

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

**Кафедра прикладних інформаційних
систем**

**ВИПУСКНА КВАЛІФІКАЦІЙНА
РОБОТА БАКАЛАВРА**

НА ТЕМУ

Чат-бот для навчальних консультацій

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Прикладне програмування»**

Освітній рівень: Бакалавр

Виконав: студент 4 курсу, групи ПП-42

Жовтецький Роман Дмитрович

Керівник к.т.н., доцент Краснощок В. М.

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри прикладних інформаційних систем
(Протокол №14 від 23.05.2023)

завідувачка кафедри  проф. Плескач В.Л.

Унікальність тексту – 94 %

Київ – 2023

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№з/п	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	14.10.2022	Виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	24.10.2022	Заява
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	31.10.2022	Виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.11.2022	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.11.2022	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2023	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2023	Виконано
9.	Подання роботи у першому варіанті	28.04.2023	Виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2023	Виконано
	Подання кваліфікаційної роботи бакалавра на попередній захист	23.05.2023	Виконано
	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	26.05.2023	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврську роботу)	12.06.2023	Виконано
14.	Захист кваліфікаційної роботи бакалавра	28.06.2023	Виконано

Здобувач вищої освіти

Керівник



ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Складові частини дипломної роботи	Обсяг, 67
Титульний аркуш	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	2
Анотація (іноземною мовою-англійською)	2
Зміст	1
Перелік скорочень, умовних позначень, термінів	1
Вступ	4
Розділ 1	8
Розділ 2	16
Розділ 3	20
Висновки	2
Список використаних джерел	3
Додатки	5

				ДП ХХХХ 00.000.00		
	ПІБ	Підпис	Дата			
Розробн.	Жовтецький Р.М			Відомість дипломної роботи	Лист	67 листів
Керівн.	Краснощок В.М.					
Н/контр.	Кравченко К.В.					
Зав.каф.	Плескач В.Л.					

АНОТАЦІЯ

Дипломна робота: 66 с., 21 рис., 8 таблиць, 20 джерел, 1 дод.

Дана дипломна робота присвячена розробці чат-боту для навчальних консультацій з використанням програмних засобів.

Для досягнення цього було проведено дослідження різних технологій та програмних засобів, що використовуються для розробки чат-ботів. Було проведено порівняння різних компонентів, методів та підходів до реалізації чат-боту з метою вибору оптимальних технологій.

Розроблене програмне забезпечення являє собою телеграм бота, який надає можливість отримувати інформацію від бота в будь який момент.

У даній дипломній роботі було розроблено архітектуру, методи отримання інформації різного типу та зворотного зв'язку.

Для розробки оптимізованої системи було проведено дослідження актуальних на даний час технологій, що дозволяють створювати телеграм ботів. Вибір технології для реалізації дипломної роботи був зроблений по таким критеріям, як наявні можливості, масштабованість, програмне забезпечення та вирішення проблем, що можуть виникнути під час розробки чат-боту.

У дипломній роботі надано детальний опис розробки чат-боту, включаючи проектування архітектури, вибір імплементаційних технологій, розробку функцій, компонентів, модулів, контролерів, сервісів та стратегій. Також наведено приклади впровадження та програмного супроводження чат-боту.

Результати досліджень та розробки представлені у роботі допоможуть покращити доступність та якість навчальних консультацій, сприятимуть покращенню ефективності навчального процесу та забезпеченню зручного та інтерактивного спілкування з користувачами.

Мета дипломної роботи- підвищення якості надання освітніх послуг за допомогою створення чат-боту.

Головними завданнями дослідження є вивчення потреб та очікувань користувачів, розробка інтерактивного інтерфейсу чат-бота, підбір та навчання

алгоритмів обробки запитань та надання консультацій, а також оцінка ефективності та задоволеності користувачів від використання чат-бота.

Об'єктом дослідження є процес консультування з питань надання освітніх послуг.

Предметом дослідження є засоби створення чат-боту щодо процесу навчання.

Методи дослідження. В роботі використовувалися наступні методи дослідження:

- порівняння;
- експеримент;
- узагальнення;
- формалізація;
- аналіз.

Ключові слова: bot, чат-бот, Python.

ABSTRACT

Thesis: 66 pages, 21 figures, 8 tables, 20 references, 1 appendix.

This thesis is devoted to the development of a chatbot for educational consultations using software tools.

To achieve this goal, a study of various technologies and software tools used to develop chatbots was conducted. Different components, methods, and approaches to chatbot implementation were compared in order to select the best technologies.

The developed software is a telegram bot that allows you to receive information from the bot at any time.

In this thesis, the architecture, methods of obtaining various types of information and feedback were developed.

In order to develop an optimized system, a study of currently relevant technologies was conducted. that allow you to create Telegram bots. The choice of technology for the implementation of the thesis was made according to such criteria as available capabilities, scalability, software and solving problems that may arise during the development of a software system.

The thesis provides a detailed description of chatbot development, including architecture design, selection of implementation technologies, development of functions, components, modules, controllers, services, and strategies. Examples of chatbot implementation and software support are also provided.

The research and development results presented in this paper will help to improve the availability and quality of educational consultations, contribute to the efficiency of the educational process, and ensure convenient and interactive communication with users.

The purpose of the thesis is to improve the quality of educational services by creating a chatbot.

The main tasks of the research are studying the needs and expectations of users, developing an interactive chatbot interface, selecting and training algorithms

for processing questions and providing advice, as well as evaluating the effectiveness and satisfaction of users from using a chatbot.

The object of research of the bachelor's qualification work is effective counseling of students based on the created chatbot.

The subject of the research is the process of consulting on the provision of educational services.

Research methods. The following research methods were used in the work:

- comparison;
- experiment;
- generalization;
- formalization;
- analysis.

Keywords: bot, chatbot, Python.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1	14
СУЧАСНІ ПІДХОДИ ДО РОЗРОБЛЕННЯ І ВПРОВАДЖЕННЯ ЧАТ-БОТІВ	14
1.1 Історія появи чат-ботів	14
1.2 Концепція чат-бота	15
1.3 Огляд існуючих рішень	19
РОЗДІЛ 2	22
АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТІВ	22
2.1. Класифікація чат-ботів	22
2.2 Вибір засобів розробки	24
2.3 Архітектура програмного забезпечення	30
Висновки до розділу 2	37
РОЗДІЛ 3	38
ОПИС РОБОТИ СИСТЕМИ	38
3.1 Постановка задач	38
3.2 Розробка програмного забезпечення	38
3.3 Інструкція користувача	48
Висновки до розділу 3	56
ВИСНОВОК	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту

HTTPS (HyperText Transfer Protocol Secure) – розширення протоколу HTTP

URL (Uniform Resource Locator) – єдиний вказівник на ресурс
API (Application Programming Interface) – програмний інтерфейс створення додатків

NLP (Natural Language Toolkit) – це набір бібліотек і програм для символної та статистичної обробки природної мови для англійської мови, написаних мовою програмування Python

HTML5 (HyperText Markup Language) – наступна версія мови HTML

JSON (JavaScript Object Notation) – текстовий формат обміну даними, заснований на JavaScript

IDE (integrated development environment) – комплексне програмне рішення для розробки програмного забезпечення.

ВСТУП

В сучасному світі, де технології швидко розвиваються та змінюють наше повсякденне життя, навчання не залишається осторонь цього тренду. Студенти та учні постійно шукають нові шляхи отримання знань і покращення своїх навичок. Одним із найперспективніших інноваційних інструментів, який став все більш популярним, є чат-боти.

Чат-боти - це програмні рішення, здатні автоматично комунікувати з людьми через текстовий інтерфейс. Вони можуть бути використані в різних сферах, включаючи освіту. І одним з найцікавіших та корисних застосувань чат-ботів є надання навчальних консультацій.

Актуальність. Актуальність даного дослідження полягає в зростаючій потребі у покращенні доступності та ефективності навчання завдяки розробці та впровадженню чат-бота для навчальних консультацій. Чат-боти стають все більш популярними, і їхнє використання в освітній сфері має значний потенціал для забезпечення швидкої та зручної комунікації зі студентами.

Завдяки чат-ботам, студенти матимуть можливість отримувати навчальну підтримку та консультації в будь-який зручний для них час, що сприятиме підвищенню доступності навчання. Вони зможуть отримувати індивідуалізовану підтримку, задавати питання, отримувати пояснення та матеріали, що допоможе покращити якість їхнього навчання.

Окрім того, дослідження актуально і з точки зору аналізу та оцінки ефективності використання чат-бота. Це дозволить з'ясувати, наскільки задоволені користувачі від такого способу навчання та консультування і виявити можливі сильні та слабкі сторони системи для подальшого покращення її функціональності.

У сучасному світі, де швидкість інформаційного потоку безперервно зростає, а людям потрібно оперативно отримувати релевантну інформацію,

створення чат-бота для отримання інформації про університет, іспити та інші навчальні аспекти є надзвичайно актуальним.

Перш за все, чат-боти забезпечують миттєвий доступ до інформації. Будь-який студент або абітурієнт може одразу отримати відповіді на свої запитання без необхідності чекати на відповідь від конкретної особи. Це особливо важливо в період підготовки до вступу чи під час навчального семестру, коли виникає багато питань щодо програм навчання, дат іспитів, термінів подачі документів та інших нюансів.

Крім того, чат-боти забезпечують індивідуалізацію інформації. Вони можуть надати користувачам персоналізовані відповіді, враховуючи їх потреби та контекст. Наприклад, студент може отримати інформацію про розклад занять та іспитів, або отримати поради щодо підготовки до конкретного іспиту. Це дозволяє ефективніше орієнтуватися у навчальному процесі та зменшує ймовірність помилок чи неправильного розуміння інформації.

Більше того, чат-боти можуть забезпечувати постійну наявність і підтримку. Вони доступні 24/7, що дає можливість користувачам отримати відповіді на свої запитання в будь-який зручний для них момент. Важливою перевагою є те, що чат-боти можуть обробляти багато запитів одночасно, не потребуючи людської інтеракції, що робить їх швидкими та ефективними у наданні інформації.

Отже, створення чат-бота для отримання інформації про університет, іспити та інші навчальні аспекти має велику актуальність в сучасному світі. Вони забезпечують швидкий доступ до інформації, індивідуалізацію відповідей та постійну наявність, сприяючи ефективному та зручному навчанню. Це інноваційний інструмент, який допомагає студентам та абітурієнтам бути краще підготовленими та успішними у своїй освітній дорозі.

На сьогодні існує ціла низка світових компаній, які так чи інакше вбудували функціонал чат-ботів у свої продукти. Для прикладу, Facebook (Facebook Messenger), Microsoft (Skype), Viber Media, Inc (Viber), Telegram FZ-LLC (Telegram). У нашому випадку мова йде про створення чат бота на платформі Telegram.

Мета дослідження полягає в аналізі, розробці та впровадженні чат-бота для надання навчальних консультацій з метою покращення доступності, ефективності та якості навчання. Головними завданнями дослідження є вивчення потреб та очікувань користувачів, розробка інтерактивного інтерфейсу чат-бота, підбір та навчання алгоритмів обробки запитань та надання консультацій, а також оцінка ефективності та задоволеності користувачів від використання чат-бота.

Предмет дослідження - функціональні можливості, ефективність та вплив чат-бота на процес навчання та отримання навчальних консультацій.

Об'єктом дослідження в даному контексті є створення чат-боту для надання навчальних консультацій.

Практичне значення дипломної роботи полягає в створенні функціонального чат-боту для навчальних консультацій. Цей чат-бот дозволить студентам отримувати навчальну допомогу та консультації у зручний для них час, сприяє покращенню доступності та якості навчання.

Методи дослідження. В роботі використовувалися наступні методи дослідження:

- порівняння;
- експеримент;
- узагальнення;
- формалізація;
- аналіз.

Структура роботи:

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, розподілених на підрозділи та висновку.

РОЗДІЛ 1

СУЧАСНІ ПІДХОДИ ДО РОЗРОБЛЕННЯ І ВПРОВАДЖЕННЯ ЧАТ-БОТІВ

1.1 Історія появи чат-ботів

Поява чат-ботів в сфері комунікацій є однією з важливих межових подій в історії інформаційних технологій. Хоча корінні етапи розвитку цих інтерактивних програм відносяться до останнього десятиліття, їх поява була попереджена довгим процесом технологічного й культурного розвитку[1].

Концепція чат-ботів виникла на початку 20-го століття. У 1950-х роках, із з'явленням перших комп'ютерів, народилося багато прагнень створити програми, здатні відтворювати розумові процеси людей. Однак, технологічні обмеження часу та обчислювальних можливостей не дозволяли реалізувати ці ідеї повністю[2].

Перші кроки у розвитку чат-ботів зроблені у 1960-х роках. ALICE (Artificial Linguistic Internet Computer Entity) стала однією з перших відомих програм, яка володіла деякими рівнями розуміння та відтворення людської мови[2]. Цей експеримент був досить примітивним і базувався на шаблонних відповідях.

У 1970-х роках почали з'являтися перші програми, які використовували штучний інтелект для моделювання діалогів з людьми. Ельїза, розроблена Шейном Бобертом, була одним з успішних прикладів таких програм. Вона вміла імітувати психотерапевтичні розмови, але її функціональність обмежувалася шаблонами[2].

Розвиток комп'ютерної обробки природної мови та штучного інтелекту продовжувався протягом наступних десятиліть. У 1990-х роках, з'явилися системи, які використовували статистичні методи та машинне навчання для поліпшення розпізнавання мови та відповідей[1]. Наприклад, програма ELIZA,

розроблена Джозефом Вайзенбаумом, стала популярною через свою здатність імітувати психотерапевтичні сесії.

У 2000-х роках народилися перші комерційні чат-боти. З'явилися інтерактивні агенти, призначені для обслуговування клієнтів на веб-сайтах. Ці програми забезпечували користувачів інформацією, відповідями на запитання, а також допомагали здійснювати покупки. Перші комерційні чат-боти були базовими та обмеженими, але вони зазнавали постійного розвитку[3].

У 2010-х роках революція в галузі машинного навчання та обробки природної мови сприяла з'яві більш складних та розумних чат-ботів. Багато компаній, таких як Apple, Google та Facebook, впровадили власні чат-боти в свої продукти та платформи. Штучний інтелект став основою для створення чат-ботів, які могли здійснювати розширені діалоги, розпізнавати інтенст користувача та надавати персоналізовані відповіді[3].

В сучасному світі чат-боти знаходять широке застосування в різних галузях, включаючи клієнтське обслуговування, медицину, фінанси, освіту та інші. Вони є важливими інструментами для автоматизації комунікацій та покращення користувацького досвіду. Подальший розвиток штучного інтелекту та нові технологічні відкриття надають великий потенціал для подальшого вдосконалення чат-ботів та їхнього використання у майбутньому.

1.2 Концепція чат-бота

Чат-бот – це тип комп'ютерного програмного забезпечення, яке імітує людську мову, як правило, за допомогою текстових обмінів. Можна створити чат-ботів, які відповідатимуть на введення користувача підготовленими повідомленнями, або вони можуть використовувати обробку природної мови (NLP), щоб зрозуміти запитання користувачів і відповідати на них більш розмовно [11].

Віртуальний помічник, який спрямовує споживачів до служби підтримки під час кризи, наприклад, стихійного лиха чи війни, є однією з ідей чат-бота. Користувачі зможуть взаємодіяти з волонтерами або іншими службами у своєму районі, використовуючи чат-бот для пошуку ресурсів, таких як їжа, притулок та медична допомога. Дані геолокації можуть використовуватися чат-ботом, щоб знаходити людей і повідомляти їм, де найближчі до них ресурси [2].

Разом із контактною інформацією для надзвичайних ситуацій і порадами про те, як бути готовими до криз, чат-бот також може надавати користувачам інформацію про те, як бути в безпеці під час кризи. Крім того, чат-бот може надавати споживачам оновлену інформацію про стан кризи та будь-яку свіжу інформацію, наприклад зміни наказів про евакуацію або перекриття доріг [3].

Ще одна ідея чат-бота — віртуальний консьєрж для готелів чи інших готельних фірм [16]. Використовуючи чат-бот, відвідувачі зможуть робити запити або запитувати зрозумілою мовою, наприклад, замовляти обслуговування номерів або отримувати поради щодо найближчих заходів. Відповіді чат-бота можуть бути покращені з часом за допомогою алгоритмів машинного навчання для вивчення взаємодії з клієнтами.

Щоб запропонувати безперебійний досвід відвідувачів, чат-бот також може взаємодіяти з іншими готельними системами, наприклад системами бронювання [4].

У кризовій ситуації чи в готельному середовищі ідея чат-бота є ефективним інструментом для надання людям індивідуальної підтримки та інформації в багатьох сценаріях. Чат-боти корисні для компаній і організацій, які прагнуть запропонувати першокласну підтримку клієнтів завдяки розробкам NLP і машинного навчання. Вони стають розумнішими та здатними до складних розмов [19].

Чат бот у контексті Telegram

У контексті Telegram чат-бот — це стороння програма, яка працює на платформі Telegram. Надаючи повідомлення, інструкції та вбудовані запити, користувачі спілкуються з ботами. Існує спеціальний API, який потрібно використовувати за допомогою протоколу HTTPS, щоб контролювати бот Telegram [5].

Telegram-боти - це автоматизовані акаунти, які можуть взаємодіяти з користувачами через Telegram-платформу. Вони можуть мати різноманітний функціонал, включаючи надсилання повідомлень, виконання запитів, збереження даних, відповіді на команди користувачів та багато іншого [5].

Деякі приклади функціоналу, який може бути реалізований за допомогою Telegram-боту[5]:

1. Отримання інформації: Бот може відповідати на запити користувачів, надавати інформацію про погоду, новини, курси валют, графіки тощо.
2. Розсилка повідомлень: Бот може надсилати регулярні оновлення або сповіщення про події користувачам, які підписалися на нього.
3. Робота з базою даних: Бот може зберігати та обробляти дані, включаючи реєстрацію користувачів, збереження їхніх налаштувань або історії повідомлень.
4. Розваги: Бот може надавати інтерактивні ігри, генерувати мему, відтворювати аудіо або відео, розповідати анекдоти тощо.
5. Інтеграція з іншими сервісами: Бот може отримувати дані з зовнішніх джерел, наприклад, відстежувати стан акцій на фондовій біржі або надавати інформацію про розклад рейсів.

Розглянемо Telegram Bot API. Кожен бот потребує авторизації з боку серверів Telegram.

Telegram Bot API - це програмний інтерфейс програмування (API), який надається Telegram для взаємодії з Telegram-ботами. Цей API дозволяє розробникам створювати, налаштовувати та керувати ботами на платформі Telegram.

Основні можливості Telegram Bot API включають [6]:

1. Отримання оновлень: Бот може отримувати повідомлення та інші оновлення, що стосуються користувачів чи чатів, через механізм вебхуків або запитів Long Polling.

2. Надсилання повідомлень: Бот може надсилати текстові повідомлення, а також повідомлення з медіафайлами (фотографії, відео, аудіо, документи), стікери, голосові повідомлення, місцезнаходження та інші типи повідомлень.

3. Керування клавіатурою: Бот може створювати візуальні клавіатури для зручної взаємодії з користувачами, включаючи клавіатури з кнопками та варіантами вибору.

4. Робота з фото та відео: Бот може отримувати, обробляти та надсилати фотографії та відеофайли. Він також може створювати альбоми з фотографій та відео.

5. Управління чатами: Бот може отримувати інформацію про чати, в яких він знаходиться, включаючи ідентифікатори чатів та користувачів. Він може здійснювати керування членством у групових чатах, налаштуванням прав доступу, отриманням списку учасників тощо.

6. Робота з клавішними командами: Бот може реагувати на клавішні команди користувачів, що починаються з "/". Наприклад, команда `/start` може ініціювати спеціальний функціонал або повідомлення від бота.

7. Інтерація з іншими сервісами: Бот може інтегруватися з іншими сервісами та API для отримання додаткової інформації або виконання певних дій. Наприклад, бот може взаємодіяти зі службою перекладу, отримувати дані зі сторонніх додатків або сервісів.

1.3 Огляд існуючих рішень

Це груповий чат для обговорення тем, пов'язаних із підтвердженням особи, наприклад підтвердження ідентифікаційного документа, процедури КУС (знай свого клієнта) та запобігання шахрайству. Чат є публічним, тобто будь-хто може приєднатися та взяти участь у обговореннях (рис.1.5).

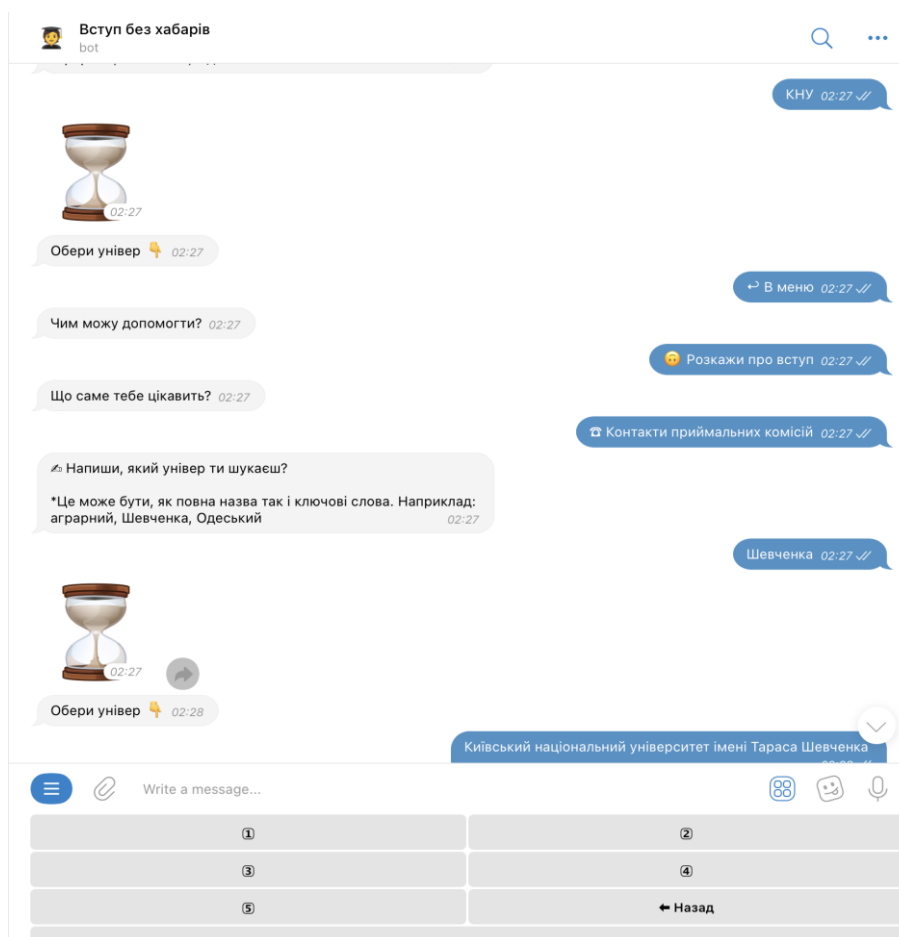


Рисунок 1.5 – Приклад чат бота

В даному чат боті можна дізнатись дані про різні університети України, отримати деякі загальновідомі дані про вступ, тощо (рис.1.6).

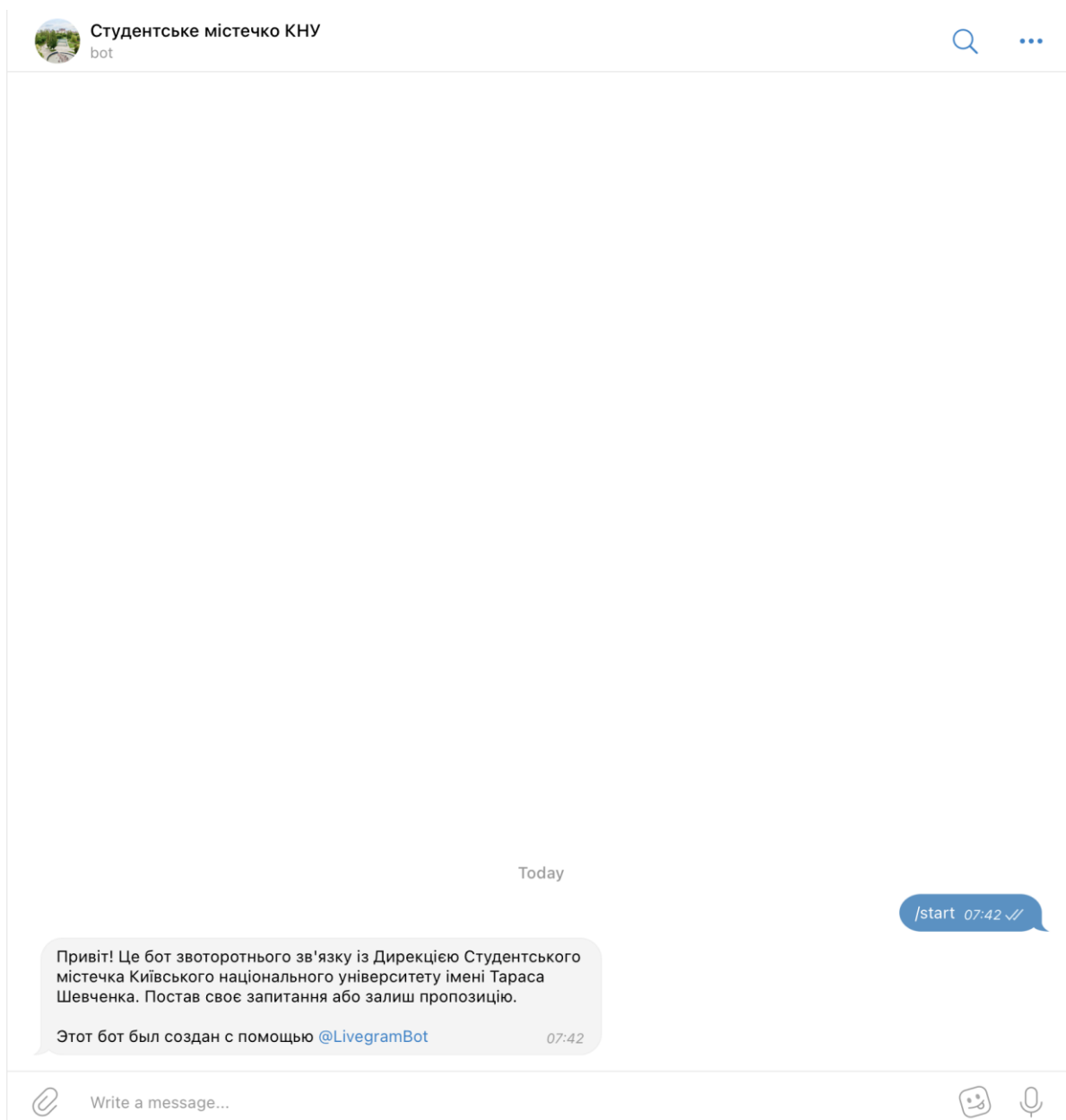


Рисунок 1.6 – Приклад чат бота

Даний чат бот надає можливість зворотного зв'язку із Дирекцією Студентського містечка Київського національного університету імені Тараса Шевченка.

Проаналізувавши телеграм на наявність чат ботів для університетів, стало зрозуміло що створення такого бота для нашої кафедри є актуальною проблемою. Адже практично у всіх університетів відсутній такий помічник для вступників.

Тому метою дипломної роботи є надання цифрового сервісу на основі створеного чат-бота щодо навчання. Для досягнення поставленої мети було сформовано наступні завдання:

- Провести аналіз концепції телеграм ботів.
- Провести аналіз допоміжних засобів для створення цифрових ботів.
- Розробити телеграм бот, що надаватиме навчальні консультації студентам та абітурієнтам факультету. Бот повинен мати зрозумілий і легкий інтерфейс, та при цьому повністю задовільняти вимоги згідно теми.

Також, на основі огляду відомих нині досліджень по темі кваліфікаційної роботи виконується аналіз поставленого завдання з посиланнями на джерела інформації.

Висновки до розділу 1

В даному розділі було описано історію появи телеграм ботів, їх концепцію і загальні поняття ботів. Загалом було описано сфери застосування чат бота для допомоги, описано концепцію чат ботів у контексті Telegram та описано існуючі рішення.

РОЗДІЛ 2

АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТІВ

2.1. Класифікація чат-ботів

Коли йдеться про розробку та впровадження чат-ботів у різні види людської діяльності, багато уваги приділяють як місцевим, так і міжнародним дослідженням.

Месенджери використовуються для різноманітних завдань, які виходять за рамки звичайного обміну текстовими повідомленнями: взаємодія з компаніями, пошук необхідних предметів і збереження вмісту таких обмінів [10].

Чат-боти розроблені як для приватного, так і для професійного спілкування. Особисті чат-боти є додатковими помічниками для кореспондента, а також можуть створювати аудіо- та відеофайли, вести календар, отримувати твіти та надсилати запити [1].

Бізнес-чат-боти створені спеціально для компаній і розроблені, щоб навчити клієнтів спілкуватися, допомагаючи їм у маркетингу, продажах та інших обов'язках (табл. 1.1).

Шлях створення та використання чат-ботів постійно змінюється, що вимагає більш глибокого вивчення та захисту використовуваних платформ, фреймворків і методологій. Ми проведемо широке дослідження чат-ботів. Спочатку необхідно з'ясувати безліч категорій і різновидів чат-ботів, які існують [1].

Можна класифікувати наступні ознаки: користувач, інтерфейс, призначення, доступ і принцип роботи. У таблиці показано, як чат-боти можна класифікувати в цілому. 1.1.

Таблиця 1.1 - Класифікація чат - ботів

Ознака	Вид
Користувач	Персональний, Бізнес
Інтерфейс користувача	Текстовий, Кнопковий, Голосовий
Призначення	Комунікаційний, Функціональний
Доступ	Доданий до групи, За підпискою, Вбудований в діалог
Принцип роботи	Шаблонний, Який навчається

Для спілкування з користувачем чат-бот може використовувати кнопки, текстові або голосові інструкції. Інтерфейс кнопкового чат-бота складається з кнопок та інструкцій. Натискаючи відповідні кнопки, користувач отримує вибір категорій, запитань або пропозицій, які можуть його зацікавити під час чату.

У цій ситуації кнопки є формою команди. Найчастіше команди визначаються під час створення; рідко вони можуть змінюватися під час виконання роботи [1].

Хоча текстовий чат-бот має певні практичні якості, спілкування з ним можна порівняти з спілкуванням із реальною людиною. Незважаючи на те, що вони потенційно більш універсальні, ніж кнопкові боти, навички цих ботів все ж обмежені їхніми знаннями та досвідом роботи з певними текстами. Експерти вважають, що найскладніше керування чат-ботом – голосове керування. Платформи відрізняються підтримкою цього типу керування. Користувач надає

аудіоповідомлення боту, який розпізнає та перетворює голос у текст перед його аналізом та формуванням відповіді [17].

Існують способи інтеграції та вдосконалення інтерфейсів чат-ботів. Доступ можна отримати трьома різними способами: груповий, за підпискою та вбудований. Декілька користувачів можуть отримувати доступ до послуг одночасно, використовуючи груповий режим доступу. Підписка на чат-бот призначена виключно для індивідуального використання.

За допомогою спеціальної команди вбудовані боти можуть бути викликані в будь-якому обговоренні. Ці чат-боти виконують певне завдання та оперативно повідомляють результат користувачеві [18]. Залежно від того, як вони функціонують, чат-боти можна розділити на дві групи: ті, які дотримуються попередньо визначеного шаблону, і ті, які навчаються через розмову.

Шаблонні чат-боти працюють лише в рамках обмежень жорстко заданого логічного автомата — дерева рішень. Їхні розмови часто організовані та лінійні. Розробка навчальних чат-ботів базується на технологіях ШІ. Їхній метод роботи побудований на постійному аналізі розмови з метою подальшого розвитку їхніх комунікаційних здібностей.

2.2 Вибір засобів розробки

Python — це популярна мова програмування для створення чат-ботів у Telegram через її читабельність, простоту та велику кількість інструментів і фреймворків, які полегшують створення та розгортання чат-ботів. Оскільки Python є інтерпретованою мовою, код можна запускати негайно без необхідності окремого етапу компіляції, що робить його ідеальним для швидкого створення прототипів і розробки. Крім того, Python має значну та активну спільноту розробників, яка пропонує велику кількість інформації та допомоги для програмістів, які використовують цю мову [8].

Розробники можуть використовувати Telegram's Bot API для створення чат-ботів Python. Розробники можуть надсилати й отримувати повідомлення, контролювати взаємодію користувачів і отримувати доступ до інших можливостей платформи Telegram через API Telegram Bot. Використовуючи лише кілька рядків коду, ви можете створити робочого чат-бота за допомогою бібліотек Python, таких як `python-telegram-bot` і `telepot`, які пропонують прості оболонки через API Telegram Bot.

Кілька модулів обробки природної мови (NLP) у Python, як-от `spaCy`, `NLTK` і `TextBlob`, спрощують розуміння та відповідь на введені користувачем дані більш розмовною манерою. Ці бібліотеки включають методи аналізу настроїв, токенизації та синтаксичного аналізу, які можна використовувати для інтерпретації введених користувачем даних і отримання відповідних відповідей [12].

Python - це потужна та популярна мова програмування з великою кількістю переваг. Однак, як і в будь-якій мові, вона також має свої плюси та мінуси. Ось кілька основних переваг і недоліків Python:

Плюси Python[13]:

1. Простота вивчення: Python має простий і лаконічний синтаксис, що дозволяє легко вивчити основи програмування. Це робить його ідеальним вибором для початківців.
2. Велика спільнота розробників: Python має активну та велику спільноту розробників, що означає наявність багатьох бібліотек, фреймворків та ресурсів для розвитку проектів. Це сприяє швидкому розвитку та підтримці мови.
3. Універсальність: Python може бути використаний для розробки різноманітних додатків, включаючи веб-програмування, наукові обчислення, штучний інтелект, обробку даних та багато іншого. Він підтримує багато платформ і операційних систем.

4. Велика кількість бібліотек: Python має велику кількість сторонніх бібліотек, що спрощують розробку і забезпечують доступ до різних функцій. Наприклад, бібліотека NumPy для обробки числових даних, бібліотека Pandas для роботи з даними та багато інших.

5. Висока продуктивність: При правильному використанні та оптимізації, Python може бути досить продуктивним. Крім того, з використанням відповідних бібліотек, таких як Cython або PyPy, можна досягти ще більшої швидкодії.

Мінуси Python[14]:

1. Швидкість виконання: Python є інтерпретованою мовою, що може призводити до певного зниження швидкості виконання порівняно з компільованими мовами, такими як C++ або Java. Однак, це часто не є критичним фактором для багатьох додатків.

2. Обмеження мобільної розробки: Python не є найпопулярнішою мовою для розробки мобільних додатків. Хоча існують фреймворки, такі як Kivy або PySide, які дозволяють розробляти мобільні додатки на Python, їх популярність ще не така велика, як у Java або Swift.

3. Високе використання пам'яті: Python має високе використання пам'яті порівняно з іншими мовами програмування. Це може бути проблемою, особливо при роботі з об'ємними даними або в обмежених середовищах.

4. Global Interpreter Lock (GIL): GIL є механізмом, який обмежує можливість одночасного виконання потоків (threads) Python на рівні інтерпретатора. Це може призводити до обмеження масштабованості багатопоточних програм.

Хоча Python має деякі недоліки, він є потужним та гнучким інструментом для багатьох видів розробки програмного забезпечення.

Загалом Python є оптимальним вибором для розробки чат-ботів Telegram завдяки його простоті, адаптивності та надійній екосистемі інструментів. Його широкий набір функцій і потужні можливості NLP дозволяють розробляти

розумні та розважальні чат-боти, які можуть надавати користувачам індивідуальну допомогу та підтримку.

Існує кілька допоміжних програм і засобів розробки, які можна використовувати для прискорення та покращення процесу розробки програми для розділення файлів. Нижче наведено кілька найпопулярніших інструментів:

1. **Інтегровані середовища розробки (IDE):** такі IDE, як Eclipse, Visual Studio та NetBeans, надають ряд функцій, які роблять розробку швидшою та легшою, наприклад підсвічування коду, автозавершення та налагодження(рис.2.1-2.3).

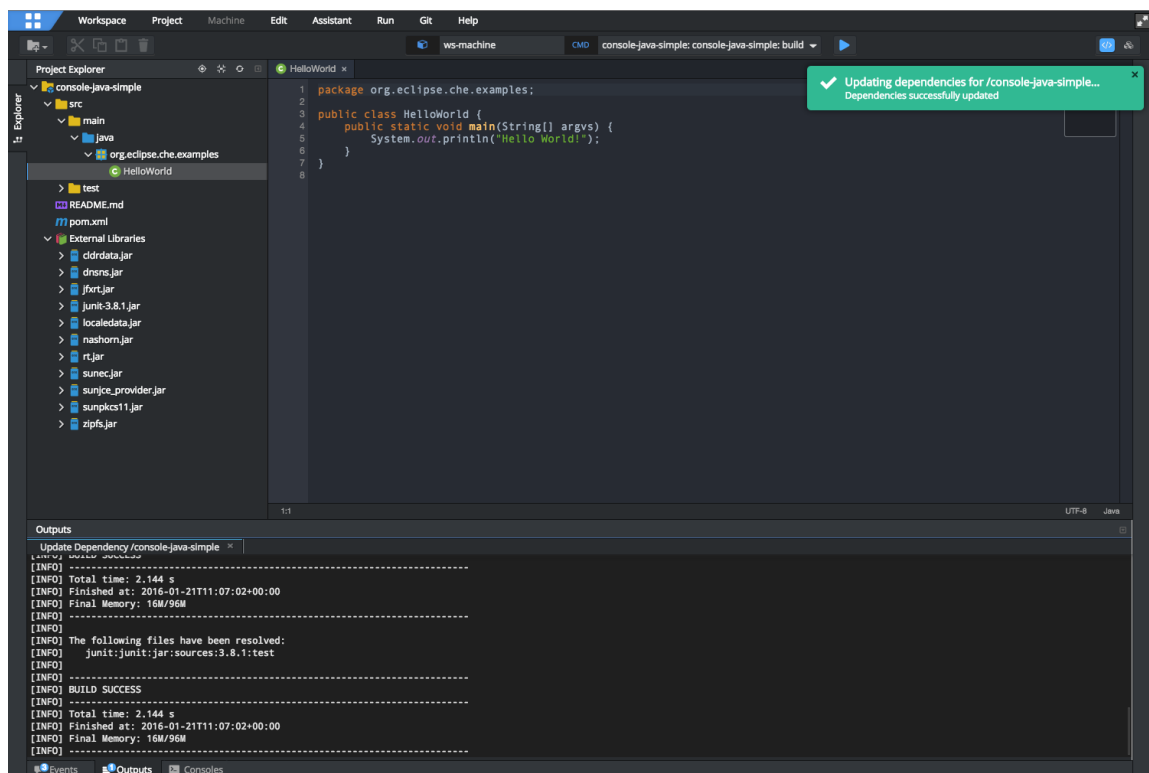


Рисунок 2.1 – Середовище розробки Eclipse

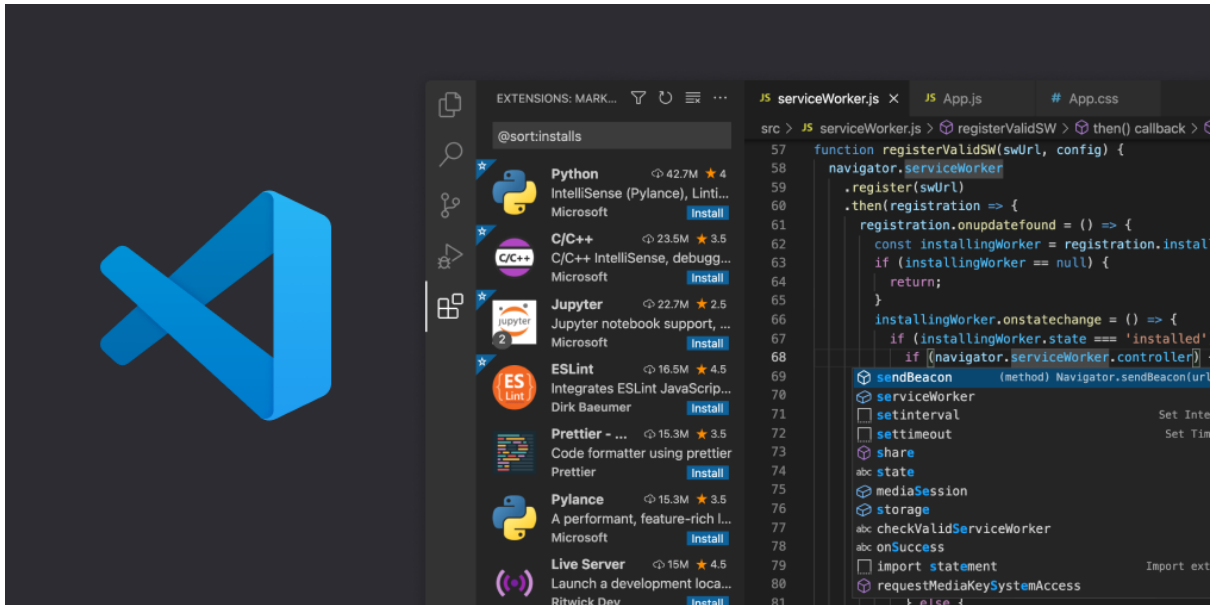


Рисунок 2.2 – Середовище розробки VS

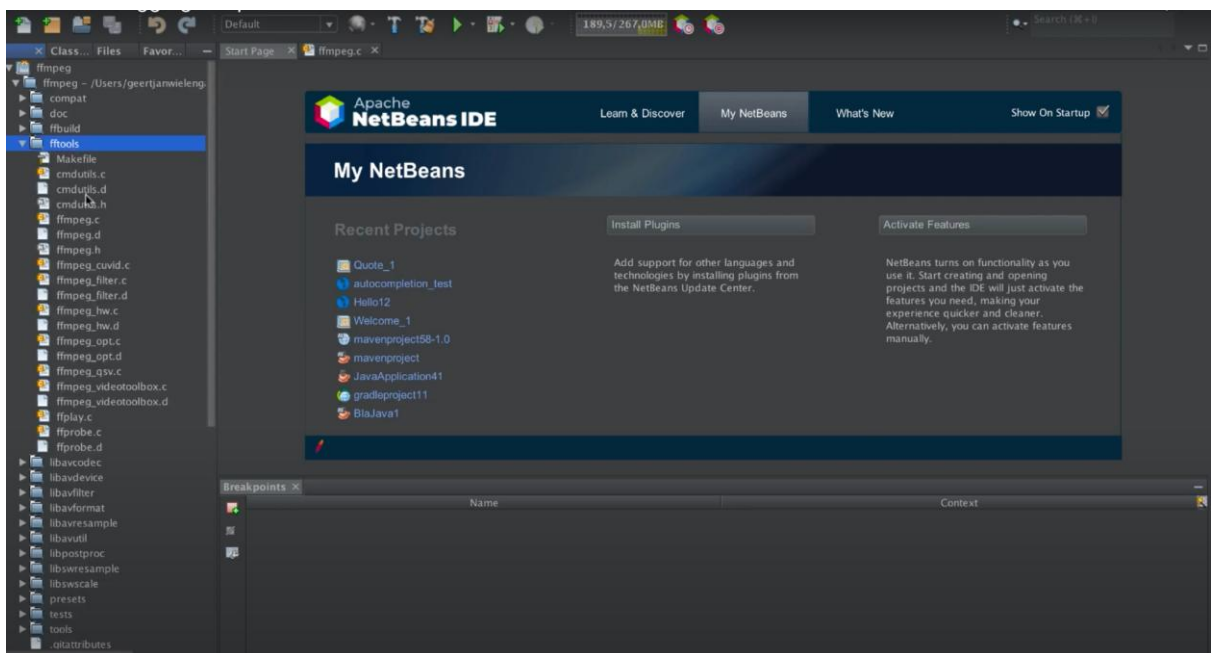


Рисунок 2.3 – Середовище розробки NetBeans

Обирати середовище розробки можна залежно від мови, вподобань, операційної системи тощо.

2. **Системи контролю версій:** Git і SVN — це два інструменти контролю версій, які використовуються для моніторингу проблем, керування змінами кодової бази та взаємодії з іншими розробниками(рис.2.4).

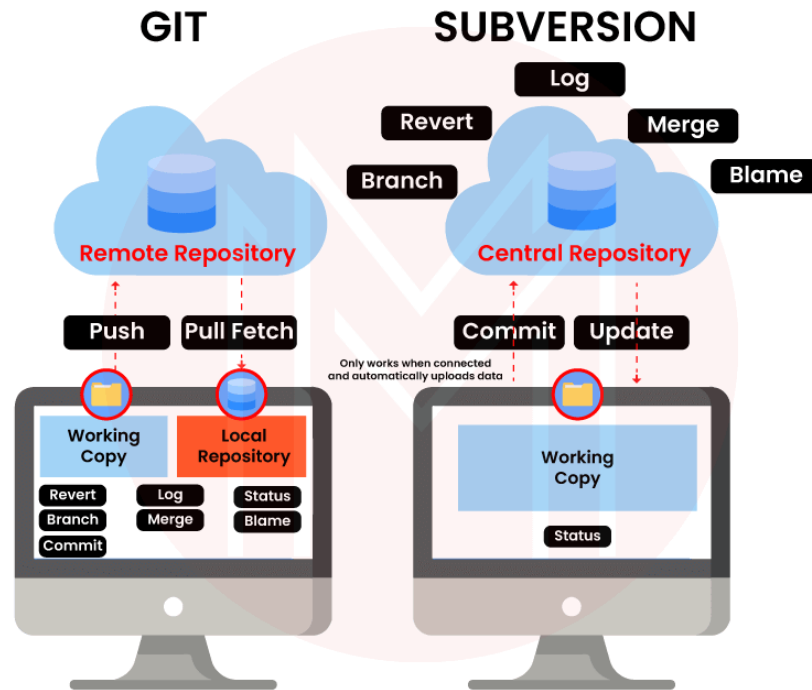


Рисунок 2.4 – Система контролю версій Git та SVN

3. **Бібліотеки та фреймворки:** Бот використовує API Telegram і бібліотеки Python, як telegram і requests для реалізації різних функцій, таких як запуск бота, надсилання довідкових повідомлень, автентифікація користувачів, пошук ресурсів за місцезнаходженням користувача та зіставлення користувачів із доступними ресурсами на основі їхнього розташування та потреб. Бібліотека telegram використовується для створення Bot об'єкта та надсилання повідомлень користувачеві, тоді як requests бібліотека використовується для надсилання HTTP-запитів до зовнішніх API, наприклад API Google Places. Бот використовує різні обробники, такі як CommandHandler, MessageHandler і Filters для обробки різних типів введених користувачами, таких як команди, повідомлення та розташування.
4. **Інструменти тестування:** інструменти тестування, такі як JUnit, NUnit і TestNG, можна використовувати для автоматизації тестування та забезпечення правильного функціонування програмного забезпечення.

5. **Інструменти збірки.** Інструменти збірки, такі як Maven, Gradle і Ant, автоматизують процес збирання, спрощуючи упаковку програмного забезпечення та розгортання його у виробництві.
6. **Інструменти документації:** такі інструменти, як Doxygen і Javadoc, можна використовувати для автоматичного створення документації для програмного забезпечення, що полегшує розробникам розуміння того, як використовувати програмне забезпечення та як воно працює.

Загалом, використання допоміжного програмного забезпечення та засобів розробки може значно покращити процес розробки утиліти розділювача файлів. Ці інструменти можуть допомогти скоротити час розробки, покращити якість коду та спростити співпрацю з іншими розробниками.

2.3 Архітектура програмного забезпечення

Розробники можуть створювати та спілкуватися з ботами Telegram за допомогою колекції API, яка називається Telegram Bot API [6]. Все більш популярний хмарний сервіс обміну повідомленнями під назвою Telegram робить сильний акцент на безпеці та анонімності. По суті, боти Telegram — це автоматизовані програми, які можуть спілкуватися з користувачами, відповідати на їхні повідомлення та виконувати різні дії відповідно до введених користувачами.

Telegram Bot API надає розробникам широкий спектр функцій і функцій, як-от [6]:

1. Надсилання та отримання повідомлень: розробники можуть використовувати API для надсилання та отримання тексту, зображень та інших типів повідомлень.
2. Керування користувачами: API дозволяє розробникам керувати користувачами, наприклад додавати або видаляти користувачів із групи чату.
3. Спеціальні клавіатури: розробники можуть створювати власні клавіатури для своїх ботів, що дозволяє користувачам легше з ними взаємодіяти.

4. Вбудований режим: вбудований режим дозволяє розробникам вставляти своїх ботів у розмову в чаті, дозволяючи користувачам взаємодіяти з ботом, перебуваючи в розмові.

5. Веб-хуки: API дозволяє розробникам налаштовувати веб-хуки, які можна використовувати для отримання оновлень про взаємодію користувача з ботом.

6. Платежі: API включає підтримку платежів, що дозволяє розробникам створювати ботів, які можуть приймати платежі за товари чи послуги.

Загалом, Telegram Bot API надає розробникам потужний набір інструментів для створення та керування ботами Telegram. Широкий набір функцій і можливостей робить його популярним вибором для розробників, які хочуть створювати ботів для різних цілей, включаючи підтримку клієнтів, доставку новин тощо.

Діаграма використання

Діаграма використання (Use Case diagram) є одним з типів діаграм UML, який моделює функціональність системи з точки зору взаємодії між акторами (користувачами) та системою [9]. Вона використовується для визначення функцій або можливостей системи та відображення взаємодії між акторами та використовуваними випадками використання[20].

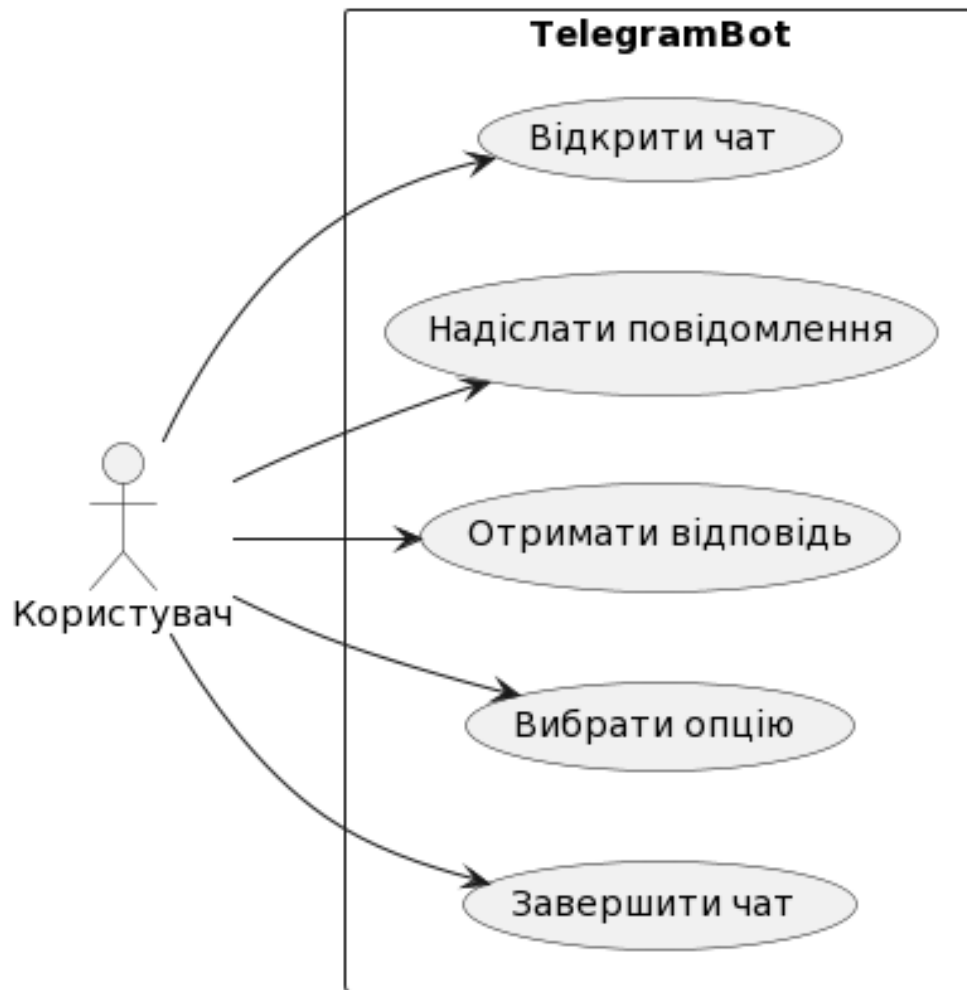


Рисунок 2.5 - Діаграма використання

Перелік юз-кейсів(рис.2.5):

1. Старт: Цей юз-кейс відображає початок взаємодії з системою. Користувач запускає додаток і бачить головне меню з доступними опціями.
2. Вибір опції: Користувач вибирає одну з доступних опцій, таких як "Вступні іспити", "FAQ", "Освітні програми" і т.д.
3. Вступні іспити: Користувач вибирає цю опцію, щоб отримати інформацію про вступні іспити. Система відображає розклад інтерв'ю, або надає необхідну інформацію про вступний процес.
4. FAQ: Користувач вибирає опцію FAQ, щоб переглянути список часто задаваних питань та відповіді на них. Система відображає список питань і

надає користувачеві можливість вибрати певне питання для перегляду відповіді.

5. Освітні програми: Користувач вибирає опцію "Освітні програми", щоб отримати інформацію про освітні програми, які пропонуються. Система надає список доступних програм та посилання на детальнішу інформацію про кожну програму.

6. Розклад: Користувач вибирає опцію "Розклад" і отримує посилання на розклад зовнішнього ресурсу, наприклад, таблиці Google Sheets з розкладом занять.

7. Зв'язатися з нами: Користувач вибирає цю опцію, щоб зв'язатися з адміністратором системи. Система пересилає надсилає повідомлення з інформацією щодо номеру телефону та електронною адресою факультета.

Діаграма послідовності

Діаграма послідовності відображає послідовність виконання повідомлень (методів) між об'єктами в часі [9]. Вона показує, як об'єкти взаємодіють між собою через виклики методів та передачу повідомлень. Діаграма може містити об'єкти, лінії життєвого циклу об'єктів, повідомлення (виклики методів), асоціації між об'єктами, умови та інші важливі деталі.

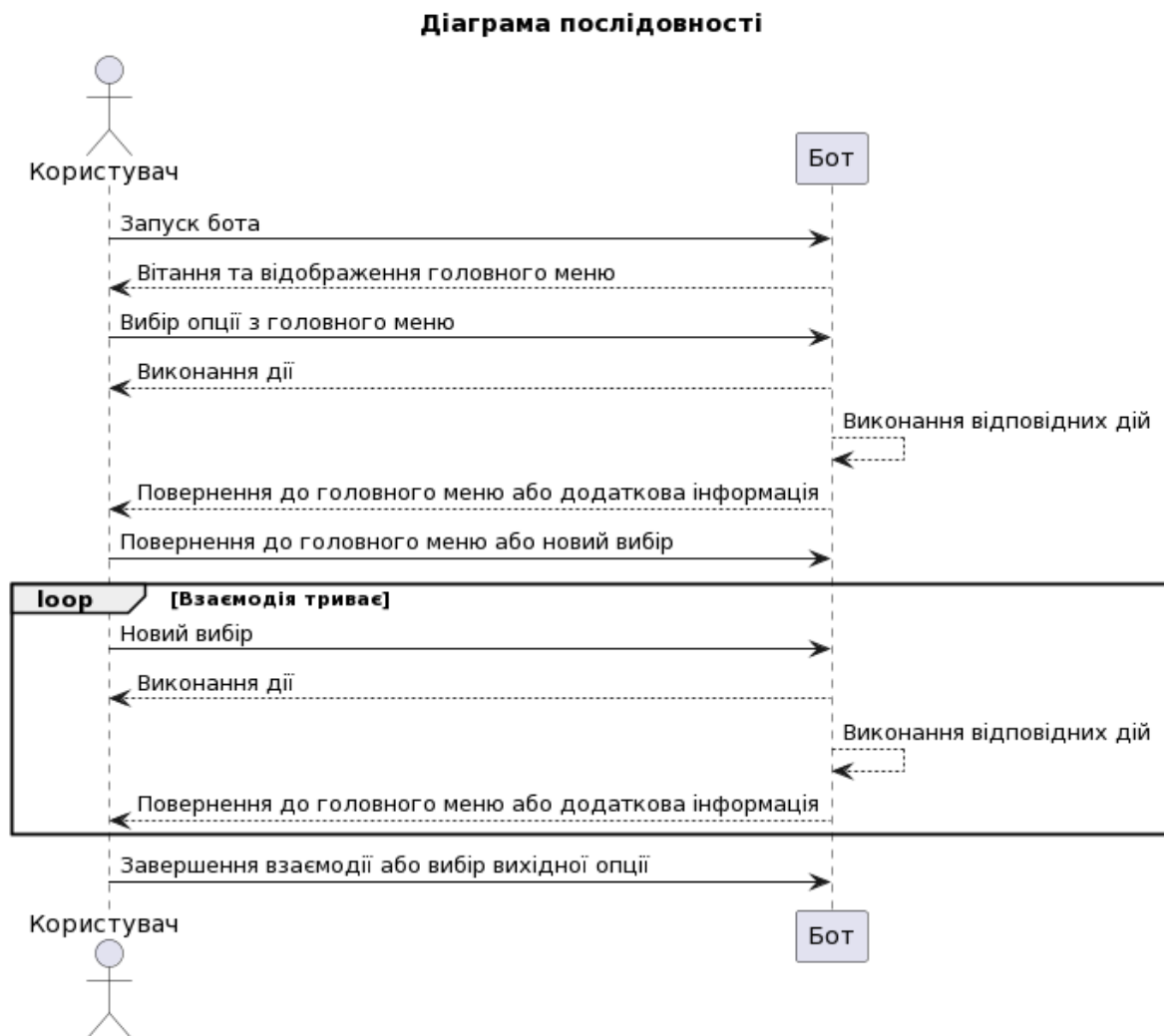


Рисунок 2.6 - Діаграма послідовності

Ця діаграма використання показує взаємодію між користувачем і телеграм-ботом. Користувач спілкується з ботом, використовуючи команду /start і вибираючи опції з меню. Бот обробляє вибір користувача і відображає відповідну інформацію про вступні іспити, FAQ та освітні програми. Користувач може повернутись до попереднього меню або завершити взаємодію з ботом (рис.2.6).

Діаграма активності

Діаграма активностей використовується для візуалізації кроків, що відбуваються в системі, а також потоку управління між цими кроками [9]. Вона

складається з активностей (дій), рішень, розгалужень, злиття, функцій, переходів та інших елементів.

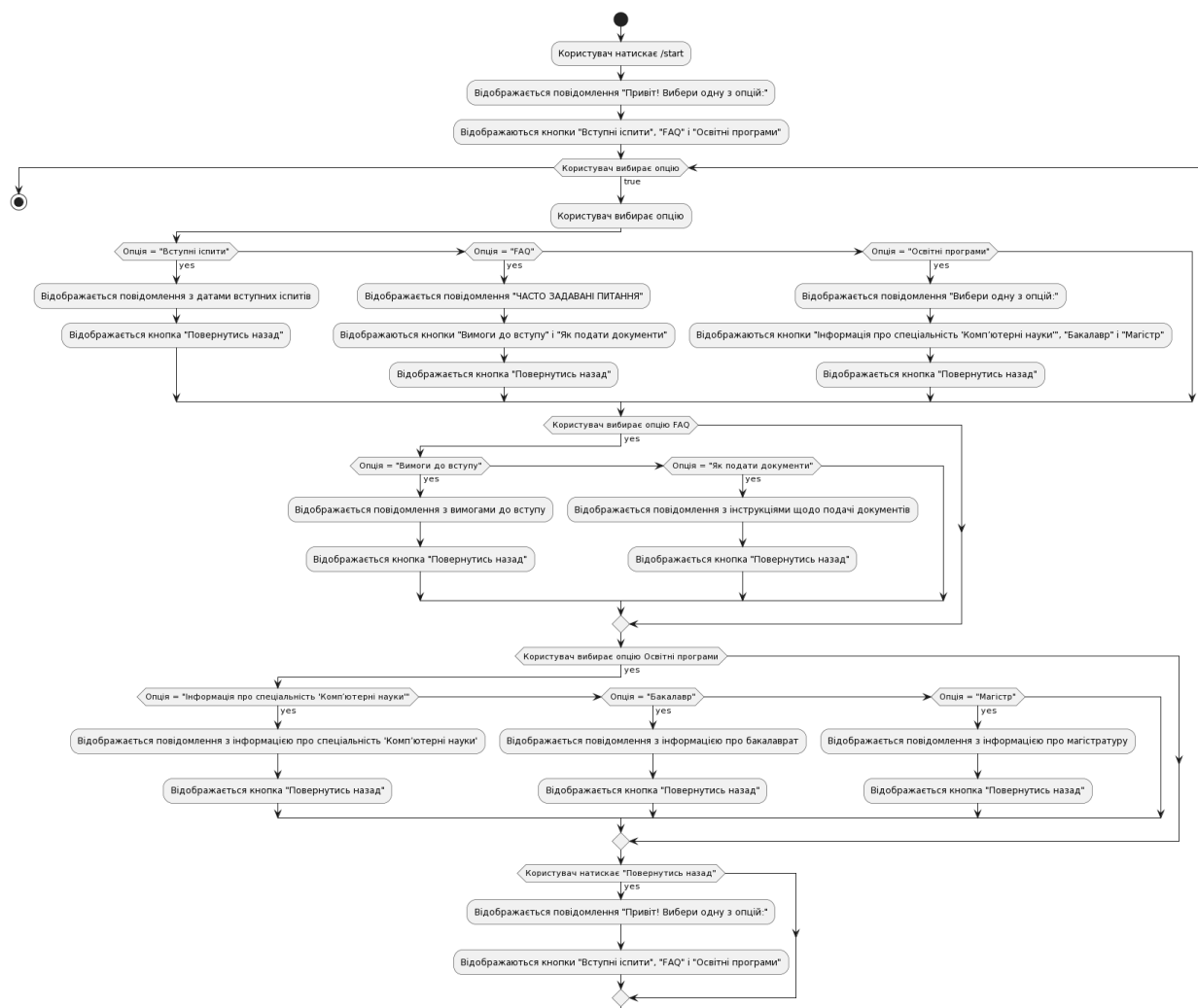


Рисунок 2.7 - Діаграма активності

Діаграма відображає взаємодію користувача з Телеграм-ботом, що надає різні опції і відповідні відповіді (рис.2.7).

1. Користувач запускає бота - це початок взаємодії користувача з ботом.
2. Бот відображає головне меню - після запуску, бот виводить головне меню з доступними опціями для користувача.
3. Користувач вибирає опцію - користувач вибирає одну з опцій, натискаючи на відповідну кнопку у головному меню.

4. Бот відображає інформацію про вступні іспити - якщо користувач обрав опцію "Вступні іспити", бот відображає інформацію про дати іспитів.

5. Користувач може повернутися до головного меню - після перегляду інформації про вступні іспити, користувач має можливість повернутися до головного меню для вибору інших опцій.

6. Бот відображає часто задавані питання - якщо користувач обрав опцію "ЧАСТО ЗАДАВАНІ ПИТАННЯ", бот відображає список часто задаваних питань.

7. Користувач може вибрати конкретне питання або повернутися до головного меню - після перегляду списку питань, користувач має можливість вибрати конкретне питання для перегляду відповіді або повернутися до головного меню.

8. Бот відображає освітні програми - якщо користувач обрав опцію "Освітні програми", бот відображає список доступних освітніх програм.

9. Користувач може вибрати програму або повернутися до головного меню - після перегляду списку освітніх програм, користувач має можливість вибрати конкретну програму для отримання додаткової інформації або повернутися до головного меню.

10. Бот відкриває розклад у веб-браузері - якщо користувач обрав опцію "Розклад", бот відкриває розклад у веб-браузері для перегляду.

11. Бот відкриває посилання на гайд у веб-браузері - якщо користувач обрав опцію "Гайд ЄВІ", бот відкриває посилання на гайд у веб-браузері для перегляду.

12. Обрано недійсну опцію - якщо користувач обрав недійсну опцію або ввів неправильну команду, бот відображає повідомлення про помилку.

13. Бот відображає повідомлення про помилку - у випадку, коли обрано недійсну опцію або введено неправильну команду, бот виводить повідомлення про помилку.

14. Користувач може повернутися до головного меню - після отримання повідомлення про помилку, користувач може повернутися до головного меню для вибору інших опцій.

Висновки до розділу 2

В даному розділі описано обрані засоби розробки, описано актуальність дослідження та створення телеграм ботів. Також було описано архітектуру програмного забезпечення, а саме побудовано діаграму використання, діаграму послідовності та діаграму активності.

РОЗДІЛ 3

ОПИС РОБОТИ СИСТЕМИ

3.1 Постановка задач

Завдання дослідження:

1. Вивчення потреб та очікувань користувачів: Здійснення опитувань, інтерв'ю та аналіз з метою розуміння потреб та очікувань студентів щодо навчальних консультацій та використання чат-бота.

2. Розробка інтерактивного інтерфейсу чат-бота: Проектування та розробка користувацького інтерфейсу чат-бота, який буде забезпечувати зручну та ефективну навігацію та взаємодію з користувачами.

3. Підбір та навчання алгоритмів обробки запитань та надання консультацій: Вибір та розробка алгоритмів обробки текстових запитань користувачів та надання відповідей з необхідною інформацією та консультаціями.

4. Оцінка ефективності та задоволеності користувачів: Здійснення експериментальних тестувань та опитувань користувачів для оцінки ефективності та якості роботи чат-бота, а також задоволеності користувачів від використання навчальних консультацій через чат-бот.

5. Впровадження та використання чат-бота: Розгортання розробленого чат-бота на платформі, активне використання його студентами для отримання навчальної підтримки та збір фідбеку для подальшого вдосконалення.

3.2 Розробка програмного забезпечення

Функціональні вимоги визначають, які конкретні функції або задачі має виконувати система або програмний продукт. Це описує очікувані результати, що повинні бути досягнуті за допомогою програмного продукту [7].

Функціональні вимоги:

1. Привітання: Чат-бот повинен вітати користувача із запитом на початку розмови.

2. Інформаційні питання: Чат-бот повинен вміти відповідати на загальні запитання вступників, такі як: дати вступних іспитів, вимоги до документів для вступу, процес подачі заявки тощо.

3. Відповіді на питання про програми навчання: Чат-бот повинен знати про різні програми навчання, які пропонуються університетом, та надавати інформацію про них, наприклад, спеціалізації, плани навчання тощо.

4. Розклад вступних іспитів: Чат-бот може надавати інформацію про розклад вступних іспитів, місце їх проведення та процедуру реєстрації на них.

Нефункціональні вимоги описують характеристики системи, які не відносяться безпосередньо до функцій, але впливають на якість, надійність та інші аспекти роботи системи [7].

Нефункціональні вимоги:

1. Доступність: Чат-бот повинен бути доступним 24/7, щоб користувачі могли отримати інформацію в будь-який час.

2. Відповіді в реальному часі: Чат-бот повинен надавати відповіді користувачам в режимі реального часу, з мінімальним часом затримки.

3. Інтуїтивний інтерфейс: Інтерфейс чат-бота повинен бути простим і легким у використанні, щоб користувачі могли з легкістю отримати потрібну інформацію.

4. Масштабованість: Система чат-бота повинна бути здатна обслуговувати одночасно багато користувачів без втрати продуктивності.

5. Безпека: Чат-бот повинен забезпечувати захист конфіденційної інформації користувачів, такої як особисті дані.

Опис методів

Основна логіка полягає в обробці натискання кнопок та відправленні відповідних повідомлень користувачеві.

У методі `start` реалізовано початкове повідомлення та створення кнопок-опцій для вибору. Кнопки створюються з використанням класу `InlineKeyboardButton`, а потім об'єднуються у відповідну розмітку за допомогою класу `InlineKeyboardMarkup`. Функція `start` обробляє команду `/start` від користувача. Вона створює повідомлення з текстом і кнопками опцій за допомогою класу `InlineKeyboardButton`. Розмітка кнопок створюється за допомогою класу `InlineKeyboardMarkup`. Повідомлення разом з розміткою кнопок відправляються користувачеві.

Метод `button_click` відповідає за обробку натискання кнопок. Залежно від вибраної опції, відправляється відповідне повідомлення. Для кожної опції створюються відповідні кнопки та розмітка, щоб забезпечити навігацію між екранами. Функція `button_click` обробляє натискання кнопок користувача. Залежно від вибраної опції, відправляється відповідне повідомлення. Також в цій функції реалізовано кнопку "Повернутись назад", яка викликає функцію `start` та змінює розмітку кнопок для попереднього екрану.

Також у методі `button_click` реалізовано кнопку "Повернутись назад", яка викликає метод `start` та змінює розмітку кнопок для попереднього екрану.

У методі `__main__` створюється екземпляр `Updater`, встановлюється токен бота та додаються обробники команд та натискань кнопок. Запускається процес отримання оновлень ботом з використанням `start_polling()`. У цій функції створюється екземпляр `Updater` з переданим токеном вашого бота. Об'єкт `dispatcher` використовується для додавання обробників команд та натискань кнопок. Встановлення `use_context=True` забезпечує використання об'єкту `context` у функціях обробників. Нарешті, запускається процес отримання оновлень ботом з використанням `start_polling()`.

Давайте розглянемо опис кожного методу і їх роль у функціонуванні чат-бота.

Метод `start(update, context)`: Цей метод викликається при команді `/start` або при першому запуску бота. Він відправляє повідомлення з вибором опцій користувачу, використовуючи клавіатуру з кнопками `InlineKeyboardButton`.

Метод `button_click(update, context)`: Цей метод викликається при натисканні на кнопки клавіатури. Він перевіряє, яку кнопку було натиснуто, і відповідно до цього виконує відповідну дію.

Метод `main()`: Цей метод є точкою входу в програму. Він налаштовує `Updater` з використанням токена `Telegram Bot API`, створює диспетчер та додає обробники команд та повідомлень. Після цього він починає постійне опитування сервера `Telegram` для отримання оновлень та обробки подій.

В таблиці 3.1 описано методи, що використовуються для забезпечення роботи телеграм бота.

Таблиця 3.1 - Методи і їх опис

Метод	Опис
start(update, context)	Відправляє користувачеві початкове повідомлення з кнопками вибору опцій.
user_message(update, context)	Отримує текстове повідомлення від користувача та пересилає його адміну.
button_click(update, context)	Обробляє натискання кнопок клавіатури та виконує відповідні дії.
reply_user(update, context)	Отримує повідомлення від адміна та пересилає його користувачу.
main()	Головний метод, який налаштовує бота та починає процес опитування сервера Telegram.

Опис методу тестування

Тестування Telegram-ботів включає декілька етапів, які допомагають забезпечити якість та надійність роботи бота перед його випуском у

виробництво. Основні процеси тестування Telegram-ботів можуть включати наступні етапи:

1. Планування тестування: У цьому етапі визначаються мета тестування, функціональні вимоги та очікувані результати. Встановлюються критерії прийняття тестування та планується розклад тестування.

2. Розробка тестових сценаріїв: Тестові сценарії описують послідовність кроків, які повинні бути виконані для перевірки функціональності бота. Вони можуть включати введення команд, надсилання повідомлень, перевірку виведення результатів та інші дії.

3. Виконання тестових сценаріїв: Тестування виконується згідно зі складеними тестовими сценаріями. Це включає взаємодію з ботом, надсилання запитів, перевірку реакції бота та порівняння отриманих результатів з очікуваними.

4. Виявлення та відстеження дефектів: Якщо під час тестування виявляються проблеми або дефекти, вони фіксуються та документуються. Це можуть бути помилки в роботі бота, некоректні відповіді, затримки чи інші проблеми.

5. Валідація та верифікація: Валідація та верифікація полягають у перевірці того, чи відповідає бот заданим вимогам та специфікаціям. Це може включати перевірку правильності відповідей, обробку помилок та дотримання стандартів програмування.

6. Тестування на різних платформах: Telegram-боти можуть працювати на різних платформах, таких як веб, мобільні пристрої тощо. Тому важливо виконати тестування на різних платформах для перевірки сумісності та впевненості в їх правильній роботі.

7. Тестування безпеки: Оскільки Telegram використовується для обміну повідомленнями та передачі даних, тестування безпеки важливо для забезпечення захисту особистих даних користувачів. Це може включати

перевірку на наявність потенційних уразливостей, атак на бота та перехоплення даних.

8. Тестування навантаження: Цей вид тестування допомагає перевірити продуктивність бота під навантаженням. Його мета - визначити, як бот реагує на велику кількість запитів та повідомлень і чи зберігає він стабільність та продуктивність.

9. Автоматизація тестування: Щоб забезпечити ефективність та швидкість тестування, можна використовувати автоматизацію тестування. Це дозволяє автоматизувати виконання тестових сценаріїв, виявлення дефектів та генерацію звітів.

Після завершення тестування зазвичай проводиться аналіз результатів, виправлення виявлених дефектів та повторне тестування, поки бот не відповідає всім вимогам та стандартам якості. Для тестування було обрано функціональне тестування.

В таблиці 3.2 описано мінімальні вимоги, що необхідні для забезпечення стійкої роботи телеграм бота.

Таблиця 3.2 – Мінімальні системні вимоги для запуску телеграм бота

Операційна система	Windows 7 , macOS 10.12, Ubuntu 16.04
CPU (процесор)	Intel Core 3 та вище
RAM (оперативна пам'ять)	2 GB
Об'єм пам'яті на диску	2 GB

В таблиці 3.3 описано тестування, що необхідні для перевірки функціоналу початку роботи бота. Тест містить назву сцени, на якій проводиться випробування, схему проведення тесту, очікуваний результат, а також результат випробування.

Таблиця 3.3 – Перевірка функціоналу запуску бота

Мета тесту	Перевірка функціоналу початку роботи бота
Назва сцени, на якій проводиться випробування	Початковий екран
Схема проведення тесту	Користувач натискає команду “Start”
Очікуваний результат	Бот надсилає повідомлення з проханням вибрати команду
Результат випробування	Результат збігається з очікуваним

В таблиці 3.4 описано тестування, що необхідні для перевірки функціоналу отримання інформації.

Таблиця 3.4 – Перевірка функціоналу отримання інформації

Мета тесту	Перевірка отримання інформації
Назва сцени, на якій проводиться випробування	Початковий екран
Схема проведення тесту	Користувач вибирає команду
Очікуваний результат	Користувач отримує інформацію
Результат випробування	Результат збігається з очікуваним

В таблиці 3.5 описано тестування, що необхідні для перевірки функціоналу роботи кнопки «Вступні іспити».

Таблиця 3.5 – Перевірка вибору команди “Вступні іспити”

Мета тесту	Перевірка роботи кнопок
Назва сцени, на якій проводиться випробування	Бот запущено
Схема проведення тесту	Користувач вибирає команду “Вступні іспити”
Очікуваний результат	Користувач отримує інформацію про “Вступні іспити”
Результат випробування	Результат збігається з очікуваним

В таблиці 3.6 описано тестування, що необхідні для перевірки функціоналу роботи кнопки «FAQ».

Таблиця 3.6 – Перевірка вибору команди “FAQ”

Мета тесту	Перевірка роботи кнопок
Назва сцени, на якій проводиться випробування	Бот запущено
Схема проведення тесту	Користувач вибирає команду “FAQ”
Очікуваний результат	Користувач отримує інформацію про “FAQ”
Результат випробування	Результат збігається з очікуваним

В таблиці 3.7 описано тестування, що необхідні для перевірки функціоналу роботи кнопки «Освітні програми».

Таблиця 3.7 – Перевірка вибору команди “Освітні програми”

Мета тесту	Перевірка роботи кнопок
Назва сцени, на якій проводиться випробування	Бот запущено
Схема проведення тесту	Користувач вибирає команду “Освітні програми”
Очікуваний результат	Користувач отримує інформацію про “Освітні програми”
Результат випробування	Результат збігається з очікуваним

3.3 Інструкція користувача

Для початку роботи потрібно відшукати телеграм ботам в пошуку, та для його запуску треба ввести команду /start.

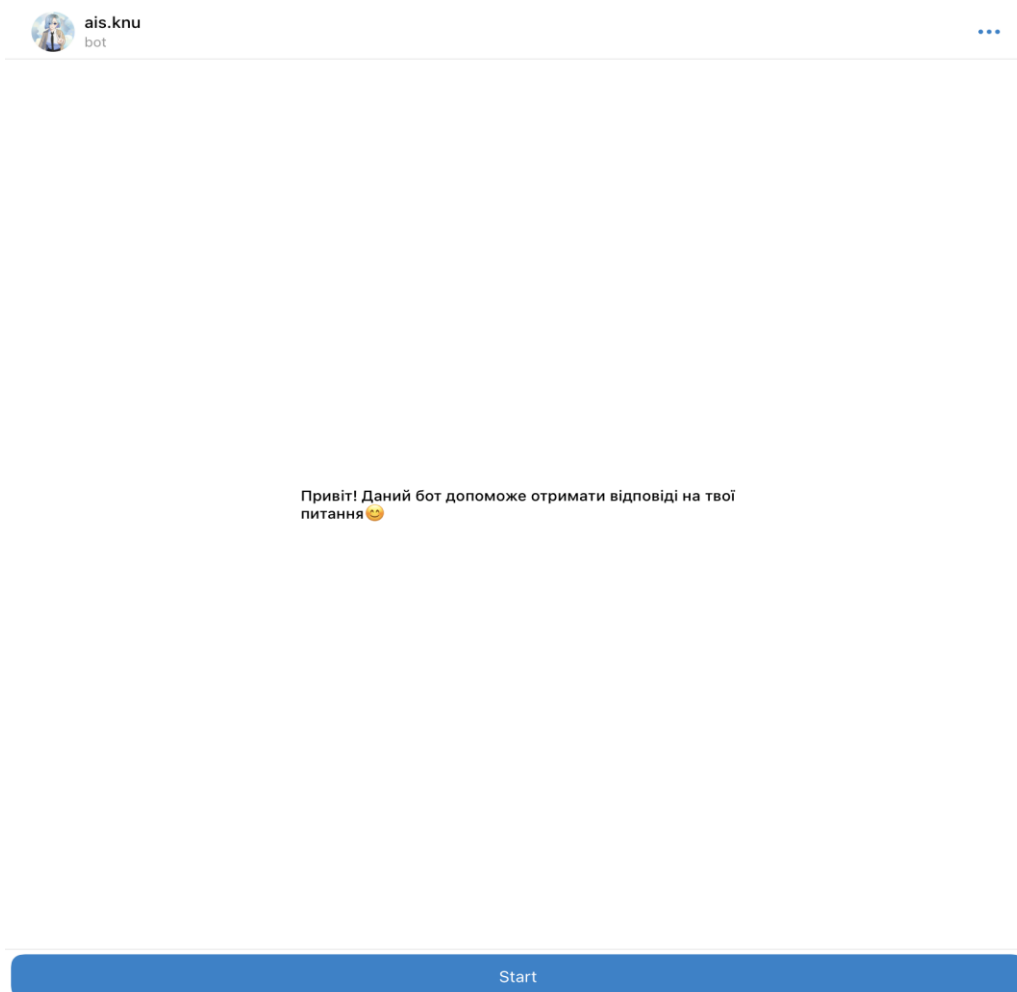


Рисунок 3.1 - Початок роботи

На рис.3.5 зображено опис телеграм бота.

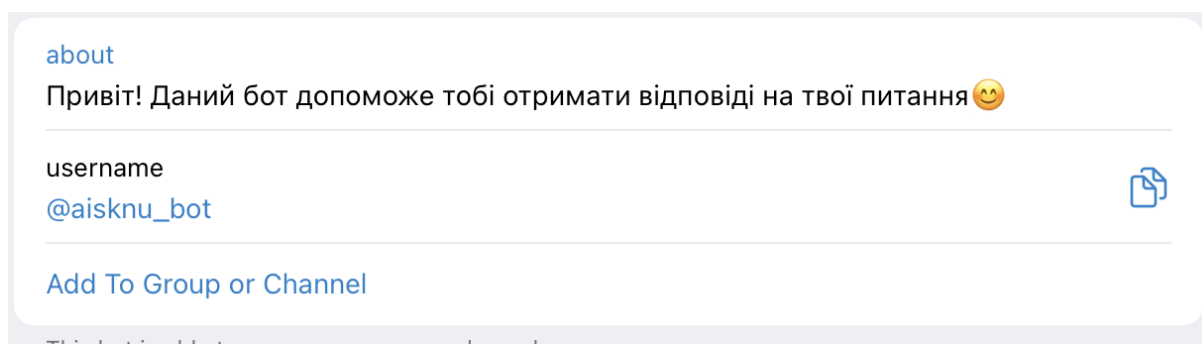


Рисунок 3.2 - Опис

Після запуску бота користувач отримує повідомлення з проханням вибрати відповідну команду.

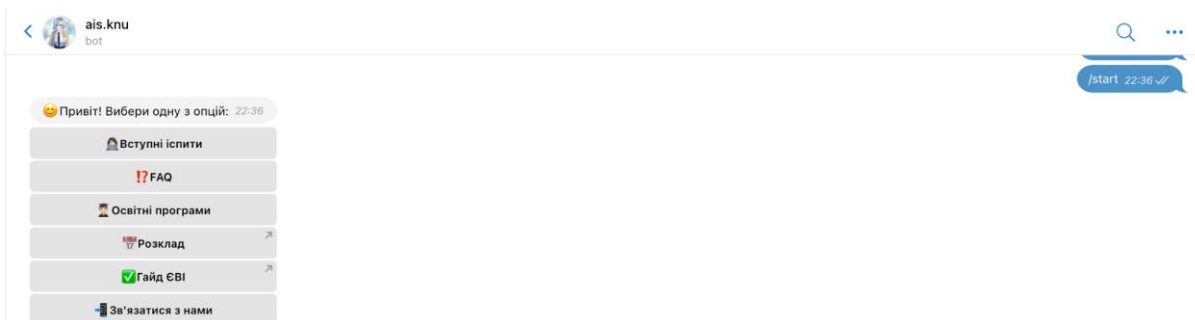


Рисунок 3.3 - Робота бота

Щоб отримати інформацію стосовно вступних іспитів потрібно натиснути кнопку “Вступні іспити” , після чого користувач отримує відповідну інформацію про вступні іспити на факультет. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

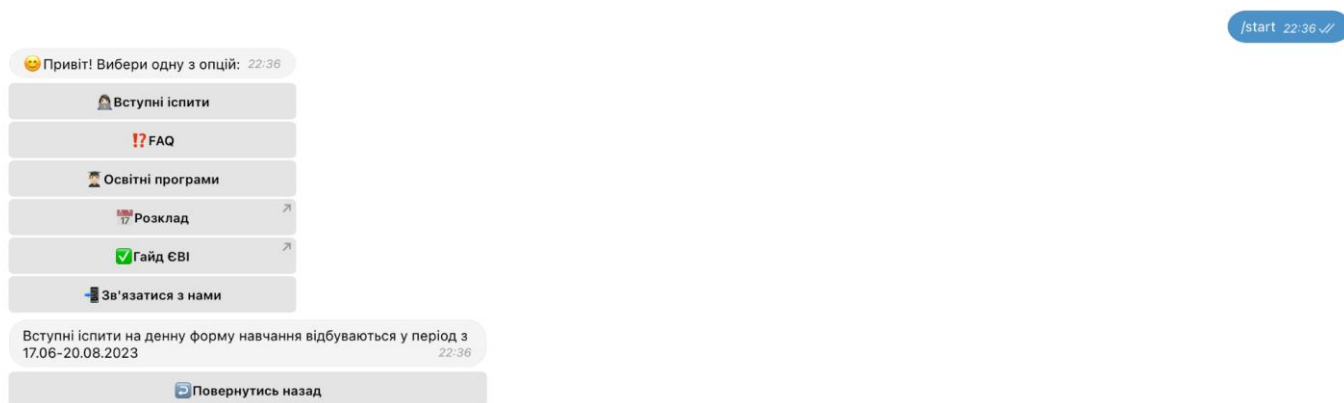
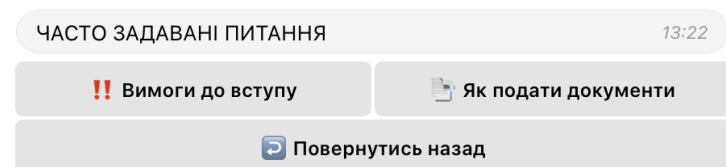


Рисунок 3.4 - Огляд функціоналу

Щоб отримати відповіді на часто задавані питання, необхідно натиснути кнопку “FAQ” , в результаті користувач отримує відповідну інформацію про часто задавані питання, такі як “Вимоги до вступу” та “Як подати документи”. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».



Write a message...  

Рисунок 3.5 - Огляд функціоналу

Щоб отримати інформацію про вимоги до вступу необхідно натиснути на кнопку “Вимоги до вступу”, і тоді користувач має можливість отримати інформацію як і для магістра так і для бакалавра.

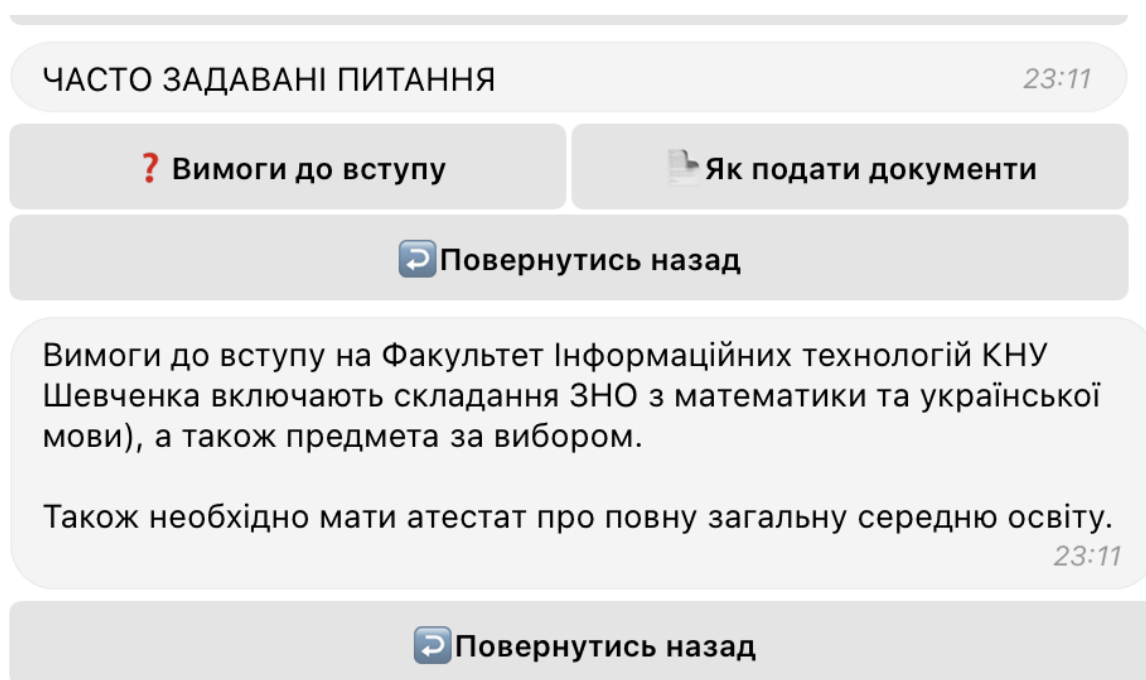


Рисунок 3.6 - Огляд функціоналу

Щоб отримати інформацію про те , як подати документи потрібно натиснути кнопку «Як подати документи», після чого бот надасть відповідну інформацію. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

Документи на вступ приймаються з 1 липня до 15 серпня 2023 року. Для подачі документів необхідно звернутися до приймальної комісії факультету. 23:11



 Повернутись назад

Рисунок 3.7 - Огляд функціоналу

При виборі кнопки “Освітні програми” отримуємо інформацію з посиланням на конкретні освітні програми, що присутні на факультеті. Щоб отримати детальну інформацію потрібно натиснути на відповідні кнопки із програмами, які ведуть до сторінки університетського сайту із актуальною інформацією. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

Факультет Інформаційних технологій КНУ Шевченка пропонує наступні освітні програми: 23:12

Бакалаврська програма з Прикладного програмування 

Магістерська програма з Інформаційних систем 

Магістерська програма з Штучного інтелекту 


 Повернутись назад

Рисунок 3.8 - Огляд функціоналу

При натисканні на кнопки бачимо перехід за посиланням на сайт університету. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

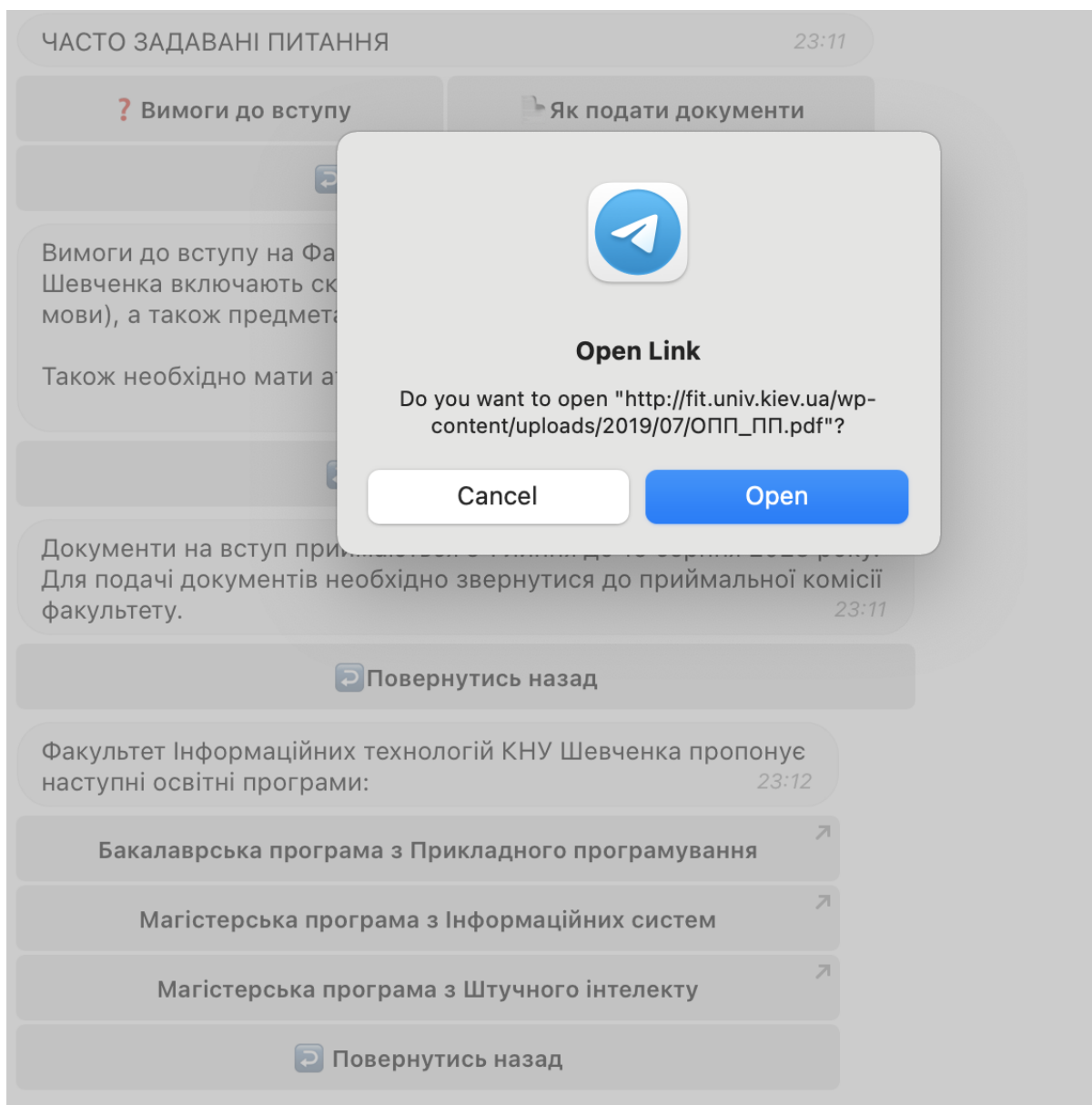


Рисунок 3.9 - Огляд функціоналу

Щоб отримати актуальний розклад, потрібно натиснути кнопку «Розклад», в результаті чого переходимо на сторінку із розкладом для всіх груп. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

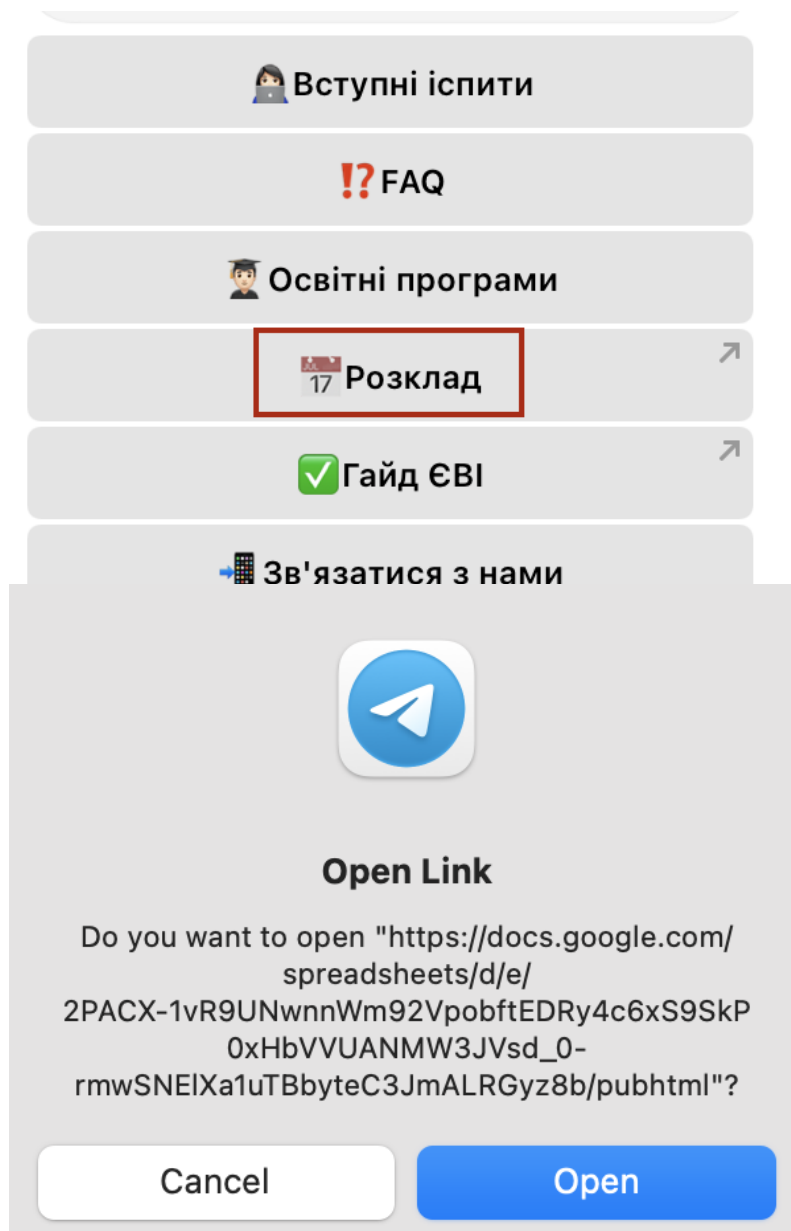


Рисунок 3.10 - Огляд функціоналу

Щоб отримати інформацію про іспит на магістратуру потрібно обрати кнопку “Гайд ЄВІ” , при натисканні на неї ми переходимо на сторінку із гайдом. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

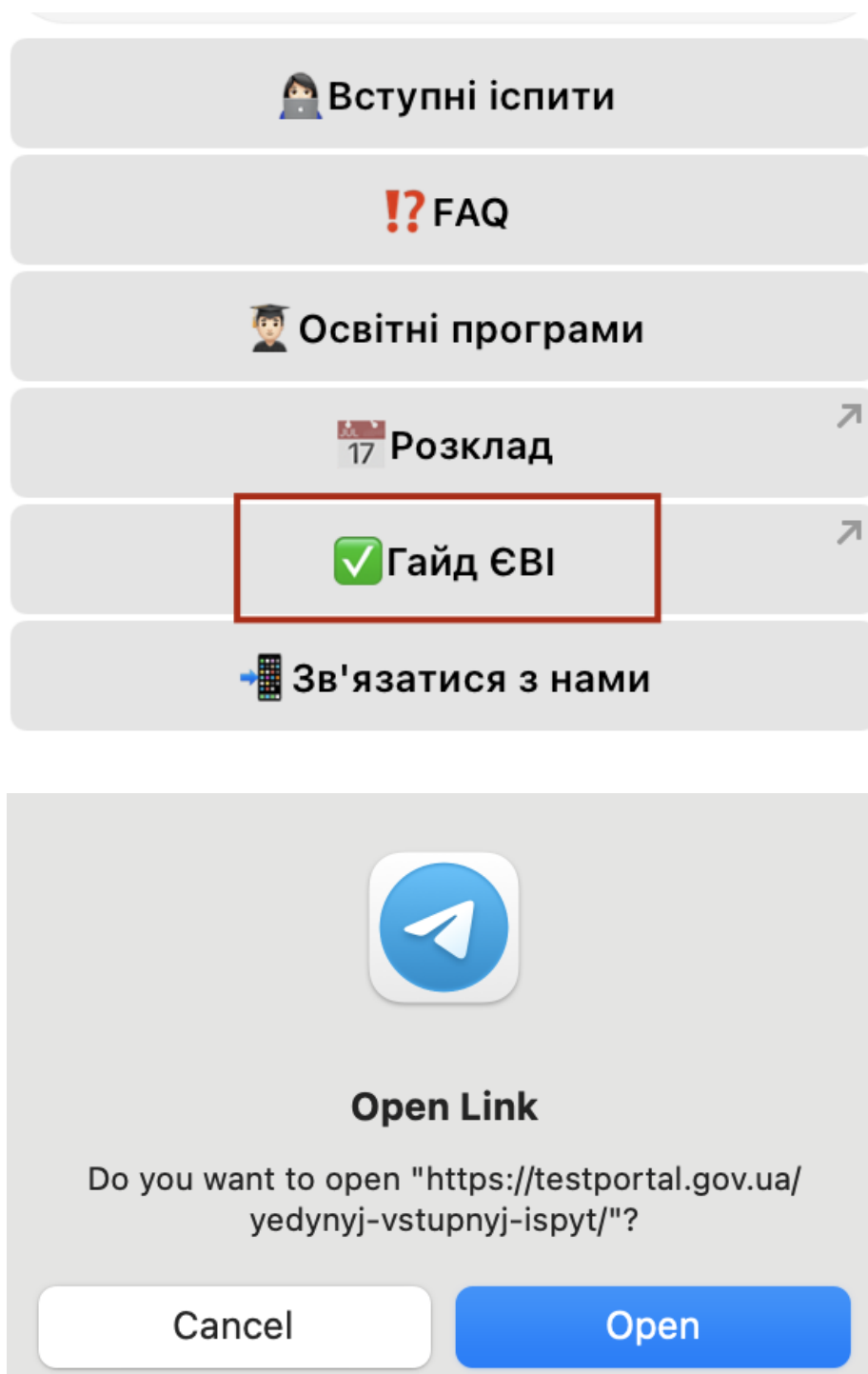


Рисунок 3.11 - Огляд функціоналу

Щоб отримати контактні дані необхідно натиснути на кнопку “Зв'язатися з нами”, у відповідь від бота отримуємо контактну інформацію про університет, та про приймальню деканату. Щоб повернутись до меню потрібно натиснути кнопку «Повернутись назад».

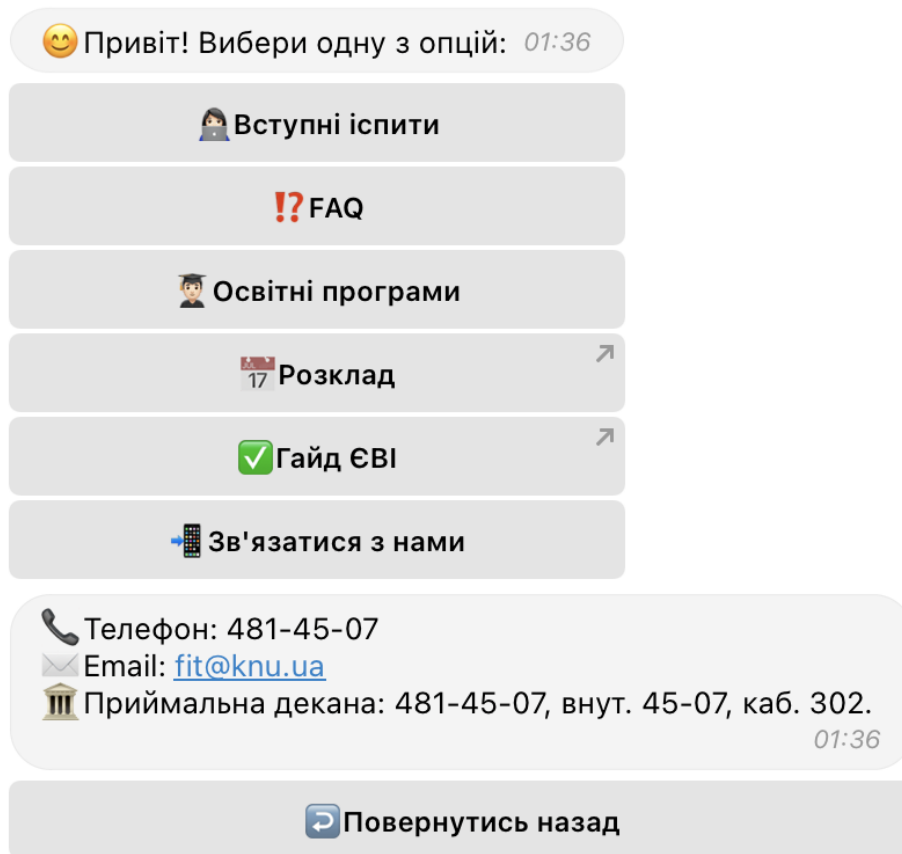


Рисунок 3.12 - Огляд функціоналу

Висновки до розділу 3

В даному розділі було описано використані засоби для реалізації телеграм бота. Загалом Python є оптимальним вибором для розробки чат-ботів Telegram завдяки його простоті, адаптивності та надійній екосистемі інструментів. Було розроблені функціональні та нефункціональні вимоги допоможуть переконатися, що чат-бот для абітурієнтів є надійним, безпечним і зручним для користувача.

Описано задачі розробки чат бота. Його широкий набір функцій і потужні можливості NLP дозволяють розробляти розумні та розважальні чат-боти, які можуть надавати користувачам індивідуальну допомогу та підтримку. Також було описано методи та відображено UML діаграми, такі як діаграма діяльності, використання та активності.

Також у розділі відображено покрокову інструкцію для користувача, реалізований програмний застосунок, виконує поставлену задачу, а отже задовольняє початковим умовам. Бот коректно надає інформацію для навчальних консультацій.

ВИСНОВОК

В наш час, стали популярними месенджери, соціальні мережі, вебсервіси тощо. Це легкий та зручний спосіб взаємодії, а отже виникла потреба впровадження веб-технологій у навчальний процес.

Завданням дипломного проекту було створення чат бота для надання навчальних консультацій.

У розділах розглянуто:

- класифікацію чат ботів;
- аналіз відомих рішень;
- аналіз інструментів розробки;
- розроблено вимоги;
- архітектуру програмного забезпечення;
- опис розробленого бота.

Детальний огляд проекту показує, що розроблена система взаємодії з користувачем є інтуїтивно зрозумілою для нього. Проект оснащений достатньою кількістю функціоналу, щоб в повній мірі, задовольняти поставлені цілі дипломної роботи.

Не дивлячись на об'єм та складність кожної функціональної частини, розроблена система залишається зрозумілою та зручною у використанні. Проект дає змогу спрощення взаємодії з ботом для отримання інформації. Завдяки мобільності, доступу у будь-який час, за межами університету, онлайн.

Бот забезпечує швидкий доступ до інформації, індивідуалізацію відповідей та постійну наявність, сприяючи ефективному та зручному навчанню. Це інноваційний інструмент, який допомагає студентам та абітурієнтам бути краще підготовленими та успішними у своїй освітній дорозі.

Підбиваючи підсумок, можна сказати, що тему роботи «Телеграм-бот для надання навчальних консультацій» розкрито, а мету дослідження досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Клочко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка.
2. A curated directory of chat bot resources & tools : веб-сайт. URL: <http://www.botsfloor.com/botstash/products/?category=Development> %20 (дата звернення: 06.03.2023)
3. Chatbot Report 2018: Current landscape of how people create chatbots and how users expect to interact with them : веб-сайт. URL: <https://elearningindustry.com/chatbots-for-learning-support-10-reasons><https://naiz.chat/NAIZ-report-18072018.pdf> (дата звернення: 10.04.2023)
4. Sheth B. The Bot Lifecycle. What to know before you make your chatbot : веб-сайт. URL: <https://chatbotsmagazine.com/the-bot-lifecycle1ff357430db7>(дата звернення: 15.04.2023)
5. Michiels. E. Modelling Chatbots with a Cognitive System Allows for a Differentiating User Experience : веб-сайт. URL: <http://eurws.org/Vol-2027/paper24.pdf> (дата звернення: 17.04.2023)
6. Telegram Bot API : веб-сайт. URL: <https://core.telegram.org/bots/api> вільний.
7. Що таке функціональні вимоги: приклади, визначення, повний посібник : веб-сайт. URL: <https://visuresolutions.com/uk/blog/functional-requirements/>. (дата звернення: 21.04.2023)
8. Путівник мовою програмування Python : веб-сайт. URL: <https://pythonguide.rozh2sch.org.ua/>. (дата звернення: 23.04.2023)

9. Types of UML Diagrams : веб-сайт. URL: <https://www.lucidchart.com/blog/types-of-UML-diagrams>. (дата звернення: 30.04.2023)

10. D.A. Urayev, Classification and methods for creating chatbot applications, International scientific review (2019) Vol. LXIV, 30-33.

11. A. Vishwakarma, A. Pandey, A Review & Comparative Analysis on Various Chatbots Design, International Journal of Computer Science and Mobile Computing (2021) Vol.10, Issue 2, 72-78. doi: 10.47760/ijcsmc.2021.v10i02.011.

12. Python Crash Course. Eric Matthes, 2015.

13. Effective Python: 59 Specific Ways to Write Better Python. Brett Slatkin, 2015.

14. Learning Python. Mark Lutz, 2000.

15. Beaver L. The Chatbots Explainer : веб-сайт. URL: <https://www.businessinsider.com/intelligence/chatbots-explainer> вільний. (дата звернення: 19.03.2023)

16. Muldowney O. Chatbots. An Introduction And Easy Guide To Making Your Own / O. Muldowney. – Dublin: Curses & Magic, 2017. – 69 p.

17. Sheth B. The Bot Lifecycle. What to know before you make your chatbot : веб-сайт. URL: <https://chatbotmagazine.com/the-bot-lifecycle1ff357430db7> (дата звернення: 22.04.2023)

18. A curated directory of chat bot resources & tools : веб-сайт. URL: [http://www.botsfloor.com/botstash/products/?category=Development %20](http://www.botsfloor.com/botstash/products/?category=Development%20)

19. Сухас У. Oracle Intelligent Bots: Чат-боти з AI : веб-сайт. URL: <https://www.oracle.com/a/ocom/docs/chatbots.pdf> (дата звернення: 03.06.2023)

20. What is a UML Diagram? : веб-сайт. URL:
<https://www.smartdraw.com/uml-diagram/>. (дата звернення: 05.06.2023)

Додатки

Додаток А

```

from telegram import InlineKeyboardButton, InlineKeyboardMarkup
from telegram.ext import Updater, CommandHandler,
CallbackQueryHandler, MessageHandler, Filters

def start(update, context):
    message = "Привіт! Вибери одну з опцій:"
    reply_markup = start_menu()
    update.message.reply_text(message, reply_markup=reply_markup)

def button_click(update, context):
    query = update.callback_query
    option = query.data

    if option == "exams":
        message = "Вступні іспити на денну форму навчання
відбуваються у період з 17.06-20.08.2023"
        buttons = [
            [InlineKeyboardButton("Новини ЗНО",
url="https://osvita.ua/test/")]
        ]
        back_button = [InlineKeyboardButton("Повернутись назад",
callback_data="start")]
        buttons.append(back_button)
        reply_markup = InlineKeyboardMarkup(buttons)
        context.bot.send_message(chat_id=update.effective_chat.id,
text=message, reply_markup=reply_markup)

    elif option == "faq":
        message = "ЧАСТО ЗАДАВАНІ ПИТАННЯ"
        back_button = [InlineKeyboardButton("Повернутись назад",
callback_data="start")]

```

```

        faq_buttons = [
            [InlineKeyboardButton("📄Вимоги до вступу",
callback_data="requirements"),
            InlineKeyboardButton("📄Як подати документи",
callback_data="documents")],
            back_button
        ]
        reply_markup = InlineKeyboardMarkup(faq_buttons)

        context.bot.edit_message_text(
            chat_id=query.message.chat_id,
            message_id=query.message.message_id,
            text=message,
            reply_markup=reply_markup
        )
        return

    elif option == "requirements":
        message = "Вимоги до вступу на Факультет Інформаційних
технологій КНУ Шевченка включають складання ЗНО з математики та
української мови), а також предмета за вибором.\n\nТакож необхідно
мати атестат про повну загальну середню освіту."

        back_button = [InlineKeyboardButton("🏠Повернутись назад",
callback_data="faq")]
        reply_markup = InlineKeyboardMarkup([back_button])
        context.bot.send_message(chat_id=update.effective_chat.id,
text=message, reply_markup=reply_markup)

    elif option == "documents":
        message = "Документи на вступ приймаються з 1 липня до 15
серпня 2023 року. Для подачі документів необхідно звернутися до
приймальної комісії факультету."

```

```

        back_button = [InlineKeyboardButton("⏪ Повернутись назад",
callback_data="faq")]
        reply_markup = InlineKeyboardMarkup([back_button])
        context.bot.send_message(chat_id=update.effective_chat.id,
text=message, reply_markup=reply_markup)

    elif option == "programs":
        message = "Факультет Інформаційних технологій КНУ Шевченка
пропонує наступні освітні програми:"
        buttons = [
            [InlineKeyboardButton("Бакалаврська програма з
Прикладного програмування", url="http://fit.univ.kiev.ua/wp-
content/uploads/2019/07/%D0%9E%D0%9F%D0%9F_%D0%9F%D0%9F.pdf")],
            [InlineKeyboardButton("Магістерська програма з
Інформаційних систем", url="http://fit.univ.kiev.ua/wp-
content/uploads/2019/07/%D0%9E%D0%9D%D0%9F_%D0%86%D0%A11.pdf")],
            [InlineKeyboardButton("Магістерська програма з
Штучного інтелекту", url="http://fit.univ.kiev.ua/wp-
content/uploads/2019/07/%D0%9E%D0%9D%D0%9F_%D0%A88.pdf")]
        ]
        back_button = [InlineKeyboardButton("⏪ Повернутись назад",
callback_data="start")]
        reply_markup = InlineKeyboardMarkup([back_button])
        context.bot.send_message(chat_id=update.effective_chat.id,
text=message, reply_markup=reply_markup)

    elif option == "contact":
        message = "Зв'язатися з нами можна за номером
+380123456789 або за електронною адресою example@example.com"
        back_button = [InlineKeyboardButton("⏪ Повернутись назад",
callback_data="start")]
        reply_markup = InlineKeyboardMarkup([back_button])
        context.bot.send_message(chat_id=update.effective_chat.id,
text=message, reply_markup=reply_markup)

```

```

elif option == "start":
    message = "Привіт! Вибери одну з опцій:"
    reply_markup = start_menu()
    context.bot.edit_message_text(
        chat_id=query.message.chat_id,
        message_id=query.message.message_id,
        text=message,
        reply_markup=reply_markup
    )
    return

query.answer()

def start_menu():
    buttons = [
        [InlineKeyboardButton("Вступні іспити",
callback_data="exams")],
        [InlineKeyboardButton("FAQ", callback_data="faq")],
        [InlineKeyboardButton("Освітні програми",
callback_data="programs")],
        [InlineKeyboardButton("Розклад",
url="https://docs.google.com/spreadsheets/d/e/2PACX-
1vR9UNwnnWm92VpobftEDRy4c6xS9SkP0xHbVVUANMW3JVsd_0-
rmwSNElXa1uTBbyteC3JmALRGyz8b/pubhtml")],
        [InlineKeyboardButton("Гайд ЄВІ",
url="https://testportal.gov.ua/yedynuj-vstupnyj-ispyt/")],
        [InlineKeyboardButton("Зв'язатися з нами",
callback_data="contact")]
    ]
    reply_markup = InlineKeyboardMarkup(buttons)
    return reply_markup

```

```
def main():
    updater = Updater(token="5618189984:AAFpLuqXS1gG1EsSsPgWpw12qQA45orzshA",
use_context=True)
    dp = updater.dispatcher

    dp.add_handler(CommandHandler("start", start))
    dp.add_handler(CallbackQueryHandler(button_click))

    updater.start_polling()
    updater.idle()

if __name__ == '__main__':
    main()
```