. [Електронний ресурс]. – Режим доступу: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>
2. Hibernate ORM. [Електронний ресурс]. – Режим доступу: <https://hibernate.org/orm/>

ПЕРЕВАГИ ВИВЧЕННЯ ДИСЦИПЛІНИ "АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ" ЗА ДОПОМОГОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Комаров О. В., Негоденко О.В.

Державний університет телекомунікацій, м. Київ

Знання алгоритмів та структур даних є необхідною складовою професійної компетентності програміста. Для розробки ефективного та швидкого програмного забезпечення вміння обирати доцільний алгоритм дій та використовувати відповідні структури даних є незамінними.

Структури даних і алгоритми допомагають розуміти, як працюють системи, додатки та програми, та як можна покращити їх ефективність.

Крім того, знання алгоритмів і структур даних дозволяє підвищити ефективність роботи програміста, розвинути аналітичні здібності та алгоритмічне мислення. Знання загальних принципів алгоритмів і структур даних допомагає програмісту ухвалювати правильні рішення та обирати для розв'язання завдань найбільш оптимальні інструменти.

Використання програмного забезпечення для вивчення будь-якої дисципліни має багато переваг. Воно дозволяє зробити процес вивчення цієї дисципліни простішим, цікавішим. [2] А також покращує організацію дистанційного та індивідуального навчання. Та допомагає організувати наочний контроль за проходженням навчання для студентів та викладачів. [1]

Крім того, це покращує рівень інформаційної компетентності викладачів та студентів, та підвищує ефективність їх взаємодії. [3]

Програмний продукт створюється на базі курсу "Алгоритми та Структури даних" Державного університету телекомунікацій.

Програма має модульну структуру, так що можна додавати навчальні модулі.

Кожна із розглянутих тем складається з таких розділів:

- Теорія
- Приклади
- Завдання
- Аналіз результатів

Перевагою даного продукту являється можливість використовувати його різними мовами.

Література:

[1] – <https://www.researchgate.net/post/What-are-the-benefits-of-using-educational-software-in-schools>

[2] – <http://www.vkmaheshwari.com/WP/?p=236>

[3] – <https://medialiteracy.org.ua/vykorystannya-suchasnyh-osvitnih-instrumentiv-dlya-pidvyshhennya-rivnya-tsyfrovoyi-kompetentnosti-pedagoga-nush/>

ВИКОРИСТАННЯ WEBSOCKET ДЛЯ КЛІЄНТ-СЕРВЕРНОЇ КОМУНІКАЦІЇ

Шовтенко В.Ю.

Державного університету телекомунікацій, м.Київ

Відповідно до загальноприйнятого визначення, WebSocket - це протокол з двонаправленою, повнодуплексною комунікацією між клієнтом і сервером, який використовується відмінно від HTTP і починається з ws:// або wss://. Цей протокол зберігає стан з'єднання, що означає, що з'єднання між клієнтом і сервером буде активним, поки не буде розірвано однією зі сторін (клієнтом або сервером). Після закриття з'єднання клієнтом або сервером, з'єднання відмовляється з обох сторін.

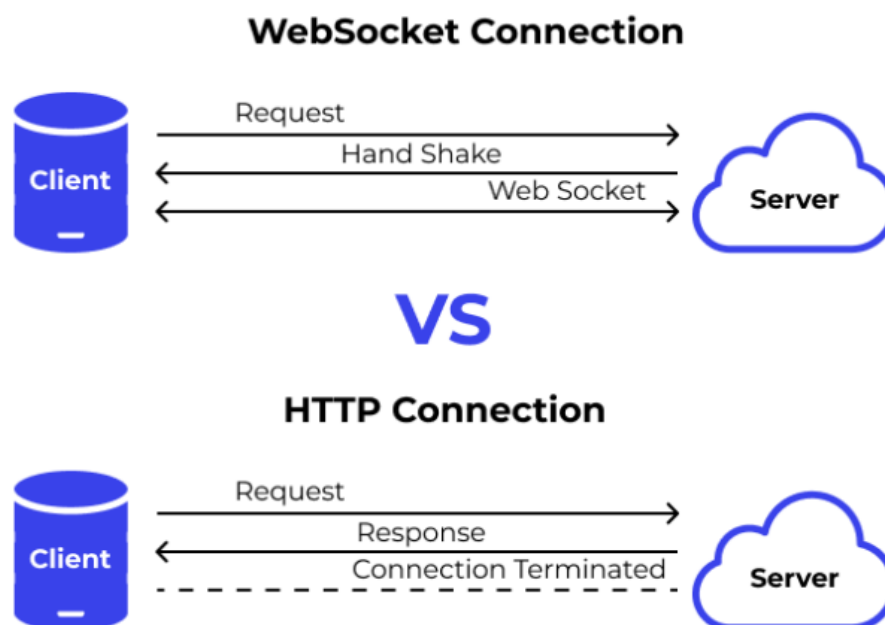


Рис. 1 Порівняння HTTP з'єднання та WebSocket з'єднання
WebSockets використовуються для таких додатків як:

- Веб-додаток у реальному часі, як приклад додатки де для відображається коливання цін на товар/валюту, постійно надсилаються сервером на клієнтську сторону.
- Ігри: в ігрових додатках де дані постійно надходять на сервер, інтерфейс користувача оновлюється автоматично, не встановлюючи нове з'єднання, що сильно знижує навантаження на сервер. Особливо якщо клієнтів декілька десятків тисяч.
- Чати: чати використовують WebSockets для встановлення з'єднання лише один раз для обміну, публікації та трансляції повідомлення серед підписників. Він повторно використовує те саме підключення WebSocket для надсилання та отримання повідомлення та для передачі повідомлення один на один.

Література:

1. A simple explanation o what a Websocket is [Електронний ресурс]. – Режим доступу: <https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is>
2. What is web socket and how it is different from the HTTP? [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>

REDUX TOOLKIT ЯК СПОСІБ ЗБЕРІГАННЯ СТАНУ У ВЕБ-ДОДАТКАХ

Шовтенко В.Ю.

Державного університету телекомунікацій, м.Київ

Redux Toolkit - це набір інструментів, який допомагає ефективно розробляти застосунки з використанням Redux. React і Redux вважаються найкращою комбінацією для керування станом великих React-програм. Однак популярність Redux знизилася через складність його налаштування.

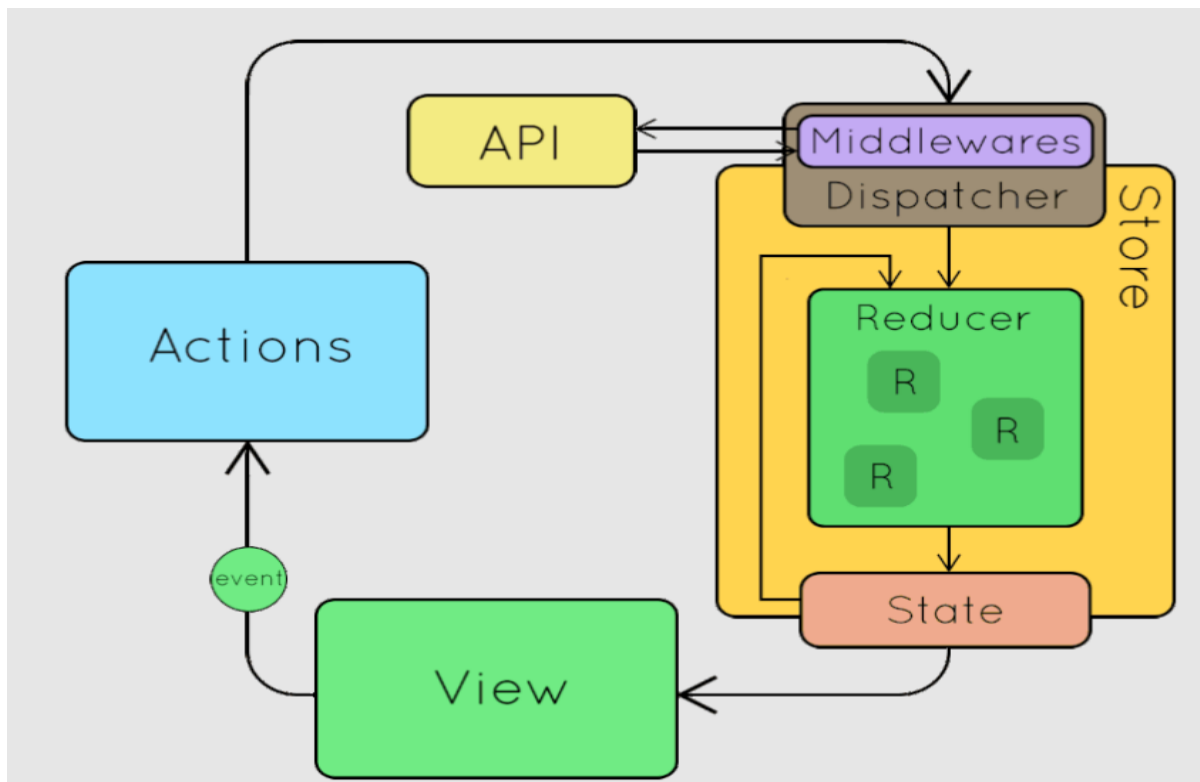


Рис. 1 Потік даних у Redux (data flow)

Redux Toolkit надає параметри для конфігурації глобального сховища, а також створення дій (actions) та редукторів (reducers) більш оптимізованих, шляхом абстрагування від Redux API, наскільки це можливо. З мого досвіду, Redux Toolkit є відмінним вибором для початку роботи з Redux. Він спрощує код і допомагає керувати станом Redux, зменшуючи кількість шаблонного коду.

Крім того, Redux Toolkit може бути використаний не лише з React, а й з іншими фреймворками, такими як Angular.

Література:

3. Simplifying Redux with Redux Toolkit [Електронний ресурс]. – Режим доступу: <https://blog.bitsrc.io/simplifying-redux-with-redux-toolkit-6236c28cdfcb>
4. Redux-toolkit docs [Електронний ресурс]. – Режим доступу: <https://redux.js.org/redux-toolkit/overview>

ВИКОРИСТАННЯ NOSQL БАЗ ДАНИХ ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ

Зайцев І.С., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

При розробці застосунку який повинен масштабуватися виникає

проблема зберігання даних.

При використанні реляційних баз даних (SQL) виникає проблема що більшість із них не підтримують масштабування та розподілення даних між серверами.

SQL бази даних гарантують ACID та доступність (Availability) але вони не гарантують поділу (Partitioning).

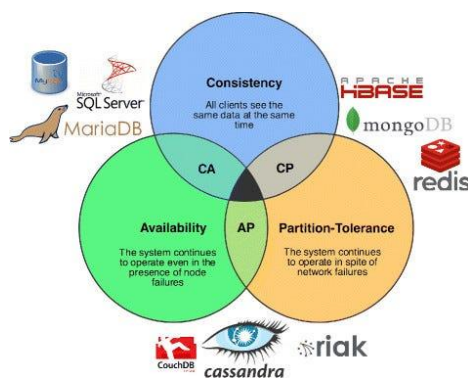


Рисунок 1 - Теорема CAP

SQL

Atomicity - база даних гарантує те що при використанні транзакції часткового результату збережено не буде, вся транзакція виконається повністю, або всі дані не будуть збережені.

Consistency - база даних гарантує що не буде збережено невідповідних даних які порушують обмеження таблиць (constraint), відносини між таблицями (primary-foreign keys).

Isolation - при виконанні транзакцій паралельно база даних гарантує що при використанні різних ступенів ізоляції одна транзакція не повинна повпливати на другі якщо вона не була збережена.

Існують такі рівні ізоляції транзакцій:

Кожен із рівнів вирішує різні проблеми (Dirty Reads, Nonrepeatable Reads, Phantom Reads) які можуть виникнути при паралельному виконанні транзакцій.

- 1) READ UNCOMMITTED
- 2) READ COMMITTED
- 3) REPEATABLE READ

4) SERIALIZABLE

Durability - база даних веде лог транзакцій та гарантує якщо раптово станеться помилка системи то збережені транзакції не буде втрачено.

NoSQL

NoSQL бази даних підтримують поділ між вузлами та масштабування, але в основному в них немає підтримки ізоляції транзакцій та в деяких базах взагалі немає підтримки консистенції даних.

Різні NoSQL бази даних імплементують різні алгоритми зберігання та обробки даних. Немає універсального інструменту який би можна було використовувати для всіх задач. При підборі бази даних потрібно ретельно обирати під які задачі база даних підходить краще за все.

Основні переваги NoSQL баз даних:

1) Partitioning

Дані зберігаються на диску в залежності від розділу, це дає змогу зберігати велику кількість даних, а також швидко отримувати дані по розділу.

Наприклад як розділ можна використовувати країну або регіон, та зберігати користувачів на різних розділах диску.

2) Sharding

Алгоритм зберігання Sharding схожий з Partitioning але дані зберігаються не на одному диску та на одному сервері, а на різних серверах та на різних розділах диску.

Це дає змогу розподілити навантаження на різні вузли бази даних для розділів.

3) Fault tolerance

Так як NoSQL бази даних мають змогу масштабуватися то навіть якщо один із вузлів вийде з ладу та буде не доступний база даних може продовжувати працювати без проблем доступності.

Висновок

Враховуючи зазначене, для забезпечення реалізації горизонтального масштабування при створенні дистрибутивної системи виконання відкладених

завдань було вибрано базу даних класу NoSQL. Із переліку NoSQL баз даних було вибрано базу даних MongoDB.

Література :

1. SQL ACID. [Електронний ресурс]. – Режим доступу:
<https://en.wikipedia.org/wiki/ACID>
2. NoSQL. [Електронний ресурс]. – Режим доступу:
<https://en.wikipedia.org/wiki/NoSQL>

РОЗРОБКА ПРОФАЙЛЕРУ ПРОДУКТИВНОСТІ SQL-ЗАПИТІВ ЗА ДОПОМОГОЮ СТРЕС-ТЕСТУВАННЯ

Бондаренко В.В., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

Тестування продуктивності SQL-запитів є безцінною практикою, яка може принести значні переваги розробнику, підчас тестування написаних запитів. Його можна використовувати для різних цілей, включаючи: продуктивність виконання запитів, планування потужності та виявлення вузьких місць.

Стрес-тестування запиту, яке також іноді називають навантажувальним тестуванням, передбачає виконання запитів з високою частотою із різних з'єднань або потоків із базою даних. Мета полягає в тому, щоб визначити, наскільки добре сервер бази даних справляється з високим стресом або високим навантаженням виконуваного запиту.

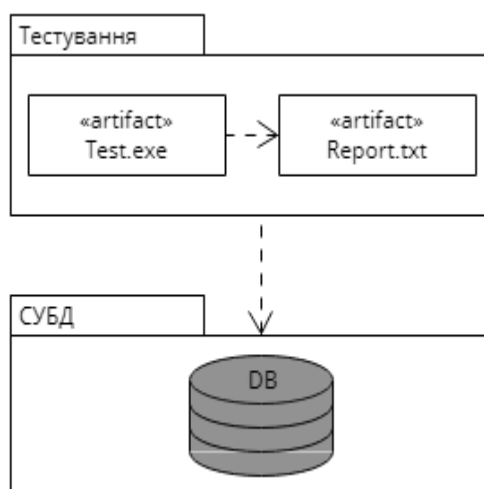


Рисунок 1.1 — Стрес-тестування запиту

Таке тестування дає картину, наскільки оптимально будуть виконуватись ті чи інші запити, в умовах приближених до реальних, коли сервер баз даних може обробляти циклічні запити, які надходять від великої кількості користувачів одночасно.

Виконання навантажувального тестування передбачає більше ніж просто здійснювати запитів до сервера баз даних. Також таке тестування повинно надавати розробнику статистику щодо продуктивності виконання навантажування, таку як: фактичний час виконання запиту, середній час виконання команди, медіана, дисперсія, кількість читань з бази даних, процесорний час та інше. Статистика може надаватись у виді числових значень, діаграм, графіків. У ході тестування слід аналізувати дані, отримані в результаті виконання кожного сценарію. Необхідно враховувати час виконання команди, кількість оброблених рядків, обсяг оперативної пам'яті, що використовується, та інші показники, які передбачені розробниками програми стрес-тестування запиту. Важливо переконатися, що SQL-команда правильно працює в умовах високого навантаження і не призводить до витоків пам'яті або збоїв у системі. Також варто звернути увагу на оптимізацію запиту та спробувати покращити його продуктивність, якщо це необхідно.

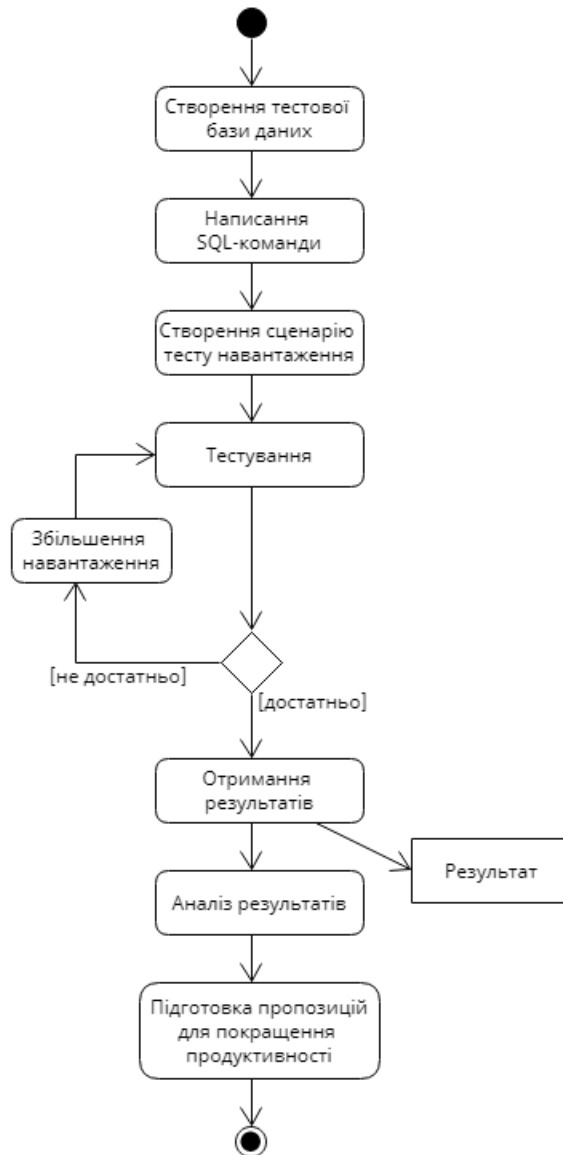


Рисунок 1.2 — Тестування SQL-команди

Процес стрес-тестування SQL-команди може виглядати так (Рис. 1.2):

- Створення тестової бази даних з великою кількістю даних, які можуть бути створені за допомогою різних інструментів.
- Написання SQL-команди, яку потрібно протестувати.
- Створення сценарію тесту навантаження за допомогою вибраного інструменту.
- Запуск тесту із поступовим збільшенням навантаження на базу даних.
- Аналіз результатів тестування.

В результаті стрес-тестування можна виявити можливі вузькі місця в роботі бази даних та оптимізувати SQL-команду для покращення її продуктивності.

Загалом, стрес-тестування запитів SQL є важливим кроком у забезпеченні продуктивності та масштабованості вашої системи баз даних.

Виявляючи та вирішуючи проблеми продуктивності на ранній стадії, можна уникнути простоїв і забезпечити кращий досвід роботи для своїх клієнтів.

Література:

1. SQL Query Stress Tool. [Електронний ресурс]. – Режим доступу: <https://www.mssqltips.com/sqlservertip/2730/sql-query-stress-tool/>
2. How to Fake Load Tests. [Електронний ресурс]. – Режим доступу: <https://www.brentozar.com/archive/2015/05/how-to-fake-load-tests-with-sqlquerystress/>
3. SQL Performance Testing Tools. [Електронний ресурс]. – Режим доступу: <https://www.dnsstuff.com/sql-performance-testing>
https://uk.wikipedia.org/wiki/Entity_Framework

ВИКОРИСТАННЯ БІБЛІОТЕКИ SCOTTPLOT.WPF ДЛЯ СТВОРЕННЯ ГРАФІКІВ ПРИ РОЗРОБЦІ ПРОФАЙЛЕРУ ПРОДУКТИВНОСТІ SQL-ЗАПИТІВ ЗА ДОПОМОГОЮ СТРЕС-ТЕСТУВАННЯ

Бондаренко В.В., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

При розробці профайлеру продуктивності SQL-запитів на платформі .NET було вирішено надати користувачу статистику виконання запитів в графічному форматі. Можливий варіант відображення проілюстровано на рисунку 1.1.

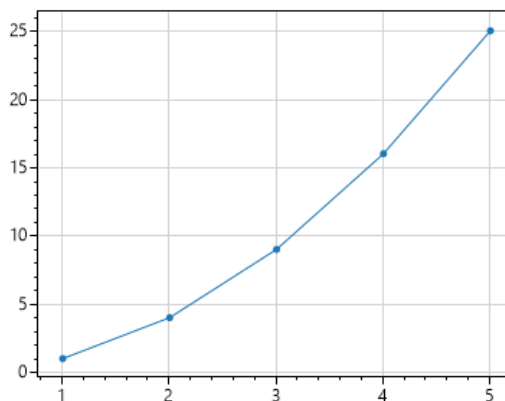


Рисунок 1.1 — Приклад графіку

Надання такої функціональності вирішено було реалізувати використовуючи бібліотеку ScottPlot.WPF.

ScottPlot.WPF — це бібліотека, яка надає прості у використанні інструменти для створення інтерактивних графіків у програмах WPF. Вона побудована на основі бібліотеки ScottPlot, яка є популярною бібліотекою візуалізації даних для .NET. Ця бібліотека дозволяє розробникам створювати лінійні діаграми, точкові діаграми, стовпчасті діаграми та інші типи діаграм лише за допомогою кількох рядків коду.

Також вона надає ряд функцій, які спрощують налаштування зовнішнього вигляду графіків, включаючи можливість змінювати кольори, шрифти та мітки.

Окрім функцій налаштування, ScottPlot.WPF також надає ряд інтерактивних функцій, включаючи можливість масштабування та панорамування, вибору точок даних і відображення підказок. Ці функції дозволяють користувачам легко досліджувати та аналізувати дані безпосередньо на графіку. На рисунку 1.2 наведено приклад використання цієї бібліотеки при розробці додатку.

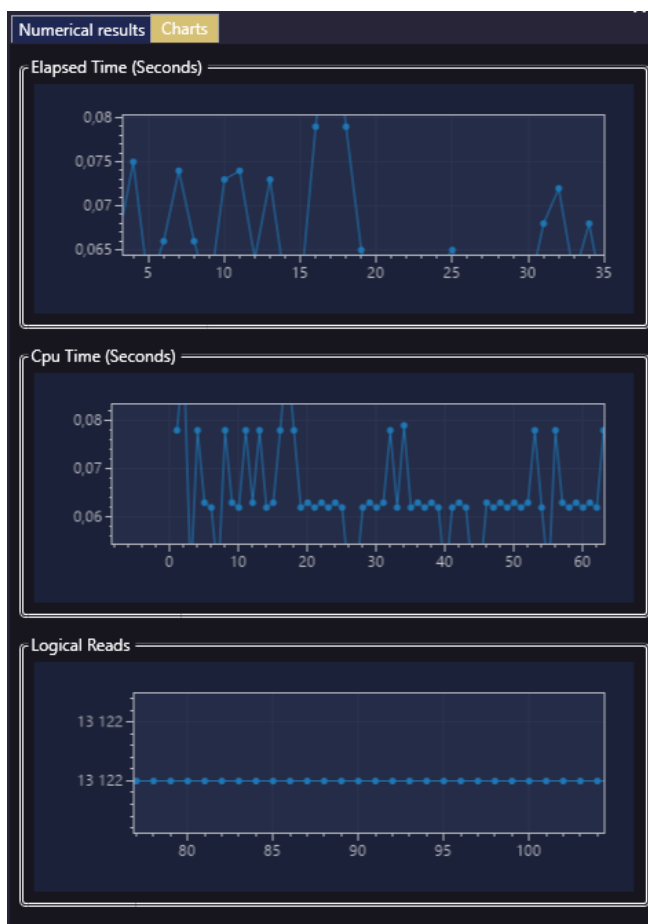


Рисунок 1.2 — Приклад використання бібліотеки

Загалом, ScottPlot.WPF — це потужний і гнучкий інструмент для створення високоякісних візуалізацій даних у програмах WPF. Його простота використання та можливості налаштування роблять його популярним вибором для розробників, яким потрібно швидко й ефективно створювати професійно виглядаючі графіки.

Література:

1. ScottPlot 4.1 Cookbook. [Електронний ресурс]. – Режим доступу: <https://scottplot.net/cookbook/4.1/#quickstart>
2. ScottPlot. [Електронний ресурс]. – Режим доступу: <https://github.com/ScottPlot/ScottPlotsqlquerystress/>

НАПРЯМ 3. СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА НЕЙРОННІ МЕРЕЖІ

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ CLOUD-TO-CLOUD У РОЗУМНИХ БУДИНКАХ

Сарафанюк Р.О., Куфтеріна С.Р.

Державного університету телекомунікацій, м. Київ

Результати досліджень. Що ж таке технологія Cloud-to-cloud? Cloud-to-Cloud (C2C) - це технологія, яка дозволяє різним хмарним сервісам/середовищам обмінюватись даними між собою безпосередньо. У використанні з розумним будинком, C2C може створити зв'язок між різними пристроями та системами управління, що поєднує їх у сітку та дозволить маніпулювати потрібним пристроєм використовуючи один застосунок [1].

Наприклад, у розумному будинку може бути встановлено декілька систем управління, які працюють на різних платформах. Використовуючи C2C, ці системи мають можливість обмінюватись даними зволікаючи різністю своїх виробників, що забезпечує зручність та ефективність використання майбутніх та теперішніх технологій. У зв'язку з цим, користувачу постає можливість віддаленого керування системою використовуючи інтернет.

Також, одним з юзкейсів є збереження даних, наприклад, зберігати дублікати своїх даних на різних серверах для збільшення процента залишення своїх даних цілими [2].

Одним за наочних варіантів використання є рішення від Google під назвою "Google Smart house". Google Smart House - це інтегрована система управління розумним будинком від компанії Google. Ця система дозволяє зібрати всі розумні пристрої під одним керуванням, щоб забезпечити користувачам зручний та легкий спосіб керування своїм домом.

Google Smart House використовує голосове керування, мобільний додаток та інші інтерфейси, щоб дозволити користувачам контролювати свої розумні пристрої з будь-якого місця. Система підтримує пристрої, які включають в себе електричні розетки, пульт дистанційного керування, термостати, системи безпеки, освітлення та багато іншого.

Google Smart House також інтегрується з іншими популярними голосовими асистентами, такими як Google Assistant, Amazon Alexa та Apple HomeKit. Це означає, що користувачі можуть взаємодіяти зі своїм розумним будинком за допомогою голосових команд на своєму смартфоні або голосовому асистенті, що дозволяє зробити досвід використання системи більш зручним та простим.

Google Smart House також пропонує інтеграцію з додатком Google Nest, який надає додаткові можливості контролю та налаштування пристроїв. Додаток дозволяє користувачам відстежувати споживання енергії, налаштувати розклад роботи пристроїв, контролювати температуру в

кімнатах та інші функції [3].

В розріз з іншими рішеннями, наприклад, Аїах систем розумного будинку, ми маємо один результат у використанні та обмеження можливості пристроїв які надає нам виробник. З одної сторони це характеризує незалежність системи та її індивідуальний підхід, але в можливостях вона не покриває всі потреби клієнту/компанії, які постануть на основі задач майбутніх проектів.

Висновок та перспективи. Проведено аналіз стану С2С в системах розумного будинку, та дослідили, що використання приладів для розумного будинку зростає, як і зростає потенціал використання цих приладів. Розбіжність приладів та виробників так само експонентно зростає. Тому, потреба у єдиному рішенні зростає, що надає зручність користувачеві у використанні системи керування. На теперішній момент, кількість можливих єдиних рішень є малою, але зі зростанням потреби та проблеми рішення буде поширюватись.

Література:

1. Cloud-to-cloud
<https://developers.home.google.com/cloud-to-cloud>
2. Get started with Cloud-to-cloud
<https://developers.home.google.com/cloud-to-cloud/get-started>
3. What is Google Home and how does it work?
<https://www.digitaltrends.com/home/google-home-defined/>

ВИКОРИСТАННЯ .NET MAUI

Луценко І.В., Гаманюк І.М.

Державний університет телекомунікацій, м. Київ

MAUI - це фреймворк, створений компанією Microsoft, який використовується для створення кросплатформних застосунків мовами програмування С# та ХАМЛ. Був вперше випущений у травні 2022 року на основі платформи .NET 6 і є еволюцією Xamarin.Forms, що дає можливість не тільки легко адаптуватися розробникам із досвідом, але і мігрувати старий код на нову платформу.

Наразі підтримується операційні системи Android, Windows, iOS та macOS.

MAUI націлений на реалізацію якомога більшої кількості спільної логіки та елементів користувацького інтерфейсу із єдиної кодової бази, але водночас надання можливості створювати код та ресурси для специфічних платформ.

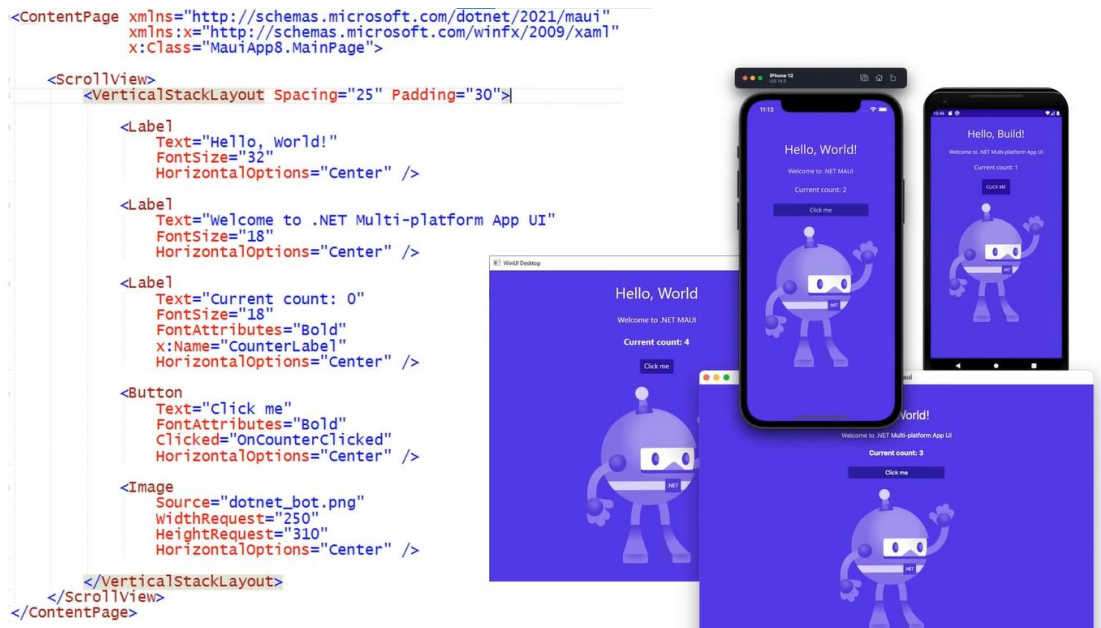


Рисунок 1 - Приклад XAML коду та його репрезентацій на різних платформах

Фреймворк базується на новій архітектурі (Рис. 2), яка абстрагує деталі конкретної платформи і дозволяє розробникам використовувати спільний API для доступу до інформації про девайс, датчиків, файлів, буфер обміну і тд.

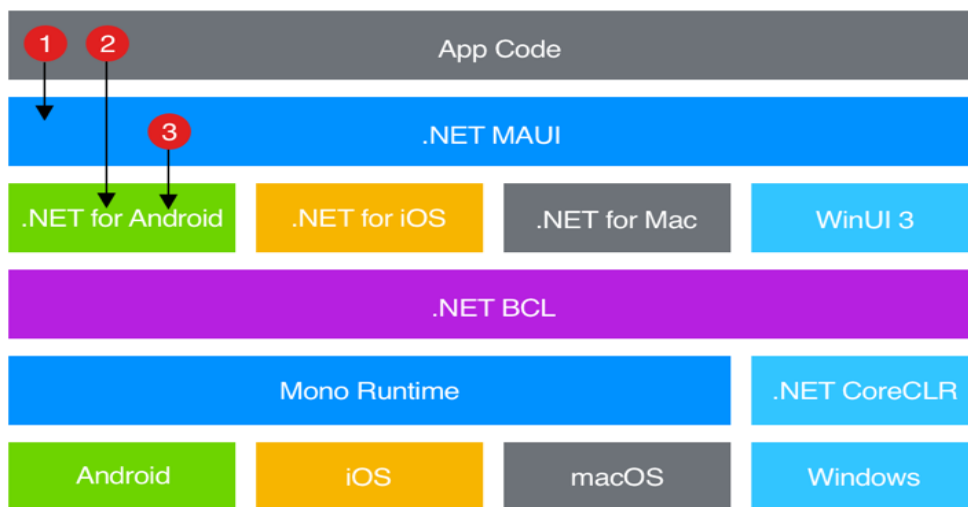


Рисунок 2 - Архітектура MAUI

Фреймворк використовує ряд бібліотек, для взаємодії з кожною конкретною платформою: .NET for Android, .NET for iOS, .NET for macOS та WinUI 3. Кожна з них має доступ до бібліотеки .NET Base Class Library (BCL), яка надає спільний абстрактний API.

На рисунку 2 наведено приклад більшості випадків, у яких код програми звертається до платформи .NET MAUI (1), а потім до бібліотеки для конкретної ОС (3). Також, якщо потрібно, код програми може звернутися відразу до бібліотеки платформи (2).

Висновок

Для розробки програмного забезпечення для моніторингу особистих спортивних тренувань найбільш раціональним вибором є використання технології для створення програмного забезпечення перш за все для мобільних платформ Android та iOS. Враховуючи зазначене, для цього було обрано .NET MAUI.

Література:

1. .NET Multi-platform App UI documentation. [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/maui/>
2. dotnet / maui. [Електронний ресурс]. – Режим доступу: <https://github.com/dotnet/maui>

ІОТ ТА ШТУЧНИЙ ІНТЕЛЕКТ

Патока В.В., Аверічев І.М.

Державний університет телекомунікацій, м. Київ

Постановка задачі

Інтернет речей (ІоТ) - технологія яка змінила сприйняття девайсів кардинально в нашому світі. Доступ до інтернету девайсів дала змогу їх використання на абсолютно новому рівні.

Штучний інтелект (ШІ) - принцип роботи нейронної мережі з додаванням можливості саморозвитку дала величезний крок для комп'ютерного навчання, і аналізу величезної купи даних.

З об'єднанням цих технологій побутові пристрої, пристрої безпеки і тому подібне створюють революцію в наш час.

Мета дослідження

В дослідженні показані можливості використання ІоТ поєднаними зі ШІ. У статті наведені розумні девайси які спрощують та покращують сучасне життя людей.

За останні кілька років ІоТ і штучний інтелект стали дуже популярними технологіями, які впливають на різні сфери нашого життя. Ці технології забезпечують підключення об'єктів до Інтернету, що дозволяє їм обмінюватися даними та керувати різними процесами. Штучний інтелект дозволяє аналізувати ці дані та приймати на їх основі рішення.

Однією з головних переваг ІоТ та ШІ є потенціал для автоматизації рутинних процесів і зниження ризиків. Більшість компаній вже використовують технології для автоматичного керування дорожнім рухом, контролю якості повітря та безпеки на робочому місці. Використання технологій ШІ зменшує кількість аварій, та покращує якість життя.

Питання конфіденційності та безпеки даних стають все більш актуальними. Збір і обробка великих обсягів даних може стати джерелом порушень конфіденційності та безпеки. Це може призвести до витоку даних та інших проблем. Крім того, існує проблема плинності робочих місць, оскільки автоматизація процесів може призвести до скорочення робочих місць. Це, у свою чергу, може негативно вплинути на соціальну стабільність і безробіття. Крім того, необхідно налагодити партнерство між галузями, які використовують технології Інтернету речей та ШІ, щоб сприяти координації та співпраці між ними. Сприяння взаємодії між медичними установами та розробниками може покращити системи охорони здоров'я. Це забезпечить підвищення якості медичної допомоги та ефективність використання ресурсів.

Надзвичайно важливо забезпечити глобальну координацію для розробки стандартів і принципів для технологій. Це має відбуватися через міжнародне співробітництво та угоди, які захищають права користувачів. Він має сприяти вільному розвитку та використанню технологій Інтернету речей та штучного інтелекту в усьому світі.

Ключовим аспектом технологій Інтернету речей та ШІ є соціальний вплив на людей і суспільство. Інтеграція таких технологій може змінити приватне та професійне життя людей. Це вимагає ретельного аналізу та планування, щоб забезпечити успішну адаптацію та мінімізувати негативний вплив.

Висновки та перспективи

Загалом IoT і ШІ мають значний потенціал, щоб змінити наше життя, покращити наші робочі процеси та забезпечити безпеку та зручність у нашому житті. В першу чергу необхідно забезпечити належну підготовку спеціалістів, які відповідатимуть за розробку та впровадження систем IoT та ШІ. Освіта майбутніх інженерів-програмістів має бути більше зосереджена на цих технологіях, щоб забезпечити відповідні знання та навички для успішного впровадження їх у майбутньому.

Література:

1. Як IoT та ШІ працюють разом.

URL: <https://www.techtarget.com/iotagenda/tip/AI-and-IoT-How-do-the-internet-of-things-and-AI-work-together>

2. Як ШІ змінює IoT.

URL: <https://www.infoworld.com/article/3663017/how-ai-is-changing-iot.html>

ЗАСОБИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТ

Патока В.В., Аверічев І.М.

Державний університет телекомунікацій, м. Київ

Постановка задачі

Програмна реалізація додатку для аналізу криптовалют має конкретні потреби для коректного функціонування. Вибір мови програмування для розробки залежить від специфіки криптовалютного ринку. Швидкість обробки великої кількості інформації є головним. Дані зазвичай надходять з високою швидкістю, що може створювати проблеми для їх обробки та зберігання.

Мета дослідження

В дослідженні показані засоби програмної реалізації додатку для аналізу криптовалют. У статті наведені засоби які пришвидшують та покращують процес реалізації програмного застосунку.

В якості мови розробки, засобів програмної реалізації додатку для аналізу криптовалют, вибрано Python. Він є однією з найпопулярніших мов програмування в світі, завдяки своїй простоті, елегантності та широкій спільноті розробників.

Одним з найбільш відомих фреймворків для створення веб-застосунків на Python являється Django. Він надає розробникам зручний інтерфейс для створення веб-додатків, забезпечуючи високу продуктивність, безпеку та масштабованість. Фреймворк містить вбудований ORM (Object-Relational Mapping), який дозволяє працювати з базами даних, такими як MySQL, PostgreSQL та SQLite, без прямої роботи з SQL.

Також допомагає в реалізації додатку для аналізу криптовалют бібліотека Django Channels. Додає до Django підтримку асинхронних запитів та WebSocket-з'єднань, що дозволяє розширити можливості Django-додатків та побудувати більш інтерактивні та реактивні веб-додатки.

Бібліотека Matplotlib, одна з найпопулярніших для візуалізації даних в Python. Вона містить багато інструментів для побудови різноманітних графіків, включаючи лінійні графіки, гистограми, діаграми розсіювання, графіки контурів та багато інших.

Для роботи з API допоможе базова бібліотека в Python requests, яка дозволяє здійснювати HTTP-запити та отримувати відповіді від сервера. Завдяки простому та зрозумілому API, бібліотека дозволяє легко взаємодіяти з web-ресурсами, зокрема з API веб-сервісів та іншими web-додатками. За допомогою requests можна відправляти HTTP-запити, такі як GET, POST, PUT, DELETE та інші, а також виконувати запити з різними параметрами, такими як параметри запиту, заголовки, файли тощо.

Візуальну частину додатку можна реалізувати використовуючи мову гіперпосилань HTML в поєднанні з мовою стилей CSS а також

шаблонізатором Jinja 2. Шаблонізатор Jinja 2 для Python, дозволяє вбудовувати в HTML-документи динамічний контент, а також підтримує вставку змінних, конструкції умов, циклів та інші елементи, які дозволяють генерувати різні типи контенту на стороні сервера.

Висновки та перспективи

Загалом засобів для програмної реалізації додатку для аналізу криптовалют, існує багато, але одним із найзручнішим вважається Python з його простотою, швидкістю створення проектів, та величезною кількістю бібліотек що допомагають вирішувати будь-які складнощі в процесі.

Література:

1. Веб-розробка Python – Підручник Django.

URL: <https://www.geeksforgeeks.org/python-web-development-django-tutorial/>

2. Чому Django є найкращим веб-фреймворком для вашого проекту.

URL: <https://steelkiwi.com/blog/why-django-best-web-framework-your-project/>

ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ ДЛЯ МОБІЛЬНОЇ РОЗРОБКИ

Полтавець Н.В., Шевченко С.М.

Державного університету телекомунікацій, м. Київ

Мобільна розробка на сьогоднішній день є важливою складовою людського життя, в нашому смартфоні можуть бути важливі мобільні додатки як банків, так і до особистих споживань, як додаток по розрахування калорій. Для розробки додатків використовують такі мови програмування як Java, Dart, Kotlin та Swift. При тому для кожного напрямку розробки існують бібліотеки, які покращують мобільний додаток та спрощують його розробку. Але мобільні стосунки мають залишатися актуальними відносно сучасних тенденцій. AI(штучний інтелект) сприяє еволюції мобільних додатків.

Для прикладу розуміння, як штучний інтелект слугує інструментом та збільшує можливості функціональності додатків. Нижче приведені декілька бібліотек машинного навчання, які можна використовувати як інструмент для створення функцій при розробці мобільного застосунку з фреймворком Flutter:

1. TensorFlow Lite

TensorFlow Lite – це набір інструментів, який забезпечує машинне навчання на пристрої, допомагаючи розробникам запускати свої моделі на мобільних, вбудованих та периферійних пристроях. TensorFlow Lite має підтримку для Flutter, що дозволяє інтегрувати моделі машинного навчання безпосередньо у Flutter додатки.

2. ML Kit

ML Kit – це мобільний SDK, який поєднує досвід Google у галузі машинного навчання з програмами для Android та iOS у потужному, але простому у використанні пакеті. ML Kit можна використовувати з Flutter за допомогою пакету Flutterfire.

3. OpenCV

OpenCV (Бібліотека комп'ютерного зору з відкритим кодом) — це бібліотека з відкритим кодом, яка містить кілька сотень алгоритмів комп'ютерного зору. Вона може бути використана для обробки зображень, визначення об'єктів на зображеннях, а також для виконання аналізу зображень. Інтегрувати OpenCV з Flutter можна за допомогою пакету opencv_2.

4. Hugging Face

Hugging Face є бібліотекою для роботи з нейронними мережами та штучним інтелектом. Вона дає змогу для обробки мови(класифікація тексту, переклад, генерація тексту тощо), класифікація зображень, автоматичне розпізнавання мовлення, та інше. Hugging Face можна використовувати з Flutter за допомогою пакету huggingface_inference.

5. PyTorch Mobile

PyTorch Mobile - це платформа глибокого навчання для навчання та запуску моделей машинного навчання (ML), що прискорює швидкість від дослідження до виробництва. Для інтегрування в Flutter можна за допомогою пакету torch_flutter.

Тож на сьогоднішній день створили велику кількість бібліотек з машинним навчанням, які можна використовувати в розробці мобільних застосунків. Оскільки розробка за допомогою фреймворку Flutter збільшує свою популярність на IT-арені мобільної розробки, багато бібліотек зі машинним навчанням можна інтегрувати при розробці з Flutter.

Література:

1. <https://www.tensorflow.org/lite/guide>
2. <https://firebase.google.com/docs/ml-kit>
3. <https://huggingface.co/docs/transformers/index>
4. <https://docs.opencv.org/4.x/d1/dfb/intro.html>

ВИКОРИСТАННЯ АЛГОРИТМУ SUPERVISED LEARNING ДЛЯ ПРОГНОЗУВАННЯ НА ВЕЛИКИХ НАБОРАХ ДАНИХ

Поночовний О.В., Садовенко В.С.

Державного університету телекомунікацій, м. Київ

Машинне навчання (МО, Machine Learning, ML) — великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. Одним з найбільш перспективних застосувань машинного навчання

є прогностична аналітика. Її можна використовувати для всього, від створення продукту до ціноутворення на нерухомість.

Найбільш успішні алгоритми машинного навчання – це ті, які автоматизують процеси прийняття рішень шляхом узагальнення відомих прикладів. У цих методах, відомих як навчання з учителем або контрольоване навчання (Supervised Learning), користувач надає алгоритму пари об'єкт – відповідь, а алгоритм знаходить спосіб отримання відповіді по об'єкту. Зокрема, алгоритм здатний видати відповідь для об'єкта, якого він ніколи не бачив раніше, без будь – якої допомоги людини.

Процес класифікації з вчителем складається з двох етапів: побудови моделі класифікації і класифікації нових екземплярів даних. Обидва етапи включають такі дії: підготовку набору навчальних даних; попередню обробку даних; визначення інформативних ознак; побудова аналітичної моделі; збір нових даних; попередню обробку нових даних; визначення інформативних ознак у нових даних; класифікацію нових даних. При навчанні з вчителем, кожен екземпляр даних є незалежним екземпляром прикладної області, яку необхідно вивчити. Звичайно, є багато задач, в яких необроблені дані не можуть бути виражені як індивідуальні, незалежні екземпляри і для формування екземплярів слід приймати до уваги довідкову інформацію про прикладну область

Магазин може передбачати кількість конкретних товарів, які треба закупити за результатами оброблення масиву попередніх продажів. Система «тренується» та вчиться враховувати сезонність, особливості конкретного бізнесу в конкретному магазині та безліч інших факторів. Можна легко взяти до уваги трендові складові. Зокрема, якщо у вас загалом зростають продажі, це також буде відображено у прогнозі.

Література:

1. Сайт SENIOR.ua [Електронний ресурс] – Режим доступу: <https://senior.ua/articles/scho-take-mashinne-navchannya-yak--de-pracyu>

НАПРЯМ 4. РОЗРОБКА ІГОР ТА ІГРОФІКАЦІЯ ОСВІТНІХ ТА БІЗНЕС ПРОЦЕСІВ

ОСОБЛИВОСТІ РОЗРОБКИ ІГОР ЗА ДОПОМОГОЮ UNITY

Білоус В.С., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Вибір двигуна – це саме перше завдання, з яким спотикається розробник ігор, бо вибір інструменту, де ти будеш робити свої ігри одне із самих головних завдань, тому потрібно віднестись до цього питання

серйозно. В даний момент вибір багатьох початківців стоїть між Unity та Unreal Engine 5, але, що там , що там є свої плюси та мінуси , та свої особливості.

Отже, двигун Unity - це один з найпопулярніших ігрових двигунів, які використовуються для створення ігор на різних платформах, включаючи ПК, мобільні пристрої та консолі.

На Unity було розроблено дуже багато гігантів таких як, Cuphead , Hearstone , Rust , Subnaftica , а з останніх дуже популярних це Among Us , гра, яка розірвала увесь ринок серед усіх поколінь.

В Unity є дуже багато особливостей при розробці , серед яких є:

- Кросплатформенність: Unity дозволяє розробникам створювати ігри на різних платформах, використовуючи єдиний код. Це спрощує процес розробки та дозволяє розробникам випускати ігри на кількох платформах.
- Asset Store: Unity Asset Store - це онлайн-магазин, де розробники можуть купувати готові ресурси для створення ігор, такі як моделі персонажів, текстури, звукові ефекти та багато іншого.
- Скриптинг: Unity використовує мову програмування C# розробки ігор. Ця мова програмування має безліч можливостей і є відносно простою для вивчення.
- Оптимізація: Unity має безліч інструментів, які дозволяють оптимізувати гру для кращої продуктивності та більш плавного геймплею.
- Велика спільнота: Unity має велику та активну спільноту, яка може допомогти у вирішенні різних проблем та дати поради щодо розробки ігор.
- Рендеринг: Unity має власний движок рендерингу, який дозволяє створювати ігри з високим рівнем деталізації та візуальною якістю.

В цілому, Unity - це потужний інструмент для створення ігор, що дозволяє розробникам швидко створювати ігри на різних платформах з високим рівнем деталізації та візуальною якістю.

Література:

1. Використання рушія Unity у мобільному геймдеві: можливості, переваги та недоліки [Електронний ресурс]. - Режим доступу до ресурсу:

<https://vokigames.com/ua/vikoristannya-rushiya-unity-u-mobilnomu-gejmdevi-mozhливosti-perevagi-ta-nedoliki/>

2. The pros and cons of the Unity game engine [Електронний ресурс]. - Режим доступу до ресурсу:

<https://www.itechart.com/blog/unity-pros-and-cons/>

ВИКОРИСТАННЯ LOW-POLY ГРАФІКИ

Павліченко В.Ю., Гребенюк В.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі:

З часів початку ігрової індустрії, а саме 70 років, графіка в іграх будь-яких жанрів мінялась по сьогоднішній день. Зараз ми можемо побачити реалістичні моделі персонажів, високоякісні текстури та круті багато часткові партикли, і з кожним новим роком технології йдуть вперед. Однак багато компаній використовують Low-Poly для своїх творінь, тому що цей вид графіки без велетенських зусиль дозволяє гравцю відчувати себе комфортно. Цей вид відрізняється від інших тим, що воно складається з малої кількості полігонів, завдяки чому, гра стає простішою, але залишається яскравою та привабливою.

Мета дослідження:

Вивчення особливостей використання Low-Poly графіки для гри в жанрі Survival, наше дослідження має виявити чи можливо досягнути привабливої, яскравої та комфортної графіки для користувача. Також оцінити вплив заданого виду графіки на гравця.

Результати дослідження:

Аналіз існуючих ігор з використанням графіки в стилі low-poly показав, що такий стиль може дозволити створити незвичайну та стильну графіку, що привертає увагу користувачів. Крім того, зменшення кількості полігонів дозволяє більш ефективно використовувати ресурси обчислювальної техніки та зменшує вимоги до технічних характеристик комп'ютерів користувачів. Більшість користувачів відзначили стильну та незвичайну графіку як перевагу гри. Крім того, користувачі відчували себе зануреними у світ гри, що свідчить про позитивний вплив графіки на іммерсію в гру.

Висновок:

Підсумки проведеного дослідження дозволяють стверджувати, що використання стилю графіки low-poly в ігровій індустрії, зокрема в жанрі survival, може бути ефективним. Зменшення кількості полігонів дозволяє оптимізувати використання обчислювальних ресурсів та зменшити вимоги до технічних характеристик комп'ютерів користувачів. Також графіка в стилі low-poly може створити незвичайний та стильний образ гри, що привертає увагу гравців та створює позитивне враження від гри.

Література:

1. Бізюк А. В. LOW POLY МОДЕЛЛІНГ ДЛЯ ІНДУСТРІЇ ІНТЕРАКТИВНИХ РОЗВАГ [Електронний ресурс] / А. В. Бізюк, В. О. Нечаєв, М. М. Глушко – Режим доступу до ресурсу: <https://openarchive.nure.ua/server/api/core/bitstreams/ac9542f5-d471-4e2e-ab10-0fa17b201561/content>.
2. Stachowiak M. What is Low Poly? Showcase of Great Low Poly Art [Електронний ресурс] / Martyna Stachowiak – Режим доступу до ресурсу: <https://blog.displate.com/low-poly-art/>.

ЗАСТОСУВАННЯ ШАБЛОНУ ПРОЕКТУВАННЯ VISITOR НА ДВИГУНІ UNITY ДЛЯ ОБРОБКИ ВЗАЄМОДІЇ ОБ'ЄКТІВ ГРИ

Бойко М.С., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

У цій тезі буде розглянуто використання шаблону проектування "Visitor" на двигуні Unity для обробки взаємодії між об'єктами гри. Шаблон дозволяє розділити алгоритм взаємодії між об'єктами та їх структурою, що дозволяє легко додавати нові типи об'єктів та алгоритми без зміни основного коду.

У данній роботі було використано шаблон для обробки взаємодії між об'єктами куль та ворогів у грі жанру Tower Defense. Кожен тип куль має унікальні характеристики та ефекти, тому кожен ворог повинен мати власну унікальну реакцію на кожен тип куль. У грі було використано інтерфейс IEnemyVisitor для взаємодії між об'єктами куль та ворогів. Коли куля торкається ворога, вона передає себе інтерфейсу IEnemyVisitor як параметр, що дозволяє ворогові унікально реагувати на кожен кулю.

Таким чином, застосування шаблону проектування "Visitor" на двигуні Unity, дозволяє ефективно управляти взаємодією між об'єктами гри, легко додавати нові типи об'єктів та алгоритмів, без зміни основного коду.

Для реалізації шаблону "Visitor" на двигуні Unity використано наступний підхід: створення інтерфейсу IEnemyVisitor, який містить методи TakeDamage, що приймають кожен тип об'єктів з якими повинні бути взаємодії. Далі було створено різні класи відвідувачів, які у випадку дотику до ворога звертаються до його інтерфейсу IEnemyVisitor і викликають метод TakeDamage, передаючи себе як параметр.

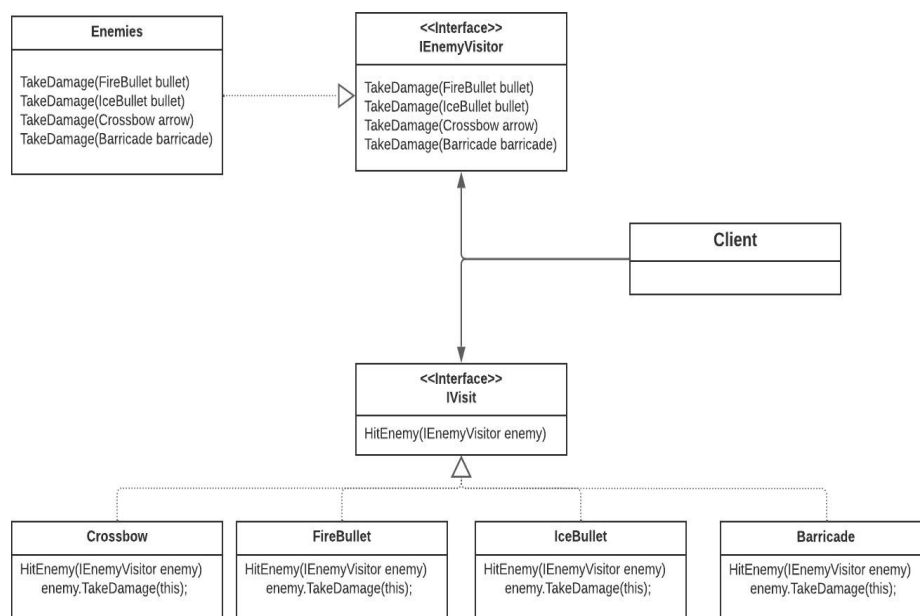


Рисунок 1 - Діаграма класів

У данному випадку, було створено інтерфейс IEnemyVisitor, який містить методи для обробки кожного типу куль та ворогів. Далі було розроблено інтерфейс IVisit, який містить метод HitEnemy. В цьому методі використовується метод TakeDamage інтерфейсу IEnemyVisitor якому передається поточний клас. Таким чином, коли куля торкається ворога, викликається метод HitEnemy, який передає кулю відповідному класу-відвідувачу, який реалізує логіку обробки взаємодії між кулею та ворогом.

Отже, використання шаблону проектування "Visitor" на двигуні Unity дозволяє з легкістю реалізувати логіку взаємодії між об'єктами, що знаходяться на сцені гри. Це дозволяє зосередитися на розробці ігрових механік, не звертаючи увагу на складну логіку взаємодії між об'єктами. Застосування шаблону проектування "Visitor" допомагає зменшити зв'язаність між класами та забезпечити легку розширюваність системи. Це дозволяє змінювати логіку гри та додавати нові елементи без великих змін в коді. Також, використання шаблону проектування "Visitor" допомагає зберегти код чистим та організованим, що робить його легким для розуміння та підтримки.

Загалом, використання шаблону проектування "Visitor" на двигуні Unity є ефективним рішенням для реалізації складної логіки взаємодії між об'єктами гри, що дозволяє зосередитися на розробці ігрових механік та зберегти код організованим та легким для підтримки.

Література:

1. Head First Design Patterns: A Brain-Friendly Guide (2004) by Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates. Ресурс доступу: <https://www.oreilly.com/library/view/head-first-design/0596007124/>
2. Refactoring Guru - Visitor Pattern [Електронний ресурс] - Ресурс доступу: <https://refactoring.guru/design-patterns/visitor>
3. Design Patterns – Visitor Pattern [Електронний ресурс] - Ресурс доступу: https://www.tutorialspoint.com/design_pattern/visitor_pattern.htm

АНАЛІЗ ВПЛИВУ МЕТАВСЕСВІТІВ ТА РОЛЬОВИХ РЕЖИМІВ НА РОЗВИТОК ІНФОКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Торжевський Я. Р., Щербина І. С.

Державного університету телекомунікацій, м. Київ

На сучасному етапі ми спостерігаємо бурхливий розвиток RP (roleplay) режимів та метавсесвітів у сфері відеоігор. Це зумовлено підвищенням у останній час інтересом користувачів до більш

іммерсивних та віртуальних середовищ, які дозволяють гравцям поринути у різні ролі та життєві ситуації. За допомогою передових технологій та широкого доступу до інтернету, RP режими та метавсесвіти стали невід'ємною частиною геймерської культури.

Переваги RP режимів та метавсесвітів полягають у тому, що вони дозволяють гравцям відчувати атмосферу віртуального світу та отримати більш насичений досвід взаємодії з іншими учасниками. Особливо важливим аспектом є розвиток комунікації та підвищення соціальних навичок. Участь у RP режимах забезпечує гравцям можливість вільно спілкуватися, обмінюватися думками та досвідом, а також вирішувати конфліктні ситуації. Гравці також можуть відпрацьовувати навички лідерства, переговорів та командної роботи в складних ситуаціях.

Такі ігри вимагають від гравців активно використовувати різні канали зв'язку, такі як текстовий, голосовий та відео чат. Це стимулює розвиток нових інфокомунікаційних рішень та покращує якість віртуального спілкування. відповідно до зростання потреб гравців, розробники ігор та інфокомунікаційних технологій постійно вдосконалюють інструменти та сервіси для спілкування, забезпечуючи кращу якість аудіо, відео та текстових повідомлень.

RP режими та метавсесвіти можуть бути інтегровані з різними платформами, такими як стрімінгові сервіси, соціальні мережі та спеціалізовані додатки для комунікації. Це розширює можливості для інновацій та стимулює розвиток нових інфокомунікаційних продуктів та сервісів. Також, це сприяє підвищенню обізнаності та залучення нових користувачів до RP режимів, розвиваючи глобальну комунікаційну мережу.

RP режими та метавсесвіти також можуть використовуватися в освітніх цілях, допомагаючи гравцям набувати нових знань та навичок. Вони можуть застосовуватися для вивчення іноземних мов, розвитку критичного мислення та підвищення культурної свідомості, сприяючи розвитку інфокомунікаційної грамотності. Ряд освітніх ігор і навчальних платформ вже інтегрували RP елементи, що забезпечують більш захоплююче та ефективно навчання.

Завдяки RP режимам та метавсесвітам, інфокомунікаційні технології стають доступнішими для широкого спектру користувачів, включаючи осіб з обмеженими можливостями. Це сприяє розвитку інклюзивних рішень та технологій, які підтримують різноманітність та доступність. Геймінг-спільнота все більше розуміє необхідність створення доступного контенту та інтерфейсів, що враховують потреби користувачів з різними можливостями. Це спонукає розробників ігор та інфокомунікаційних продуктів до покращення та адаптації своїх рішень для різних груп користувачів.

Особливо великий внесок RP режими та метавсесвіти роблять в розвиток креативності та віртуального мистецтва. Гравці мають можливість створювати унікальні персонажів, дизайн інтер'єрів, а також

складати музику та знімати кінопродукцію. Це спонукає розвиток нових інфокомунікаційних інструментів та сервісів, спрямованих на підтримку та розповсюдження цих творчих робіт.

Отже, бурхливий розвиток RP режимів та метавсесвітів у сфері відеоігор має значний вплив на розвиток інфокомунікаційних технологій. Вони сприяють розширенню можливостей комунікації, інтеграції з іншими технологіями, підтримці освітнього процесу, інклюзивності та доступності, стимулюванню розвитку ігрової індустрії та розвитку креативності та віртуального мистецтва. Це свідчить про важливість подальшого дослідження та розвитку RP режимів та метавсесвітів, які мають значно більший потенціал, ніж ігри минулих поколінь, для зміни сучасних інфокомунікаційних технологій та робити їх більш ефективними, доступними та корисними для широкого спектру користувачів.

Література:

1. Communication in Multi-player Role Playing Games - The Effect of Medium, Anders Drachen, Michael Hitchens [Електронний ресурс]
https://www.researchgate.net/publication/221274381_Communication_in_Multi-player_Role_Playing_Games_-_The_Effect_of_Medium_Pre-print
2. Reinventing Ourselves: Contemporary Concepts of Identity in Virtual Worlds, Mark Childs, Anna Peachey [Електронний ресурс]
https://www.researchgate.net/publication/321594202_Reinventing_Ourselves_Contemporary_Concepts_of_Identity_in_Virtual_Worlds
3. Virtual Worlds: Concepts, Applications and Future Directions Karen B. Monkeovich, Nolan H. Bard

РОЛЬОВІ МОДИФІКАЦІЇ У ВІДЕОІГРАХ: РОЗГЛЯД ВПЛИВУ ТА ЗНАЧЕННЯ В ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЯХ

Щукін І.Ю., Щербина І.С.

Державного університету телекомунікацій, м. Київ

За останні кілька десятиліть відеоігри значно еволюціонували, ставши частиною у секторі інформаційних та комунікаційних технологій (ІКТ). Модифікації(моди) рольових ігор, або моди, відіграли ключову роль у цій еволюції, покращуючи ігровий досвід, надаючи гравцям додатковий контент і можливості ігрового процесу. Ці моди дозволяють гравцям взаємодіяти з іграми у способи, які не були спочатку заплановані розробниками, тим самим стимулюючи творчість та інновації в ігровій спільноті. У цій тезі ми заглиблюємося у взаємозв'язок між рольовими модифікаціями та ІКТ, досліджуючи, як ці моди сприяють зростанню та розвитку сектору.

Роль рольових модів в ІКТ можна проаналізувати за чотирма основними напрямками: розробка ігор, досвід гравців, ширша індустрія та безпосередній

вплив на сектор ІКТ.

Рольові моди стали важливим аспектом розробки ігор, оскільки вони дозволяють розробникам продовжити життєвий цикл своїх ігор. Використовуючи творчість та навички спільноти розробників модів, вони можуть постійно впроваджувати новий контент, ігрові механіки та функції, зберігаючи свої ігри актуальними та цікавими. Крім того, спільноти любителів модів можуть слугувати випробувальним майданчиком для розробників, дозволяючи їм збирати цінний зворотній зв'язок щодо потенційних оновлень та вдосконалень гри. Цей ітеративний процес призводить до покращення ігрового дизайну та кращого узгодження з уподобаннями гравців, що в кінцевому підсумку приносить користь усій екосистемі ІКТ.

Моди для рольових ігор мають значний вплив на ігровий досвід, оскільки вони пропонують гравцям можливість налаштувати та персоналізувати своїх ігрових персонажів, оточення та оповідання. Ці моди дають більше відчуття занурення та залучення. Більше того, рольові моди можуть сприяти соціальній взаємодії між гравцями, оскільки вони заохочують до співпраці, обміну модами та формування спільнот, заснованих на спільному ігровому досвіді. Покращений ігровий досвід підвищує рівень залученості та задоволеності користувачів, сприяючи зростанню ІКТ-сектору.

Поява рольових модів у відеоіграх призвела до створення процвітаючої екосистеми модів. Ця екосистема створила нові можливості як для аматорів, так і для професійних розробників, а також для компаній, що виробляють програмне та апаратне забезпечення. Моддинг-спільнота стимулює інновації в індустрії, оскільки розробники заохочуються до створення ігор з підтримкою модів, а виробники обладнання повинні розробляти технологію, здатну запускати дедалі складніші моди. Крім того, популярність рольових модів сприяла зростанню ігрової індустрії в цілому, залучаючи нових гравців і генеруючи додаткові потоки доходів.

Рольові моди мають безпосередній вплив на сектор ІКТ, стимулюючи попит на більш досконалі апаратні та програмні рішення. Оскільки ці моди стають більш складними та ресурсно-витратними, вони розширюють межі існуючих технологій, стимулюючи інновації та розвиток в ІКТ-секторі. Попит на вдосконалене обладнання та програмне забезпечення призводить до збільшення інвестицій у дослідження та розробки, сприяючи технологічному прогресу, який приносить користь не лише ігровій індустрії, а й ширшому сектору ІКТ.

Модифікації рольових ігор стали невід'ємною складовою ІКТ-сектору, впливаючи на розробку ігор, досвід гравців та індустрію в цілому. Заохочуючи творчість, інновації та створення спільнот, рольові моди значно покращили ігровий ландшафт і, ймовірно, продовжуватимуть формувати майбутнє індустрії. Безпосередній вплив цих модів на ІКТ-сектор проявляється у попиті на більш досконалі апаратні та програмні рішення, а також у появі нових ринків, що обслуговують спільноту гравців, які цікавляться модами.

Оскільки сектор ІКТ продовжує зростати і розвиватися, дуже важливо визнати вплив і значення рольових модів у формуванні майбутнього ігор і цифрового світу в цілому. Розуміючи складну взаємодію між рольовими іграми та ІКТ, зацікавлені сторони можуть використовувати ці моди для стимулювання подальших інновацій, покращення користувацького досвіду та створення нових можливостей в ігровому та технологічному секторах. Зрештою, рольові моди можуть трансформувати не лише ігрову індустрію, а й ширшу екосистему ІКТ, сприяючи цифровій революції, що триває, та формуванню дедалі більш взаємопов'язаного і захоплюючого світу.

Література:

1. Video games are transforming how we communicate with each other - and they could fix a range of other global issues too, Susanna Pollack [Електронний ресурс] <https://www.weforum.org/agenda/2019/12/video-games-culture-impact-on-society/>
2. Video game hardware sales increased by 112% in June, Sam Desatoff [Електронний ресурс] <https://www.gamedaily.biz/report-video-game-hardware-sales-increased-by-112-in-june-npd/>

РОЗРОБКА МОБІЛЬНОЇ ВІДЕОГРИ НА ЮНІТІ

Розгон Д.А., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Розробка мобільної гри є комплексною задачею, що включає в себе різні етапи розробки, проектування і підтримки проєкту. І на завжди фінальний продукт буде задовольняти користувача. Такий продукт має відповідати певним вимогам, які визначають розробники, такі як бистродія, відмовостійкість, вимоги до графіки і можливості розширення в майбутньому.

Важливою частиною розробки гри є вибір ігрового рушія для реалізації ідей. Він використовується для спрощення процесу розробки і його пришвидшення. Це досягається за рахунок того, що ігровий рушій вже має готові рішення наприклад для фізики, освітлення, кросплатформеності і інших процесів, що займали б багато часу при їх ручній розробці. Проаналізувавши мої потреби, а саме мобільна гра з простою фізикою і мовою програмування C# я обрав ігровий рушій Unity.

Unity - це середовище для розробки відеоігор, що розробляються і орієнтуються відразу на декілька платформ. Створила цей рушій американська компанія Unity Technologies. Unity є надзвичайно простим і безкоштовним.

Unity підтримує систему компонентів, що значно спрощує процес розробки.

Компоненти є функціональними частинами кожного GameObject. Компоненти містять властивості, які можна редагувати, щоб визначити

поведінку GameObject. Щоб отримати додаткові відомості про зв'язок між компонентами та GameObjects, перегляньте GameObjects

Щоб переглянути список компонентів, прикріплених до GameObject в інспекторі виберіть GameObject у вікні Hierarchy або Scene

Ви можете приєднати багато компонентів до GameObject, але кожен GameObject повинен мати один і тільки один компонент Transform. Це тому, що компонент Transform визначає розташування, обертання та масштаб GameObject.

Таким чином система компонентів спрощує розробку ігор, а також покращує зрозумілість проекту для інших розробників.

Література:

1. Unity Documentation. [Електронний ресурс]. – Режим доступу:

<https://docs.unity3d.com/Manual/Components.html>

АНІМАЦІЯ В UNITY

Левчук М.П., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Функції анімації Unity включають анімацію з можливістю перенацілювання, повний контроль ваги анімації під час виконання, виклик подій із відтворення анімації, складні ієрархії та переходи кінцевого автомата, форми змішування для анімації обличчя та багато іншого.

Unity має багату та складну систему анімації. Вона забезпечує:

- Простий робочий процес і налаштування анімації для всіх елементів Unity, включаючи об'єкти, персонажів і властивості.
- Підтримка імпортованих анімаційних кліпів і анімації, створеної в Unity
- Ретаргетинг гуманоїдної анімації - можливість застосовувати анімацію з однієї моделі персонажа на іншу.
- Спрощений робочий процес для вирівнювання анімаційних кліпів.
- Зручний перегляд анімаційних кліпів, переходів та взаємодій між ними. Це дозволяє аніматорам працювати більш незалежно від програмістів, прототипувати та переглядати анімації до підключення ігрового коду.
- Управління складними взаємодіями між анімаціями за допомогою інструмента візуального програмування.
- Анімація різних частин тіла з різною логікою.
- Особливості шарування та маскування

Система анімації Unity базується на концепції анімаційних кліпів, які містять інформацію про те, як певні об'єкти мають змінювати своє положення, обертання чи інші властивості з часом. Кожен кліп можна розглядати як один лінійний запис. Анімаційні кліпи із зовнішніх джерел створюються художниками чи аніматорами за допомогою інструментів сторонніх розробників, таких як Autodesk® 3ds Max® або Autodesk® Maya®, або надходять із студій захоплення руху чи інших джерел.

Потім анімаційні кліпи організовуються у структуровану систему, схожу на блок-схему, яка називається контролером анімації. Контролер Animator діє як «State Machine», який відстежує, який кліп зараз має відтворюватися, а також коли анімації мають змінюватися або змішуватися.

Дуже простий контролер анімації може містити лише один або два кліпи, наприклад, щоб керувати обертанням і підстрибуванням посилення або для анімації відкриття та закриття дверей у потрібний час. Досконаліший контролер анімації може містити десятки гуманоїдних анімацій для всіх дій головного героя та може поєднуватися між кількома кліпами одночасно, щоб забезпечити плавний рух, коли гравець рухається сценою.

Система анімації Unity також має численні спеціальні функції для роботи з гуманоїдними персонажами, які дають вам можливість перенацілювати гуманоїдну анімацію з будь-якого джерела (наприклад: захоплення руху; Asset Store або будь-яка інша бібліотека анімації третьої сторони) на вашу власну модель персонажа, а також коригування визначення м'язів. Ці особливі функції доступні системою Avatar Unity, де персонажі-гуманоїди відображаються у загальному внутрішньому форматі. Кожна з цих частин – анімаційні кліпи, контролер аніматора та аватар – об'єднані в GameObject за допомогою компонента аніматора. Цей компонент містить посилання на контролер аніматора та (за потреби) аватар для цієї моделі. Контролер Animator, у свою чергу, містить посилання на анімаційні кліпи, які він використовує.

Література:

1. Animation [Електронний ресурс]:
<https://docs.unity3d.com/ru/2019.4/Manual/AnimationSection.html>
2. Animation System Overview [Електронний ресурс]:
<https://docs.unity3d.com/ru/2019.4/Manual/AnimationOverview.html>

СУЧАСНІ РІШЕННЯ ДЛЯ РОЗРОБКИ ІГОР

Малярчук М.М., Іщеряков С.М.

Державного університету телекомунікацій, м. Київ

Постановка завдання. Аналіз існуючих технологій для розробки ігор під будь-які платформи.

Мета дослідження. Дослідити та обґрунтувати вибір тих чи інших сучасних засобів для розробки ігор.

Результати дослідження. Відеоігри, як програмний продукт, вимагають від розробника багато навичок та часу, так як для створення повноцінної гри її код повинен мати багато підсистем як для конкретних низькорівневих задач так і для ігрових речей.

Розробник може вибрати шлях повного написання гри з нуля, в такому випадку, крім суто ігрових систем, механік та розробки ігрового процесу, йому також доведеться розробити всі базові низькорівневі системи. Перевагою такого підходу є те, що в ньому розробник має максимум контролю над необхідними йому процесами, що надає можливість ідеально оптимізувати роботу цих процесів під задуми розробника, та легко реалізувати нестандартні функції та ефекти. Недоліком такого підходу, в свою чергу, є те, що для кожної гри реалізація базових систем, як наприклад рендеринг зображення, буде займати багато часу.

Для спрощення написання відеоігор розробляють ігрові рушії, які одразу реалізують всі опорні системи, як наприклад системи рендерингу, звуку, вводу/виводу на різні пристрої, обробки фізики, анімацій, вбудовані інструменти для створення штучного інтелекту, гри по мережі, керування пам'яттю, а також підтримують локалізацію на різні мови. Вони дозволяють перейти одразу до розробки суто ігрових систем, та при необхідності, надають можливості для внесення змін у базові системи.

Кожен розробник може сам написати власний рушій, реалізуючи всі потрібні йому системи та доповнювати його з плином часу. Це забере багато часу, але це необхідно бути зробити лише один раз. Проте також чудовим варіантом є використання вже існуючого рушія. Існує багато рушіїв, які кожен може використовувати, а деякі навіть модифікувати за бажанням.

Серед них варто виділити:

- Unreal Engine. Один із найпопулярніших рушіїв який є мульти-платформенною системою розробки ігор будь-яких розмірів. Проте він має доволі великі вимоги до навичок розробників.
- Unity. Цей мульти-платформенний ігровий рушій поєднує у собі доволі низький поріг входу для розробників та багатий функціонал здатний задовольнити майже будь-які вимоги, але він не дуже підходить для розробки великих проєктів.
- Godot. Цей рушій є повністю безкоштовним, та не накладає ніяких обмежень на використання. Він є доволі простим, гнучким та

ідеально підійде для новачків. Проте для багатьох проектів він може виявитись надто простим, але нічого не заважає розширити його самому.

В загальному випадку, для розробки гри потрібно вирішити який рушій використовувати та чи використовувати його взагалі. Для одноразового проекту, який служить для навчання/дослідження або для тестування певних алгоритмів/ідей можна обійтись зовсім без рушія. Проте, якщо проект замислювався як серйозний або комерційний, варто звернути увагу на ігрові рушії. Для нестандартних ігор, які наприклад використовують неевклідову геометрію або мають особливості рендерингу слід розглянути варіант з власним рушієм, або вибрати рушій з відкритим вихідним кодом для модифікації, у інших випадках готові рушії на кшталт Unreal або Unity є чудовим вибором для більшості розробників.

Література:

1. Game Engines: Driving the Future of Gaming [Електронний ресурс] <https://ancient8.gg/research/en/articles/game-engines-driving-the-future-of-gaming>
2. Колекція рушіїв з відкритим кодом на GitHub [Електронний ресурс] <https://github.com/collections/game-engines>
3. How to Choose the Best Video Game Engine [Електронний ресурс] <https://www.gamedesigning.org/career/video-game-engines/>

РОЗРОБКА ГРИ МОНОПОЛІЯ

Ярошевський О.В., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Монополія - це популярна настільна гра, яка дозволяє гравцям зануритися в світ бізнесу та взаємодіяти між собою. З метою покращення досвіду гри та забезпечення більшої реалістичності, було поставлено задачу розробити комп'ютерну версію гри Монополія.

Головною метою дослідження є розробка комп'ютерної версії гри Монополія, яка забезпечує реалістичну імітацію бізнес-середовища, взаємодію між гравцями та приємний користувацький інтерфейс.

Результатом дослідження є розробка комп'ютерної версії гри Монополія з використанням сучасних технологій та алгоритмів, яка забезпечує реалістичну імітацію бізнес-середовища та взаємодію між гравцями. Крім того, гра має інтуїтивний користувацький інтерфейс та покращені функції, які забезпечують більшу зручність користування.

Розробка комп'ютерної версії гри Монополія дозволить гравцям насолоджуватися цією популярною грою, забезпечує реалістичну імітацію бізнес-середовища та взаємодію між гравцями. За результатами дослідження можна зробити висновок, що розроблена гра має перспективи для подальшого

вдосконалення та розвитку, зокрема за рахунок розширення функціональності гри. Також, розробка подібних комп'ютерних версій настільних ігор може стати популярною тенденцією в галузі індустрії розваг. У майбутньому, можна очікувати розвиток технологій та наукових підходів, які дозволять створювати ще більш реалістичні та захоплюючі комп'ютерні версії настільних ігор.

Література:

1. Monopoly (game) [Електронний ресурс] // Wikipedia. Режим доступу: [https://en.wikipedia.org/wiki/Monopoly_\(game\)](https://en.wikipedia.org/wiki/Monopoly_(game)) (Дата звернення: 04.04.2023)

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІГРОВОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ UNITY 2D

Карасовський В.В., Гребенюк В.В.

Державного університету телекомунікацій, м. Київ

Постановка задачі

Задачами проекту є:

- Розробка застосунку в Unity;
- Вивчення та використання мови програмування C#;
- Вдосконалення навичок роботи з Adobe Photoshop та інтеграції в застосунок;
- Вдосконалення навичок роботи з Unity ;
- Вдосконалення методики створення ігрового застосунку.

Метою дослідження є розробка ігрового застосунку в середовищі Unity 2D, з використанням мови програмування C#.

Результати дослідження

Реалізований ігровий застосунок в середовищі Unity 2D.

Висновки та перспективи

Під час розробки та проектування ігрового застосунку, ми розвинемо навички зі створення 2D ігрового застосунку в середовищі Unity з використанням мови програмування C#.



Рисунок 1 – Ігровий застосунок

Література:

1. Unity Manual - Introduction to 2D. [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/2D-introduction.html>

ДОСЛІДЖЕННЯ ВІДЕОІГОР В ЖАНРІ “ВИЖИВАННЯ”

Вдовін Д.Е., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Виживання — це жанр, який зосереджується на боротьбі героя за виживання в екстремальних ситуаціях. Часто в цьому жанрі герой опиняється в космосі або далекому світі, де він повинен боротися з силами зла, щоб вижити. Виживання жанр дуже популярний в відеоіграх.

Однією з характеристик життя є загроза життю головного героя. Герой стикається зі слабкими енергіями, такими як дикі тварини чи монстри, які намагаються його з'їсти, голод через спеку та холод, зневоднення або зневоднення.

Ще один важливий аспект жанру полягає в тому, що герой повинен збирати і використовувати різні предмети, щоб вижити. Це їжа, вода, медикаменти, зброя, обладнання, пальне тощо. це може бути.

Герой також може використовувати свої навички та знання, щоб вижити та дійти до кінця. Інший аспект життя – це страх, викликаний відсутністю найнеобхіднішого, постійними погрозами та бажанням жити героїчним життям. Ця напруга може бути виражена глядачеві чи грацю, дозволяючи зрозуміти ефект спостереження героя.

У цій ситуації виникають моральні проблеми, коли герой вирішує, що важливіше – його життя чи моральні цінності.

Жанр виживання можна розділити на декілька піжжанрових напрямків, таких як постапокаліптика, виживання на природі, наукова фантастика, жахи та багато інших. У кожному з цих напрямків присутні свої особливості, що роблять гру унікальною та цікавою для гравців.

Unity — це середовище розробки ігор, яке дозволяє створювати ігри для багатьох платформ, таких як ПК, консолі, мобільні пристрої та віртуальна реальність. Unity має потужні інструменти для графіки, фізики, аудіо та інших завдань, які дозволяють розробникам створювати насичені та складні ігри. Unity підтримує такі мови програмування, як C#, що дозволяє розробникам вибирати різні шляхи розробки гри, різними шляхами.

Література:

1. Plarium.com [Електронний ресурс] – Ресурс доступу: <https://plarium.com/en/blog/survival-games/>
2. Unity Multiplier Networking [Електронний ресурс] – Ресурс доступу: <https://docs-multiplayer.unity3d.com>

ТВІНЕРИ І DOTWEEN – ЦЕ ЩО

Осьмак В.Р., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Ні для кого не секрет що зараз ігри займають досить велику частину ринку. Тому на сьогоднішній день двигун Unity є досить затребованим і стрімко набираючим обертів і саме тому для нього з'являються нові технології які допомагають в розробці. У минулому, розробники часто створювали анімацію вручну, що вимагало багато часу та зусиль. Зараз же, завдяки появі твінерів, створення складної анімації можливе у декілька кліків миші. Твінер допомагає їм досягти цієї мети, надаючи потужні функції для створення складних анімацій з мінімальними зусиллями. Завдяки цьому, розробники можуть сконцентруватись на творчому процесі та створювати кращі застосунки.

Твінер - це спосіб анімації, який використовується для плавного переходу між двома значеннями в часі. У розробці ігор, твінер дозволяє робити анімацію об'єктів, персонажів та ефектів, що забезпечує покращення геймплею та досвіду гравців.

Твінінг дає можливість програмістам та дизайнерам ігор створювати анімацію без необхідності вручну задавати кожен кадр. Замість цього, твінер автоматично розраховує та створює плавний рух об'єктів між початковим та кінцевим станами. Наприклад, можна використовувати твінер для створення плавного руху персонажа, зміни його розміру або форми, анімації зіткнення двох об'єктів, зміни кольору фону та багато іншого.

Один з популярних твінерів це DOTween. Він дозволяє робити плавну

анімацію будь-яких властивостей гри, включаючи рух, повороти, зміни розміру, колір та інше. Крім того, DOTween має інтерфейс користувача, що спрощує процес розробки, дозволяє зберігати анімацію та легко управляти її параметрами

DOTween має багато готових функцій сповільнення (easing), які дозволяють створювати різні ефекти анімації, такі як "затухання", "еластичність", "викид", "різкий старт" та інші. Крім того, DOTween підтримує корутини (coroutines), що дозволяє створювати складні асинхронні анімації та легко керувати їхнім потоком.

Ось для прикладу як просто буде виглядати код який анімує рух об'єкта вгору (по осі Y) на відстань 100 одиниць протягом 1 секунди.

```
transform.DOMoveY(100, 1);
```

Література:

1. DOTween (HOTween v2) [Електронний ресурс] – Режим доступу до ресурсу: <http://dotween.demigiant.com/index.php>
2. How to Use DOTween in Unity: Basics - Yarsa DevBlog [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.yarsalabs.com/dotween-unity-animation-part2/>

РОЗРОБКА ГРИ "LASTBATTLE" З ВИКОРИСТАННЯМ ІГРОВОГО РУШІЯ UNITY МОВОЮ C#

Марковський П.П., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Розробка ігор та ігрофікація процесів стають все більш популярними в різних сферах, включаючи освіту та бізнес. Гра "LastBattle" є однією з таких ігор, яка може бути використана для ігрофікації різних процесів.

"LastBattle" - це гра в жанрі shoot 'em up, де гравець контролює космічний корабель і знищує ворожі кораблі, використовуючи різні види зброї та уникає ворожі атаки. Ця гра може бути застосована для ігрофікації процесів навчання, в основному підвищення стратегічного мислення та реакції гравців.

У даній темі буде розглянуто процес розробки гри "LastBattle" з використанням ігрового рушія Unity та мови програмування C#. Розробка гри включає в себе такі етапи, як проектування гри, розробка графічних

елементів та ефектів, створення логіки гри та інтерфейсу користувача.

В рамках даної теми будуть розглянуті такі аспекти розробки гри "LastBattle", як:

- Проектування гри та розробка графічних елементів та ефектів
- Створення логіки гри та інтерфейсу користувача
- Використання ігрового рушія Unity та мови програмування C#
- Оптимізація та тестування гри

Результатом роботи буде готова гра "LastBattle", яка може бути використана для ігрофікації процесів навчання.

Література:

1. Unity. URL: <https://unity.com/>
2. Unity Documentation:
<https://docs.unity3d.com/ScriptReference/Accessibility.VisionUtility.html>

РОЗРОБКА ГРИ "ЗМІЙКА" ЖАНРУ АРКАДА МОВОЮ PYTHON"

Носач Д.М., Аверічев І.М.

Державного університету телекомунікацій, м. Київ

Постановка завдання

Завдання полягає у розробці гри "Змійка" в жанрі аркада за допомогою мови програмування Python. Гра повинна мати наступні функціональні вимоги:

Гравець повинен керувати змією, яка рухається по ігровому полю, збираючи їжу та уникаючи перешкод.

Їжа повинна з'являтися на ігровому полі випадковим чином, і змія повинна збирати її, збільшуючи свою довжину.

Якщо змія зіткнеться зі стіною або своїм тілом гра закінчується.

Гра повинна мати рівні складності, які змінюються з кожним рівнем. Наприклад, рівень може змінювати швидкість руху змійки, розмір поля, кількість їжі та перешкод.

Гра повинна мати систему балів та рейтингу. Гравець повинен отримувати очки за збирання їжі та їхня кількість повинна відображатися на екрані. Рейтинг повинен зберігатися між іграми та бути доступним для перегляду.

Гра повинна мати графічний інтерфейс користувача з декількома елементами: ігрове поле, змія, їжа та перешкоди.

Гра повинна мати звуковий ефект під час зіткнення змійки з перешкодою

або збирання їжі.

Мета дослідження

Метою даного дослідження є розробка гри "Змійка" в жанрі аркада за допомогою мови програмування Python. Головною метою гри є набрати якомога більше очок, збираючи їжу та уникаючи перешкод, а також досягти якомога вищого рівня складності.

Результат дослідження

Дослідження розробки гри "Змійка" в жанрі аркада мовою програмування Python має на меті розробити програму, що дозволить користувачам насолоджуватися геймплеєм та конкурувати з іншими гравцями. Результатом дослідження є готова програма з графічним інтерфейсом, яка включає елементи гри, такі як ігрове поле, змійку, їжу та перешкоди.

Програма дозволяє гравцям керувати змійкою, збирати їжу та уникати перешкод, отримуючи очки за кожен зібраний об'єкт. Гра містить звукові ефекти, які додають геймплею більшої реалістичності. Рейтинг гравця зберігається між іграми, щоб дозволити гравцям порівнювати свої результати та покращувати свої навички.

У процесі розробки гри використовувались різноманітні функції та бібліотеки мови програмування Python, такі як Pygame, яка використовується для роботи з графікою та звуком, та Random, яка використовується для генерації випадкових чисел.

Результатом дослідження є готова програма, яка може бути використана для насолоди геймплеєм та конкурування з іншими гравцями. Дана програма може бути використана в навчанні молодших школярів основ програмування, а також в сфері розважальних заходів, які залучають увагу молодіжної аудиторії.

Висновки та перспективи

Під час дослідження розробки гри "Змійка" в жанрі аркада мовою програмування Python було успішно розроблено програму з графічним інтерфейсом, що дозволяє користувачам насолоджуватися геймплеєм та конкурувати з іншими гравцями. Програма містить елементи гри, такі як ігрове поле, змійку, їжу та перешкоди, звукові ефекти, які додають геймплею більшої реалістичності, а також рейтинг гравця, що зберігається між іграми.

В подальшому можна розширити функціональність гри, додавши нові елементи, наприклад, бонуси, що дозволять гравцеві отримувати додаткові бали та покращувати свої результати. Також можна розробити версію гри, яка дозволить гравцям грати онлайн з іншими користувачами, що дозволить збільшити інтерактивність та зробити гру більш цікавою.

Література:

1. "Python Game Programming by Example" автора Alejandro Rodas de Paz
2. "Invent Your Own Computer Games with Python" автора Al Sweigart

ПРОЦЕС РОЗРОБКИ ІГОР ЗА ДОПОМОГОЮ PYTHON І PYGAME

Носач Д.М., Аверічев І.М.

Державного університету телекомунікацій, м. Київ

Постановка завдання

Розробка класичної гри "Змійка" жанру аркада мовою Python з використанням модуля Pygame. Гравець керує змією та збирає їжу, збільшуючи довжину змійки. Гра закінчується, якщо змійка зіткнеться зі стіною або зі своїм власним хвостом. Для реалізації необхідно створити класи для змійки, їжі та стін, а також основний цикл гри, що оновлюватиме стан гри на кожному кроці.

Мета дослідження

Дослідження включатиме кілька етапів і метою цього дослідницького проекту є розробка класичної аркадної гри під назвою «Змійка» з використанням мови програмування Python і модуля Pygame. У грі гравець буде керувати змією та збирати їжу, щоб збільшити її довжину. Гра закінчиться, якщо змія зіткнеться зі стіною або власним хвостом. Для досягнення цієї мети проект передбачає розробку класів для змії, їжі та стін, а також основного ігрового циклу, який оновлює стан гри на кожній ітерації.

Дослідження включатиме вивчення фундаментальних концепцій розробки ігор, включаючи об'єктно-орієнтоване програмування, ігрову механіку, графіку та ведення гри користувачами. Проект вимагатиме глибокого розуміння програмування на Python, включаючи структури даних, цикли, умови та функції. Модуль Pygame також буде детально досліджено, включаючи його вбудовані функції для обробки графіки, звуку та введення користувача.

Дослідження включатиме кілька етапів, включаючи планування та розробку гри, впровадження коду, а також тестування та налагодження гри, щоб переконатися, що вона працює належним чином. Гру розроблятимуть за допомогою ітераційного процесу, причому кожна ітерація передбачає додавання нових функцій, тестування гри та внесення необхідних коригувань і вдосконалень.

Кінцева мета проекту — створити функціональну та приємну гру, у яку зможуть грати користувачі будь-якого рівня кваліфікації. Дослідження дасть цінну інформацію про процес розробки ігор за допомогою Python і Pygame, а також зробить внесок у наявні знання про розробку ігор. Гра також може мати практичне застосування в освіті та розвагах і потенційно може бути вдосконалена та модифікована для створення нових ігор або програм.

Результат дослідження

Дослідження включало кілька етапів, включаючи планування та розробку гри, впровадження коду, а також тестування та налагодження гри, щоб переконатися, що вона працює належним чином. Гру було розроблено за

допомогою ітераційного процесу, причому кожна ітерація передбачала додавання нових функцій, тестування гри та внесення необхідних коригувань і вдосконалень.

Результатом проекту стала функціональна та цікава гра, в яку можуть грати користувачі будь-якого рівня кваліфікації. Дослідження дало цінну інформацію про процес розробки ігор за допомогою Python і Pygame, а також допомогло поповнити наявні знання про розробку ігор. Гра також може мати практичне застосування в освіті та розвагах і потенційно може бути вдосконалена та модифікована для створення нових ігор або програм.

Висновки та перспективи

Результатом проекту стала функціональна та цікава гра, в яку можна грати користувачам будь-якого рівня кваліфікації. Дослідження дало цінну інформацію про процес розробки ігор за допомогою Python і Pygame, а також допомогло доповнити наявні знання про розробку ігор. Гра також може мати практичне застосування в освіті та розвагах і може бути вдосконалена та модифікована для створення нових ігор або програм.

Література:

1. "Python Game Programming by Example" автора Alejandro Rodas de Paz
2. "Invent Your Own Computer Games with Python" автора Al Sweigart

ДОСЛІДЖЕННЯ ВІДЕОІГОР В ЖАНРІ “ШУТЕР”

Олексенко Р.Г., Дібрівний О.А.

Державного університету телекомунікацій, м. Київ

Навчально-науковий інститут інформаційних технологій

Шутер - це жанр відеоігор, в яких головними механіками є використання зброї та бойові вміння, щоб вбити ворогів. Це може бути як перша, так і третя особа, залежно від конкретної гри. Жанр шутерів поділяється на декілька видів, таких як екшн-шутери, тактичні шутери, кіберпанк-шутери та інші, але найбільш популярні з них є шутери в першій особі.

У шутерах в першій особі гравець зображується як персонаж у грі, і все, що він бачить, відбувається через його очі. Це дає можливість бути більш іммерсивним і відчувати себе як частина гри. Більшість шутерів в першій особі мають одиночну кампанію, в якій гравець проходить різні рівні та зустрічається з різними ворогами, та багатокористувацький режим, в якому гравці з різних частин світу можуть змагатися один з одним у режимах збереження прапора, знищення ворожих команд та інших.

Більшість шутерів мають широкий арсенал зброї, від пістолетів та гвинтівок до мечів та різноманітних бойових інструментів. Крім того, гравці можуть використовувати спеціальні вміння, такі як гранати, підривники та бойові

дрони, для того, щоб знищити ворогів та вижити в битві. У шутерах також можуть бути різні умови для перемоги, від того, щоб вбити певну кількість ворогів, до знищення ворожих баз або захисту об'єктів.

Шутери - це один з найпопулярніших жанрів відеоігор, і вони мають велику аудиторію серед геймерів. Цей жанр зазвичай приваблює гравців, які люблять динамічну бойову систему та швидкість дій. Крім того, шутери можуть бути використані як засіб для соціалізації з іншими гравцями, зокрема, в багатокористувацьких режимах, де гравці можуть об'єднуватися в команди та працювати разом, щоб перемогти ворогів.

Одним з найбільш популярних шутерів в першій особі є серія "Call of Duty", яка має велику аудиторію гравців по всьому світу. Інші популярні шутери включають серію "Battlefield", "Counter-Strike", "Overwatch" та "Destiny". Шутери також використовуються в багатьох професійних турнірах з електронних спортивних ігор, де гравці змагаються за гроші та славу.

В цілому, шутери - це жанр відеоігор, який надає гравцям можливість відчувати себе частиною динамічної бойової системи та забезпечує широкий арсенал зброї та умінь, щоб перемогти ворогів. Цей жанр продовжує еволюціонувати та ставати все більш популярним серед геймерів.

Література:

1. Video game https://en.wikipedia.org/wiki/Video_game

2. Shooter games

<https://www.quora.com/Shooter-games-are-basically-killing-simulators-Is-it-bad-to-play-them>

НАПРЯМ 5. ХМАРНІ ТЕХНОЛОГІЇ

ХМАРНІ ТЕХНОЛОГІЇ

Виговський Б.С., Залива В.В.

Державного університету телекомунікацій, м. Київ

Хмарні технології – це технології розподіленої обробки цифрових даних, за допомогою яких комп'ютерні ресурси надаються інтернет-користувачеві як онлайн-сервіс. Програми запускаються і видають результати роботи в вікні web-браузера на локальному ПК. При цьому всі необхідні для роботи програми та їх дані знаходяться на віддаленому інтернет-сервері і тимчасово кешуються на клієнтській стороні: на ПК та інших пристроях.

Види хмарних технологій:

1. Публічна хмара - одночасний доступ багатьох користувачів до ІТ-інфраструктури. Але можливості управляти і обслуговувати дану хмару у користувачів немає, вся відповідальність покладена на її власника. Абонентом пропонованих сервісів може стати будь-яка компанія або приватна особа.
2. Приватна хмара - ІТ-інфраструктура, яку контролює і експлуатує тільки один абонент у власних інтересах. Інфраструктура для управління приватною хмарою може розміщуватися або в приміщеннях користувача, або у зовнішнього оператора, або частково у користувача і оператора.
3. Гібридна хмара - це ІТ-інфраструктура, в якій поєднані кращі якості публічної і приватної хмари. Така композиція має унікальні об'єкти, пов'язані між собою стандартизованими або власними технологіями, які дозволяють переносити дані або програми між компонентами.

Види хмарних послуг:

1. Software as a Service (програмне забезпечення як послуга). Даний вид передбачає надання готового рішення для клієнту з мінімальною необхідністю налаштування. Тобто теоретично, підписуючись на такий сервіс, керувати ним може будь-який користувач з мінімальним залученням системного адміністратора або взагалі без нього. Найвідоміші представники такого сервісу у корпоративному середовищі — це Office 365.
2. Platform as a Service (платформа як послуга). Послуги типу **PaaS** розраховані в першу чергу на розробників. Вони являють собою набори готових компонентів для створення додатків, а також фреймворки для керування платформою. В даному випадку компонентами будуть сервіси даних, репозиторії, інструменти автоматизованого деплою, середовища тестування і тому подібні сервіси.
3. Infrastructure as a Service (інфраструктура як послуга). Інфраструктура як послуга за своїми об'єктами та характеристиками найбільш наближена до володіння власним "залізом" та віртуалізацією.

У випадку з IaaS ви отримуєте у своє розпорядження хмарні процесори, пам'ять, диски та мережі, з яких згодом ви створюєте сервери-маршрутизатори та налаштовуєте мережеву топологію так, як вам необхідно.

Переваги та недоліки хмарних технологій.

Основні переваги:

- не потрібні великі обчислювальні потужності ПК - по суті будь-який смартфон, планшет і т.д., при відкритті вікна браузера отримує величезний потенціал.
- відмовостійкість;
- певний рівень безпеки;
- висока швидкість обробки даних;
- економія на покупці софта - всі необхідні програми вже є в сервісі, де будуть працювати додатки;
- Ваш власний вінчестер не наповнюється - всі дані зберігаються в мережі..

Основні недоліки:

- хмарна послуга надається завжди якоюсь компанією, відповідно, збереження даних користувача залежить від цієї компанії;
- поява хмарних монополістів;
- необхідність завжди бути в мережі для роботи;
- небезпека хакерських атак на сервер (при зберіганні даних на комп'ютері ви в будь-який час можете відключитися від мережі і очистити систему за допомогою антивірусу);
- можлива подальша монетизація ресурсу - цілком можливо, що компанії надалі вирішать брати за послуги плату з користувачів.

Література:

- 1.[Електронний ресурс]. – Режим доступу до ресурсу:
<https://edin.ua/shho-take-xmarni-texnologii-i-navishho-voni-potribni/>
2. [Електронний ресурс]. – Режим доступу до ресурсу:
<https://denovo.ua/blog/vidi-hmarnih-servisiv-yakij-obrati-ta-oglyad-hmarnih-provaid-8>
- 3.[Електронний ресурс]. – Режим доступу до ресурсу:
<https://studfile.net/preview/11198634/page:2/>

НАПРЯМ 6. АВТОМАТИЗАЦІЯ ТА УПРАВЛІННЯ БІЗНЕС ПРОЦЕСАМИ

АКТУАЛЬНІСТЬ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ В СУЧАСНОМУ СВІТІ

Медко М.М., Мудрик Я.Ю.

Державного університету телекомунікацій, м. Київ

Наразі існують готові рішення, програмні модулі або цілі екосистеми, які заточені під стандартний набір інструментів в сфері освіти кожної окремої країни. В ході аналізу ринку і конкурентів, було прийнято рішення створити українську платформу дистанційного навчання.

На сьогоднішній день, системи Е-навчання мають наступні недоліки:

- один з головних недоліків – іноземне походження ПЗ;
- недостатньо автоматизації і інформування в деяких процесах;
- всі подібні ПЗ не мають зручного або зовсім відсутній мобільний клієнт;
- можлива підтримка для розширення функціоналу ПЗ.
- Тому, мета даної публікації є розробка платформи для дистанційного навчання, яка має наступні особливості:
 - можливість використання passwordless для користувачів;
 - автоматизація процесу створення групи та приєднання користувачів до неї;
 - автоматизований процес створення звіту успішності студентів;
 - зосередження таких компонентів:
 - структура навчального закладу;
 - розклад;
 - функціонал навчального процесу.
 - покращений процес інформування користувачів системи.

Наразі існує прототип системи у вигляді поєднання мобільного і

Література:

1. Платформи та сервіси дистанційного навчання 11.03.2023 [Електронний ресурс] // Житомирський обласний спортивний ліцей житомирської обласної ради. – 2023. – Режим доступу до ресурсу: <https://regional-lyceum.zt.ua/pedahoham/platformy-ta-servisy-dystantsiinoho-navchannia.html>.
2. Що таке Moodle 11.03.2023 [Електронний ресурс] // Moodle. – 2020. – Режим доступу до ресурсу: <https://moodle.org/mod/page/view.php?id=8174>.
3. Microsoft learn Educator Center 11.03.2023 [Електронний ресурс] // MICROSOFT. – 2023. – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/training/educator-center/?source=mc>.
4. IEEE Std 610.12.1990 09.04.2023 [Електронний ресурс] // ieeexplore. – 2002. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/159342>

АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ДЛЯ РОЗРОБКИ WEB-СЕРВІСІВ

Ковалевська Ю.С., Дзядович О.С.

Державного університету телекомунікацій, м. Київ

Розробка успішних проектів вимагає використання найкращих архітектурних стилів, що дозволять компаніям впроваджувати їх продукти швидше і легше пристосовуватись до ринку.

Метою дослідження є аналіз мікросервісної архітектури, її переваг та актуальності при розробці WEB-сервісів.

Мікросервісна архітектура являє собою набір пов'язаних між собою автономних сервісів, кожен з яких виконує бізнес задачі свого субдомену.

Прикладом може послугувати інтернет-магазин, де будуть окремі мікросервіси для аутентифікації, оплати товару та видачі інвойсів, менеджменту товарів та інші.

Операції в подібних системах в основному будуть виконуватись локально, або будуть розподілені між пов'язаними мікросервісами. Такі розподілені операції можуть виконуватись синхронно з допомогою протоколу HTTP, або асинхронно з допомогою брокерів повідомлень, як Apache Kafka, RabbitMQ, Apache ActiveMQ та інших. Перевагами мікросервісної архітектури є:

- Швидкість розгортання;
- Команди невеликі і розподілені між мікросервісами;

- Слабка зв'язність коду;
- Можливість вибору різних технологій для кожного мікросервісу;
- Масштабованість: легко додавати нові мікросервіси;
- Можливість ізоляції даних: створення різних баз даних для кожного мікросервісу;
- Надійність: якщо один з мікросервісів вийде з ладу, інші залишаться справними і можуть далі виконувати свої функції.

Варто зазначити, що дана архітектура має свої недоліки:

- Складність: система в цілому стає складнішою;
- Складність тестування мікросервісів і їх зв'язків;
- Важливо запроваджувати стандарти для усього проекту, аби полегшити подальшу підтримку
- Ланцюжок залежностей між мікросервісами може стати надто великим, повільним і важко підтримуваним
- Потрібно ретельно слідкувати за цілісністю даних
- Потрібна висококваліфіковані команди, знайомі з даною архітектурою

Якщо компанії хочуть швидко впроваджувати надійні гнучкі системи, то їм варто ознайомитись із мікросервісною архітектурою, адже переваги даної архітектури надають можливість легко створювати легко масштабовані та стабільні WEB-сервіси.

Література:

1. Microservice architecture style. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>.
2. What are microservices?. microservices.io. URL: <https://microservices.io/>.

ВИКОРИСТАННЯ ESP-IDF ДЛЯ АВТОМАТИЗАЦІЇ ФУНКЦІОНУВАННЯ МЕРЕЖІ ЗАРЯДНИХ СТАНЦІЙ ДЛЯ ЕЛЕКТРОМОБІЛІВ

Сігута Д.К., Широкопетлева М.С.

Харківський національний університет радіоелектроніки

Останніми роками завдяки технологічному прогресу та зростаючій турботі про навколишнє середовище стрімко зростає кількість електромобілів. Однією з проблем, пов'язаних з даною тенденцією, є потреба в надійній та зручній зарядній інфраструктурі. Тож розвиток мережі зарядних станцій із використанням автоматизаційних засобів має важливе значення для задоволення

потреб водіїв, а також вирішує такі бізнес-задачі як-от:

- зниження операційних витрат за рахунок мінімізації потреб в обслуговуючому персоналі;
- ефективне масштабування діяльності, оскільки інтеграція нових зарядних станцій до мережі не потребує значних додаткових ресурсів;
- покращення якості обслуговування клієнтів;
- можливість збору даних про моделі електротранспортів, тарифи та інші важливі показники, на основі яких можна корегувати власну діяльність та швидко реагувати на потреби ринку.

Спираючись на раніше проведене дослідження, в якості IoT пристрою для автоматизації функціонування мережі зарядних станцій було обрано мікроконтролер ESP32 [1]. Головним завданням даного дослідження є вибір технології, за допомогою якої буде запрограмовано даний девайс.

Існує декілька фреймворків для розробки прошивки під мікроконтролери ESP32, кожен з яких має свої переваги та недоліки. Розглянемо найпопулярніші з них:

- ESP-IDF (Espressif IoT Development Framework) – офіційна бібліотека для розробки під ESP32 від компанії Espressif, яке надає повний доступ до низькорівневих апаратних функцій і розширених можливостей. Вихідний код пишеться на C/C++ [2].

- Micropython – полегшена реалізація мови програмування Python, призначена для роботи на мікроконтролерах, у тому числі – ESP32. Надає інтерфейс REPL (Read-Eval-Print-Loop), що дозволяє легко експериментувати з пристроєм [3].

- Arduino-ESP32 – складова частина популярного фреймворку (Arduino API), призначеного для спрощеної розробки додатків для вбудованих систем, включаючи пристрої ESP32. Надає інтерфейс до апаратного забезпечення та мову програмування більш високого рівня, що дозволяє легко розпочати розробку [4].

ESP-IDF вимагає більше технічних знань і досвіду, через що процес розробки дещо ускладнюється. Micropython та Arduino-ESP32, у свою чергу, є відносно простими у вивченні та використанні, але власне за рахунок цього вони надають обмежений доступ до низькорівневих апаратних функцій та не гарантують високої продуктивності.

З точки зору бізнесу, ефективність роботи вбудованих систем є надто критичним фактором, який має враховуватись у першу чергу. Спираючись на функціонал, який має забезпечувати кожен IoT пристрій в контексті мережі зарядних станцій для електромобілів [1], мікроконтролер ESP32 одночасно виконуватиме 2 ролі: HTTPS-клієнт для взаємодії з серверною частиною програмної системи та HTTPS-сервер для взаємодії з мобільним клієнтом. Це вимагає деяких ресурсів, яких у вбудованих системах і без того обмаль. Тому в якості ключового інструменту для програмування пристроїв ESP32 варто обрати саме ESP-IDF. Тим паче, дана бібліотека добре задокументована та активно

підтримується як компанією Espressif, так і спільнотою розробників.

Використання ESP-IDF передбачає створення системи за архітектурою, яка складається з наступних рівнів:

- Прикладний рівень: містить основну логіку програми, включаючи код, який ініціалізує апаратне забезпечення та налаштовує різні компоненти системи.

- Компонентний рівень: містить різні програмні компоненти, з яких складається система, такі як драйвери, бібліотеки та ППЗ.

- Рівень драйверів: містить низькорівневі драйвери пристроїв, які безпосередньо взаємодіють з обладнанням, наприклад, драйвери UART, SPI та ін.

- Рівень ППЗ: містить програмні модулі, які забезпечують функціональність вищого рівня, наприклад, мережу, доступ до файлової системи.

- Рівень операційної системи: забезпечує базову операційну систему та керує системними ресурсами, такими як пам'ять, завдання та переривання.

Отже, дана архітектура розроблена з метою забезпечення модульної та гнучкої основи для побудови складних вбудованих систем, зокрема, тих, що можуть відіграти ключову роль в автоматизації функціонування мережі зарядних станцій для електромобілів. Таким чином, у ході даного дослідження ми провели порівняльний аналіз технологій, які можуть бути використані для програмування пристроїв ESP32 та дійшли висновку, що для вирішення бізнес-задач обраної предметної області найефективнішим інструментом є бібліотека ESP-IDF, застосування якої забезпечить надійну та високопродуктивну роботу мікроконтролерів ESP32.

Література:

1. Сігута Д. К., Широкопетлева М. С. Використання IoT пристроїв на основі ESP32 як спосіб автоматизації внутрішніх процесів мережі зарядних станцій для електромобілів. *Матеріали IV форуму «Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології» AERT-2022: Наук. конф.*, м. Харків, 24–25 листоп. 2022 р. Харків, 2022. С. 94–95.

2. ESP-IDF Programming Guide – ESP32. ESP-IDF Programming Guide latest documentation. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/> (дата звернення: 11.04.2023).

3. Quick reference for the ESP32 – MicroPython latest documentation. URL: <https://docs.micropython.org/en/latest/esp32/> (дата звернення: 11.04.2023).

4. Welcome to ESP32 Arduino Core's documentation. URL: https://espressif-docs.readthedocs-hosted.com/projects/arduino-esp32/en/latest/getting_started.html (дата звернення: 11.04.2023).

ДОСЛІДЖЕННЯ АВТОМАТИЗАЦІЇ БІЗНЕСУ ЗА ДОПОМОГОЮ TELEGRAM ТА TELEGRAM ЧАТ-БОТУ

Данилюк О. С., Золотухіна О.А
Державній університет телекомунікацій, м. Київ

Що таке Автоматизація бізнес процесів та для чого це потрібно?

Автоматизація бізнес-процесів — це набір алгоритмів і програмного забезпечення, які компанія використовує для автоматизації повторюваних завдань для досягнення стратегічних операційних і фінансових цілей.

Тобто автоматизацію бізнес процесу використовують для вирішення таких завдань:

- скоротити час проходження бізнес-процесу;
- зменшити людський фактор;
- оптимізувати штат компанії;
- збільшити продуктивність праці;
- зменшити кількість рутинних операцій у щоденній роботі працівників.

Що таке Telegram?

Telegram – популярний та відомий багатьом багатофункціональний месенджер, за допомогою якого можна обмінюватися повідомленнями та медіафайлами, здійснювати відео- та аудіодзвінки, організувати конференції, створювати розраховані на багато користувачів групи та канали.

Аудиторія Telegram

Ядро аудиторії Телеграм — користувачі від 25 до 34 років (30,6%), інші групи приблизно рівні. Окрім підлітків, яких тут лише 7,7%. Тобто, аудиторія у месенджера досить доросла. У телеграмів більше чоловіків, ніж жінок, яких лише 39%. За сферами переважають представники ІТ (77%), 22,5% зайняті у виробництві та 10,6% у торгівлі. Також у месенджері є маркетологи, спеціалісти зі сфер освіти, охорони здоров'я, фінансів та інших. 46% користувачів – наймані працівники, 11% з них – керівники. Бізнесменів у Телеграм небагато – лише 5%, а ось самозайнятих та фрілансерів більше – 15%. Телеграм використовує освічена аудиторія: 60% мають вищу освіту, чверть з них продовжують навчатися після закінчення університету на другій вищій або в магістратурі. Близько 11% користувачів – студенти.

Що таке Telegram чат бот? Та як він допомагає в автоматизації бізнесу ?

Телеграм бот — чат телеграм, який реагує на певні дії користувачів та видає потрібну інформацію автоматично. Прості роботи виконують одну функцію: перевірку тексту, пошук інформації тощо. Більш складні спілкуються з

користувачами та відповідають на запитання. Боти — альтернатива мобільним програмам для компаній. На їх розробку потрібно менше ресурсів бізнесу, а користувачеві не потрібно встановлювати ще одну програму на смартфон. Роботи візьмуть на себе спілкування з клієнтами та навіть оформлять замовлення з оплатою. Тобто, за допомогою телеграм чат боту людина зможе виконати такі дії:

- Подивитися які є нові фільми у кінотеатрі та замовити білети на нього не виходячи з дому.
- Телеграм чат бот може допомогти знайти інформацію або послуги компанії

Література:

1. Сайт Aboutmarketing.info [Електронний ресурс] – Режим доступу: <https://aboutmarketing.info/>

2. Сайт Snovio [Електронний ресурс] – Режим доступу: <https://snov.io/glossary/business-process-automation/>

ВИКОРИСТАННЯ КАФКА ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ТА КОМУНІКАЦІЇ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ

Зайцев І.С., Гаманюк І.М.

Державній університет телекомунікацій, м. Київ

Apache Kafka - це дистрибутивна система для обміну повідомленнями, яка дозволяє ефективно передавати великі об'єми даних між різними застосунками та системами.

При розробці застосунку який повинен масштабуватися виникає декілька проблем:

- Комунікації та обміну повідомлення між вузлами та гарантії доставки повідомлень (Delivery guarantee)
- Рівномірного розподілення навантаження між усіма вузлами застосунку. (Load Balancing)

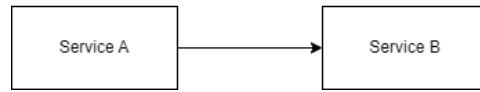


Рисунок 1 - Приклад проблеми комунікації

Використовувати традиційний HTTP для комунікації та Service Discovery для виявлення сервісів проблематично:

- Немає гарантії що повідомлення буде доставлено або що взагалі не буде доставлено декілька раз в разі проблеми з мережею або якщо відправник або отримувач буде не доступний.

- Немає високої пропускної здатності та низької затримки.

Рішення

При використанні Kafka застосунки не комунікують на пряму, замість цього один застосунок відправляє повідомлення в Kafka брокер, другі застосунки підписуються на певні повідомлення та можуть отримувати їх, це дає змогу одному застосунку відправляти повідомлення одночасно для декількох отримувачів.



Рисунок 2 - Приклад комунікації використовуючи Kafka

Kafka зберігає всі отримані повідомлення на протязі періоду який можливо сконфігурувати, якщо якийсь застосунок був не доступний на момент відправки повідомлення то в нього є час для того щоб прочитати повідомлення в майбутньому. (Data retention)

Так як архітектурно додатки не комунікують на пряму це дає змогу підтримувати високу пропускну здатність та низьку затримку. (High throughput and lower latency)

Kafka підтримує масштабування, декілька вузлів може бути запущено для

балансування навантаження між брокерами Kafka. Навіть якщо один із вузлів вийде з ладу та буде не доступний Kafka може продовжувати працювати. (Fault tolerance)

Всі дані в Kafka зберігаються в топіках (topic) та розділах (partition). Топік це як таблиця в базі даних, відправник має змогу відправити повідомлення в топік, а підписники мають змогу підписатися на отримання повідомлень з конкретного топіку. Розділи використовуються для того щоб розподілити навантаження між кількома брокерами Kafka та забезпечити паралельну обробку даних між підписниками, також це є ключовий механізм для підтримки масштабування. (Scalability)

В Kafka є декілька режимів гарантії доставки повідомлень:

Максимум один раз (At most once):

При використанні цього підходу повідомлення можуть бути втрачені, якщо під час обробки станеться збій. Ця гарантія забезпечує найвищу пропускну здатність, але не гарантує 100% доставку повідомлень.

Принаймні один раз (At least once):

При використанні цього підходу повідомлення можуть дублюватися, якщо під час обробки станеться збій, але є гарантія що вони 100% не будуть втрачені.

Точно один раз (Exactly once):

При використанні цього підходу повідомлення не будуть втрачені або дубльовані, якщо під час обробки станеться збій. Щоб досягнути цієї гарантії Kafka використовує транзакції (transaction) та ідемпотентних отримувачів (Idempotent consumer).

Висновок

Враховуючи зазначене, для забезпечення реалізації горизонтального масштабування при створенні дистрибутивної системи виконання відкладених завдань було вибрано Apache Kafka.

Література:

1. Apache Kafka. [Електронний ресурс]. – Режим доступу: <https://kafka.apache.org>

ОСОБЛИВОСТІ СУЧАСНОГО ПЗ В ІКТ

Юхта М.А., Бондарчук А.П.

Державній університет телекомунікацій, м. Київ

Постановка задачі. Застосування програмного забезпечення в ІКТ має великий потенціал для покращення ефективності та продуктивності в різних сферах, включаючи подорожі. Однак, з розвитком Інтернету речей та зростанням популярності мобільних пристроїв, виникає необхідність у використанні спеціально розробленого програмного забезпечення для подорожей.

Мета дослідження. Метою цього дослідження є розгляд застосування програмного забезпечення в ІКТ для цілей подорожей та аналіз його можливостей для покращення ефективності та зручності в подорожах.

Результат дослідження. Застосування програмного забезпечення в ІКТ для цілей подорожей може включати різноманітні програми та додатки, які дозволяють забезпечити маршрутне планування, бронювання готелів та авіаквитків, пошук транспорту, ресторанів та розважальних закладів, а також стеження за погодними умовами та забезпечення безпеки.

Одним з найбільш важливих аспектів використання програмного забезпечення в ІКТ для цілей подорожей є забезпечення високої точності та швидкості збору та обробки даних, що дозволяє зменшити час та зусилля, які необхідні для планування та здійснення подорожі.

При використанні програмного забезпечення в ІКТ для цілей подорожей, важливо також забезпечити безпеку та конфіденційність даних, особливо в тому випадку, якщо вони містять особисту інформацію про користувача.

Висновок. Розробка та використання програмного забезпечення в ІКТ для цілей подорожей має великий потенціал для покращення ефективності та зручності в подорожах, забезпечуючи точне та швидке збору та обробку даних, що дозволяє користувачам зменшити зусилля, які необхідні для планування та здійснення подорожі. Крім того, використання програмного забезпечення дозволяє забезпечити безпеку та конфіденційність даних, що є важливим аспектом для користувачів.

Інтернет речей є новою технологією, яка дозволяє збільшити можливості для використання програмного забезпечення в ІКТ для цілей подорожей. Наприклад, додатки для моніторингу стану транспорту та прогнозування трафіку можуть бути розроблені для використання даних з сенсорів, які забезпечують безпеку на дорогах та контролюють рух транспорту.

У майбутньому, програмне забезпечення в ІКТ для цілей подорожей буде продовжувати розвиватися, забезпечуючи

користувачам нові можливості та функціональність. Таким чином, використання програмного забезпечення в ІКТ для цілей подорожей може бути важливим фактором для покращення ефективності та зручності.

Література:

1. Varga, L. (2016). The role of ICT in tourism industry. *Procedia Economics and Finance*, 39, 679-686.
2. Buhalis, D. (2003). *eTourism: information technology for strategic tourism management*. Pearson Education.
3. Huang, C. D., Li, X., & Li, H. (2017). The Impact of Information and Communication Technology on Tourism Development: A Review from Tourism Resource-Based View. *Sustainability*, 9(9), 1667.
4. Neuhofer, B., Buhalis, D., & Ladkin, A. (2015). Technology as a catalyst of change: enablers and barriers of the tourist experience and their consequences. *The Routledge Handbook of Transport Economics*, 25

ВИКОРИСТАННЯ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ ДЛЯ РОЗРОБКИ WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ

Чернявський Ж.А., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Розробка web-додатку для складського обліку з використанням UML діаграм прецедентів для покращення процесу розробки та підвищення якості програмного продукту

Мета дослідження. Дослідити можливості використання UML діаграм прецедентів для розробки web-додатку для складського обліку, а також визначити переваги використання цих діаграм

Результат дослідження. В результаті було проведено дослідження можливостей використання UML діаграм прецедентів для розробки web-додатку для складського обліку. Було побудовано діаграму UML для мого проекту та описано кожен прецедент, що допомогло при розробці та продемонструвало переваги використання UML діаграми прецедентів. Переваги використання цих діаграм є можливість легко зрозуміти функціональність системи, виявлення проблем та уточнення вимог до системи.

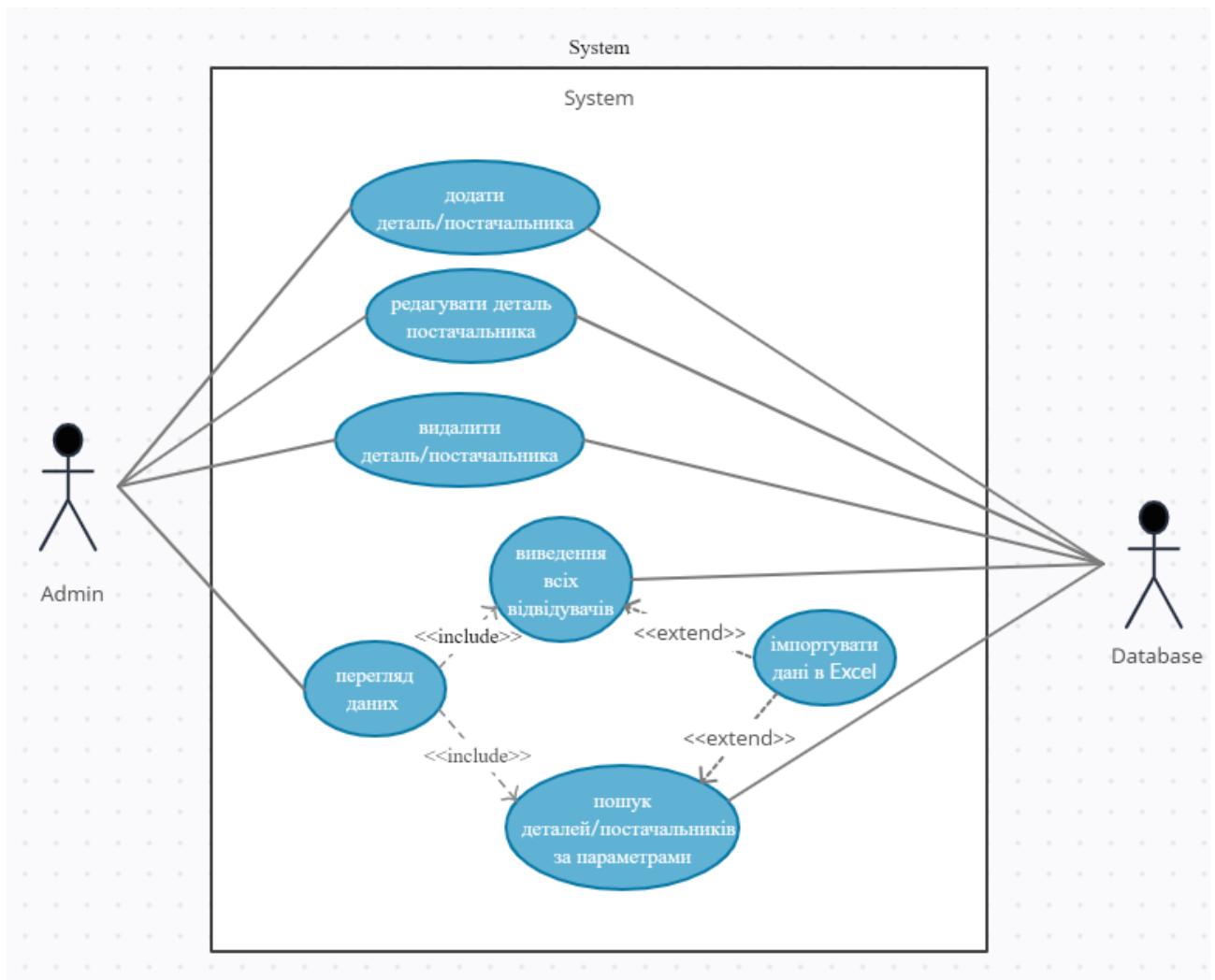


Рисунок 1 – Діаграма прецедентів web-додатку для складського обліку

Опис прецедентів:

5. Додати деталь/постачальника – адміністратор додає нову деталь або постачальника, вписує дані і зберігає зміни.
6. Редагувати деталь/постачальника – адміністратор редагує дані деталі або постачальника і зберігає зміни.
7. Видалити деталь/постачальника - адміністратор видаляє деталь або постачальника і зберігає зміни.
8. Перегляд даних - адміністратор переглядає дані про деталі або постачальників, включаючи перегляд усіх деталей та постачальників, або пошук за конкретними даними з можливістю експортувати обрані дані в Excel таблицю.

Висновок. використання UML діаграм прецедентів може бути корисним для проектування web-додатку для складського обліку. Це допомагає уникнути можливих помилок, що можуть виникнути при неправильному зрозумінні вимог до системи, та дає змогу забезпечити повну та правильну функціональність системи і є корисним інструментом проектування.

Література:

- 1.ASP.NET documentation : <https://learn.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-7.0>
- 2.ASP.NET Core Blazor documentation : <https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-7.0>
- 3.Use Case Diagram notations guide : <https://circle.visual-paradigm.com/docs/uml-and-sysml/use-case-diagram/use-case-diagram-notations-guide/>

АНАЛІЗ ПЕРЕВАГ ВИКОРИСТАННЯ UI-ФРЕЙМВОРКУ RAZOR PAGES У РОЗРОБЦІ WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ НА ПЛАТФОРМІ ASP.NET CORE

Чернявський Ж.А., Гаманюк І.М.

Державного університету телекомунікацій, м. Київ

Постановка задачі. Розробка веб-додатків зазвичай вимагає використання різних інструментів, технологій та середовищ розробки, щоб задовольнити різноманітні вимоги та потреби проекту, що в свою чергу вимагає від розробника значно більших навичок і володінь різними мовами програмування. І для того щоб полегшити розробку можна використовувати UI-фреймворк Razor Pages.

Мета дослідження. Метою цього дослідження є аналіз можливості технології Razor Pages для розробки веб-додатку для складського обліку на платформі ASP.NET Core, Визначити його переваги використання.

Результат дослідження.

Razor Pages - це фреймворк для розробки веб-додатків, що ґрунтується на платформі ASP.NET Core. Цей фреймворк має багато переваг, які роблять його зручним для використання при розробці web-додатку для складського обліку.

Однією з основних переваг Razor Pages є те, що він є частиною платформи .NET, що дозволяє розробникам використовувати різноманітні інструменти та середовища розробки, які підтримують .NET. Наприклад Visual Studio, популярне інтегроване середовище розробки для .NET, що надає повний набір інструментів для розробки веб-додатків на основі Razor Pages.

Платформа .NET дозволяє мені використовувати найкраще знайому мені програмування мову C#.

У веб-розробці з використанням фреймворку ASP.NET Core та технології Razor Pages існує можливість вставки C# коду безпосередньо в HTML сторінки. Це дає можливість уникнути потреби використовувати JavaScript для динамічного змінення вмісту сторінки.

```

70
71  @* will render city checkboxes with names in format of 'CitiesList[i].HasGoodWeather'
72      which will bind to viewModel *@
73  <div>
74      Cities (for):
75      <div>
76          @for (int i = 0; i < Model.CitiesList.Count(); i++)
77          {
78              <label>@Model.CitiesList[i].CityName</label>
79              @Html.CheckBoxFor(m => m.CitiesList[i].HasGoodWeather)
80          }
81      </div>
82  </div>
83
84

```

Рисунок 1 – приклад вставки C# коду в HTML сторінку в моїй дипломній роботі

Всі компоненти додатку, включаючи стилі, скрипти, розмітку та бізнес-логіку, можуть бути розміщені в одному проекті, бо Razor Pages дозволяє розробляти веб-додатки в одному solution в Visual Studio Це полегшує розробку, тестування та супровід додатку, оскільки всі компоненти доступні в одному місці.

Крім того, Razor Pages забезпечує зручний механізм для обробки запитів, який не вимагає використання контролерів та дій. Замість цього, він дозволяє використовувати сторінки (Pages), які містять весь необхідний код для обробки запитів. Це дозволяє скоротити кількість коду, який потрібно написати, та полегшує розробку.

Також перевагою є те, що Razor Pages має зручний шаблон для розробки CRUD-операцій, який можна використовувати для швидкої розробки сторінок додатку. Цей шаблон автоматично створює сторінки для створення, читання, оновлення та видалення записів з бази даних, що дозволяє зосередитися на бізнес-логіці додатку, а не на написанні багатьох однотипних сторінок.

Висновок. Razor Pages є зручним та ефективним вибором для розробки складського обліку на платформі ASP.NET Core, з великою кількістю переваг, наприклад інструментів які підтримують .NET, що є великим плюсом для програмістів, що мають більший фокус на використання більш знайомої мови програмування C#.

Література:

1. ASP.NET documentation : <https://learn.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-7.0>
2. Razor Pages in ASP.NET Core documentation : <https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-7.0&tabs=visual-studio>

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ENTITY FRAMEWORK В РОЗРОБЦІ БАЗ ДАНИХ НА ПЛАТФОРМІ .NET

Решетнік Н.О., Трінтіна Н.А.

Державного університету телекомунікацій, м. Київ

Майже кожен перший додаток який вже існує або тільки створюється, потребує зберігання різного роду даних, для цього розробники використовують бази даних. В даному прикладі розглянемо реляційні бази даних, для в взаємодії з якими використовується мова запитів SQL. Тобто і будь-який додаток який потребує доступу до БД, теж має його використовувати, відсилати різні запити мовою SQL і отримувати результати. Але у цього примітивного підходу є серйозні мінуси, наприклад: SQL код треба зберігати в кодї додатку, що робить код додатку роздутим, також цей код тяжче зробити динамічним і оптимізованим, і ще й з можливістю перевикористання. В додачу до цього, SQL запити в кодї додатку будуть виглядати як просто великі стрічки тексту, що не дуже лягає на концепцію ООП які використовують зараз більшість мов програмування та звикли оперувати об'єктами. І тому використання реляційної бази даних для зберігання об'єктно-орієнтованих даних приводить до семантичного провалу, примушуючи програмістів писати програмне забезпечення, яке повинне вміти як обробляти дані в об'єктно-орієнтованому вигляді, так і вміти зберегти ці дані в реляційній формі. Тому, за для вирішення цих проблем було винайдено технологію ORM.

ORM - технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». В даному випадку ми будемо використовувати Entity Framework як приклад.

Entity Framework (EF) - це об'єктно-орієнтований інструмент для роботи з БД на платформі .NET, та який створений для вирішення проблем вище. EF підтримує 3 моделі синхронізації з БД для подальшої роботи:

1. Code First – у випадку коли готової бази даних ми ще не маємо, то ми можемо створити сутності якими ми будемо оперувати, налаштувати їх зв'язки та обмеження, і потім за допомогою Entity Framework згенерувати базу даних яка буде містити в собі всі необхідні таблиці та зв'язки
2. Database First - передбачає створення бази даних спочатку (або вже мати створену), а потім моделі бази даних EDMX. Файл .edmx містить інформацію про структуру бази даних, модель даних та її відображення, і може бути використаний у середовищі для програмування для генерації класів у кодї.
3. Model First – передбачає першочергове створення моделі графіку EDMX, а потім класи у кодї створюються автоматично за лаштунками. А База даних створюється з графіка EDMX.

Також одним із плюсів використання Entity Framework є його інтегрованість з мовою запитів LINQ що дозволяє писати запити до бази даних в термінах об'єктів що дозволяє зробити код більше зрозумілим, та зменшити його об'єм. Ще одною перевагою можна назвати внутрішню оптимізацію запитів до бази даних, яка хоч і не завжди зробить запити більш швидкими ніж прямі запити до бази даних за допомогою SQL, але скоріш зробить їх краще ніж їх би написав середньо статистичний розробник.

Серед інших переваг використання технології Entity Framework є підтримка транзакцій та контексту. Контекст - це об'єкт, який зберігає зміни до об'єктів моделі даних та здійснює їх синхронізацію з базою даних. Завдяки цьому можна забезпечити консистентність даних в базі даних та уникнути проблем з її цілісністю.

Однак, на жаль, використання ORM має й свої недоліки. Наприклад, збільшення часу виконання запиту до бази даних, або складність використання Entity Framework в середовищах з великою кількістю даних. На великих обсягах даних може виникнути проблема з продуктивністю та ресурсами системи, що може призвести до зависання додатку на деякий час.

Також, використання технології Entity Framework вимагає додаткового знання технологій та практик для забезпечення безпеки та цілісності даних. Для успішного використання Entity Framework, необхідно мати досвід роботи з реляційними базами даних, знати основи об'єктно-орієнтованого програмування та бути знайомим з патернами проектування.

На завершення хочу сказати, що технологія ORM, зокрема Entity Framework є потужним інструментом для розробки, та взаємодії з базами даних на платформі .NET. Вона забезпечує зручний та ефективний доступ до даних, спрощує роботу з базами даних та дозволяє зосередитися на розробці бізнес-логіки. Однак, перед використанням технології, необхідно ознайомитись з її основами і методиками, та знати, які проблеми можуть виникнути в процесі розробки та експлуатації. Також, варто бути обережним при побудуванні запитів та звернути увагу на продуктивність та безпеку даних при використанні в середовищах з великою кількістю даних.

Література:

1. Pros and Cons of Entity Framework - Режим доступу до ресурсу: <https://www.ablison.com/pros-and-cons-of-entity-framework/>
2. Об'єктно-реляційне відображення - Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Об%27єктно-реляційне_відображення

Зміст

НАПРЯМ 1. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕЛЕКОМУНІКАЦІЙ	5
РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ ФІНАНСАМИ РОСКЕТPENNY МОВОЮ PYTHON	5
Диндар А.В., Трінтіна Н.А.....	5
ВИЗНАЧЕННЯ ВИМОГ ДО ТЕЛЕГРАМ-БОТУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ ЗІ СТУДЕНТАМИ . Анісімов Б. А., Золотухіна О. А.	6
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЩОДО АНАЛІЗУ КОМУНАЛЬНИХ ВИТРАТ З ВИКОРИСТАННЯМ UML ДІАГРАМИ СТАНІВ	7
Висоцький В.В., Гаманюк І.М.	7
СХЕМОТЕХНІКА: БЮЧП.....	8
Кужентський О.Г., Лемешко А.В.	8
РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ ПРАВИЛ ЕТИКЕТУ МОВОЮ C#.....	10
Король Р.Є., Аверічев І.М.....	10
ВИМОГИ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА ПРИ РОЗРОБЦІ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ	11
Думенко І.О., Гаманюк І.М.....	11
АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЇ ENTITY FRAMEWORK CORE ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ВІДВІДУВАЧІВ ВИСТАВКИ.....	12
Михайленко В.Д., Гаманюк І.М.	12
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ВІДВІДУВАЧІВ ВИСТАВКИ З ВИКОРИСТАННЯМ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ	15
Михайленко В.Д., Гаманюк І.М.	15
ВИКОРИСТАННЯ HTML, CSS, JAVASCRIPT ПРИ РОЗРОБЦІ FRONTEND-ЧАСТИНИ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ.....	16
Думенко І.О., Гаманюк І.М.....	16
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЩОДО АНАЛІЗУ КОМУНАЛЬНИХ ВИТРАТ З ВИКОРИСТАННЯМ UML ДІАГРАМИ СТАНІВ	17
Висоцький В.В., Гаманюк І.М.	17
АНАЛІЗ СУЧАСНИХ ІНСТРУМЕНТІВ ТА МЕТОДІВ СТВОРЕННЯ ВЕБСАЙТІВ	19
Малюк В.В., Яскевич В.О.....	19
РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ СТВОРЕННЯ ІНДИВІДУАЛЬНОГО ПЛАНУ ХАРЧУВАННЯ ТА ТРЕНУВАНЬ ЛЮДИНИ	21
Малюк В.В., Яскевич В.О.....	21
ЗАБЕЗПЕЧЕННЯ ВІДПОВІДНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО СТАНДАРТІВ БЕЗПЕКИ, ЩО СПРИЯЄ ПІДВИЩЕННЮ НАДІЙНОСТІ ТА ДОВГОВІЧНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	22
Кужентський О.Г., Лемешко А.В.	22
ВИМОГИ ДО BACKEND- ЧАСТИНИ ПРИ РОЗРОБЦІ ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ.....	24
Дубей Р.Р., Гаманюк І.М.....	24
ВИКОРИСТАННЯ .NET CORE ПРИ РОЗРОБЦІ BACKEND-ЧАСТИНИ ЗАСТОСУНКУ ДЛЯ	

ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ	
Дубей Р.Р., Гаманюк І.М.....	25
АНАЛІЗ ТА РОЗРОБКА ВЕБ-РЕСУРСІВ З ПІДТРИМКОЮ ІНКЛЮЗИВНОГО ІНТЕРФЕЙСУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ З ВИКОРИСТАННЯМ СПЕЦІАЛІЗОВАНИХ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ.....	
Юхимець Б.С.....	26
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ПРОЦЕСУ НАДАННЯ ПЕРШОЇ ДОПОМОГИ.....	28
Спіцин А.Я., Трінтіна Н.А.	28
ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ТА ПІДТРИМУВАНOSTІ СЕРВЕРНОГО ДОДАТКУ ЗА ДОПОМОГОЮ БАГАТОРІВНЕВОЇ АРХІТЕКТУРИ.....	
Решетнік Н.О., Трінтіна Н.А.	29
АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБМІНУ НЕПОТРІБНИМИ ПОДАРУНКАМИ. Конох В.В., Золотухіна О.А.....	31
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОНТРОЛЮ ЗДОРОВ'Я ДОМАШНІХ ТВАРИН МОВОЮ С#.....	
Смирнова К.Б., Негоденко О.В.....	33
РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ КАЛЕНДАРНОГО ПЛАНУВАННЯ ОСОБИСТИХ ЗАДАЧ МОВОЮ С# НА БАЗІ ПЛАТФОРМИ ASP.NET	
Скидан П.В., Негоденко О.В.	Error! Bookmark not defined.
МІКРОСЕРВІСНА АРХІТЕКТУРА ПРИ РОЗРОБЦІ ТЕЛЕГРАМ-БОТІВ	
Скидан П.В., Негоденко О.В.	34
ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ТА ЗАБЕЗПЕЧЕННЯ БІЛЬШ ДОВГОГО ЖИТТЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
Кужентський О.Г., Лемешко А.В.	35
ВИВЧЕННЯ МЕТОДІВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЇХ ВИКОРИСТАННЯ В ПРОЦЕСІ РОЗРОБКИ ПЗ.....	
Іванов О. П.	36
РОЗРОБКА ІТ-РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ГОДУВАННЯ ТВАРИН НА ОСНОВІ СИСТЕМИ VARF	
Місіна Н.В., Яскевич В.О.	37
АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ТА ІНСТРУМЕНТІВ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІКТ.....	
Студент Б.М., Негоденко О.В.....	39
РОЗРОБКА WEB-ДОДАТКУ ДЛЯ ДОПОМОГИ СЕРВІСНИМ ЦЕНТРАМ.....	
Кутняк М.Ю., Трінтіна Н.А.	42
ФРЕЙМВОРКИ REACT ТА ANGULAR.....	
Жовніренко Є.В., Яскевич В.О.....	43
РОЗРОБКА WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ МОВОЮ С# НА ПЛАТФОРМІ ASP.NET CORE ТА BLAZOR	
Чернявський Ж.А., Гаманюк І.М.....	45
РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ХАРЧУВАННЯ ДОМАШНІХ УЛЮБЛЕНЦІВ ПО СИСТЕМІ VARF	
Місіна Н.В., Яскевич В.О.	46

АКТУАЛЬНІСТЬ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ПОШУКУ ПОПУТНОГО ТРАНСПОРТУ ДЛЯ ПЕРЕДАЧІ ПОСИЛОК.....	
Горбань А.М., Золотухіна О.А.	48
АНАЛІЗ І ПОРІВНЯННЯ РІЗНИХ ПІДХОДІВ ДО ПРОЕКТУВАННЯ RESTFUL ВЕБ-СЕРВІСІВ...	
Кужентський О.Г., Лемешко А.В.	50
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНИТОРИНГУ ОСОБИСТИХ СПОРТИВНИХ ТРЕНУВАНЬ З ВИКОРИСТАННЯМ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ.....	
Луценко І.В., Гаманюк І.М.	52
ФРЕЙМВОРКИ ASP.NET CORE та JAVA SPRING.....	
Каліш В. В., Яскевич В.О.	54
ВИКОРИСТАННЯ SERVICE DISCOVERY ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ	
Зайцев І. С., Гаманюк І.М.	55
ПЕРЕВАГИ БІБЛІОТЕКИ REACT ДЛЯ РОЗРОБКИ WEB-ДОДАТКІВ.....	
Федоренко А.А., Трінтіна Н.А.....	58
РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ ОРГАНІЗАЦІЇ ОПИТУВАНЬ СТУДЕНТІВ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ REACT. JS	
Шевчук Ю.О., Залива В.В.....	59
СУЧАСНІ ЗАСОБИ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ В ІКТ	
Волошин В.В., Негоденко О.В.	61
РОЗРОБКА НАВЧАЛЬНОГО МОДУЛЯ ЗА ТЕМОЮ"КОМБІНАТОРИКА" МОВОЮ С#	
Черкасов К.А., Корецька В.О.....	62
ВИДИ ТА МЕТОДИ ТЕСТУВАНЬ ПРОДУКТИВНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ	
Бобков Р.А.....	64
РОЗРОБКА МЕТОДІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ RESTFUL ВЕБ-СЕРВІСІВ.....	
Кужентський О.Г., Лемешко А.В.	66
РОЗРОБКА МЕСЕНДЖЕРУ НА ANDROID	
Виходцев М.М., Трінтіна Н.А.	67
ВИКОРИСТАННЯ ІКТ ДЛЯ ПОКРАЩЕННЯ ДІАГНОСТИКИ ТА ЛІКУВАННЯ ДІАБЕТУ	
Петренко Е.Д., Трінтіна Н.А.....	69
РОЗРОБКА TELEGRAM-БОТА ДЛЯ ПОШУКУ КУЛІНАРНИХ РЕЦЕПТІВ МОВОЮ PYTHON....	
Ялович Д.В, Негоденко О.В.	71
АКТУАЛЬНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ПРОЦЕСУ НАДАННЯ ПЕРШОЇ ДОПОМОГИ.....	
Спіцин А.Я., Трінтіна Н.А.	72
МОНИТОРИНГ АКТИВНОСТІ КОРИСТУВАЧІВ ВЕБ-РЕСУРСУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ AJAX	
Волошко Д.С.	73
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ DRAFTSIGHT	
Степанова А.В.....	75
ОБСЛУГОВУВАННЯ ПК ДЛЯ ЇХНЬОЇ БЕЗПОСЕРЕДНЬОЇ РОБОТИ.....	
Меркулов М.І.	76
DATABASE CONNECTION SOLUTIONS IN SYS ENGINEERING	

Churakhin N.S	78
АКТУАЛЬНІСТЬ КОНТЕЙНЕРИЗАЦІЇ ДОДАТКІВ ЗА ДОПОМОГОЮ DOCKER	
Федоренко М.А, Бурлака В.М.	80
PYTHON – НАЙКРАЩІЙ ВИБІР ДЛЯ СУЧАСНОЇ РОЗРОБКИ ПО	
Бородін Н.В, Лемешко А.В.	81
ТЕСТУВАННЯ ПРОДУКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АБО Ж ПЕРФОРМЕНС ТЕСТУВАННЯ СИСТЕМА МОНІТОРИНГУ МЕРЕЖІ ZABBIX	
Сараненко А.Д.	84
НАПРЯМ 2. БАЗИ ДАНИХ ТА ЗНАНЬ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА ОБРОБКА ІНФОРМАЦІЇ	86
РОЗРОБКА WEB-ПЛАТФОРМИ ДЛЯ АГРЕГАЦІЇ ДАНИХ.....	
Лебідь М.М., Трінтіна Н.А.	86
АНАЛІЗ МОЖЛИВОСТЕЙ ТЕХНОЛОГІЙ ІОТ ДЛЯ РЕАЛІЗАЦІЇ ПЛАТФОРМ З ПЛАНУВАННЯ ПОДОРОЖЕЙ.....	
Юхта М.А., Бондарчук А.П.	87
РОСКЕТРЕННУ: ЯК МОБІЛЬНИЙ ДОДАТОК ДОПОМАГАЄ ЛЮДЯМ КЕРУВАТИ СВОЇМИ ФІНАНСАМИ	
Диндар А.В., Трінтіна Н.А.....	88
ВИКОРИСТАННЯ ЗАСОБІВ МОВИ PYTHON ДЛЯ ВИРІШЕННЯ ЗАДАЧ ОБРОБКИ ТЕКСТУ	
Лукашенко І.П., Золотухіна О.А.	89
ВИКОРИСТАННЯ JPA ТА HIBERNATE ДЛЯ СПРОЩЕННЯ РОБОТИ З БАЗАМИ ДАНИХ У ВЕБ ДОДАТКАХ.....	
Левченко О.О., Трінтіна Н.А.	91
ПЕРЕВАГИ ВИВЧЕННЯ ДИСЦИПЛІНИ "АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ" ЗА ДОПОМОГОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
Комаров О. В., Негоденко О.В.	92
ВИКОРИСТАННЯ WEBSOCKET ДЛЯ КЛІЄНТ-СЕРВЕРНОЇ КОМУНІКАЦІЇ.....	
Шовтенко В.Ю.	93
REDUX TOOLKIT ЯК СПОСІБ ЗБЕРІГАННЯ СТАНУ У ВЕБ-ДОДАТКАХ	
Шовтенко В.Ю.	94
ВИКОРИСТАННЯ NOSQL БАЗ ДАНИХ ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ	
Зайцев І.С., Гаманюк І.М.	95
РОЗРОБКА ПРОФАЙЛЕРУ ПРОДУКТИВНОСТІ SQL-ЗАПИТІВ ЗА ДОПОМОГОЮ СТРЕС- ТЕСТУВАННЯ	
Бондаренко В.В., Гаманюк І.М.	98
ВИКОРИСТАННЯ БІБЛІОТЕКИ SCOTTPLOT.WPF ДЛЯ СТВОРЕННЯ ГРАФІКІВ ПРИ РОЗРОБЦІ ПРОФАЙЛЕРУ ПРОДУКТИВНОСТІ SQL-ЗАПИТІВ ЗА ДОПОМОГОЮ СТРЕС- ТЕСТУВАННЯ	
Бондаренко В.В., Гаманюк І.М.	101
НАПРЯМ 3. СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА НЕЙРОННІ МЕРЕЖІ	103
ВИКОРИСТАННЯ ТЕХНОЛОГІЇ CLOUD-TO-CLOUD У РОЗУМНИХ БУДИНКАХ	
Сарафанюк Р.О., Куфтеріна С.Р.	103
ВИКОРИСТАННЯ .NET MAUI	

Луценко І.В., Гаманюк І.М.	104
ІОТ ТА ШТУЧНИЙ ІНТЕЛЕКТ.....	
Патока В.В., Аверічев І.М.	106
ЗАСОБИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ ДЛЯ АНАЛІЗУ КРИПТОВАЛЮТ	
Патока В.В., Аверічев І.М.	108
ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ ДЛЯ МОБІЛЬНОЇ РОЗРОБКИ.....	
Полтавець Н.В., Шевченко С.М.	109
ВИКОРИСТАННЯ АЛГОРИТМУ SUPERVISED LEARNING ДЛЯ ПРОГНОЗУВАННЯ НА ВЕЛИКИХ НАБОРАХ ДАНИХ	
Поночовний О.В., Садовенко В.С.	110
НАПРЯМ 4. РОЗРОБКА ІГОР ТА ІГРОФІКАЦІЯ ОСВІТНІХ ТА БІЗНЕС ПРОЦЕСІВ..	111
ОСОБЛИВОСТІ РОЗРОБКИ ІГОР ЗА ДОПОМОГОЮ UNITY	
Білоус В.С., Дібрівний О.А.....	111
ВИКОРИСТАННЯ LOW-POLY ГРАФІКИ	
Павліченко В.Ю., Гребенюк В.В.	113
ЗАСТОСУВАННЯ ШАБЛОНУ ПРОЕКТУВАННЯ VISITOR НА ДВИГУНІ UNITY ДЛЯ ОБРОБКИ ВЗАЄМОДІЇ ОБ'ЄКТІВ ГРИ	
Бойко М.С., Дібрівний О.А.....	114
АНАЛІЗ ВПЛИВУ МЕТАВСЕСВІТІВ ТА РОЛЬОВИХ РЕЖИМІВ НА РОЗВИТОК ІНФОКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ	
Торжевський Я. Р., Щербина І. С.	115
РОЛЬОВІ МОДИФІКАЦІЇ У ВІДЕОІГРАХ: РОЗГЛЯД ВПЛИВУ ТА ЗНАЧЕННЯ В ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЯХ	
Щукін І.Ю., Щербина І.С.....	117
РОЗРОБКА МОБІЛЬНОЇ ВІДЕОГРИ НА ЮНІТІ	
Розгон Д.А., Дібрівний О.А.	119
АНІМАЦІЯ В UNITY	
Левчук М.П., Дібрівний О.А.	120
СУЧАСНІ РІШЕННЯ ДЛЯ РОЗРОБКИ ІГОР	
Малярчук М.М., Іщераков С.М.	122
РОЗРОБКА ГРИ МОНОПОЛІЯ	
Ярошевський О.В., Дібрівний О.А.	123
ПРОЕКТУВАННЯ ТА РОЗРОБКА ІГРОВОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ UNITY 2D.....	
Карасовський В.В., Гребенюк В.В.	124
ДОСЛІДЖЕННЯ ВІДЕОІГОР В ЖАНРІ “ВИЖИВАННЯ”	
Вдовін Д.Е., Дібрівний О.А.	125
ТВІНЕРИ І DOTWEEN – ЦЕ ЩО	
Осьмак В.Р., Дібрівний О.А.....	126
РОЗРОБКА ГРИ "LASTBATTLE" З ВИКОРИСТАННЯМ ІГРОВОГО РУШЯ UNITY МОВОЮ C#	
Марковський П.П., Дібрівний О.А.	127
РОЗРОБКА ГРИ "ЗМІЙКА" ЖАНРУ АРКАДА МОВОЮ PYTHON"	
Носач Д.М., Аверічев І.М.	128

ПРОЦЕС РОЗРОБКИ ІГОР ЗА ДОПОМОГОЮ PYTHON І PYGAME	
Носач Д.М., Аверічев І.М.	130
ДОСЛІДЖЕННЯ ВІДЕОІГОР В ЖАНРІ “ШУТЕР”.....	
Олексенко Р.Г., Дібрівний О.А.....	131
НАПРЯМ 5. ХМАРНІ ТЕХНОЛОГІЇ	133
ХМАРНІ ТЕХНОЛОГІЇ	
Виговський Б.С., Залива В.В.	133
НАПРЯМ 6. АВТОМАТИЗАЦІЯ ТА УПРАВЛІННЯ БІЗНЕС ПРОЦЕСАМИ	135
АКТУАЛЬНІСТЬ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ В СУЧАСНОМУ СВІТІ.....	
Медко М.М., Мудрик Я.Ю.	135
АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ДЛЯ РОЗРОБКИ WEB-СЕРВІСІВ.....	
Ковалевська Ю.С., Дзядович О.С.....	137
ВИКОРИСТАННЯ ESP-IDF ДЛЯ АВТОМАТИЗАЦІЇ ФУНКЦІОНУВАННЯ МЕРЕЖІ ЗАРЯДНИХ СТАНЦІЙ ДЛЯ ЕЛЕКТРОМОБІЛІВ	
Сігута Д.К., Широкопетлева М.С.....	138
ДОСЛІДЖЕННЯ АВТОМАТИЗАЦІЇ БІЗНЕСУ ЗА ДОПОМОГОЮ TELEGRAM ТА TELEGRAM ЧАТ-БОТУ	
Данилюк О. С., Золотухіна О.А.....	141
ВИКОРИСТАННЯ КАФКА ДЛЯ ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ ТА КОМУНІКАЦІЇ ПРИ СТВОРЕННІ ДИСТРИБУТИВНОЇ СИСТЕМИ ВИКОНАННЯ ВІДКЛАДЕНИХ ЗАВДАНЬ	
Зайцев І.С., Гаманюк І.М.	142
ОСОБЛИВОСТІ СУЧАСНОГО ПЗ В ІКТ	
Юхта М.А., Бондарчук А.П.	145
ВИКОРИСТАННЯ UML ДІАГРАМИ ПРЕЦЕДЕНТІВ ДЛЯ РОЗРОБКИ WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ	
Чернявський Ж.А., Гаманюк І.М.....	146
АНАЛІЗ ПЕРЕВАГ ВИКОРИСТАННЯ UI-ФРЕЙМВОРКУ RAZOR PAGES У РОЗРОБЦІ WEB-ДОДАТКУ ДЛЯ СКЛАДСЬКОГО ОБЛІКУ НА ПЛАТФОРМІ ASP.NET CORE	
Чернявський Ж.А., Гаманюк І.М.....	148
ВИКОРИСТАННЯ ТЕХНОЛОГІЇ ENTITY FRAMEWORK В РОЗРОБЦІ БАЗ ДАНИХ НА ПЛАТФОРМІ .NET.....	
Решетнік Н.О., Трінтіна Н.А.	150