

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА TELEGRAM ЧАТ-БОТУ ДЛЯ ПІДТРИМКИ
НАВЧАЛЬНОГО ПРОЦЕСУ НА КУРСІ "КОМП'ЮТЕРНІ ДИСКРЕТНІ
СТРУКТУРИ" МОВОЮ C#»

Виконав: студент 4 курсу, групи ПД– 44

спеціальності

121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Утешов В.С.

(прізвище та ініціали)

Керівник Садовенко В.С.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

«___» _____ 2023 року

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Утешов Владислав Сергійович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка Telegram чат-боту для підтримки навчального процесу на курсі "Комп'ютерні дискретні структури" мовою C#»

Керівник роботи _____ К. ф.м. наук, доцент Садовенко В.С

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «24» лютого 2023 року №26.

2. Строк подання студентом роботи «1» червня 2023 року

3. Вихідні дані до роботи:

3.1.Офіційна документація C#

3.3.Офіційна документація Microsoft Visual Studio

3.4.Науково-технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1.Аналіз актуальності та огляд існуючих додатків

- 4.2. Аналіз інструментів реалізації та розробка структури чат-боту
- 4.3. Реалізація програмної частини додатку.
- 4.4. Висновки

5. Перелік графічного матеріалу

- 5.1. Титульний слайд
- 5.2. Мета, об'єкт та предмет дослідження
- 5.3. Задачі дипломної роботи
- 5.4. Аналіз аналогів
- 5.5. Вимоги до чат-боту
- 5.6. Програмні засоби для розробки
- 5.7. Екрані форми
- 5.8. Апробація результатів дослідження
- 5.9. Висновки

6. Дата видачі завдання «25» лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Утвердження теми бакалаврської роботи	24.02.2023	Виконано
2	Підбір науково-технічної літератури	26.02.2023 – 15.03.2023	Виконано
3	Аналіз актуальності та дослідження існуючих додатків	15.03.2023 – 20.03.2023	Виконано
4	Аналіз та вибір інструментів для розробки Telegram чат боту	20.03.2023 – 14.04.2023	Виконано
5	Розробка Telegram чат-боту для підтримки навчального процесу	14.04.2023 – 04.05.2023	Виконано
6	Висновки та оформлення роботи	04.05.2023 – 12.05.2023	Виконано
8	Попередній захист	1.06.2023	
9	Захист роботи	16.06.2023	

Студент _____
(підпис)

Утешов В.С
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Садовенко В.С
(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 55 с., 7 табл., 29 рис., 1 дод., 20 джерел.

ТЕХНОЛОГІЯ LTE, OFDM, БАГАТОПОЗИЦІЙНІ СИГНАЛИ, ОПТИМАЛЬНІ МЕТОДИ, КОГЕРЕНТНИЙ, НЕКОГЕРЕНТНИЙ, АВТОКОРЕЛЯЦІЙНИЙ ПРИЙОМ СИГНАЛІВ, ЗАВАДОСТІЙКІСТЬ, ПАРАМЕТРИ.

Об'єкт дослідження – використання Telegram чат-бота для підтримки навчального процесу на курсі "Комп'ютерні дискретні структури".

Предмет дослідження – Telegram чат-бот, розроблений мовою C# для підтримки навчального процесу.

Мета роботи – підтримка навчального процесу, за допомогою Telegram чат боту

Методи дослідження – методи аналізу бізнес-процесів, програмування, системного аналізу, методи тестування та відладки програмного забезпечення.

Визначено основні аспекти та вимоги до чат-боту, а також вибрано технології та інструменти для його розробки.

Здійснено проектування та розробку чат-боту, а також проведено його тестування.

На основі результатів виконаних досліджень розроблено Telegram чат-бота, який підтримує навчальний процес на курсі "Комп'ютерні дискретні структури". Цей бот включає різні можливості, зокрема подання навчальних матеріалів, нагадування про терміни виконання завдань, проведення тестування знань тощо.

Упровадження розробленого Telegram чат-боту дозволяє полегшити процес навчання, підвищуючи ефективність спілкування між студентами та викладачами, та забезпечуючи легкий доступ до навчальних матеріалів.

ЗМІСТ

ВСТУП.....	8
1. АНАЛІЗ ОСНОВНИХ АСПЕКТІВ ПРЕДМЕТНОЇ ОБЛАСТІ «НАВЧАЛЬНИЙ ПРОЦЕС В ЗАКЛАДІ ВИЩОЇ ОСВІТИ (ЗВО)».....	11
1.1 Дослідження основних бізнес процесів предметної області.....	11
1.2 Аналіз програмних рішень для предметної області.....	16
1.3 Постановка задачі на розробку Telegram-бота	20
2. ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ НАВЧАЛЬНОГО ПРОЦЕСУ	22
2.1 Вибір інструментів розробки	22
2.2 Аналіз вимог. Use-case діаграми. Основні прецеденти.....	25
2.3 Архітектура проекту	30
2.3.1 Особливості розробки рівня BLL	31
2.3.2 Особливості розробки рівня UI	33
2.3.3 Особливості розробки DAL	36
3. РОЗРОБКА, ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ НАВЧАЛЬНОГО ПРОЦЕСУ	38
3.1 Проектування бази даних	38
3.2 Розробка програмних модулів системи	39
3.3 Результати функціонального та модульного тестування.....	46
3.4 Інструкція користувачеві програми для підтримки навчального процесу	49
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ	61
ДОДАТОК А	63
ДОДАТОК Б	71

ВСТУП

В епоху інформаційних технологій і цифрової економіки важливою є не лише можливість отримувати великі обсяги інформації, але й вміння оперувати нею: швидко знаходити необхідні дані, вміти їх аналізувати та використовувати. Саме цим і займається дисципліна «Комп'ютерні дискретні структури», вивчення якої є важливим етапом в підготовці майбутніх спеціалістів в галузі комп'ютерних наук.

Актуальність теми полягає в нагальній потребі вдосконалення навчального процесу за допомогою сучасних цифрових технологій. У сучасних умовах, коли усе більше навчальних процесів переходить в онлайн-формат, потреба в технологіях, які б дозволили оптимізувати процес навчання, зростає. Чат-боти в Telegram можуть виконувати роль особистих асистентів для студентів, нагадувати про важливі дати і терміни, розсилати навчальні матеріали та навіть проводити невеликі тести для перевірки знань.

Використання чат-ботів у навчальному процесі було предметом ряду досліджень. Зокрема, в роботах [1] вивчається використання чат-ботів для підтримки навчання та взаємодії студентів. Проте, більшість досліджень фокусуються на використанні чат-ботів як додаткового інструменту навчання, а не як основного механізму підтримки навчального процесу. Використання чат-ботів в рамках конкретних курсів, особливо в таких специфічних як «Комп'ютерні дискретні структури», залишається недостатньо дослідженим.

Інформаційною базою для цієї роботи послужили наукові дослідження в галузі використання чат-ботів у навчальному процесі, матеріали по розробці чат-ботів для Telegram, а також конкретні приклади чат-ботів, що використовуються в навчальному процесі. До основних джерел відносяться роботи [2-3], документація Telegram Bot API та інші технічні ресурси.

Об'єктом дослідження в даній роботі є процес навчання студентів курсу «Комп'ютерні дискретні структури». Це комплексний процес, що включає в себе різні аспекти: передачу знань, самостійну роботу студентів, перевірку знань, та використання різноманітних технологій для підтримки навчального процесу.

Предметом дослідження є використання Telegram чат-бота як інструменту підтримки навчального процесу на курсі «Комп'ютерні дискретні структури». Зокрема, досліджується використання чат-бота для автоматизації надсилання навчальних матеріалів, нагадування про терміни та організації тестування знань.

Метою даного дослідження є розробка та впровадження Telegram чат-бота для підтримки навчального процесу на курсі «Комп'ютерні дискретні структури». Ця мета орієнтована на поліпшення якості процесу навчання та підвищення ефективності взаємодії між студентами та викладачами через використання новітніх технологій.

Для досягнення цієї мети ставляться наступні завдання:

- вивчення потреб користувачів (студентів та викладачів) для розробки вимог до чат-бота;
- розробка архітектури програми, що відповідає визначеним вимогам;
- програмування чат-бота, включаючи реалізацію його основних функцій: відправка навчальних матеріалів, нагадування про терміни, організація тестування знань;
- тестування розробленого чат-бота на користувачах для перевірки його функціональності та виявлення можливих проблем.

Методика цього дослідження базується:

- аналітичний метод. Використовується на етапі аналізу потреб користувачів та вивчення існуючих технологій для розробки чат-ботів. Даний метод дозволяє визначити ключові вимоги до чат-бота та найефективніші технології його реалізації;
- метод проектування та моделювання. При розробці архітектури та дизайну чат-бота використовуються методи проектування систем і методи моделювання. Вони дозволяють відобразити структуру і взаємодію різних компонентів чат-бота.

Наукова новизна дослідження полягає у створенні чат-бота для підтримки навчального процесу на курсі «Комп'ютерні дискретні структури». Це дозволяє перенести частину навчального процесу в цифрове середовище, забезпечуючи

більш зручний доступ до матеріалів курсу і надаючи можливість здійснювати інтерактивне тестування. Хоча розробка чат-ботів для навчання не є новим явищем, саме такий підхід є новим для даного курсу, що дозволяє краще розуміти взаємодію студентів з матеріалом курсу та підвищує ефективність навчання.

Практична значущість результатів дослідження полягає в забезпеченні ефективного інструменту для викладання курсу «Комп'ютерні дискретні структури». Чат-бот дозволяє викладачам надавати студентам швидкий і доступний доступ до матеріалів курсу, а також здійснювати інтерактивне тестування, що допомагає перевірити рівень засвоєння матеріалу. Використання такого чат-бота може значно покращити якість навчання, зробити його більш гнучким і адаптивним до потреб кожного студента.

1. АНАЛІЗ ОСНОВНИХ АСПЕКТІВ ПРЕДМЕТНОЇ ОБЛАСТІ «НАВЧАЛЬНИЙ ПРОЦЕС В ЗАКЛАДІ ВИЩОЇ ОСВІТИ (ЗВО)»

1.1 Дослідження основних бізнес процесів предметної області

Будь-яка система навчання має на меті створити цілісну структуру універсальних знань, умінь, навичок та досвіду самостійної діяльності й особистої відповідальності учнів - ключових компетенцій, що визначають якість освіти у сучасному світі.

В умовах інформатизації та масової комунікації суспільства, підготовка молодого покоління набуває все більшого значення. Розглянемо особливості навчання, які роблять його найбільш адаптованим до життя в інформаційному суспільстві.

У педагогіці розрізняють навчальні вміння і навички на загальні (загальнонавчальні) та спеціальні (предметні) (див. рис. 1.1).

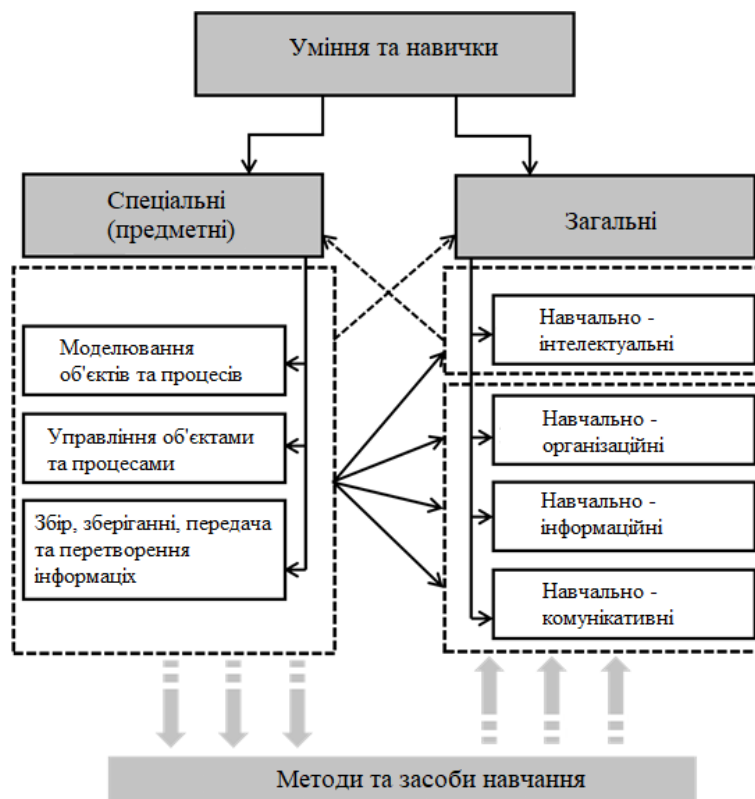


Рисунок 1.1 – Модель процесу навчання

Знання загальнонавчальних умінь і навичок є важливим фактором успіху у навчанні всіх предметів, тоді як спеціальні уміння і навички визначають успішність у конкретних навчальних предметах. Сучасні навчальні заклади приділяють все більше уваги формуванню як загальнонавчальних, так і спеціальних умінь і навичок [1].

Володіння широким спектром умінь і навичок визначається як здатність до навчання. Існує кілька підходів до класифікації загальнонавчальних умінь і навичок. Наприклад, в одному джерелі виділяються такі типи:

- навчально-організаційні уміння і навички;
- навчально-інформаційні уміння і навички;
- навчально-інтелектуальні (навчально-логічні) уміння і навички;
- навчально-комунікативні уміння і навички.

Особлива увага повинна бути приділена розвитку навчально-інтелектуальних умінь та навичок, які включають такі аспекти: уміння свідомо вивчати матеріал і виділяти головне; вміння аналізувати, порівнювати, класифікувати, встановлювати причинно-наслідкові зв'язки та інші; вміння будувати зв'язну мову, відповіді, аргументацію; вміння формулювати висновки і т.д. Формування цих навчально-інтелектуальних умінь і навичок є завданням для всіх учнів з різними рівнями.

Уміння розпізнавати суттєві властивості об'єкта та встановлювати причинно-наслідкові зв'язки можуть бути розвинуті шляхом визначення понять та формулювання законів, теорем; учні можуть представляти великі обсяги текстової інформації за допомогою таблиць і схем; після вивчення кожної теми рекомендується давати учням завдання, пов'язані з класифікацією вивчених об'єктів, тощо.

У той же час, навчально-інтелектуальні уміння, які розвиваються в рамках інформатики на матеріалі з міжпредметним змістом, набувають більшої ваги як загальнонавчальні уміння у свідомості учнів [2].

У контексті навчального матеріалу нового покоління, ми розуміємо поняття навчально-методичних комплектів (НМК), як сукупність матеріалів та обладнання,

що мають цілісну структуру і придатні для організації та проведення навчального процесу в новому освітньому середовищі, що базується на використанні засобів ІКТ. Ці комплекти сприяють досягненню традиційних результатів навчання, таких як знання, уміння та навички, а також розвитку освітніх результатів, які є компетенціями, необхідними для підготовки учнів до життя в інформаційному суспільстві (рис. 1.2).

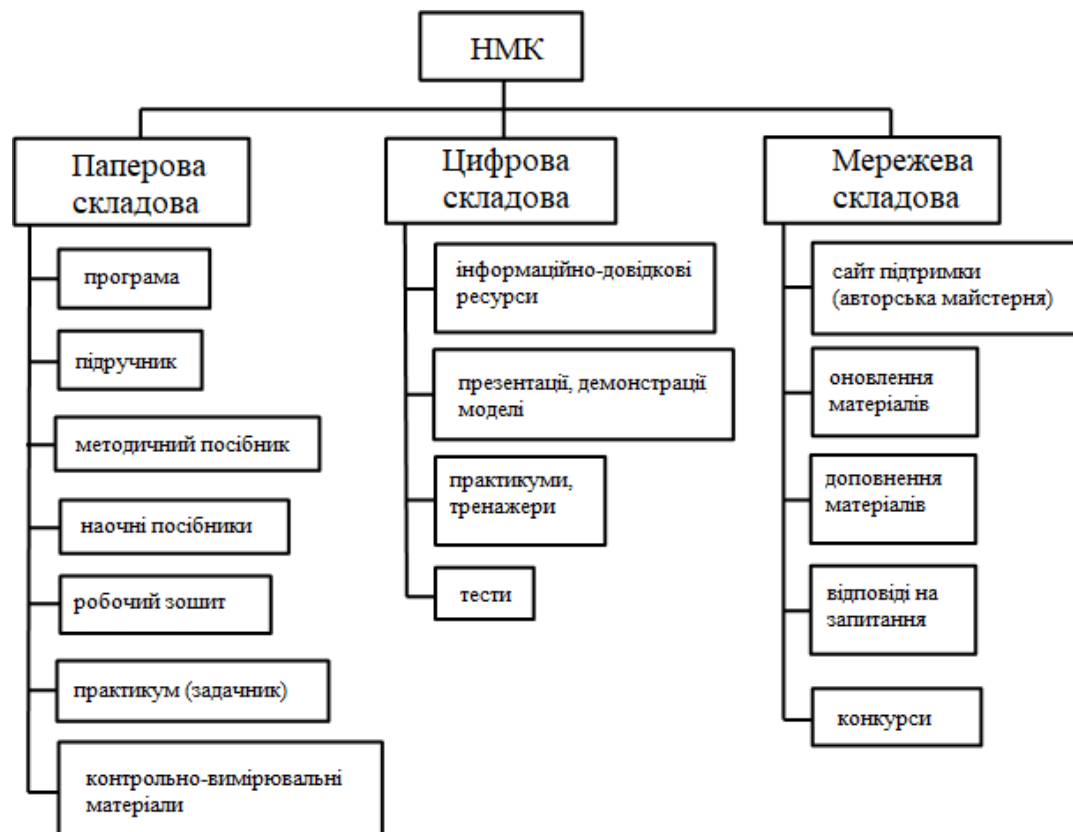


Рисунок 1.2 – Інформаційна складова НМК

У сучасних умовах, ще одним важливим елементом будь-якого навчально-методичного комплексу (НМК) є його мережна складова, яка дозволяє вчителям систематично отримувати консультації від автора, завантажувати оновлені версії планування, нові цифрові освітні ресурси та різноманітні методичні матеріали.

Використання навчальних матеріалів нового покоління сприяє підвищенню ефективності освітнього процесу, що дозволяє підготувати учнів до життя в інформаційному суспільстві шляхом розвитку таких компетенцій, як: здатність до

навчання, комунікабельність, спроможність працювати в колективі, самостійне мислення і діяти, здійснювати вибір та нести відповідальність, розв'язувати нетрадиційні завдання з використанням набутих знань, умінь та навичок. У цьому контексті, важливим завданням є формування широкого спектру компетенцій на основі навчальних матеріалів нового покоління, зокрема: навчальних (організація навчального процесу та вибір власної траєкторії розвитку; розв'язування навчальних та самоосвітніх проблем; поєднання окремих компонентів знання для практичного застосування); дослідницьких (знаходження та обробка інформації, використання різних джерел даних, робота з документами); соціально-особистісних (критичне мислення щодо запропонованих знань, встановлення та інтерпретація зв'язків між навчальними знаннями та реальними явищами, застосування знань у соціальному, політичному, економічному, культурному, екологічному та технологічному контекстах, формування власної думки та участь у дискусіях); комунікативних (уважне сприйняття інших поглядів, аналіз їх підґрунтя, розуміння графіків, діаграм та таблиць даних, публічне виступання з використанням мультимедійних засобів, організація та участь у дискусіях, забезпечення комунікації та співпраці в команді) [3].

У новому поколінні навчально-методичних комплексів доцільно включати різні компоненти, такі як: традиційні навчальні матеріали (підручники, навчальні посібники, робочі зошити, наочні посібники); цифрові освітні ресурси (інформаційні джерела та інструменти); дидактичні та методичні матеріали для організації навчального процесу (формулювання дослідницьких завдань, методичні рекомендації для їх виконання з використанням доступних джерел, інструментів, зовнішнього матеріалу, планування уроків та тестові завдання) у традиційній та цифровій формах; а також цифрове та нецифрове навчальне обладнання.

Таким чином, сучасний навчально-методичний комплект є логічним розвитком традиційного комплекту, збагаченим використанням інформаційних технологій і призначеним для відкритої системи освіти.

Методичні проблеми, які потребують вирішення, включають:

– проблема обсягу матеріалу: В програмі передбачений значний обсяг фактичного матеріалу, але обмежений навчальний час. Викладач повинен забезпечити формування стійких практичних навичок учнів у роботі з інформаційними технологіями, одночасно розвиваючи загальну теоретичну освіченість. У таких умовах це стає викликом;

– проблема організації практичної роботи: Учні мають різні рівні навичок роботи з комп'ютером. Традиційний метод, коли вчителю слід показати і пояснити певні операції, які учні повторюють, може бути малоефективним. Застосування особистісно-орієнтованого та індивідуально-діяльнісного підходів, зміна ролі вчителя, вимагають від вчителя відповідних методичних умінь;

– проблема оцінки знань: Як оцінити опанування учнями інформаційних технологій? Використання методики технологічного навчання, що використовується у професійному навчанні, може бути ефективним рішенням. Але ця методика ще недостатньо використовується в загальноосвітніх закладах.

Мета навчання учнів інформаційних технологій - формування інформаційно-технологічних умінь створення інформаційних продуктів за допомогою комп'ютера. Це можливо досягти за допомогою системи спеціальних навчальних завдань, що моделюють реальні ситуації та застосуванням відповідних підходів до навчання. Формування умінь базується на принципах науковості, систематичності, практичності, самосвідомості учнів, диференціації, індивідуалізації, доступності, міцності, високого рівня складності та наочності.

Для забезпечення ефективності моделі технологічного навчання важливі такі дидактичні умови:

– науково-технологічна спрямованість та міжпредметні зв'язки у навчально-виховному процесі;

– відбір і структурування навчального матеріалу відповідно до конкретної технологічної діяльності;

– систематичне використання інформаційно-комунікаційних технологій (ІКТ).

– практична спрямованість занять та розвиток самостійної роботи учнів.

– управління навчально-пізнавальною діяльністю зі зворотним зв'язком та рефлексією результатів навчання.

Модель методичної системи технологічного навчання має такі завдання:

- формування інформаційно-технологічних знань, вмінь та навичок;
- розвиток навичок роботи з інформацією;
- розвиток інформаційно-технологічної діяльності.

У моделі методичної системи технологічного навчання також важливий формувальний компонент, який включає зміст навчання, методичні підходи, форми організації навчання та засоби навчання. Зміст навчання включає інформаційно-технологічні знання, вміння та навички [4].

1.2 Аналіз програмних рішень для предметної області

Аналіз програмних рішень дозволяє визначити найкращі програмні рішення, що відповідають потребам предметної області та мають потрібні функціональні можливості.

MoodleBot

MoodleBot - це готовий Telegram чат-бот, який інтегрується з платформою навчання Moodle. Він надає зручний доступ до курсового матеріалу, домашніх завдань, тестів та спілкування з викладачем. Бот дозволяє студентам зручно отримувати інформацію про навчальний процес та брати участь у ньому безпосередньо через Telegram.

Переваги MoodleBot включають:

– інтеграція з платформою навчання Moodle. Забезпечує пряму інтеграцію з популярною платформою навчання Moodle. Це означає, що студенти можуть зручно отримувати доступ до матеріалів курсу, завдань, тестів та комунікувати з викладачем, не виходячи з Telegram;

- надання інформації про курс. Може надати студентам докладну інформацію про курс, таку як розклад занять, назви модулів, викладачів, вимоги до оцінювання та багато іншого;

- організація домашніх завдань та тестів. Дозволяє студентам зручно отримувати домашні завдання та проходити тести через Telegram. Викладачі можуть легко створювати та надсилати завдання, а студенти можуть здавати їх та отримувати результати безпосередньо у месенджері.

Недоліки MoodleBot:

- залежність від платформи. Оскільки MoodleBot інтегрується з платформою навчання Moodle, його ефективність та можливості обмежені функціоналом цієї платформи. Якщо Moodle не відповідає конкретним потребам курсу або не використовується в навчальному закладі, це може становити обмеження для використання MoodleBot;

- необхідність конфігурації та налаштування. Для використання MoodleBot необхідно налаштувати інтеграцію з платформою Moodle та налаштувати чат-бота для конкретного курсу. Це може вимагати деякого технічного досвіду та зусиль від адміністратора системи.

- обмеження функціоналу. Хоча MoodleBot надає широкий спектр функціональності, його можливості можуть бути обмежені стандартними можливостями платформи Moodle [5].

EduBots

EduBots - це платформа для розробки навчальних чат-ботів, яка може бути використана для створення Telegram чат-боту для підтримки курсу «Комп'ютерні дискретні структури». Ця платформа надає готові шаблони та інструменти для створення чат-бота, який може надавати інформацію про курс, відповідати на запитання студентів, проводити тести та надавати навчальний матеріал.

Переваги EduBots:

- адаптований до освітніх потреб. Розроблені спеціально для освітніх цілей, і вони мають функціонал, що сприяє підтримці навчального процесу;

- освітній контент. Має вбудований освітній контент, який може бути доступним для студентів. Це може включати навчальні матеріали, рекомендації, посилання на корисні ресурси та інше;

- інтерактивність та залучення студентів. Може надавати інтерактивні функції, які сприяють залученню студентів до навчального процесу.

Недоліки EduBots:

- обмежений функціонал. Має обмежені можливості порівняно з іншими програмними рішеннями. Вони можуть не мати всіх необхідних функцій або бути обмеженими в плані інтеграції з іншими платформами або інструментами;

- вимоги до технічного обладнання. Використання EduBots може вимагати наявності певного технічного обладнання, такого як комп'ютери, планшети або смартфони, для доступу до програмного забезпечення та можливості його повноцінного використання;

- потреба у навчанні та підтримці. Використання EduBots може вимагати навчання викладачів та студентів, які повинні ознайомитися з функціоналом та можливостями цих програмних рішень [6].

Botpress

Botpress - це відкрите програмне забезпечення для розробки чат-ботів. Воно дозволяє створити Telegram чат-бота зі спеціалізованим функціоналом для навчання. За допомогою Botpress можна налаштувати бота для надання навчального матеріалу, проведення віртуальних лекцій, відповідей на запитання та навіть оцінювання студентів.

Переваги Botpress:

- відкрите джерело. Botpress є програмним забезпеченням з відкритим вихідним кодом, що означає, що його можна змінювати та налаштовувати відповідно до власних потреб;

- легка налагодженість та розробка. Botpress надає розширений набір інструментів для розробки та налагодження чат-ботів;

– розширені можливості: Botpress має розширені можливості для створення складних чат-ботів. Він підтримує інтеграцію зі сторонніми сервісами, такими як бази даних, CRM-системи, електронна пошта тощо.

Недоліки Botpress:

– вимоги до технічного обладнання. Використання Botpress вимагає певного рівня технічних знань та наявності певного обладнання, такого як сервери або веб-хостинг;

– навчання та оновлення. Використання Botpress вимагає вивчення його функціоналу та можливостей;

– обмеженість готових модулів. Хоча Botpress надає багато інструментів для розробки, він має обмеженість у готових модулях та компонентах, які можуть бути доступні для безпосереднього використання [7].

У табл. 1.1 проведено порівняльний аналіз розглянутих програмних рішень.

Таблиця 1.1 – Порівняльний аналіз програмних рішень

Основні характеристики	MoodleBot	EduBots	Botpress
Відкрите джерело	Так	Ні	Так
Легкість розробки	Середня	Висока	Висока
Розширені можливості	Середня	Висока	Висока
Технічні вимоги	Низькі	Середні	Середні
Готові модулі	Так	Так	Ні
Навчання та оновлення	Середні	Високі	Середні

Проведений аналіз показав, що MoodleBot є відкритим джерелом з легким розробленням та низькими технічними вимогами, а також має готові модулі. Однак, він обмежений у розширених можливостях. EduBots, з іншого боку, має високу легкість розробки та розширені можливості, але не є відкритим джерелом і має середні технічні вимоги. Botpress, у свою чергу, є відкритим джерелом з

високою легкістю розробки та розширеними можливостями, але має середні технічні вимоги та не має готових модулів.

На підставі проведеного аналізу можна зробити висновок, що немає одного ідеального рішення, яке б виконувало всі потреби підтримки навчального процесу на курсі «Комп'ютерні дискретні структури». Кожен з розглянутих варіантів має свої переваги та недоліки. Отже, виникає необхідність у розробці нового програмного рішення.

1.3 Постановка задачі на розробку Telegram-бота

Для підтримки навчального процесу на курсі «Комп'ютерні дискретні структури» пропонується розробити Telegram-бота, який повинен мати ряд важливих функцій та можливостей:

- нагадування про терміни. Бот повинен надсилати користувачам нагадування про майбутні лекції, дедлайни домашніх завдань та інші важливі дати та події. Необхідно розробити алгоритм, який буде автоматично відправляти такі нагадування за попередньо заданим графіком;

- розсилка навчальних матеріалів. Чат-бот повинен мати можливість розсилки матеріалів для навчання, включаючи лекції, записи, слайди, та інші навчальні ресурси. Система повинна підтримувати різні формати файлів та дозволяти викладачам додавати нові матеріали за потреби;

- тестування та відповіді на запитання. Чат-бот повинен мати інтегровану систему тестування, яка дозволить студентам перевіряти свої знання та отримувати зворотний зв'язок. Також бот повинен мати можливість відповідати на запитання студентів або перенаправляти їх до викладача;

- перегляд довідкових матеріалів. Бот повинен надавати користувачам доступ до важливих довідкових матеріалів, що можуть бути корисні в процесі навчання.

- керування обліковими записами користувачів. Чат-бот повинен мати здатність керувати обліковими записами користувачів, включаючи реєстрацію

нових користувачів, збереження їхніх результатів тестування, відстеження їхнього прогресу в навчанні;

– налаштування нагадувань. Бот повинен надавати можливість користувачам налаштовувати власні нагадування щодо подій, дедлайнів та інших важливих моментів.

Таким чином, основними задачами розробки є створення відповідного інтерфейсу для виконання зазначених функцій, інтеграція бота з відповідними даними курсу, розробка алгоритмів для роботи з навчальними матеріалами та користувачами.

2. ВИБІР ТЕХНОЛОГІЙ ТА ОСОБЛИВОСТІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ НАВЧАЛЬНОГО ПРОЦЕСУ

2.1 Вибір інструментів розробки

Мова програмування

Враховуючи необхідність досягнення високої продуктивності, масштабованості та сумісності з різними платформами, можна розглядати наступні мови програмування: C#, C++ та Java.

Мова програмування C# представляє собою об'єктно-орієнтовану мову програмування, розроблену корпорацією Microsoft. Ця мова була створена з метою розробки програм для платформи .NET, що забезпечує можливість виконання програм на різних операційних системах. C# має синтаксис, який схожий на мову програмування C++, але відрізняється вищим рівнем абстракції, що дозволяє зменшити кількість коду, необхідного для вирішення поставлених завдань. Крім того, C# підтримує різні парадигми програмування, такі як процедурне програмування, функціональне програмування та інші [8].

Мова програмування C++ є універсальною мовою програмування з високим рівнем абстракції. Вона була розроблена як розширення мови C, щоб надати можливості об'єктно-орієнтованого програмування та інші функціональні можливості. C++ знаходить застосування у розробці різноманітних програм, включаючи операційні системи, відеоігри, банківські системи та інші застосування. Мова має широку підтримку та багатий набір функцій, що дозволяє розробникам ефективно використовувати пам'ять та системні ресурси [9].

Мова програмування Java є об'єктно-орієнтованою мовою програмування, що була розроблена компанією Sun Microsystems [10]. Однією з визначальних особливостей Java є високий рівень переносимості, оскільки програми, написані на цій мові, можуть бути виконані на різних операційних системах та пристроях. Java широко використовується для розробки різноманітних програм, включаючи веб-

додатки, мобільні додатки, сервіси хмарних технологій та інші. Особлива увага була приділена вбудованій підтримці багатопоточності, мережових додатків та інших функцій в мові Java, що дозволяє розробникам ефективно використовувати ресурси комп'ютера [11].

У табл. 2.1 наведено табличний порівняльний аналіз функціональних можливостей кожної з розглянутих мов програмування.

Таблиця 2.1 – Порівняльний аналіз мов програмування

Характеристика	C#	Java	C++
Парадигми програмування	Об'єктно-орієнтоване, процедурне, функціональне	Об'єктно-орієнтоване, процедурне, функціональне	Об'єктно-орієнтоване, процедурне, функціональне
Переносимість	Висока, завдяки платформі .NET	Низька, оскільки код має компілюватись для кожної операційної системи окремо	Висока, завдяки вбудованій віртуальній машині Java
Використання пам'яті	Автоматична збірка сміття	Ручне керування пам'яттю	Автоматична збірка сміття
Швидкодія	Середня	Висока	Середня

Виходячи з проведеного аналізу мова програмування C# є хорошим вибором для розробки різноманітних програм з високим рівнем складності та вимог до безпеки. Вибір мови програмування C# бути обґрунтований через платформу .NET, підтримку об'єктно-орієнтованого підходу, вбудовану підтримку мультипоточності, можливість розробки веб-додатків з підтримкою мультипоточності та відкритий код та підтримку спільноти розробників.

Середовище розробки

Середовище розробки дозволяє розробникам створювати, відлагоджувати та тестувати програми у зручному та ефективному середовищі.

У випадку розробки Telegram чат-боту для підтримки навчального процесу на курсі «Комп'ютерні дискретні структури» можна розглянути наступні середовища розробки:

- Visual Studio. Це інтегроване середовище розробки, розроблене компанією Microsoft, яке надає підтримку для розробки програм на різних мовах програмування, включаючи C#, C++, Java, Python та інші [12].

- Eclipse. IDE, яке розроблене фондом Eclipse, яке зосереджено на розробці програм на мові програмування Java та інших мовах. Воно має різноманітні інструменти для розробки, налагодження та тестування програм [13].

- PyCharm. IDE, яке розроблене компанією JetBrains, яке спеціалізується на розробці програм на мові програмування Python. Воно також надає різноманітні інструменти для розробки, налагодження та тестування програм, написаних на мові Python [14-15].

Таблиця 2.2 – Порівняльний аналіз середовищ розробки

Характеристика	Visual Studio	Eclipse	PyCharm
Мови програмування	C#, C++, Java	Java	Python
Платформи	Windows	Cross-OS	Cross-OS
Підтримка Git	Так	Так	Так
Вбудований дебагер	Так	Так	Так
Редагування коду	Так	Так	Так
Аналіз коду	Так	Так	Так
Підтримка тестування	Так	Так	Так
Підтримка мультиплатформи	Ні	Так	Так
Візуалізація даних	Так	Ні	Так

Для розробки застосунку вибір був здійснений у сторону Visual Studio, оскільки воно забезпечує оптимальну та надійну розробку програм на мові C#, а також має розширений набір інструментів для ефективної розробки, відлагодження та тестування програм.

2.2 Аналіз вимог. Use-case діаграми. Основні прецеденти

З метою детального опису функціональності розробленої системи та забезпечення зрозуміння її використання користувачами, використання діаграм прецедентів виявляється важливим та потужним інструментом. Кожен прецедент може бути розглянутий як окремий сервіс, який надається системою та визначає, як користувач може взаємодіяти з системою з метою досягнення певного результату. Діаграма прецедентів дозволяє виокремити та описати поведінку системи з точки зору користувачів, уточнити вимоги до системи та забезпечити ефективну взаємодію між користувачами та системою.

У табл. 2.3 описані функціональні вимоги до застосунку.

Таблиця 2.3 – Функціональні вимоги до застосунку

Вимоги	Опис
REQ-1	Можливість реєстрації нових користувачів та авторизації вже зареєстрованих користувачів у системі
REQ-2	Система повинна здійснювати запис та зберігання журналу подій, що відбулися в системі, з метою моніторингу та аудиту активності користувачів
REQ-3	Додавати та редагувати інформацію облікових записів користувачів
REQ-4	Додати та редагувати інформацію про матеріали курсу

Продовження таблиці 2.3 - Функціональні вимоги до застосунку

REQ-5	Додати та редагувати інформацію тестових запитань курсу
REQ-6	Додати та редагувати інформацію про нагадування на курси
REQ-7	Можливість запуску телеграм-бота, який надає підтримку користувачам у роботі з системою, включаючи відповіді на запити та надання необхідної інформації

Наступним кроком є визначення нефункціональних вимоги до застосунку (табл. 2.4).

Таблиця 2.4 – Нефункціональні вимоги до додатку

Вимоги	Опис
REQ-8	Додаток повинен працювати швидко та ефективно в умовах потенційно великого навантаження користувачів, забезпечуючи швидкий відгук на запити та мінімальні часи відповіді
REQ-9	Додаток повинен бути стійким до помилок та неперервно працювати без несправностей. Він повинен коректно обробляти можливі виняткові ситуації та запобігати витоку даних або системних збоїв
REQ-10	Додаток повинен забезпечувати захист конфіденційності даних користувачів та забезпечувати механізми автентифікації та авторизації, щоб запобігти несанкціонованому доступу до системи
REQ-11	Додаток повинен мати зрозумілий, інтуїтивно зрозумілий та легко використовуваний користувацький інтерфейс, що сприяє зручній навігації та забезпечує зручне взаємодію користувача з системою

Для ідентифікації основних учасників (акторів), що взаємодіють з системою, та визначення їх головних цілей у контексті програмного додатку була розроблена таблиця 2.5. Кожен актор може мати одну або декілька цілей, які він намагається досягти завдяки використанню системи. Ці цілі, в свою чергу, стають основою для визначення головних варіантів використання (use-cases), що допомагають визначити функціональні можливості та поведінку системи.

Таблиця 2.5 – Актори та цілі додатку

Актори	Цілі
Адміністратор	Виконання операцій, пов'язаних з обробкою та керуванням обліковими записами у системі. Його завдання включає управління правами доступу задля безпеки системи та вирішення пов'язаних з аутентифікацією та авторизацією завдань.
Користувач	Взаємодія з системою та виконання різноманітних завдань, пов'язаних зі своїми потребами та вимогами
База даних	Збереженні, організації та забезпеченні доступу до інформації, яка використовується системою

У таблиці 2.6 наведено деталізований перелік сценаріїв використання системи різними акторами. Кожен варіант використання містить унікальне ім'я, яке однозначно ідентифікує його, опис, що відображає основну мету або результат використання, а також послідовність кроків, які актор повинен виконати для досягнення даної мети.

Таблиця 2.6 – Опис варіантів використання додатку

Варіант використання	Ім'я	Опис
UC1	Реєстрація в системі	Дозволяє користувачам пройти процедуру реєстрації для користування застосунком
UC2	Автентифікація в системі	Дозволяє користувачам пройти процедуру автентифікації, що підтверджує їхню ідентичність
UC3	Вивід каталогу користувачів	Користувачі з правами системного адміністратора мають можливість переглядати каталог всіх користувачів системи
UC4	Додати користувача	Користувачі з правами системного адміністратора можуть додавати нових користувачів до системи
UC5	Редагувати користувача	Користувачі з правами системного адміністратора мають можливість редагувати інформацію про обраного користувача зі списку
UC6	Вивід каталогу запитань	Дозволяє користувачеві вивести каталог всіх запитань курсу «Комп'ютерні дискретні структури»
UC7	Додати нове запитання	Дозволяє користувачеві додати інформацію про нове запитання курсу
UC8	Редагувати запитання	Дозволяє користувачеві редагувати інформацію вибраного із списку запитання

Продовження таблиці 2.6 - Опис варіантів використання додатку

UC9	Формування матеріалів курсу	Дозволяє користувачеві здійснити налаштування матеріалів курсу
UC10	Зміна параметрів нагадування	Дозволяє користувачеві змінювати налаштування для нагадування про найближчі іспити та здачу домашніх завдань для студентів курсу
UC11	Запуск телеграм-бота	Дозволяє здійснити запуск телеграм-бота, який надає підтримку користувачам у роботі з системою
UC12	Вивід системних подій	Дозволяє вивести всі події, які відбулися в системі

На підставі наданих даних були розроблені use-case (діаграми прецедентів) для системного адміністратора та користувача, які відображені на рисунках 2.1 та 2.2 відповідно.



Рисунок 2.1 – Діаграма use-case для ролі «системний адміністратор»



Рисунок 2.2 – Діаграма use-case для ролі «користувач»

2.3 Архітектура проекту

Для розробки даного проекту було використано трьохрівневу архітектуру, яка є поширеним типом архітектури програмного забезпечення. Цей підхід базується на поділі системи на три рівні: клієнтський, логічний та даних, кожен з яких має свої відповідності, відповідальності та обмеження.

На першому рівні, клієнтському, знаходяться компоненти, які відповідають за інтерфейс користувача та забезпечують взаємодію з користувачем. Це можуть бути різні типи інтерфейсів, включаючи веб-сторінки, додатки для мобільних пристроїв або десктопні додатки. На цьому рівні зосереджено засоби, які дозволяють користувачам спілкуватися з системою та використовувати її функціональні можливості.

На другому рівні, логічному, знаходяться компоненти, які відповідають за бізнес-логіку системи. Цей рівень включає в себе обробку даних, логіку додатку та взаємодію з базою даних. Його основна функція - забезпечити виконання бізнес-правил та операцій, обробку даних та керування станом системи. На цьому рівні

розташовані сервіси, які надають інтерфейси для взаємодії між клієнтським та даних рівнями, а також забезпечують контроль доступу до даних та забезпечують їх цілісність.

На третьому рівні, рівні даних, знаходиться база даних або інші джерела даних, які використовуються системою. Цей рівень відповідає за збереження та управління даними, які використовуються на інших рівнях. Він забезпечує доступ до даних та виконання операцій збереження, оновлення та видалення даних.

2.3.1 Особливості розробки рівня BLL

Розробка бізнес-логіки сприяє розділенню функціоналу додатку на логічні блоки, що сприяє досягненню більшої модульності та гнучкості в майбутньому. Цей підхід дозволяє незалежно змінювати окремі частини функціоналу без впливу на інші компоненти додатку, спрощуючи підтримку та подальший розвиток проекту.

Крім того, відокремлення бізнес-логіки від бази даних та користувацького інтерфейсу дозволяє використовувати різні системи управління базами даних та різні типи інтерфейсів без необхідності внесення змін до самої бізнес-логіки. Це забезпечує більшу гнучкість та адаптивність додатку до змін у вимогах бізнесу та технологічному середовищі [16].

Такий підхід до структури додатку максимально розділяє логічні компоненти, забезпечуючи їх взаємну незалежність та сприяючи модульності та гнучкості проекту. В результаті, зміни у одному компоненті можуть бути внесені без необхідності внесення змін у інші компоненти, що дозволяє ефективно управляти та розвивати додаток з урахуванням змінних потреб бізнесу та технологій.

Діаграму даного шару показано на рис. 2.3.

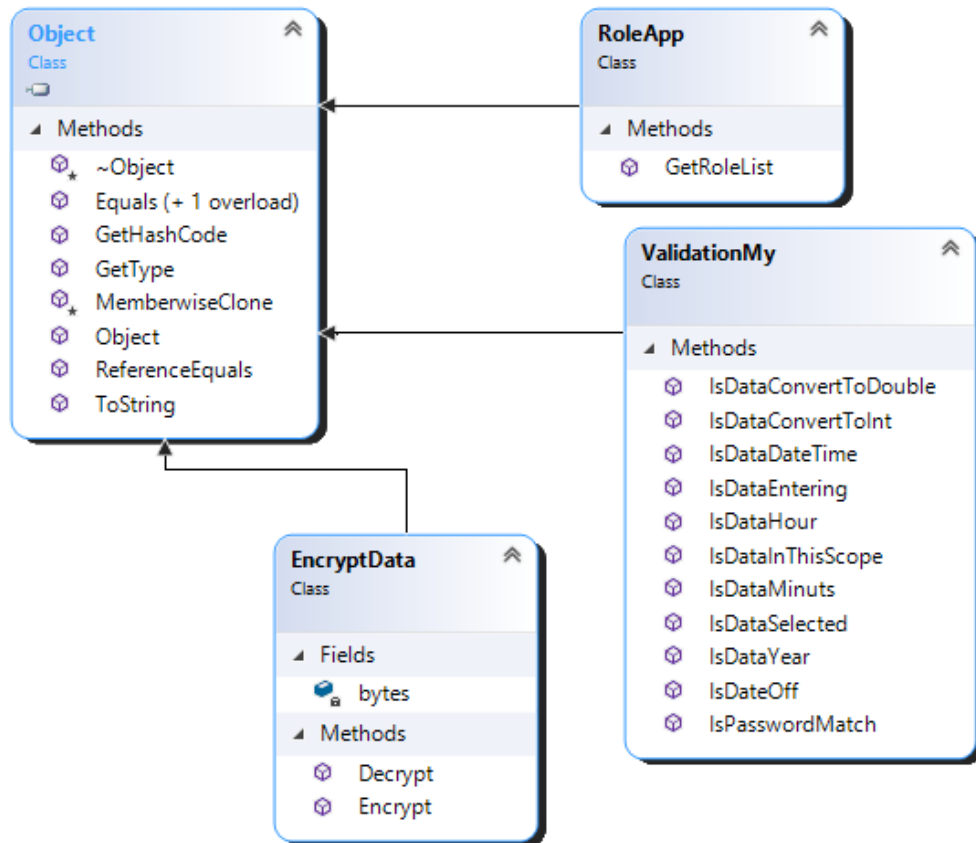


Рисунок 2.3 – Діаграма класів бізнес-логіки

Даний шар архітектури складається із наступних класів:

- клас «EncryptData» відповідає за шифрування та розшифрування даних в системі підтримки навчального процесу на курсі «Комп'ютерні дискретні структури». Цей клас містить методи та алгоритми для забезпечення безпеки та конфіденційності інформації, що зберігається в системі. Він використовує різні криптографічні алгоритми, такі як AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman) для шифрування та розшифрування даних, забезпечуючи їхню надійність та захищеність під час зберігання та передачі;

- клас «RoleApp» відповідає за управління ролями користувачів в системі підтримки навчального процесу. Він містить методи та функціональність, що дозволяють присвоювати, змінювати та видаляти ролі користувачів в системі. Клас забезпечує перевірку прав доступу для кожної ролі та контролює, які функції та можливості доступні для користувачів з різними ролями;

Діаграма даного рівня складається з одинадцяти класів рівня UI та є похідними від класу Form, тобто мають графічний інтерфейс.

Клас «DiscStructMDI» відповідає за головне вікно додатку. Він інкапсулює основний робочий простір додатку та надає користувачу можливість взаємодіяти з різними модулями та функціями, які стосуються навчального процесу. Клас «DiscStructMDI» включає в себе меню, панелі інструментів та інші елементи, які допомагають користувачу здійснювати навігацію, відкривати й керувати різними формами та функціями додатку.

Клас «TelegramBotForm» відповідає за інтеграцію та взаємодію з телеграм-ботом у додатку «підтримка навчального процесу». Він надає користувачу можливість бачити всі повідомлення, що отримує бот. Клас «TelegramBotForm» включає в себе відповідні елементи, які дозволяють користувачу ввести повідомлення, отримати відповідь від бота та виконати інші дії, пов'язані з функціоналом телеграм-бота.

Клас «MaterialsForm» відповідає за роботу з матеріалами курсу в додатку. Він надає користувачу доступ до різних матеріалів, таких як навчальні посібники, презентації, відеоуроки тощо. Клас «MaterialsForm» включає в себе елементи для перегляду та завантаження матеріалів, пошуку за ключовими словами, сортування тощо.

Клас «RemindersForm» відповідає за роботу з нагадуваннями в додатку «підтримка навчального процесу на курсі «Комп'ютерні дискретні структури»». Він надає можливість користувачам встановлювати та керувати нагадуваннями для різних подій, таких як дедлайни, зустрічі, важливі дати тощо. Клас «RemindersForm» містить в собі функціонал для додавання та редагування нагадувань.

Клас «TestFormForm» відповідає за додавання нових завдань для проведення тестування знань на курсі. Він надає можливість викладачам або адміністраторам курсу створювати тестові завдання, формулювати питання, встановлювати правильні відповіді.

Клас «UpdateTestForm» відповідає за редагування тестових завдань в додатку. Він надає можливість викладачам або адміністраторам курсу вносити зміни до наявних тестових завдань, наприклад, змінювати питання, відповіді, правильні відповіді та оцінки. Клас «UpdateTestForm» містить функціонал для вибору тесту, редагування питань та відповідей, а також збереження оновлених даних тесту.

Клас «LoginForm» відповідає за автентифікацію та вхід користувачів до додатку «підтримка навчального процесу». Він надає можливість користувачам ввести свої облікові дані, такі як ім'я користувача та пароль, для отримання доступу до функціоналу системи. Клас «LoginForm» перевіряє введені дані на правильність та здійснює авторизацію користувача, дозволяючи або обмежуючи його доступ до різних ресурсів та функцій системи.

Клас «PersonalizationForm» відповідає за персоналізацію налаштувань та профілю користувача в додатку. Він надає можливість користувачам налаштовувати особисті дані, які пов'язані з їх профілем, наприклад, змінювати інформацію про себе, змінювати пароль тощо. Клас «PersonalizationForm» містить в собі відповідні елементи, що дозволяють користувачу внести зміни до свого профілю та налаштувань.

Клас «SystemLogForm» відповідає за ведення системного журналу подій в додатку. Він забезпечує реєстрацію та зберігання інформації про різноманітні події та дії, які відбуваються в системі. Клас «SystemLogForm» дозволяє переглядати журнал подій, включаючи інформацію про входи, виходи тощо.

Клас «UpdateUsersForm» відповідає за редагування інформації користувачів. Він надає можливість викладачам або адміністраторам курсу змінювати дані, пов'язані з обліковими записами користувачів. Клас «UpdateUsersForm» містить функціонал для вибору конкретного користувача, редагування його персональних даних та збереження оновлених інформаційних записів.

Клас «UsersForm» відповідає за відображення списку користувачів в додатку «підтримка навчального процесу на курсі «Комп'ютерні дискретні структури»». Він надає можливість переглядати інформацію про користувачів, таку як їх імена,

ролі, дати реєстрації тощо. Клас «UsersForm» містить функціонал для відображення списку користувачів, а також взаємодії з іншими компонентами системи для отримання та відображення інформації про користувачів.

2.3.3 Особливості розробки DAL

Метою компоненти DAL (Data Access Layer) є забезпечення доступу до даних, розташованих у базі даних, з різних рівнів системи, зокрема з сервісного рівня (BLL). У контексті системи, DAL було реалізовано за допомогою технології ADO.NET. Для цього використовувався об'єкт System.Data.OleDb, що дозволяє встановлювати з'єднання з базою даних за допомогою різних провайдерів даних (див. рис. 2.5). Компонента DAL виконує завдання роботи з даними, такі як отримання, зміна, видалення та додавання даних у базу даних. Вона забезпечує абстракцію бази даних від решти системи, що дозволяє забезпечити незалежність від конкретного постачальника бази даних та спрощує взаємодію з даними [18].

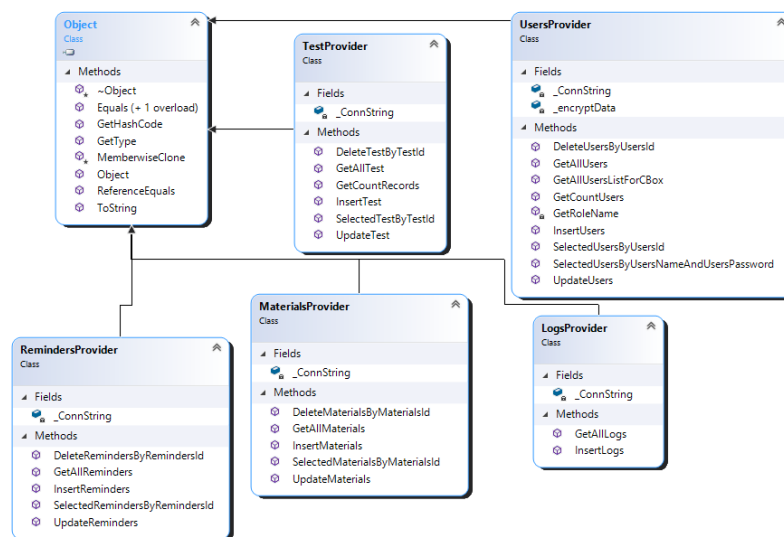


Рисунок 2.5 – Діаграма класів користувацького інтерфейсу

Як видно із рис. 2.5, діаграма класів рівня даних складається із п'яти основних класів:

- клас «LogsProvider» відповідає за надання доступу до журналу подій в системі. Його головна функція полягає в зборі, зберіганні та наданні інформації про

події, які відбулися в системі. Цей клас надає методи для додавання нових записів у журнал, отримання списку подій;

– клас «MaterialsProvider» відповідає за надання доступу до матеріалів курсу в системі. Його головна функція полягає в зборі, зберіганні та наданні інформації про навчальні матеріали, які використовуються на курсі «Комп'ютерні дискретні структури»;

– клас «RemindersProvider» відповідає за надання доступу до нагадувань на курсі. Його основна функція полягає в зборі, зберіганні та наданні інформації про нагадування, пов'язані з різними аспектами навчального процесу;

– клас «TestProvider» відповідає за надання доступу до тестових запитань та виконання тестування в системі. Його основна функція полягає в зборі, зберіганні та наданні інформації про тестові запитання, а також проведенні тестування знань користувачів. Клас «TestProvider» надає методи для додавання нових тестових запитань, редагування існуючих запитань, отримання списку запитань;

– клас «UsersProvider» відповідає за надання доступу до інформації про користувачів системи. Його головна функція полягає в зборі, зберіганні та наданні даних про користувачів, включаючи їх особисту інформацію, облікові записи та ролі. Клас «UsersProvider» надає методи для додавання нових користувачів, редагування існуючих записів, отримання списку користувачів, а також фільтрації та сортування користувачів за певними критеріями. Також, даний клас може використовуватись іншими компонентами системи для отримання інформації про користувачів та керування обліковими записами.

3. РОЗРОБКА, ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ НАВЧАЛЬНОГО ПРОЦЕСУ

3.1 Проектування бази даних

Розробка проекту розпочинається із проектування бази даних. У даному випадку було використано метод «сутність-зв'язок» (ER метод) - це підхід до проектування баз даних, що передбачає визначення сутностей, їх атрибутів та зв'язків між ними. Застосування цього методу дозволяє створити логічну модель бази даних, що відображає предметну область та відносини між її елементами [19].

У процесі проектування бази даних за методом «сутність-зв'язок» для предметної області було виділено наступні сутності та атрибути:

користувачі системи: дані про облікові записи користувачів, такі як прізвище, ім'я, ім'я облікового запису, пароль, загальна інформація та ідентифікатор ролі.

- матеріали (інформація про матеріали курсу): назва та опис та шлях до матеріалу;
- нагадування (інформація про нагадування): назва нагадування, дата та опис;
- завдання на тести (інформація про завдання на тести): запитання, варіант відповіді а, варіант відповіді б, варіант відповіді с, варіант відповіді д;
- події (інформація про події в системі) ідентифікатор користувача, опис події, дата/час події.

Організовано такі зв'язки між таблицями:

- «Users» (поле «UsersId») – «Logs» (поле «UsersId»), вид зв'язку 1:N.

ER-діаграма є значущим етапом у проектуванні бази даних, оскільки вона дозволяє графічно відобразити взаємозв'язки між сутностями та їх атрибутами. Під час проектування була створена ER-діаграма для бази даних, яка ілюструє зв'язки між таблицями та їх полями (див. рис. 3.1). Ця модель є важливою для подальшого проектування та розробки функціоналу додатку. Вона сприяє виявленню залежностей між сутностями та допомагає визначити ключові аспекти системи. Це

забезпечує підґрунтя для ефективного впровадження та управління базою даних у межах проекту.

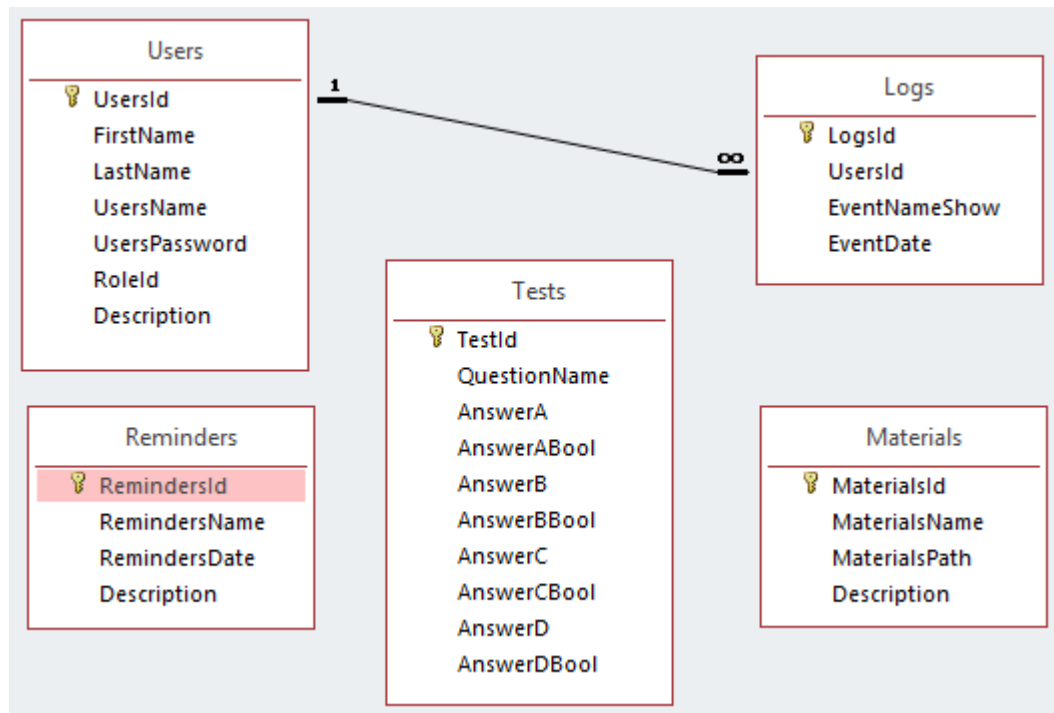


Рисунок 3.1 – Діаграма бази даних

3.2 Розробка програмних модулів системи

Наступним кроком після створення бази даних є підключення її до системи з використанням інтегрованого середовища розробки Visual Studio 2019 [20]. Це дозволяє зв'язати всі таблиці з формами, на яких можна додавати, редагувати та видаляти дані. Для здійснення підключення необхідно створити змінну «CONNECT» у файлі конфігурації проекту «App.config» і встановити значення параметрів, які можна переглянути на рисунку 3.2.

Цей крок забезпечує встановлення з'єднання між програмним додатком та базою даних, що дозволяє зчитувати та записувати дані. Підключення до бази даних використовується для взаємодії з інформацією, збереженою у базі даних, та забезпечує функціональність для роботи з даними у програмі.

```

<appSettings>
  <!-- Підключення до бази даних -->
  <add key="CONNECT" value="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirectory|\DataBase.mdb;" />
  <add key="ClientSettingsProvider.ServiceUri" value="" />
</appSettings>

```

Рисунок 3.2 – Змінна із параметрами налаштування бази даних

Для оптимальної роботи з базою даних в проєкті використовується простір імен System.Data.OleDb. Цей простір містить класи, такі як OleDbConnection, які забезпечують взаємодію з даними у базі даних, виконання запитів та збереження змін. Шляхом додавання елемента menuStrip до головного вікна проєкту створюється меню, яке надає користувачам можливість вибирати опції та викликати функції програми за допомогою зручного інтерфейсу. Результат такого додавання можна побачити на рисунку 3.3.

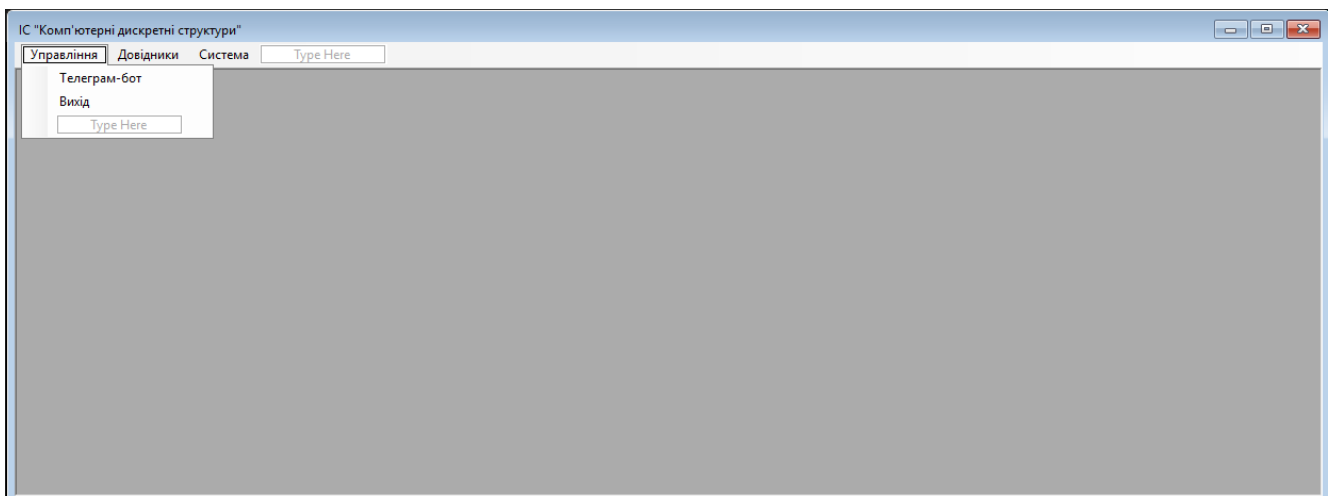


Рисунок 3.3 – Додавання головного меню

Кожному пункту меню було надано відповідний код, який створює екземпляр відповідної форми. При виборі певного пункту меню, відповідна форма відкривається, і її вікно стає активним. Додатково, був включений код, який забезпечує закриття попередньо відкритого вікна перед відкриттям нового. Це гарантує правильне відображення вікон та допомагає уникнути можливих

конфліктів. Аналогічний код застосовується для всіх пунктів меню, забезпечуючи їх належну роботу та взаємозв'язок з формами. На рис. 3.4 наведений приклад такого коду для одного з пунктів меню.

```
private void телеграмботToolStripMenuItem_Click(object sender, EventArgs e) {
    CloseAllWindows();
    TelegramBotForm telegramBotForm = new TelegramBotForm();
    telegramBotForm.MdiParent = this;
    telegramBotForm.WindowState = FormWindowState.Maximized;
    telegramBotForm.Show();
}
```

Рисунок 3.4 – Код пунктів меню

Після цього були створені класи для взаємодії з базою даних. Нижче наведені часткові реалізації методів різних класів разом з детальним описом. Наприклад, для додавання інформації про нове тестове завдання до бази даних був створений метод з назвою «InsertTest». Код цього методу наведено на рисунку 3.5.

```
public void InsertTest(string QuestionName, string AnswerA, bool AnswerABool, string AnswerB, bool AnswerBBool,
    string AnswerC, bool AnswerCBool, string AnswerD, bool AnswerDBool) {
    string SqlString = "INSERT INTO Tests (QuestionName, AnswerA, AnswerABool, AnswerB, AnswerBBool, AnswerC, AnswerCBool, " +
    "AnswerD, AnswerDBool) Values(?, ?, ?, ?, ?, ?, ?, ?, ?)";

    using (OleDbConnection conn = new OleDbConnection(_ConnString)) {
        using (OleDbCommand cmd = new OleDbCommand(SqlString, conn)) {
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("QuestionName", QuestionName);
            cmd.Parameters.AddWithValue("AnswerA", AnswerA);
            cmd.Parameters.AddWithValue("AnswerABool", AnswerABool);
            cmd.Parameters.AddWithValue("AnswerB", AnswerB);
            cmd.Parameters.AddWithValue("AnswerBBool", AnswerBBool);
            cmd.Parameters.AddWithValue("AnswerC", AnswerC);
            cmd.Parameters.AddWithValue("AnswerCBool", AnswerCBool);
            cmd.Parameters.AddWithValue("AnswerD", AnswerD);
            cmd.Parameters.AddWithValue("AnswerDBool", AnswerDBool);
            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();
        }
    }
}
```

Рисунок 3.5 – Код методу «InsertTest»

Метод «InsertTest», відповідає за вставку нового запису з питанням тесту та його відповідями до таблиці «Tests» у базі даних.

У першому рядку методу визначається рядок SQL-запиту, який містить команду INSERT INTO, яка вказує, куди та які значення потрібно вставити.

Далі, використовуючи об'єкт OleDbConnection, створюється з'єднання з базою даних. У блоку using створюється об'єкт OleDbCommand, який представляє SQL-запит та виконується на з'єднанні.

Для параметрів запиту використовуються методи AddWithValue, де перший аргумент вказує ім'я параметра, а другий - його значення.

Після встановлення всіх параметрів, викликається метод ExecuteNonQuery, який виконує SQL-запит, не повертаючи дані. У кінці методу з'єднання з базою даних закривається, використовуючи ключове слово Close.

Для видалення даних про завдання, також був розроблений метод, код якого представлено на рис. 3.6.

```
public void DeleteTestById(int TestId) {
    string SqlString = "DELETE FROM Tests WHERE TestId=" + TestId.ToString();
    using (OleDbConnection conn = new OleDbConnection(_ConnString)) {
        using (OleDbCommand cmd = new OleDbCommand(SqlString, conn)) {
            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();
        }
    }
}
```

Рисунок 3.6 – Код методу «DeleteTestById»

Метод DeleteTestById виконує видалення запису з таблиці «Tests» за допомогою SQL-запиту DELETE FROM, де ідентифікатор тесту (TestId) відповідає заданому значенню. З'єднання з базою даних відкривається, виконується запит за допомогою методу ExecuteNonQuery, а потім з'єднання закривається.

У наступному кроці було реалізовано форму під назвою «TelegramBotForm» для керування телеграм ботом. Ця форма створена з метою забезпечення взаємодії з ботом під назвою «DiscreteStructures», який був створений за допомогою сервісу «BotFather». Основною метою цього бота є обробка запитів користувачів та надання їм відповідей.

На початку було створено обробник подій «TelegramBotForm_Load» код реалізації якого показано на рис. 3.7.

```
1 reference
private void TelegramBotForm_Load(object sender, EventArgs e) {
    botClient = new TelegramBotClient(_Token);
    botClient.OnMessage += Bot_OnMessage;
    botClient.OnCallbackQuery += Bot_OnCallbackQuery;
    botClient.StartReceiving();
    ReportTBox.Text = "Запущено ТГ бота\r\n";
}
```

Рисунок 3.7 – Код події подій «TelegramBotForm_Load» для старту ТГ боту

Обробник подій «TelegramBotForm_Load» виконується при завантаженні форми «TelegramBotForm». У його роботі відбувається наступне:

- створюється екземпляр класу TelegramBotClient з використанням збереженого токена (_Token), який використовується для авторизації та взаємодії з телеграм ботом;
- встановлюється обробник події OnMessage, який виконується при отриманні нового повідомлення від користувача;
- встановлюється обробник події OnCallbackQuery, який виконується при отриманні відповіді на клавіші з меню бота;
- запускається процес отримання повідомлень з телеграм бота за допомогою методу StartReceiving(). Це дозволяє боту почати слухати та обробляти вхідні повідомлення.
- відображається повідомлення «Запущено ТГ бота» в текстовому полі ReportTBox, що підтверджує успішний запуск бота.

Для обробки повідомлень від користувачів і керування логікою тестування через телеграм бота розроблено подію «Bot_OnMessage». Код події «OnMessage», що призначена для аналізу введених даних користувачів показана на рис. 3.8.

```

async void Bot_OnMessage(object sender, MessageEventArgs e) {
    if (e.Message.Text != null) {
        // Updating the TextBox from the UI thread
        ReportTBX.Invoke((MethodInvoker)delegate {
            // Set the new value to TextBox
            ReportTBX.Text += e.Message.Chat.Id.ToString() + "\r\n";
            ReportTBX.Text += e.Message.Text + "\r\n";
        });

        if (!userStates.ContainsKey(e.Message.Chat.Id)) {
            userStates[e.Message.Chat.Id] = new UserTestState();
        }

        UserTestState userState = userStates[e.Message.Chat.Id];
        if (userState.InTest) { // Process the user's answer
            userState.QuestionIndex++;
            if (userState.QuestionIndex >= _TestList.Count) // assuming the test has Count questions
            {
                userState.InTest = false;
                string testComplite = "Тестування завершено!" + "\r\n";
                testComplite += "Ви правильно відповіли на " + userStates[e.Message.Chat.Id].Rating + " з " + _TestList.Count + " запитань" + "\r\n";
                await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: testComplite);
            } else {
                if (_validation.IsDataConvertToInt(e.Message.Text)) {
                    RightAnswer(e.Message.Chat.Id, Convert.ToInt32(e.Message.Text), userState.QuestionIndex - 1);

                    string textQuestion = _TestList[userState.QuestionIndex].QuestionName + "\r\n";
                    textQuestion += "1. " + _TestList[userState.QuestionIndex].AnswerA + "\r\n";
                    textQuestion += "2. " + _TestList[userState.QuestionIndex].AnswerB + "\r\n";
                    textQuestion += "3. " + _TestList[userState.QuestionIndex].AnswerC + "\r\n";
                    textQuestion += "4. " + _TestList[userState.QuestionIndex].AnswerD + "\r\n";
                    textQuestion += "Вкажіть № відповіді та натисніть 'Enter' для продовження";

                    await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: textQuestion);
                } else {
                    await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: "Будь-ласка введіть число у діапазоні від 1 до 4!");
                }
            }
        } else if (e.Message.Text == "/start") {
            await SendMainMenu(e.Message.Chat.Id);
        } else {
            // handle non-test messages here
        }
    }
}

```

Рисунок 3.8 – Код події «OnMessage» для аналізу запитів

Подія «Bot_OnMessage» виконується при отриманні повідомлення від користувача через телеграм бота. Основна логіка методу наступна:

- Перевіряється, чи не є текст повідомлення порожнім.
- Застосовується механізм «Invoke» для оновлення текстового поля ReportTBX з основного потоку користувацького інтерфейсу. Повідомлення відправника та текст повідомлення додаються до текстового поля.
- Перевіряється, чи вже існує поточний стан користувача в словнику userStates. Якщо ні, створюється новий об'єкт UserTestState для даного користувача.
- Перевіряється, чи користувач знаходиться у режимі проходження тесту (userState.InTest). Якщо так, виконується логіка обробки відповіді користувача на питання:

- збільшується індекс поточного питання (userState.QuestionIndex);
- перевіряється, чи користувач відповів на всі питання тесту. Якщо так, тестування завершується і відправляється повідомлення з результатами тестування;
- якщо не всі питання відповідної до поточного індексу, перевіряється, чи введена користувачем відповідь є числом;
- якщо так, викликається метод RightAnswer, який перевіряє правильність відповіді користувача на попереднє питання;
- потім відправляється наступне питання користувачу;
- якщо введена відповідь не є числом, відправляється відповідне повідомлення.
- Якщо користувач не знаходиться у режимі проходження тесту, перевіряється, чи текст повідомлення є командою «/start». Якщо так, викликається метод SendMainMenu для надсилання головного меню користувачу.
- У випадку, якщо повідомлення не відповідає тестовим повідомленням, можна розмістити код для обробки інших типів повідомлень.

Після цього було розроблено метод «SendMainMenu», що дозволяє користувачу обрати дію з головного меню, надсилаючи відповідне значення кнопки через телеграм бота (рис. 3.9).

```

async System.Threading.Tasks.Task SendMainMenu(long chatId) {
    var inlineKeyboard = new InlineKeyboardMarkup(new[] {
        // First row
        new [] {
            InlineKeyboardButton.WithCallbackData("Нагадування", "reminders"),
            InlineKeyboardButton.WithCallbackData("Матеріал", "materials"),
            InlineKeyboardButton.WithCallbackData("Довідка", "help")
        },
        // Second row
        new [] {InlineKeyboardButton.WithCallbackData("Тестування", "test")});
    await botClient.SendTextMessageAsync( chatId: chatId, text: "Оберіть дію:", replyMarkup: inlineKeyboard );
}

```

Рисунок 3.9 – Код події «SendMainMenu» для виведення меню у чаті

Основна логіка методу наступна:

- створюється об'єкт inlineKeyboard типу InlineKeyboardMarkup, який представляє собою розташування кнопок у вигляді клавіатури;

- кожен рядок клавіатури містить один або кілька `InlineKeyboardButton`, які представляють окремі кнопки з текстом і відповідними значеннями;
- використовуючи `botClient`, викликається метод `SendTextMessageAsync` для надсилання повідомлення користувачу;
- передається `chatId`, щоб визначити отримувача повідомлення;
- передається текст повідомлення, що містить інструкцію для користувача;
- передається розмітка клавіатури (`inlineKeyboard`) для відображення кнопок у повідомленні.

Отже, в даному розділі було проведено частковий опис основним моментів функціонування розробленого програмного забезпечення.

3.3 Результати функціонального та модульного тестування

Тестування відіграє важливу роль у розробці програмного забезпечення, спрямованого на перевірку відповідності вимогам та очікуванням користувачів. Цей процес дозволяє виявити потенційні проблеми та помилки в роботі програми, а також забезпечити їх виправлення. У контексті розробки додатку «ІС «Комп'ютерні дискретні структури»» було проведено функціональне тестування, яке включало перевірку основних функцій додатку та його відповідність вимогам користувачів. Для цього були розроблені тест-кейси, що відображали функціональні вимоги програмного забезпечення. Основною метою тестування було визначення рівня стабільності та надійності додатку. В результаті проведеного тестування було досягнуто підвищення якості програмного забезпечення та забезпечено його коректну роботу.

Тест-кейс №1. Вхід до системи. Перевірка на наявність правильної авторизації користувача:

- відкрити форму авторизації;
- ввести існуючий логін та пароль користувача;
- перевірити, що користувач має доступ до головної форми застосунку.

В результаті, користувач має мати доступ до головної форми застосунку.

Тест-кейс №2. Додавання запису про нове запитання для проведення тестування знань. Перевірка на правильність заповнення полів:

- відкрити форму додавання нового запису запитання;
- заповнити всі обов'язкові поля (запитання, варіанти відповіді, та правильну відповідь);
- натиснути кнопку «Додати»;
- перевірити, що новий запис тестування знань збережений у базі даних.

В результаті, запис тестування знань має бути доданий в базу даних та відображається у списку всіх запитань.

Тест-кейс №3. Редагування запису запитання. Перевірка на правильність редагування полів:

- вибрати необхідний запис;
- змінити значення одного з полів (наприклад, варіант відповіді);
- натиснути кнопку «Зберегти»;
- перевірити, що відредагований запис вибраного запитання збережений у БД.

В результаті, запис вибраного запитання має бути відредагований в базі даних та відображається у списку всіх записів із запитаннями.

Тест-кейс №3. Видалення запису доданого запитання. Перевірка на правильність видалення:

- відкрити форму перегляду записів запитань тестування знань;
- вибрати необхідне запитання із списку, яке потрібно видалити;
- натиснути кнопку «Видалити»;
- перевірити, що обраний запис був успішно видалений з бази даних.

В результаті, інформація про запитання тестування знань має бути видалена коректно та більше не відображається у списку з іншими записами.

Після завершення функціонального тестування системи, наступним кроком є модульне тестування, яке є важливим етапом у розробці програмного забезпечення.

Модульне тестування дозволяє перевірити роботу окремих модулів або компонентів програми з метою визначення відповідності вимогам та специфікації.

Для проведення модульного тестування в проекті було створено новий проект типу «Unit Test Project» у середовищі Visual Studio. Далі були визначені тести, які охоплюють функції програми, та запущені для перевірки їх правильності та працездатності. Результати модульного тестування, включаючи пройдені тести та можливі помилки, можна переглянути на рис. 3.10.

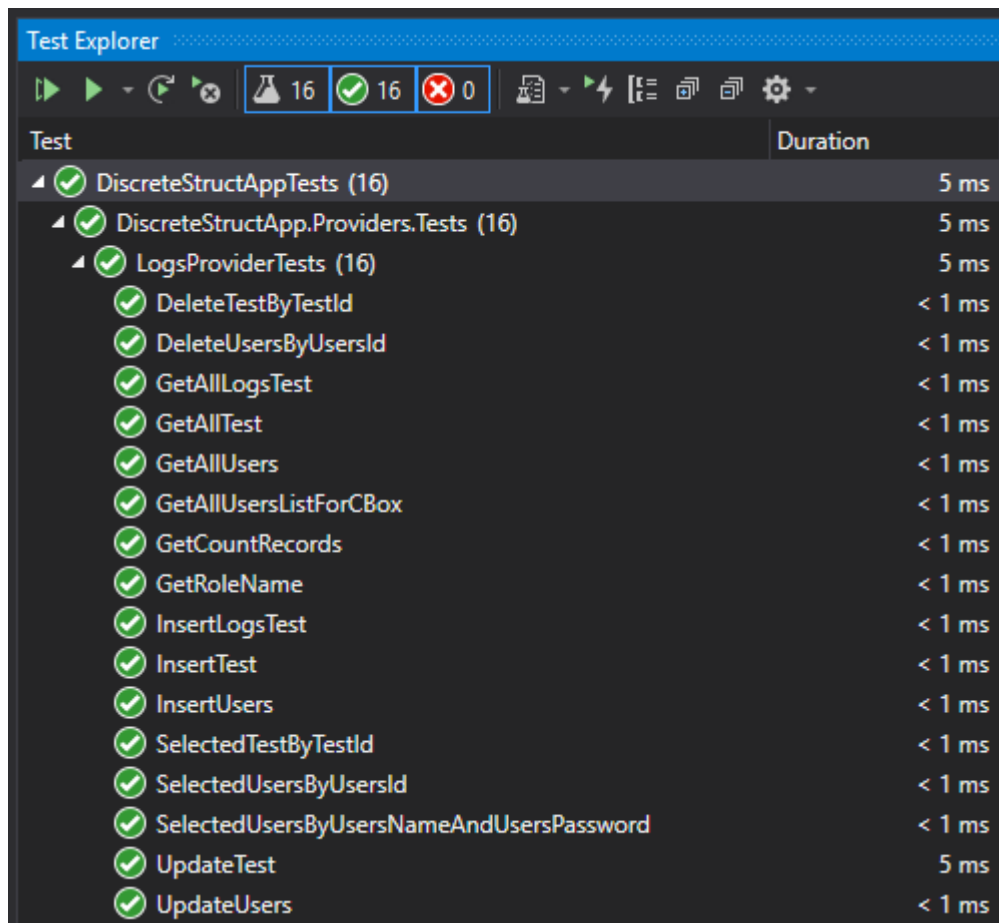


Рис. 3.10. Результати проведення модульного тестування

Модульне тестування дозволило виявити проблеми та помилки в окремих модулях програми, що сприяло поліпшенню якості програмного забезпечення та забезпечило його коректну роботу. Результати тестування є важливою інформацією для подальшої оптимізації та вдосконалення програми перед її релізом.

3.4 Інструкція користувачеві програми для підтримки навчального процесу

Після виконання програми, з'являється вікно авторизації, де користувач має можливість ввести свій логін та пароль (рис. 3.11). Цей крок є необхідним для ідентифікації користувача та забезпечення безпеки його персональних даних. Введені дані будуть перевірені на відповідність збереженим обліковим записам, що дозволить користувачеві отримати доступ до відповідних функцій та ресурсів програми. Авторизаційне вікно забезпечує приватність та конфіденційність користувача, а також забезпечує захист від несанкціонованого доступу до системи.

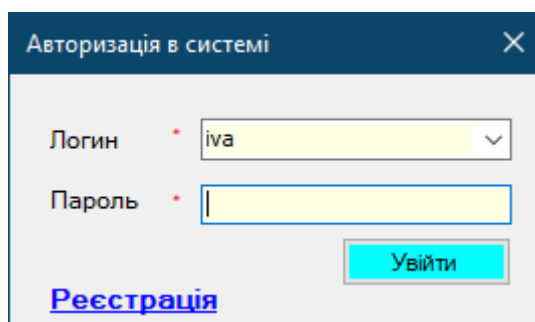
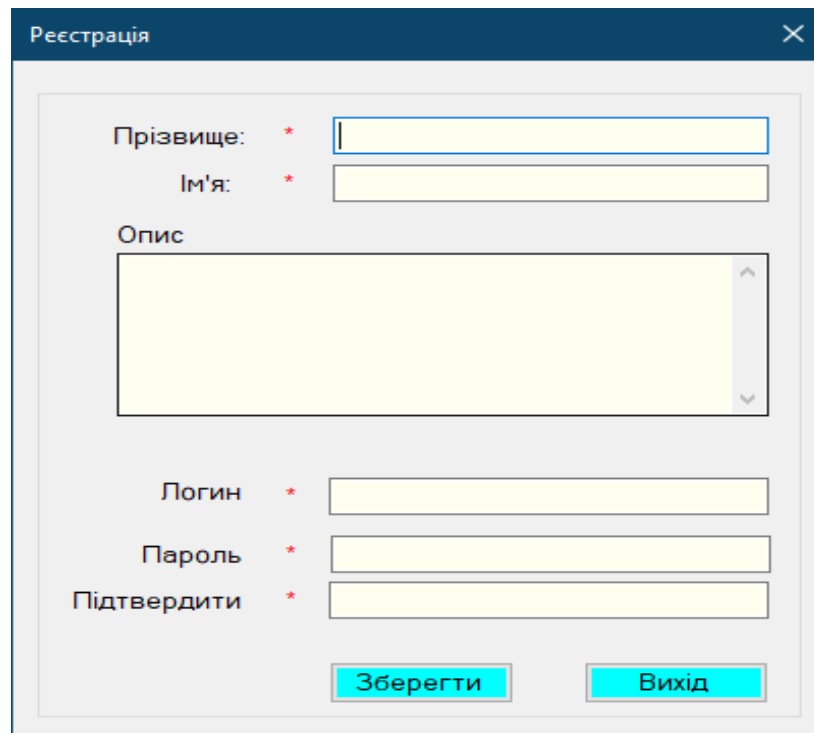


Рисунок 3.11 – Автентифікація користувача в системі

У межах програмного застосунку "ІС «Комп'ютерні дискретні структури»" надається можливість користувачам створити свій власний обліковий запис шляхом використання форми реєстрації. Після натискання кнопки "Реєстрація" на екрані авторизації, користувач буде автоматично перенаправлений на форму реєстрації, де йому необхідно буде ввести своє повне ім'я, прізвище, електронну адресу та обрану пароль. Після заповнення усіх відповідних полів, користувач повинен натиснути кнопку "Зберегти", щоб зареєструвати свій обліковий запис (рис. 3.12). З метою максимальної безпеки, система зберігає паролі користувачів у зашифрованому вигляді та забезпечує надійний захист від несанкціонованого доступу до особистої інформації користувачів.



Реєстрація

Прізвище: *

Ім'я: *

Опис

Логин *

Пароль *

Підтвердити *

Рисунок 3.12 – Форма реєстрації користувачів

У програмі, якщо вона ще не має жодного зареєстрованого користувача, при реєстрації першого користувача система автоматично присвоює йому роль системного адміністратора, в той час як іншим користувачам присвоюються ролі звичайних користувачів. Це важливо для забезпечення безпеки та надійності системи з самого початку її функціонування. Для полегшення пошуку користувача програма надає можливість використовувати випадаючий список з наявними іменами користувачів, а також автоматично піднімає введене користувачем ім'я, що допомагає прискорити процес пошуку потрібного користувача.

Після успішної авторизації користувач отримує доступ до головного інтерфейсу додатку (рис. 3.13), де він може переглядати та редагувати інформацію, пов'язану з підтримкою навчального процесу. Цей інтерфейс надає користувачеві зручність у взаємодії з програмою та забезпечує доступ до всіх необхідних функцій та даних, пов'язаних з навчальним процесом.

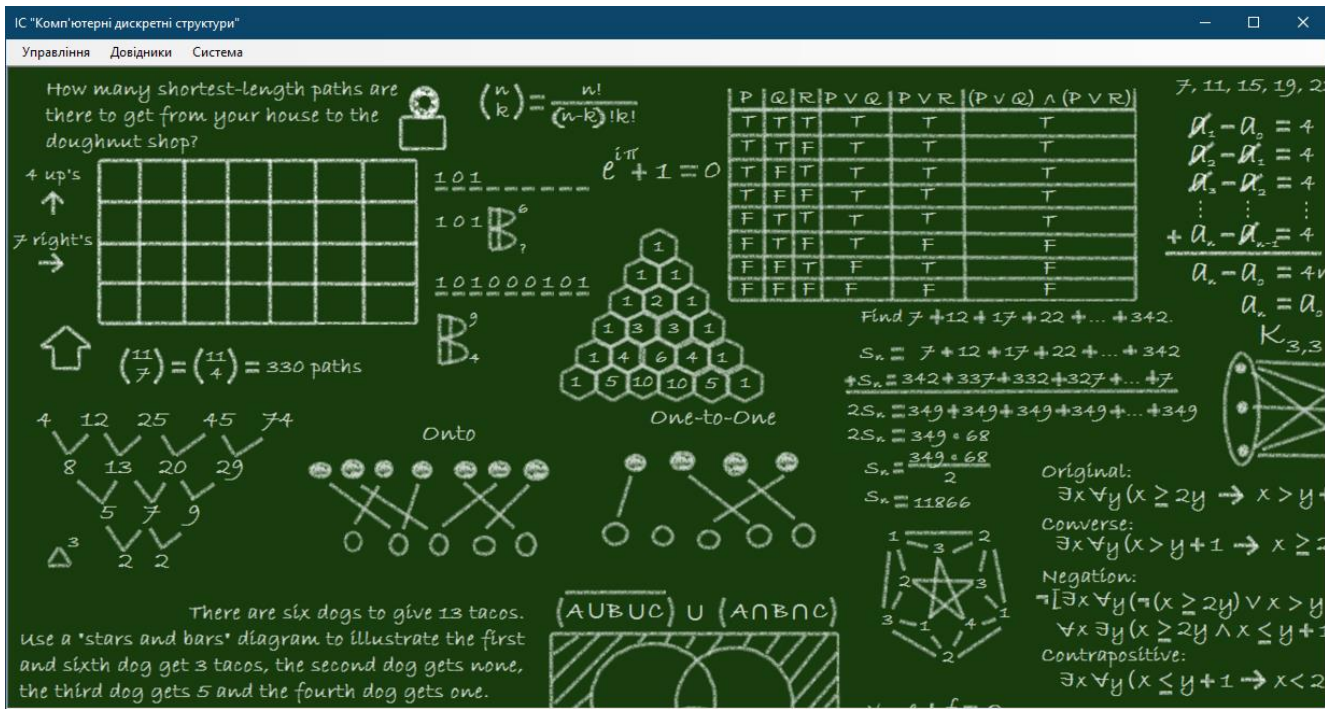


Рисунок 3.13 – Головне меню програми

У даній системі використовуються дві ролі: системного адміністратора та звичайного користувача. Роль системного адміністратора надається користувачам з особливими привілеями та дозволяє їм здійснювати певні додаткові функції. Основні можливості системного адміністратора включають керування обліковими записами та перегляд подій системи.

Система передбачає заповнення певних полів з необхідною інформацією. Під час додавання матеріалів користувачу потрібно (рис. 3.14):

- вибрати файл. Користувач може вибрати файл, який містить матеріали курсу "Комп'ютерні дискретні структури". Це може бути файл у будь-якому форматі, такому як PDF, DOC, PPT і т.д. Користувач може використовувати діалогове вікно для вибору файлу з локального пристрою або інших джерел, наприклад, хмарного сховища;
- вказати Назву матеріалу. Користувач повинен ввести назву матеріалу курсу, що ясно і конкретно відображає його зміст та тематику. Назва повинна бути інформативною та лаконічною, щоб дати користувачам чітке уявлення про зміст матеріалу;

– провести опис матеріалу. Користувач має можливість надати додатковий опис матеріалу. Цей опис може містити деталі про зміст, цільову аудиторію, рекомендації щодо використання та будь-яку іншу важливу інформацію, яка допоможе користувачам краще розуміти та оцінити матеріал.

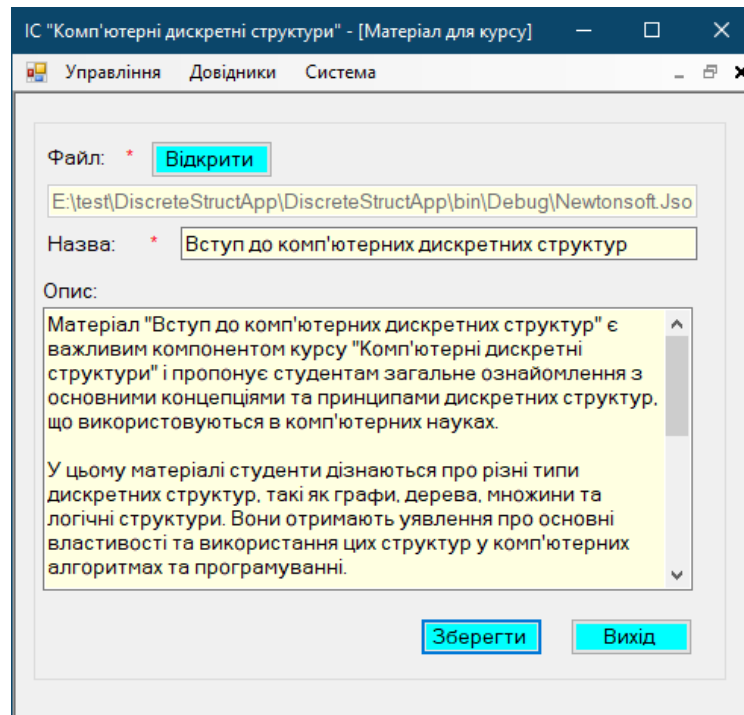


Рисунок 3.14 – Вікно для зміни інформації про курси

Процес створення нагадування для курсу "Комп'ютерні дискретні структури" включає заповнення необхідних полів з інформацією про нагадування. Під час створення нагадування користувачу необхідно вказати дату, назву та опис нагадування (рис. 3.15). Інформація про нагадування включає:

- дата нагадування. Користувач повинен вибрати дату, коли нагадування має бути відправлене. Це може бути дата, коли відбудеться важлива подія, дедлайн або будь-який інший часовий пункт, який вимагає уваги користувачів курсу;
- назва нагадування. Користувач повинен ввести коротку, але змістовну назву нагадування. Це дозволяє користувачам швидко ідентифікувати та розуміти зміст нагадування. Назва повинна бути чіткою і інформативною;

– опис нагадування. Користувач має можливість надати додатковий опис для нагадування. Цей опис може містити деталі, пояснення або конкретні вказівки, пов'язані з нагадуванням. Опис допомагає користувачам краще розуміти важливість та зміст нагадування.

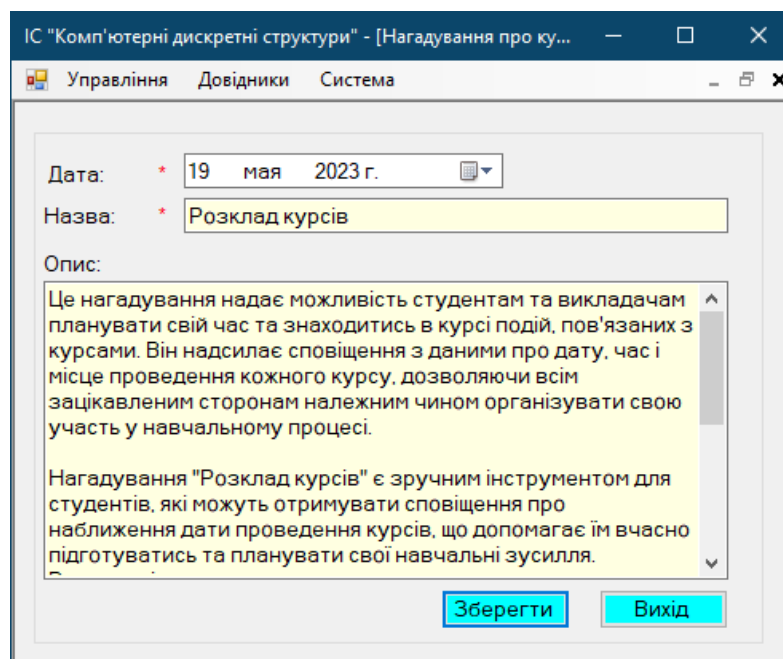


Рисунок 3.15 – Вікно для зміни інформації про нагадування

Після введення необхідних даних користувач має можливість зберегти створене нагадування. Це може бути здійснено за допомогою кнопки "Зберегти". Після збереження нагадування буде доступне для перегляду та відображення користувачам, що використовують Telegram чат-бот.

Процес додавання інформації для тестування знань на курсі "Комп'ютерні дискретні структури" включає введення запитання, варіантів відповідей та позначення правильних відповідей. Описуємо цей процес крок за кроком (рис. 3.16):

– запитання. Користувач повинен ввести запитання, яке буде поставлено в тестуванні. Запитання повинно бути зрозумілим, конкретним і пов'язаним з вивченим матеріалом курсу "Комп'ютерні дискретні структури";

- варіанти відповідей. Користувач повинен надати 4 варіанти відповідей для запитання. Кожен варіант відповіді повинен бути коротким і зрозумілим. Вони можуть бути позначені як А, В, С, D або використовувати іншу систему позначень для ідентифікації варіантів;
- правильні відповіді. Користувач повинен вказати, які з варіантів відповідей є правильними. Це може бути одна або більше правильних відповідей. Користувач може використовувати позначення, наприклад, вибирати номери варіантів відповідей, які є правильними;
- збереження тесту. Після введення всієї необхідної інформації, користувач має можливість зберегти тест, щоб додати його до курсу "Комп'ютерні дискретні структури". Збереження може бути реалізовано за допомогою кнопки "Зберегти" або подібного елемента у інтерфейсі додатку.

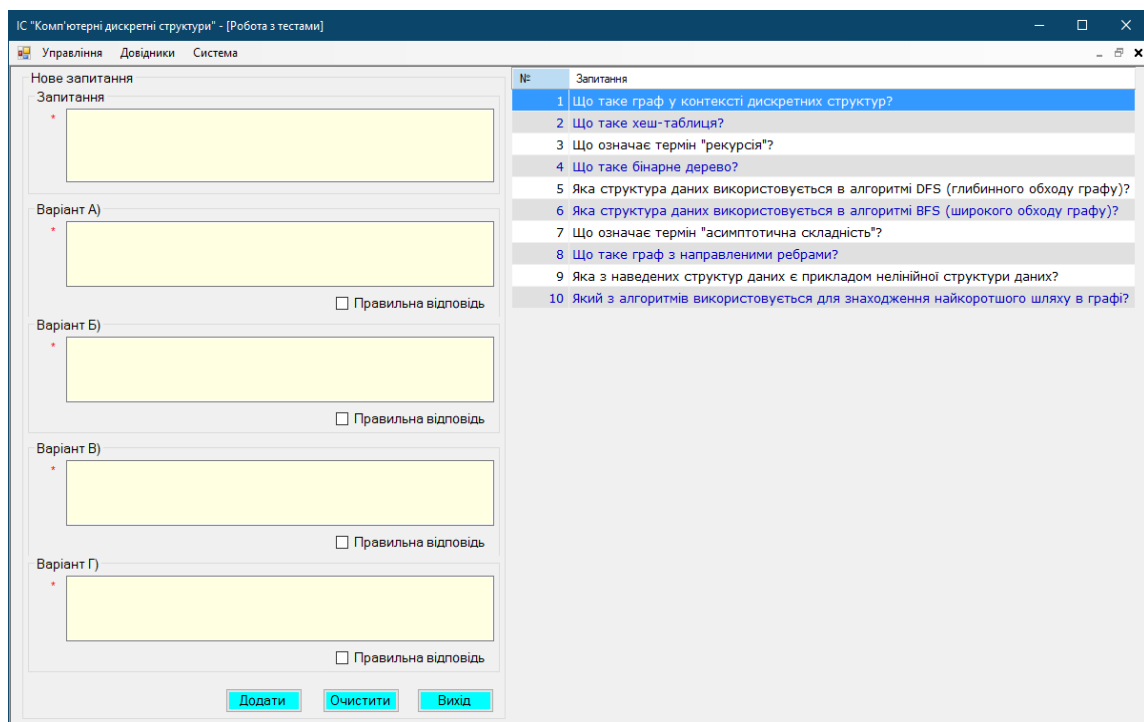


Рисунок 3.16 – Вікно для додавання інформації для тестування знань

Щоб користуватись Telegram -ботом користувачу потрібно знайти чат із назвою «DiscreteStructures» та відкривши його відправити команду «/start», після чого з'явиться головне меню чату (рис. 3.17).

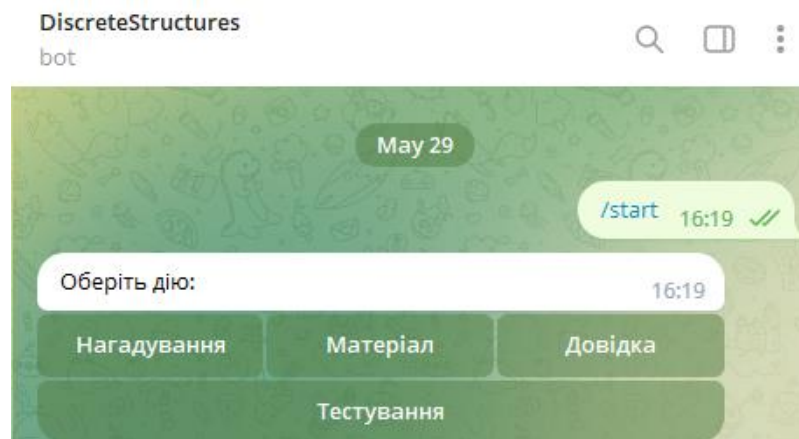


Рисунок 3.17 – Головне меню Telegram –бота

При натисканні на пункт меню "Нагадування" користувач отримує доступ до функціоналу пов'язаного з нагадуваннями про курси (рис. 3.18).

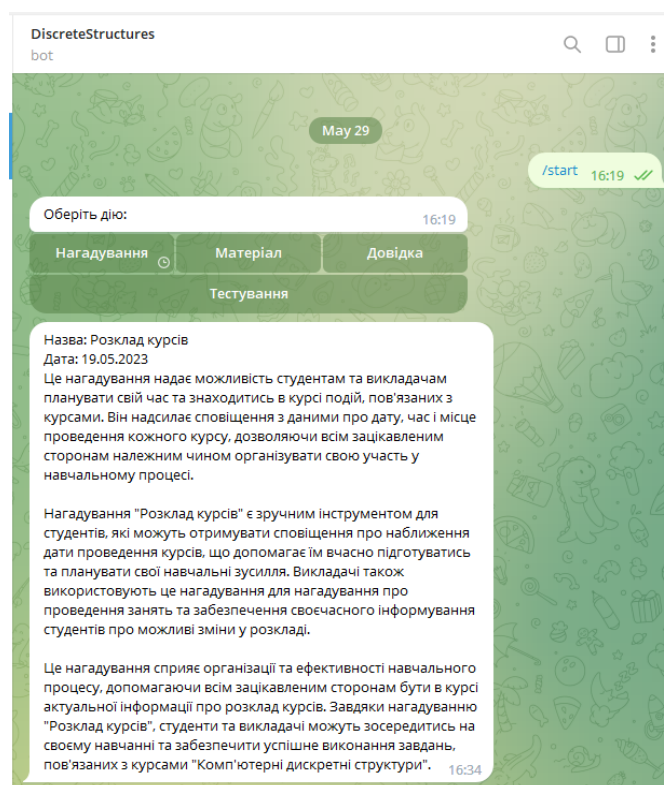


Рисунок 3.18 – Використання пункту меню «Нагадування»

Після натискання на пункт меню "Матеріал" користувач отримує доступ до матеріалів, пов'язаних з курсом "Комп'ютерні дискретні структури". Це можуть бути лекції, практичні завдання, посібники або інші навчальні ресурси (рис. 3.19).

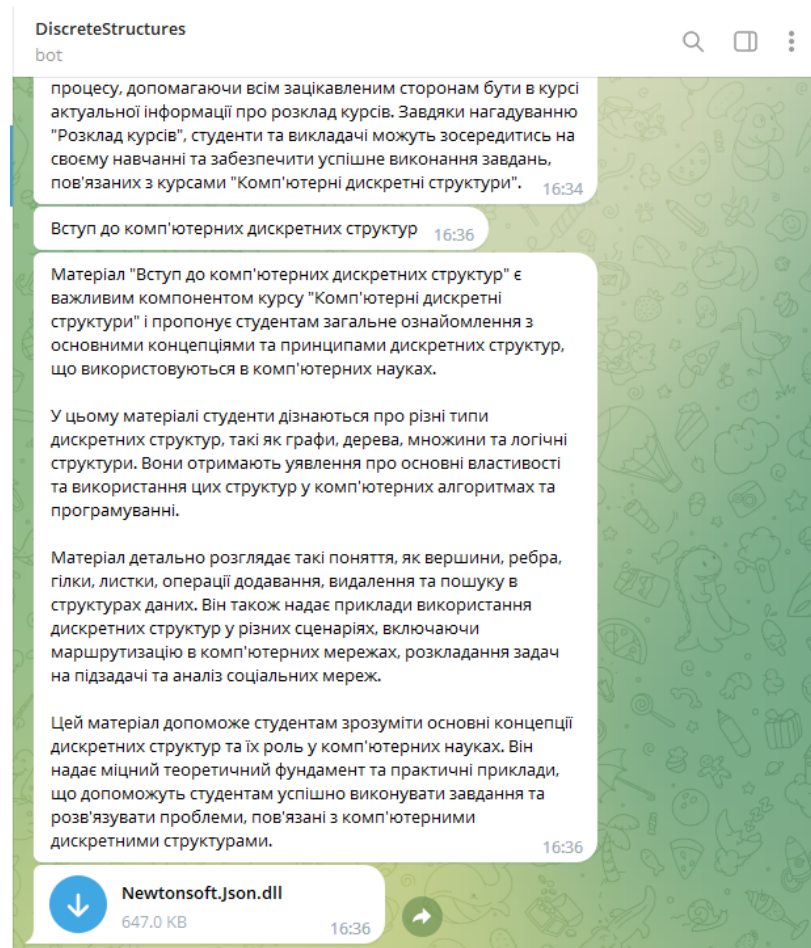


Рисунок 3.19 – Використання пункту меню «Матеріал»

Також, у чаті реалізована довідкова система. При натисканні на пункті меню «Довідка», користувач отримує доступ до довідкової інформації про призначення пунктів меню.

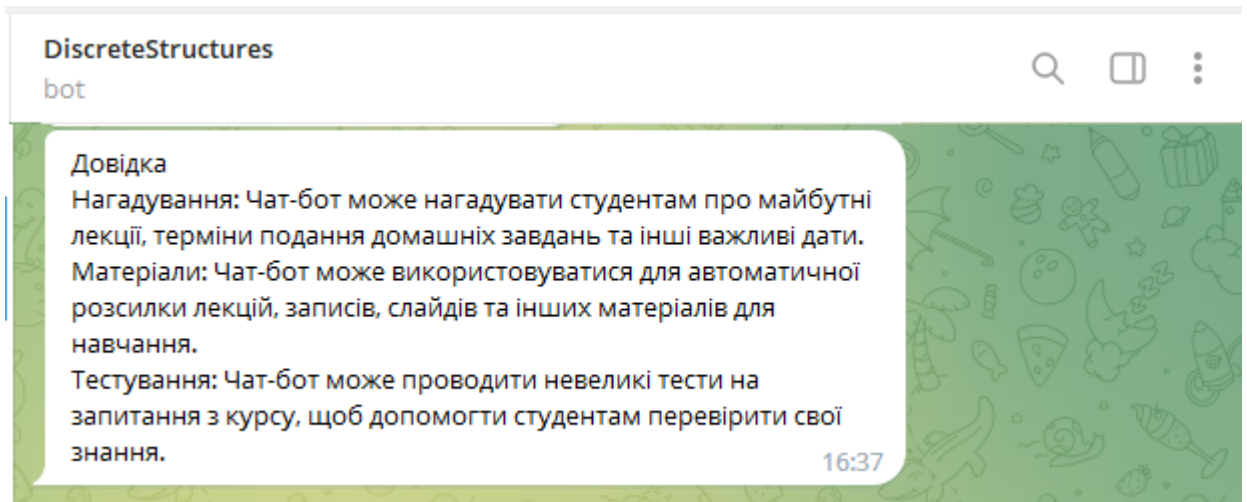


Рисунок 3.20 – Використання пункту меню «Довідка»

При натисканні на пункт меню "Тестування" користувач отримує доступ до функціоналу тестування з курсу "Комп'ютерні дискретні структури" (рис. 3.21). Бот може пропонувати користувачу запитання з тем, пов'язаних з курсом, і в кінці тестування виводити оцінку знань (рис. 3.22).

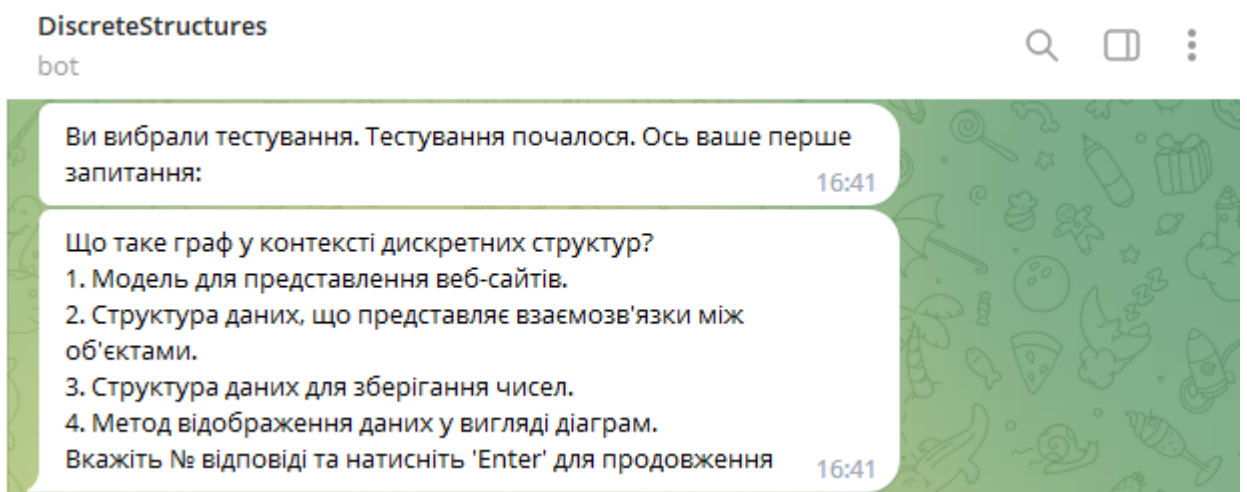


Рисунок 3.21 – Використання пункту меню «Тестування»

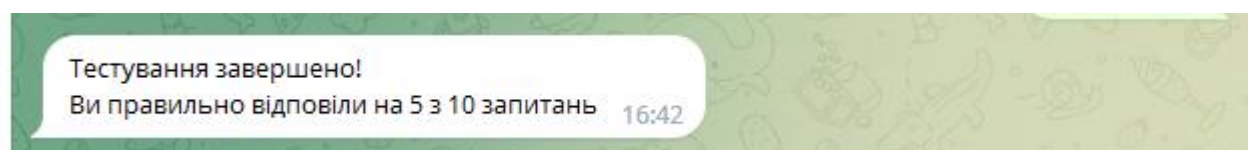


Рисунок 3.22 – Виведення результатів тестування

Треба сказати, що дана система є не сильно функціональною, але вона є досить простою в користуванні. Її інтерфейс є інтуїтивно зрозумілим для користувача.

ВИСНОВКИ

За результатами аналізу потреби в підтримці навчального процесу, було визначено, що чат-бот Telegram може слугувати ефективним засобом для цієї мети. Було досліджено бізнес-процеси для підтримки навчального процесу, включаючи використання існуючих чат-ботів. Було проаналізовано декілька сучасних програмних рішень, таких як MoodleBot, EduBots, та Botpress, що вже використовуються для підтримки навчального процесу. За результатами аналізу було визначено, що розробка спеціалізованого Telegram чат-боту для курсу "Комп'ютерні дискретні структури" може виявитися більш цілеспрямованою та ефективною.

Вибір технологій для розробки чат-бота було виконано з урахуванням потужності, гнучкості та рівня підтримки, що надається спільнотою. Мова програмування C# та середовище Visual Studio були обрані внаслідок їх відмінних характеристик та широкого застосування в сфері розробки програмного забезпечення. Вони дозволяють швидко та ефективно реалізувати необхідну функціональність і водночас забезпечують високу безпеку та стабільність роботи програмного продукту.

Архітектура програмного забезпечення була розроблена з урахуванням специфічних вимог до чат-бота. Вона включає BLL (Business Logic Layer) для виконання основних бізнес-операцій, UI (User Interface Layer) для взаємодії з користувачами, та DAL (Data Access Layer) для взаємодії з базою даних. Така структура дозволяє забезпечити гнучкість, ефективність та легке супроводження системи.

У процесі розробки програмного забезпечення було проєктовано базу даних (MS Access), що дозволяє зберігати та обробляти необхідну інформацію. Розроблено програмні модулі системи та проведено їх тестування. Виконано функціональне та модульне тестування за допомогою Unit Test для забезпечення якості розробленого програмного забезпечення.

Розроблено інструкцію користувача для підтримки навчального процесу, що включає детальні пояснення та рекомендації щодо використання чат-бота. Це забезпечує комфортне використання чат-бота як викладачами, так і студентами.

Проведена робота демонструє значний потенціал Telegram чат-бота для підтримки навчального процесу. Далі планується впровадження чат-бота в реальне навчальне середовище, збір відгуків від користувачів для подальшого вдосконалення системи, а також розширення функціональності бота для підтримки інших курсів або навчальних модулів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Курлянд З.Н., Осипова Т.Ю., Гурін Р.С. Теорія і методика професійної освіти: навч. посіб. Київ, 2012. 390 с.
2. Стинська В. Теорія і практика вищої професійної освіти в Україні: навч.-метод. посіб. Івано-Франківськ, 2021. 250 с.
3. Зайченко І. В. Теорія і методика професійного навчання. 2-е вид., доповн. і переробл. Київ: Видавництво Ліра-К, 2016. 580 с.
4. Теорія і практика вищої професійної освіти: навч. посіб. / уклад.: М. О. Давидюк. Вінниця, 2015. 94 с.
5. MoodleBot [Електронний ресурс] – Режим доступу: <https://moodle.org/course/view.php?id=5#section-0>
6. EduBots [Електронний ресурс] – Режим доступу: <https://devpost.com/software/edubot-wxgqeб>
7. Botpress [Електронний ресурс] – Режим доступу: <https://botpress.com/>
8. MATT WATSON. What is C# used for? [Електронний ресурс] / MATT WATSON // STACKIFY PRODUCT & COMPANY UPDATES. – 2020. – Режим доступу до ресурсу: <https://stackify.com/what-is-c-used-for/> .
9. C++ [Електронний ресурс] – Режим доступу: http://www.znannya.org/?view=Cpp_basics
10. Sun Microsystems [Електронний ресурс] – Режим доступу: <https://www.britannica.com/topic/Sun-Microsystems-Inc>
11. Java [Електронний ресурс] – Режим доступу: <https://dou.ua/lenta/articles/how-to-learn-java/>
12. Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visualstudio.com/> .
13. Eclipse [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eclipse.org/ide/>
14. PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/idea/pycharm-pro/>

15. IDE [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/uk-ua/visualstudio/extensibility/internals/ide-defined-commands-for-extending-project-systems?view=vs-2019>
16. Creating a Business Logic Layer [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/aspnet/web-forms/overview/data-access/introduction/creating-a-business-logic-layer-cs>
17. UML examples and algorithms [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <http://www.omg-portal.ru/articlematerial17> (дата звернення 01.05.2018)
18. Creating a Data Access Layer [Електронний ресурс]. – 2020. – Режим доступу до ресурсу <https://learn.microsoft.com/en-us/aspnet/web-forms/overview/data-access/introduction/creating-a-data-access-layer-cs>
19. Perkins B. What is ERP? Key features of top enterprise resource planning systems [Електронний ресурс] / Bart Perkins. – 2020. – Режим доступу до ресурсу: <https://www.cio.com/article/2439502/what-is-erp-key-features-of-top-enterprise-resourceplanning-systems.html>
20. Visual Studio Code [Електронний ресурс]. — Режим доступу: <https://code.visualstudio.com/>

ДОДАТОК А

Лістинг 1. Код класу «TelegramBotForm»

```

using AccountingBoat.AppCode;
using DiscreteStructApp.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Telegram.Bot;
using Telegram.Bot.Args;
using Telegram.Bot.Types.InputFiles;
using Telegram.Bot.Types.ReplyMarkups;

namespace DiscreteStructApp.Forms.Control {
    public partial class TelegramBotForm : Form {

        //Name: DiscreteStructures
        //UserName: DiscreteStructures_bot
        private ITelegramBotClient botClient;
        private string _Token = "6173079810:AAF1pwmmWddrkoP-ixRRBhj6DF2LOXWDFD0";
        private Dictionary<long, UserTestState> userStates = new Dictionary<long, UserTestState>();

        private MaterialsProvider _MaterialsProvider = new MaterialsProvider();
        private Materials _selectedMaterials = new Materials();
        private RemindersProvider _RemindersProvider = new RemindersProvider();
        private Reminders _selectedReminders = new Reminders();
        private ValidationMy _validation = new ValidationMy();

        private TestProvider _TestProvider = new TestProvider();
        private List<Test> _TestList = new List<Test>();

        public TelegramBotForm() {
            InitializeComponent();
            LoadDate();
        }

        private void LoadDate() {
            _selectedMaterials = _MaterialsProvider.SelectedMaterialsByMaterialsId(1);
            _selectedReminders = _RemindersProvider.SelectedRemindersByRemindersId(1);
            _TestList = _TestProvider.GetAllTest();
        }
    }
}

```

```

private void TelegramBotForm_Load(object sender, EventArgs e) {
    botClient = new TelegramBotClient(_Token);
    botClient.OnMessage += Bot_OnMessage;
    botClient.OnCallbackQuery += Bot_OnCallbackQuery;
    botClient.StartReceiving();
    RaportTextBox.Text = "Запущено ТГ бота\r\n";
}

async void Bot_OnMessage(object sender, MessageEventArgs e) {
    if (e.Message.Text != null) {
        // Updating the TextBox from the UI thread
        RaportTextBox.Invoke((MethodInvoker)delegate {
            // Set the new value to TextBox
            RaportTextBox.Text += e.Message.Chat.Id.ToString() + "\r\n";
            RaportTextBox.Text += e.Message.Text + "\r\n";
        });

        if (!userStates.ContainsKey(e.Message.Chat.Id)) {
            userStates[e.Message.Chat.Id] = new UserTestState();
        }

        UserTestState userState = userStates[e.Message.Chat.Id];
        if (userState.InTest) { // Process the user's answer
            userState.QuestionIndex++;
            if (userState.QuestionIndex >= _TestList.Count) // assuming the test has Count questions
            {
                userState.InTest = false;
                string testComplite = "Тестування завершено!" + "\r\n";
                testComplite += "Ви правильно відповіли на " + userStates[e.Message.Chat.Id].Rating + " з "
+ _TestList.Count + " запитань" + "\r\n";
                await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: testComplite);
            } else {

                if (_validation.IsDataConvertToInt(e.Message.Text)) {

                    RightAnswer(e.Message.Chat.Id, Convert.ToInt32(e.Message.Text), userState.QuestionIndex
- 1);

                    string textQuestion = _TestList[userState.QuestionIndex].QuestionName + "\r\n";
                    textQuestion += "1. " + _TestList[userState.QuestionIndex].AnswerA + "\r\n";
                    textQuestion += "2. " + _TestList[userState.QuestionIndex].AnswerB + "\r\n";
                    textQuestion += "3. " + _TestList[userState.QuestionIndex].AnswerC + "\r\n";
                    textQuestion += "4. " + _TestList[userState.QuestionIndex].AnswerD + "\r\n";
                    textQuestion += "Вкажіть № відповіді та натисніть 'Enter' для продовження";

                    await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: textQuestion);
                } else {
                    await botClient.SendTextMessageAsync(chatId: e.Message.Chat, text: "Будь-ласка введіть
число у діапазоні від 1 до 4!");
                }
            }
        }
    }
}

```



```

    } else if (e.Message.Text == "/start") {
        await SendMainMenu(e.Message.Chat.Id);
    } else {
        // handle non-test messages here
    }
}
}
}

```

```

async void Bot_OnCallbackQuery(object sender, CallbackQueryEventArgs e) {
    switch (e.CallbackQuery.Data) {
        case "reminders":
            await StartReminderProcess(e.CallbackQuery.Message.Chat.Id);
            break;
        case "materials":
            await StartSendingMaterialsProcess(e.CallbackQuery.Message.Chat.Id);
            break;
        case "test":
            await StartTestingProcess(e.CallbackQuery.Message.Chat.Id);
            break;
        case "help":
            await StartHelpProcess(e.CallbackQuery.Message.Chat.Id);
            break;
    }
}
}

```

```

async System.Threading.Tasks.Task SendMainMenu(long chatId) {
    var inlineKeyboard = new InlineKeyboardMarkup(new[] {
        // First row
        new [] {
            InlineKeyboardButton.WithCallbackData("Нагадування", "reminders"),
            InlineKeyboardButton.WithCallbackData("Матеріал", "materials"),
            InlineKeyboardButton.WithCallbackData("Довідка", "help")
        },
        // Second row
        new [] {InlineKeyboardButton.WithCallbackData("Тестування", "test")});
    await botClient.SendTextMessageAsync(chatId: chatId, text: "Оберіть дію:", replyMarkup:
inlineKeyboard );
}

```

```

async System.Threading.Tasks.Task StartReminderProcess(long chatId) {
    // Your logic here for Reminders
    string remindersText = "Назва: " + _selectedReminders.RemindersName + "\r\n";
    remindersText += "Дата: " + _selectedReminders.RemindersDate.ToShortDateString() + "\r\n";
    remindersText += _selectedReminders.Description;

    await botClient.SendTextMessageAsync(chatId: chatId, text: remindersText);
}

```

```

async System.Threading.Tasks.Task StartHelpProcess(long chatId) {
    // Довідка
    string sendText = "Довідка" + "\r\n";
}

```

```

    sendText += "Нагадування: Чат-бот може нагадувати студентам про майбутні лекції, терміни подання домашніх завдань та інші важливі дати." + "\r\n";
    sendText += "Матеріали: Чат-бот може використовуватися для автоматичної розсилки лекцій, записів, слайдів та інших матеріалів для навчання." + "\r\n";
    sendText += "Тестування: Чат-бот може проводити невеликі тести на запитання з курсу, щоб допомогти студентам перевірити свої знання." + "\r\n";
    await botClient.SendTextMessageAsync(chatId: chatId, text: sendText);
}

```

```

async System.Threading.Tasks.Task StartSendingMaterialsProcess(long chatId) {
    // Відправка матеріалів курсів
    await botClient.SendTextMessageAsync(chatId: chatId, text: _selectedMaterials.MaterialsName);
    await botClient.SendTextMessageAsync(chatId: chatId, text: _selectedMaterials.Description);
    SendFile(chatId, _selectedMaterials.MaterialsPath);
}

```

```

async System.Threading.Tasks.Task StartTestingProcess(long chatId) {
    // logic here for Testing
    if (!userStates.ContainsKey(chatId)) {
        userStates[chatId] = new UserTestState();
    }
}

```

```

UserTestState userState = userStates[chatId];
userState.InTest = true;
userState.QuestionIndex = 0;

```

```

await botClient.SendTextMessageAsync(chatId: chatId, text: "Ви вибрали тестування. Тестування почалося. Ось ваше перше запитання:");

```

```

string textQuestion = _TestList[userState.QuestionIndex].QuestionName + "\r\n";
textQuestion += "1. " + _TestList[userState.QuestionIndex].AnswerA + "\r\n";
textQuestion += "2. " + _TestList[userState.QuestionIndex].AnswerB + "\r\n";
textQuestion += "3. " + _TestList[userState.QuestionIndex].AnswerC + "\r\n";
textQuestion += "4. " + _TestList[userState.QuestionIndex].AnswerD + "\r\n";
textQuestion += "Вкажіть № відповіді та натисніть 'Enter' для продовження";

```

```

await botClient.SendTextMessageAsync(chatId: chatId, text: textQuestion);

```

```

}

```

```

async private void RightAnswer(long chatId, int answer, int questionId) {
    switch (answer) {
        case 1:
            if (_TestList[questionId].AnswerABool) {
                userStates[chatId].Rating++;
            }
            break;
        case 2:
            if (_TestList[questionId].AnswerBBool) {
                userStates[chatId].Rating++;
            }
    }
}

```

```

        break;
    case 3:
        if (_TestList[questionId].AnswerCBool) {
            userStates[chatId].Rating++;
        }
        break;
    case 4:
        if (_TestList[questionId].AnswerDBool) {
            userStates[chatId].Rating++;
        }
        break;
    default:
        await botClient.SendTextMessageAsync(chatId: chatId, text: "Введене число виходить за
межі допустимого діапазону (1-4)");
        break;
    }
}

```

```

async Task SendFile(long chatId, string filePath) {
    using (var fileStream = new FileStream(filePath, FileMode.Open, FileAccess.Read,
FileShare.Read)) {
        var fileName = Path.GetFileName(filePath);
        await botClient.SendDocumentAsync(chatId, new InputOnlineFile(fileStream, fileName));
    }
}
}
}

```

```

public class UserTestState {
    public bool InTest { get; set; }
    public int QuestionIndex { get; set; }

    public int Rating { get; set; }
}

```

ЛІСТИНГ 2. Код класу «MaterialsForm»

```

using AccountingBoat.AppCode;
using DiscreteStructApp.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiscreteStructApp.Forms.Dictionary {
    public partial class MaterialsForm : Form {
        private Materials _selectedMaterials = new Materials();
    }
}

```

```

private MaterialsProvider _MaterialsProvider = new MaterialsProvider();
private ValidationMy _Validation = new ValidationMy();
public MaterialsForm() {
    InitializeComponent();
    LoadAllDate();
}

private void OpenBtn_Click(object sender, EventArgs e) {
    OpenFileDialog openFileDialog = new OpenFileDialog();

    openFileDialog.Filter = "All files (*.*)|*.*";
    openFileDialog.FilterIndex = 2;
    openFileDialog.RestoreDirectory = true;
    if (openFileDialog.ShowDialog() == DialogResult.OK) {
        try {
            MaterialsPathTBox.Text = openFileDialog.FileName;
        } catch (Exception ex) {
            MessageBox.Show(ex.Message);
        }
    }
}

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        _MaterialsProvider.UpdateMaterials(MaterialsNameTBox.Text, MaterialsPathTBox.Text,
        DescriptionTBox.Text, 1);
        MessageBox.Show("Дані успішно збережено!");
        this.Close();
    }
}

private void ExitBtn_Click(object sender, EventArgs e) {
    this.Close();
}

private void LoadAllDate() {
    _selectedMaterials = _MaterialsProvider.SelectedMaterialsByMaterialsId(1);
    MaterialsNameTBox.Text = _selectedMaterials.MaterialsName;
    MaterialsPathTBox.Text = _selectedMaterials.MaterialsPath;
    DescriptionTBox.Text = _selectedMaterials.Description;
}

private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (_Validation.IsDataEntering(MaterialsNameTBox.Text)) {
        MaterialsNameValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        MaterialsNameValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataEntering(MaterialsPathTBox.Text)) {

```

```

        MaterialsPathValidtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        MaterialsPathValidtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }

    return isCorrect;
}

}
}

```

ЛІСТИНГ 3. Код класу «RemindersForm»

```

using AccountingBoat.AppCode;
using DiscreteStructApp.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DiscreteStructApp.Forms.Dictionary {
    public partial class MaterialsForm : Form {
        private Materials _selectedMaterials = new Materials();
        private MaterialsProvider _MaterialsProvider = new MaterialsProvider();
        private ValidationMy _Validation = new ValidationMy();
        public MaterialsForm() {
            InitializeComponent();
            LoadAllDate();
        }

        private void OpenBtn_Click(object sender, EventArgs e) {
            OpenFileDialog openFileDialog = new OpenFileDialog();

            openFileDialog.Filter = "All files (*.*)|*.*";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK) {
                try {
                    MaterialsPathTBox.Text = openFileDialog.FileName;
                } catch (Exception ex) {
                    MessageBox.Show(ex.Message);
                }
            }
        }

        private void SaveBtn_Click(object sender, EventArgs e) {
            if (IsDataEnteringCorrect()) {

```

```

        _MaterialsProvider.UpdateMaterials(MaterialsNameTBox.Text, MaterialsPathTBox.Text,
DescriptionTBox.Text, 1);
        MessageBox.Show("Дані успішно збережено!");
        this.Close();
    }
}

private void ExitBtn_Click(object sender, EventArgs e) {
    this.Close();
}

private void LoadAllDate() {
    _selectedMaterials = _MaterialsProvider.SelectedMaterialsByMaterialsId(1);
    MaterialsNameTBox.Text = _selectedMaterials.MaterialsName;
    MaterialsPathTBox.Text = _selectedMaterials.MaterialsPath;
    DescriptionTBox.Text = _selectedMaterials.Description;
}

private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (_Validation.IsDataEntering(MaterialsNameTBox.Text)) {
        MaterialsNameValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        MaterialsNameValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataEntering(MaterialsPathTBox.Text)) {
        MaterialsPathValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        MaterialsPathValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }

    return isCorrect;
}
}
}

```

ДОДАТОК Б



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА TELEGRAM ЧАТ-БОТУ ДЛЯ ПІДТРИМКИ НАВЧАЛЬНОГО ПРОЦЕСУ НА КУРСІ "КОМП'ЮТЕРНІ ДИСКРЕТНІ СТРУКТУРИ" МОВОЮ С#

Виконав студент 4 курсу

Групи ПД-44

Утешов Владислав Сергійович

Керівник роботи

К. ф.м. наук, доцент Садовенко В.С

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** - підтримка навчального процесу, за допомогою Telegram чат боту
- **Об'єкт дослідження** – використання Telegram чат-боту для підтримки навчального процесу на курсі "Комп'ютерні дискретні структури".
- **Предмет дослідження** - Telegram чат-бот, розроблений мовою С# для підтримки навчального процесу.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Аналіз існуючих Telegram чат-ботів для підтримки навчального процесу.
2. Розглянути актуальність чат-ботів.
3. Розглянути ресурси, для розробки чат-боту.
4. Сформулювати вимоги до чат-боту на основі аналізу аналогів.
5. Розробити Telegram чат-бот для підтримки навчального процесу мовою C#.
6. Провести тестування.

3

АНАЛІЗ АНАЛОГІВ

Основні характеристики	<u>MoodleBot</u>	<u>EduBot</u>	<u>BotPress</u>	Розроблений бот
Нагадування	-	+	+	+
Розсилка	-	-	+	+
Тестування та відповіді на запитання	+	+	+	+
Керування обліковим записом	+	+	-	+
Налаштування	+	+	+	+

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Нагадування про терміни.
2. Розсилка навчальних матеріалів.
3. Тестування та відповіді на запитання.
4. Керування обліковими записами користувачів.
5. Перегляд довідкових матеріалів.
6. Налаштування нагадувань.

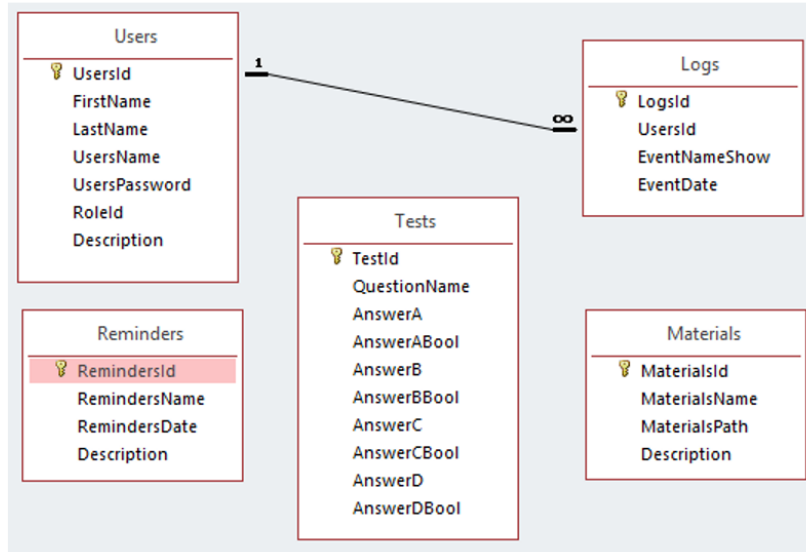
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



6

ДІАГРАМА БАЗ ДАНИХ



7

USE CASE ДІАГРАМА ДЛЯ РОЛІ «КОРИСТУВАЧ»

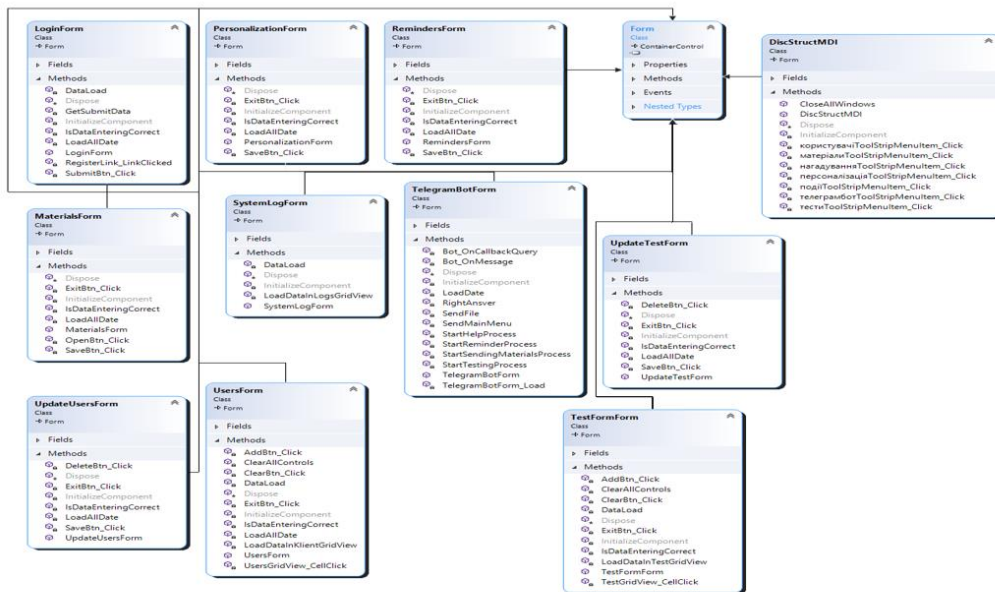


8

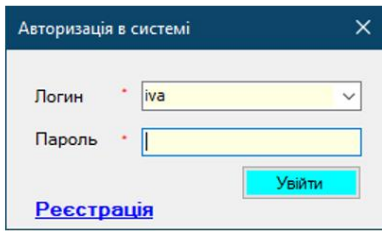
USE CASE ДІАГРАМА ДЛЯ РОЛІ «СИСТЕМНИЙ АДМІНІСТРАТОР»



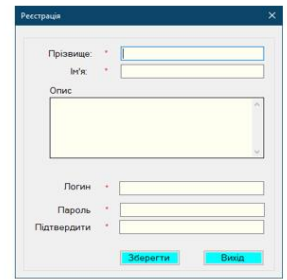
ДІАГРАМА КЛАСІВ ІНТЕРФЕЙСУ



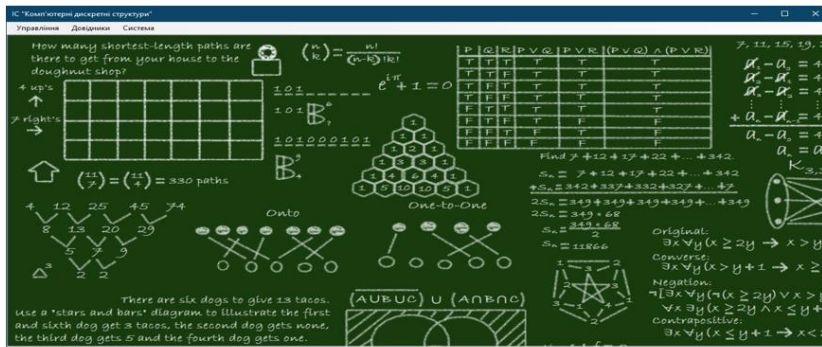
ЕКРАННІ ФОРМИ



Авторизація

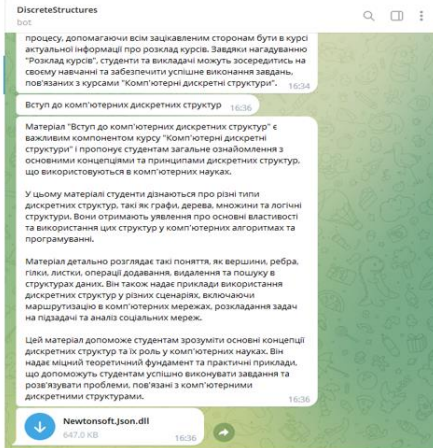


Реєстрація

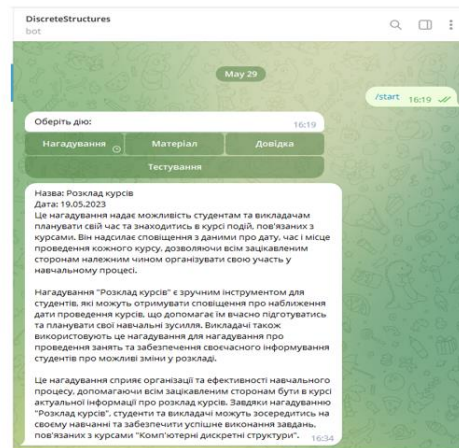


Головне меню

ЕКРАННІ ФОРМИ

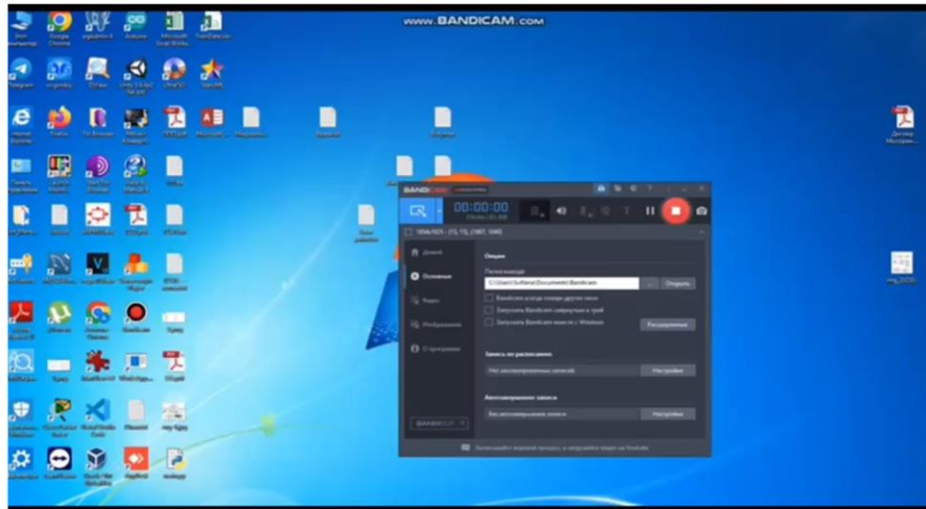


Використання пункту меню «Матеріал»



Використання пункту меню «Нагадування»

ПРИКЛАД ВИКОРИСТАННЯ



13

ВИСНОВКИ

1. Досліджено і проаналізовано існуючі Telegram чат-боти: [MoodleBot](#), [eduBot](#), [BotPress](#). Також було розглянуто основні механіки [геймплею](#) для створення відчуття динамічності.
2. Чат-боти мають значний потенціал для [підтримки навчального процесу](#) та можуть бути дуже [актуальними у сучасному освітньому середовищі](#).
3. Розглянуто ресурси, для розробки чат боту. Було розглянуто мови програмування C++, Java, C#. В результаті було обрано мову C# та середовище розробки Microsoft Visual Studioю.
4. Розроблено вимоги до чат боту на основі аналізу аналогів. Вимоги включають в себе наступні пункти: [нагадування](#), [розсилка](#), [тестування](#), [керування](#), [перегляд](#), [налаштування](#).
5. Розроблено Telegram чат-бот для підтримки навчального процесу за допомогою мови C# перевагами якого є: [нагадування про заняття](#), [розсилка інформації](#).
6. Проведено мануальне тестування додатку продемонструвало відсутність помилок та [невалідної поведінки](#)

14