

ВСТУП

Актуальність теми. В наші часи з'являється все більше і більше нових пристроїв. Вже важко собі уявити людину, що живе в місті яка не користується гаджетами. На сьогоднішній день вже майже у всіх є мобільний телефон, вдома стоїть комп'ютер, чи в портфелі знаходиться ноутбук. Але чим більше цих девайсів випускається, тим більше вони виходять з ладу. На жаль наразі не існує вічного пристрою в кишені кожної людини який не можливо розбити, залити рідиною чи ідеального технологічного процесу при якому не існує браку. Також не можливо не зауважити що зовнішня середа також є досить агресивною для електроніки: підвищена вологість, температура яка в наших широтах може коливатися від -30 до +40, також пил який існує всюди та проводить електричний струм. Тому з'являється все більше і більше сервісних центрів як офіційних так і неофіційних. З офіційними все зрозуміло, вони працюють з компаніями які випускають всі ці пристрої напряду та мають всю необхідну інформацію щодо ремонту цих пристроїв та у разі коли виникають питання мають шляхи зв'язку з цими компаніями. Але неофіційні сервісні центри цією інформацією не володіють та вимушені збирати всю цю інформацію по крупицям в інтернеті. Цей процес довгий та займає не малу частину процесу ремонту. А що якщо об'єднати велику кількість цієї інформації в один сайт? Тоді цей час скоротився значно. Тому веб-додаток для сервісного центру з цією інформацією може бути актуальними та затребуваним серед сервісних центрів.

Об'єкт дослідження – процес допомоги сервісним центрам.

Предмет дослідження – програмне забезпечення для допомоги сервісним центрам.

Мета роботи – спрощення процесу діагностики відео карт та ноутбуків за рахунок веб-додатку мовою Java.

Методи дослідження – методи розробки веб-додатків, методи розробки програмного забезпечення, методи проектування, методи тестування веб-додатків.

Для того щоб досягнути поставленої мети необхідно:

1. Проаналізувати існуючі аналоги і визначити переваги та недоліки
2. Розробити вимоги для веб додатку взявши за основу переваги та недоліки існуючих аналогів та потреб майбутніх користувачів
3. Знайти інформацію для заповнення веб-додатку
4. Проаналізувати технічні засоби для розробки веб-додатку
5. Спроектувати новий веб-додаток
6. Розробити веб-додаток

Галузь використання – додаток може відкрити кожен користувач будь-якої операційної системи застосовуючи як Microsoft Edge Browser, Google Chrome, Apple Safari, Opera, Mozilla Firefox та інші. Додаток допоможе людям які займаються ремонтами в пошуку інформації щодо ремонту техніки.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Поняття «веб-додаток»

Веб-додаток – це програма або програмне забезпечення, що можливо запустити за допомогою браузера. Для більшості веб додатків зовнішній вигляд створюється за допомогою мови розмітки: на сьогоднішній день найпопулярнішою такою мовою є HTML, що розшифровується як HyperText Markup Language та останньою його версією є HTML5. також разом з HTML зазвичай використовують спеціальну мову, яка використовується для опису зовнішнього вигляду сторінок, називається ця мова CSS (Cascading Style Sheets) та за необхідності скриптові мови програмування, такі як: JavaScript, PHP та інші. У той час як для написання серверної частини використовують такі мови програмування як : Java, C#, Ruby, C++, Go та інші.

До основних переваг веб-додатків відносять:

- Можливість застосування на будь-якій операційній системі, на якій підтримуються сучасні браузери
- Відсутність необхідності встановлювати додаткове програмне забезпечення користувачу для доступу до різних веб-додатків
- Користувачу не потрібно встановлювати нове оновлення веб-додатку на свій пристрій
- Менші витрати на розробку та підтримку у порівнянні з настільними програмами

Наразі існує декілька видів веб-додатків:

SPA (Single Page Application) – це одно сторінковий веб-додаток. Його особливістю є те що він взаємодіє з користувачем та динамічно переписує поточну сторінку. Основними перевагами таких додатків є те що їх можна використовувати, розмір додатка не обмежений пам'яттю пристрою та SPA додатки працюють швидше ніж інші оскільки спочатку завантажуються не всі модулі програмного

забезпечення та підгружаються в процесі використання додатку. Яскравими прикладами таких додатків є: Trello, Facebook, Twitter, Gmail та інші.

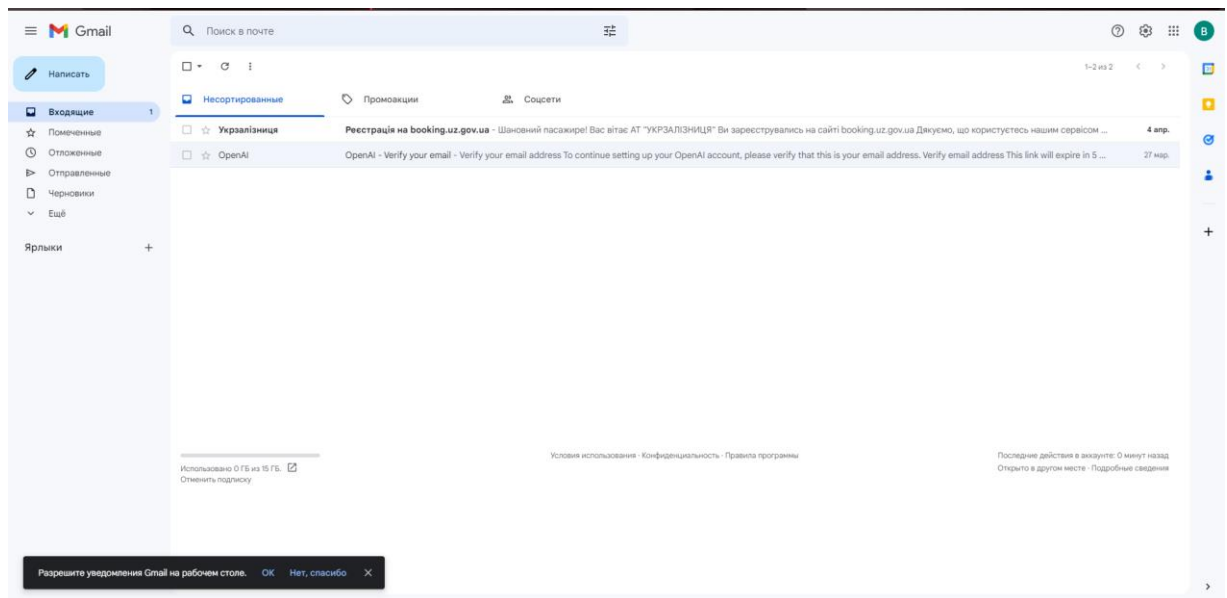


Рисунок 1.1 Приклад інтерфейсу Gmail

MPA (Multi-Page Application) – багатосторінковий веб-додаток. Більшість веб-додатків працюють саме за цим принцип. На відміну від SPA, де при взаємодії з користувачем підвантажується певний модуль на тій само сторінці, у MPA відкривається нова сторінка. Перевагами такого виду веб-додатка можна назвати: легкість оптимізування сторінок для пошукових систем, оскільки кожна сторінка аналізується пошуковою системою окремо та вона чіткіше розуміє що знаходиться на цій сторінці у порівнянні з іншими видами, також MPA зручніше розширювати оскільки у вас немає ліміту на кількість сторінок, також можна назвати плюсом те що більшість сайтів наразі розроблено саме на принципі MPA через що спеціалістів з розробки таких додатків більше, всі методи давно відомі та користувачі звикли до такого виду веб-додатків. Проте у MPA є і недоліки: у цьому виді додатків важко відокремити бек-енд від фронт-енду через що робота розробників ускладнюється, також через це MPA веб-додаток досить важко конвертувати в мобільний додаток та іноді приходиться мобільний додаток писати з нуля. Яскравими прикладами є Amazon, Rozetka, Wikipedia та інші.

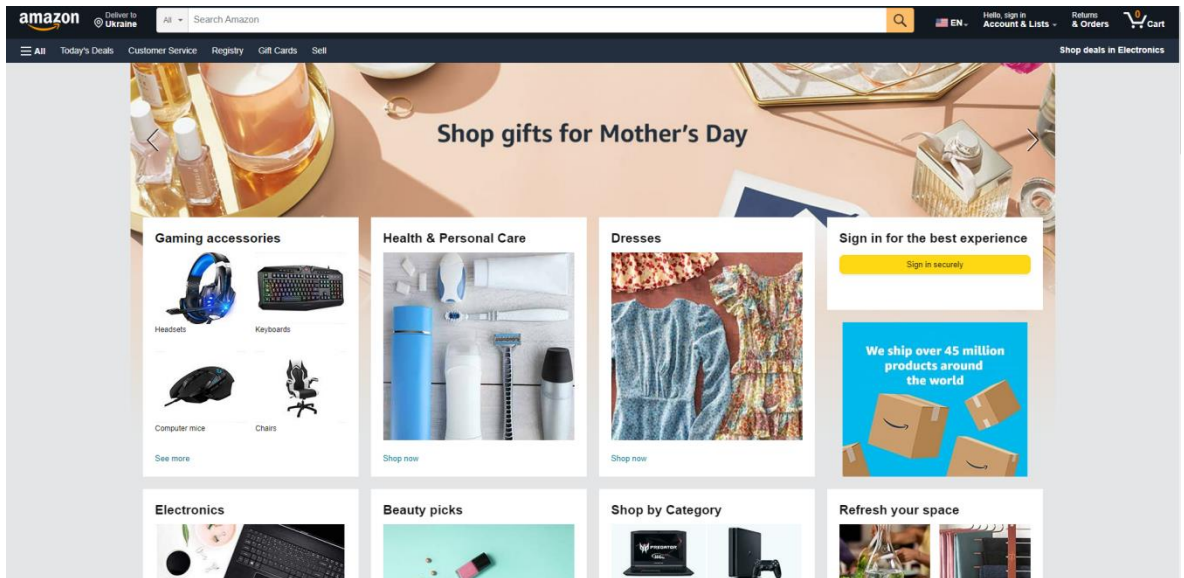


Рисунок 1.2 – Приклад інтерфейсу Amazon

PWA (Progressive Web Application) – це вид веб-додатку який необхідно встановити на свій пристрій, поводить він себе так само як і звичайна веб-сторінка, але також можуть мати функціонал як в мобільних додатках: працювати в автономному режимі, надсилати push-повідомлення та інші. Їх перевагами виділяють: кросплатформеність, оскільки вони все ще веб-додатки та для їх роботи на різних пристроях та операційних системах не потрібно переписувати багато коду, їх можна встановити безпосередньо з сайту без необхідності заходити в магазини додатків, вони займають менше пам'яті у порівнянні з повноцінними мобільними додатками та вони можуть працювати швидше ніж веб-сайти оскільки частина даних може міститися на пристрої користувача та не підгружатися з серверу. Прикладами таких додатків є: Twitter Lite, Uber та інші.

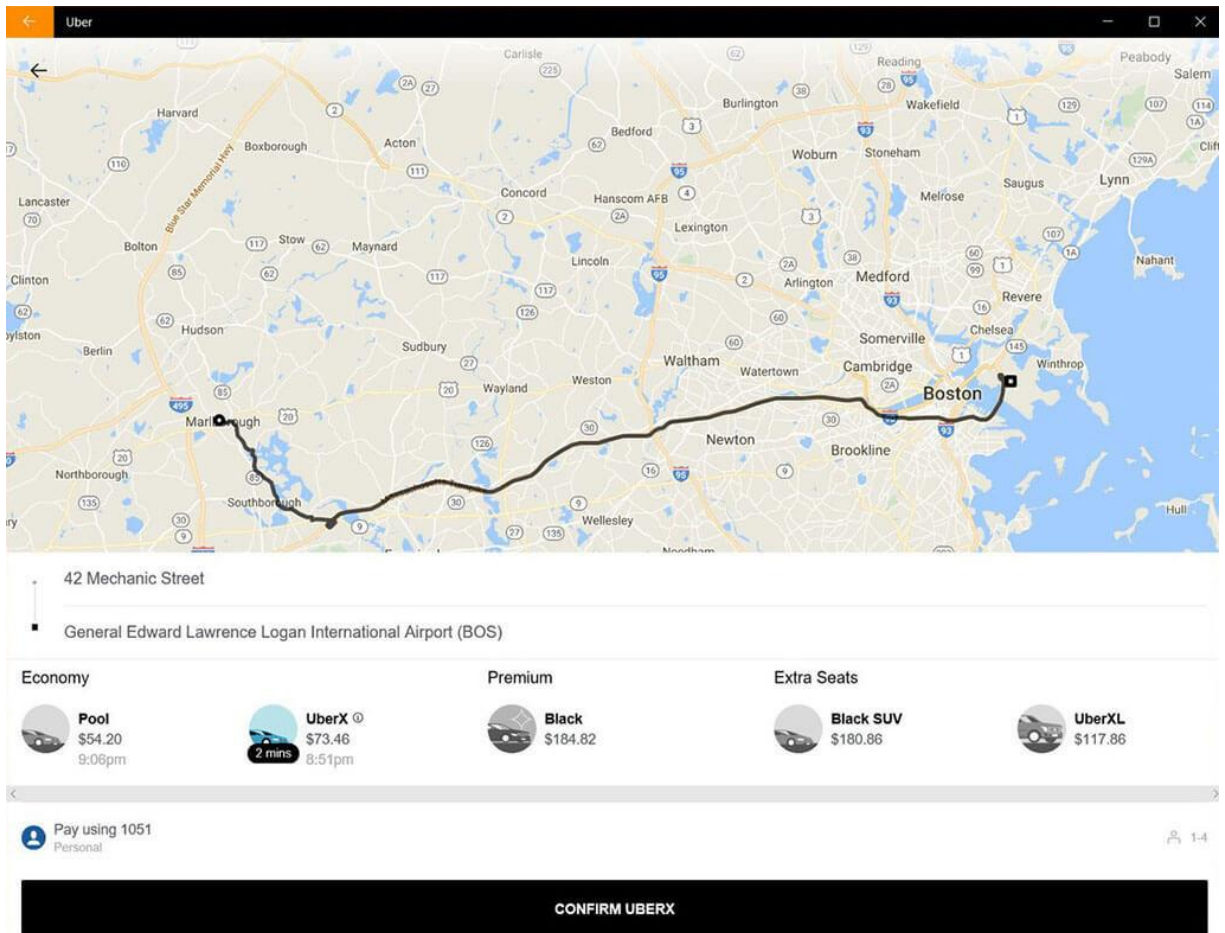


Рисунок 1.3 – Приклад інтерфейсу Uber

1.2. Огляд існуючих аналогів

1.2.1. TechPowerUp

TechPowerUp – це веб-сайт, який спеціалізується на тестах та оглядах комп’ютерного заліза. Сайт був створений у 2004 році та за цей час став одним з найпопулярніших джерел інформації для комп’ютерних ентузіастів та професіоналів. В них на сайті можна знайти тести різних комп’ютерних компонентів, таких як відео карти, процесори, материнські плати та інше. Також у них на сайті можна скачати їх власну програму GPU-Z – програму для отримання інформації про свою відеокарту та моніторингу її стану. Також в них на сайті є форум де користувачі можуть спілкуватися та отримувати допомогу від інших членів спільноти. В них на сайті можна знайти огляди на відео карти з фото компонентної бази їх. Також в них є досить велика база даних з біосами на велику

кількість відео карт. На жаль в них на сайті майже відсутня інформація стосовно ноутбуків. Також відсутня більш детальна інформація щодо замірів супротиву відеокарт, їх boardview та вольтажі які повинні бути присутні на відеокарті. Також відсутня українська мова.

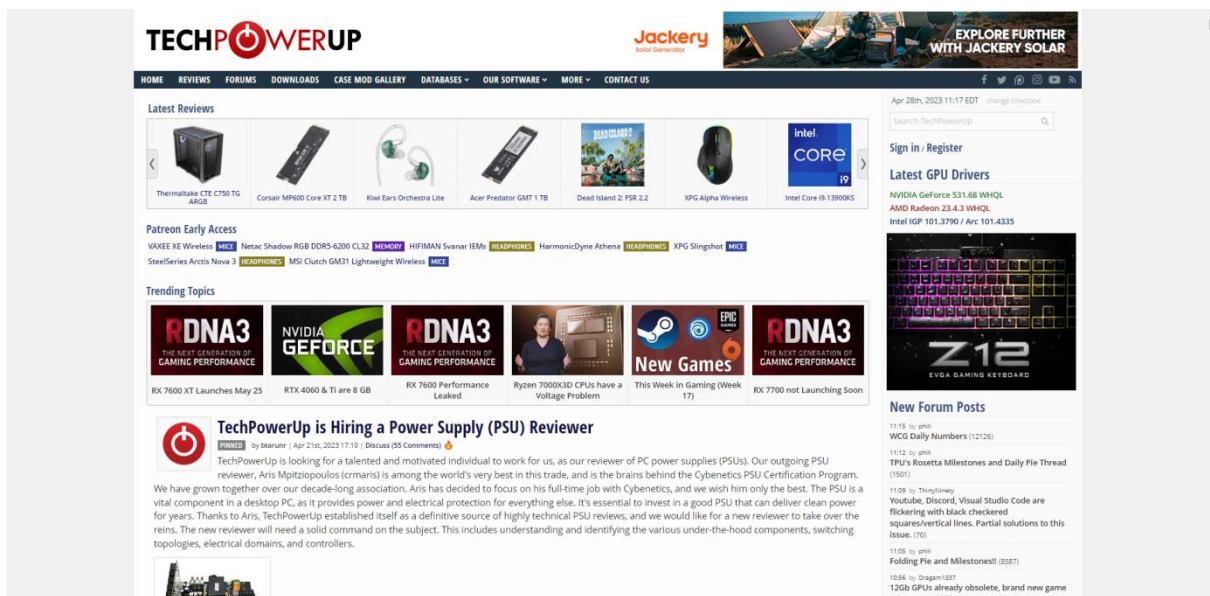


Рисунок 1.4 – Приклад інтерфейсу TechPowerUp

1.2.2. Overclockers.ua

Overclockers – це веб сайт який спеціалізується на тестах та оглядах комп'ютерного заліза. На ньому є можливість прочитати статті українською мовою. На ньому також є фото на яких чітко видно компонентну базу відео карт. На жаль в них так само відсутня інформація щодо ноутбуків, також відсутні біоси, немає замірів та boardview.

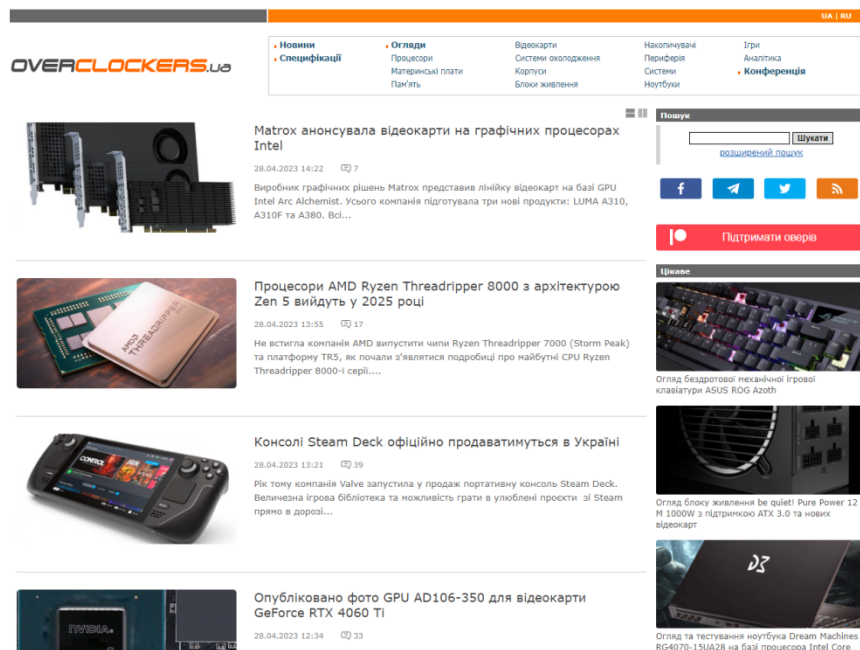


Рисунок 1.5 – Приклад інтерфейсу Overclockers.ua

1.2.3. Linus Tech Tips

Linus Tech Tips – це форум який зв'язаний з однойменним каналом на платформі Youtube. На ньому є підрозділи де люди обговорюють проблеми з комп'ютерним залізом, але структурованої готової інформації щодо ремонту тієї чи іншої відео карти чи іншого заліза немає. Однак у порівнянні з попередніми ресурсами є хоча б підрозділ у вигляді форуму щодо ноутбуків.

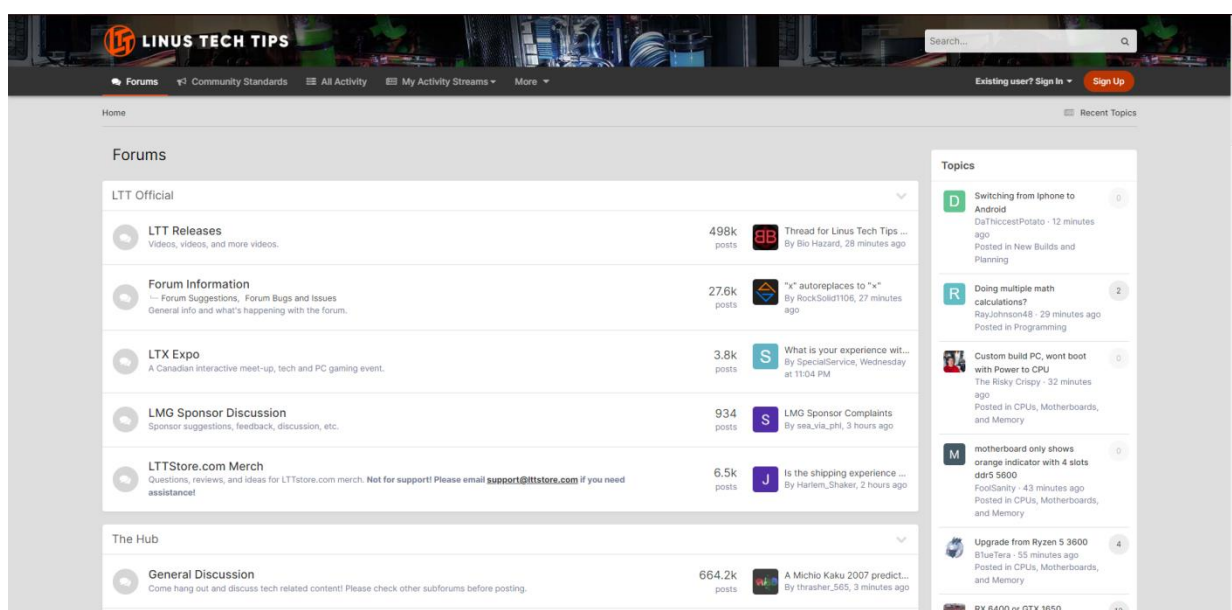


Рисунок 1.6 – Приклад інтерфейсу Linus Tech Tips

Таблиця 1.1 - Результат аналізу порівняння аналогів

	Українська мова	Наявність можливості завантажити біоси	Наявність форумів щодо ремонту комп'ютерного заліза	Наявність підрозділу форуму з ноутбуками
TechPowerUp	-	+	+	-
Overclockers	+	-	+	-
Linus Tech Tips	-	-	+	+

Висновки до першого розділу

У першому розділі було розглянуто тему що таке веб-додаток, проаналізовано актуальність розробки веб-додатку для допомоги сервісним центрам у результаті якого було з'ясовано що такий додаток може бути актуальним та затребуваним та було порівняно аналоги у результаті чого ми бачимо що у більшості з них немає української мови, готових підрозділів з біосами та тільки в одного з них є хоч якась інформація про ноутбуки.

РОЗДІЛ 2. ВИБІР ЗАСОБІВ ТА ТЕХНОЛОГІЙ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

1.1. Середовища розробки

1.1.1. IntelliJ IDEA

IntelliJ IDEA – це інтегроване середовище розробки для різних мов програмування таких як: Java, Python, Scala та інші. Розробляє це програмне забезпечення компанія JetBrains.

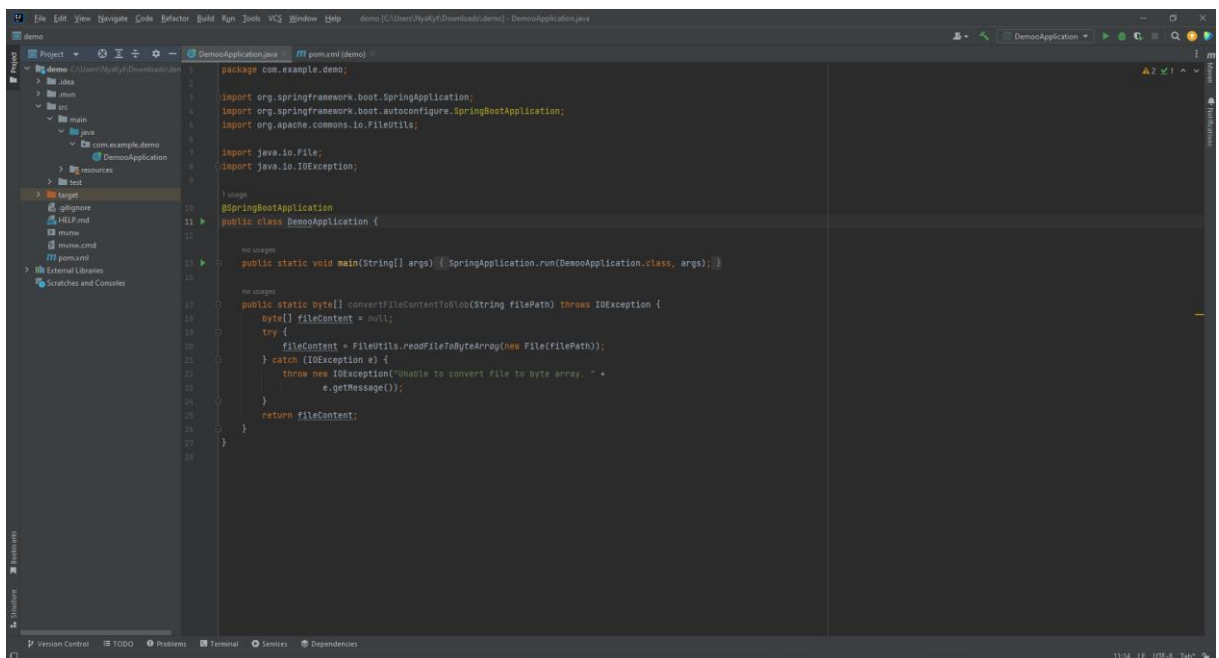


Рисунок 2.1 - Приклад інтерфейсу IntelliJ IDEA

Перша версія програми з'явилася у 2001 році та дуже швидко набрала популярність. На сьогодні останньою версією цього середовища є 2023.1 та воно регулярно оновлюється. Є версії для Linux, Windows та MacOS. Системні вимоги є досить низькими.

Таблиця 2.1 - Системні вимоги IntelliJ IDEA

	Мінімальні	Рекомендовані
ОЗП	2 ГБ вільної пам'яті	8 ГБ наявної в комп'ютері
Процесор	Будь-який сучасний	Багатоядерний процесор. Також IntelliJ IDEA підтримує багато поточність
Місце на жорсткому дисці	3.5 ГБ	SSD накопичувач з вільними 5 ГБ місця
Роздільна здатність монітора	1024x768	1920x1080
Операційна система	<p>64 бітна версія наступних ОС:</p> <ul style="list-style-type: none"> - Microsoft Windows 10 1809 чи новіше - macOS 10.15 чи новіше - Будь-який дистрибутив Linux що підтримує GNOME, KDE чи Unity DE <p>IntelliJ IDEA не підтримується на дистрибутивах Linux, які не включають GLIBC 2.27 чи новіше</p>	Останні 64 бітні версії Windows, macOS чи Linux (наприклад Debian, Ubuntu, RHEL)

Також у IntelliJ IDEA є власний магазин з плагінами.

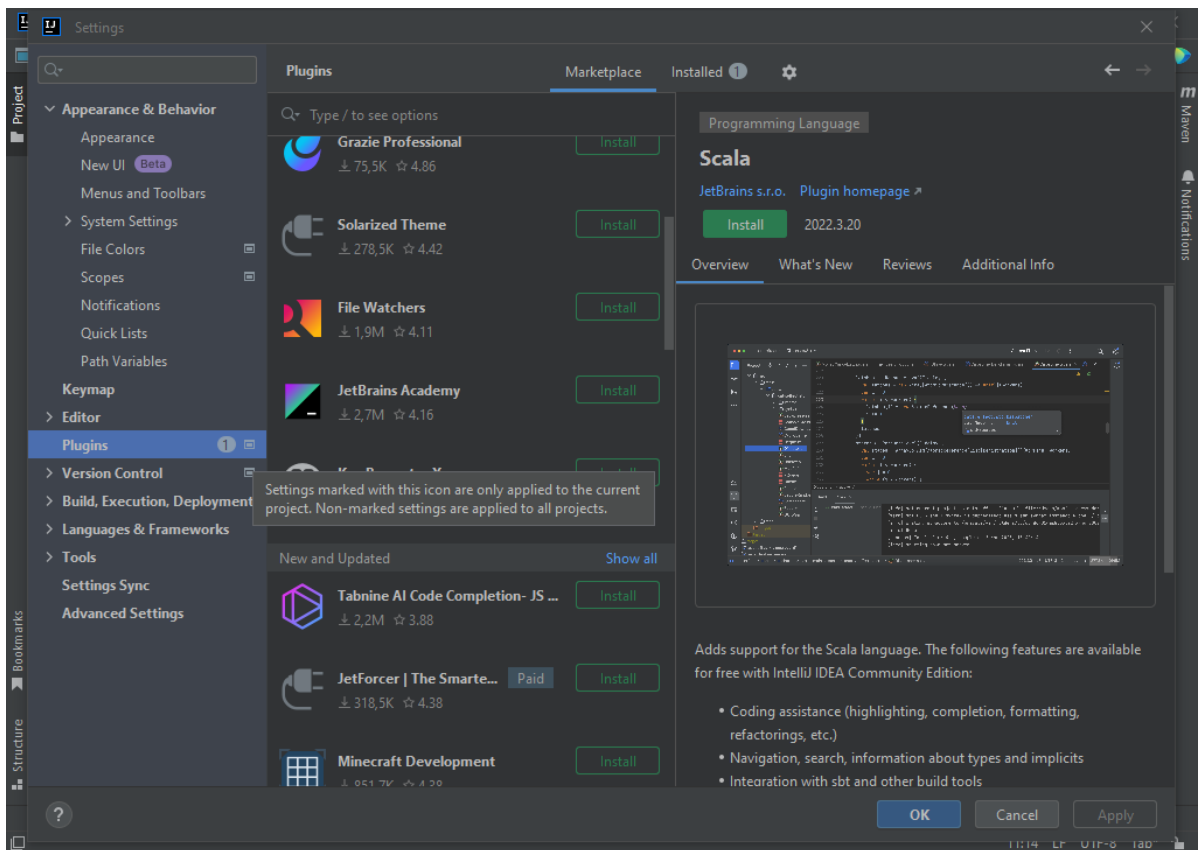


Рисунок 2.2 - Магазин плагінів у IntelliJ IDEA

Постачається IntelliJ IDEA у двох версіях: Ultimate та Community Edition. Порівняння їх наведено у таблиці 2.2.

Таблиця 2.2 - Порівняння версій IntelliJ IDEA

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition
Вартість	599 доларів на рік	Безкоштовно
Java, Kotlin, Groovy, Scala	Підтримується	Підтримується
Maven, Gradle, sbt	Підтримується	Підтримується
Git, GitHub, SVN, Mercurial, Perforce	Підтримується	Підтримується
Відладчик	Підтримується	Підтримується
Docker	Підтримується	Підтримується в межах плагіну

Інструменти профілювання	Підтримується	Не підтримується
HTTP-клієнт	Підтримується	Не підтримується
JavaScript, TypeScript, HTML, CSS, Node.js, Angular, React, Vue.js	Підтримується	Не підтримується
Віддалена розробка (бета)	Підтримується	Не підтримується
Сумісна розробка	Підтримується	Підтримується сеанси до 3 людей, не більше 30 хвилин

1.1.2. Visual Studio Code

Visual Studio Code (скор. VS Code) – це редактор вихідного коду. Він був створений компанією Microsoft сумісно з Electron Framework. Вперше був представлений у 2015 році ті вже у 2022 році при опитуванні розробників був названий найпопулярнішим середовищем розробника. У VS Code є вбудована функція IntelliSense яка виявляє не завершені фрагменти коду, неправильний синтаксис та помилки в коді та підказує як ці помилки вирішити. У VS Code також є вбудований власний магазин розширень.

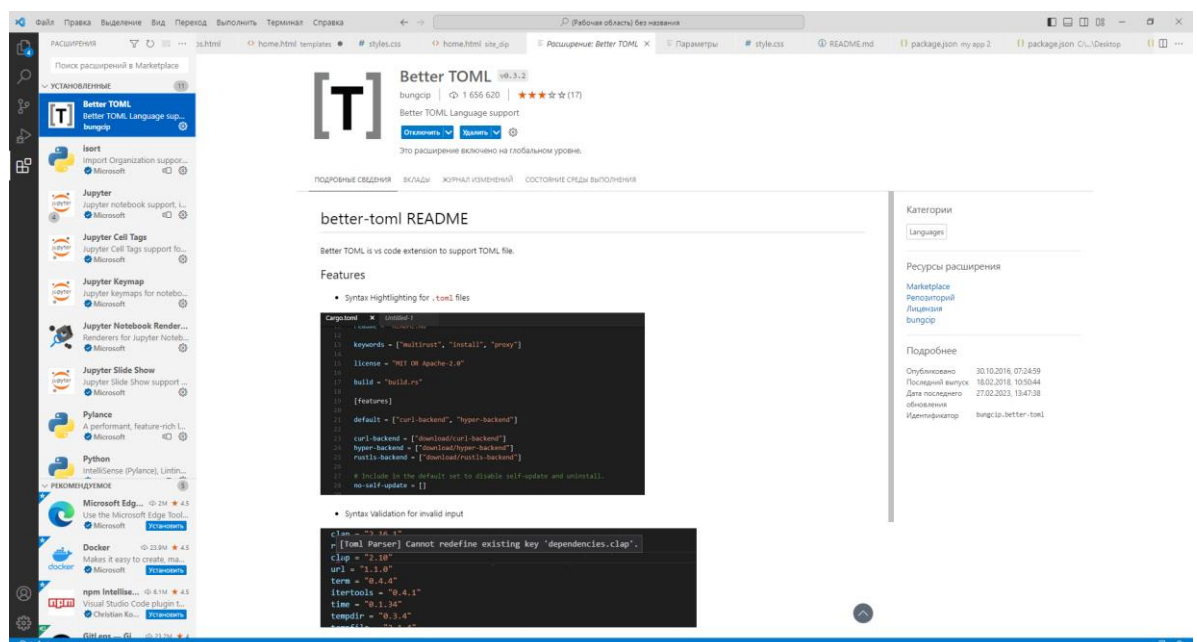


Рисунок 2.3 – Магазин розширень Visual Studio Code

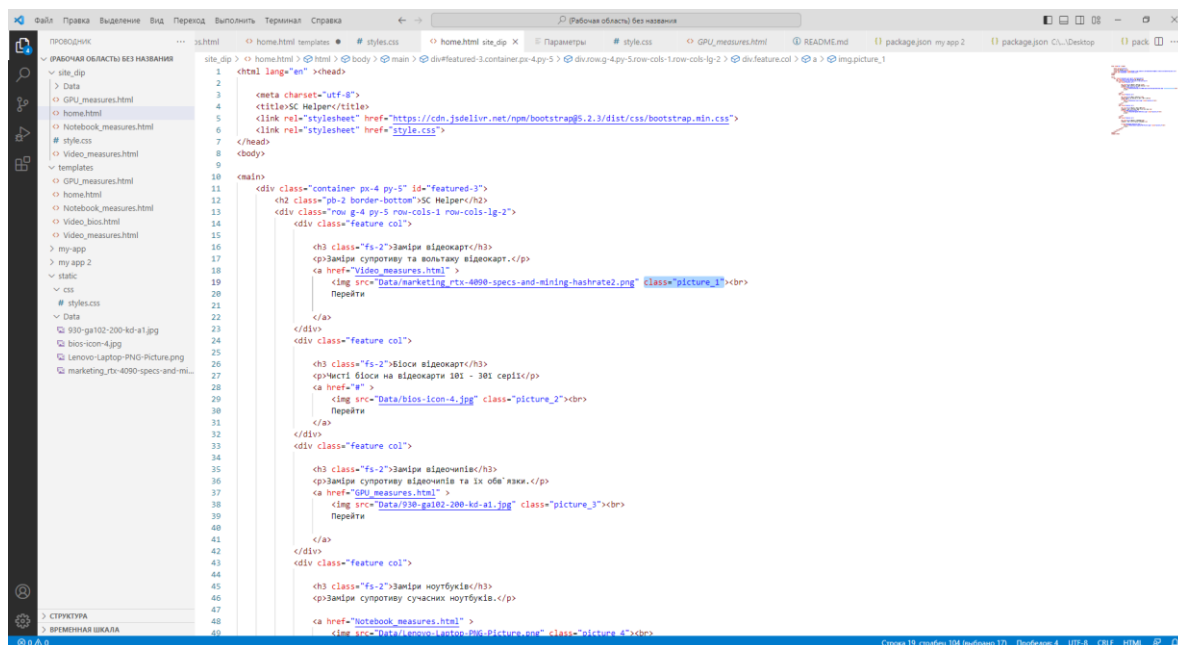


Рисунок 2.4 – Приклад інтерфейсу Visual Studio Code

1.1.3. MySQL Workbench

MySQL Workbench – це інструмент для візуального проектування баз даних. Він в себе інтегрує проектування, моделювання, створення та експлуатацію баз даних. Вперше MySQL Workbench була випущена в 2005 році. У MySQL Workbench є можливість подивитись швидкодію та ефективність бази даних.

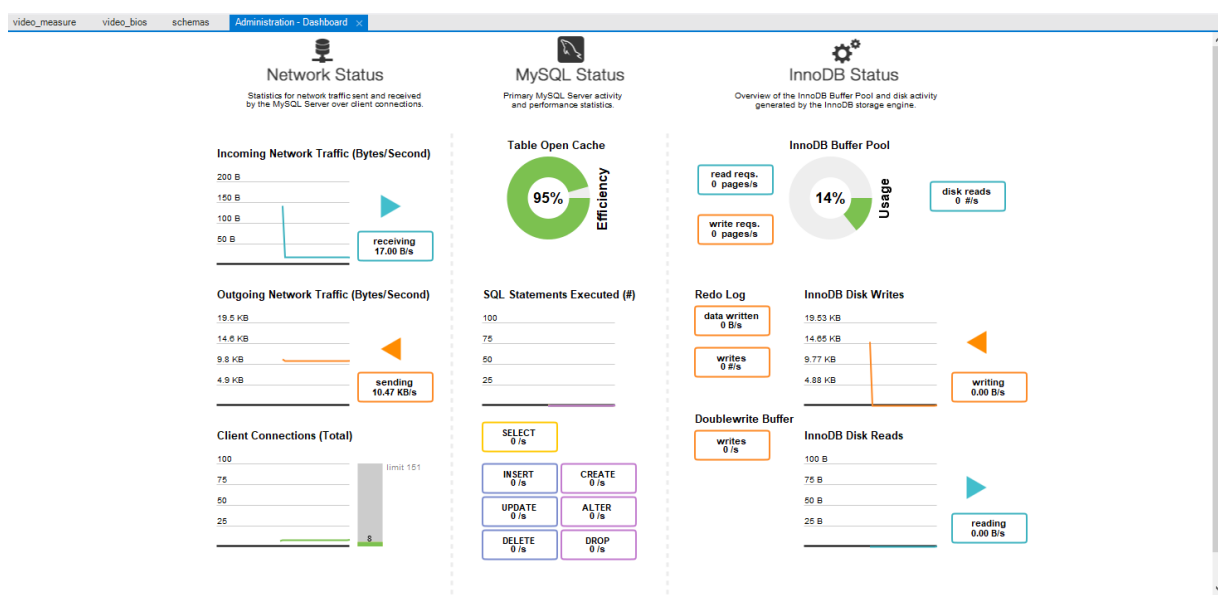


Рисунок 2.5 – Вікно швидкодії у MySQL Workbench

Серед MySQL Workbench має багато функцій та можливостей які роблять роботу з базами даних простіше та ефективніше, ось декілька з них:

- Візуальний конструктор для створення таблиць та моделювання бази даних за допомогою діаграми Entity-relationship
- Панель управління базою даних, яка дозволяє адмініструвати та моніторити базу даних
- Відладчик SQL-запитів, який перевіряє їх та оптимізує їх
- Інструменти для імпорту та експорту даних у різні формати
- Підтримка роботи с декількома серверами та можливість підключення до віддаленої бази даних

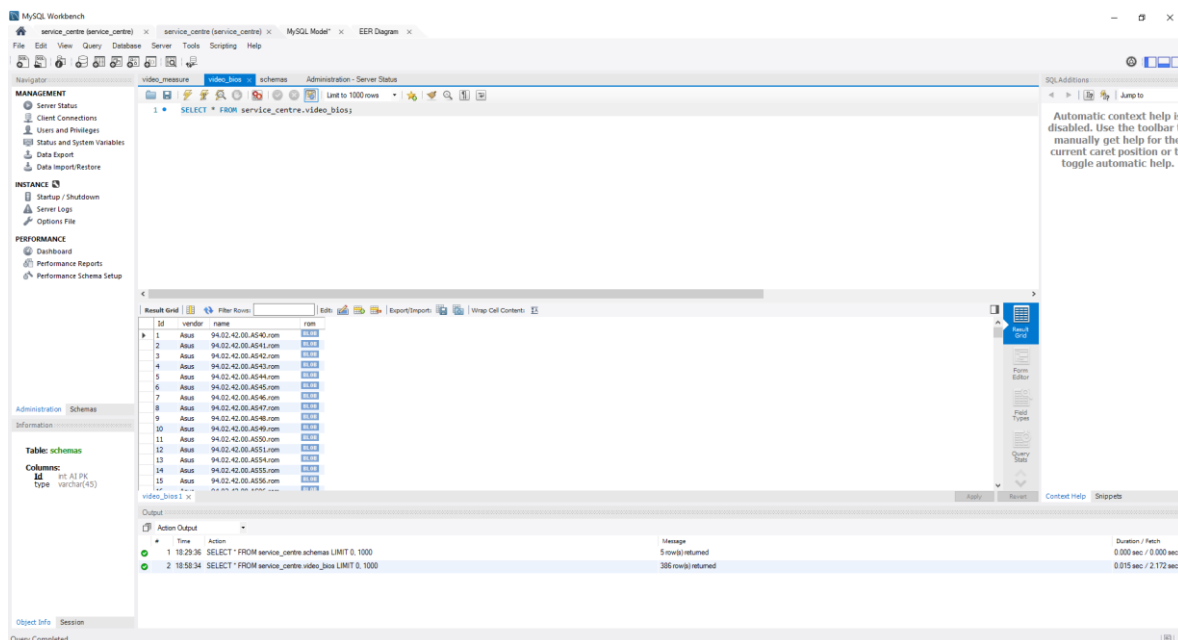


Рисунок 2.6 - Приклад інтерфейсу MySQL Workbench

1.2. Обрані технології розробки

1.2.1. Мова програмування Java

Java – це об’єктно-орієнтована мова програмування. Вперше була випущена ще в 1995 році компанією Sun Microsystems, яку в подальшому викупила компанія Oracle. На сьогодні Java є найпопулярнішою об’єктно-орієнтованою мовою програмування.

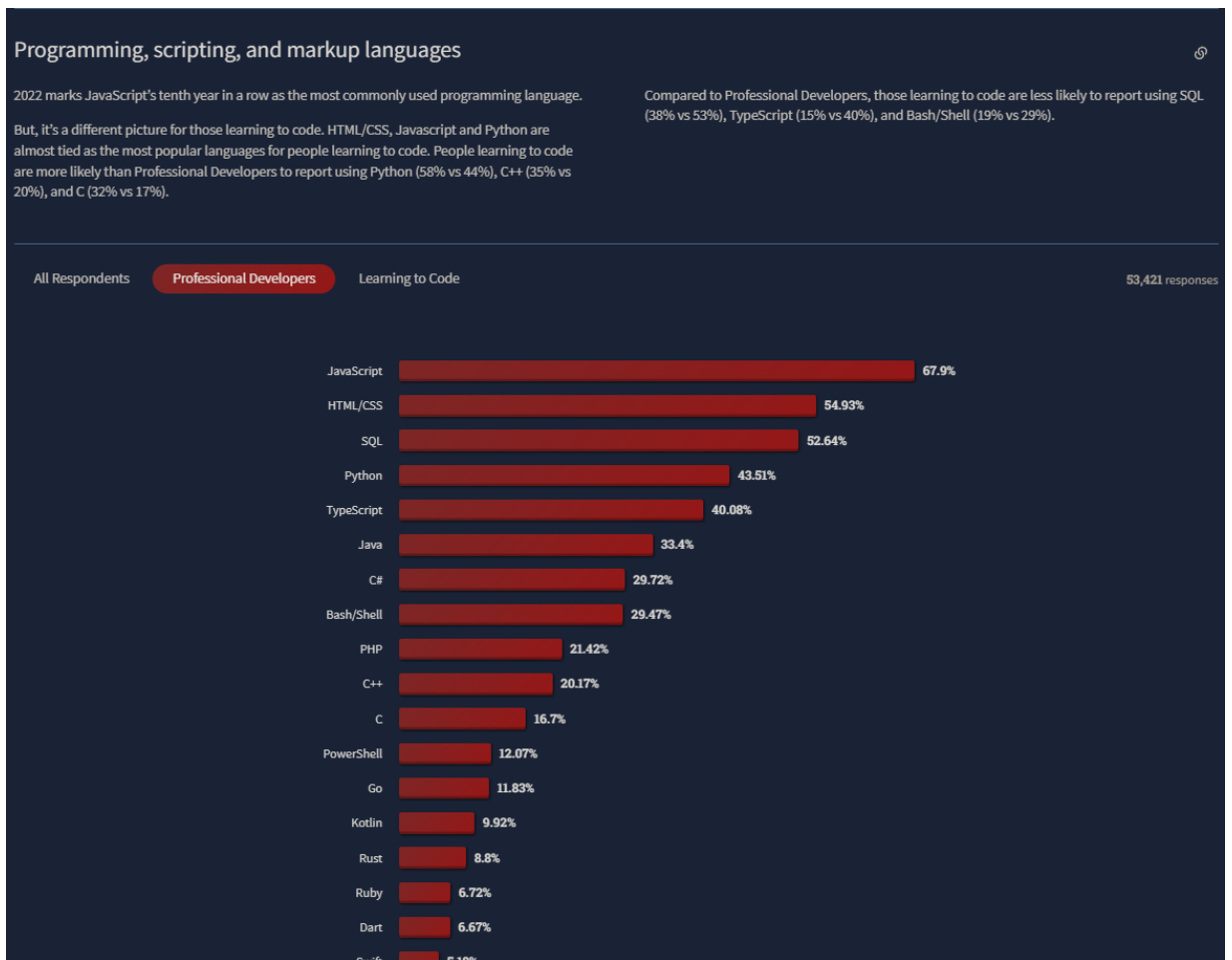


Рисунок 2.7 – Результати опитування «Найпопулярнішої мови програмування, скриптової мови та мови розмітки» на Stack Overflow

Мова Java відрізняється від багатьох мов тим що код написаний на цій мові компілюється в байт-код, тому він може бути виконаний на будь-якій платформі, що підтримує Java Virtual Machine.

Мову Java широко використовують для різних додатків, починаючи від веб-додатків, закінчуючи мобільними. Ось деякі особливості Java які роблять її популярною серед розробників:

- Java була розроблена як мова програмування повністю орієнтована на об'єкти. Це дозволяє розробникам створювати великі та масштабуємі додатки, які легко розширювати.
- Додатки написані на мові Java можуть бути запущенні на будь-якій платформі, що підтримує Java Virtual Machine. Це дає розробникам

можливість створювати додатки, які можуть бути запуснені для декількох платформ, без необхідності переписувати код під кожену.

- Java має вбудовану багато поточність, що дозволяє створювати які виконують декілька задач одночасно, через що програми з великими об'ємами даних та важкими операціями працюють значно скоріше.
- Java має вбудовані механізми безпеки, які захищають додаток від вірусів та інших небезпечних програм. Код Java виконується в ізольованому середовищі від всієї системи.

1.2.2. MySQL

MySQL – це безкоштовна, відкрита система керування реляційними базами даних. Вперше була випущена в 1995 році. Вона була розроблена компанією ТсХ, яку в 2008 році викупила компанія Sun Microsystems, яку поглинула компанія Oracle у 2009 році. Згідно опитування у 2022 році на ресурсі Stack Overflow MySQL є самою популярною системою керування базою даних.

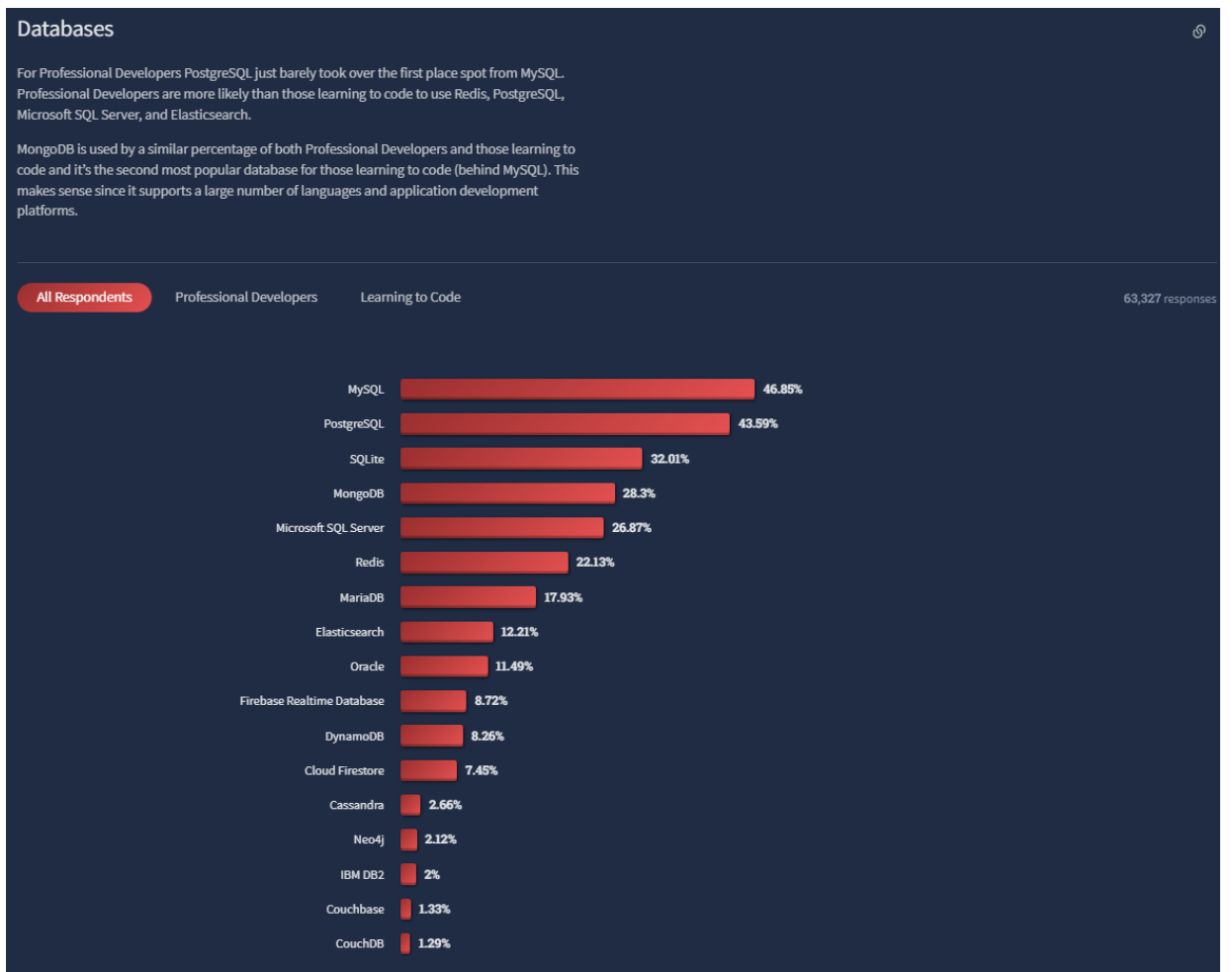


Рисунок 2.8 – Результати опитування «Найпопулярнішої системи керування базою даних» на Stack Overflow

Ось деякі особливості MySQL, які роблять її популярною серед розробників:

- MySQL може працювати з великими об'ємами даних та забезпечувати високу продуктивність.
- MySQL підтримує різні типи даних і має багато функцій, які дозволяють легко керувати даними.
- MySQL має вбудовані механізми, які дають можливість захистити від неакціонованого доступу та злому.
- MySQL може бути розширена за допомогою додаткових модулів та плагінів, які розширюють її функціонал

1.2.3. Spring Boot Framework

Spring Boot – це фреймворк для створення веб-додатків мовою Java, Kotlin та Groovy. Він дозволяє швидко та легко створювати додатки. Він має багато готових компонентів які можливо додати в проект на початку створення на їх веб-сайті та вони будуть автоматично імплементовані в проект, або під час роботи з проектом розробник вручну може їх додати. Одним з найчастіших застосувань цього фреймворку є створення веб-додатків. У 2022 році було проведено опитування, за результатами якого Spring зайняв четверте місце по популярності серед фреймворків та бібліотек.

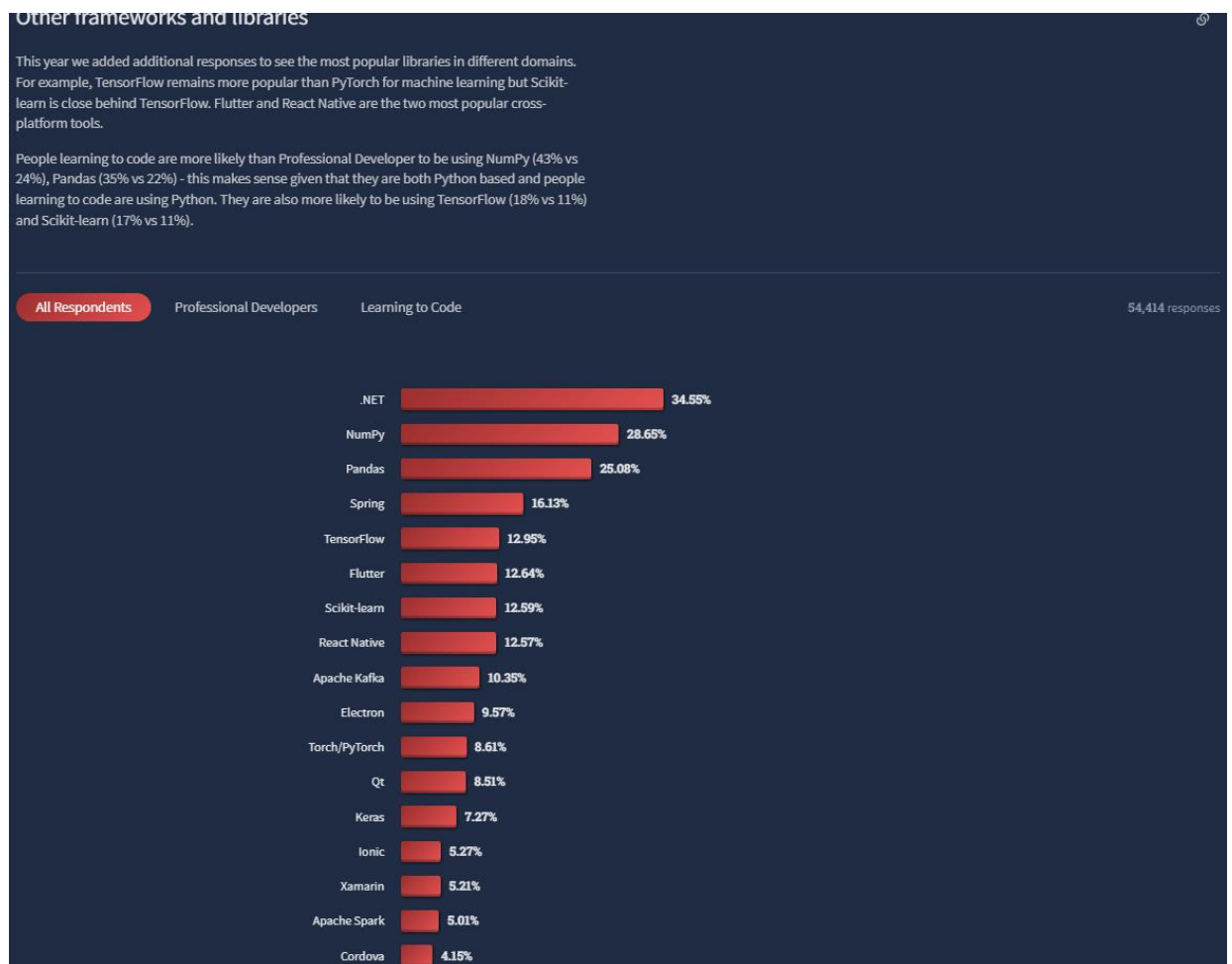


Рисунок 2.9 - Результати опитування «Найпопулярнішої Фреймворки та бібліотеки» на Stack Overflow

Можна виділити деякі особливості фреймворку Spring Boot через які розробники частіше вибирають його:

- Гнучкі бібліотеки, які можна використовувати як для трансляції телепередач, так і для онлайн шопінгу, та безлічі ідей які можна реалізувати за допомогою Spring
- Гнучкий і повний набір розширень і сторонніх бібліотек Spring дозволяє розробникам створювати майже будь-яку програму, яку тільки можна уявити. По суті, функції інверсії керування (IoC) і впровадження залежностей (DI) Spring Framework забезпечують основу для широкого набору функцій і функцій. Незалежно від того, створюєте ви безпечні, реактивні, хмарні мікросервіси для Інтернету або складні потокові потоки даних для підприємства, у Spring є інструменти, які допоможуть.
- Spring Boot змінює ваш підхід до завдань програмування на Java, радикально оптимізуючи ваш досвід. Spring Boot поєднує в собі такі необхідні елементи, як контекст програми та автоматично налаштований вбудований веб-сервер, щоб зробити розробку мікросервісу простою. Щоб працювати ще швидше, ви можете об'єднати Spring Boot із багатим набором допоміжних бібліотек, серверів, шаблонів і шаблонів Spring Cloud, щоб безпечно розгорнути цілі архітектури на основі мікросервісів у хмарі за рекордно короткий час.
- З Spring ви помітите швидкий запуск, швидке завершення роботи та оптимізоване виконання за замовчуванням. Все частіше проекти Spring також підтримують реактивну (неблокуючу) модель програмування для ще більшої ефективності. Вбудовані веб-сервери, автоматичне налаштування та «fat jars» допомагають швидко розпочати роботу, а такі інновації, як LiveReload у Spring DevTools, означають, що розробники можуть виконувати ітерацію швидше, ніж будь-коли

раніше. Ви навіть можете розпочати новий проект Spring за лічені секунди за допомогою Spring Initializr на start.spring.io.

- Комітери Spring працюють із професіоналами з безпеки, щоб виправити та перевірити будь-які зареєстровані вразливості. Залежності від сторонніх розробників також ретельно відстежуються, і регулярні оновлення випускаються, щоб зберегти ваші дані та програми якомога безпечніше. Крім того, Spring Security спрощує інтеграцію з галузевими стандартними схемами безпеки та надає надійні рішення, безпечні за замовчуванням.
- Спільнота Spring є величезною, глобальною, різноманітною та охоплює людей різного віку та здібностей, від повних новачків до досвідчених професіоналів. Незалежно від того, де ви перебуваєте на своєму шляху, ви можете знайти підтримку та ресурси, необхідні для переходу на наступний рівень: швидкі старту, відео, зустрічі, підтримку або навіть формальне навчання та сертифікацію.

1.2.4. Controller-Service-Repository

Controller-Service-Repository – це архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Сутність цього шаблону полягає в поділенні системи на 3 рівні:

- Контроллер – це найвищий рівень і цієї архітектурі, який обробляє всі події та дії користувача та передає їх в Сервіс
- Сервіс –це наступний рівень який відповідає за бізнес-логіку системи та у випадку якщо необхідно прочитати, створити, оновити чи видалити дані, то він передає ці запити в Репозиторій
- Репозиторій, який читає, створює, оновлює та видаляє дані в системі

Схематично цей шаблон виглядає так (Рисунок 2.10)

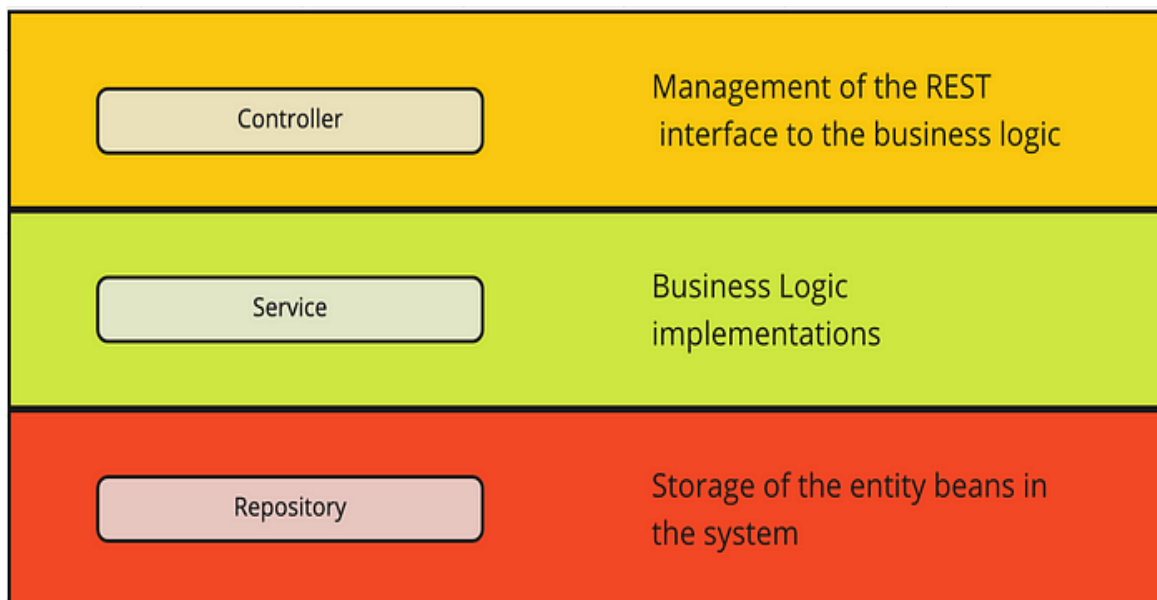


Рисунок 2.10 – Схема рівнів шаблону Controller-Service-Repository
Цей шаблон можна дуже часто спостерігати у Spring Boot додатках.

Висновки до другого розділу

У другому розділі було розглянуті засоби та технології розробки веб-додатку та обрано їх для реалізації поставленої мети.

Для розробки веб-додатку для допомоги було обрано мову програмування Java, яка популярною та гнучкою мовою програмування та підходить для створення веб-додатків. Для написання коду мовою Java було обрано найпопулярніше середовище розробки на мові Java – IntelliJ IDEA. Також був обраний найпопулярніший фреймворк який працює на мові Java – Spring Boot. Шаблоном проектування та розробки був обраний Controller-Service-Repository, який дуже часто зустрічається у Spring Boot додатках. Для написання фронт-енду, тобто HTML та CSS коду була обрана середа Visual Studio Code. Та для зберігання була обрана система керування базою даних MySQL та середа розробки для керування нею MySQL Workbench.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ

3.1. Розробка концепції

Проаналізувавши аналоги та актуальність додатку, було прийнято рішення розробки web-додатку.

Наступним етапом стане проголошення концепції.

Концептуальна модель – це один з початкових кроків для створення проекту. На етапі створення моделі розробник створює й описує свої ідеї в певний документ або таблицю. У результаті має з'явитися документ або таблиця яка описує проект як кінцевий продукт. В подальшому ця модель буде використовуватися членами групи розробки програмного забезпечення та демонструватися інвесторам чи замовникам. Створюваний web-додаток створюється однією людиною, тому в ній можуть бути відображені лише основні особливості веб-додатку. У процесі розробки деякі з них можуть змінюватися.

Таблиця 3.1 Концептуальна модель web-додатку для допомоги сервісним центрам

Категорія	Web-додаток
Цільова аудиторія	Неофіційні сервісні центри
Платформи	Microsoft Windows, Linux, Mac OS, Android
Мінімальні версії браузерів для роботи	Google Chrome версії 63 та вище Mozilla Firefox версії 57 та вище Safari версії 11 та вище Microsoft Edge версії 16 та вище Opera версії 50 та вище
Головні сторінки	Сторінка з замірами супротиву Графічних процесорів

	Сторінка з замірами супротиву ноутбучних материнських плат Сторінка з Біосами для відео карт Сторінка з замірами супротиву та вольтажу відео карт
Мова програмування	Java

3.2. Діаграма використання

Діаграма використання, також відома як use case diagram – це діаграма, що описує функціональність системи з точки зору користувачів. Ця діаграма описує як користувач взаємодіє з системою. Вперше цю діаграму запропонував використовувати А. Джекобсон у 1994 році.

Діаграма використання складається із таких елементів:

- Актор – це хтось або щось, що взаємодіє із системою. Актором не обов’язково повинна бути людина. Актором може бути інша система, підсистема, клас чи будь-що, що взаємодіє з системою, але не належить до неї. На діаграмі він позначається у вигляді чоловічка.
- Випадок використання (Use Case, прецедент) – це подія, що відбувається при взаємодії актора з системою. Зазвичай зображується у вигляді еліпса з текстом всередині.
- Система – це те що моделюється та з чим взаємодіє актор, це може бути як сайт, додаток, програмне забезпечення тощо. Зображується у вигляді прямокутника.
- Зв’язки (відношення) – це зв’язки між актором та прецедентом. Усі актори повинні бути пов’язані з прецедентами, але не усі прецеденти повинні бути пов’язані з акторами.

У діаграмі використання існують такі типи зв’язків:

- Асоціація – це звичайний зв’язок актора та прецедента. Цим типом зв’язку можуть бути зв’язані декілька акторів з одним з одним або декількома прецедентами. Зображується він суцільною лінією без напису.
- Розширення – цей тип зв’язку показує додаткову функціональність прецедента. Він зображується пунктирною лінією з написом <<extend>>
- Включення – цей тип зв’язку показує що поведінка одного прецеденту включається як складовий елемент іншого. Він використовується для уникнення дублювання однакових прецедентів. Зображується він пунктирною лінією з написом <<include>>

Користувачем даної системи буде майстер з ремонту відео карт та ноутбуків.

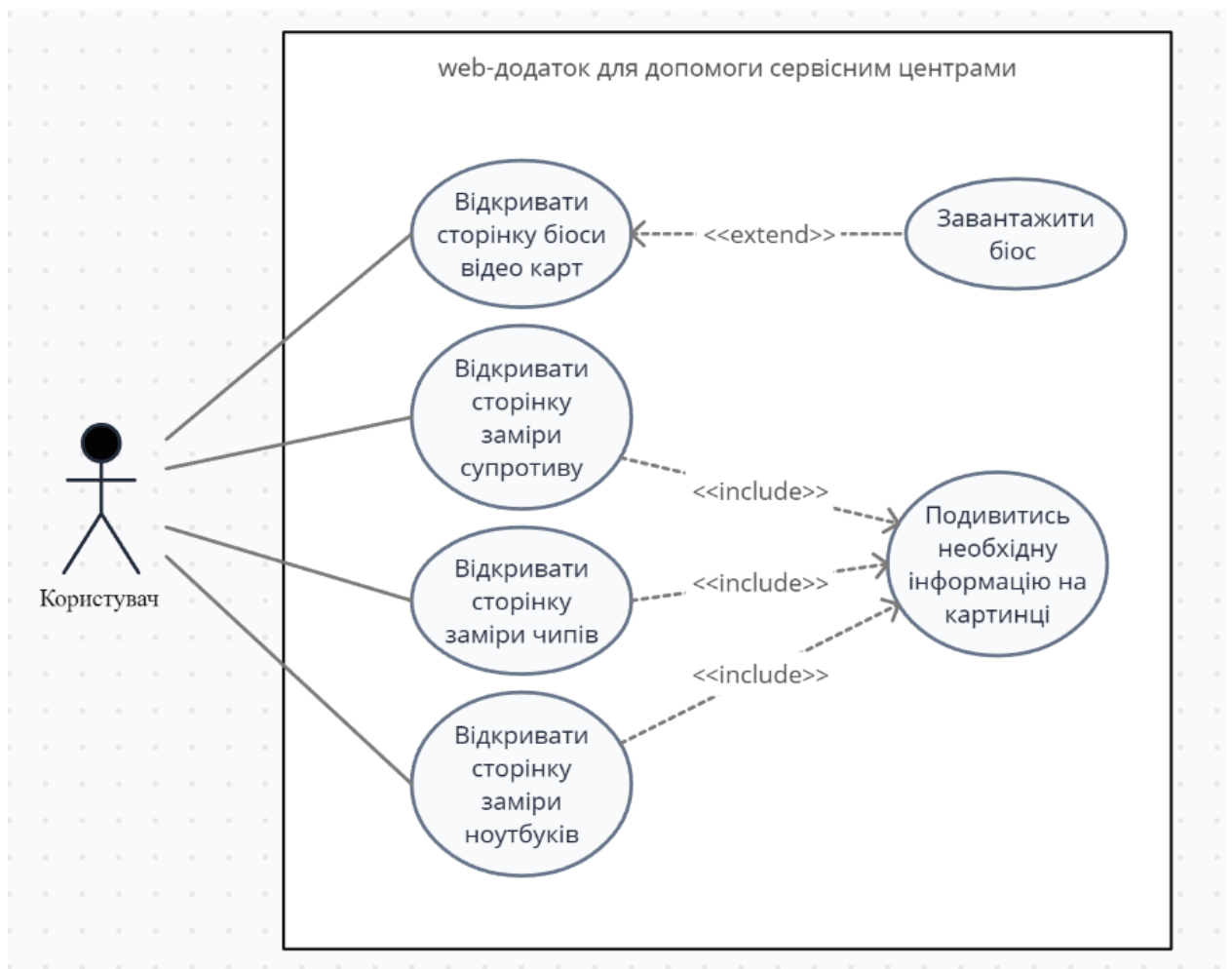


Рисунок 3.1 – UML діаграма використання

3.3. Діаграма діяльності

Діаграма діяльності (Діаграма активності, Activity diagram) – це діаграма, що використовується для послідовності дій та процесів в системі. Вона зображує рішення, дії та послідовність кроків у процесі виконання.

Діаграми діяльності складаються з таких елементів:

- Зафарбований круг – початок діаграми
- Зафарбований круг у колі – кінець діаграми
- Прямокутник з заокругленими кряями – дії що виконуються в процесі
- Ромб – розгалуження вибору
- Товста продовгувата риска з декількома виходами – це вилка, що розділяє потоки на декілька паралельні без прийняття рішення
- Товста продовгувата риска з одним виходом – це злиття, що об'єднують декілька паралельних потоків в один

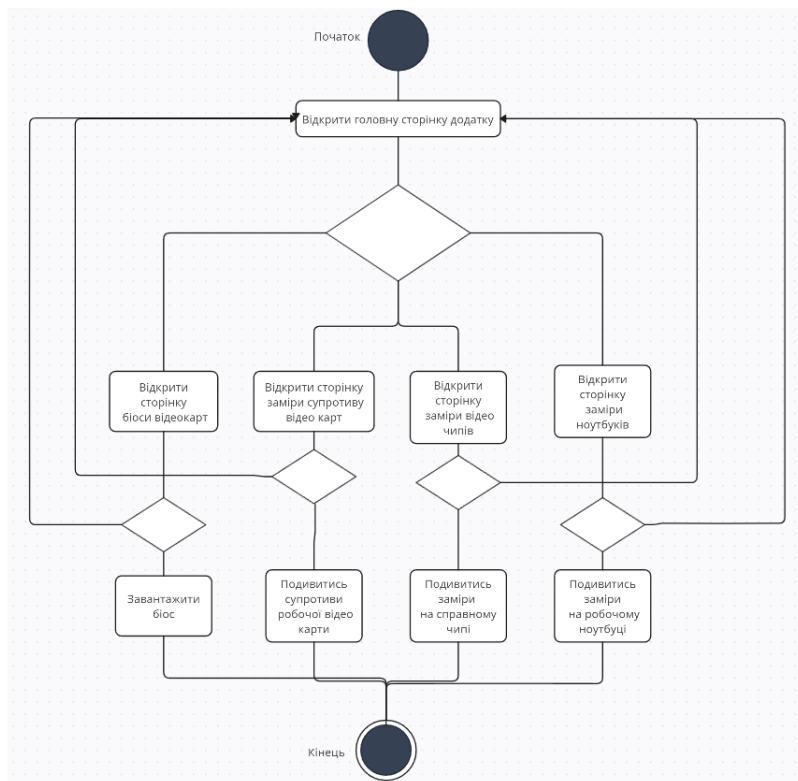


Рисунок 3.2 UML діаграма діяльності

Video_Meas – це клас що обробляє сторінку Video_measures та виводить на неї всі заміри відео карт. Знаходиться він на рівні Сервіс у архітектурі.

Video_M – це інтерфейс класу Video_Meas. Він служить для того щоб передати інформацію з MainController до Video_Meas.

GPU_Meas – це клас що обробляє сторінку GPU_measures та виводить на неї всі заміри графічних чипів. Знаходиться він на рівні Сервіс у архітектурі.

GPU_M – це інтерфейс класу GPU_Meas. Він служить для того щоб передати інформацію з MainController до GPU_Meas.

BiosRepository – це інтерфейс. Він відповідає за збереження та доступ до даних що знаходяться у сутності video_bios.

GPU_measRepository – це інтерфейс, який відповідає за збереження та доступ до даних що знаходяться у сутності gru_measures.

Notebook_measRepository - це інтерфейс, який відповідає за збереження та доступ до даних що знаходяться у сутності notebook_measures.

Video_measRepository - це інтерфейс, який відповідає за збереження та доступ до даних що знаходяться у сутності video_measure.

gru_measures – це клас що відповідає за взаємодію з однойменною таблицею в базі даних. В архітектурі знаходиться на рівні Repository.

notebook_measures – це клас що відповідає за взаємодію з однойменною таблицею в базі даних. В архітектурі знаходиться на рівні Repository.

video_bios – це клас що відповідає за взаємодію з однойменною таблицею в базі даних. В архітектурі знаходиться на рівні Repository.

Video_measure – це клас що відповідає за взаємодію з однойменною таблицею в базі даних. В архітектурі знаходиться на рівні Repository.

3.5. Планування розробки

Для планування задач та розробки було обрано програмне забезпечення Jira Software. Це інструмент для управління проектами та прослідковування задач. Він був розроблений компанією Atlassian. В нього широкий функціонал та багато готових шаблонів для різних напрямків. Ось напрямки для яких існує шаблон в Jira Software:

- Розробка програмного забезпечення
- Управління послугами
- Управління роботою
- Управління продуктом
- Маркетинг
- Робота з кадрами
- Фінанси
- Проектування
- Особистий
- Операції
- Юриспруденція
- Продажі
- Аналітика
- ИТ
- Адміністративно-господарська служба

У Jira Software є великий набір функціональності який надає можливість розробникам та проектним групам ефективно організовувати та керувати своїми проектами. Також у напрямку Розробка програмного забезпечення є готові шаблони для розробки за допомогою методологій Scrum та Kanban. Під час розробки веб-додатку була використана методологія Kanban.

Kanban – це метод управління розробкою, головним принципом якого є вираз «якраз вчасно». При цьому підході весь процес розробки прозорий для всіх членів команди. Задачі по мірі по мірі заносяться в окремий список, звідки потім

члени команди їх беруть. Цей метод вперше з'явився на заводі «Toyota», але в наші дні широко використовується і командами розробників програмного забезпечення. Девід Андерсон вперше ідентифікував п'ять ключових властивостей методу Канбан, пізніше їх назвали практиками та додали ще одну:

- Візуалізація процесів роботи
- Обмеження задач в процесі виконання
- Керування потоком
- Прозоре виконання задач
- Створення циклів зворотнього зв'язку
- Вдосконалення співпрацюючи

В команді з однієї людини основні принципи також працюють. Такі як: візуальне зображення робочого потоку на дошці та обмеження кількості завдань.

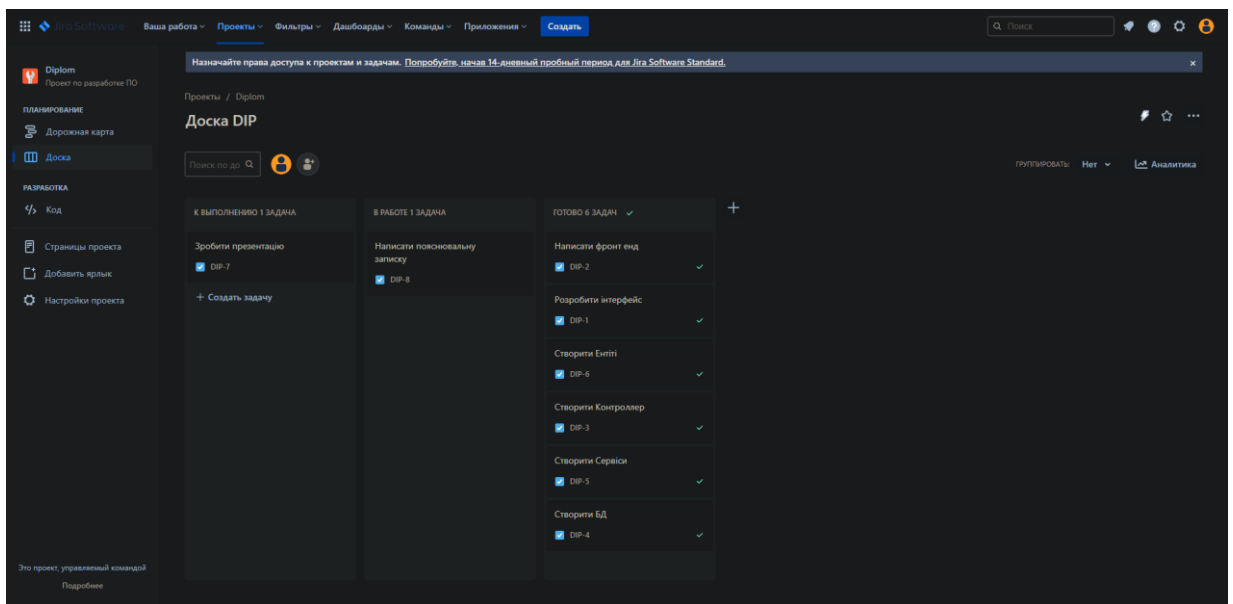


Рисунок 3.4 Kanban дошка проекту

3.6. Тестування web-додатку

Для тестування та перевірки якості розробленого web-додатку було вирішено протестувати наступним чином:

- Зовнішнє тестування (GUI) – це тестування зовнішнього вигляду та функцій, з якими буде взаємодіяти звичайний користувач
- Внутрішнє тестування (Functional) – це тестування всіх функцій web-додатку
- Стійкість – це тестування стійкості додатку до навантаження
- Сумісність – це тестування перевірки сумісності з сучасними браузерами

3.6.1. Зовнішнє тестування

До GUI тестування було включено: наявність орфографічних помилок, робота посилань, масштабованість. Говорячи про перевірку зовнішнього вигляду, то для створення його було використано HTML та CSS разом з шаблоном Thymeleaf.

```
templates > > home.html > @html > @body
1 <html xmlns:th="http://www.thymeleaf.org" lang="en">
2
3 <head>
4
5
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8
9 <title>SC Helper</title>
10
11
12 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
13
14 <link rel="stylesheet" th:href="@{css/styles.css}">
15
16 </head>
17
18 <body>
19
20
21 <main>
22 <div class="container px-4 py-5" id="featured-3">
23 <h2 class="pb-2 border-bottom">SC Helper</h2>
24 <div class="row g-4 py-5 row-cols-1 row-cols-lg-2">
25 <div class="feature col">
26 <h3 class="fs-2">Заміри відеокарт</h3>
27 <p>Заміри супротиву та вольтажу відеокарт.</p>
28 <a href="#video_measures" class="icon-link">
29 <br>
30 Перейти
31 </a>
32 </div>
33 </div>
34 <div class="feature col">
35 <h3 class="fs-2">Біоси відеокарт</h3>
36 <p>Чисті біоси на відеокарти 101 - 301 серії</p>
37 <a href="#video_bios" class="icon-link">
38 <br>
39 Перейти
40 </a>
41 </div>
42 </div>
43 <div class="feature col">
44 <h3 class="fs-2">Заміри відеочипів</h3>
45 <p>Заміри супротиву відеочипів та їх обо'яжж.</p>
46 <a href="#GPU_measures" class="icon-link">
47 <br>
48 Перейти
49 </a>
50 </div>
51 </div>
52 </main>
53 </body>
54 </html>
```

Рисунок 3.5 HTML код головної сторінки

```
49     <a href="#">Перейти
50
51     </a>
52 </div>
53 <div class="feature col">
54
55     <h3 class="fs-2">Заміри ноутбуків</h3>
56     <p>Заміри супротиву сучасних ноутбуків.</p>
57
58     <a href="Notebook_measures" class="icon-link">
59         <br>
60         Перейти
61     </a>
62 </div>
63 </div>
64 </div>
65 </main>
66 </body>
67
68 </html>
```

Рисунок 3.6 HTML код головної сторінки

Шаблон Thumeleaf зручно та швидко передавати дані з серверної частини проекту до користувацького інтерфейсу. У процесі тестування були перевірено інтерфейс на орфографічні помилки які не були знайдені. Також було перевірено додаток на масштабованість

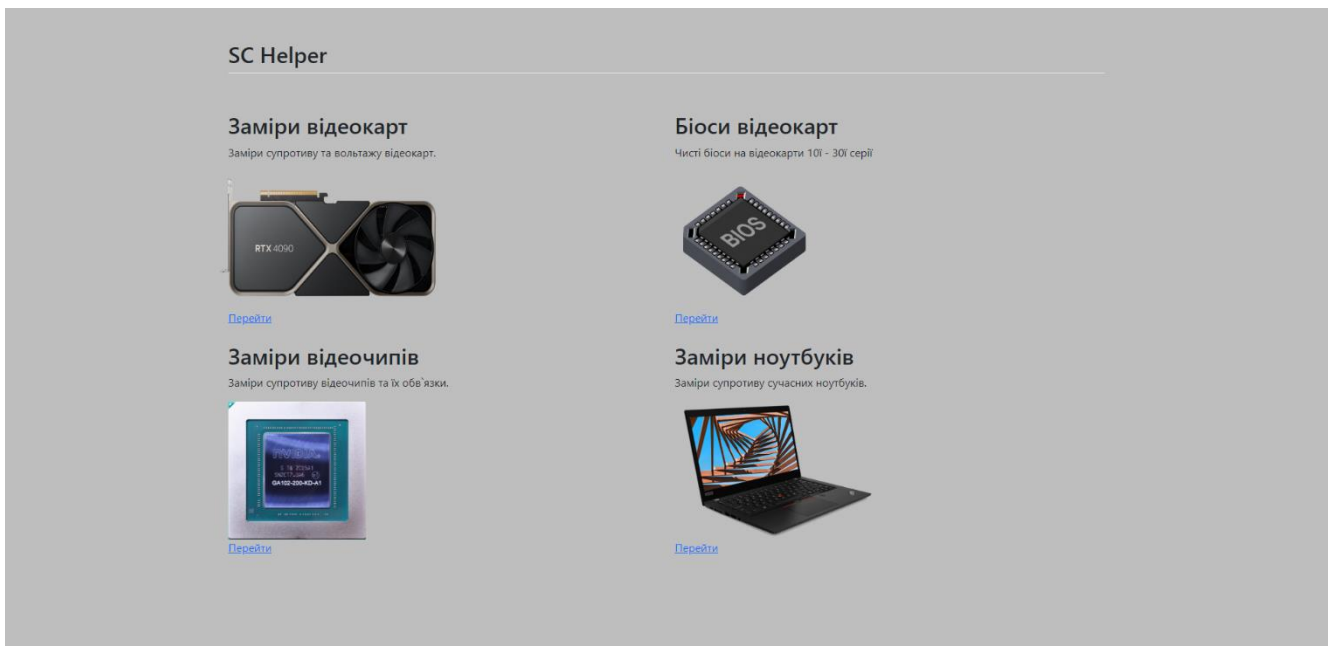


Рисунок 3.7 інтерфейс web-додатку на моніторі зі стандартним співвідношенням сторін 16:9

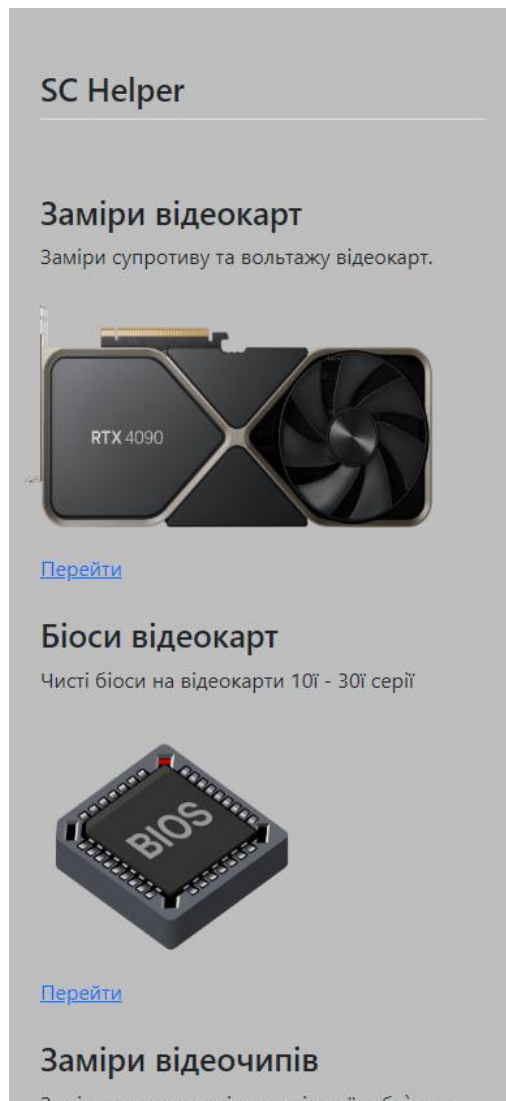


Рисунок 3.8 інтерфейс web-додатку на телефоні

3.6.2. Внутрішнє тестування

Для внутрішнього тестування необхідно створити набір тест кейсів та перевірити їх. Для створення тест кейсів був використаний додаток Qase де були створенні тест кейси та протестовані.

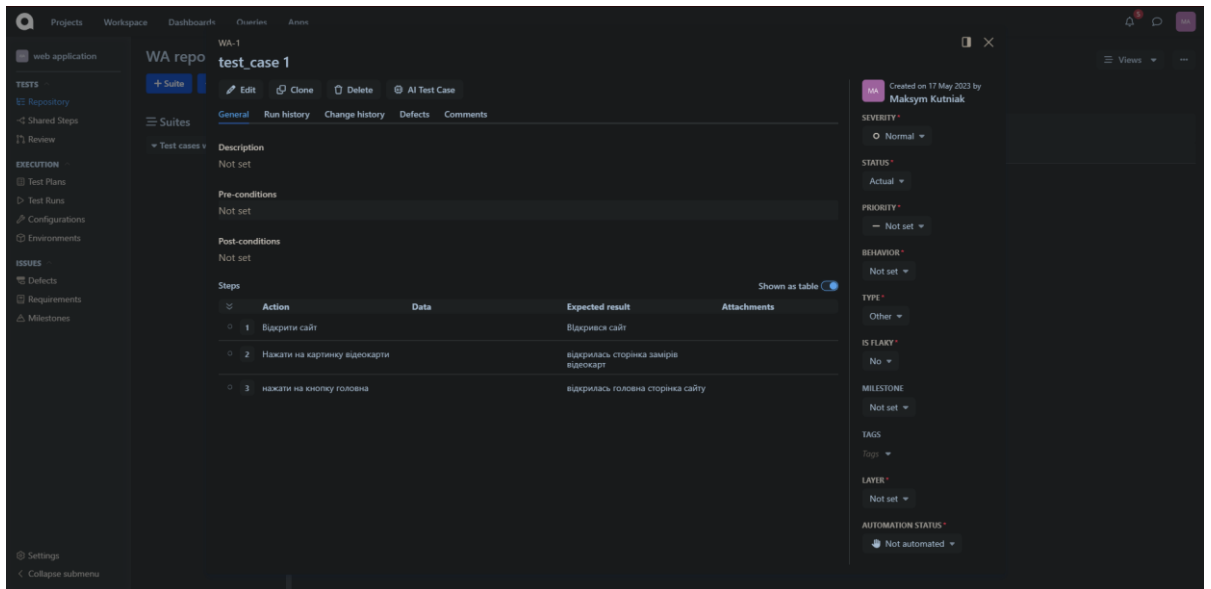


Рисунок 3.9 Приклад тест-кейсу на платформі Qase

3.6.3. Стійкість

Для перевірки web-додаток на стійкість до навантаження було використано додаток Apache JMeter, за допомогою якого можна створити навантаження на додаток.

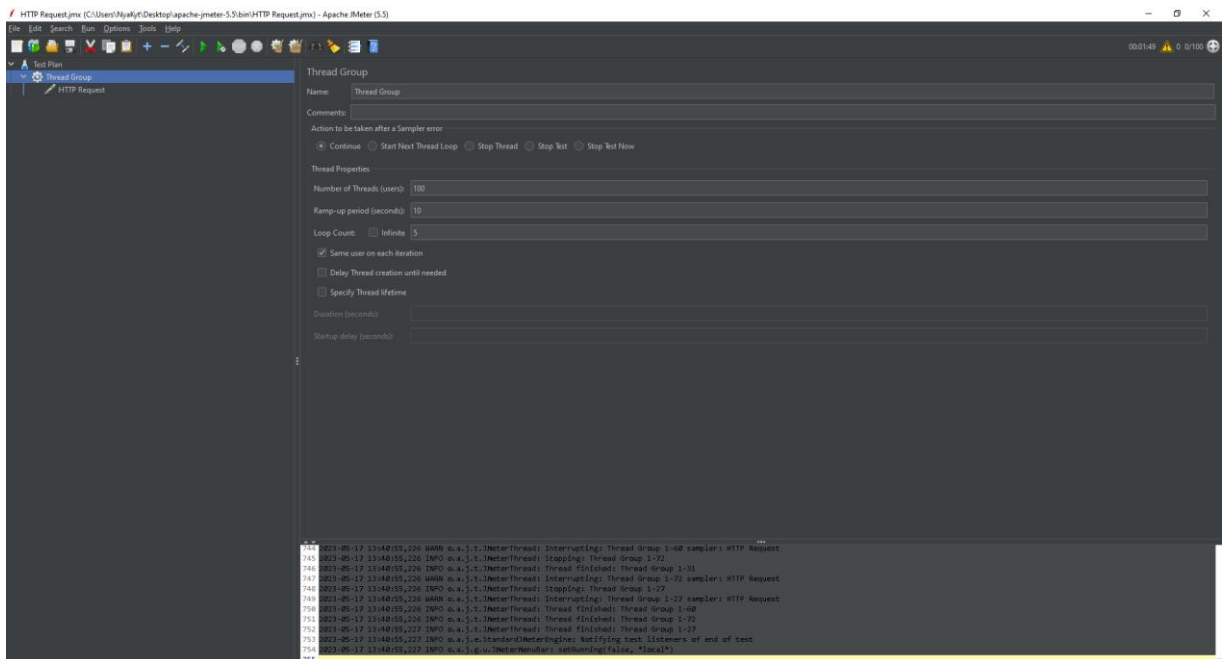


Рисунок 3.10 Приклад інтерфейсу Apache JMeter

3.6.4. Сумісність

Для перевірки web-додатку на сумісність його було відкрито у останніх версіях браузерів: Opera, Google Chrome, Microsoft Edge та Mozilla Firefox та не було помічено проблем.

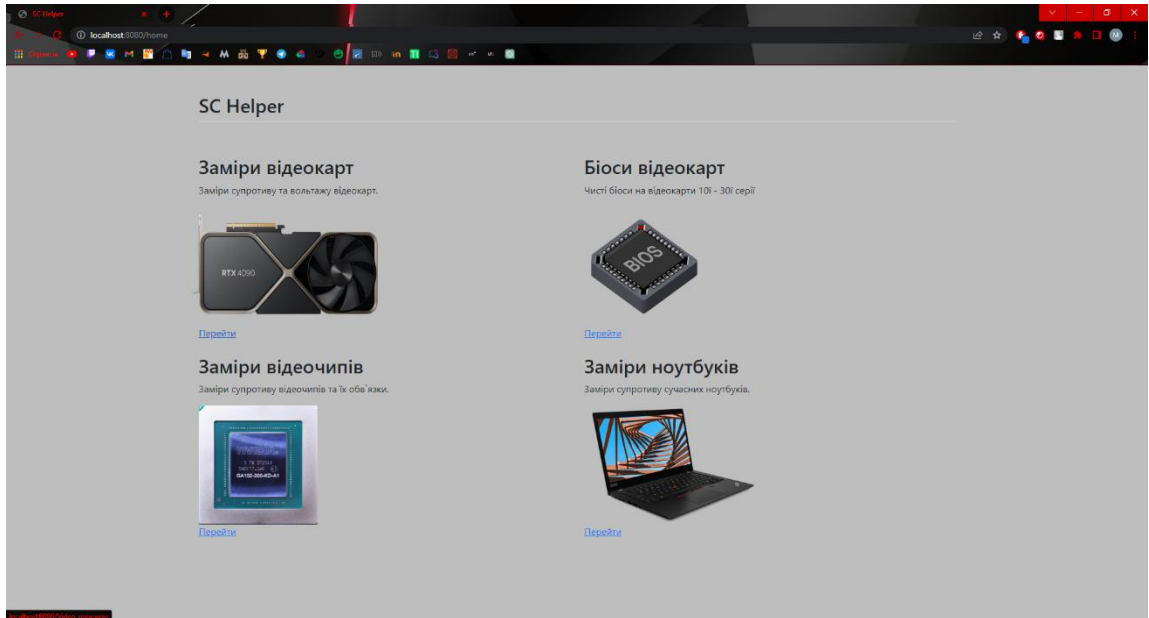


Рисунок 3.11 Приклад роботи додатку у останній версії Google Chrome

ВИСНОВКИ

У результаті виконання даної дипломної роботи було спроектовано та виконано розробку web-додатку для допомоги сервісним центрам.

Проведено аналіз предметної галузі що до існуючих додатків для допомоги сервісним центрам. Ключові проблеми, які можна виділити, пов'язані з недостатньою кількістю інформації на одному ресурсі , відсутність в більшості з них української мови та майже відсутність інформації про ноутбуки. Це призводить до витрачання великої кількості часу на пошук інформації.

Розглянуто програмне забезпечення, яке може бути використано для допомоги сервісним центрам. Ключовими недоліками існуючих систем є недостатня кількість інформації на одному ресурсі , відсутність української мови та майже відсутня інформація про ноутбуки. Таким чином, майбутній програмний продукт повинен включати наступні ключові функції: мати розділ з замірами супротиву та напруг ноутбуків, мати українську мову та повинна мати базу даних з біосами та заміри напруг та супротиву відео карт .

Проведено аналіз засобів розробки програмного забезпечення. обрано мову програмування Java, що забезпечує кросплатформеність, масштабуємість, багато поточність та вбудовані механізми безпеки.