

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на тему

РОЗРОБКА ЧАТ-БОТА СТУДЕНТА ПУЕТ

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Береговий Ростислав Романович
_____ « ____ » _____ 2023 р.
(підпис)

Науковий керівник зав. каф., к.ф.-м.н., Ольховська Олена Володимирівна
_____ « ____ » _____ 2023 р.
(підпис)

Рецензент

ПОЛТАВА 2023 р.

ЗАТВЕРДЖУЮЗавідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 202__ р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему Розробка Чат-бота студента ПУЕТ

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Береговий Ростислав Романович

Затверджена наказом ректора № 144-Н від «01» вересня 2022 р.

Термін подання студентом роботи «___» _____ 202__ р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, системи дистанційного навчання.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Типи Telegram-ботів

2.2. Системи, середовища програмування, середовища для розробки програмного забезпечення

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд та аналіз відомих чат-ботів

3.2. Аналіз відомих освітніх чат-ботів

3.3. Актуальність використання чат-ботів в освітньому процесі

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Алгоритм роботи чат-бота

4.2. Робота чат-боту

4.3. Програмна реалізація чат-бота

4.4. UML діаграма прецедентів

4.5. Тестування

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТОК А. КОД ПРОГРАМИ

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Ольховська О.В.		
Інформаційний огляд	Ольховська О.В.		
Теоретична частина	Ольховська О.В.		
Практична частина	Ольховська О.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постанова задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 202__ р.

Здобувач вищої освіти _____ Береговий Ростислав Романович
(підпис)Науковий керівник _____ зав. каф., к.ф.-м.н., Ольховська Олена Володимирівна
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «____» _____ 2023 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

Зав. кафедрою _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
«_____» _____ 202__ р.

Погоджено

Науковий керівник _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
«_____» _____ 202__ р.

План

кваліфікаційної роботи на тему
Розробка Чат-бота студента ПУЕТ
зі спеціальності 122 Комп'ютерні науки
освітня програма 122 «Комп'ютерні науки»
ступеня бакалавр
Прізвище, ім'я, по батькові Береговий Ростислав Романович

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД****2.1. Типи Telegram-ботів**

2.2. Системи, середовища програмування, середовища для розробки програмного забезпечення

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА**3.1. Огляд та аналіз відомих чат-ботів****3.2. Аналіз відомих освітніх чат-ботів****3.3. Актуальність використання чат-ботів в освітньому процесі****РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА****4.1. Алгоритм роботи чат-бота****4.2. Робота чат-боту****4.3. Програмна реалізація чат-бота****4.4. UML діаграма прецедентів****4.5. Тестування****ВИСНОВКИ****СПИСОК ЛІТЕРАТУРИ****ДОДАТОК А. КОД ПРОГРАМИ**

Здобувач вищої освіти _____ Ростислав БЕРЕГОВИЙ
«_____» _____ 202__ р.

РЕФЕРАТ

Записка: 46 с., 21 рис., 1 додаток, 16 джерел.

Розробка Чат-бота студента ПУЕТ.

Об'єкт розробки – алгоритм, UML діаграма, розробка програмного забезпечення тренажера.

Мета роботи – проектування та реалізація телеграм чат-боту, який може взаємодіяти з користувачами за допомогою текстових повідомлень та надавати корисну інформацію.

Предмет розробки – проектування чат-боту студента ПУЕТ.

Методи дослідження – чат-бот, середовище візуальної розробки програм Visual Studio Code та мова програмування Python.

Зроблено інформаційний огляд матеріалу по темі чат-боту. Розроблено алгоритм чат-боту студента ПУЕТ.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	8
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	10
2.1. Типи Telegram-ботів	10
2.2. Системи, середовища програмування, середовища для розробки програмного забезпечення	13
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	16
3.1. Огляд та аналіз відомих чат-ботів.....	16
3.2. Аналіз відомих освітніх чат-ботів.....	17
3.3. Актуальність використання чат-ботів в освітньому процесі	18
РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ	20
4.1. Алгоритм роботи чат-бота	20
4.2. Робота чат-боту	21
4.3. Програмна реалізація чат-бота	30
4.4. UML діаграма прецедентів	35
4.5. Тестування	38
ВИСНОВКИ.....	40
СПИСОК ЛІТЕРАТИ.....	41
ДОДАТОК А. КОД ПРОГРАМИ.....	44

ВСТУП

У сучасному світі інформаційні технології активно використовуються в різних сферах життя, у тому числі й у навчанні. Розвиток мобільних технологій та популярність месенджерів призвели до зростання популярності чат-ботів як зручного інструменту комунікації з користувачами. Телеграм чат-боти відносяться до найпопулярніших типів чат-ботів, які можуть вирішувати різні завдання та забезпечувати швидкий доступ до інформації.

Мета проекту – проектування та реалізація телеграм чат-боту, який може взаємодіяти з користувачами за допомогою текстових повідомлень та надавати корисну інформацію.

Об'єктом розробки є алгоритм роботи чат-бота.

Предмет розробки — чат-бот Телеграм.

Перелік використаних методів полягає в застосуванні граматик, мов, що задаються граматиками.

Проект складається з трьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі розглянуто типи чат-ботів, системи, середовища програмування, середовища для розробки програмного забезпечення. У третьому розділі представлено огляд матеріалу з теми чат-бот.

У додатку міститься диск з матеріалами проекту.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Головним завданням курсового проекту з фаху – є створення елементів чат-боту, програмування, елементів програмного забезпечення тренажеру з теми «Проектування Чат-бота студента ПУЕТ»

Ціллю даної дипломної роботи є дослідження та розробка телеграм чат-бота з метою впровадження його в реальні умови. Головним завданням роботи є розробка ефективного чат-бота, здатного забезпечувати якісну комунікацію з користувачами та забезпечувати їх потреби шляхом надання необхідної інформації, виконання різних завдань та забезпечення зручного інтерфейсу взаємодії.

Для досягнення поставленої мети в роботі будуть розглянуті основні принципи роботи чат-ботів, методи та алгоритми їх розробки. Використовуючи знання з програмування та розробки програмного забезпечення, буде реалізовано телеграм чат-бота, що відповідає поставленим вимогам та задачам.

Результатом дипломної роботи буде готовий телеграм чат-бот, який демонструє високу якість комунікації з користувачами та забезпечує потрібні функції його використання.

Завдання дипломного проекту з фаху:

- Розглянути матеріал з теми «Проектування Чат-бота студента ПУЕТ»;
- Розробити елементи чат-бота;
- Запрограмувати елементи чат-бота;
- Переглянути результати створення елементів чат-бота.

Основними етапами розробки телеграм боту є:

- визначення аудиторії;
- розробка концепції;
- формування технічного завдання;

- розробка;
- тестування;
- відлагодження;
- реєстрація та реліз;
- взаємодія з користувачами;
- періодичні оновлення.

Основні вимоги до Telegram чат-боту включають:

- Взаємодія з користувачем: Бот повинен мати можливість приймати повідомлення від користувачів і відправляти їм відповіді.
- Обробка повідомлень: Бот повинен мати функціонал для обробки отриманих повідомлень. Це може включати аналіз тексту, розпізнавання команд, обробку зображень або відео тощо.
- Маршрутизація запитів: Чат-бот може мати різні функції або послуги. Важливо мати систему маршрутизації запитів, щоб керувати повідомленнями користувачів до відповідних функцій.
- Інтеграція зі зовнішніми сервісами: Чат-бот може виконувати запити до зовнішніх API або сервісів, таких як бази даних, онлайн-платформи або інші API-інтерфейси.
- Аналітика та статистика: Важливо відстежувати використання бота, кількість користувачів, їх запити, ефективні

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Типи Telegram-ботів

Чат-боти класифікують за кількома критеріями: алгоритму роботи, формату взаємодії та способу призначення.

Залежно від алгоритму роботи чат-боти бувають:

- обмежені – відповідають на конкретні запити користувачів за заздалегідь розробленим скрипту, мають обмежену кількість відповідей;
- саморозвиваючі – працюють на основі штучної нейронної мережі, можуть розуміти суть бесіди і вести реалістичні розмови з покупцем, з часом навчаються і дають більш релевантні відповіді на запити.

За форматом взаємодії з користувачем існують такі типи чат-ботів:

- Кнопковий. Використовується найчастіше в месенджерах. Комунікація з клієнтом відбувається через кнопки з варіантами дій. Бот реагує на них, як на команди, і пропонує користувачеві уточнюючі кнопки або дає відповідь на поставлене запитання;



Рисунок 2.1.1 – Кнопковий чат-бот

- **Текстовий.** Самий функціональний вид віртуального співрозмовника. Спілкування з ним близько до людського, так як робот розпізнає запит, аналізує інформацію і підбирає для нього найбільш релевантний відповідь із заготовлених;

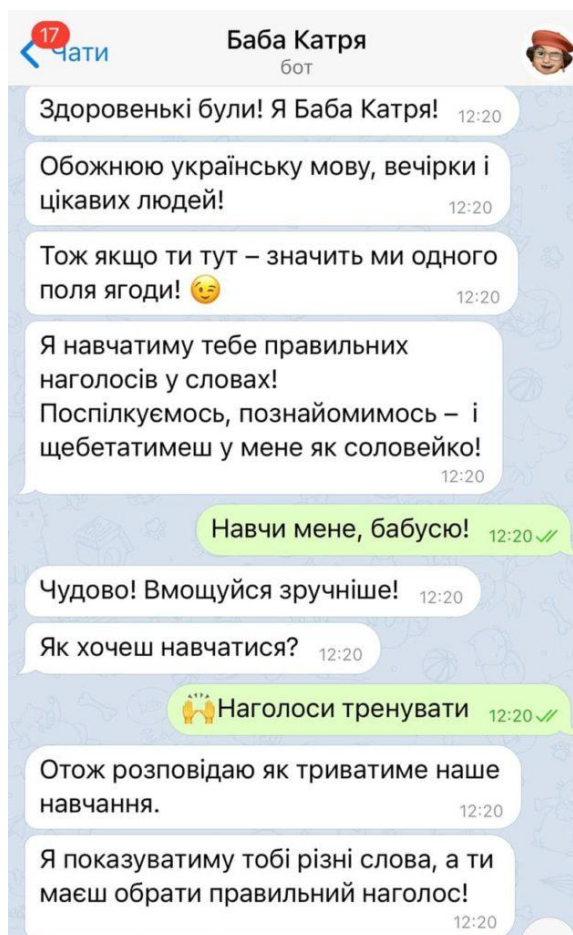


Рисунок 2.1.2 – Текстовий чат-бот

- **Вбудований (inline).** З'являється усередині діалогу в месенджері після виклику і пропонує варіанти дій. Результатом можна поділитися зі співрозмовником, з яким йшов діалог. Зазвичай використовується для пошуку відповідних локацій, замовлення їжі та інших схожих послуг.



Рисунок 2.1.3 – Вбудований чат-бот

- Комунікаційні боти відповідають своїй назві і забезпечують спілкування компанії з клієнтами. Вони можуть бути дуже примітивними. Наприклад, відповідати на запитання й відповіді шаблонними фразами, пропонувати зворотний дзвінок або переадресовувати на справжнього менеджера. Але також це можуть бути і маркетингові чат-боти. Вони за замовчуванням виконують функцію двостороннього каналу зв'язку з клієнтами. Надають їм інформацію про послуги та спецпропозиції, відповідають на зустрічні питання по заздалегідь вибудованої архітектурі.
- Функціональні чат-боти є заміною повноцінним мобільним додаткам. Уже зараз більшість програм дозволяє в одному вікні шукати, консультуватися, бронювати, купувати, виконувати банківські операції, а також пропонувати користувачам інтерактивні дії і персоналізовані відповіді. Далі спектр операцій можна розширювати до нескінченності [2].

2.2. Системи, середовища програмування, середовища для розробки програмного забезпечення

Python — це мова програмування, яка широко використовується в інтернет-додатках, розробці програмного забезпечення, науці даних і машинному навчанні (ML). Розробники використовують Python, тому що він ефективний, простий у вивченні та працює на різних платформах. Програми мовою Python можна завантажити безкоштовно, вони сумісні з усіма типами систем та підвищують швидкість розробки.

Мова Python має такі переваги:

- Розробники можуть легко читати та розуміти програми на Python, оскільки мова має базовий синтаксис, схожий на синтаксис англійської.
- Python допомагає розробникам бути більш продуктивними, оскільки вони можуть писати програми на Python, використовуючи менше рядків коду, ніж іншими мовами.
- Python має велику стандартну бібліотеку, що містить багаторазові коди практично для будь-якого завдання. В результаті розробникам не потрібно писати код із нуля.
- Розробники можуть легко поєднувати Python з іншими популярними мовами програмування: Java, C та C++.
- Активна спільнота Python складається з мільйонів розробників з усього світу. У разі виникнення проблем співтовариство допоможе в їх вирішенні.
- Крім того, в Інтернеті є безліч корисних ресурсів для вивчення Python. Наприклад, ви можете легко знайти відеоролики, навчальні посібники, документацію та посібники для розробників.
- Python можна переносити на різні операційні системи: Windows, MacOS, Linux і Unix [3].

Бот – це спеціалізований робот-помічник. Він допомагає виконувати будь-які рутинні заняття. Боти здатні реалізовувати програмні коди, які будуть відповідати за активацію різних команд користувачів. Листування з таким ПЗ здійснюється безпосередньо через чат. Клієнт дає роботу команди, які він обробляє і виконує в режимі 24/7. Ключове завдання “робота” – дати відповідь питання клієнта, спираючись на задану програму. З допомогою них вдається економити як час, а й інші ресурси.

До переваг ботів Телеграм відносять:

- цілодобову допомогу – функціонування bots припиняють лише у разі аварій на серверах, які трапляються вкрай рідко;
- зручність та простоту використання – для більшості команд достатньо вибрати із запропонованого списку відповідну операцію;
- миттєве отримання відповіді;
- відсутність вимог до потужності задіяного пристрою – це з тим, що з роботи роботів використовуються можливості сторонніх серверів;
- високий рівень безпеки;
- відсутність необхідності ініціалізації додаткового ПЗ для запуску виду «помічника», що розглядається [4].

Visual Studio Code - засіб для створення, редагування та налагодження сучасних вебзастосунків і програм для хмарних систем. Visual Studio Code поширюється безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

Компанія Microsoft представила Visual Studio Code в квітня 2015 на конференції Build 2015. Це середовище розробки стало першим кросплатформовим продуктом в лінійці Visual Studio.

За основу для Visual Studio Code використовуються напрацювання вільного проекту Atom розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, який використовує браузерні движок Chromium і Node.js. Примітно, що про використання напрацювань вільного

проекту Atom і на сайті Visual Studio Code, і в пресрелізи, і в офіційному блозі немає згадки.

Редактор містить вбудований відладчик, інструменти для роботи з Git і засоби рефакторинга, навігації по коду, автозаповнення типових конструкцій і контекстної підказки. Продукт підтримує розробку для платформ ASP.NET і Node.js, і позиціонується як легкий рішення, що дозволяє обійтися без повного інтегрованого середовища розробки. Серед підтримуваних мов і технологій:

- JavaScript;
- C ++;
- C #;
- TypeScript;
- PHP;
- Python;
- Java та інші [5].

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд та аналіз відомих чат-ботів

Огляд та аналіз відомих чат-ботів Telegram:

- **@Manybot:** Manybot є одним з найпопулярніших та функціональних чат-ботів в Telegram. Він надає користувачам можливість створювати власні чат-боти з різними функціями, такими як опитування, розсилка повідомлень, автоматичні відповіді тощо. Manybot пропонує зручний інтерфейс для налаштування бота та добре підходить для некодерів.
- **@BotFather:** BotFather є офіційним ботом Telegram, який дозволяє користувачам створювати своїх власних чат-ботів. Він надає API-інтерфейс та можливості для налаштування різних параметрів бота, таких як назва, аватар, команди та багато іншого. BotFather є відмінним вибором для розробників, які мають досвід в програмуванні.
- **@PollBot:** PollBot є добре відомим чат-ботом для проведення опитувань в Telegram. Він дозволяє користувачам створювати опитування з різними типами питань та варіантами відповідей. Користувачі можуть голосувати за варіанти, а результати опитування можуть бути надіслані в груповий чат або приватно.
- **@TriviaBot:** TriviaBot є чат-ботом, який пропонує користувачам викликати розумові гри та питання-відповіді. Він надає різні категорії питань, такі як спорт, історія, наука тощо, і дозволяє користувачам змагатися з іншими у відповідях на запитання.
- **@WeathermanBot:** WeathermanBot є простим і корисним чат-ботом, який надає поточну погоду та прогноз на кілька днів вибраного місця. Користувачі можуть запитувати погодні умови для свого місцезнаходження або вказаного місця.

Після огляду відомих чат-ботів Telegram можна зробити деякі спостереження. Багато з цих ботів пропонують широкий спектр функцій, включаючи створення опитувань, голосувань, грошових операцій, погодних звітів та багато іншого. Вони надають зручний спосіб взаємодії з користувачами та автоматизують багато завдань. Деякі з цих ботів також надають API-інтерфейси для розробників, що дозволяє створювати власні чат-боти з розширеними функціями та інтеграцією з іншими сервісами.

Аналізуючи ці відомі чат-боти, можна виявити певні схожі та відмінні риси в їх функціональності, взаємодії з користувачами та варіантах використання. Все це відкриває широкі можливості для розвитку та вдосконалення власного телеграм чат-бота з урахуванням потреб користувачів та конкретної сфери його використання.

3.2. Аналіз відомих освітніх чат-ботів

Освітні чат-боти стали популярними інструментами в сфері навчання та освіти. Вони поєднують у собі штучний інтелект із можливістю комунікації з користувачами через текстові повідомлення, дозволяючи забезпечити доступ до освітніх матеріалів та відповідати на запитання в режимі реального часу. Давайте розглянемо кілька відомих освітніх чат-ботів і їхні особливості:

- **Duolingo Chatbot:** Duolingo є однією з найпопулярніших мовних платформ, а їхній чат-бот допомагає користувачам вивчати іноземні мови. Він пропонує інтерактивні уроки з різних тем та рівнів складності, де користувачі можуть відповідати на питання, виконувати завдання та отримувати негайну зворотну відповідь. Чат-бот користується популярністю завдяки своїй доступності та ефективності.
- **Brainly Bot:** Brainly є платформою, де користувачі можуть задавати питання з різних навчальних предметів, а інші користувачі надають на них відповіді. Brainly Bot допомагає автоматизувати цей процес, надаючи

користувачам швидкі відповіді на їхні запитання. Він використовує штучний інтелект для пошуку в базі даних та надає відповіді з великою кількістю інформації, що допомагає учням розібратися у складних питаннях.

- **Mitsuku:** Mitsuku є одним з найвідоміших чат-ботів, розробленим для загального спілкування та розваги. Він має широкий спектр функцій, включаючи відповіді на питання, розповідь жартів та навіть проведення філософських дискусій. Він використовує навчання з підсиленням для постійного вдосконалення своїх відповідей та може викликати в користувачів враження розмови з реальною особою.

- **Woebot:** Woebot є освітнім чат-ботом, спрямованим на підтримку психічного здоров'я. Він використовує техніки психотерапії, такі як підтримка на основі когнітивно-поведінкової терапії, для надання порад і підтримки користувачам, які мають стрес або інші емоційні проблеми. Woebot також відстежує настрій користувачів та надає практичні рекомендації для поліпшення їхнього самопочуття.

Ці освітні чат-боти представляють лише деякі приклади того, як штучний інтелект може бути використаний для покращення процесу навчання та надання підтримки учням та студентам. Вони дозволяють користувачам отримувати індивідуальну підтримку, навчатися власним темпом та мати доступ до знань у будь-який зручний для них час. За допомогою таких чат-ботів можна покращити доступність освіти та навчання, забезпечити персоналізований підхід та сприяти активній взаємодії між користувачами та технологіями.

3.3. Актуальність використання чат-ботів в освітньому процесі

Використання чат-ботів в освітньому процесі є актуальним і має значний потенціал у багатьох аспектах. Ось деякі з переваг і актуальних аспектів використання чат-ботів у навчанні:

- **Персоналізація та індивідуалізація:** Чат-боти можуть надати індивідуальну підтримку та навчання для кожного учня. Вони можуть адаптуватися до потреб та рівня знань кожного учня і надавати наступні кроки або додаткові матеріали, що відповідають їхнім потребам. Це допомагає кожному учневі розвиватися власним темпом і досягати кращих результатів.

- **Негайна зворотна відповідь:** Чат-боти забезпечують миттєву відповідь на запитання користувача. Учні можуть одразу отримати пояснення або допомогу з будь-яким навчальним матеріалом. Це зменшує залежність від вчителя або викладача та дозволяє учням самостійно вирішувати свої проблеми.

- **Доступність та гнучкість:** Чат-боти доступні 24/7 і можуть бути використані в будь-який зручний для учня час. Вони можуть бути доступні на різних платформах, включаючи мобільні пристрої, що дозволяє учням вчитися з будь-якого місця і в будь-який час.

- **Підтримка самостійного навчання:** Чат-боти можуть стимулювати самостійне навчання, спонукати учнів до активної взаємодії та пошуку відповідей на власні запитання. Вони можуть надати корисні матеріали, завдання та викликати інтерес до дослідження нових тем.

- **Оцінювання та відстеження прогресу:** Чат-боти можуть допомагати в оцінюванні знань та навичок учнів, а також відстежувати їх прогрес. Вони можуть надавати зворотний зв'язок щодо помилок та вказівки для поліпшення.

- **Психологічна підтримка:** Деякі освітні чат-боти спеціалізуються на психологічній підтримці учнів, допомагаючи їм управляти стресом, тривогами та іншими емоційними проблемами, що можуть впливати на навчання.

Використання чат-ботів в освітньому процесі допомагає покращити доступність, ефективність та індивідуальний підхід до навчання. Вони створюють нові можливості для учнів та студентів і сприяють покращенню якості освіти.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1. Алгоритм роботи чат-бота

Алгоритм роботи Telegram чат-бота може бути описаний таким чином:

- **Реєстрація та налаштування бота:** щоб створити чат-бота в Telegram, необхідно зареєструватися в Telegram та створити новий бот через BotFather. Потім потрібно налаштувати параметри бота, такі як його ім'я, аватар, опис та команди.
- **Налаштування API-ключа:** після створення бота в Telegram необхідно налаштувати API-ключ для взаємодії з Telegram API. Це дозволить боту отримувати та надсилати повідомлення через Telegram.
- **Визначення функціональності:** визначте, які функції ви хочете додати до вашого робота. Деякі можливості, які можуть бути реалізовані в Telegram-боті, включають відповідь на команди, відправлення повідомлень, завантаження файлів, роботу з базою даних і багато іншого.
- **Написання коду:** напишіть код для вашого бота за допомогою мови програмування, яка підтримує Telegram API (наприклад, Python). Використовуйте відповідну бібліотеку, яка допоможе вам взаємодіяти з Telegram API.
- **Тестування:** протестуйте ваш бот, щоб переконатися, що він працює коректно і виконує всі необхідні функції. Перевірте також, що бот безпечний і не загрожує користувачам.
- **Розгортання:** розгорніть ваш бот на сервері, щоб він міг бути доступний для користувачів.

4.2. Робота чат-боту

На головному екрані нас зустрічає інформаційний надпис про бота та велика кнопка «Розпочати». (Рисунок 3.2.1)

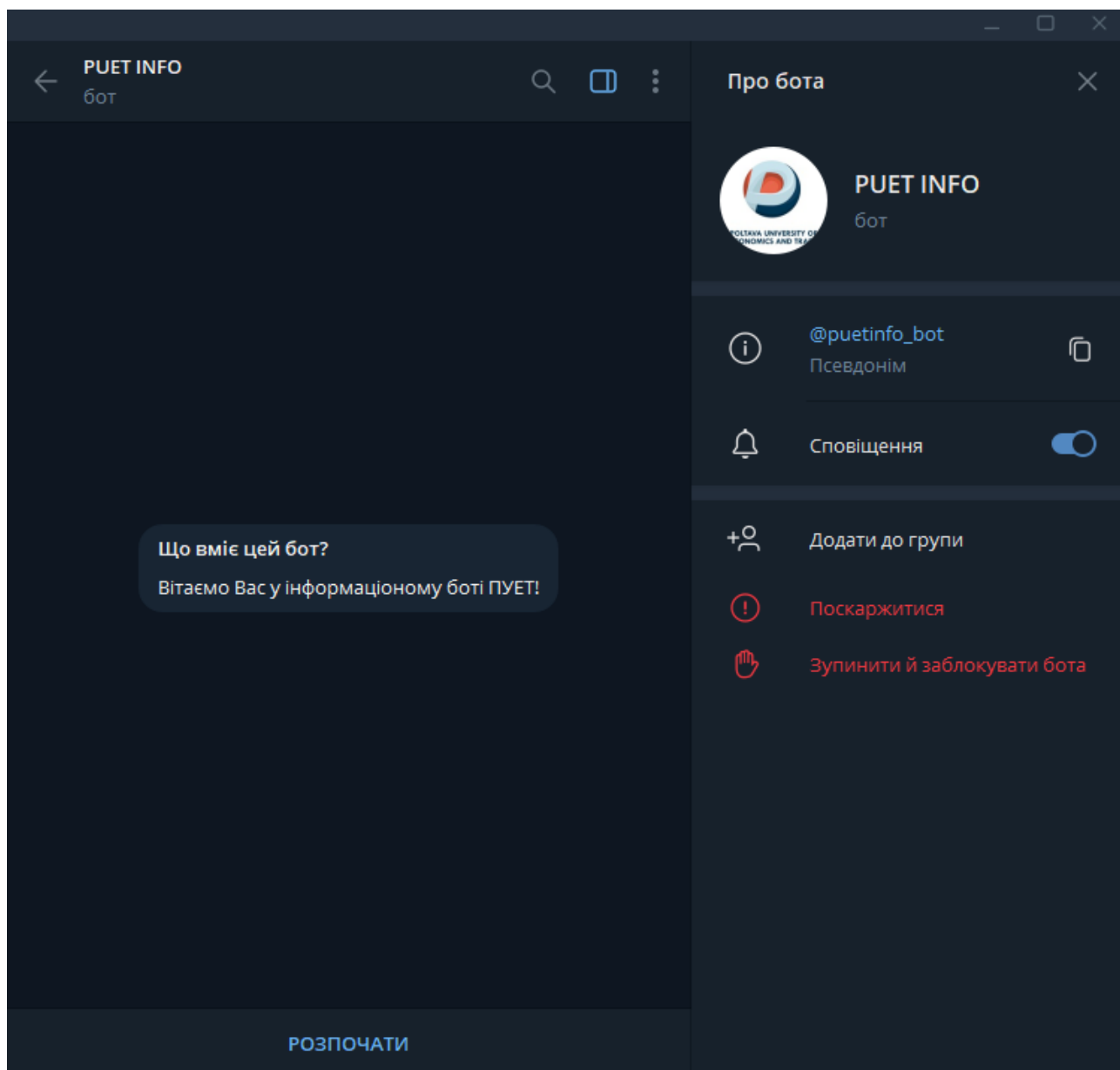


Рисунок 4.2.1 – Головний екран

Після натиснення кнопки «Розпочати», у чат автоматично відправляється команда, яка розпочинає роботу бота. Внизу користувач може ознайомитися із

кнопками із доступними для нього функціями. (Рисунок 4.2.2) Основними функціями для користувача є:

- Оплата.
- Розклад.
- Дистанційний курс.
- ChatGPT.
- Пошук викладача.
-

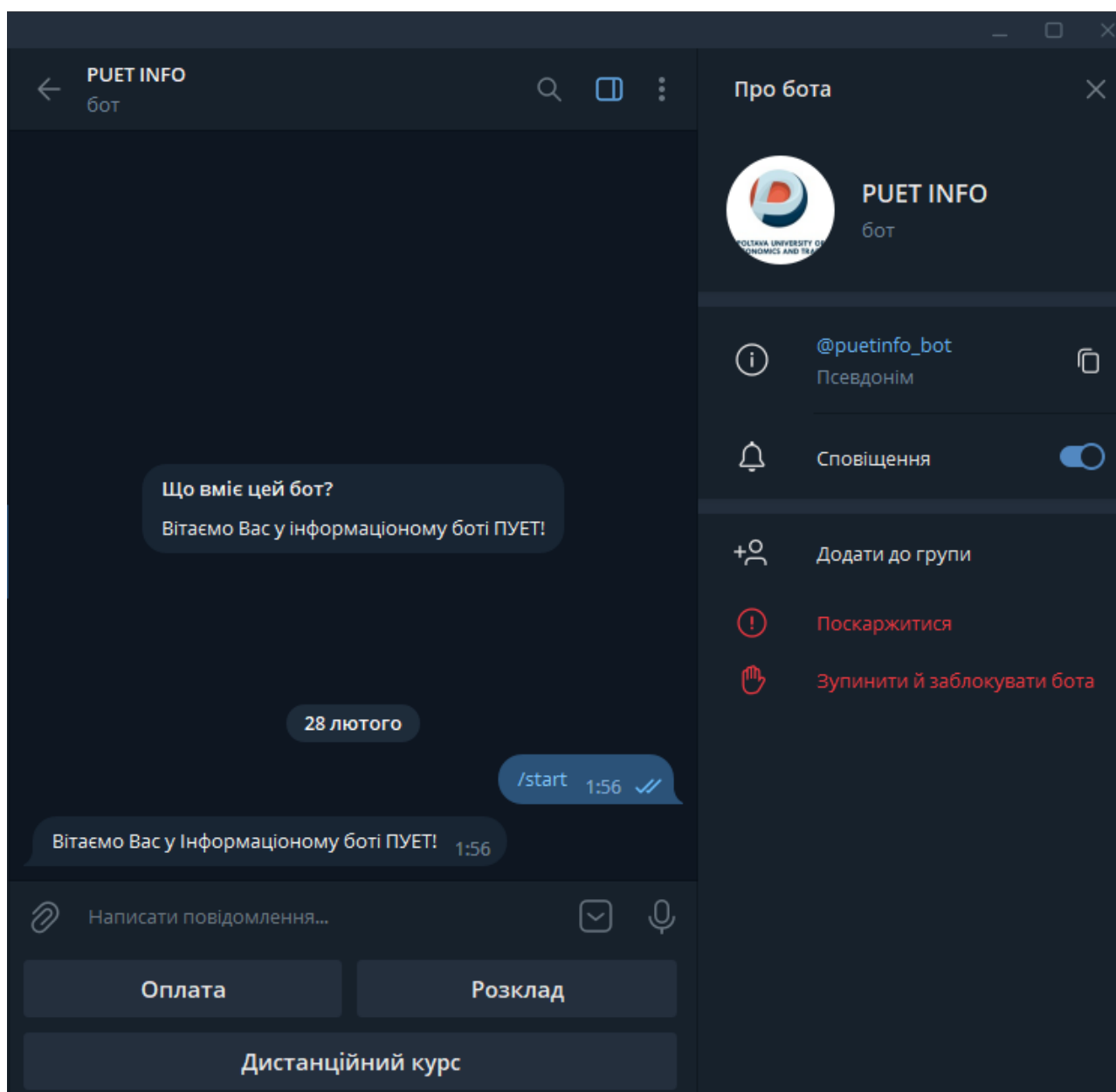


Рисунок 4.2.2 – Основні функції

Оплата – це кнопка, що викликає повідомлення, під яким є кнопки із можливими видами оплати. (Рисунок 4.2.3) Серед цих кнопок, користувач може бачити усі можливі види оплати за навчання, а саме:

- Приват банк.
- Кредіт Агріколь.
- Ощадбанк.
- ОТП банк.
- Укрсиббанк.
- Укрексим.
- Посилання на сайт.

При натисканні на один із видів банку, користувач отримує повідомлення із реквізитами для оплати. Реквізити відправляються окремо у кожному повідомленні для зручного копіювання тексту, у разі потреби. (Рисунок 4.2.4).

При натисканні на кнопку «Посилання на сайт», користувач отримує посилання на сайт Приват банку. На сайті користувач має можливість обрати зручний банк для сплати та не вводити реквізити, а лише ввести свої данні. (Рисунок 4.2.5)

Дистанційний курс – це кнопка, що викликає повідомлення із посиланням на дистанційний курс. (Рисунок 4.2.7)

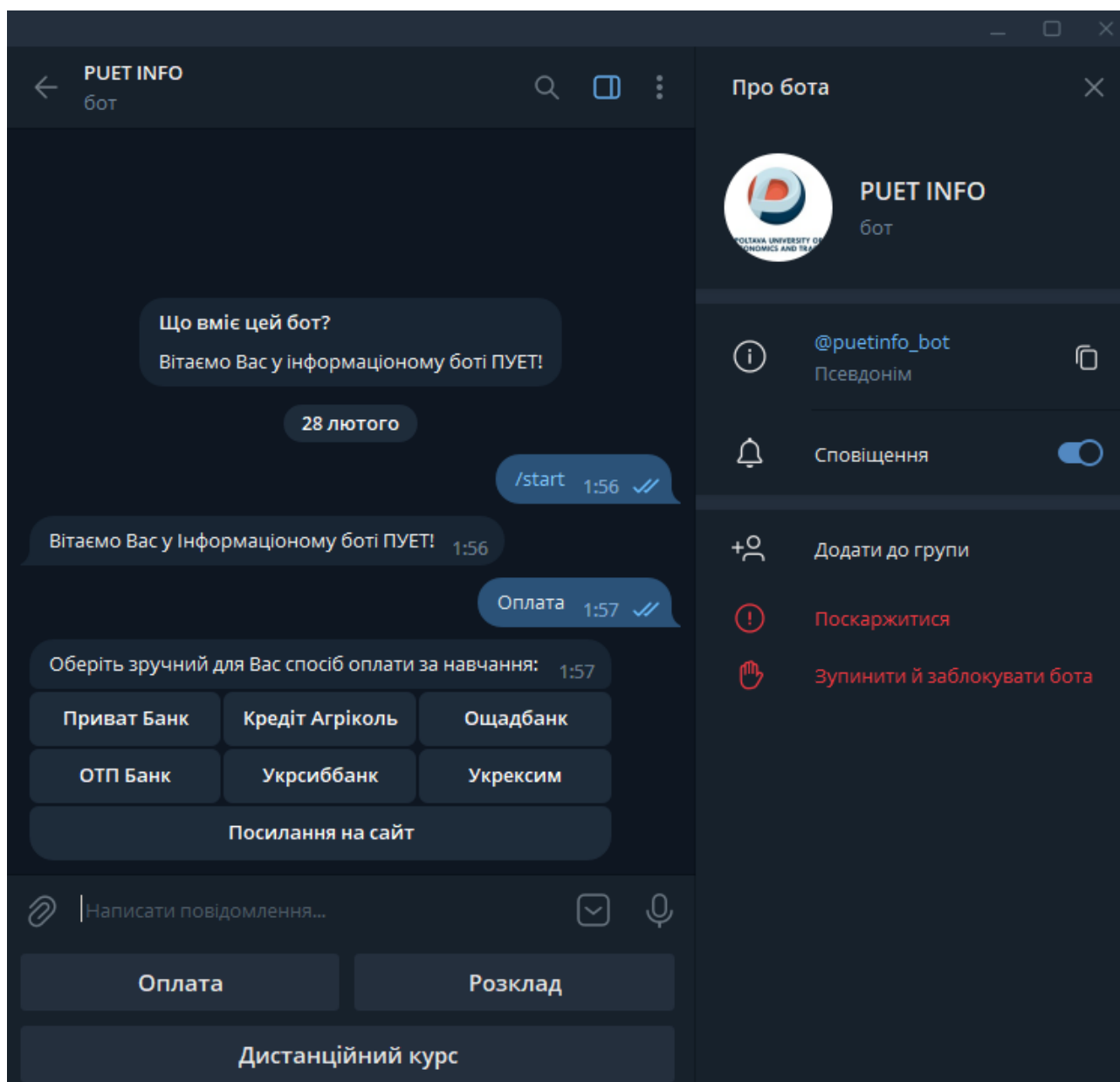


Рисунок 4.2.3 – Оплата за навчання

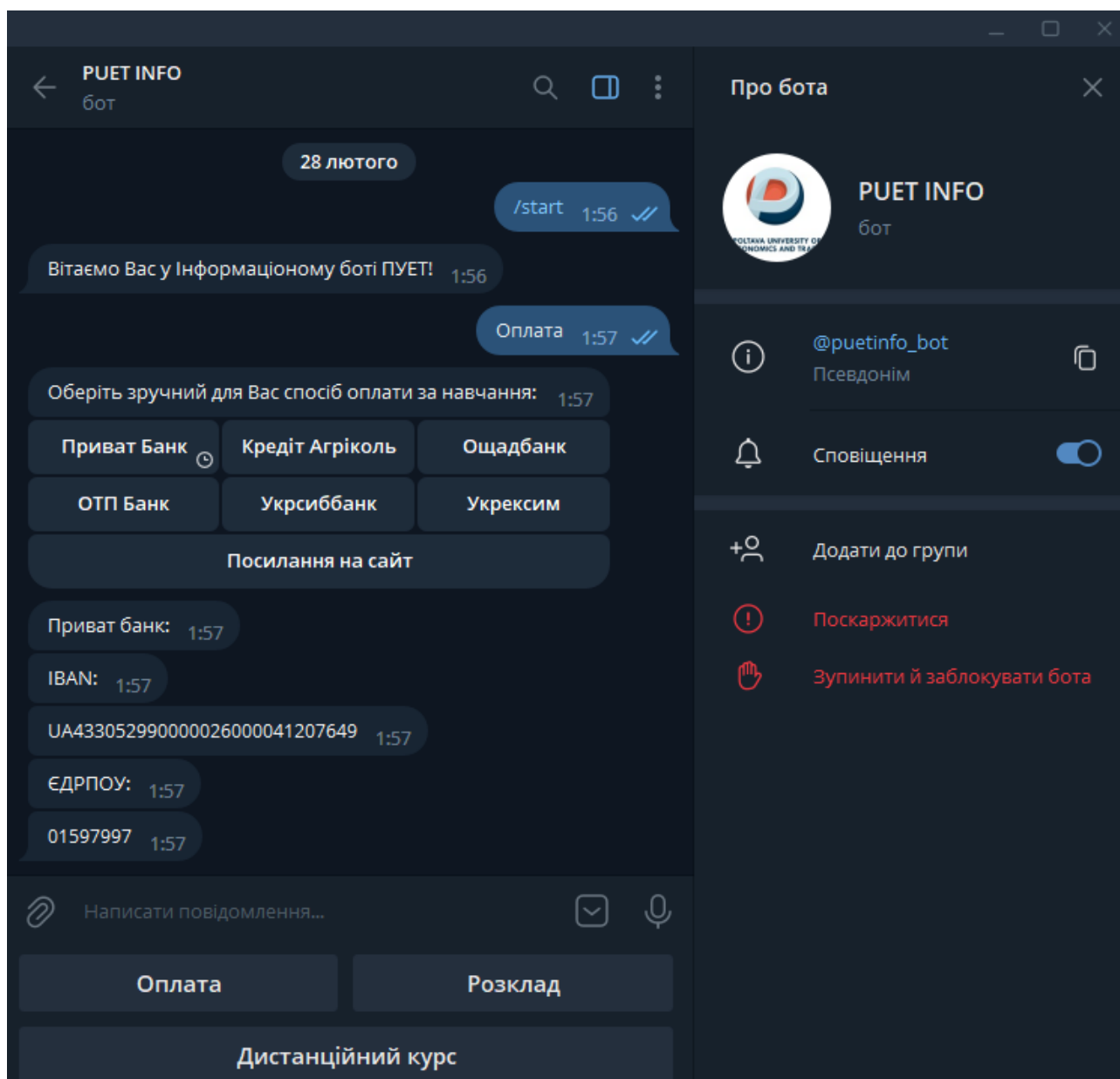


Рисунок 4.2.4 – Приклад повідомлення реквізитів оплати

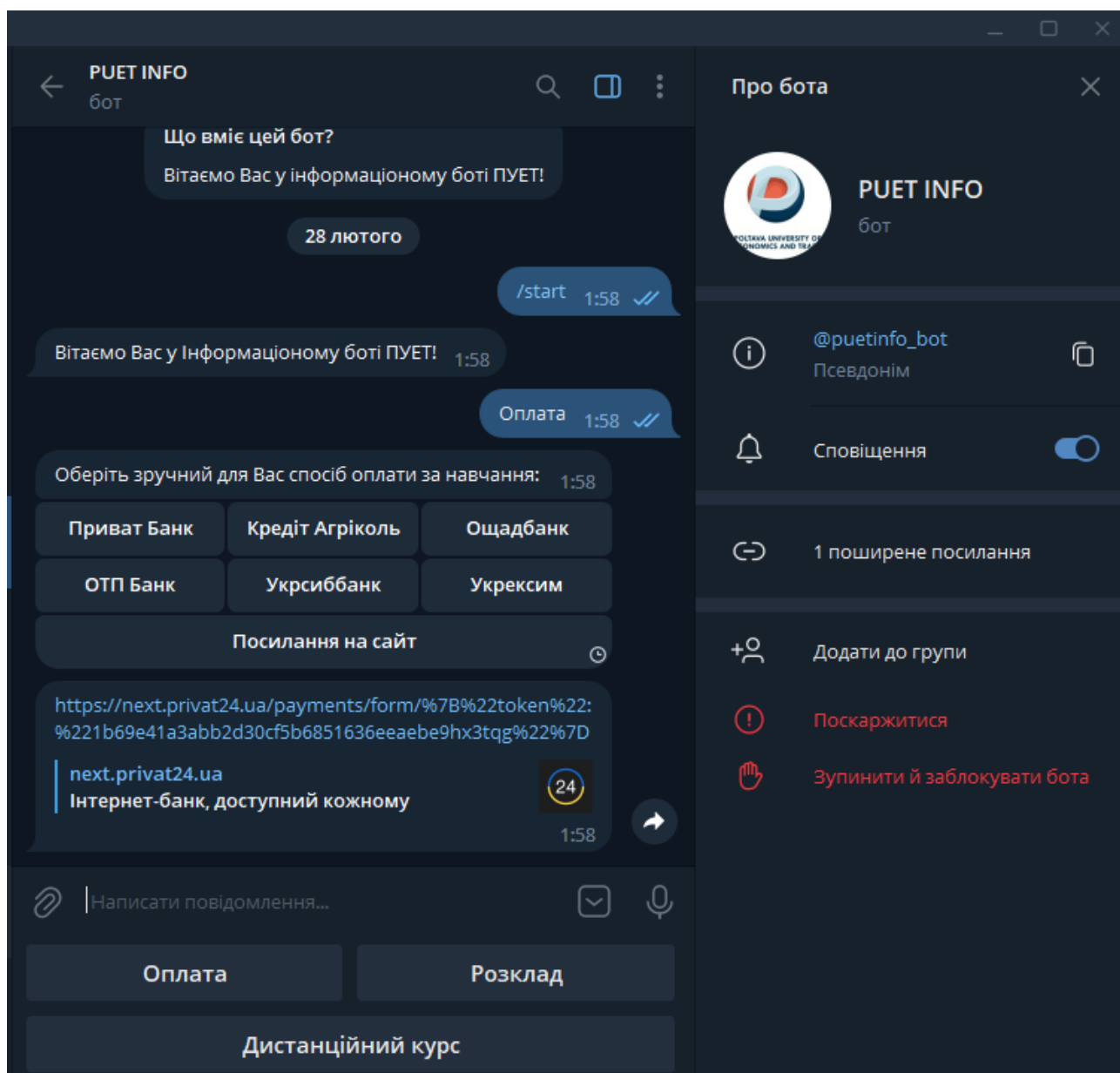


Рисунок 3.2.5 – Посилання на банк

Розклад – це кнопка, що викликає повідомлення із посиланням на розклад. (Рисунок 4.2.6)

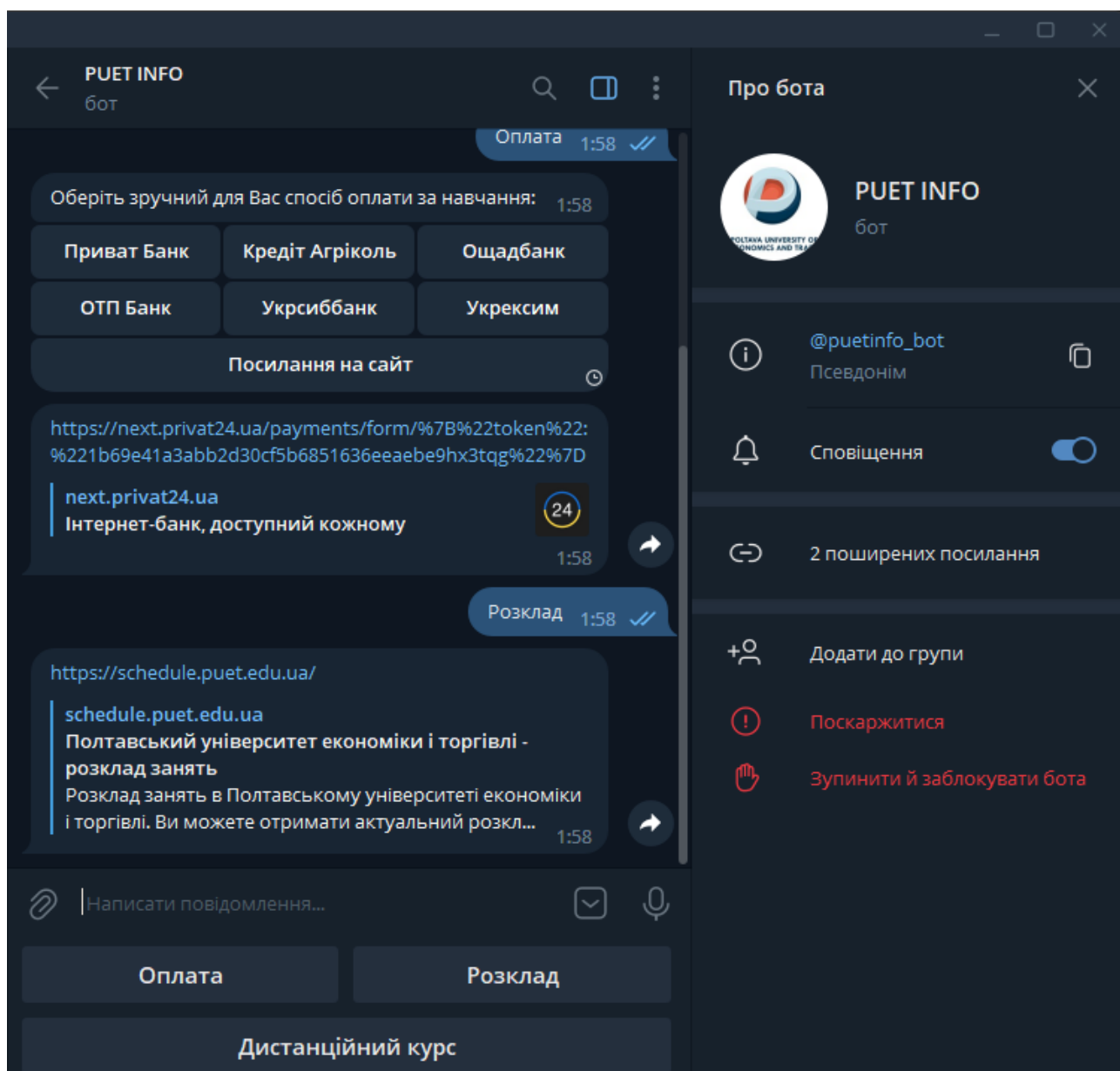


Рисунок 4.2.6 – Розклад

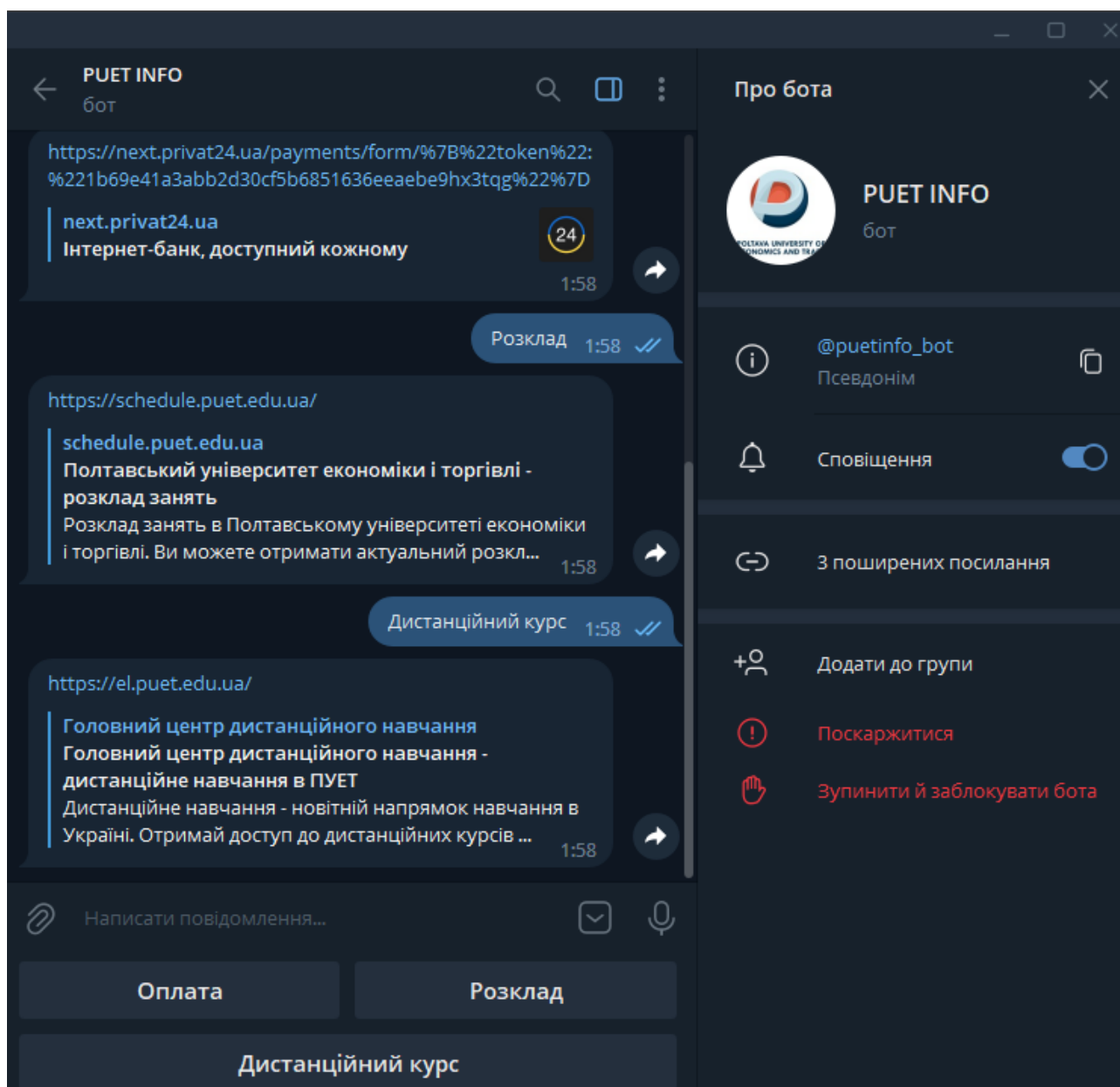


Рисунок 4.2.7 – Дистанційний курс

Chat GPT – це кнопка, що дозволяє задати запитання штучному інтелекту. Натиснувши кнопку, користувач отримує повідомлення, у якому пропонується задати запитання. Після того, як користувач напише текст запитання, бот почне обмірковувати відповідь. Чат завершується автоматично. Після наданої відповіді, під повідомленням з'являється кнопка, яка пропонує продовжити діалог зі штучним інтелектом. (Рисунок 4.2.8)

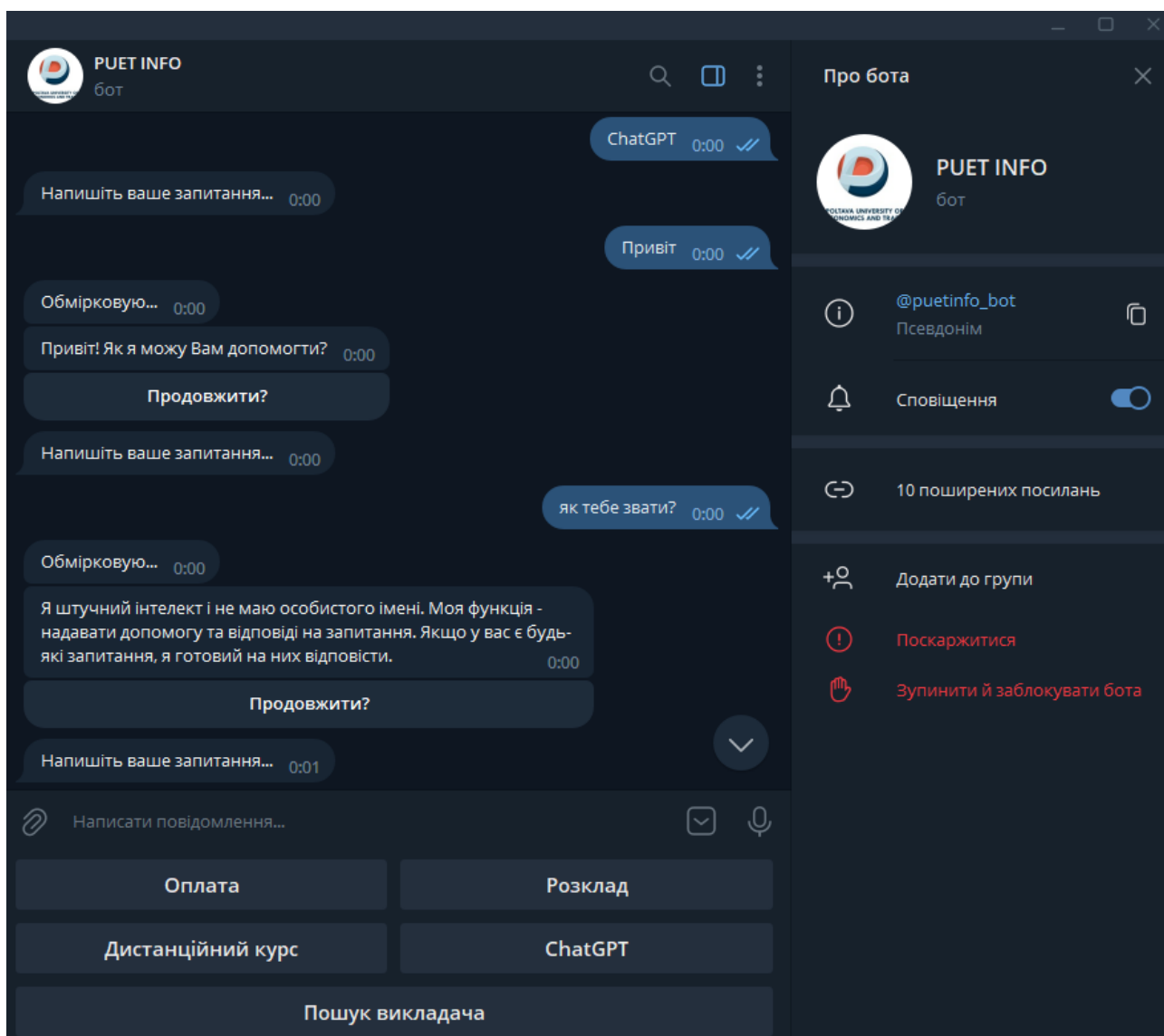


Рисунок 4.2.8 – Chat GPT

Пошук викладача – це кнопка, що дозволяє знайти електронну пошту викладача. Після натискання на кнопку, користувачу пропонується написати прізвище викладача. Після чого, користувач отримає прізвище, ім'я та по батькові, електронну пошту викладача. (Рисунок 3.2.9)

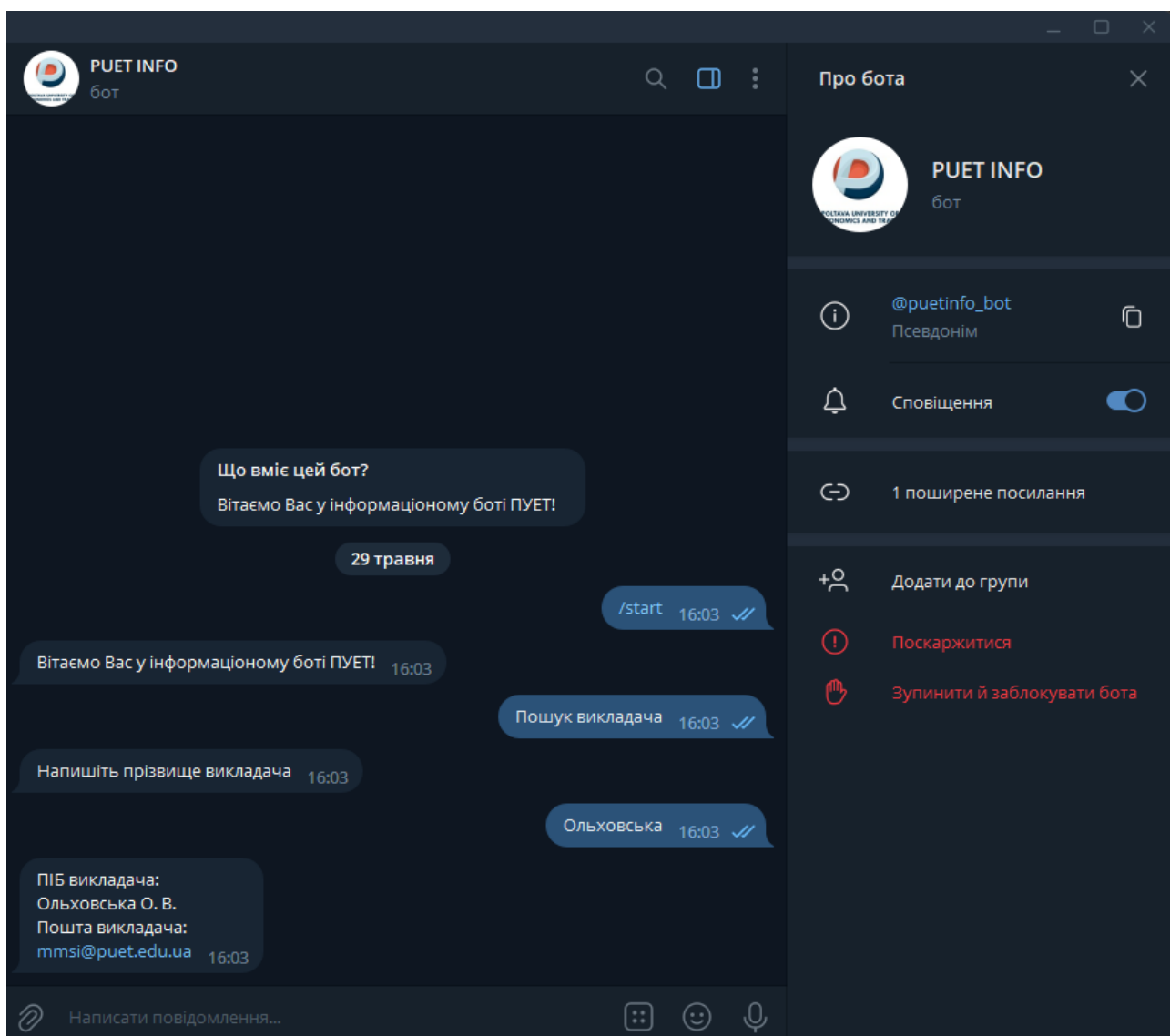


Рисунок 4.2.9 – Пошук викладача

4.3. Програмна реалізація чат-бота

Розробка чат-бота розпочалася зі створення бота в Telegram за допомогою іншого чат-бота BotFather. (Рисунок 4.3.1)

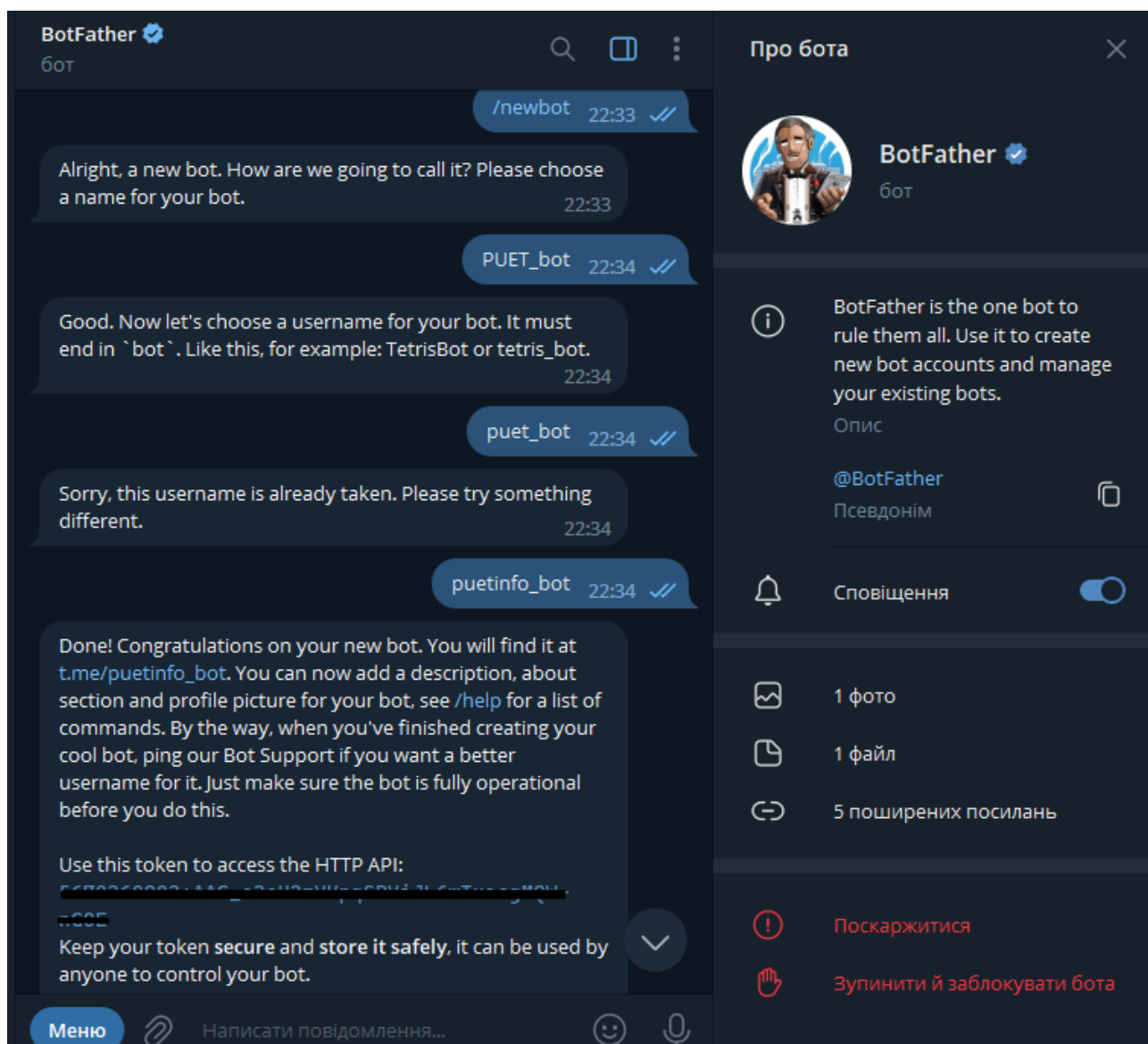


Рисунок 4.3.1 – Створення бота

Наступним кроком є додавання `pyTelegramBotAPI` до проекту. Після цього, першим кроком є створення бота в самому проекті та додавання токена бота, який був отриманий в результаті створення бота у BotFater. (Рисунок 4.3.2)

```
# KEYS

bot = telebot.TeleBot( keys.bot_key, parse_mode=None)
openai.api_key = keys.api_key
```

Рисунок 4.3.2 – Створення бота в проекті

Першим, що бачить користувач – це стартове меню. За допомогою функції “bot.message_handler” бот буде реагувати на команду “/start” та виводити набір кнопок. В нашому випадку це кнопки: “Оплата”, “Розклад”, “Дистанційний курс”, “ChatGPT”, “Пошук викладача”. Ці кнопки створюються за допомогою функції “types.ReplyKeyboardMarkup”. (Рисунок 4.3.3)

```
@bot.message_handler(commands=['start'])
# СТАРТОВИЙ ВИБІР КНОПОК
def startButton(message):
    markup_StartButton = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard= True)
    markup_StartButton.add(KeyboardButton('Оплата'),
                           KeyboardButton('Розклад'),
                           KeyboardButton('Дистанційний курс'),
                           KeyboardButton('ChatGPT'),
                           KeyboardButton('Пошук викладача'))
    bot.send_message(message.chat.id, 'Вітаємо Вас у інформаційному боті ПУЕТ!', reply_markup=markup_StartButton)
```

Рисунок 4.3.3 – Стартове меню

Наступним кроком в розробці бота було створення обробки та відповідей на стартові кнопки. Була створена функція “bot_message”, в якій бот отримував повідомлення, оброблював його зміст та надавав відповідь. За допомогою конструкції “if...else” бот розуміє яка саме кнопка, з існуючих, була натиснута та яку відповідь йому потрібно надати. Також до відповідей на деякі кнопки, а саме “Оплата”, були розроблені додаткові кнопки для можливості вибору банку для реквізитів. Це було розроблено за допомогою функції “types.InlineKeyboardMarkup”. (Рисунок 4.3.4)

```
@bot.message_handler(content_types=['text'])
def bot_message(message):
    if message.text == 'Онлайн':
        markup_PaymentMethods = types.InlineKeyboardMarkup(row_width=3)
        item_privatBank = types.InlineKeyboardButton(text='Приват Банк', callback_data='privatBank')
        item_ukrExim = types.InlineKeyboardButton(text='Укрексим', callback_data='ukrExim')
        item_creditAgrecol = types.InlineKeyboardButton(text='Кредіт Агріколь', callback_data='creditAgrecol')
        item_oshadBank = types.InlineKeyboardButton(text='Ошадбанк', callback_data='oshadBank')
        item_ukrSib = types.InlineKeyboardButton(text='Укрсиббанк', callback_data='ukrSib')
        item_otpBank = types.InlineKeyboardButton(text='ОТП Банк', callback_data='otpBank')
        item_linkToBank = types.InlineKeyboardButton(text='Посилання на сайт', callback_data='linkToBank')
        markup_PaymentMethods.add(item_privatBank, item_creditAgrecol, item_oshadBank, item_otpBank, item_ukrSib, item_linkToBank)
        bot.send_message(message.chat.id, 'Оберіть зручний для Вас спосіб оплати за навчання:', reply_markup=markup_PaymentMethods)
    elif message.text == 'Дистанційний курс':
        bot.send_message(message.chat.id, 'https://el.puet.edu.ua/')
    elif message.text == 'Розклад':
        bot.send_message(message.chat.id, 'https://schedule.puet.edu.ua/')
    elif message.text == 'ChatGPT':
        msg = bot.send_message(message.chat.id, "Напишіть ваше запитання...")
        bot.register_next_step_handler(msg, chat)
    elif message.text == 'Пошук викладача':
        msg = bot.send_message(message.chat.id, "Напишіть прізвище викладача")
        bot.register_next_step_handler(msg, searchingTeacher)
```

Рисунок 4.3.4 – Обробка та відповіді

Наступним кроком є додавання штучного інтелекту ChatGPT. Для початку потрібно зареєструватися на сайті openai.com, отримати API KEY, який далі ми використовуємо для додавання штучного інтелекту до бота. За допомогою команди «`openai.api_key`» ми додаємо API штучного інтелекту. Створюємо функцію «`chat`» у якій приймаємо повідомлення, на яке надалі штучний інтелект буде надавати відповідь. Функція «`ChatCompletion.create`» відповідає за оброблення повідомлення із запитанням від користувача. Далі штучний інтелект обдумує відповідь на запитання та відправляє повідомлення із відповіддю. (Рисунок 4.3.5)

```
#CHAT GPT
def chat(message : telebot.types.Message):
    markup_chatContinue = types.InlineKeyboardMarkup(row_width=3)
    item_chatContinue = types.InlineKeyboardButton(text='Продовжити?', callback_data='chatContinue')
    markup_chatContinue.add(item_chatContinue)
    bot.send_message(message.chat.id, "Обмірковую...")
    response = openai.ChatCompletion.create(
        model = 'gpt-3.5-turbo',
        messages = [
            {'role' : 'user', 'content' : message.text}
        ]
    )
    bot.send_message(message.chat.id, response['choices'][0]['message']['content'], reply_markup=markup_chatContinue)
```

Рисунок 4.3.5 – ChatGPT

Функція «`searchingTeacher`» створена для пошуку викладача. Для початку потрібно додати базу даних, що видно на рисунку 4.3.6.

```
# DB CONNECT
connect = sqlite3.connect('teachers.db', check_same_thread=False)
cursor = connect.cursor()

connect.commit()
```

Рисунок 4.3.6 – Додавання бази даних

Далі функція отримує повідомлення із прізвищем викладача. За допомогою запити «`cursor.execute`» функція шукає в базі даних потрібного викладача. Наступним кроком є відправлення знайденої інформації користувачу за допомогою функції «`send_message`». (Рисунок 4.3.7)

4.4. UML діаграма прецедентів

Зазвичай чат-боти – складні і динамічні структури. На їх розробку виділяють короткий час, щоб додаток як можна швидше запрацював і давав ефект. Часто розробники відразу ж сідають за написання коду, навіть не обдумавши, що вони збираються створювати, і як вони це збираються зробити. Однак за допомогою простого моделювання і суворого підходу до програмування можна зробити процес розробки гладким і гарантувати, що створену систему буде просто підтримувати в подальшому.

UML (Unified Modeling Language) – уніфікована мова моделювання – це мова, за допомогою якої описуються системи. Отже ця мова може чудово допомогти описати і відобразити майбутню систему.

Мова UML може бути складною в засвоєнні, але деякими частинами UML користуватися дуже легко, а це допоможе створювати кращі системи в менші терміни. [16]

Недоліки та переваги UML проектування:

Недоліки:

- трата часу;
- необхідність знання різних діаграм і їх нотацій.

Переваги:

- можливість подивитися на завдання з різних точок зору;
- іншим програмістам легше зрозуміти суть завдання і спосіб її реалізації;
- діаграми порівняно прості для читання після досить швидкого ознайомлення з їх синтаксисом.

При розробці будь-якої інформаційної системи всі необхідні елементи й структури даних повинні бути об'єднані у відповідну модель даних. Така модель даних являє собою інформаційну модель певної предметної області й служить для рішення відповідних завдань.

Наявність моделі даних дозволяє надалі адаптувати існуючу інформаційну систему до змін, що виникають у предметній області.

Найбільш інформативними й корисними при розробці баз даних є описи предметної області, виконані за допомогою спеціалізованих графічних нотацій.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання.

Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання: асоціації, включення, розширення та узагальнення. Діаграму прецедентів наведено в додатку Б.

Діаграма ж послідовностей відображає взаємодію об'єктів в динаміці.

В UML взаємодія об'єктів розуміється як обмін інформацією між ними. При цьому інформація набуває вигляду повідомлень. Крім того, що повідомлення несе якусь інформацію, воно певним чином також впливає на одержувача. Як бачимо, в цьому плані UML повністю відповідає основним принципам ООП, відповідно до яких інформаційну взаємодію між об'єктами зводиться до відправки і прийому повідомлень.

Діаграма послідовностей відноситься до діаграм взаємодії UML, що описує поведінкові аспекти системи, але розглядає взаємодію об'єктів в часі.

Іншими словами, діаграма послідовностей відображає тимчасові особливості передачі і прийому повідомлень об'єктами.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють в рамках сценарію, повідомлення, якими вони обмінюються, і повертають результати, пов'язані з повідомленнями.

Об'єкти позначаються прямокутниками з підкресленими іменами (щоб відрізнити їх від класів), повідомлення (виклики методів) – лініями зі стрілками, які повертаються результати – пунктирними лініями зі стрілками. Прямокутники на вертикальних лініях під кожним з об'єктів показують "час життя" (фокус) об'єктів. Втім, досить часто їх не зображують на діаграмі, все це залежить від індивідуального стилю проектування.

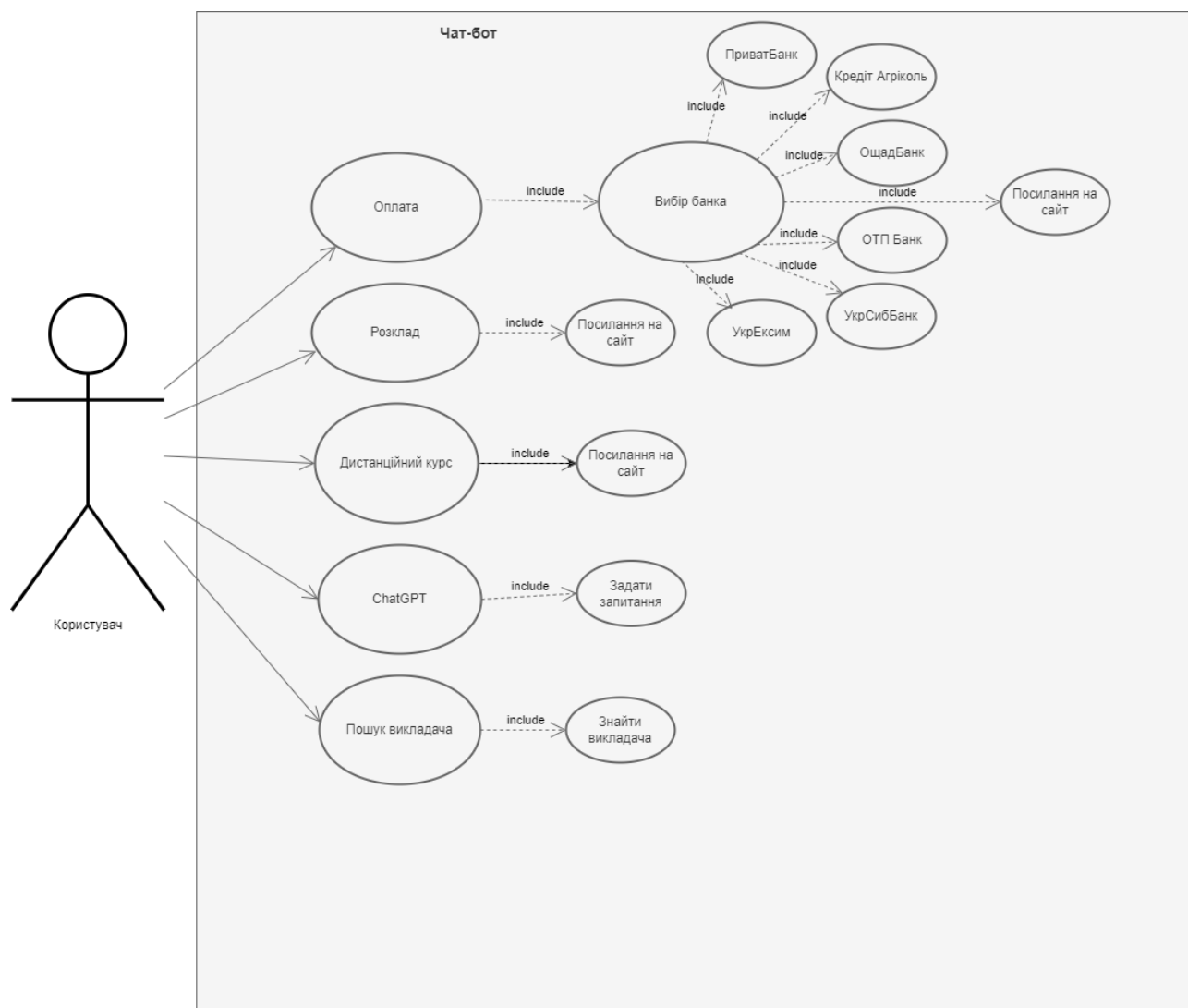


Рисунок 4.4.1 – UML діаграма прецедентів

4.5. Тестування

Тестування графічного інтерфейсу мобільного-додатку проводилось на емуляторі Pixel 3a з операційною системою Android. Під час тестування було перевірено коректність розміщення усіх елементів інтерфейсу та їх відповідність заданим стилям й визначеній функціональності.

У процесі планування були заплановані такі методи тестування:

- функціональне тестування;
- юзабіліті тестування.

Функціональне тестування проводилося вручну, методом «неформального» тестування (ad-hoc testing) з позиції кінцевого користувача.

Ціллю функціонального тестування є виявлення функціональних помилок, невідповідність з технічним завданням.

Функціональне тестування – передбачає аналіз функціональних характеристик додатку та перевірку на невідповідності між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і бізнес-вимоги. Функціональне тестування фактично імітує фактичне використання системи.

Функціональні тести можуть проводитися на усіх рівнях тестування (компонентному, інтеграційному, системному, приймальному).

Ціллю юзабіліті тестування є перевірка на легкість та зручність у використанні.

У даній роботі було застосоване ручне тестування програмного продукту. У додатку Г в таблиці Г.1 було обрано та представлено набір Test Cases – описує сукупність кроків, конкретних умов та параметрів, необхідних для перевірки реалізації функції, що тестується чи її частини.

ВИСНОВКИ

У даній роботі було проєктовано та реалізовано телеграм чат-бота, який може взаємодіяти з користувачами за допомогою текстових повідомлень та надавати корисну інформацію. Було розглянуто питання вибору технологій, архітектури та реалізації програмного забезпечення.

Результатом проведених досліджень є підтвердження того, що телеграм чат-боти можуть стати ефективним інструментом для взаємодії з користувачами. Чат-боти дозволяють користувачам швидко та ефективно отримувати інформацію.

Таким чином, розробка телеграм чат-бота є актуальною темою для дослідження, оскільки вона може допомогти користувачам, а саме студентам, швидко та ефективно отримувати інформацію. Отримані знання та досвід можуть бути використані в майбутньому при розробці більш складних проєктів в сфері розробки інформаційних технологій.

У проєкті з фаху описано такі пункти:

- Постановка задачі.
- Типи Telegram-ботів.
- Системи, середовища програмування, середовища для розробки програмного забезпечення.
- Алгоритм роботи чат-бота.
- Проектування чат-бота.
- Робота чат-боту.

СПИСОК ЛІТЕРАТИ

1. Ольховська О. В. Методичні рекомендації до виконання кваліфікаційної роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» ступеня бакалавра / О. В. Ольховська, О. О. Черненко – Полтава: ПУЕТ, 2022. – 71с.
2. Опис середовища розробки Visual Studio [Електронний ресурс] – Режим доступу: https://studbooks.net/2258619/informatika/opisanie_sredy_razrabotki_microsoft_visual_studio – Назва з екрану.
3. 10 Best Ide in the World [Електронний ресурс] – Режим доступу: <https://medium.com/issuehunt/10-best-idein-the-world-c4875d1300c5%20%E2%80%93%20%D0%94%D0%B0%D1%82%D0%B0%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D1%83%2001.06.2019> – Назва з екрану.
4. ISO/IEC 23270:2006 – Information technology – Programming languages – C# [Електронний ресурс] – Режим доступу: <https://www.iso.org/standard/42926.html> – Назва з екрану.
5. Значення блок-схеми [Електронний ресурс] – Режим доступу: <https://uk.warbletoncouncil.org/diagrama-de-flujo-388> – Назва з екрану.
6. Цикозін О. В. Пояснювальна записка до бакалаврської роботи на тему «Програмна реалізація та алгоритмізація тренажера дистанційного навчального курсу «Обчислювальні методи» з теми «Розв'язування диферівнянь методом Ейлера» / О. В. Цикозін. – Полтава: Кафедра ММСІ ПУЕТ, 2020 – 64 с.
– Режим доступу: <http://dSPACE.puet.edu.ua/bitstream/123456789/9029/1/%d0%a6%d0%b8%d0%ba%d0%be%d0%b7%d1%96%d0%bd.pdf>
7. Кулинич М. К. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення тренажеру з теми «Злиття

впорядкованих послідовностей» дистанційного навчального курсу «Алгоритми та структури даних» / М. К. Кулинич. – Полтава: Кафедра ММСІ ПУЕТ, 2020 – 39 с. – Режим доступу:

<http://dspace.puet.edu.ua/bitstream/123456789/9006/1/%d0%9a%d1%83%d0%bb%d0%b8%d0%bd%d0%b8%d1%87%20%d0%b0%d0%bd%d1%82%d0%b8%d0%bf%d0%bb%d0%b0%d0%b3%d1%96%d0%b0%d1%82.pdf>

8. Скромінський М. В. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення для тренажеру з теми «Контекстові граматики» дистанційного навчального курсу «Теорія програмування» / М. В. Скромінський. – Полтава: Кафедра ММСІ ПУЕТ, 2020 – 39 с. – Режим доступу:

http://dspace.puet.edu.ua/bitstream/123456789/9025/1/Skrominskyi_Mykhailo_Vitaliiovych-KN21i-1%20%281%29.pdf

9. Хрупа В. О. Розробка програмного забезпечення з теми «Турнірне сортування» дистанційного навчального курсу «Алгоритми та структури даних» / В. О. Хрупа, Ол-ра О. Ємець. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – 3 с. – Режим доступу:

10. Python [Електронний ресурс]. – Режим доступу: URL: <https://aws.amazon.com/ru/what-is/python> – Назва з екрану.

11. Visual Studio Code [Електронний ресурс]. – Режим доступу: URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code – Назва з екрану.

12. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.– Суми: Вид-во СумДУ, 2009. – 181 с.

13. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://ttp.unicyb.kiev.ua/doc/TOP.pdf>.

14. Гусак Т. М., Малінко О. Г. Підвищення самостійності студентів під час вивчення іноземних мов // Педагогіка і психологія, Т. М. Гусак. – К., 2000. – 56 с.

15. Самостійна робота студентів. Фактори успіху [Електронний ресурс] – Режим доступу: https://dut.edu.ua/ua/news-1-525-7802-samostiyna-robota-studentiv-faktori-uspihu_kafedra-inozemnih-mov – Назва з екрану.

16. UML [Електронний ресурс]. – Режим доступу: URL: <https://habr.com/ua/post/458680/> (дата звернення 26.05.2021). – Назва з екрану.

ДОДАТОК А. КОД ПРОГРАМИ

```

import telebot
from telebot import types
from telebot.types import KeyboardButton
import openai
import sqlite3
import keys.py

# DB CONNECT
connect = sqlite3.connect('teachers.db', check_same_thread=False)
cursor = connect.cursor()

connect.commit()

# KEYS

bot = telebot.TeleBot( keys.bot_key, parse_mode=None)
openai.api_key = keys.api_key

@bot.message_handler(commands=['start'])
# СТАРТОВИЙ ВИБІР КНОПОК
def startButton(message):
    markup_StartButton = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard= True)
    markup_StartButton.add(KeyboardButton('Оплата'),
                           KeyboardButton('Розклад'),
                           KeyboardButton('Дистанційний курс'),
                           KeyboardButton('ChatGPT'),
                           KeyboardButton('Пошук викладача'))

    bot.send_message(message.chat.id, 'Вітаємо Вас у інформаційному боті ПУЕТ!',
                      reply_markup=markup_StartButton)
#NAVBAR
@bot.message_handler(content_types=['text'])
def bot_message(message):
    if message.text == 'Оплата':
        markup_PaymentMethods = types.InlineKeyboardMarkup(row_width=3)
        item_privatBank = types.InlineKeyboardButton(text='Приват Банк',
        callback_data='privatBank')
        item_ukrExim = types.InlineKeyboardButton(text='Укрексим', callback_data='ukrExim')
        item_creditAgrecol = types.InlineKeyboardButton(text='Кредіт Агріколь',
        callback_data='creditAgrecol')
        item_oshadBank = types.InlineKeyboardButton(text='Ощадбанк', callback_data='oshadBank')
        item_ukrSib = types.InlineKeyboardButton(text='Укрсиббанк', callback_data='ukrSib')
        item_otpBank = types.InlineKeyboardButton(text='ОТП Банк', callback_data='otpBank')
        item_linkToBank = types.InlineKeyboardButton(text='Посилання на сайт',
        callback_data='linkToBank')

```

```

markup_PaymentMethods.add(item_privatBank, item_creditAgrecol, item_oshadBank,
item_otpBank, item_ukrSib, item_ukrExim, item_linkToBank)
bot.send_message(message.chat.id, 'Оберіть зручний для Вас спосіб оплати за навчання:',
reply_markup=markup_PaymentMethods)
elif message.text == 'Дистанційний курс':
bot.send_message(message.chat.id, 'https://el.puet.edu.ua/')
elif message.text == 'Позклад':
bot.send_message(message.chat.id, 'https://schedule.puet.edu.ua/')
elif message.text == 'ChatGPT':
msg = bot.send_message(message.chat.id, "Напишіть ваше запитання...")
bot.register_next_step_handler(msg, chat)
elif message.text == 'Пошук викладача':
msg = bot.send_message(message.chat.id, "Напишіть прізвище викладача")
bot.register_next_step_handler(msg, searchingTeacher)
#CHAT GPT
def chat(message : telebot.types.Message):
markup_chatContinue = types.InlineKeyboardMarkup(row_width=3)
item_chatContinue = types.InlineKeyboardButton(text='Продовжити?',
callback_data='chatContinue')
markup_chatContinue.add(item_chatContinue)
bot.send_message(message.chat.id, "Обмірковуюю...")
response = openai.ChatCompletion.create(
model = 'gpt-3.5-turbo',
messages = [
{'role' : 'user' , 'content' : message.text}
]
)
bot.send_message(message.chat.id, response['choices'][0]['message']['content'],
reply_markup=markup_chatContinue)
#ПОШУК ВИКЛАДАЧА
def searchingTeacher(message):
cursor.execute(f'SELECT * FROM teachers WHERE lastname = "{message.text}")
msg = cursor.fetchall()
for i in msg:
bot.send_message(message.chat.id,"ПІБ викладача: \n" + i[0] + " " + i[1] + "\nПошта
викладача: \n" + i[2])
#ВІДПОВІДІ НА КНОПКИ БОТА
@bot.callback_query_handler(func=lambda call: True)
def callback(call):
if call.message:
if call.data == 'linkToBank':
bot.send_message(call.message.chat.id,
'https://next.privat24.ua/payments/form/%7B%22token%22:%221b69e41a3abb2d30cf5b6851636ee
aebe9hx3tqg%22%7D')
elif call.data == 'privatBank':
bot.send_message(call.message.chat.id, "Приват банк:")
bot.send_message(call.message.chat.id, "IBAN:")
bot.send_message(call.message.chat.id, "UA433052990000026000041207649")
bot.send_message(call.message.chat.id, "ЄДРПОУ:")
bot.send_message(call.message.chat.id, "01597997")
elif call.data == "ukrExim":

```

```

    bot.send_message(call.message.chat.id, "УКрексим:")
    bot.send_message(call.message.chat.id, "IBAN:")
    bot.send_message(call.message.chat.id, "UA663223130000026008000019421")
    bot.send_message(call.message.chat.id, "ЄДРПОУ:")
    bot.send_message(call.message.chat.id, "01597997")
elif call.data == 'creditAgrecol':
    bot.send_message(call.message.chat.id, "Кредіт Агріколь:")
    bot.send_message(call.message.chat.id, "IBAN:")
    bot.send_message(call.message.chat.id, "UA653006140000026003500152765")
    bot.send_message(call.message.chat.id, "ЄДРПОУ:")
    bot.send_message(call.message.chat.id, "01597997")
elif call.data == 'oshadBank':
    bot.send_message(call.message.chat.id, "Ощадбанк:")
    bot.send_message(call.message.chat.id, "IBAN:")
    bot.send_message(call.message.chat.id, "UA873314670000026004300878326")
    bot.send_message(call.message.chat.id, "ЄДРПОУ:")
    bot.send_message(call.message.chat.id, "01597997")
elif call.data == 'ukrSib':
    bot.send_message(call.message.chat.id, "Укрсиббанк:")
    bot.send_message(call.message.chat.id, "IBAN:")
    bot.send_message(call.message.chat.id, "UA543510050000026005208016500")
    bot.send_message(call.message.chat.id, "ЄДРПОУ:")
    bot.send_message(call.message.chat.id, "01597997")
elif call.data == 'otpBank':
    bot.send_message(call.message.chat.id, "ОТП Банк:")
    bot.send_message(call.message.chat.id, "IBAN:")
    bot.send_message(call.message.chat.id, "UA363005280000026001455058062")
    bot.send_message(call.message.chat.id, "ЄДРПОУ:")
    bot.send_message(call.message.chat.id, "01597997")
elif call.data == 'chatContinue':
    msgStop = bot.send_message(call.message.chat.id, "Напишіть ваше запитання...")
    bot.register_next_step_handler(msgStop, chat)

bot.polling(none_stop=True, interval=0)

```