

Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка програмного забезпечення Telegram-боту для сайту університету

Студента 4 курсу, 6 групи,
спеціальності «Інженерія
програмного забезпечення»

підпис студента

Дьякова Дениса
Юрійовича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Затверджую

Зав. кафедри інженерії
програмного
забезпечення
та кібербезпеки
Криворучко О.В.
"20" жовтня 2020 р.

Завдання на випускню кваліфікаційну роботу студентіві

Дьякову Денису Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи Розробка програмного
забезпечення Telegram-боту для сайту університету

Затверджена наказом ректора від "30" жовтня 2020 р. № 3225

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи

Мета роботи отримання повнофункціонального Telegram бота для сайту
університету з власною програмною реалізацією і логікою

Об'єкт дослідження процес навчання

Предмет дослідження процес розробки Telegram бота у середовищі Visual
Studio за допомогою мови програмування C#

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ
ДИПЛОМНОГО ПРОЕКТУ

1.1 Аналіз популярних месенджерів і визначення найбільш підходящого для
створення боту зі сповіщенням

1.2 Огляд платформ для написання програм-ботів

1.3 Загальний алгоритм реалізації проекту

1.4 Огляд можливостей месенджера Telegram

1.5 Конфіденційність та безпека

Висновки до розділу 1

РОЗДІЛ 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 UML Діаграми

2.2 База даних

2.3 Опис коду

Висновки до розділу 2

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ

3.1 Опис функціонування чат-боту

Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. «АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ»</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. «ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. «ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ»</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021– 21.05.2021	18.05.2021
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	24.05.2021	24.05.2021
11.	<i>Здача прожитої випускної кваліфікаційної роботи на кафедру</i>	25.05.2021	25.05.2021
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи

Цензура М.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М.О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Дьяков Д. Ю.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

Випускна кваліфікаційна робота студента Дьяков Д. Ю.
(прізвище, ініціали)

Гарант освітньої програми Цензура М. О.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« » 2021 p.

АНОТАЦІЯ

Дипломна робота на тему «Розробка програмного забезпечення Telegram-боту для сайту університету» містить 60 сторінок, 41 рисуноків, 3 таблиці, 12 додатків. Перелік посилань налічує 5 найменувань.

Метою дослідження є отримання повнофункціонального Telegram бота для сайту університету з власною програмною реалізацією і логікою.

Об'єктом дослідження є процес навчання.

Предметом дослідження є процес розробки Telegram бота у середовищі Visual Studio за допомогою мови програмування C#.

У першому розділі було проведено аналіз існуючих месенджерів, досліджено їх безпеку та конфіденційність, вибрано месенджер, обрано мову програмування.

У другому розділі була проаналізована та вибрано база даних, побудована система класів з використанням Telegram API, розроблені алгоритми обробки даних та створення pdf файлу з оцінками.

У третьому розділі подану інформацію про інтерфейс програми, описані та реалізовані команди, продемонстрований алгоритм взаємодії з Telegram-ботом.

ANNOTATION

The thesis on the topic "Development of Telegram bot software for the university website" contains 60 pages, 41 drawings, 3 tables, 12 appendix. The list of links includes 5 items.

The purpose of the study is to obtain a full-featured Telegram bot for the university website with its own software implementation and logic.

Research subject is the learning process.

The study's subject is a Telegram bot developing process using Visual Studio via the programming language C#.

The first section analyzes the existing popular messengers, the technical task, studied their safety and privacy, has chosen messenger and programming language.

The second section analyzes information about database, created class system with using Telegram API, describes the data processing and file building algorithms.

The third section provides information about the program interface, described and realized commands, demonstrated the interaction algorithm with the Telegram bot.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	6
1.1 Аналіз популярних месенджерів і визначення найбільш підходящого для створення боту зі сповіщенням.	6
1.2 Огляд платформ для написання програм-ботів	11
1.3 Загальний алгоритм реалізації проекту	14
1.4 Огляд можливостей месенджера Telegram.....	15
1.5 Конфіденційність та безпека.....	19
Висновки до розділу 1	23
РОЗДІЛ 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	24
2.1 UML Діаграми	24
2.2 База даних	33
2.3 Опис коду	36
Висновки до розділу 2	48
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ	49
3.1 Опис функціонування чат-боту	49
Висновки до розділу 3	57
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ.....	61

					<i>КНТЕУ 121 06-09.БР</i>			
Зм.	Арк.	№ докум	Підпис	Дата	Розробка програмного забезпечення Telegram- боту для сайту університету	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.				Зміст	2	60
Керівник		Цензура М.О.			Зміст	Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розроб		Дьяков Д.Ю.						

ВСТУП

На сьогоднішній день месенджери користуються великим попитом, це пов'язано зі зміною в області мобільного інтернету: високі швидкості, низька ціна і широке поширення смартфонів. Прогресивність месенджерів можна порівняти з явищем десятирічної давності – вибухом соціальних медіа. Уже зараз додатками для обміну повідомленнями користуються 2 мільярди людей. Саме тому месенджери все більше і більше замінюють телефонні розмови. Адже це набагато зручніше для обох сторін. Ми можемо відповісти тільки у зручний для нас час, подумати над відповіддю, або не відповідати зовсім. Це зберігає наші нерви, а найголовніше – наш час. Але прогрес не стоїть на місці, і тепер нашим співрозмовником, навіть, не обов'язково має бути людина, її може замінити так званий Чат-бот. Чат-бот - це помічник, який спілкується з користувачами за допомогою повідомлень і має безліч специфічних функцій. Чат-бота можна використовувати як для розсилки інформації, так і для її збору.

AI (Artificial intelligence) використовується в основному для скорочення витрат і підвищення якості послуг. Найбільш популярними напрямками є розпізнавання голосу і текстове спілкування за допомогою роботів. Як приклад останнього можна навести знамениті спливаючі віконця "консультантів" на багатьох сайтах. Звичайно, про якість таких консультацій можна посперечатися, але тенденція очевидна і від неї не збираються відмовлятися.

Прикладом автоматизації можна назвати величезну кількість ботів для зручної та швидкої взаємодії у власному кабінеті інтернет провайдера, прийому заявок на доставку їжі, замовлення столиків в ресторанах, розсилки реклами і багато іншого.

					<i>КНТЕУ 121 06-09.БР</i>				
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>					
Зав. Каф.		Криворучко О.В.			<i>Розробка програмного забезпечення Telegram-боту для сайту університету</i>		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Цензура М.О.					<i>В</i>	<i>3</i>	<i>60</i>
Гарант		Цензура М.О.			<i>Вступ</i>		<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
Розроб		Дьяков Д.Ю.							

Такі боти дозволяють збільшити прибуток компаній, так як бот може обробляти заявки з більшою швидкістю, ніж людина, і зменшити навантаження на робочий персонал.

Враховуючи позитивні сторони вказані вище, було прийнято рішення створити Чат-бота для більш зручної взаємодії з веб сайтом Київського національного торговельно-економічного університету.

Платформою для створення було обрано Telegram, адже це багатоплатформний месенджер з функціями VoIP, що дозволяє обмінюватися текстовими, голосовими та відео повідомленнями, стікерами та фотографіями, файлами багатьох форматів. Також можна здійснювати відео та аудіо-дзвінки, організовувати конференції і розраховані на багато користувачів групи і канали. Але найголовніше там є можливість створювати потрібних нам віртуальних співрозмовників.

Актуальність створення чат-бота є безперечною, адже це не тільки полегшить студентам взаємодію з веб сайтом університету, а й зробить цей процес більш швидким та зручним.

Метою випускної кваліфікаційної роботи є набуття нових навичок при розробці Telegram Чат ботів, вдосконалення знань HTTP протоколу, консолідація та отримання нових знань з мови програмування C#, а також підвищення навичок роботи з середовищем Visual Studio. Отримання повнофункціонального Telegram бота для сайту університету з власною програмною реалізацією і логікою.

Об'єкт дипломної роботи - процес навчання.

Предмет дослідження – процес розробки Telegram бота у середовищі Visual Studio за допомогою мови програмування C#.

Завдання дослідження:

- Опис складових функціоналу
- Аналіз інструментальних засобів
- Розробка алгоритмів
- Розробка Telegram бота
- Складання звіту

Зм.	Лист	№ докум	Підпис	Дата

КНТЕУ 121 06-09.БР

Лист

5

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

Месенджер (англ. Messenger – посланник, instant messaging – обмін миттєвими повідомленнями) – це програма для миттєвого обміну повідомленнями, яка встановлюється на комп'ютер, ноутбук, смартфон або планшет. Дозволяє обмінюватися між користувачами текстовими, голосовими повідомленнями, а також прикріплювати фото, відео, різні документи. Перші версії месенджерів могли обмінюватися тільки текстовими повідомленнями. Сучасні системи, крім перерахованих вище функцій, дозволяють здійснювати аудіо та відео дзвінки, створювати групові чати, відеоконференції, відправляти автоматичні повідомлення, зберігати дані в «хмарі» і багато іншого.

1.1 Аналіз популярних месенджерів і визначення найбільш підходящого для створення боту зі сповіщенням.

Найпопулярнішими меседжерами в Україні вважаються: Viber (з часткою 87%), Telegram (40%) та WhatsApp (31%). Розглянемо кожен більш докладно.

1.1.1 Viber

Рік створення: 2010

Кількість користувачів: 900 мільйонів

За допомогою цієї програми користувачі можуть:

- Створювати групові чати для до 200 осіб.

					<i>КНТЕУ 121 06-09.БР</i>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. Каф.	Криворучко О.В.				Розробка програмного забезпечення Telegram-боту для сайту університету	Стадія	Аркуш
Керівник	Цензура М.О.					P1	60
Гарант	Цензура М.О.				Аналіз існуючих рішень та обґрунтування теми дипломного проекту	Факультет інформаційних технологій, 4 курс, 6 група	
Розроб	Дьяков Д.Ю.						

- Підписуватися на публічні чати
- Здійснювати грошові перекази.
- Здійснювати умовно безкоштовні телефонні та відеодзвінки між користувачами Messenger
- Надсилати один одному необмежену кількість повідомлень, фотографій, документів, презентацій, аудіо- та графічних файлів;
- Дзвінки на стаціонарні та мобільні номери по всьому світу за мінімальними тарифами (послуга Viber Out);

Переваги:

- Простий і зрозумілий інтерфейс користувача;
- Велика кількість смайлів, наклейок та фонів чату;
- Відсутність офіційного розголосу;
- Наявність власної ігрової платформи;
- Можливість спілкування під час відкритих тематичних дискусій.

Недолік:

- Проблеми з безпекою виникають час від часу.
- Тривале завантаження ігор;
- Багато спаму.

Viber для бізнесу

Ця популярна послуга обміну повідомленнями є більш ефективною у малому та середньому бізнесі, але її також успішно використовують великі бренди. Наприклад, для підвищення лояльності та помітності бренду вони розробляють корпоративні наклейки спеціально для Viber.

Такі великі компанії, як Fanta та Sprite, музичний агрегатор ELLO та журнал MAXIM створили громадські котеджі, на які кожен може підписатися.

Крім того, Viber конкурує зі Skype за ціни на міжнародні телефонні дзвінки до клієнтів.

1.1.3. WhatsApp

Рік заснування: 2009

1,5 мільярда користувачів.

WhatsApp Messenger - універсальний Додаток для смартфонів, який працює на всіх мобільних платформах. Програма здійснює пошук у телефонній книзі та додає тих, хто вже працює з WhatsApp Messenger, до списку контактів.

За допомогою цієї програми користувачі можуть:

- Обмінюватися повідомленнями та робити умовно безкоштовні дзвінки;
- Синхронізувати обмін повідомленнями з комп'ютером (скануючи QR-код на екрані комп'ютера за допомогою смартфона).
- Створити групові чати для до 256 осіб.
- Надсилати необмежену кількість текстових та голосових повідомлень, фотографій, документів, аудіо- та відеофайлів, місцезнаходження та історії чату.
- Доступ до повідомлень, надісланих, коли програма була офлайн.

Переваги:

- Швидкість;
- Простота;
- Популярність.

Недоліки:

- Відсутність ігрової платформи;
- Час від часу виникають проблеми з безпекою
- При відправленні якості медіафайлів знижується

WhatsApp для бізнесу

WhatsApp частіше використовується в малому та середньому бізнесі. Месенджер можна використовувати всередині компанії (для організації відеоконференцій та вебінарів) та поза нею. Можете використовувати додаток для підтримки клієнтів та просування брендових продуктів. Згідно з опитуванням, потенційні клієнти на 40% активніше взаємодіють з бізнесом через WhatsApp Mobile Messenger, ніж за допомогою дзвінків. Користувачам простіше швидко відправити миттєві повідомлення, ніж зателефонувати на гарячу лінію, яка часто перевантажена великою кількістю вхідних повідомлень. Постачальники можуть надсилати рекламу, повідомляти про створення нового товару та вести комунікації з ЦА.

1.1.4. Telegram

Рік створення дійсний до: 2013

2 мільярди користувачів.

Telegram - це месенджер, який дозволяє безкоштовно ділитися текстовими, аудіо- та графічними файлами та телефонувати іншим користувачам програми.

Переваги:

- Безпека;
- Відсутність реклами;
- Таємні чати та повідомлення, що самовидаляються;
- Створення та обслуговування розумних ботів;
- Швидкість і простота;
- Наявність власної бази даних стікерів (наклейок);
- Швидкий пошук повідомлень та таймерів для їх видалення;
- Шифрування повідомлень;

- Вдосконалена програма для редагування зображень;
- Автоматична синхронізація між пристроями.

Недоліки:

- Регулярні відключення через велике навантаження сервера;
- Відсутність відеодзвінків;

Telegram для бізнесу

Функціональність Telegram може бути ефективна для всіх типів бізнесу. Мессенджер має високий рівень конфіденційності і можливістю проводити банківські операції – все це може активно використовуватися фінансовими компаніями. Мессенджер забезпечить миттєву доставку інформації в службу підтримки. Telegram також корисний для великих компаній з великими потоками документів і великими базами даних, оскільки програма може зберігати великі обсяги інформації через хмарні сервіси.

Особливістю Telegram, яка пропонує великий потенціал для використання в компаніях, є підтримка розумних ботів. Це облікові записи, які автоматично обробляють повідомлення і відповідають на них. Функціонал ботів практично необмежений. Ви можете виконувати як прості, так і складні операції: від перегляду останніх новин, замовлення вечері в найближчій кав'ярні всього за кілька кліків до управління «розумним будинком».

Спеціальні API дозволяють стороннім розробникам створювати «боти», спеціальні запити, які можуть управляти програмою. Типові боти реагують на певні команди в особистих і групових чатах і можуть виконувати пошук в Інтернеті або виконувати інші завдання, які використовуються в розважальних або ділових цілях (Таблиця 1.1 – Основні характеристики).

Основні характеристики

Таблиця 1.1

Особливість	Telegram	Viber	WhatsApp
Швидкість відправки повідомлень	≈ 150 мілісекунд	≈ 900 мілісекунд	≈ 550 мілісекунд
Обсяг трафіку для відправки повідомлення	Мінімальний	Середній	Середній
Відмовостійкість та надійність	Висока	Середня	Середня
Вид синхронізації	Хмарна і миттєва	1 первинний пристрій (смартфон / планшет) і кілька вторинних (планшети / комп'ютери), які синхронізуються з первинним пристроєм	1 смартфон і 1 пов'язаний з ним комп'ютер
Завантаження вмісту	Динамічне в режимі реального часу, на вимогу	Одноразова доставка з сервера	Одноразова доставка з сервера

Ознайомившись з інформацією щодо кожного месенджера і виходячи з даних у таблицях 1.1 і 1.2 можна зробити такі висновки: враховуючи швидкість відправки повідомлення, високу надійність і відмовостійкість, миттєву хмарну синхронізацію, динамічне завантаження вмісту, простоту і багатоплановість у створенні ботів для розроблення боту моніторингу було обрано месенджер Telegram.

1.2 Огляд платформ для написання програм-ботів

Існує два способи написання ботів: за допомогою онлайн конструктора або використовуючи мову програмування.

1.2.1 За допомогою конструктора

В основі розробки за допомогою конструкторів лежить простий принцип: ви збираєте з готових цеглинок ту поведінку, яка вам потрібна. Якщо якоїсь цеглинки немає, то варіанти два: знайти інший конструктор або придумати, як можна обійтися без нього. У даного підходу є свої плюси і мінуси.

Плюси:

- Простота розробки
- Не потрібні навички програмування, або потрібні в мінімальній кількості

Мінуси:

- Функціонал обмежений платформою розробки
- Плата за використання, або обмежений функціонал з рекламою

Сервіси, які дозволяють створити Telegram бота без навичок програмування:

1. Manybot

Вартість: безкоштовно, але з рекламними повідомленнями.

Manybot – єдиний сервіс в нашій добірці, де немає візуального конструктора, а вся робота йде через телеграм-чат. Сервіс підійде тим, кому потрібні прості і передбачувані діалоги, вибір з невеликого числа варіантів і маленьке меню. Навряд чи він підійде для серйозних проектів, тому що простіше вивчити Python, ніж керувати складним ботом через команди в чаті. Платної версії немає, тому бот буде іноді відправляти свої рекламні повідомлення всім читачам, а відключити це не можна.

2. Bottap

Вартість: безкоштовна версія з рекламою або 150 € в місяць.

Мультиплатформовий конструктор допомагає створити боти, які можуть одночасно працювати в Telegram, Viber і у Skype.

На відміну від попереднього сервісу, тут є блоки, які можна перетягувати мишкою і налаштовувати зв'язку між ними.

З можливостей варто відзначити вбудовану CRM, необмежену глибину відповідей бота і форми зворотного зв'язку.

3. Botmother

Вартість: безкоштовно з обмеженнями, але без реклами, або 469 € в місяць.

Хороший сервіс для тих, кому потрібно багато функцій і детальна документація. Це вже повноцінний конструктор, де можна програмувати складну логіку. У сервісі є вбудовані шаблони для типових ситуацій і готові боти, які можна переробити під свої завдання. Ще можна відстежити кожен діалог з користувачем і втрутитися, якщо бот не справляється зі спілкуванням.

4. PuzzleBot

Вартість: безкоштовно з рекламними повідомленнями або від 210 € в місяць.

Потужний сервіс, де легко створити бота будь-якого рівня. Можна робити складні умови, налаштовувати тригери, автоматизувати збір статистики, передавати відповіді користувачів в інші сервіси і що завгодно ще.

Єдиний мінус - для роботи з сервісом все-таки потрібно знати основи програмування: що таке змінна, вкладені умови і цикли.

1.2.2 За допомогою мови програмування

В якості мови програмування було вибрано C#, так як дана мова є об'єктно-орієнтованою, має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML.

Для реалізації проекту було вирішено зупинити свій вибір на середовищі розробки Microsoft Visual Studio Community 2019, так як вона досить широко поширена, має безкоштовну версію для використання в освітніх цілях і до того ж підтримує кілька мов.

Модель даного середовища використовує такі поняття, як рішення (solution), проект, простір імен (namespace) і збірка (assembly). Поняття проекту присутня в багатьох середовищах, наприклад, в середовищі Delphi. Файл проекту містить перерахування вихідних файлів і інших ресурсів, з яких система буде будувати додаток. У рішення середовища Visual Studio можуть входити кілька проектів, які будуть залежними або незалежними один від одного в залежності від задач що вирішуються.

Інтегроване середовище розробки має велику кількість інструментів і функцій: може створювати як додатки в консолі, так і програми з графічним інтерфейсом, і навіть з допомогою технології Windows Forms. Microsoft Visual Studio хоча і працює з повним функціоналом відразу після установки, але при відмові одного з компонентів перестає працювати весь продукт. Зручний інтерфейс і легкість в розумінні роботи продукту.

Для реалізації клієнтської частини необхідно створення та налаштування бота в Telegram. Ця можливість реалізована всередині самого месенджера.

Так само для організації роботи програми з Telegram необхідна API, які дозволяють реалізовувати різні дії з цими сервісами. Для роботи з Telegram був обраний Telegram.Bot API, так як він підтримує мову програмування C#, і має безліч навчальних матеріалів.

1.3 Загальний алгоритм реалізації проекту

Перед написанням коду необхідно вивчити всю наявну документації по темі. Після навчання необхідно записати основний код бота. Далі йде підключення засадничих функцій бота, таких як авторизація у власному

кабінеті, реакція на повідомлення в Telegram. При коректній роботі основи починається період тестування наявних функцій і додавання нових. Коли бот буде повністю працездатний і протестований, починається етап апробування бота.

1.4 Огляд можливостей месенджера Telegram

Telegram - унікальний за своєю структурою месенджер, та є кросплатформним додатком. Крім стандартного обміну повідомленнями в діалогах і групах, в месенджері можна зберігати необмежену кількість файлів, вести канали (мікроблоги), створювати і використовувати боти. За допомогою спеціального API сторонні розробники можуть створювати боти, спеціальні акаунти, керовані програмами. Типові боти відповідають на спеціальні команди в персональних і групових чатах, також вони можуть здійснювати пошук в інтернеті або виконувати інші завдання.

Також реалізована функція використання socks5 проху, яка дозволяє використовувати месенджер навіть у важких для підключення умовах.

SOCKS 4/5 – це поширене позначення двох версій мережевого протоколу SOCKS4 і SOCKS5. Основною перевагою SOCKS (від. Англ. «Sockets» – гнізда) є можливість розробляти клієнт-серверні додатки за межами міжмережевого екранування, тобто SOCKS проксі може прийняти запит від клієнта, що знаходиться за файрволом, переглянути його права доступу і передати запит на зовнішній сервер.

Крім обміну повідомленнями, Telegram дозволяє створювати ботів. Telegram-боти - різновид чат-ботів. Їх суть полягає в розгалуженій реакції на певні повідомлення, які можуть взаємодіяти з користувачем, відповідати на їх повідомлення, команди і відправляти користувачеві у відповідь повідомлення.

Для початку треба визначимось, що таке чат-бот. Chatbot – це програма, яка дозволяє користувачам взаємодіяти зі сторонніми службами у вигляді добре відомого інтерфейсу чату. Основна мета чат-боту – допомогти

користувачам. Кожен чат-бот має свої специфічні функції, які виконуються після виклику спеціальних команд. Таким чином, ви можете швидко виконувати різні дії: перекладати, коментувати, знаходити, тестувати, досліджувати, навчати, транслювати, інтегрувати в інші сервіси та платформи, взаємодіяти з датчиками та об'єктами, підключеними до Інтернету.

Telegram Messenger був обраний платформою для створення чат-бота. Telegram – одна з найпопулярніших програм обміну повідомленнями. Окрім версій для мобільних операційних систем, існують версії для Windows, Linux та MacOS, існує також версія цього месенджера для браузера.

Повідомлення, команди і запити, надіслані користувачами, відправляються безпосередньо на сервер розробникам пошукової системи. Необхідно звернути увагу, що сервер Telegram є тільки посередником і надає розробникам свій API. Він доступний через інтерфейс HTTPS, який пропонує спрощену версію Telegram API, так званий «Bot API». Щоб взаємодіяти з ботом, користувач повинен спочатку додати його в свій чат або групу. Потім команди чату відправляються безпосередньо конкретному боту з використанням символу «@» перед його ім'ям в командному повідомленні. Все, що потрібно врахувати в цьому проекті, – це можливість додати бота в чат або групу. Потім він може відправляти туди повідомлення з відхиленнями від показників, що аналізуються.

Для початку необхідно визначитися з етапами створення нового бота Telegram. Щоб створити бота згідно з інструкціями на офіційному сайті Telegram API, необхідно виконати наступні дії.

а) Необхідно додати в чат основного телеграм-бота, якого можна знайти, виконавши пошук за запитом «@BotFather».

б) Наступним кроком буде робота з «@BotFather», для якого відправлена команда «/ newbot». Потім треба ввести ім'я. Потім він запросить коротку назву створеного бота (для швидкого відправлення команд).

Якщо збігів немає, формується ідентифікаційний токен, який потрібен при відправці запитів на телеграми API (Рис 1.1 – Реєстрація Telegram бота).

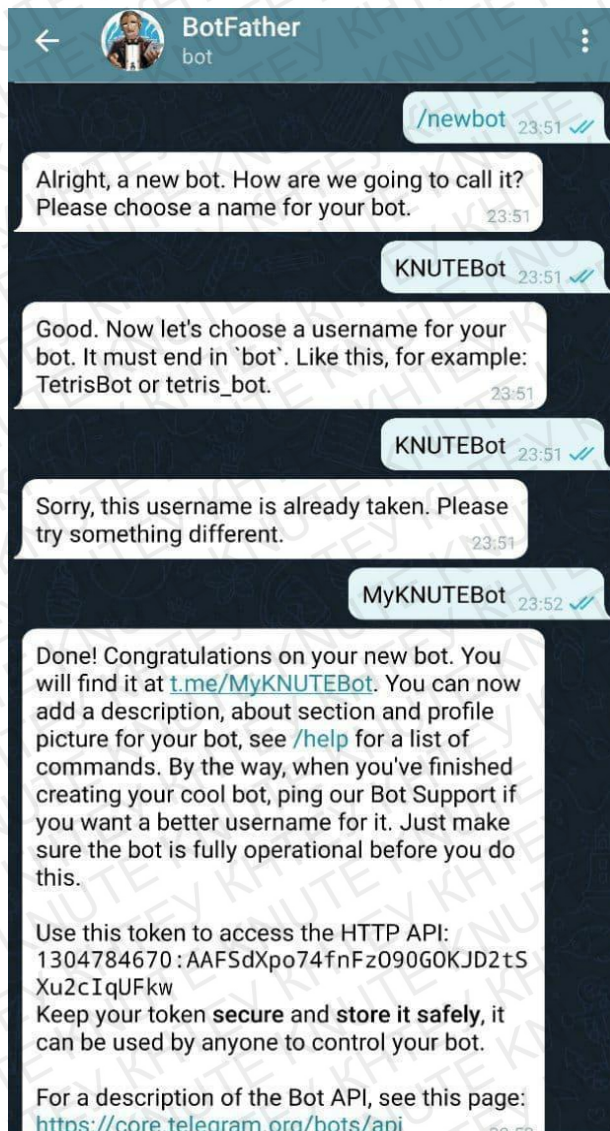


Рис 1.1 Реєстрація Telegram бота

Таблиця 1.2

Основні характеристики

Особливість	Telegram	Viber	WhatsApp
Основна концепція	Інтеграція месенджера з будь-якими сервісами, виконання найрізноманітніших задач в межах Telegram	Чатботи для оптимізації бізнесу та підтримки користувачів при взаємодії з публічними акаунтами	Не підтримується
Реалізація	Особливі Telegram-акаунти без номерів телефону, що контролюються програмами та відповідають чи звертаються до користувачів в рамках можливостей Bot API та фантазії розробників	Програми, що через інтерфейс Viber API відповідають користувачам, надаючи зокрема різноманітні бізнес пропозиції та пов'язану інформацію	
Обмеження на створення	Майже відсутні	Є певні вимоги до ботів	
Підтримувані типи вмісту	Всі (що підтримуються Telegram), зокрема місцезнаходження та номер	Текст, Зображення, відео, контакт, URL, каруселі, графічний зміст, а також місцезнаходження	
Комунікація через вбудований режим	Підтримується (відправка запитів і отримання результатів відповіді)	Обмежена реалізація через розширення	
Додавання ботів до інших чатів	Підтримується (до груп / каналів)	Не підтримується	
Варіанти використання	Отримання новин, інформації, повідомлень, покупка товарів і послуг, створення нових інструментів, ігор, соціальних сервісів, оптимізація користувацького досвіду (IV, попередній перегляд посилань), навчання, автоматизація, зворотній зв'язок, авторизація та безліч іншого	Отримання новин, інформації, реклами, зв'язок з публічним акаунтом, підтримка	

Типи робіт:

а) Роботи, які розуміють природну мову. Зазвичай такі програми базуються на нейронній мережі, яка логічно та структурно реагує на повідомлення користувачів.

б) Скриптові боти – це боти, які не розуміють природну мову і реагують лише на певний список команд. Діалоги з цими роботами часто є відкритими, зрозумілими та структурованими.

У нашому випадку краще використовувати скрипт робота.

1.5 Конфіденційність та безпека

Відразу два важливих пункти. Саме захист повідомлень від перехоплення і сторонніх очей стала основою популярності месенджера. Після множинних витоків інформації, виникає питання, чим би таким скористатися, щоб персональні дані не витекли в мережу. Тому важливо знати, що листи в Telegram надійно зашифровані.

1.5.1 Як влаштована безпека в телеграм

Якщо коротко, то безпеку в телеграм ґрунтується на криптографічному протоколі MTProto Proху. Особливості даного протоколу такі:

- Використання секретного ключа замість стандартної зв'язки "логін + пароль" (створення ключа відбувається в момент реєстрації клієнта і піддається надійному шифруванню);
- Використання двох видів шифрування: клієнт-сервер (для звичайного обміну повідомленнями) і клієнт-клієнт (для секретних чатів);
- Передача трафіку через канали телеграм, замаскована під стандартне TLS / SSL-з'єднання, але всередині дані зашифровані таким чином, що нічого неможливо розпізнавати;

MTProto Proxy - це спеціальний тип проксі, розроблений конкретно під месенджер телеграм, а це значить, що інші додатки на ньому не працюють, а звичайний інтернет-трафік через цей протокол не можна передавати.

1.5.2 Шифрування

Як було показано вище, телеграм використовує два види шифрування: "клієнт-сервер" і "клієнт-клієнт". Розглянемо докладніше, в чому їх призначення і їх відмінності.

1) Клієнт-сервер

Більшість месенджерів використовують найпоширеніший вид шифрування "клієнт-сервер". Алгоритм має такий вигляд:

- Спочатку клієнт генерує випадковий ключ для симетричного шифрування. Далі клієнт зашифровує цей ключ за допомогою асиметричного алгоритму, використовуючи відомий йому заздалегідь публічний ключ сервера.
- Зашифрований ключ відправляється сервера.
- Сервер отримує і розшифровує отриманий ключ і відповіддю повідомляє клієнту, що все добре.

Подальший трафік між сервером і клієнтом шифрується за допомогою обраного симетричного алгоритму.

2) Клієнт-клієнт

Даний вид шифрування називається крайовим або наскрізним, так як передача даних відбувається від пристрою відправника до пристрою користувача "крізь" сервер. Тобто вони передаються через сервер, але сам сервер не може розшифрувати ці дані, тому що ключі шифрування знаходяться тільки на пристроях користувачів, а самі дані зберігаються на сервері в зашифрованому вигляді. Такий вид шифрування гарантує що всі повідомлення, дзвінки, аудіофайли, відеофайли, зображення, документи та

інші дані захищені від попадання в треті руки - вони доступні тільки користувачам листування і є найбезпечнішими. Такий вид шифрування використовується в секретних чатах телеграм.

1.5.3 Хмарний пароль

Кожен раз входячи в телеграм з нового пристрою, Додаток вимагає від користувача введення одноразового коду підтвердження, який приходить на телефон користувача у вигляді SMS-повідомлення або повідомлення в телеграм.

Але якщо зловмисник має можливість перехопити SMS-повідомлення, то даний спосіб захисту не особливо допоможе. Тому рекомендується використовувати двоетапну перевірку, яку називають хмарним паролем. Це значно підвищує безпеку в телеграм.

Встановивши хмарний пароль, вхід в телеграм з нового пристрою стане більш безпечним, так як крім одноразового коду підтвердження доведеться ввести єдиний додатковий пароль, який відомий тільки власникові облікового запису телеграм. Також якщо треті особи спробують змінити даний пароль, на адресу електронної пошти користувача прийде повідомлення про зміну пароля, тим самим повідомляючи про підозрілих дії.

1.5.4 Код-пароль

Розробники також подбали про те, щоб ускладнити доступ зловмисників до листування користувача, якщо пристрій потрапив до них у руки. Встановивши код-пароль, доступ до чатах буде заблокований. Код-пароль може бути встановлений у вигляді ПІН-коду або звичайного пароля. Але його легко скинути, перевстановити додаток на пристрої.

Код-пароль дуже зручний для захисту листувань від цікавих очей колег або домочадців. Так що наявність коду-пароля не буде зайвою мірою безпеки в телеграм.

Тому настійно рекомендується використовувати опцію хмарного пароля, який завадить зловмисникові увійти в обліковий запис користувача телеграм.

Висновки до розділу 1

В наші дні прогрес не стоїть на місці, тому все більше і більше процесів стають автоматизованими. Наприклад, замовлення товару через інтернет, або отримання інформації щодо послуг інтернет провайдера. Тепер для досягнення таких цілей можна скористуватися чат-ботами. Провівши аналіз існуючих месенджерів, враховуючи такі фактори, як: популярність, зручність, надійність, а найголовніше можливість створювати власних чат-ботів – було вибрано месенджер Telegram. Мовою програмування було вибрано C#, оскільки ця мова є об'єктно-орієнтованою, має статичну типізацію, підтримує поліморфізм, перевантаження операторів (включаючи явні та неявні оператори типізації), делегатів, атрибути, події, властивості, узагальнені типи та методи, ітератори, анонімні функції, LINQ.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 UML Діаграми

Проект має клієнт-серверну архітектуру:

- Telegram є клієнтом, користувацьким інтерфейсом через який відбувається взаємодія з сервером.
- Сервер, на якому відбувається розрахунки та відправляється результат клієнту.

2.1.1 USE CASE

Use Case: 1 Зайти у персональний кабінет

ХАРАКТЕРИСТИЧНА ІНФОРМАЦІЯ

Мета в контексті: Додати у базу даних логін та пароль для подальшої роботи у власному кабінеті.

Сфера застосування: Сайт університету

Рівень: Telegram

Передумови: Ми знаємо email або номер перепустки та пароль користувача.

Успішна стадія завершення: Користувач записав свої дані для авторизації у базу даних.

Основний актор: Студент

Тригер: При першому запуску бота система просить користувача ввести email або номер перепустки та пароль.

					<i>КНТЕУ 121 06-09.БР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка програмного забезпечення Telegram-боту для сайту університету</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. Каф.		Криворучко О.В.				<i>P2</i>	<i>24</i>	<i>60</i>
Керівник		Цензура М.О.			<i>Проектування програмного забезпечення</i>	<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
Гарант		Цензура М.О.						
Розроб		Дьяков Д.Ю.						

Або студент натискає на кнопку «Перезайти у кабінет».

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ УСПІХУ

1. Студент вводить email або номер перепустки
2. Студент вводить пароль
3. Система вносить дані для авторизації у базу даних

ВІДПОВІДНА ІНФОРМАЦІЯ

Пріоритет: ТОП

Цільова ефективність: 5 секунд на введення та обробку інформації

Частота: 2 / рік

Канал для основного актора: будь-який пристрій, підтримуючий Telegram

Use Case: 2 Отримати оцінки за навчання

ХАРАКТЕРИСТИЧНА ІНФОРМАЦІЯ

Мета в контексті: Студент натискає на кнопку «Отримати оцінки», очікує отримати файл з оцінками у форматі PDF.

Сфера застосування: Сайт університету

Рівень: Telegram

Передумови: Ми знаємо email або номер перепустки та пароль користувача.

Успішна стадія завершення: Користувач отримав PDF файл з оцінками за модульний контроль, користувач зареєстрований на сайті університету, логін та пароль є правильними.

Невдале завершення стану: Невірний логін або пароль.

Основний актор: Студент

Тригер: Студент натискає на кнопку «Отримати оцінки»

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ УСПІХУ

1. Студент натискає на кнопку «Отримати оцінки»
2. Телеграм бот опрацьовує команду.
3. Перевіряємо чи є соокіє користувача у базі даних у форматі масиву байтів.

4. Десеаріалізуємо масив байтів у об'єкт cookie
5. Відправляємо запит на сайт університету разом з cookie користувача.
6. Отримуємо оцінки за модульний контроль у форматі JSON
7. Парсимо строку у динамічний об'єкт
8. Сортируємо значення по даті
9. Будуємо таблицю та розфарбовуємо поля
10. Створюємо PDF файл та записуємо туди таблицю.
11. Відправляємо файл користувачу через Telegram

Розширення

3а. Відправити запит до сайту Кнтеу для авторизації у власному кабінеті та отримати cookie.

3а1. Зберегти cookie у базі даних

ВІДПОВІДНА ІНФОРМАЦІЯ

Пріоритет: ТОП

Цільова ефективність: 2 секунди на очікування документа

Частота: 6 / місяць

Канал для основного актора: будь-який пристрій, підтримуючий Telegram

Use Case: 5 Лого

ХАРАКТЕРИСТИЧНА ІНФОРМАЦІЯ

Мета в контексті: Студент натискає на кнопку «Показати найкращий університет», телеграм бот опрацьовує команду та відправляє лого “КНТЕУ” що складається з емодзі.

Сфера застосування: Telegram

Рівень: Telegram

Успішна стадія завершення: Користувач отримав лого з емодзі.

Основний актор: Студент

Тригер: Студент натискає на кнопку «Показати найкращий університет»

					КНТЕУ 121 06-09.БР	Аркуш
Зм..	Аркуш	№ докум	Підпис	Дата		26

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ УСПІХУ

1. Студент натискає на кнопку «Показати найкращий університет».
2. Телеграм бот опрацьовує команду.
3. Бот відправляє лого “КНТЕУ” що складається з емодзі.

ВІДПОВІДНА ІНФОРМАЦІЯ

Пріоритет: Середній

Цільова ефективність: 1 секунда на очікування лого

Частота: 20 / місяць

Канал для основного актора: будь-який пристрій, підтримуючий Telegram

Use Case: 6 Допомога

ХАРАКТЕРИСТИЧНА ІНФОРМАЦІЯ

Мета в контексті: Студент натискає на кнопку «Допомога», очікує отримати докладний опис всіх доступних команд.

Сфера застосування: Telegram

Рівень: Telegram

Успішна стадія завершення: Користувач отримав докладний опис всіх доступних команд.

Основний актор: Студент

Тригер: Студент натискає на кнопку «Допомога»

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ УСПІХУ

1. Студент натискає на кнопку «Допомога».
2. Телеграм бот опрацьовує команду.
3. Бот відправляє докладний опис всіх доступних команд

4. ВІДПОВІДНА ІНФОРМАЦІЯ

Пріоритет: Середній

Цільова ефективність: 1 секунду на очікування опис

Канал для основного актора: будь-який пристрій, підтримуючий Telegram (Рис 2.1 – Діаграма сценаріїв використання)

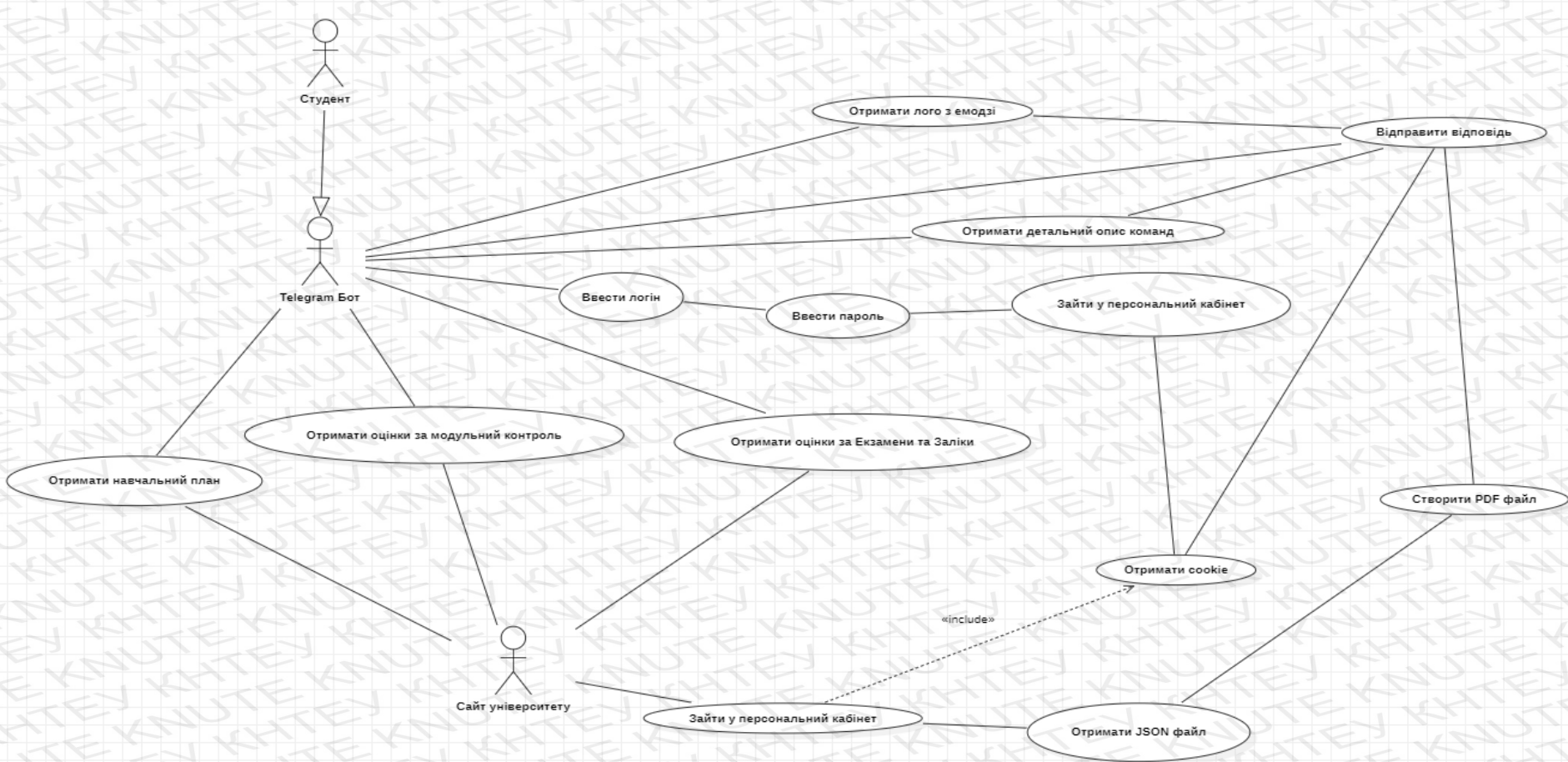


Рис. 2.1 – Діаграма сценаріїв використання

2.1.2 Діаграма Послідовностей

У разі, якщо користувач хоче зайти у персональний кабінет, то програма надішле повідомлення з проханням ввести логін і буде чекати відповіді. Після отримання відповіді надішлеться повідомлення з проханням ввести пароль. Після закінчення діалогу відбудеться запис до бази даних. Тепер після того, як користувач запросить свої оцінки, програма, спочатку перевірить наявність cookie у базі даних, і якщо їх не буде, або cookie будуть застарілі, то перед запитом отримання оцінок відправиться запит для авторизації у власному кабінеті (Рис 2.2 – Діаграма послідовностей частина 1)

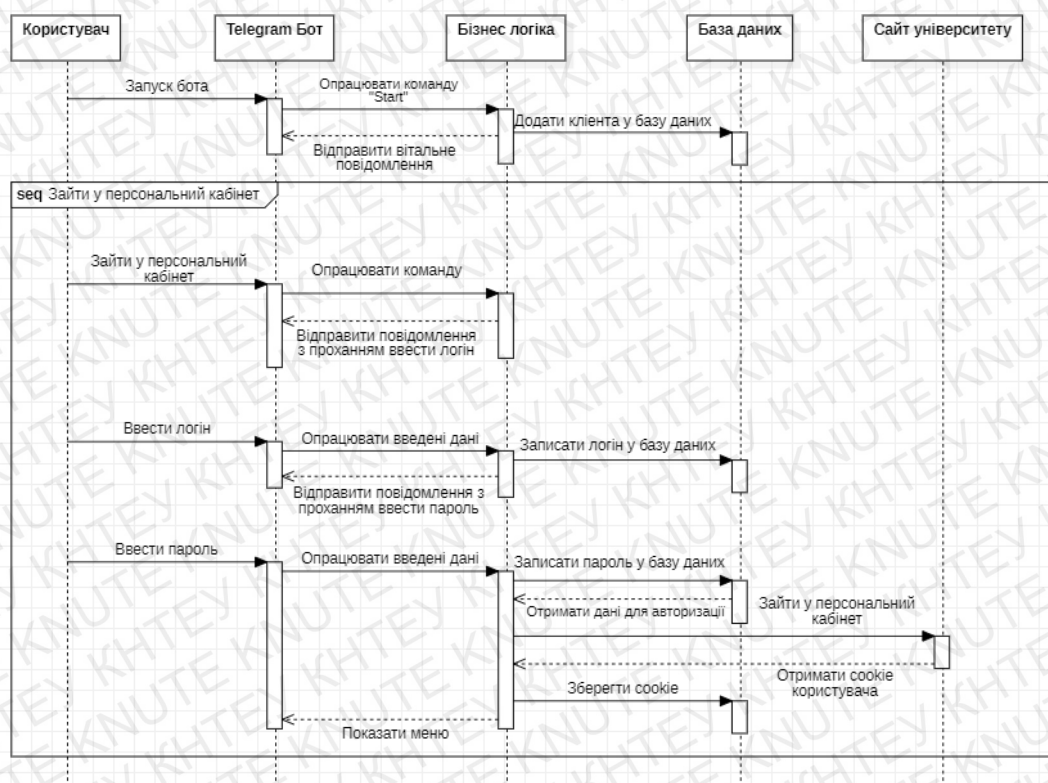


Рис. 2.2 – Діаграма послідовностей частина 1

Якщо користувач хоче отримати, відправляється запит до бази даних, щоб отримати збережені cookie користувача. Після їх отримання, передаємо ці дані методу для роботи з веб запитом та отримуємо відповідь у вигляді JSON. Після успішного отримання оцінок у форматі JSON, на їх основі створюється PDF файл та відправляється клієнту (Рис. 2.3 – Діаграма послідовностей частина 2).

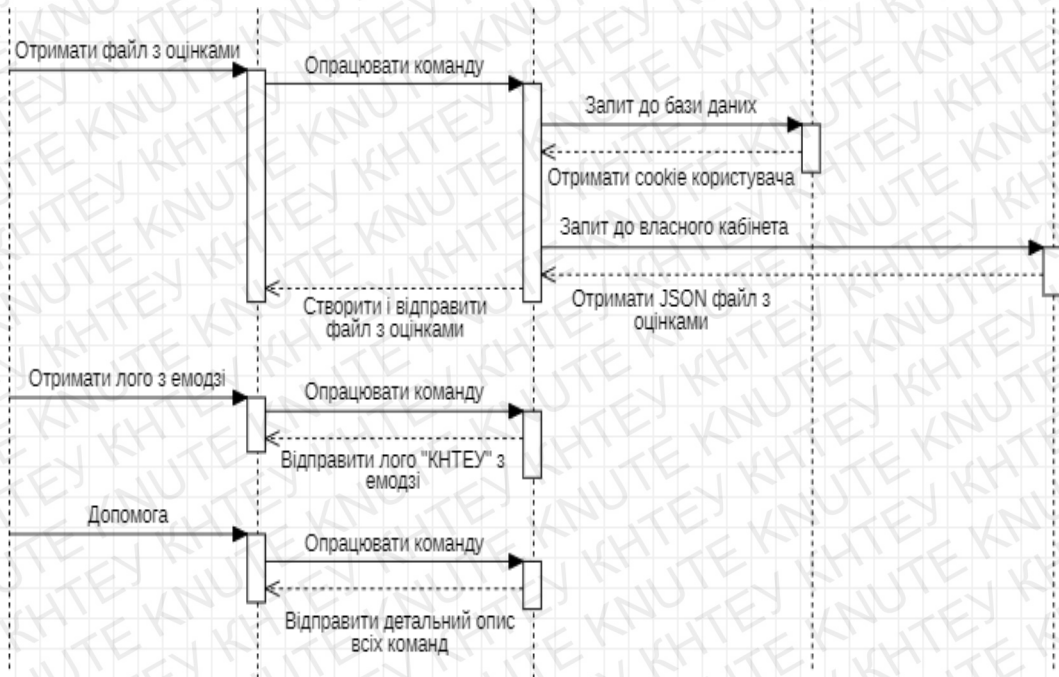


Рис. 2.3 – Діаграма послідовностей частина 2

2.1.3 Діаграма класів

Проект складається з таких класів: Web, Document, Bot, AppSettings, класів командами, моделей (Рис 2.4 - Діаграма класів).

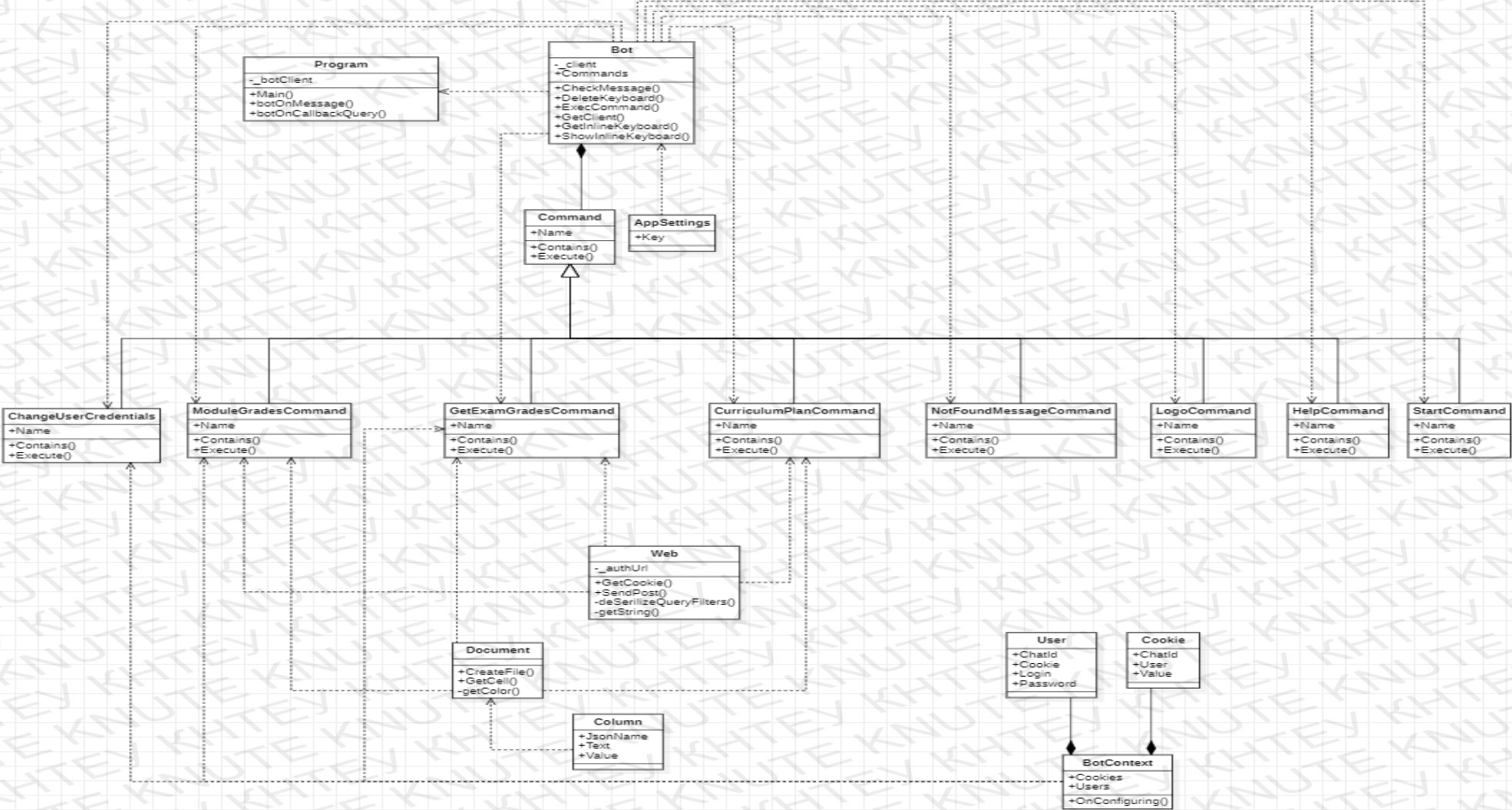


Рис 2.4 - Діаграма класів

2.1.4 Діаграма станів

Діаграма станів — діаграма, що визначає зміну станів об'єкту у часі, одна з діаграм моделювання поведінки в UML. Подає об'єкт як автомат з теорії автоматів зі стандартизованими умовними позначеннями (Рис 2.5 – Діаграма станів).

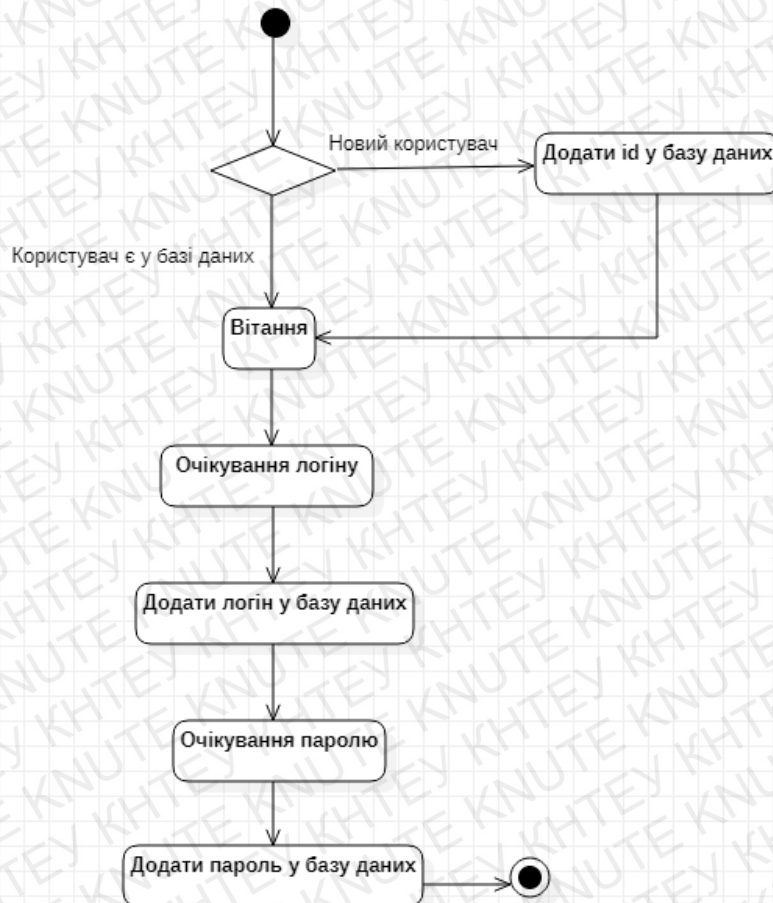


Рис 2.5 – Діаграма станів

2.1.5 Діаграма діяльності

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій (Рис 2.6 – Діаграма діяльності)

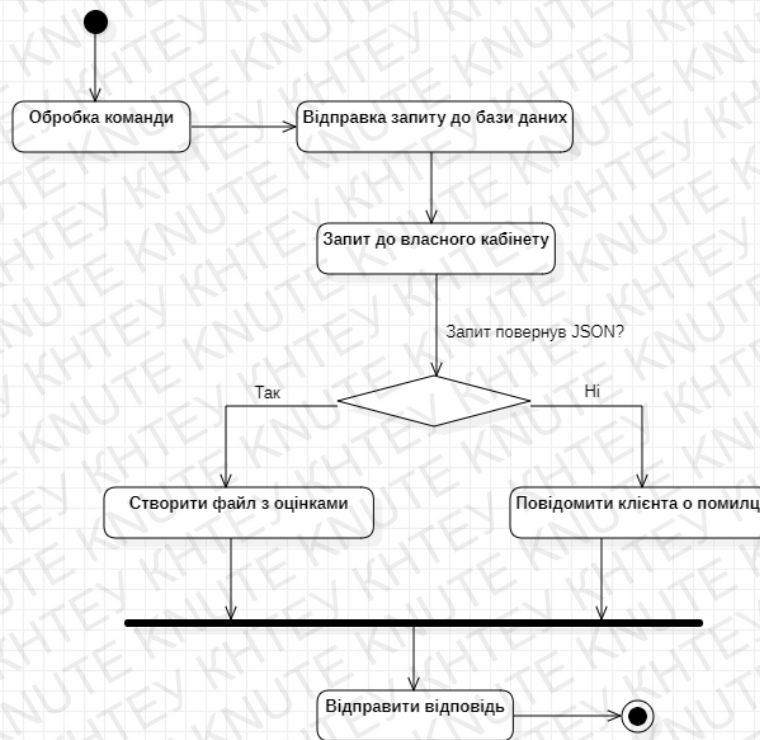


Рис 2.6 – Діаграма діяльності

2.2 База даних

Базою даних було вибрано SQLite. SQLite – компактна вбудована реляційна база даних. СУБД є безкоштовною і має відкритий вихідний код. Є чисто реляційної базою даних.

Слово «вбудована» означає, що SQLite не використовує парадигму клієнт-сервер. Тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якої програма компонується і движок стає складовою частиною програми. Таким чином, в якості протоколу обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід

зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції записи весь файл, який зберігає базу даних, блокується.

2.2.1 Структура таблиць

База даних Telegram бота містить дві таблиці: Users та Cookie. Таблиця Users містить в собі інформацію про користувача Telegram бота. Cookie зберігає куки користувача, отримані після авторизації у власному кабінеті сайту Кнтеу.

Users

User складається з таких полів:

- ChatId – id клієнта в Telegram. Дане поле є первинним ключем
- Login – логін користувача із власного кабінету
- Password – пароль користувача із власного кабінету
- State – Стан користувача

Cookie

- ChatId є одночасно і первинним та зовнішнім ключем
- Value зберігає cookie користувача у вигляді масиву байт

Більш детальна характеристика таблиць показана на рисунку 2.7

Name	Type	Schema
Cookies		CREATE TABLE "Cookies" ("ChatId" INTEGER NOT NULL CONSTRAINT "PK_Cookies" PRIMARY KEY, "Value" BLOB NULL, CONSTRAINT "FK_Cookies_Users_ChatId" FOREIGN KEY ("ChatId") REFERENCES "Users" ("ChatId") ON DELETE CASCADE)
ChatId	INTEGER	"ChatId" INTEGER NOT NULL
Value	BLOB	"Value" BLOB
Users		CREATE TABLE "Users" ("ChatId" INTEGER NOT NULL CONSTRAINT "PK_Users" PRIMARY KEY AUTOINCREMENT, "Login" TEXT NULL, "Password" TEXT NULL, "State" INTEGER NOT NULL)
ChatId	INTEGER	"ChatId" INTEGER NOT NULL
Login	TEXT	"Login" TEXT
Password	TEXT	"Password" TEXT
State	INTEGER	"State" INTEGER NOT NULL

Рис. 2.7 – Структура таблиц

Таблиці мають зв'язок одна з одною, адже у користувача може бути тільки один cookie.

2.2.2 Entity Framework

Entity Framework – це інструмент, який спрощує зіставлення об'єктів в програмному забезпеченні з таблицями і стовпцями реляційної бази даних. Воно дозволяє взаємодіяти з СУБД за допомогою сутностей (entity), а не таблиць. Також код з використанням EF пишеться набагато швидше. Наприклад, працюючи з базами даних безпосередньо, розробник повинен турбуватися про підключення, підготовці SQL і параметрів, відправлення запитів і транзакцій. На Entity Framework все це робиться автоматично – програміст же працює безпосередньо з сутностями і тільки говорить EF, що потрібно зробити.

Цей підхід надає ряд істотних переваг: не потрібно турбуватися про код доступу до даних, не потрібно знати деталей роботи СУБД SQL Server і синтаксис мови запитів T-SQL, замість цього користувач працює з таблицями бази даних як з класами C #, з полями цих таблиць - як з властивостями класів, а синтаксис SQL-запитів, який в ADO.NET раніше потрібно було вставляти в код C # у вигляді команд, замінений на більш зручний підхід з LINQ. Entity Framework бере на себе обов'язки по перетворенню коду C # в SQL-інструкції.

При роботі з Entity Framework розробнику надаються величезні можливості по створенню моделі бази даних за допомогою інтегрованого середовища розробки (IDE) Visual Studio.

2.3 Опис коду

Проект складається з таких класів, як: Web, Document, Bot, AppSettings, класів з командами бота, класів моделей, ну і самого головного класу – Program. Розглянемо детальніше кожний із них. Структура проекту в цілому відображається в додатку А (Додаток А.1 - Структура проекту).

2.3.1 Program

Program – це основний клас програми, в якому знаходиться головний метод Main. Даний метод є точкою входу, з нього починається виконання коду (Додаток А.2 –Метод Main).

Подія On-Message викликає метод botOnMessage кожного разу, коли боту буде приходити повідомлення (Додаток А.3 – Метод обробки повідомлень).

Подія OnCallbackQuery викликає метод botClient_OnCallbackQuery коли користувач натискає на будь-яку кнопку у головному меню (Рис 2.8 – Метод обробки натиску кнопок).

```
private static void botOnCallbackQuery
(object sender, Telegram.Bot.Args.CallbackQueryEventArgs e)
{
    var callback = e.CallbackQuery;
    Bot.CheckMessage(callback.Message, callback.Data);
}
```

Рис. 2.8 – Метод обробки натиску кнопок

Далі за допомогою метода StartReceiving починається періодична перевірка сервера Telegram на наявність оновлень. Якщо на сервері Telegram є оновлення, буде встановлено з'єднання на короткий час. Протягом цього часу, всі оновлення одночасно надсилаються боту за допомогою технології Long Polling.

2.3.2 Bot

Клас Bot уособлює сам Telegram бот. Він містить метод отримання об'єкта бота. Саме тут користувач отримаємо об'єкт телеграм бота для подальшої роботи з ним, а також заповнює колекцію команд бота.

Методи GetInlineKeyboard, ShowInlineKeyboard та DeleteKeyboard допомагають проводити маніпуляції з inline клавіатурою (Додаток Б.1 – Метод реєстрації та отримання об'єкта бота).

GetInlineKeyboard повертає екземпляр об'єкту inline клавіатури(меню) для подальшої відправки її користувачу (Додаток Б.2 – Метод отримання inline клавіатури).

У методі ShowInlineKeyboard саме і виконується відправка клавіатурі вказівки вибрати один з пунктів меню (Додаток Б.3 – Метод відправки клавіатури користувачу).

Після натиску на кнопку, поки клієнт очікує результат виконання команди, меню стає недоступним і пропадає за допомогою метода DeleteKeyboard (Додаток Б.4 - Видалити inline клавіатуру).

Але найголовнішим у даному класі є метод обробки повідомлень. Саме тут відбувається аналіз, обробка відправленого тексту, та виклик відповідної команди. Спершу ми перевіряємо наявність даного користувача у базі даних. Якщо такого клієнта не знайдено – відбувається запис у базу даних (Додаток В.1 – Метод обробки і виклику команд).

Далі викликається відповідна команда в залежності від відправленого тексту.

У разі якщо користувач спробував виконати неіснуючу команду, або тип відправленого повідомлення не є текстом (аудіофайл, зображення), то з'явиться меню з доступними командами та повідомлення з проханням вибрати з існуючого списку (Додаток В.2 – Метод обробки і виклику команд частина 2).

2.3.3 AppSettings

Клас AppSettings використовується для початкового налаштування бота. У даному класі міститься одна властивість що зберігає токен бота для доступу до HTTP API (Рис 2.9 – Властивість, що зберігає токен бота).

```
class AppSettings
{
    public static string Key { get; } =
        "1304784670:AAFsdXpo74fnFz090G0KJD2tSXu2cIqUFkw";
}
```

Рис. 2.9 – Властивість, що зберігає токен бота

2.3.4 Command

Командою називають функцію, яку виконує бот. Кожна команда є класом. Всі класи команд успадковуються від абстрактного класу Command. Даний клас має таку структуру:

- Назву команди
- Метод виклику
- Метод порівняння з назвою команди (для пошуку у колекції всіх команд)

Структура відображена на рисунку 2.10.

```

public abstract class Command
{
    public abstract string Name { get; set; }

    public abstract void Execute(Message message, TelegramBotClient client);

    public bool Contains(string command)
    {
        return command.Contains(this.Name);
    }
}

```

Рис. 2.10 – Структура абстрактного класу команди

Таблиця 2.1

Команди Telegram бота

Команда	Опис
/start	Розпочати роботу з ботом
/setcredentials	Повторно зайти у кабінет
/module	Оцінки за модульний контроль
/exams	Оцінки за Екзамени/Заліки
/curriculum	Навчальний план
/help	Допомога
/logo	Показати найкращий університет
/keyboard	Показати меню
/commandnotfoundmessage	Відправити повідомлення про неіснуючу команду

2.3.5 StartCommand

Команда /start викликається після першого запуску бота. Вона вітає користувача, змінює його стан на "Start" та викликає команду "/setcredentials", яка запускає низку подій для авторизації у персональному кабінеті (Додаток Г – Команда запуску бота).

2.3.6 ChangeUserCredentials

Команда "/setcredentials" потрібна для того, щоб перезайти у персональний кабінет. Після виклику команди розпочинається діалог з користувачем. Діалог реалізований за допомогою алгоритму кінцевого автомату.

Кінцевий автомат (або просто FSM – Finite-state machine) – це модель обчислень, яка заснована на гіпотетичній машині станів. В один момент часу тільки один стан може бути активним. Отже, для виконання будь-яких дій програма повинна змінювати свій стан. Кінцеві автомати зазвичай використовуються для організації та подання потоку виконання чого-небудь. Це особливо корисно при реалізації штучного інтелекту в іграх. Наприклад, при написанні штучного інтелекту ігрових персонажів.

Кінцевий автомат можна представити у вигляді графа, вершини якого є станами, а ребра - переходи між ними. Кожне ребро має мітку, що інформує про те, коли повинен відбутися перехід.

Всього існує 4 стана: початок роботи, введення логіну, введення пароля, очікування команди (Рис. 2.11 – Можливі стани користувача).

```
public enum State
{
    Start,
    Login,
    Password,
    Command
}
```

Рис. 2.11 – Можливі стани користувача

З кожним наступним кроком стан змінюється на наступний, і в залежності від цього змінюється логіка обробки повідомлення. Наприклад, якщо у користувача стан "Start" програма відправить повідомлення з проханням ввести логін та буде очікувати відповідь. Якщо стан «Login» то це означає, що логін ми вже маємо и просимо користувача відправити пароль.

Таким чином відбувається діалог між клієнтом і ботом (Додаток Д – Структура методу зміни даних для авторизації).

2.3.7 GetExamGradesCommand

Команда «Отримати оцінки за Екзамени/Заліки» допомагає розкрити головний потенціал Telegram бота – отримати PDF файл з оцінками за Екзамени/Заліки із власного кабінету користувача сайту університету (Додаток Е.1 - Структура методу отримання оцінок за Екзамени/Заліки).

Спочатку відправляється запит до бази даних, щоб отримати збережені cookie користувача. Вони знадобляться у наступному кроці, коли буде відправлятися запит до сайту університету. Щоб кожного разу не авторизуватися на сайті – зберігаємо cookie і таким чином працюємо в межах однієї сесії.

Після отримання об'єкту користувача та його cookie, передаємо ці дані методу для роботи з веб запитами та отримуємо відповідь у вигляді JSON.

У разі якщо метод повернув null – відправити повідомлення клієнту «Невірний логін або пароль».

Після успішного отримання оцінок у форматі JSON, створюється колекція з потрібними нам колонками у форматі (Назва колонки – назва колонки у JSON – довжина). Колекція з колонками відображена на рисунку 2.12.

```
var columns = new List<Column>()
{
    new Column("Дисципліна", "disc", 5f),
    new Column("Дата контролю", "Data_kontr", 2f),
    new Column("Результат", "NAC", 3f),
    new Column("ECTS", "ECTS", 1f),
    new Column("Лектор", "vLek", 4f),
    new Column("Годин", "god", 1f),
    new Column("Семестр", "Semestr", 1f)
};
```

Рис. 2.12 – Колекція з колонками

Наступним кроком відбувається десеріалізація строки JSON у об'єкт. У разі успішного перетворення масиву бітів у об'єкт, передається колекція з колонками та властивість з даними у метод створення pdf файла. Після деяких маніпуляцій отримаємо потрібний файл у форматі масиву байт, який десеріалізуємо у PDF файл та відправимо клієнту з указаною сьогоднішньою датою (Додаток Е.2 - Створення та відправка файлу з оцінками клієнту).

2.3.8 ModuleGradesCommand

Команда для отримання оцінок по модульному контролю має подібну структуру. Відрізняється лише властивістю з даними та колекцією колонок (Додаток Ж – Команда отримання оцінок за модульний контроль).

2.3.9 CurriculumPlanCommand

Аналогічним чином працює і команда отримання навчального плану. Змінюються лише параметри, а саме: url по якому відправляється запит та колекція колонок (Рис. 2.13 – Url для отримання JSON строки з оцінками).

```
const string errorMessage = "Невірний логін або пароль!";  
const string url = "https://knute.edu.ua/student-office/my-curriculum/?format=json";
```

Рис. 2.13 – Url для отримання JSON строки з оцінками

Колекція з колонками для PDF файлу зображена на рисунку 2.14.

```
var columns = new List<Column>()
{
    new Column("Назва дисципліни", "disc", 5f),
    new Column("Годин", "god", 1f),
    new Column("Лабораторні", "lab", 2f),
    new Column("Практики", "pra", 2f),
    new Column("Самостійно", "sam", 2f),
    new Column("Оцінювання", "VidOcin", 2f),
    new Column("Семестр", "Semestr", 2f),
    new Column("Лектор", "vLek", 4f),
    new Column("Асистенти", "vAss1", 4f),
    new Column("", "vAss2", 4f)
};
```

Рис. 2.14 – Налаштування команди отримання навчального плану

2.3.10 LogoCommand

Команда "/logo" відправляє користувачу лого з надписом "КНТЕУ", що складається з емодзі. (Додаток І – Команда відправки лого "КНТЕУ").

2.3.11 NotFoundMessageCommand

Дана команда, на відміну від інших, викликається не клієнтом, а ботом, після того, як користувач ввів невідомі системі дані. Наприклад, відправив зображення замість повідомлення, або спробував викликати неіснуючу команду (Рис 2.15 – Повідомлення при невідомій команді).

```
public override string Name { get; set; } = "/commandnotfoundmessage";

public override async void Execute(Message message, TelegramBotClient client)
{
    Bot.ShowInlineKeyboard(message, "Оберіть одну з кнопок нижче");
}
```

Рис. 2.15 – Повідомлення при невідомій команді

У відповідь клієнту на дану подію буде відправлене меню зі всіма командами та повідомленням з проханням обрати команду із цього списку.

2.3.12 HelpCommand

Для отримання опису можливостей бота та його команд використовується команда `"/help"`. (Додаток К – Команда допомоги).

2.3.13 Web

За допомогою класу Web відбувається взаємодія з власним кабінетом користувача. У метод SendPost передається url до якого треба відправити POST запит, словник для передачі параметрів у тіло запиту та cookie. В першу чергу будується рядок з аргументами для запиту та виконується серіалізація його у масив байтів. (Додаток Л.1 – Побудова тіла запиту).

Відправляємо запит. У разі успішного виконання операції налаштовується метод для отримання відповіді. Після цього користувач отримає масив байтів у відповідь. (Додаток Л.2 – Відправка POST запиту).

Для авторизації на сайті треба отримати cookie користувача. Для цього відправляємо запит на API авторизації. (Додаток Л.3 – Отримання cookie).

У разі успішної операції зберігається cookie у бази даних. (Додаток Л.4 - Збереження cookie у бази даних).

У метод GetRequest передається об'єкт користувача з його cookie та url до якого треба відправити post запит.

Алгоритм дії :

- 1) У разі якщо cookie null – користувач вперше зайшов у власний кабінет
- 2) У такому разі отримати cookie та записати у БД
- 3) Отримати cookie з БД
- 4) Десеріалізувати cookie

- 5) Відправити запит до власного кабінету, прикріпивши cookie до запиту, з метою отримати оцінки у форматі JSON (Додаток Л.5 – Отримання оцінок у JSON).
- 6) Якщо запит повернув null це означає що щось пішло не так (cookie застарів)
- 7) У цьому разі видаляємо неправильний cookie з БД та отримуємо новий.
- 8) Якщо знову відповідь не у форматі JSON, це означає що справа у неправильному логіну або пароллю.
- 9) У разі успішної операції повернути JSON, інакше null. (Додаток Л.6 – Валідація)

Отримаємо рядок відповіді за допомогою приватного метода getString(Додаток Л.7 – Десеріалізація відповіді у строку).

Метод deSerilizeQueryFilters переводить збережений cookie з формату масив байтів в об'єкт. (Додаток Л.8 – Десеріалізація cookie у об'єкт).

2.3.14 Document

Клас Document має метод CreateFile за допомогою якого і створюється pdf файл. (Додаток М.1 – Метод створення документу).

У якості аргументів метод приймає:

- колекцію колонок
- колекцію з даними для заповнення таблиці.

Спочатку налаштовується механізм будівлі таблиць, заповнюючи відповідними властивостями.

Після цього у першому циклі будуємо заголовок таблиці. У другому ж циклі заповнюємо таблицю даними. (Додаток М.2 – Побудова таблиці).

Після усіх ітерацій закривається файловий потік а створений файл повертається у форматі масиву байтів.

Метод GetCell будує та повертає розфарбовану комірку з текстом. (Додаток М.3 – Побудова комірки).

Розфарбовувати допомагає метод getColor, який повертає код кольору у форматі RGB(red, green, blue) в залежності від величини балу. (Додаток М.4 – Розфарбовування комірки).

2.3.15 Column

Клас Column є допоміжним для класу Document. Він слугує моделлю для побудови таблиці. Та має таку структуру: заголовок, назву атрибуту в json дереві, ширину колонки та значення. (Додаток Н.1 – Модель колонки).

2.3.16 Root

Root є системою класів для уособлення та представлення рядка JSON у вигляді об'єкта. (Додаток Н.2 – Модель JSON файлу).

2.3.17 BotContext

Клас BotContext використовується для підключення до бази даних, та називається контекстом. Саме тут налаштовується робота з базою даних та реєстрація властивостей таблиць. (Додаток Н.4 – Клас контексту бази даних).

2.3.18 User

Так як C# є об'єктно-орієнтованою мовою програмування, то сутність повинна бути класом. Клас User і є описом сутності. Так як для роботи з базою даних використовується Entity Framework, то замість полів таблиці взаємодія проходить через відповідні властивості.

Структура класу зображена у Додатку Н.5 – Модель класу користувача.

User складається з таких полів:

- ChatId – id клієнта в Telegram. Дане поле є первинним ключем
- Login – логін користувача із власного кабінету
- Password – пароль користувача із власного кабінету
- State – стан користувача
- Cookie – навігаційна властивість для побудови зв'язку з Cookie.

2.3.19 Cookie

Клас Cookie має таку структуру:

- ChatId є одночасно і первинним та зовнішнім ключем
- Value зберігає cookie користувача у вигляді масиву байт
- User – навігаційна властивість для побудови зв'язку з User

Структура відображена на Додатку Н.6 - Модель класу cookie

Висновки до розділу 2

Отже, була вибрана база даних SQLite. Адже це легка, але в одночас потужна база даних, що не потребує встановлення окремого сервера для її роботи. Для створення Telegram бота була побудована система класів з використанням Telegram API. Також були розроблені алгоритми обробки даних та створення pdf файлу з оцінками.

Зм..	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-09.БР

Аркуш

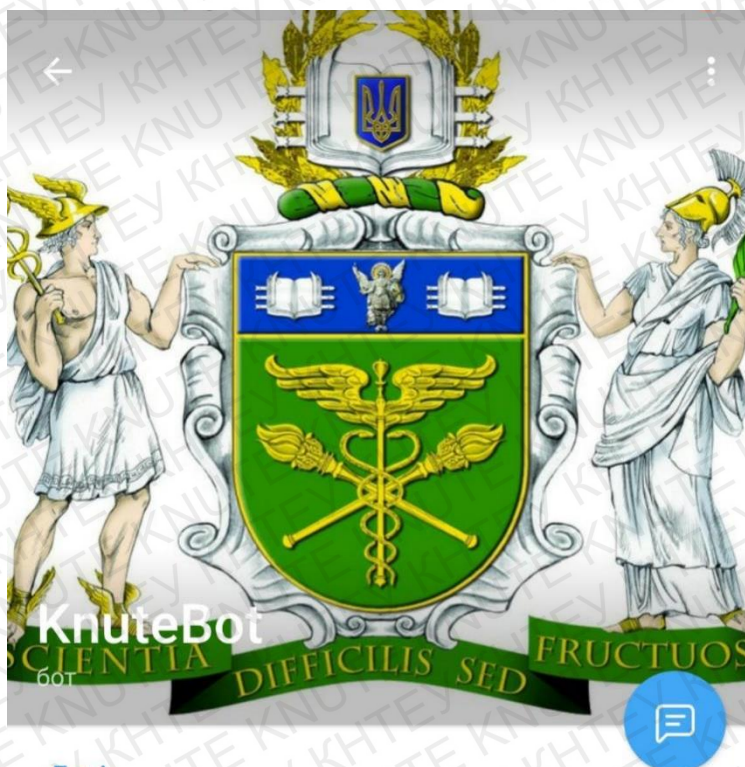
48

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ

3.1 Опис функціонування чат-боту

Для того, щоб розпочати роботу з ботом, потрібно спочатку його знайти по імені "@MyKNUTeBot" (Рис. 3.1 – Опис Telegram бота).



Дані

Даний бот був створений у рамках випускної кваліфікаційної роботи студента 4 курсу ФІТ Дьякова Дениса Юрійовича.

Про себе

@MyKNUTeBot

Псевдонім

Рис. 3.1 – Опис Telegram бота

					<i>КНТЕУ 121 06-09.БР</i>				
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>					
Зав. Каф.		Криворучко О.В.			<i>Розробка програмного забезпечення Telegram-боту для сайту університету</i>		Стадія	Аркуш	Аркушів
Керівник		Цензура М.О.					РЗ	49	60
Гарант		Цензура М.О.			<i>Програмна реалізація чат-боту</i>		<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
Розроб		Дьяков Д.Ю.							

Якщо користувач ще не взаємодіяв з ботом, то замість чату буде відображатись опис "Що вміє цей бот" (Рис. 3.2 – Початкове меню).

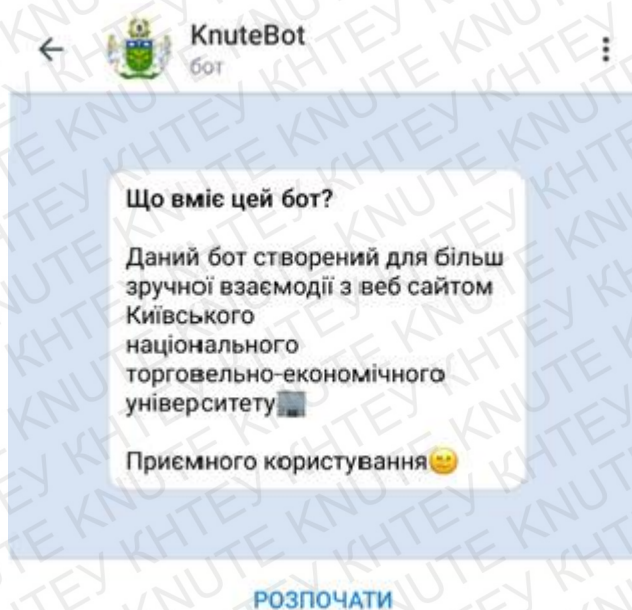


Рис. 3.2 – Початкове меню

Після натиску на кнопку "Розпочати" на екрані з'являється опис та команда «Start», яка і розпочинає роботу алгоритмів бота. Бот вітає студента, коротко описує свій функціонал та просить ввести логін від власного кабінету сайту КНТЕУ (Рис. 3.3 – Команда "Start").

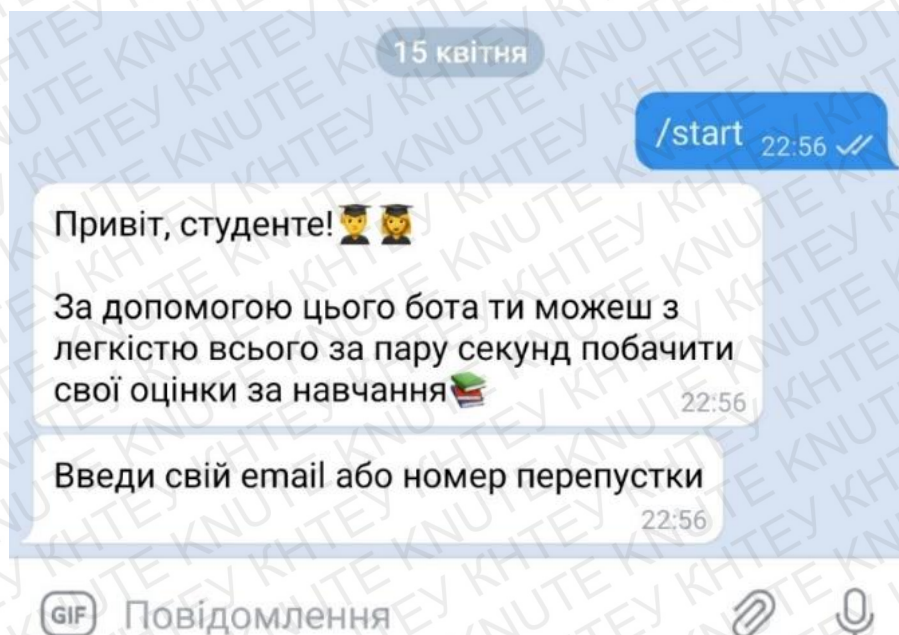


Рис. 3.3 – Команда "Start"

Коли ми закінчили діалог з ботом, а саме відправили йому свої дані для авторизації, бот представляє користувачу меню з можливістю виклику окремих команд (Рис. 3.4 – Меню).

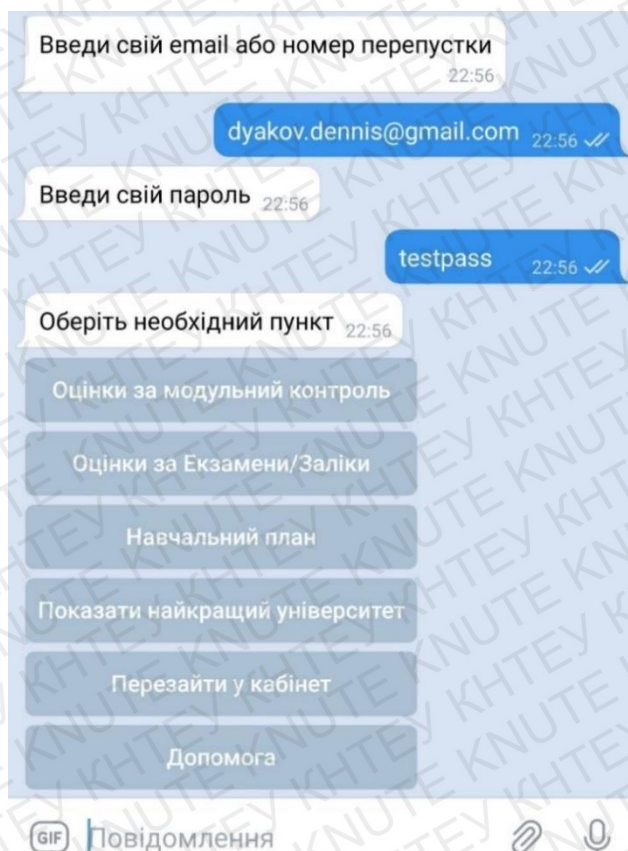


Рис. 3.4 – Меню

Меню складається з таких кнопок:

- Оцінки за модульний контроль
- Оцінки за Екзамени/Заліки
- Навчальний план
- Показати найкращий університет
- Перезайти у кабінет
- Допомога

Для того, щоб дізнатися про всі можливості бота, треба натиснути в кнопку "Допомога" (Рис. 3.5 – Команда "Допомога").

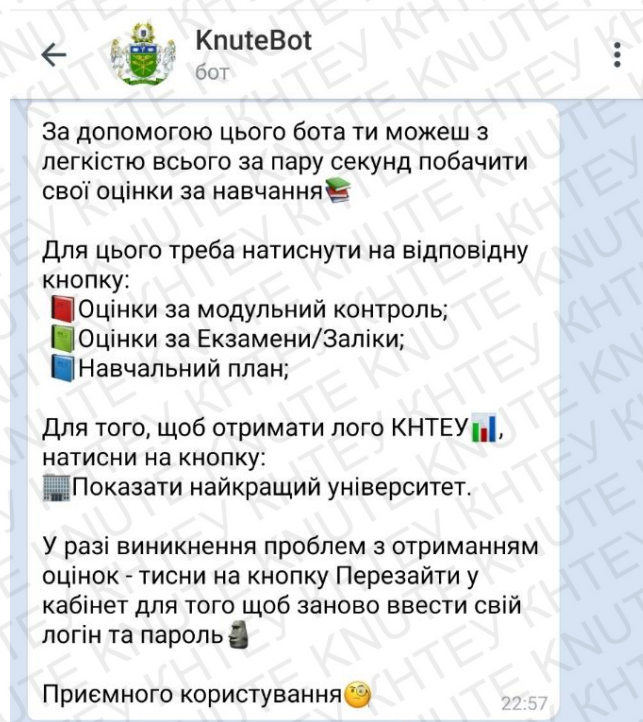


Рис. 3.5 – Команда "Допомога"

Під час виконання команд меню пропадає, а по завершенню знову з'являється. Для того, щоб отримати оцінки, треба натиснути на одну з кнопок відповідно. Структура файлів показана на рисунках: 3.6, 3.7, 3.8, 3.9, 3.10.

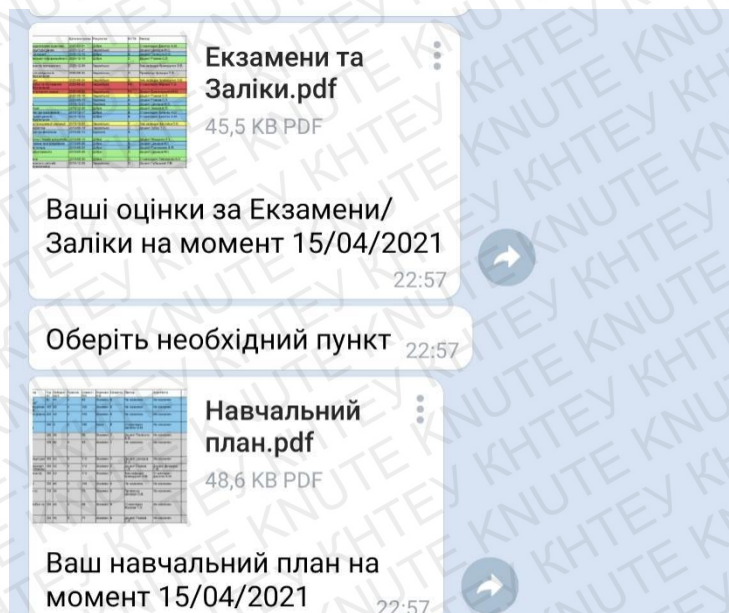


Рис. 3.6 – Pdf файли з оцінками

Дисципліна	Дата контролю	Результат	Всього балів	ECTS	Лектор	Годин	Семестр
Виробнича переддипломна практика	2021-03-01	Добре	75	C	Ст.викладач Десятко А.М.	180	8
Алгоритми та структури даних	2020-12-07	Задовільно	60	E	Доцент Цензура М.О.	180	7
Економіка і організація інформаційного бізнесу	2020-12-04	Задовільно	60	E	Доцент Рзаєва С.Л.	180	7
Менеджмент проектів програмного забезпечення	2020-12-04	Задовільно	65	E	Зав.кафедри Криворучко О.В.	180	7
Програмування Інтернет	2020-12-03	Добре	81	C	Доцент Палагута К.О.	180	7
КР з Баз даних	2020-06-10	Відмінно	90	A	Доцент Рзаєва С.Л.	30	6
Технологія розробки та тестування програмного забезпечення	2020-06-05	Незадовільно	22	F	Ст.викладач Жирова Т.О.	135	6
Стандартизація та метрологія програмного забезпечення	2020-06-04	Задовільно	60	E	Професор Цюцюра С.В.	135	6
Експертні системи	2020-06-04	Задовільно	60	E	Зав.кафедри Криворучко О.В.	180	6
Організація комп'ютерних мереж	2020-06-04	Незадовільно	20	F	Доцент Браїловський М.М.	180	6
Бази даних	2020-05-13	Незадовільно	55	FX	Доцент Рзаєва С.Л.	150	6
Архітектура та проектування програмного забезпечення	2019-12-05	Добре	75	C	Ст.викладач Десятко А.М.	180	5
Технологія Java	2019-12-05	Відмінно	90	A	Доцент Цензура М.О.	180	5
Емпіричні методи програмної інженерії	2019-12-04	Незадовільно	50	FX	Зав.кафедри Шетініна О.К.	180	5
Операційні системи	2019-12-03	Задовільно	72	D	Доцент Звєрєв В.П.	180	5
Математична логіка і теорія алгоритмів	2019-05-31	Задовільно	70	D	Доцент Мащенко Л.З.	180	4
Об'єктно орієнтоване програмування	2019-05-30	Добре	81	C	Доцент Цюцюра М.І.	150	4
КР з Об'єктно орієнтованого програмування	2019-05-30	Добре	81	C	Доцент Цюцюра М.І.	30	4
Фізичне виховання	2019-05-30	Добре	75	C	Ст.викладач Пивоваров А.А.	68	4
Економіка підприємства	2019-05-27	Незадовільно	40	FX	Доцент Зубко Т.Л.	180	4
Архітектура комп'ютера	2019-05-21	Добре	82	B	Доцент Рассамкін В.Я.	180	4
Іноземна мова (за професійним спрямуванням)	2019-05-17	Відмінно	90	A		180	4
Web дизайн і Web програмування	2019-03-04	Добре	75	C	Ст.викладач Котенко Н.О.	180	5
Охорона праці	2018-12-06	Добре	75	C	Доцент Палієнко О.О.	135	3
Об'єктно орієнтоване програмування	2018-12-06	Незадовільно	50	FX	Зав.кафедри Криворучко О.В.	180	3

Національні інтереси в світовій геополітиці та геоekonomіці	2018-12-05	Задовільно	60	E	Доцент Губицький Л.В.	135	3
Комп'ютерна дискретна математика	2018-12-04	Добре	75	C	Доцент Котляр В.Ю.	135	3
Філософія	2018-12-03	Задовільно	70	D	Доцент Литвин Н.М.	135	3
Іноземна мова (за професійним спрямуванням)	2018-11-23	Відмінно	90	A		180	3
Фізичне виховання	2018-06-01	Відмінно	90	A	Ст.викладач Пивоваров А.А.	90	2
Математичний аналіз	2018-05-31	Добре	75	C	Доцент Михайленко С.В.	270	2
Основи інженерії програмного забезпечення	2018-05-24	Добре	82	B	Доцент Чернякін В.П.	180	2
Безпека життєдіяльності	2018-05-24	Незадовільно	39	FX	Ст.викладач Болілий О.С.	90	2
Українська мова (за професійним спрямуванням)	2018-05-17	Задовільно	60	E		90	2
Право (Підприємницьке право)	2018-05-16	Задовільно	70	D	Доцент Можайкіна О.С.	90	2
Іноземна мова (за професійним спрямуванням)	2018-05-14	Відмінно	90	A		90	2
Історія економіки та економічної думки	2017-12-08	Задовільно	60	E	Професор Ніколаєць К.М.	135	1
Інженерна та комп'ютерна графіка	2017-12-08	Добре	75	C	Доцент Григоренко О.М.	135	1
Фізичне виховання	2017-12-08	Задовільно	70	D	Ст.викладач Пивоваров А.А.	90	1
Лінійна алгебра та аналітична геометрія	2017-12-06	Задовільно	65	E	Доцент Білоусова С.В.	90	1
Основи інженерії програмного забезпечення	2017-12-05	Задовільно	70	D	Доцент Чернякін В.П.	135	1
Правознавство	2017-12-05	Задовільно	73	D	Ст.викладач Хромей В.В.	135	1
Історія української культури	2017-11-28	Добре	75	C	Доцент Кизименко І.О.	90	1
Іноземна мова (за професійним спрямуванням)	2017-11-22	Добре	87	B		90	1

Рис. 3.7 – Структура pdf файла

Дисципліна	Дата контролю	Результат	Всього балів	ECTS	Лектор	Годин	Семестр
Виробнича переддипломна практика	2021-03-01	Добре	75	C	Ст.викладач Десятко А.М.	180	8
Алгоритми та структури даних	2020-12-07	Задовільно	60	E	Доцент Цензура М.О.	180	7
Економіка і організація інформаційного бізнесу	2020-12-04	Задовільно	60	E	Доцент Рзаєва С.Л.	180	7
Менеджмент проєктів програмного забезпечення	2020-12-04	Задовільно	65	E	Зав.кафедри Криворучко О.В.	180	7
Програмування Інтернет	2020-12-03	Добре	81	C	Доцент Палагута К.О.	180	7
КР з Баз даних	2020-06-10	Відмінно	90	A	Доцент Рзаєва С.Л.	30	6
Технологія розробки та тестування програмного забезпечення	2020-06-05	Незадовільно	22	F	Ст.викладач Жирова Т.О.	135	6
Стандартизація та метрологія програмного забезпечення	2020-06-04	Задовільно	60	E	Професор Цюцюра С.В.	135	6
Експертні системи	2020-06-04	Задовільно	60	E	Зав.кафедри Криворучко О.В.	180	6
Організація комп'ютерних мереж	2020-06-04	Незадовільно	20	F	Доцент Браїловський М.М.	180	6
Бази даних	2020-05-13	Незадовільно	55	FX	Доцент Рзаєва С.Л.	150	6
Архітектура та проектування програмного забезпечення	2019-12-05	Добре	75	C	Ст.викладач Десятко А.М.	180	5
Технологія Java	2019-12-05	Відмінно	90	A	Доцент Цензура М.О.	180	5
Емпіричні методи програмної інженерії	2019-12-04	Незадовільно	50	FX	Зав.кафедри Щетініна О.К.	180	5
Операційні системи	2019-12-03	Задовільно	72	D	Доцент Зверев В.П.	180	5
Математична логіка і теорія алгоритмів	2019-05-31	Задовільно	70	D	Доцент Мащенко Л.З.	180	4
Об'єктно орієнтоване програмування	2019-05-30	Добре	81	C	Доцент Цюцюра М.І.	150	4
КР з Об'єктно орієнтованого програмування	2019-05-30	Добре	81	C	Доцент Цюцюра М.І.	30	4

Рис. 3.8 – Оцінки за модульний контроль

Назва дисципліни	Годин	Лаб	Практики	Самостійно	Оцінювання	Семестр	Лектор	Асистенти
Практичний курс "Бізнес-симуляція"	80	40	0	40	Екзамен	8	Не назначен	Не назначен
Безпека інформаційних систем та мереж	180	48	0	100	Екзамен	8	Не назначен	Не назначен
Моделювання та аналіз програмного забезпечення	225	48	0	145	Екзамен	8	Не назначен	Не назначен
Виробнича переддипломна практика	180	0	0	180	Зачет	8	Ст.викладач Десятко А.М.	Не назначен
Програмування Інтернет	180	56	0	96	Екзамен	7	Доцент Палагута К.О.	Не назначен
Моделювання фінансово-господарської діяльності підприємства	100	60	0	40	Екзамен	7	Не назначен	Не назначен
Алгоритми та структури даних	180	42	0	110	Екзамен	7	Доцент Цензура М.О.	Не назначен
Економіка і організація інформаційного бізнесу	180	42	0	110	Екзамен	7	Доцент Рзаєва С.Л.	Доцент Демедюк С.В.
Менеджмент проєктів програмного забезпечення	180	42	0	110	Екзамен	7	Зав.кафедри Криворучко О.В.	Ст.викладач Десятко А.М.
Методи і засоби передачі даних	180	46	0	104	Екзамен	6	Не назначен	Не назначен
Стандартизація та метрологія програмного забезпечення	135	30	0	75	Екзамен	6	Професор Цюцюра С.В.	Не назначен
Технологія розробки та тестування програмного забезпечення	135	46	0	59	Екзамен	6	Ст.викладач Жирова Т.О.	Не назначен

Рис. 3.9 – Навчальний план

Для того, щоб змінити дані для авторизації треба натиснути на "Перезайти у кабінет". У разі якщо невірний пароль або логін, при спробі отримати оцінки бот відправляє повідомлення о помилці (Рис. 3.10 – Спроба отримати оцінки з невірними даними для авторизації).



Рис. 3.10 – Спроба отримати оцінки з невірними даними для авторизації

При натисканні на клавішу «Показати найкращий університет» отримуємо лого КНТЕУ, яке складається з емодзі (Рис. 3.11 - Лого КНТЕУ з емодзі).



Рис. 3.11 – Лого КНТЕУ з емодзі

У разі, якщо користувач відправить невідому команду, бот попросить користувача скористатися існуючим функціоналом та відправить меню (Рис. 3.12 – Користувач ввів непередбачений запит).

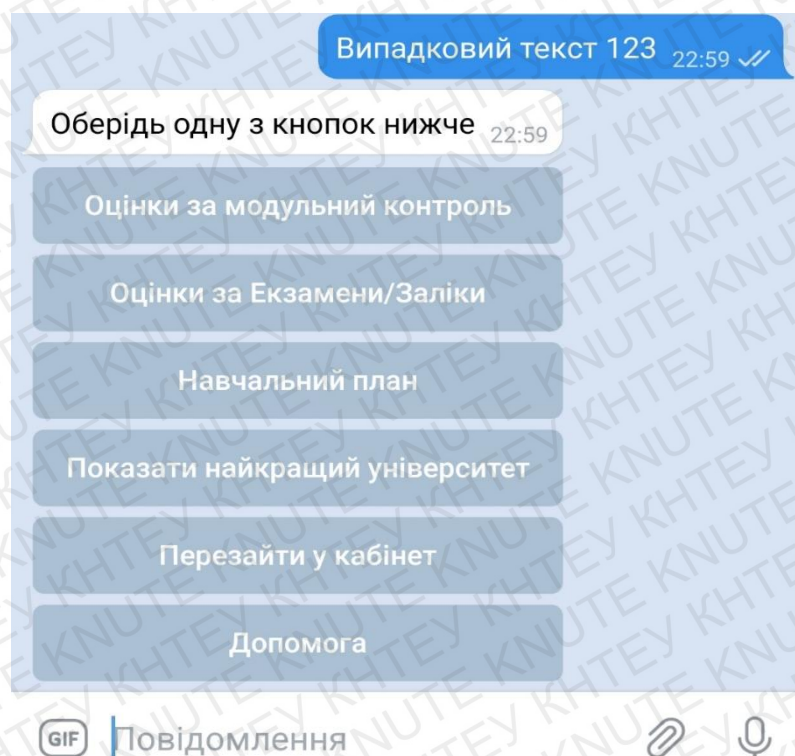


Рис. 3.12 – Користувач ввів непередбачений запит

Висновки до розділу 3

В результаті був створений повнофункціональний чат-бот для більш зручної взаємодії з сайтом Київського національного торговельно-економічного університету. Був розроблений інтерфейс програми, описані та реалізовані команди. Інтерфейс складається з кнопок, які і відповідають за виклик відповідних команд. В кінці було проведене тестування функціоналу та виправлення знайдених багів.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Результатом виконання дипломного проекту став повнофункціональний чат-бот для більш зручної взаємодії з веб-сайтом Київського національного торговельно-економічного університету. У якості об'єктно-орієнтованої мови програмування було обрано C#. Адже на сьогоднішній момент дана мова програмування є однією з найпотужніших та затребуваних мов в IT-галузі. На даний момент на C# пишуться найрізноманітніші програми: від невеликих десктопних програм до великих веб-порталів і веб-сервісів, які обслуговують щодня мільйони користувачів.

В якості бази даних було обрано SQLite. SQLite – це компактна вбудована реляційна база даних. Вона є безкоштовною та відкритою. SQLite не використовує парадигму клієнт-сервер. Це означає, що движок SQLite – це не окремий процес, з яким програма взаємодіє, а є бібліотекою, з якою програма збирається і механізм стає частиною програми. Тому виклики функцій SQLite (API) використовуються як протокол обміну. Цей підхід зменшує накладні витрати, час відгуку та спрощує програму.

Був проведений аналіз безпеки використання месенджера Telegram та безпосередньо чат-боту. Безпеку в Telegram ґрунтується на криптографічному протоколі MTProto Proху завдяки якому передача трафіку через канали телеграм, замаскована під стандартне TSL / SSL-з'єднання, але всередині дані зашифровані таким чином, що нічого неможливо розпізнавати. Також месенджер використовує шифрування клієнт-клієнт, клієнт-сервер, а ще дає можливість встановити хмарний та код паролі.

					<i>КНТЕУ 121 06-09.БР</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка програмного забезпечення Telegram-боту для сайту університету</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. Каф.		Криворучко О.В.				ВП	58	60
Керівник		Цензура М.О.			<i>Висновки та пропозиції</i>	<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
Гарант		Цензура М.О.						
Розроб		Дьяков Д.Ю.						

Також завдяки мультиплатформеності месенджера Telegram користуватись функціоналом чат-бота можливо на багатьох видах пристроїв: починаючи від смартфонів, та закінчуючи ноутбуками, планшетами та персональними комп'ютерами.

Додаток розроблений таким чином, що при необхідності можна з легкістю додати новий функціонал просто створюючи нові класи команд з відповідною логікою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Джеффрі Ріхтер CLR via C# – В.: «Діалектика», 2019. – 1161 ст.
2. Грофф Д. SQL. Повне керування. / Д. Грофф, П. Вайнберг, Е. Дж. Оппель. - В.: «Вільямс», 2015. – 960 ст.
3. Стілмен Е. Вивчаємо C#. Включаючи C# .NET 4.0 і Visual Studio 2010. 2-е видання / Е. Стілмен., Дж. Грін - O'Reilly, 2012. – 689 с.
4. Шілдт Герберт. C# 4.0: Повне керівництво / Герберт Шілдт. – Вільямс, 2011. – 1056 с.
5. Молінаро Ентоні. SQL. Збірник рецептів / Ентоні Молінаро. – O'Reilly, 2009. – 672 с.

Інтернет ресурси:

6. Сайт SQLite // SQLite Home Page [Електронний ресурс]- Режим доступу: <http://sqlite.org/>.
7. Сайт Newtonsoft.Json // Json.NET [Електронний ресурс]- Режим доступу: <http://www.newtonsoft.com/json>.
8. Галерея додатків VisualStudio // NuGet [Електронний ресурс]- Режим доступу: <https://www.nuget.org/packages/>.
9. Документація компілятора C# Roslyn // Github [Електронний ресурс]- Режим доступу: <https://github.com/dotnet/roslyn>.
10. Повне керівництво по C# 9 і .NET 5 [Електронний ресурс]- Режим доступу: <https://metanit.com/sharp/tutorial/>.

					КНТЕУ 121 06-09.БР				
Зм.	Арк.	№ докум	Підпис	Дата					
Зав. Каф.		Криворучко О.В.			Розробка програмного забезпечення Telegram-боту для сайту університету		Стадія	Аркуш	Аркушів
Керівник		Цензура М.О.					СД	60	60
Гарант		Цензура М.О.			Список використаних джерел		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб		Дьяков Д.Ю.							

ДОДАТКИ

Додаток А

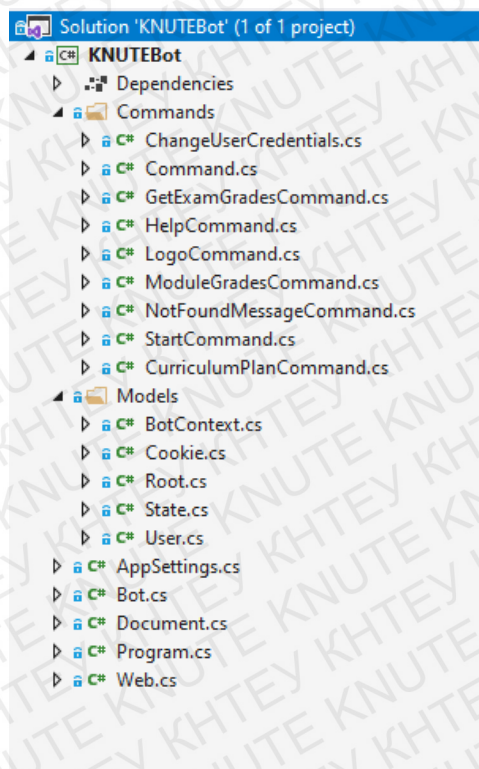


Рис. А.1 – Структура проекту

```
public static void Main(string[] args)
{
    _botClient = Bot.GetClient();
    _botClient.OnMessage += botOnMessage;
    _botClient.OnCallbackQuery += _botClient_OnCallbackQuery;
    var me = _botClient.GetMeAsync().Result;
    _botClient.StartReceiving();
    Console.WriteLine("Bot id - {0}, Bot Name - {1}", me.Id, me.FirstName);
    Console.ReadKey();
}
```

Рис. А.2 – Метод Main

```

private static void botOnMessage(object sender, Telegram.Bot.Args.MessageEventArgs e)
{
    var message = e.Message;
    if (message == null || message.Type != MessageType.Text)
    {
        Bot.ExecCommand(message, Bot.GetClient(), "/commandnotfoundmessage");
        return;
    }
    Console.WriteLine("Message " + e.Message.Text);
    Console.WriteLine("Id" + e.Message.Chat.Id);
    Bot.CheckMessage(e.Message);
}

```

Рис. А.3 – Метод обробки повідомлень

```

public static IReadOnlyList<Command> Commands { get => _commands.AsReadOnly(); }

public static TelegramBotClient GetClient()
{
    if (_client != null)
        return _client;
    _commands = new List<Command>()
    {
        new StartCommand(),
        new GetExamGradesCommand(),
        new ChangeUserCredentials(),
        new HelpCommand(),
        new ShowInlineKeyboard(),
        new NotFoundMessageCommand(),
        new ModuleGradesCommand(),
        new LogoCommand(),
        new CurriculumPlanCommand()
    };
    _client = new TelegramBotClient(AppSettings.Key) { Timeout = TimeSpan.FromSeconds(10) };
    return _client;
}

```

Рис. Б.1 – Метод реєстрації та отримання об'єкта бота

```

public static InlineKeyboardMarkup GetInlineKeyboard()
{
    Dictionary<string, string> buttons = new Dictionary<string, string>
    {
        { "Оцінки за модульний контроль", "/module" },
        { "Оцінки за Екзамени/Заліки", "/exams" },
        { "Навчальний план", "/curriculum" },
        { "Показати найкращий університет", "/logo" },
        { "Перезайти у кабінет", "/setcredentials" },
        { "Допомога", "/help" }
    };
    InlineKeyboardButton[][] ik = buttons.Select(item => new[]
    {
        InlineKeyboardButton.WithCallbackData(item.Key, item.Value)
    }).ToArray();
    return new InlineKeyboardMarkup(ik);
}

```

Рис. Б.2 – Метод отримання inline клавіатури

```

public static async void ShowInlineKeyboard(Message message, string text = "Оберіть необхідний пункт")
{
    var keyboard = GetInlineKeyboard();
    Thread.Sleep(1000);
    try
    {
        await _client.SendTextMessageAsync(
            chatId: message.Chat.Id,
            text: text,
            replyMarkup: keyboard
        );
    }
    catch (Exception)
    {
    }
}

```

Рис. Б.3 – Метод відправки клавіатури користувачу

```

public static async void DeleteKeyboard(Message message, TelegramBotClient client)
{
    try
    {
        await client.EditMessageReplyMarkupAsync(message.Chat.Id, message.MessageId);
    }
    catch (Exception)
    {
    }
}

```

Рис. Б.4 – Видалити inline клавіатуру

```

public static void CheckMessage(Message message, string commandText = null)
{
    string text = message.Text;
    if (!string.IsNullOrEmpty(commandText))
    {
        text = commandText;
    }

    long id = message.Chat.Id;
    User user = null;
    using (BotContext db = new BotContext())
    {
        user = db.Users.Find(id);
        if (user == null)
        {
            var cookie = new Cookie
            {
                ChatId = id,
                Value = null
            };

            user = new User
            {
                ChatId = id,
                State = State.Start,
                Cookie = cookie
            };
            db.Users.Add(user);
            db.SaveChanges();
        }
    }
}

```

Рис. В.1 – Метод обробки і виклику команд частина 1

```

var client = GetClient();
foreach (var command in Commands)
{
    if (user.State != State.Command)
    {
        ExecCommand(message, client, "/setcredentials");
        return;
    }
    if (command.Contains(text))
    {
        command.Execute(message, client);
        return;
    }
}
ExecCommand(message, client, "/commandnotfoundmessage");
}

```

Рис. В.2 – Метод обробки і виклику команд частина 2

```

class StartCommand : Command
{
    public override string Name { get; set; } = "/start";

    public override async void Execute(Message message, TelegramBotClient client)
    {
        long id = message.Chat.Id;
        using (BotContext db = new BotContext())
        {
            var user = db.Users.Find(id);
            user.State = State.Start;
            db.SaveChanges();
        }
        await client.SendTextMessageAsync(
            chatId: id,
            text: @"Привіт, студенте! 🤖👋");

        За допомогою цього бота ти можеш з легкістю всього за пару секунд побачити свої оцінки за навчання! 📊
        "
    );
    Thread.Sleep(500);
    Bot.ExecCommand(message, client, "/setcredentials");
}
}

```

Рис. Г.1 – Команда запуску бота

```

const string loginMessage = "Введіть ваш email або номер перепустки";
const string passwordMessage = "Введіть ваш пароль";

long id = message.Chat.Id;
using (BotContext db = new BotContext())
{
    var user = db.Users.Find(id);
    State state = user.State;
    string credentials = message.Text;

    if (state == State.Start || state == State.Command)
    {
        user.State = State.Login;
        user.Cookie.Value = null;
        await client.SendTextMessageAsync(
            chatId: id,
            text: loginMessage
        );
    }
    else if (state == State.Login)
    {
        user.Login = credentials;
        user.State = State.Password;
        await client.SendTextMessageAsync(
            chatId: id,
            text: passwordMessage
        );
    }
    else if (state == State.Password)
    {
        user.Password = credentials;
        user.State = State.Command;
        Bot.ShowInlineKeyboard(message);
    }
    db.Users.Update(user);
    db.SaveChanges();
}

```

Рис. Д.1 – Структура методу зміни даних для авторизації

```

const string errorMessage = "Невірний логін або пароль!";
const string url = "https://knute.edu.ua/student-office/my-stat/?format=json#";

long id = message.Chat.Id;
Models.User user = null;
using (BotContext db = new BotContext())
{
    user = db.Users.Include(x => x.Cookie).FirstOrDefault(x => x.ChatId == id);
}

string json = Web.GetRequest(user, url);

if (string.IsNullOrEmpty(json))
{
    Bot.DeleteKeyboard(message, client);
    await client.SendTextMessageAsync(
        chatId: id,
        text: errorMessage
    );
    Bot.ShowInlineKeyboard(message);
    return;
}

Bot.DeleteKeyboard(message, client);

```

Рис. Е.1 – Структура методу отримання оцінок за Екзамени/Заліки

```

Root knute = JsonConvert.DeserializeObject<Root>(json);

if (knute != null)
{
    var file = Document.CreateFile(columns, knute.data.module_controll_lessons.OrderByDescending(x => x.Data_kontr).ToList());
    using (var stream = new MemoryStream(file))
    {
        await client.SendDocumentAsync(
            chatId: id,
            document: new InputOnlineFile(stream, "ExamMarks.pdf"),
            caption: $"Ваші оцінки за Екзамени/Заліки на момент {DateTime.Now.ToShortDateString()}"
        );
    }
}

Bot.ShowInlineKeyboard(message);

```

Рис. Е.2 – Створення та відправка файлу з оцінками клієнту

```

var columns = new List<Column>()
{
    new Column("Дисципліна", "disc", 5f),
    new Column("Дата контролю", "Data_kontr", 2f),
    new Column("Результат", "NAC", 3f),
    new Column("Всього балів", "ball", 1f),
    new Column("ECTS", "ECTS", 1f),
    new Column("Лектор", "vLek", 4f),
    new Column("Годин", "god", 1f),
    new Column("Семестр", "Semestr", 1f)
};

Root knute = JsonConvert.DeserializeObject<Root>(json);

if (knute != null)
{
    var file = Document.CreateFile(columns, knute.data.offset_lessons.OrderByDescending(x => x.Data_kontr).ToList());
    using (var stream = new MemoryStream(file))
    {
        await client.SendDocumentAsync(
            chatId: id,
            document: new InputOnlineFile(stream, "Модульний контроль.pdf"),
            caption: $"Ваші оцінки за модульний контроль на момент {DateTime.Now.ToShortDateString()}")
        );
    }
}

```

Рис. Ж.1 – Команда отримання оцінок за модульний контроль

```
class LogoCommand : Command
{
    public override string Name { get; set; } = "/logo";

    public override async void Execute(Message message, TelegramBotClient client)
    {
        const string text = @"
██████████
████♥♥♥♥♥████
████♥♥♥♥♥████
████♥♥♥♥♥████
████♥KHTEY♥████
████♥♥♥♥♥████
████♥♥♥♥♥████
██████████♥████
██████████";

        Bot.DeleteKeyboard(message, client);
        await client.SendTextMessageAsync(
            chatId: message.Chat.Id,
            text: text
        ); ;

        Bot.ShowInlineKeyboard(message);
    }
}
```

Рис. И.1 – Команда відправки лого "КНТЕУ"

```

class HelpCommand : Command
{
    public override string Name { get; set; } = "/help";

    public override async void Execute(Message message, TelegramBotClient client)
    {
        Bot.DeleteKeyboard(message, client);
        await client.SendTextMessageAsync(
            chatId: message.Chat.Id,
            text: @"За допомогою цього бота ти можеш з легкістю всього за пару секунд побачити свої оцінки за навчання"
        );

        Для цього треба натиснути на відповідну кнопку:
        ☐ Оцінки за модульний контроль;
        ☐ Оцінки за Екзамени/Заліки;
        ☐ Навчальний план;

        Для того, щоб отримати лого КНТЕУ, натисни на кнопку:
        ☐ Показати найкращий університет.

        У разі виникнення проблем з отриманням оцінок - тисни на кнопку Перезайти у кабінет для того щоб заново ввести свій логін та пароль.

        Приємного користування
    }
}
Bot.ShowInlineKeyboard(message);

```

Рис. К.1 – Команда допомоги

```

public static HttpResponseMessage SentPost(string url, Dictionary<string, string> _formValues, CookieContainer cookies)
{
    StringBuilder postString = new StringBuilder();
    bool first = true;
    foreach (KeyValuePair<string, string> pair in _formValues)
    {
        if (first)
            first = false;
        else
            postString.Append("&");
        postString.AppendFormat("{0}={1}", pair.Key, HttpUtility.UrlEncode(pair.Value));
    }
    ASCIIEncoding ascii = new ASCIIEncoding();
    byte[] postBytes = ascii.GetBytes(postString.ToString());
}

```

Рис. Л.1 - Побудова тіла запиту

```

HttpRequest request;
try
{
    request = WebRequest.Create(url) as HttpRequest;
}
catch (UriFormatException)
{
    throw new ApplicationException("Invalid URL: " + url);
}

request.Method = "POST";
request.CookieContainer = cookies;
request.ContentType = "application/x-www-form-urlencoded";
request.ContentLength = postBytes.Length;

Stream postStream = request.GetRequestStream();
postStream.Write(postBytes, 0, postBytes.Length);
postStream.Close();

HttpResponse response = new HttpResponseMessage();
try
{
    response = request.GetResponse() as HttpResponseMessage;
}
catch (Exception)
{
    return null;
}
return response;
}

```

Рис. Л.2 – Відправка POST запиту

```

private const string _authUrl = "https://knute.edu.ua/student-office/auth/?format=json";

public static bool GetCookie(User user)
{
    Dictionary<string, string> loginPass = new Dictionary<string, string>();
    loginPass["login"] = user.Login;
    loginPass["pass"] = user.Password;

    HttpResponseMessage auth = SentPost(_authUrl, loginPass);

    if (auth.Cookies.Count < 3)
    {
        return false;
    }

    CookieContainer cookies = new CookieContainer();
    cookies.Add(auth.Cookies);
}

```

Рис. Л.3 – Отримання cookie

```

string uniText = getString(auth);
using (var reader = new StreamReader(auth.GetResponseStream()))
{
    uniText = reader.ReadToEnd();
}

if (uniText.Contains("false"))
{
    return false;
}

BinaryFormatter bf = new BinaryFormatter();
byte[] byteArray = null;
using (MemoryStream ms = new MemoryStream())
{
    bf.Serialize(ms, cookies);
    byteArray = ms.GetBuffer();
}

using (BotContext db = new BotContext())
{
    user.Cookie.Value = byteArray;
    db.Users.Update(user);
    db.SaveChanges();
}

return true;
}

```

Рис. Л.4 – Збереження cookie у базі даних

```

public static string GetRequest(User user, string url)
{
    Dictionary<string, string> loginPass = new Dictionary<string, string>();
    loginPass["login"] = user.Login;
    loginPass["pass"] = user.Password;

    for (int i = 0; i < 2; i++)
    {
        if (user.Cookie.Value == null)
        {
            bool isErrors = GetCookie(user);
            if (!isErrors)
            {
                return null;
            }
        }
        byte[] bytes = null;
        using (BotContext db = new BotContext())
        {
            bytes = user.Cookie.Value;
        }
        var cookieContainer = deSerializeQueryFilters(bytes);
        HttpWebResponse data = SentPost(url, loginPass, cookieContainer);
        if (data == null)
        {
            if (i == 0)
            {
                using (BotContext db = new BotContext())
                {
                    user.Cookie.Value = null;
                    db.Users.Update(user);
                    db.SaveChanges();
                }
                continue;
            }
            else
            {
                return null;
            }
        }
    }
}

```

Рис. Л.5 – Отримання оцінок у JSON

```

string unicodeText = getString(data);

if (unicodeText.Contains("DOCTYPE"))
{
    if (i == 0)
    {
        using (BotContext db = new BotContext())
        {
            user.Cookie.Value = null;
            db.Users.Update(user);
            db.SaveChanges();
        }
        continue;
    }
    else
    {
        return null;
    }
}
return unicodeText;
}
return string.Empty;
}

```

Рис. Л.6 – Валідація

```
private static string GetString(HttpWebResponse response)
{
    string responseString = null;
    using (var reader = new StreamReader(response.GetResponseStream()))
    {
        responseString = reader.ReadToEnd();
    }
    return responseString;
}
```

Рис. Л.7 – Десеріалізація відповіді у строку

```
private static CookieContainer deSerializeQueryFilters(byte[] items)
{
    BinaryFormatter bf = new BinaryFormatter();
    try
    {
        using (MemoryStream ms = new MemoryStream())
        {
            ms.Write(items, 0, items.Length);
            ms.Position = 0;

            return bf.Deserialize(ms) as CookieContainer;
        }
    }
    catch (Exception)
    {
        return null;
    }
}
```

Рис. Л.8 – Десеріалізація cookie у об'єкт

```

class Document
{
    public static byte[] CreateFile<T>(List<Column> columns, List<T> data) where T : IRoot
    {
        iTextSharp.text.Document doc = new iTextSharp.text.Document(PageSize.A4.Rotate());

        var file = new MemoryStream();
        var writer = PdfWriter.GetInstance(doc, file);

        float[] widths = columns.Select(x => x.Width).ToArray();
        PdfPTable table = new PdfPTable(widths.Length);
        table.WidthPercentage = 100;
        table.SetWidths(widths);

        foreach (var i in columns)
        {
            addCell(i.Text, "column");
        }

        foreach (var item in data)
        {
            string dangerLevel = "column";
            try
            {
                dangerLevel = item.GetType().GetProperty("classes").GetValue(item).ToString();
            }
            catch (Exception)
            {
            }
        }
    }
}

```

Рис. М.1 – Метод створення документу

```

        foreach (var column in columns)
        {
            try
            {
                column.Value = item.GetType().GetProperty(column.JsonName).GetValue(item).ToString();
            }
            catch (Exception)
            {
                column.Value = "";
            }
            addCell(column.Value, dangerLevel);
        }
    }
    try
    {
        doc.Open();
        doc.Add(table);
        doc.Close();
        writer.Close();
        file.Close();
    }
    catch (Exception)
    {
    }
    return file.ToArray();
}

void addCell(string text, string dangerLevel)
{
    var cell = GetCell(text, dangerLevel);
    table.AddCell(cell);
}
}

```

Рис. М.2 – Побудова таблиці

```

public static PdfPCell GetCell(string text, string dangerLevel)
{
    string ttf = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF");
    var baseFont = BaseFont.CreateFont(ttf, BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
    var font = new Font(baseFont, Font.DEFAULTSIZE, Font.NORMAL);

    var cell = new PdfPCell(new Phrase(text, font));
    cell.BackgroundColor = getColor(dangerLevel);

    return cell;
}

```

Рис. М.3 – Побудова комірки

```

private static BaseColor getColor(string dangerLevel)
{
    switch (dangerLevel)
    {
        case "danger":
            return new BaseColor(220, 91, 96);
        case "info":
            return new BaseColor(144, 200, 238);
        case "success":
            return new BaseColor(169, 238, 144);
        case "warning":
            return new BaseColor(255, 252, 107);
        case "column":
            return new BaseColor(255, 255, 255);
        default:
            return new BaseColor(212, 212, 212);
    }
}

```

Рис. М.4 – Розфарбовування комірки

```

class Column
{
    public string Text { get; set; }

    public string JsonName { get; set; }

    public float Width { get; set; }

    public string Value { get; set; }

    public Column(string text, string name, float width)
    {
        Value = string.Empty;
        Text = text;
        JsonName = name;
        Width = width;
    }
}

```

Рис.. Н.1 – Модель колонки

```

public class OffsetLesson : IRoot
{
    public string kod_st { get; set; }
    public string Kod_disc { get; set; }
    public string NomSem { get; set; }
    public string NomKont { get; set; }
    public string Data_kontr { get; set; }
    public string ball { get; set; }
    public string z_ball { get; set; }
    public string Godin { get; set; }
    public string rezult { get; set; }
    public string ECTS { get; set; }
    public string NAC { get; set; }
    public string kod_spec { get; set; }
    public string kod_disc { get; set; }
    public string typ { get; set; }
    public string disc { get; set; }
    public string god { get; set; }
    public string kred { get; set; }
    public string lek { get; set; }
    public string lab { get; set; }
    public string pra { get; set; }
    public string sam { get; set; }
    public string ksam { get; set; }
    public string VidOcin { get; set; }
    public string Semestr { get; set; }
    public string kafedra { get; set; }
    public string vLek { get; set; }
    public string vAss1 { get; set; }
    public string vAss2 { get; set; }
    public string classes { get; set; }
}

```

Рис Н.2 – Модель JSON файлу частина 1

```

public class Data
{
    public List<OffsetLesson> offset_lessons { get; set; }
    public List<ModuleControllLesson> module_controll_lessons { get; set; }
}

public class Root
{
    public int status { get; set; }
    public Data data { get; set; }
    public string msg { get; set; }
}

public class Datum : OffsetLesson
{
    public string Rik { get; set; }
    public string td_class { get; set; }
}

public class CurriculumRoot
{
    public List<Datum> data { get; set; }
    public int status { get; set; }
    public string msg { get; set; }
}

public interface IRoot
{
}

```

Рис. Н.3 – Модель JSON файлу частина 2

```

namespace KNUTEBot.Models
{
    class BotContext : DbContext
    {
        public DbSet<User> Users { get; set; }
        public DbSet<Cookie> Cookies { get; set; }

        public BotContext()
        {
            Database.EnsureCreated();
        }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            var connectionStringBuilder = new SqlConnectionStringBuilder { DataSource = "BotDb.db" };
            var connection = new SqlConnection(connectionStringBuilder.ToString());
            optionsBuilder.UseLazyLoadingProxies().UseSqlite(connection);
        }
    }
}

```

Рис. Н.4 – Клас контексту бази даних

```

public class User
{
    [Key]
    public long ChatId { get; set; }

    public string Login { get; set; }

    public string Password { get; set; }

    public State State { get; set; }

    public virtual Cookie Cookie { get; set; }

    public User()
    {
        Cookie = new Cookie();
    }
}

```

Рис. Н.5 – Модель класу користувача

```

public class Cookie
{
    [Key]
    [ForeignKey("User")]
    public long ChatId { get; set; }

    public byte[] Value { get; set; }

    public virtual User User { get; set; }
}

```

Рис. Н.6 – Модель класу cookie