

Харківський національний університет міського господарства
імені О.М. Бекетова

ННІ енергетичної, інформаційної та транспортної інфраструктури
кафедра управління проєктами в міському господарстві і будівництві

Пояснювальна записка до кваліфікаційної роботи

на тему

**УПРАВЛІННЯ ПРОЄКТОМ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО
ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ**

Виконав: студент 4 курсу групи гр. УП(кн) 2022-1
Спеціальність 122 Комп'ютерні науки
Освітня програма «Комп'ютерні науки. Управління
проєктами»

Будніченко Максим Юрійович 
(ПІБ повністю, підпис)

Керівник Гусєва Ю.Ю. 
(прізвище та ініціали, підпис)

Рецензент Косенко В.В. 
(прізвище та ініціали)

Харків – 2026 року

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО
ГОСПОДАРСТВА імені О.М. БЕКЕТОВА

Факультет ННІ енергетичної, інформаційної та транспортної інфраструктури

Кафедра управління проєктами в міському господарстві і будівництві

Освітньо-кваліфікаційний рівень бакалавр

Освітня програма Комп'ютерні науки. Управління проєктами

ЗАТВЕРДЖУЮ

Завідувач кафедри УПМГБ

д.т.н., проф. Чумаченко І.В.

«16» червня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
студенту

Будніченку Максиму Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Управління проєктом створення інтелектуального чатбота підтримки вступників університету

керівник проєкту Гусєва Юлія Юріївна, д-р. техн. наук, доцент.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від № 440-03 від 22.05.2026 р.

2. Строк подання студентом проєкту 16.06.2026 р.

3. Вихідні дані до роботи: матеріали переддипломної практики, інформація періодичних видань, книг, монографій, вихідні дані про проєкт, web-джерела.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Теоретичні основи управління проєктами розробки AI-орієнтованих інформаційних систем для закладів вищої освіти 2. Дослідження та проєктування gpt-чатбота підтримки вступників університету 3. Управління проєктом розробки gpt-чатбота підтримки вступників університету.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація в форматі MS Power Point. Обов'язкові елементи – діаграма Ганта, сітловий графік.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Косенко Н.В., доцент каф. УПМГБ		

7. Дата видачі завдання 12.05.2026 р.

КАЛЕНДАРНИЙ ПЛАН

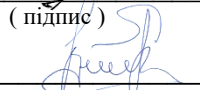
№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Розробка 1го розділу кваліфікаційної роботи	23.05.2026	виконано
2	Розробка 2го розділу кваліфікаційної роботи	30.05.2026	виконано
3	Розробка 3го розділу кваліфікаційної роботи	16.06.2025	виконано
5	Нормоконтроль	16-18.06.2026	виконано
6	Попередній захист кваліфікаційної роботи	16.06.2026	виконано
7	Рецензування кваліфікаційної роботи	16-18.06.2026	виконано
8	Захист на ДЕК	24.06.2026	

Керівник роботи


(підпис)

Гусєва Ю.Ю.
(прізвище та ініціали)

Студент


(підпис)

Будніченко М.Ю.
(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра робота: 77 с., 3 розділи, 6 рис., 17 табл., 18 бібліографічних джерел.

Метою роботи є управління проектом розробки GPT-чатбота підтримки вступників університету на основі сучасних технологій штучного інтелекту та гнучких методологій управління IT-проектами.

Об'єкт дослідження – процес інформаційної підтримки вступників закладу вищої освіти.

Предмет дослідження – методи, моделі та інструменти управління проектом розробки GPT-чатбота підтримки вступників університету.

У межах роботи сформовано проект створення GPT-чатбота підтримки вступників університету, який передбачає використання Telegram Bot API, OpenAI API, технології Retrieval-Augmented Generation (RAG), бази знань університету та сучасних засобів серверної розробки. Розроблено структуру робіт проекту, виконано планування команди, ресурсів, розкладу та бюджету проекту, проведено аналіз ризиків та організовано управління проектом на основі Agile/Scrum.

Практичне значення отриманих результатів полягає у можливості використання запропонованих проектних рішень для створення інтелектуальної системи підтримки вступників, що дозволить автоматизувати консультаційне обслуговування, підвищити якість інформаційної підтримки та знизити навантаження на працівників приймальної комісії.

GPT-ЧАТБОТ, ШТУЧНИЙ ІНТЕЛЕКТ, ВСТУПНА КАМПАНІЯ, ЗАКЛАД ВИЩОЇ ОСВІТИ, TELEGRAM-БОТ, OPENAI API, RAG, УПРАВЛІННЯ ПРОЄКТОМ, SCRUM, ЦИФРОВІЗАЦІЯ ОСВІТИ.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЄКТАМИ РОЗРОБКИ AI-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ	9
1.1 Особливості цифровізації закладів вищої освіти в сучасних умовах	9
1.2 Аналіз сучасних інформаційних систем підтримки вступників	16
1.3 Аналіз технологій штучного інтелекту та GPT-моделей у чатботах	20
1.4 Аналіз підходів та методологій управління IT-проєктами	24
1.6 Висновки до розділу 1	29
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТА ПРОЄКТУВАННЯ GPT-ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ	31
2.1 Постановка задачі розробки GPT-чатбота підтримки вступників	31
2.2 Аналіз функціональних та нефункціональних вимог до системи	34
2.3 Аналіз та вибір технологій реалізації системи	39
2.4 Моделювання функціонування системи	44
2.5 Проєктування архітектури GPT-чатбота	48
2.6 Висновки до розділу 2	54
РОЗДІЛ 3 УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ GPT-ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ	56
3.1 Ініціація та характеристика проєкту	56
3.2 Формування структури робіт проєкту	60
3.3 Планування команди та ресурсів проєкту	63
3.4 Планування розкладу проєкту	67
3.5 Планування вартості проєкту	69
3.6 Управління ризиками проєкту	72
3.7 Організація Agile/Scrum управління проєктом	73
3.8 Висновки до розділу 3	75

	6
ВИСНОВКИ.....	77
БІБЛІОГРАФІЧНИЙ СПИСОК.....	78
Додаток А Структура робіт проєкту GPT-чатбота	80
Додаток Б Діаграма Ганта.....	81
Додаток В Сітьовий графік.....	82
Додаток Г Графічні матеріали	83

ВСТУП

Сучасний розвиток інформаційних технологій та активна цифровізація закладів вищої освіти зумовлюють необхідність впровадження нових інструментів взаємодії з абітурієнтами. Вступна кампанія є одним із найбільш відповідальних процесів діяльності університету, під час якого вступники потребують оперативного доступу до актуальної інформації щодо спеціальностей, освітніх програм, правил прийому, термінів подання документів та інших аспектів вступу. Традиційні засоби консультування, такі як вебсайти, телефонні консультації та електронна пошта, не завжди забезпечують необхідний рівень оперативності та персоналізації взаємодії.

Одним із перспективних напрямів вирішення зазначеної проблеми є використання технологій штучного інтелекту та великих мовних моделей для створення інтелектуальних чатботів. GPT-чатботи дозволяють забезпечити природну взаємодію з користувачами, автоматизувати консультаційну підтримку та надавати відповіді в режимі реального часу. Особливо актуальним є використання таких рішень у закладах вищої освіти, які прагнуть підвищити якість цифрових сервісів та ефективність комунікації зі вступниками.

Метою кваліфікаційної роботи є управління проектом розробки GPT-чатбота підтримки вступників університету на основі сучасних технологій штучного інтелекту та гнучких методологій управління IT-проектами.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати особливості цифровізації закладів вищої освіти;
- дослідити сучасні інформаційні системи підтримки вступників;
- проаналізувати технології штучного інтелекту та GPT-моделі для створення чатботів;
- дослідити сучасні підходи до управління IT-проектами;
- сформулювати функціональні та нефункціональні вимоги до системи;
- розробити архітектуру GPT-чатбота;
- спроектувати функціонування інформаційної системи;

- виконати планування команди, ресурсів, розкладу та бюджету проєкту;
- розробити систему управління ризиками та організацію Agile/Scrum управління проєктом.

Об'єктом дослідження є процес інформаційної підтримки вступників закладу вищої освіти.

Предметом дослідження є методи, моделі та інструменти управління проєктом розробки GPT-чатбота підтримки вступників університету.

У роботі використано методи системного аналізу, методи проєктування інформаційних систем, а також Agile/Scrum-підхід до організації процесу розробки програмного забезпечення.

Практичне значення роботи полягає у можливості використання запропонованих проєктних рішень під час створення інтелектуальної системи підтримки вступників, що дозволить підвищити якість консультаційного супроводу та знизити навантаження на працівників приймальної комісії.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ УПРАВЛІННЯ ПРОЄКТАМИ РОЗРОБКИ AI-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

1.1 Особливості цифровізації закладів вищої освіти в сучасних умовах

Сучасний етап розвитку суспільства характеризується стрімким поширенням інформаційно-комунікаційних технологій у всіх сферах людської діяльності. Глобальна цифровізація, яку дослідники все частіше називають «четвертою промисловою революцією», кардинально змінює не лише виробничі процеси, а й підходи до освіти, науки та управління. Заклади вищої освіти (ЗВО), будучи одночасно об'єктами і суб'єктами цих змін, опиняються перед необхідністю системної трансформації власної діяльності відповідно до вимог цифрової епохи [1].

Поняття «цифровізація» (англ. digitalization) у контексті вищої освіти дослідники трактують по-різному. У широкому сенсі воно охоплює будь-яке застосування цифрових технологій в освітньому процесі. У вузькому розумінні – передбачає цілеспрямовану трансформацію бізнес-моделей, організаційних структур і процесів університету на основі цифрових технологій з метою підвищення ефективності та якості послуг [2]. Слід відрізняти цифровізацію від суміжних понять: «оцифрування» (digitization) означає переведення аналогової інформації у цифровий формат, тоді як «цифрова трансформація» (digital transformation) є найширшим поняттям, що передбачає докорінну зміну операційних моделей і ціннісних пропозицій організації завдяки цифровим технологіям [3].

Теоретичне підґрунтя дослідження цифровізації освіти закладене у роботах низки провідних науковців. Зокрема, К. Шваб у праці «Четверта промислова революція» обґрунтовує необхідність переосмислення змісту освіти та компетентностей, які мають формувати університети в умовах масштабного впровадження штучного інтелекту, робототехніки та Інтернету речей [1]. Автори

[4] вказують на те, що цифрова трансформація ЗВО – це не просто технологічне оновлення, а зміна парадигми освіти, де центральною фігурою стає студент із його індивідуальними потребами та траєкторією навчання. Вітчизняні дослідники В. Биков та М. Лещенко розглядають цифровізацію освіти як процес побудови відкритого освітнього середовища на основі хмарних технологій, що забезпечує рівний доступ до якісних освітніх ресурсів [5].

Нормативно-правове забезпечення цифровізації вищої освіти в Україні.

Процес цифровізації вищої освіти в Україні регулюється системою нормативно-правових актів різних рівнів. Основоположним документом є Закон України «Про вищу освіту» (2014, зі змінами), який визначає принципи автономії університетів та їхню відповідальність за якість освітніх послуг, що безпосередньо пов'язано з необхідністю впровадження сучасних цифрових інструментів [6]. Закон України «Про освіту» (2017) закріплює право здобувачів освіти на доступ до освітніх, інформаційних та комунікаційних технологій і визнає дистанційну форму навчання однією з основних [7].

Стратегічним документом у сфері цифровізації є Концепція розвитку цифрової економіки та суспільства України на 2018–2020 роки, схвалена розпорядженням Кабінету Міністрів України від 17 січня 2018 р. № 67-р., яка визначила пріоритети цифрової трансформації різних секторів, включно з освітою [8]. Подальший розвиток цього напрямку знайшов відображення в Указі Президента України «Про цілі сталого розвитку України на період до 2030 року» (2019), де підвищення якості та доступності освіти через цифровізацію визначено як один із пріоритетів [9].

Важливим орієнтиром для розвитку вітчизняної вищої освіти є також «Стратегія розвитку вищої освіти в Україні на 2022–2032 роки», яка ставить за мету побудову сучасного університетського середовища, здатного адаптуватися до викликів цифрової економіки та забезпечувати підготовку конкурентоспроможних фахівців [10]. Документ акцентує увагу на необхідності формування цифрових екосистем університетів, розвитку е-інфраструктури,

впровадження адаптивних систем навчання та інтелектуальних сервісів підтримки студентів.

На рівні Євросоюзу, до стандартів якого прагне наблизитись Україна у рамках євроінтеграційного курсу, ключовим орієнтиром є «Цифровий план дій у галузі освіти» (Digital Education Action Plan 2021–2027), прийнятий Європейською комісією. Документ визначає два стратегічні пріоритети: сприяння розвитку цифрової освітньої екосистеми та підвищення цифрових компетентностей і навичок для цифрової трансформації [11].

Рівні та моделі цифрової трансформації університетів.

Дослідники пропонують різні моделі для опису ступеня цифрової зрілості університетів. Одна з найпоширеніших – «Модель цифрової зрілості університету» (University Digital Maturity Model) – виділяє п'ять рівнів: початковий, розвиваючий, визначений, оптимізований та інноваційний [12]. Більшість українських університетів наразі знаходяться на другому-третьому рівнях, тоді як провідні світові ЗВО досягли четвертого-п'ятого.

З практичної точки зору, цифровізацію ЗВО можна структурувати за такими функціональними напрямками:

1. Цифровізація навчального процесу. Впровадження систем управління навчанням (LMS – Learning Management System) є першим і найбільш поширеним кроком. Найпопулярніші платформи, зокрема Moodle, забезпечують організацію дистанційного та змішаного навчання, управління навчальними матеріалами, контроль успішності та комунікацію між викладачами і студентами.

Паралельно розвиваються технології адаптивного навчання (Adaptive Learning), що використовують алгоритми машинного навчання для персоналізації освітнього контенту відповідно до індивідуального прогресу та стилю навчання кожного студента.

2. Цифровізація управлінських процесів. Університетські ERP-системи (Enterprise Resource Planning) інтегрують фінансові, кадрові, академічні та адміністративні процеси в єдиний інформаційний простір. Впровадження таких

систем дозволяє суттєво скоротити паперовий документообіг, підвищити прозорість і швидкість адміністративних процесів.

3. Цифровізація бібліотечних сервісів. Електронні бібліотеки, системи управління бібліографічними даними та доступ до міжнародних баз даних (Web of Science, Scopus, IEEE Xplore) стали невід'ємним елементом цифрової інфраструктури сучасного університету. Відкритий доступ (Open Access) до наукових публікацій поступово стає стандартом, що відповідає принципам Будапештської ініціативи відкритого доступу.

4. Цифровізація комунікацій. Університети активно використовують офіційні сайти, мобільні застосунки, соціальні мережі, месенджери та електронну пошту для взаємодії зі студентами, абітурієнтами, випускниками та партнерами. Дедалі більшого поширення набувають інтелектуальні системи підтримки – чат-боти та віртуальні асистенти на основі технологій обробки природної мови (Natural Language Processing, NLP).

5. Аналітика даних і прийняття рішень. Аналітика навчальних даних (Learning Analytics) дозволяє університетам отримувати дієві інсайти для вдосконалення навчального процесу, завчасного виявлення студентів, що перебувають у групі ризику відрахування, та оптимізації розподілу ресурсів. Концепція «університету, керованого даними» (data-driven university) набуває все більшого визнання як ефективна модель управління ЗВО.

Роль штучного інтелекту в цифровій трансформації університетів.

Особливе місце серед технологій, що визначають сучасний вектор цифровізації ЗВО, займає штучний інтелект (ШІ). За визначенням ЮНЕСКО, штучний інтелект в освіті охоплює широкий спектр застосувань – від інтелектуальних навчальних систем та автоматизованого оцінювання до аналітики даних та розмовних агентів.

Великі мовні моделі (Large Language Models, LLM) – зокрема GPT-4 компанії OpenAI, Claude від Anthropic, Gemini від Google – відкривають принципово нові можливості для автоматизації комунікативних процесів у ЗВО. На відміну від традиційних чат-ботів, що діяли за наперед визначеними

сценаріями (rule-based chatbots), GPT-орієнтовані системи здатні генерувати зв'язні, контекстуально релевантні відповіді практично на будь-яке запитання природною мовою. Це робить їх особливо цінними для застосування у сферах, де потрібна обробка великого обсягу різноманітних запитів – зокрема, в системах інформаційної підтримки абітурієнтів.

Цифровізація процесу вступу до університету: стан і тенденції.

Процес вступу до закладу вищої освіти є одним із найбільш інформаційно насичених етапів взаємодії університету із зовнішнім середовищем. Абітурієнти стикаються з необхідністю обробки великого масиву інформації: умови прийому та прохідні бали, перелік спеціальностей і освітніх програм, вимоги до документів, строки подання заяв, порядок зарахування, можливості отримання гуртожитку та стипендій тощо. Дослідження компанії Salesforce (2021) «Connected Student Report» показало, що 68% абітурієнтів відчують труднощі з пошуком актуальної та зрозумілої інформації на сайтах університетів, а 54% повідомили, що отримували суперечливу інформацію з різних офіційних джерел [13].

Традиційна модель інформаційного забезпечення вступної кампанії передбачала особистий прийом абітурієнтів у приймальній комісії, телефонні консультації та електронне листування. Ця модель є ресурсоемною: під час пікових навантажень (червень–серпень) приймальні комісії великих університетів отримують тисячі однотипних запитань щодня. Аналіз практики провідних університетів України свідчить про те, що до 70–80% таких запитань стосуються стандартних тем і потенційно можуть бути оброблені автоматично.

Цифровізація вступного процесу в Україні розпочалася з впровадженням системи «Вступ.ОСВІТА.UA» – єдиної електронної бази даних, що забезпечує подання вступниками заяв в електронній формі та моніторинг рейтингових списків у режимі реального часу. Ця система суттєво підвищила прозорість і зручність вступної кампанії, проте не вирішила проблему персоналізованої інформаційної підтримки абітурієнтів [14].

Наступним кроком стало впровадження університетами власних цифрових каналів комунікації з абітурієнтами – сторінок у соціальних мережах (Facebook, Instagram, Telegram-каналів), де публікуються актуальні новини та відповіді на поширені запитання. Однак соціальні мережі мають суттєві обмеження: неструктурованість інформації, необхідність постійного модерування, відсутність можливості вести персоналізований діалог у масштабі.

Незважаючи на очевидні переваги, цифровізація ЗВО стикається з низкою системних викликів і бар'єрів, які необхідно враховувати під час планування та реалізації відповідних проєктів.

Технологічні виклики включають необхідність забезпечення надійності та безперервності роботи цифрових систем, захисту персональних даних користувачів відповідно до вимог Загального регламенту про захист даних (GDPR) та вітчизняного законодавства, а також забезпечення сумісності (інтероперабельності) різних програмних рішень у єдиній інформаційній екосистемі університету.

Організаційні бар'єри пов'язані з опором змінам з боку частини викладацького та адміністративного персоналу, недостатнім рівнем цифрових компетентностей співробітників, відсутністю чіткої стратегії цифрової трансформації та недостатньою координацією між підрозділами університету.

Фінансові обмеження є особливо актуальними для більшості українських університетів. Впровадження та підтримка сучасних цифрових систем вимагають значних інвестицій – як у технологічну інфраструктуру, так і в навчання персоналу та управління змінами. Пошук оптимального співвідношення між вартістю рішення та його функціональністю є одним із ключових завдань проєктного менеджменту у сфері цифровізації ЗВО.

Специфічні виклики воєнного часу. В умовах повномасштабного вторгнення росії в Україну (з 24 лютого 2022 р.) вітчизняні університети зіштовхнулися з безпрецедентними викликами. Внутрішнє переміщення студентів та персоналу, руйнування або пошкодження інфраструктури, необхідність забезпечення освітнього процесу для студентів за кордоном – усе це

різко підвищило попит на надійні та гнучкі цифрові рішення. Разом з тим воєнний час став каталізатором прискорення цифрової трансформації: університети, що раніше відкладали впровадження дистанційних технологій, були змушені зробити це в надкороткі терміни [15].

Аналіз актуальних трендів у сфері EdTech дозволяє виділити кілька напрямів, що визначатимуть вектор цифровізації університетів у найближчі роки. По-перше, генеративний штучний інтелект (Generative AI) – і зокрема великі мовні моделі класу GPT – стрімко виходить за межі експериментального використання та стає інструментом масового застосування в освіті: для написання навчальних матеріалів, автоматичного зворотного зв'язку, персоналізованого тьюторингу та інформаційної підтримки. По-друге, мікрокреденціали та модульні освітні програми набувають популярності як гнучка альтернатива традиційним ступеневим програмам, що потребує відповідної цифрової інфраструктури для управління мікроакредитацією. По-третє, концепція «цифрового двійника університету» (Digital Twin of a University) передбачає створення детальної цифрової моделі закладу освіти для симуляції та оптимізації всіх процесів – від навчальних до господарських.

Зважаючи на вищевикладене, можна зазначити, що цифровізація закладів вищої освіти є незворотним і багатовимірним процесом, що охоплює технологічну, організаційну, правову та культурну складові. Ефективна цифрова трансформація університету потребує системного підходу, стратегічного планування та компетентного управління проєктами впровадження цифрових рішень. Одним із пріоритетних напрямів цієї трансформації є автоматизація інформаційної підтримки абітурієнтів на основі GPT-технологій, що дозволяє одночасно підвищити якість сервісу, знизити операційне навантаження на персонал та забезпечити рівний доступ до інформації для всіх потенційних студентів незалежно від їх місцезнаходження.

1.2 Аналіз сучасних інформаційних систем підтримки вступників

У сучасних умовах цифрової трансформації закладів вищої освіти важливого значення набуває впровадження інформаційних систем, спрямованих на підтримку вступників під час вступної кампанії. Значне зростання кількості онлайн-комунікацій між університетами та абітурієнтами зумовлює необхідність використання сучасних цифрових сервісів, здатних забезпечити швидкий доступ до інформації, автоматизацію консультаційної підтримки та підвищення ефективності взаємодії з користувачами.

Інформаційні системи підтримки вступників являють собою комплекс програмних, веборієнтованих та комунікаційних засобів, які забезпечують надання довідкової інформації, автоматизацію консультацій, супровід процесу подання документів та підтримку комунікації між вступником і закладом освіти. Основною метою таких систем є спрощення процесу вступу та підвищення доступності інформації для користувачів.

На сьогоднішній день більшість закладів вищої освіти використовують різні цифрові інструменти для підтримки вступників. До найбільш поширених належать:

- офіційні вебсайти університетів;
- FAQ-системи;
- електронні кабінети вступників;
- чатботи;
- Telegram-боти;
- системи онлайн-консультування;
- AI-асистенти.

Найбільш базовим інструментом інформаційної підтримки вступників є офіційний вебсайт університету. Саме вебсайт виступає основним джерелом інформації про освітні програми, умови вступу, перелік необхідних документів, строки подачі заяв, вартість навчання та особливості вступної кампанії. Однак традиційні вебресурси мають низку недоліків. Значний обсяг інформації

ускладнює її пошук для користувача, а структура сайтів не завжди є інтуїтивно зрозумілою. Крім того, статичний формат подання інформації не забезпечує персоналізованої взаємодії з користувачем.

Для спрощення доступу до інформації багато університетів впроваджують FAQ-системи (Frequently Asked Questions), які містять перелік найбільш поширених запитань і відповідей. Такі системи дозволяють зменшити навантаження на працівників приймальної комісії та забезпечують швидкий доступ до типової інформації. Водночас FAQ-системи мають обмежений рівень інтерактивності та не можуть враховувати індивідуальні особливості запитів користувачів.

Важливим етапом розвитку цифрових сервісів стала поява чатботів. Чатботи являють собою програмні системи, здатні автоматизовано взаємодіяти з користувачами через текстові повідомлення. У сфері освіти чатботи активно використовуються для надання консультацій вступникам, відповіді на поширені питання та підтримки комунікації через месенджери або вебінтерфейси.

Одними з найбільш популярних платформ для реалізації чатботів є Telegram, Facebook Messenger та вебчат-системи. Telegram-боти набули значного поширення завдяки простоті використання, доступності та високій популярності месенджера серед молоді. Багато університетів використовують Telegram-ботів для автоматизації консультаційної підтримки вступників.

Наприклад, окремі заклади вищої освіти України реалізують Telegram-ботів для інформування вступників про:

- строки вступної кампанії;
- перелік спеціальностей;
- вартість навчання;
- прохідні бали;
- необхідні документи;
- контакти приймальної комісії.

Основною перевагою чатботів є можливість забезпечення швидкої взаємодії з користувачами в режимі реального часу. Крім того, чатботи можуть

працювати цілодобово, що дозволяє вступникам отримувати інформацію незалежно від часу доби. Це особливо актуально під час активної фази вступної кампанії, коли кількість звернень значно збільшується.

Водночас традиційні чатботи, побудовані на основі сценарного підходу, мають суттєві обмеження. Їхня робота базується на заздалегідь визначених правилах та шаблонах відповідей. У разі, якщо запит користувача не відповідає передбаченому сценарію, система може надати некоректну відповідь або взагалі не обробити звернення. Це знижує ефективність використання таких систем та негативно впливає на користувацький досвід.

Суттєвим етапом розвитку інформаційних систем підтримки вступників стало впровадження технологій штучного інтелекту та обробки природної мови (Natural Language Processing, NLP). AI-чатботи дозволяють аналізувати зміст текстових повідомлень, визначати наміри користувача та формувати більш природні відповіді. Використання технологій штучного інтелекту значно покращує якість взаємодії між користувачем та інформаційною системою.

Особливого поширення набули GPT-моделі компанії OpenAI, які демонструють високий рівень ефективності у задачах генерації тексту та підтримки діалогу природною мовою. GPT-чатботи здатні підтримувати контекст розмови, аналізувати складні запити та генерувати більш гнучкі й інформативні відповіді порівняно з класичними сценарними системами.

Використання GPT-технологій у сфері освіти відкриває нові можливості для автоматизації підтримки вступників. Такі системи можуть:

- консультувати щодо умов вступу;
- рекомендувати освітні програми;
- пояснювати правила подачі документів;
- допомагати у виборі спеціальності;
- надавати інформацію щодо вартості навчання;
- підтримувати багатомовну взаємодію.

На відміну від традиційних FAQ-систем, GPT-чатботи забезпечують більш природний формат спілкування, що позитивно впливає на рівень задоволеності

користувачів. Крім того, такі системи здатні адаптувати відповіді до контексту діалогу та уточнювати інформацію у разі необхідності.

Порівняльний аналіз основних інформаційних систем підтримки вступників наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз основних інформаційних систем підтримки вступників

Тип системи	Переваги	Недоліки
Офіційний вебсайт	Повний обсяг інформації	Складність пошуку інформації
FAQ-система	Простота реалізації	Відсутність інтерактивності
Telegram-бот	Швидка комунікація	Обмежена логіка відповідей
Сценарний чатбот	Автоматизація типових запитів	Низька гнучкість
GPT-чатбот	Природна мова, адаптивність, персоналізація	Залежність від AI-моделі та актуальності даних

Важливим аспектом сучасних інформаційних систем підтримки вступників є інтеграція з іншими цифровими сервісами університету. Сучасні чатботи можуть взаємодіяти з базами даних, електронними кабінетами вступників, CRM-системами та вебресурсами університету. Це дозволяє автоматизувати значну частину інформаційної підтримки та забезпечити актуальність даних.

Окрему увагу слід приділити питанням користувацького досвіду (User Experience, UX). Інформаційна система підтримки вступників повинна бути інтуїтивно зрозумілою, швидкою та зручною у використанні. Молодь переважно використовує мобільні пристрої та месенджери, тому інтеграція чатботів із популярними платформами є важливим чинником ефективності таких систем.

Ще одним важливим напрямом розвитку інформаційних систем підтримки вступників є персоналізація. Сучасні AI-системи можуть враховувати інтереси користувача, історію попередніх звернень та індивідуальні потреби вступника. Це дозволяє формувати персоналізовані рекомендації та покращувати якість консультаційної підтримки.

Попри значні переваги AI-орієнтованих систем, їх впровадження супроводжується певними труднощами. Однією з основних проблем є

забезпечення актуальності інформації. Правила вступу, перелік спеціальностей та умови прийому можуть змінюватися щороку, тому база знань чатбота повинна регулярно оновлюватися. Крім того, важливими є питання інформаційної безпеки, захисту персональних даних та контролю якості відповідей AI-систем.

Таким чином, сучасні інформаційні системи підтримки вступників є важливим елементом цифрової трансформації закладів вищої освіти. Найбільш перспективним напрямом розвитку таких систем є використання технологій штучного інтелекту та GPT-моделей, які дозволяють забезпечити більш ефективну, персоналізовану та інтерактивну взаємодію між університетом і вступниками. Використання GPT-чатботів дозволяє автоматизувати консультаційну підтримку, зменшити навантаження на приймальні комісії та підвищити якість цифрових сервісів університету.

1.3 Аналіз технологій штучного інтелекту та GPT-моделей у чатботах

Одним із найбільш перспективних напрямів використання штучного інтелекту є створення інтелектуальних чатботів, здатних здійснювати автоматизовану взаємодію з користувачами природною мовою.

Штучний інтелект (Artificial Intelligence, AI) являє собою сукупність методів, алгоритмів і програмних засобів, які забезпечують можливість моделювання інтелектуальної діяльності людини. До основних напрямів розвитку AI належать машинне навчання, нейронні мережі, комп'ютерний зір, аналіз даних та обробка природної мови (Natural Language Processing, NLP).

У контексті створення чатботів найбільше значення мають технології NLP, які забезпечують можливість аналізу, інтерпретації та генерації текстових повідомлень природною мовою. Основною метою NLP є забезпечення ефективної взаємодії між людиною та комп'ютерною системою без використання спеціальних команд або формалізованих мов.

Сучасні системи NLP використовують алгоритми машинного навчання та нейронні мережі для аналізу текстових даних. Це дозволяє системам:

- визначати зміст повідомлення;
- аналізувати наміри користувача;
- виділяти ключові сутності;
- підтримувати контекст діалогу;
- формувати відповіді природною мовою.

Одним із найбільш важливих етапів розвитку технологій NLP стало створення великих мовних моделей (Large Language Models, LLM). Великі мовні моделі являють собою нейронні мережі, навчені на значних обсягах текстових даних. Основним завданням таких моделей є прогнозування наступного слова або фрагмента тексту на основі контексту попередніх повідомлень.

Серед найбільш відомих великих мовних моделей особливе місце займають GPT-моделі (Generative Pre-trained Transformer), розроблені компанією OpenAI. GPT-моделі базуються на архітектурі Transformer, запропонованій у 2017 році. Основною перевагою архітектури Transformer є можливість ефективної обробки великих обсягів тексту та врахування контекстних зв'язків між словами.

Принцип роботи GPT-моделей ґрунтується на попередньому навчанні (pre-training) на великих текстових корпусах та подальшому налаштуванні для виконання конкретних задач. У процесі навчання модель аналізує мільйони текстових прикладів, вивчаючи закономірності природної мови, структуру речень та семантичні зв'язки між словами.

GPT-моделі демонструють високий рівень ефективності у задачах:

- генерації тексту;
- підтримки діалогу;
- автоматизованого консультування;
- перекладу;
- класифікації текстів;
- аналізу інформації;
- пошуку відповідей на запитання.

Важливою особливістю GPT-моделей є здатність підтримувати контекст діалогу. Це дозволяє чатботу враховувати попередні повідомлення користувача

та формувати більш логічні й природні відповіді. На відміну від класичних сценарних чатботів, GPT-системи не обмежуються заздалегідь визначеними правилами та можуть працювати з великою кількістю варіантів формулювання запитів.

Традиційні чатботи переважно будуються на основі сценарного підходу. Їхня робота базується на використанні шаблонів, ключових слів та жорстко визначених правил обробки повідомлень. Такі системи можуть ефективно працювати лише в межах обмеженого набору сценаріїв. У разі нестандартного формулювання запиту система часто не здатна надати коректну відповідь.

На відміну від сценарних систем, GPT-чатботи використовують генеративний підхід. Це означає, що відповідь формується динамічно на основі аналізу контексту та змісту повідомлення користувача. Такий підхід забезпечує:

- більш природну комунікацію;
- гнучкість обробки запитів;
- підтримку складних діалогів;
- персоналізацію відповідей;
- можливість роботи з непередбачуваними запитами.

Порівняльну характеристику сценарних та GPT-орієнтованих чатботів наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика сценарних та GPT-орієнтованих чатботів

Характеристика	Сценарний чатбот	GPT-чатбот
Принцип роботи	Шаблони та правила	Генерація тексту AI-моделлю
Гнучкість	Низька	Висока
Робота з контекстом	Обмежена	Повноцінна
Обробка складних запитів	Частково можлива	Ефективна
Природність відповідей	Середня	Висока
Персоналізація	Мінімальна	Розширена

Використання GPT-моделей у чатботах має особливо важливе значення для сфери освіти. Університети та освітні установи активно впроваджують AI-

технології для автоматизації консультативної підтримки студентів та вступників.

GPT-чатботи можуть використовуватися для:

- надання інформації щодо вступної кампанії;
- підтримки дистанційного навчання;
- автоматизації відповідей на типові питання;
- допомоги у виборі освітньої програми;
- підтримки студентів під час навчання.

Однією з головних переваг GPT-чатботів є можливість цілодобової підтримки користувачів. Це особливо актуально для вступних кампаній, під час яких кількість запитів до приймальних комісій значно зростає. Використання AI-чатбота дозволяє зменшити навантаження на персонал та забезпечити оперативне надання інформації.

Сучасні GPT-чатботи можуть працювати як окремі вебсистеми або інтегруватися з популярними платформами:

- Telegram;
- Facebook Messenger;
- Discord;
- вебсайтами;
- мобільними додатками.

Важливим напрямом розвитку GPT-чатботів є використання технології RAG (Retrieval-Augmented Generation). Даний підхід передбачає поєднання генеративної AI-моделі з механізмом пошуку інформації у зовнішніх джерелах даних. Завдяки цьому чатбот може використовувати актуальну інформацію з баз знань, документів або вебресурсів університету.

Використання RAG-технологій є особливо важливим для освітніх систем, оскільки правила вступу, перелік спеціальностей та умови навчання можуть регулярно змінюватися. Інтеграція GPT-моделі з актуальною базою знань дозволяє зменшити ризик генерації застарілої або некоректної інформації.

Попри значні переваги GPT-моделей, їх використання супроводжується певними проблемами та обмеженнями. Однією з основних проблем є можливість

генерації неточних або помилкових відповідей. AI-модель не завжди гарантує абсолютну достовірність інформації, тому результати роботи чатбота потребують контролю та перевірки.

Іншим важливим аспектом є питання захисту персональних даних та інформаційної безпеки. У разі використання зовнішніх AI-сервісів необхідно забезпечити захист конфіденційної інформації користувачів та дотримання вимог законодавства щодо обробки персональних даних.

Крім того, використання GPT-моделей пов'язане з певними технічними та фінансовими обмеженнями. Для роботи AI-систем можуть використовуватися хмарні API-сервіси, які потребують стабільного інтернет-з'єднання та фінансових витрат на обробку запитів. Це необхідно враховувати під час планування проєкту розробки чатбота.

Окремої уваги потребують етичні аспекти використання штучного інтелекту в освіті. Важливо забезпечити прозорість роботи AI-систем, уникати дискримінаційних відповідей та контролювати якість генерованого контенту.

Таким чином, технології штучного інтелекту та GPT-моделі є одним із найбільш перспективних напрямів розвитку сучасних чатботів. Використання GPT-технологій дозволяє забезпечити природну комунікацію, підтримку складних діалогів, персоналізацію відповідей та автоматизацію консультаційної підтримки користувачів. Для закладів вищої освіти впровадження GPT-чатботів відкриває можливості підвищення ефективності взаємодії зі вступниками та розвитку сучасної цифрової інфраструктури університету.

1.4 Аналіз підходів та методологій управління IT-проєктами

Ефективне управління IT-проєктами є одним із ключових факторів успішної реалізації сучасних інформаційних систем. Особливо важливого значення управління проєктами набуває у сфері розробки програмного забезпечення, де процес створення продукту супроводжується високим рівнем невизначеності, необхідністю швидкої адаптації до змін та постійною взаємодією

із замовником. Для забезпечення ефективної реалізації ІТ-проектів використовуються різні підходи та методології управління, які визначають організацію робіт, розподіл ресурсів, контроль виконання завдань та взаємодію учасників проекту.

Методологія управління проектами являє собою сукупність принципів, методів, процесів та інструментів, які використовуються для планування, організації, реалізації та контролю виконання проекту. Вибір методології залежить від типу проекту, рівня складності, стабільності вимог, особливостей команди та очікувань замовника.

У сфері ІТ найбільш поширеними є традиційні та гнучкі підходи до управління проектами. До традиційних методологій належить каскадна модель Waterfall, а до гнучких – Agile, Scrum, Kanban та їх комбіновані варіанти.

Однією з перших та найбільш відомих методологій управління ІТ-проектами є каскадна модель Waterfall. Даний підхід передбачає послідовне виконання етапів проекту, де кожна фаза починається лише після завершення попередньої. Типовими етапами Waterfall є:

- аналіз вимог;
- проектування;
- розробка;
- тестування;
- впровадження;
- супровід.

Основною перевагою каскадної моделі є чітка структура процесів та детальне документування всіх етапів проекту. Waterfall добре підходить для проектів із чітко визначеними та стабільними вимогами, де зміни під час реалізації є мінімальними. Крім того, такий підхід спрощує планування бюджету, термінів та ресурсів.

Водночас використання Waterfall у сучасних ІТ-проектах має низку недоліків. Однією з основних проблем є низька гнучкість щодо змін вимог. У разі виявлення нових потреб або помилок на пізніх етапах проекту внесення змін

може потребувати значних витрат часу та ресурсів. Крім того, замовник отримує готовий продукт лише наприкінці проєкту, що ускладнює оперативне коригування функціоналу.

Для проєктів, пов'язаних із розробкою сучасних інформаційних систем та AI-рішень, більш доцільним є використання гнучких методологій управління проєктами. Гнучкі підходи (Agile methodologies) орієнтовані на швидку адаптацію до змін, постійну взаємодію із замовником та поетапне створення програмного продукту.

Agile не є окремою методологією, а являє собою сукупність принципів та підходів до організації процесу розробки програмного забезпечення. Основні принципи Agile були сформульовані в Agile Manifesto у 2001 році. Відповідно до Agile-підходу, пріоритет надається:

- взаємодії між людьми;
- працюючому програмному забезпеченню;
- співпраці із замовником;
- готовності до змін.

Головною особливістю Agile є ітеративний підхід до розробки. Проєкт поділяється на короткі цикли – ітерації, або спринти, у межах яких команда реалізує окрему частину функціоналу. Після завершення кожного спринту формується проміжний результат, який може бути продемонстрований замовнику.

Однією з найбільш популярних Agile-методологій є Scrum. Scrum широко використовується у сфері розробки програмного забезпечення завдяки своїй гнучкості, простоті організації процесів та ефективності управління командою.

У Scrum процес розробки поділяється на короткі спринти тривалістю від одного до чотирьох тижнів. Наприкінці кожного спринту команда формує готовий інкремент продукту. Основними ролями у Scrum є:

- Product Owner;
- Scrum Master;
- команда розробки.

Product Owner відповідає за формування вимог до продукту та визначення пріоритетів завдань. Scrum Master забезпечує організацію процесу роботи команди та дотримання принципів Scrum. Команда розробки безпосередньо виконує завдання зі створення програмного продукту.

До основних переваг Scrum належать:

- швидка адаптація до змін;
- постійний контроль прогресу;
- регулярна взаємодія із замовником;
- раннє виявлення проблем;
- висока прозорість процесів;
- поступове формування готового продукту.

Scrum є особливо ефективним для проєктів із високим рівнем невизначеності вимог, що характерно для AI-орієнтованих систем. Під час розробки GPT-чатбота вимоги до функціоналу можуть уточнюватися у процесі реалізації проєкту, а окремі рішення – змінюватися після тестування взаємодії користувачів із системою. У таких умовах використання Scrum дозволяє оперативно вносити зміни та поступово вдосконалювати продукт.

Ще одним популярним Agile-підходом є Kanban. На відміну від Scrum, Kanban не використовує фіксовані спринти та орієнтується на безперервний потік виконання завдань. Основним інструментом Kanban є Kanban-дошка, на якій відображається поточний стан задач.

Основними перевагами Kanban є:

- гнучкість;
- простота впровадження;
- візуалізація процесу роботи;
- можливість оперативного управління завданнями.

Kanban добре підходить для команд підтримки та супроводу програмних систем, однак для великих проєктів розробки програмного забезпечення частіше використовується Scrum або комбінований підхід Scrum-Kanban.

У сучасних IT-проектах також активно використовуються hybrid-підходи, які поєднують елементи традиційних та гнучких методологій. Наприклад, загальне планування проєкту може виконуватися за принципами Waterfall, тоді як безпосередня розробка програмного забезпечення здійснюється за Scrum. Такий підхід дозволяє поєднати переваги детального планування та гнучкості управління.

Порівняльну характеристику основних методологій управління IT-проектами наведено в таблиці 1.3.

Таблиця 1.3 – Порівняльна характеристика основних методологій управління IT-проектами [17-18]

Методологія	Переваги	Недоліки
Waterfall	Чітка структура, прогнозованість	Низька гнучкість
Agile	Адаптивність, швидка реакція на зміни	Складність прогнозування термінів
Scrum	Контроль спринтів, постійний зворотний зв'язок	Потребує активної участі команди
Kanban	Гнучкість, візуалізація процесів	Менш ефективний для складних проєктів
Hybrid	Поєднання переваг різних підходів	Складність організації процесів

Для проєкту розробки GPT-чатбота підтримки вступників найбільш доцільним є використання Agile-підходу із застосуванням Scrum. Це пояснюється кількома факторами.

По-перше, AI-проекти характеризуються високим рівнем невизначеності. На етапі розробки можуть виникати нові вимоги до функціоналу, змінюватися сценарії взаємодії користувачів або уточнюватися вимоги до інтеграції із зовнішніми сервісами.

По-друге, GPT-чатбот потребує поступового тестування та вдосконалення. Якість роботи AI-системи значною мірою залежить від результатів взаємодії з користувачами, тому можливість поетапного вдосконалення продукту є важливою перевагою Agile-підходу.

По-третє, Scrum дозволяє забезпечити постійний контроль процесу розробки та швидке виявлення проблем. Регулярні зустрічі команди,

демонстрація результатів спринтів та аналіз виконаної роботи сприяють підвищенню ефективності управління проєктом.

Крім вибору методології, важливим аспектом управління ІТ-проєктами є використання спеціалізованих програмних інструментів. Для планування та контролю проєктів широко використовуються:

- Microsoft Project;
- Jira;
- Trello;
- Asana;
- ClickUp;
- GitHub Projects.

Зазначені системи дозволяють здійснювати:

- планування розкладу;
- управління ресурсами;
- контроль виконання завдань;
- формування backlog;
- моніторинг ризиків;
- командну взаємодію.

Таким чином, сучасні ІТ-проєкти потребують використання гнучких підходів до управління, які забезпечують адаптацію до змін, постійну взаємодію із замовником та можливість поетапного вдосконалення продукту. Для проєкту розробки GPT-чатбота підтримки вступників найбільш доцільним є застосування Agile-методології Scrum, яка дозволяє забезпечити ефективну організацію процесу розробки, швидку реакцію на зміни та поступове формування якісного програмного продукту.

1.6 Висновки до розділу 1

У першому розділі було проведено аналіз особливостей цифровізації закладів вищої освіти та сучасних тенденцій розвитку інформаційних систем

підтримки вступників. Визначено, що в умовах активної цифрової трансформації університетів особливого значення набуває автоматизація комунікаційних процесів і впровадження інтелектуальних сервісів для взаємодії зі вступниками.

Проаналізовано сучасні інформаційні системи підтримки вступників, зокрема офіційні вебсайти університетів, FAQ-системи, Telegram-боти та AI-орієнтовані чатботи. Встановлено, що традиційні засоби консультування мають обмежений рівень інтерактивності та не забезпечують достатньої гнучкості під час обробки складних або нестандартних запитів користувачів.

Досліджено технології штучного інтелекту та GPT-моделі, які використовуються у сучасних чатботах. Визначено, що використання великих мовних моделей дозволяє забезпечити більш природну взаємодію з користувачами, підтримку контексту діалогу та персоналізацію відповідей. Водночас встановлено, що реалізація AI-проєктів у сфері освіти супроводжується низкою ризиків, пов'язаних із точністю відповідей, актуальністю даних, інформаційною безпекою та залежністю від зовнішніх сервісів.

Також було проаналізовано сучасні підходи та методології управління IT-проєктами. Встановлено, що для проєктів розробки AI-орієнтованих систем найбільш доцільним є використання гнучких методологій управління, зокрема Scrum, які забезпечують адаптивність до змін, поетапну реалізацію функціоналу та постійний контроль процесу розробки.

Отже, результати проведеного аналізу підтверджують актуальність розробки GPT-чатбота підтримки вступників університету та доцільність використання сучасних AI-технологій і гнучких підходів до управління IT-проєктами.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТА ПРОЄКТУВАННЯ GPT-ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ

2.1 Постановка задачі розробки GPT-чатбота підтримки вступників

У сучасних умовах цифровізації закладів вищої освіти особливого значення набуває автоматизація процесів інформаційної підтримки вступників. Вступна кампанія є одним із найбільш відповідальних та інформаційно навантажених процесів діяльності університету, оскільки саме на цьому етапі формується взаємодія між закладом освіти та потенційними здобувачами вищої освіти. Ефективність інформаційної підтримки вступників значною мірою впливає на імідж університету, якість комунікації з абітурієнтами та загальний рівень організації вступної кампанії.

У процесі вступу абітурієнти стикаються з необхідністю отримання значного обсягу інформації щодо:

- правил прийому;
- переліку спеціальностей;
- освітніх програм;
- вартості навчання;
- термінів подачі документів;
- конкурсного відбору;
- прохідних балів;
- особливостей навчального процесу;
- можливостей працевлаштування;
- наявності бюджетних місць.

При цьому вступники очікують швидкого отримання актуальної інформації у зручному цифровому форматі. Основними каналами комунікації сьогодні є офіційні вебсайти університетів, соціальні мережі, месенджери та електронні сервіси. Однак традиційні засоби підтримки вступників мають низку недоліків.

Інформація на офіційних сайтах університетів часто має складну структуру та великий обсяг, що ускладнює її пошук користувачами. Крім того, значна кількість вступників звертається до приймальної комісії з однотипними питаннями, що створює додаткове навантаження на працівників університету, особливо у період активної фази вступної кампанії.

Традиційні методи консультування, такі як телефонна підтримка або електронна пошта, не забезпечують достатньої швидкості обробки звернень та цілодобової доступності сервісу. В умовах великої кількості запитів це може призводити до затримок у наданні відповідей, перевантаження персоналу та зниження якості комунікації з вступниками.

Одним із перспективних напрямів вирішення зазначених проблем є використання технологій штучного інтелекту та GPT-орієнтованих чатботів. Такі системи дозволяють автоматизувати процес інформаційної підтримки користувачів, забезпечити цілодобову доступність сервісу та значно підвищити швидкість обробки запитів.

GPT-чатбот являє собою інтелектуальну інформаційну систему, побудовану на основі великих мовних моделей, здатних аналізувати текстові повідомлення природною мовою та формувати відповіді відповідно до контексту діалогу. На відміну від традиційних сценарних чатботів, GPT-системи забезпечують більш гнучку та природну взаємодію з користувачами.

Розробка GPT-чатбота підтримки вступників дозволить:

- автоматизувати надання консультацій;
- зменшити навантаження на приймальну комісію;
- забезпечити швидкий доступ до інформації;
- покращити якість комунікації зі вступниками;
- підвищити рівень цифровізації університету;
- забезпечити цілодобову підтримку користувачів.

Основними користувачами системи є:

- вступники;
- батьки вступників;

- працівники приймальної комісії;
- адміністратори системи.

Передбачається, що система повинна забезпечувати підтримку текстового діалогу природною мовою та надавати користувачам актуальну інформацію щодо вступної кампанії. Чатбот має працювати через вебінтерфейс або месенджер Telegram, що забезпечить зручність використання та доступність сервісу для широкого кола користувачів.

До основних функцій системи належать:

- обробка текстових запитів користувачів;
- генерація відповідей за допомогою GPT-моделі;
- надання інформації щодо спеціальностей та освітніх програм;
- консультування щодо правил вступу;
- підтримка діалогу природною мовою;
- інтеграція з базою знань університету;
- адміністрування інформаційного наповнення системи.

Під час розробки GPT-чатбота необхідно враховувати низку обмежень. Одним із ключових обмежень є необхідність забезпечення актуальності інформації, оскільки правила вступу можуть змінюватися щороку. Крім того, система повинна забезпечувати захист персональних даних користувачів і стабільність роботи за значного навантаження.

Також важливим аспектом є забезпечення достатньої точності відповідей AI-системи. GPT-модель повинна використовувати актуальні дані з бази знань університету та формувати коректні відповіді відповідно до запитів користувачів.

З огляду на високий рівень невизначеності та необхідність поступового вдосконалення функціоналу системи, доцільним є використання гнучких підходів до управління проектом, зокрема методології Scrum. Це дозволить поетапно реалізовувати функціональні можливості чатбота, тестувати роботу системи та оперативно реагувати на зміни в вимогах.

Отже, задача розробки GPT-чатбота підтримки вступників є актуальною в умовах цифрової трансформації закладів вищої освіти та потребує комплексного

підходу до проєктування інформаційної системи, вибору технологій штучного інтелекту й організації процесу управління IT-проєктом.

2.2 Аналіз функціональних та нефункціональних вимог до системи

Одним із ключових етапів проєктування інформаційної системи є визначення та аналіз вимог до її функціонування. Саме вимоги формують основу для подальшого проєктування архітектури системи, вибору технологій реалізації та організації процесу розробки програмного забезпечення. Для GPT-чатбота підтримки вступників правильне формування вимог має особливе значення, оскільки система повинна забезпечувати ефективну взаємодію з користувачами, підтримку природної мови та високий рівень доступності інформації.

Загалом вимоги до інформаційної системи поділяються на функціональні та нефункціональні.

Функціональні вимоги визначають перелік функцій і сервісів, які система має забезпечувати. Нефункціональні вимоги описують характеристики якості системи, умови її роботи, обмеження та параметри ефективності.

Основною метою GPT-чатбота є автоматизація процесу інформаційної підтримки вступників університету. Система повинна забезпечувати швидке надання актуальної інформації щодо вступної кампанії та підтримувати взаємодію з користувачами у форматі текстового діалогу природною мовою.

До основних функціональних вимог системи належать:

1. Обробка текстових запитів користувачів.
2. Підтримка діалогу природною мовою.
3. Генерація відповідей за допомогою GPT-моделі.
4. Надання інформації щодо вступної кампанії.
5. Пошук інформації у базі знань.
6. Підтримка багатокрокового діалогу.
7. Інтеграція з Telegram.
8. Адміністрування системи.

9. Оновлення інформаційної бази.

Однією з базових функцій системи є обробка текстових повідомлень користувачів. Чатбот повинен аналізувати введений текст, визначати зміст запиту та формувати відповідь відповідно до контексту діалогу. При цьому система повинна підтримувати різні варіанти формулювання запитів у природній мові.

Наступною важливою функцією є генерація відповідей за допомогою моделі GPT. Система повинна використовувати AI-модель для формування інформативних і логічно узгоджених відповідей на запити користувачів. Для забезпечення актуальності інформації GPT-модель має використовувати базу знань університету, що містить дані про:

- спеціальності;
- освітні програми;
- умови вступу;
- терміни подачі документів;
- вартість навчання;
- контакти приймальної комісії;
- правила прийому.

Важливою функціональною вимогою є підтримка багатокрокового діалогу. Система повинна враховувати контекст попередніх повідомлень користувача та підтримувати логічну послідовність спілкування. Це дозволить забезпечити більш природний формат взаємодії та покращити користувацький досвід.

Оскільки більшість вступників активно використовує месенджери, система повинна підтримувати інтеграцію з Telegram. Telegram-бот має забезпечувати отримання повідомлень від користувачів; передачу запитів до AI-модуля; відображення відповідей чатбота; підтримку команд та меню навігації.

Окрему увагу необхідно приділити функціоналу адміністрування системи. Адміністратор повинен мати можливість: оновлювати інформацію у базі знань; додавати нові дані; редагувати правила роботи системи; контролювати роботу чатбота; аналізувати статистику використання системи.

Для забезпечення ефективної роботи GPT-чатбота доцільно реалізувати функцію пошуку інформації у внутрішній базі знань університету. Це дозволить використовувати актуальні дані та зменшити ризик генерації неточних відповідей.

Функціональні вимоги до системи наведено у таблиці 2.1.

Таблиця 2.1 – Функціональні вимоги

№	Функціональна вимога	Опис
1	Обробка текстових запитів	Аналіз повідомлень користувача
2	Генерація відповідей	Формування відповідей GPT-моделлю
3	Пошук інформації	Отримання даних із бази знань
4	Підтримка діалогу	Врахування контексту розмови
5	Telegram-інтеграція	Робота через месенджер
6	Адміністрування	Оновлення інформації та контроль системи
7	Консультації щодо вступу	Надання інформації вступникам

Крім функціональних вимог, важливе значення мають нефункціональні вимоги, які визначають характеристики якості роботи системи. До основних нефункціональних вимог GPT-чатбота належать:

1. Швидкодія.
2. Доступність.
3. Масштабованість.
4. Безпека.
5. Надійність.
6. Зручність використання.
7. Актуальність інформації.

Однією з ключових нефункціональних вимог є швидкодія системи. Час формування відповіді чатбота повинен бути мінімальним, оскільки затримки негативно впливають на користувацький досвід. Бажано, щоб середній час відповіді системи не перевищував кількох секунд.

Важливою характеристикою є доступність системи. GPT-чатбот повинен забезпечувати цілодобовий доступ користувачів до сервісу незалежно від часу доби. Це особливо актуально у період активної фази вступної кампанії.

Ще однією важливою вимогою є масштабованість. Система повинна підтримувати можливість збільшення кількості користувачів без суттєвого погіршення продуктивності. У період вступної кампанії навантаження на чатбот може значно зростати, тому система має бути готовою до одночасної обробки великої кількості запитів.

Важливу роль відіграє інформаційна безпека системи. GPT-чатбот повинен забезпечувати захист персональних даних; безпечну передачу інформації; контроль доступу до адміністративних функцій; а також захист від несанкціонованого втручання.

Особливого значення для AI-систем має вимога актуальності інформації. База знань чатбота повинна регулярно оновлюватися відповідно до змін правил вступу, освітніх програм та нормативної документації університету.

Нефункціональні вимоги до системи наведено у таблиці 2.2.

Таблиця 2.2 – Нефункціональні вимоги

№	Нефункціональна вимога	Опис
1	Швидкодія	Мінімальний час формування відповіді
2	Доступність	Цілодобова робота системи
3	Масштабованість	Підтримка великої кількості користувачів
4	Безпека	Захист персональних даних
5	Надійність	Стабільна робота системи
6	Зручність використання	Простий та зрозумілий інтерфейс
7	Актуальність даних	Своєчасне оновлення бази знань

Для забезпечення ефективної роботи системи також необхідно враховувати технічні обмеження. GPT-чатбот залежить від стабільності роботи зовнішнього AI-сервісу та доступності API. Крім того, використання GPT-моделей може супроводжуватися фінансовими витратами на обробку запитів.

Важливим аспектом є забезпечення якості відповідей AI-системи. Чатбот повинен надавати зрозумілі, логічні та коректні відповіді без використання некоректної або суперечливої інформації. Для цього необхідно передбачити механізми тестування та контролю якості роботи системи.

Під час формування вимог також необхідно враховувати особливості цільової аудиторії. Основними користувачами системи є вступники та абітурієнти, тому інтерфейс чатбота повинен бути простим, інтуїтивно зрозумілим та адаптованим для мобільних пристроїв.

Таким чином, аналіз функціональних і нефункціональних вимог дозволив визначити основні характеристики GPT-чатбота підтримки вступників університету. Система повинна забезпечувати автоматизовану обробку запитів природною мовою, інтеграцію з Telegram, підтримку актуальної бази знань та високий рівень доступності й безпеки. Отримані результати створюють основу для подальшого проєктування архітектури системи та вибору технологій її реалізації, зокрема, для формування User Stories (табл. 2.3).

Таблиця 2.3 – User Stories для GPT-чатбота підтримки вступників

ID	User Story	Пріоритет
US-01	Як вступник, я хочу отримувати інформацію про спеціальності університету, щоб обрати напрям навчання	Високий
US-02	Як вступник, я хочу дізнаватися про умови вступу, щоб правильно підготувати документи	Високий
US-03	Як вступник, я хочу отримувати інформацію про терміни подачі заяв, щоб не пропустити дедлайни	Високий
US-04	Як вступник, я хочу ставити запитання природною мовою, щоб швидко отримувати зрозумілі відповіді	Високий
US-05	Як вступник, я хочу користуватися чатботом через Telegram, щоб мати швидкий доступ до сервісу	Високий
US-06	Як вступник, я хочу отримувати інформацію про вартість навчання, щоб оцінити можливість вступу	Середній
US-07	Як вступник, я хочу отримувати рекомендації щодо спеціальностей, щоб обрати найбільш відповідну освітню програму	Середній
US-08	Як вступник, я хочу отримувати контакти приймальної комісії, щоб мати можливість звернутися до представників університету	Середній
US-09	Як вступник, я хочу отримувати інформацію про бюджетні місця, щоб оцінити свої шанси на вступ	Середній
US-10	Як вступник, я хочу отримувати відповіді цілодобово, щоб користуватися сервісом у будь-який час	Високий
US-11	Як адміністратор системи, я хочу оновлювати базу знань чатбота, щоб система надавала актуальну інформацію	Високий
US-12	Як адміністратор системи, я хочу переглядати статистику використання чатбота, щоб аналізувати ефективність системи	Середній
US-13	Як адміністратор системи, я хочу редагувати інформацію про вступну кампанію, щоб підтримувати актуальність даних	Високий
US-14	Як працівник приймальної комісії, я хочу зменшити кількість типових запитів, щоб оптимізувати навантаження на персонал	Високий
US-15	Як користувач, я хочу отримувати швидкі відповіді на запити, щоб економити час	Високий

Сформовані user stories дозволяють визначити основні потреби користувачів системи та забезпечують основу для формування backlog проєкту. Використання user stories відповідає принципам Agile та Scrum і дозволяє орієнтувати процес розробки на потреби кінцевих користувачів системи.

2.3 Аналіз та вибір технологій реалізації системи

Одним із ключових етапів проєктування інформаційної системи є вибір технологій її реалізації. Від правильності вибору програмних засобів, мов програмування, платформ та сервісів залежать ефективність роботи системи, можливість масштабування, стабільність функціонування та якість взаємодії з користувачами. Для GPT-чатбота підтримки вступників особливого значення набуває вибір технологій, здатних забезпечити обробку природної мови, інтеграцію із зовнішніми сервісами та підтримку сучасних каналів комунікації.

Проєктована система повинна забезпечувати підтримку текстового діалогу природною мовою; інтеграцію з AI-моделлю; обробку великої кількості запитів; взаємодію з Telegram; роботу з базою знань; можливість адміністрування; а також масштабованість і надійність.

Для реалізації зазначених функцій необхідно обрати відповідні технології для серверної частини системи; AI-модуля; інтеграції з месенджерами; зберігання даних; адміністрування системи.

Одним із найважливіших компонентів системи є AI-модуль, який забезпечує генерацію відповідей на запити користувачів. Для реалізації цього функціоналу доцільно використовувати GPT-моделі компанії OpenAI через OpenAI API.

OpenAI API забезпечує доступ до великих мовних моделей, здатних аналізувати текстові повідомлення; підтримувати контекст діалогу; генерувати відповіді природною мовою; працювати з багатомовними запитами; інтегруватися із зовнішніми інформаційними системами.

Основними перевагами використання OpenAI API є:

- висока якість генерації тексту;
- підтримка природної мови;
- масштабованість;
- простота інтеграції;
- наявність документації та бібліотек для розробки.

Разом із тим використання OpenAI API має певні обмеження, серед яких:

- залежність від зовнішнього сервісу;
- необхідність доступу до мережі Інтернет;
- фінансові витрати на обробку запитів;
- можливі обмеження кількості API-запитів.

Для реалізації серверної частини системи доцільно використовувати мову програмування Python. Python є однією з найбільш популярних мов розробки AI-орієнтованих систем та вебзастосунків. Основними перевагами Python є:

- простота синтаксису;
- велика кількість бібліотек;
- підтримка AI та NLP;
- активна спільнота розробників;
- зручність інтеграції із зовнішніми сервісами.

Python широко використовується для створення чатботів, AI-систем та вебсервісів, що робить його оптимальним вибором для реалізації GPT-чатбота підтримки вступників.

Для створення серверного API системи доцільно використовувати вебфреймворк FastAPI. Даний фреймворк забезпечує:

- високу швидкодію;
- підтримку REST API;
- асинхронну обробку запитів;
- автоматичне формування документації;
- простоту інтеграції з AI-сервісами.

FastAPI є сучасним фреймворком для Python, який активно використовується для розробки AI-орієнтованих вебсистем та мікросервісної архітектури.

Альтернативою FastAPI може бути Flask – легкий вебфреймворк для Python. Flask є простішим у використанні, однак поступається FastAPI за рівнем продуктивності та підтримкою асинхронної обробки даних. З огляду на необхідність роботи з великою кількістю запитів та інтеграції з AI-моделлю, FastAPI є більш доцільним рішенням.

Для забезпечення взаємодії користувачів із системою передбачається інтеграція чатбота з платформою Telegram. Telegram є одним із найбільш популярних месенджерів серед молоді та забезпечує зручний формат текстової взаємодії.

Для інтеграції з Telegram доцільно використовувати Telegram Bot API, яке дозволяє:

- отримувати повідомлення користувачів;
- надсилати відповіді;
- реалізовувати команди;
- підтримувати меню навігації;
- організовувати взаємодію з серверною частиною системи.

Основними перевагами Telegram Bot API є:

- безкоштовне використання;
- простота інтеграції;
- підтримка вебхуків;
- швидка обробка повідомлень;
- популярність серед користувачів.

Для зберігання інформації системи необхідно використовувати систему управління базами даних. Оскільки GPT-чатбот повинен працювати з базою знань університету, доцільно використовувати реляційну базу даних PostgreSQL.

PostgreSQL є сучасною системою управління базами даних, яка забезпечує:

- високу надійність;

- масштабованість;
- підтримку складних запитів;
- безпечне зберігання даних;
- інтеграцію з Python.

У базі даних можуть зберігатися:

- інформація про спеціальності;
- правила вступу;
- FAQ;
- історія взаємодії користувачів;
- адміністративні налаштування системи.

Для покращення якості відповідей GPT-чатбота доцільно використовувати технологію RAG (Retrieval-Augmented Generation). Даний підхід поєднує генеративну AI-модель із механізмом пошуку інформації у базі знань.

Використання RAG дозволяє:

- забезпечити актуальність відповідей;
- зменшити ризик генерації помилкової інформації;
- використовувати внутрішню базу знань університету;
- покращити якість консультацій.

Для реалізації RAG-системи доцільно використовувати бібліотеку LangChain. LangChain забезпечує інтеграцію з GPT-моделями; роботу з базами знань; побудову AI-ланцюгів обробки інформації; підтримку Retrieval-підходів.

Для розробки адміністративного інтерфейсу системи можуть використовуватися вебтехнології HTML, CSS та JavaScript.

За необхідності можлива інтеграція frontend-фреймворків, наприклад React або Vue.js. Однак для бакалаврської роботи достатнім є створення базового вебінтерфейсу адміністрування.

Порівняльний аналіз основних технологій реалізації системи наведено в таблиці 2.3.

Архітектура системи передбачає взаємодію кількох основних компонентів:

- користувача;

- Telegram-бота;
- серверної частини;
- AI-модуля;
- бази знань;
- адміністративного модуля.

Таблиця 2.3 – Порівняльний аналіз основних технологій реалізації системи

Технологія	Призначення	Переваги
OpenAI API	Генерація відповідей	Висока якість NLP
Python	Серверна логіка	Простота та підтримка AI
FastAPI	Backend API	Висока швидкодія
Telegram Bot API	Взаємодія з користувачами	Простота інтеграції
PostgreSQL	Зберігання даних	Надійність та масштабованість
LangChain	Реалізація RAG	Інтеграція AI та баз знань

Після надсилання повідомлення користувачем Telegram-бот передає запит до серверної частини системи. Backend здійснює обробку повідомлення, за необхідності виконує пошук інформації у базі знань та передає запит до GPT-моделі через OpenAI API. Після генерації відповіді система повертає результат користувачу через Telegram.

Під час вибору технологій також необхідно враховувати ризики та обмеження. Основними ризиками є:

- залежність від OpenAI API;
- можливі фінансові витрати;
- обмеження швидкодії при високому навантаженні;
- потреба в регулярному оновленні бази знань.

Попри зазначені обмеження, обраний технологічний стек забезпечує достатній рівень функціональності, масштабованості та зручності реалізації GPT-чатбота для підтримки вступників.

Таким чином, у результаті аналізу було обрано технології реалізації системи, які забезпечують підтримку AI-функціоналу, інтеграцію з Telegram, обробку природної мови та роботу з базою знань університету. Використання Python, FastAPI, OpenAI API, PostgreSQL та LangChain дозволяє створити

сучасну AI-орієнтовану інформаційну систему підтримки вступників із можливістю подальшого масштабування та вдосконалення функціоналу.

2.4 Моделювання функціонування системи

Одним із важливих етапів проєктування інформаційної системи є моделювання її функціонування. Моделювання дозволяє формалізувати основні процеси взаємодії користувачів із системою, визначити структуру інформаційних потоків, описати сценарії роботи та забезпечити більш детальне розуміння логіки функціонування програмного продукту. Для GPT-чатбота підтримки вступників моделювання є необхідним етапом, оскільки система повинна забезпечувати складну взаємодію між користувачем, AI-модулем, базою знань та зовнішніми сервісами.

Для моделювання функціонування GPT-чатбота доцільно використовувати UML-діаграми. Застосування UML дозволяє візуалізувати структуру взаємодії користувачів із системою та відобразити логіку обробки запитів.

Однією з основних UML-діаграм для опису функціональності системи є діаграма варіантів використання (Use Case Diagram). Ця діаграма дозволяє визначити основних користувачів системи та перелік функцій, доступних їм.

У проєктованій системі основними акторами є вступник, адміністратор системи та працівник приймальної комісії (рис. 2.1).

Після визначення основних варіантів використання необхідно змоделювати процес взаємодії користувача із системою. Для цього доцільно використовувати UML Activity Diagram, яка дозволяє описати послідовність виконання дій у системі.

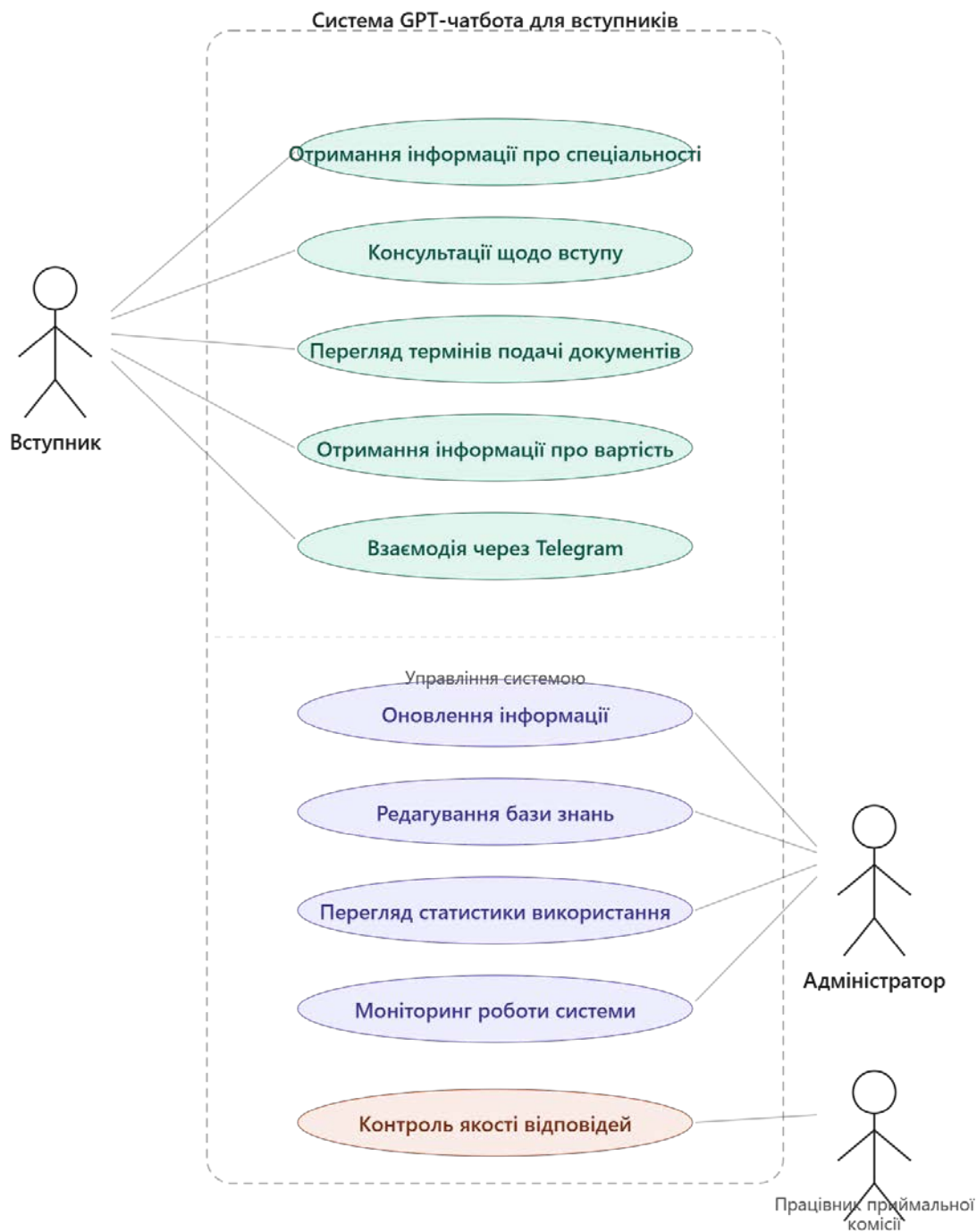


Рисунок 2.1 – UML Use Case Diagram системи

Процес взаємодії вступника з GPT-чатботом включає такі етапи (рис. 2.2).

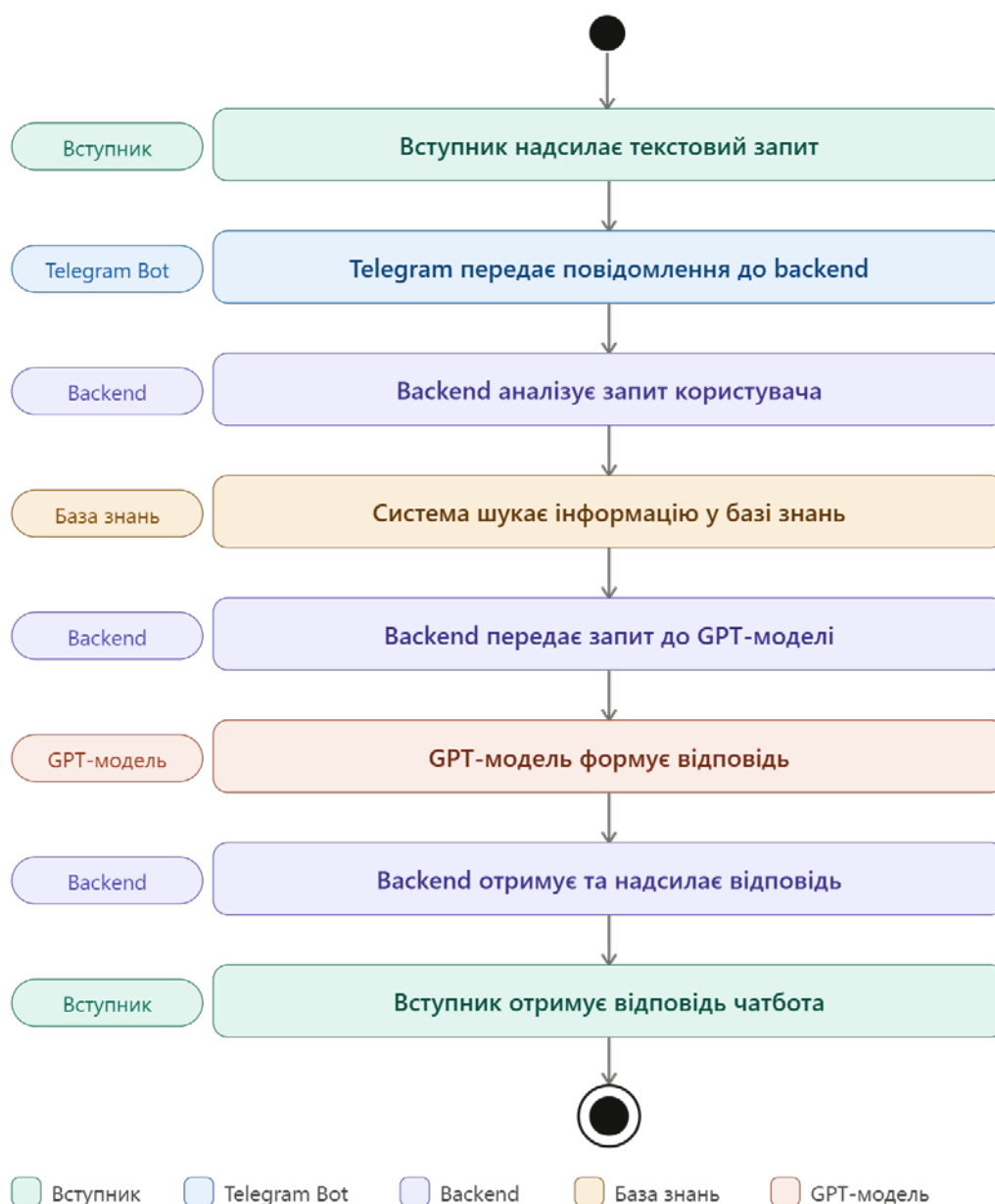


Рисунок 2.2 – Activity Diagram процесу взаємодії користувача із чатботом

Для більш детального опису взаємодії компонентів системи доцільно використовувати UML Sequence Diagram (рис. 2.3). Ця діаграма відображає послідовність обміну повідомленнями між компонентами системи в часовому порядку.

Основними компонентами взаємодії є:

- користувач;

- Telegram Bot API;
- backend-сервер;
- GPT API;
- база знань.

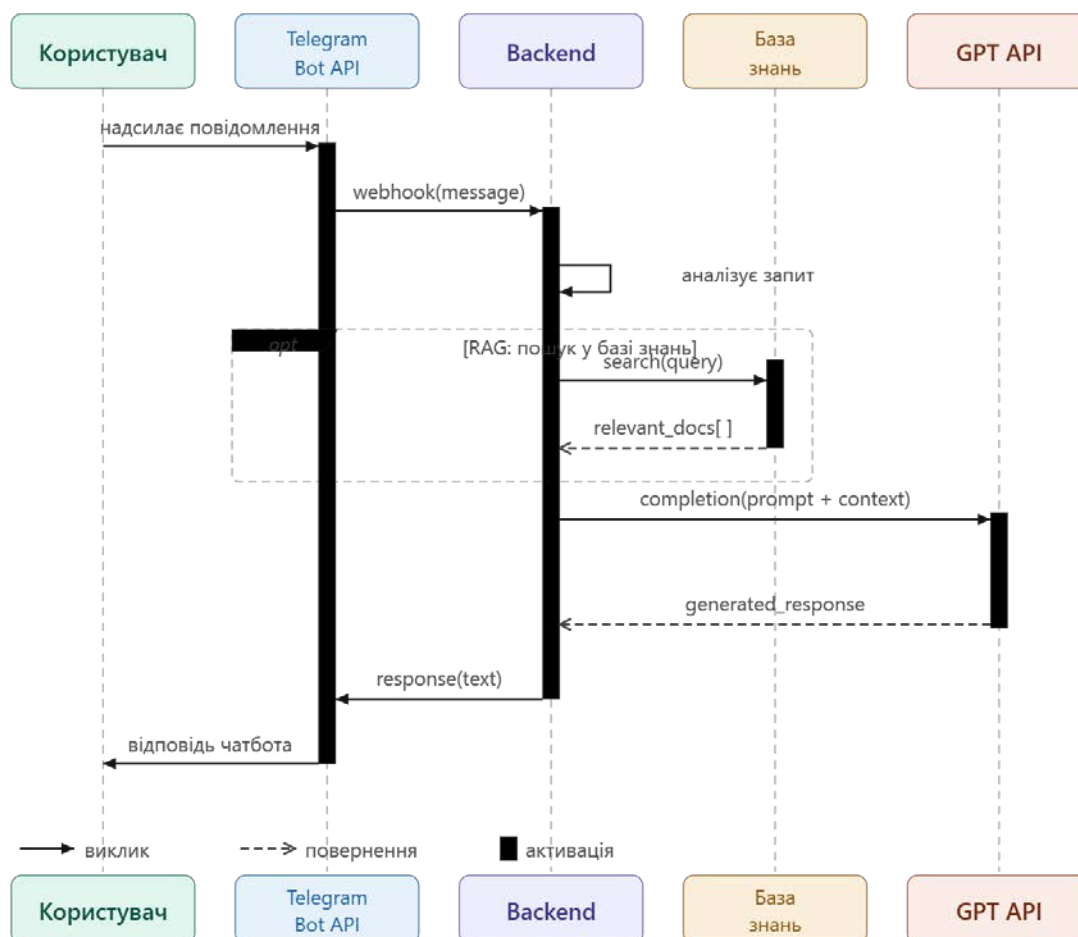


Рисунок 2.3 – UML Sequence Diagram

Послідовність взаємодії між компонентами системи є такою:

1. Користувач надсилає повідомлення чатботу.
2. Telegram Bot API передає повідомлення backend-серверу.
3. Backend аналізує запит.
4. За необхідності виконується пошук інформації у базі знань.
5. Backend надсилає запит до OpenAI API.
6. GPT-модель формує відповідь.
7. Backend отримує результат.

8. Відповідь передається користувачу через Telegram.

Sequence Diagram дозволяє визначити механізм взаємодії між окремими модулями системи та виявити критичні точки обробки інформації.

Важливим елементом моделювання є опис інформаційних потоків системи. Основними потоками даних є:

- повідомлення користувачів;
- запити до AI-моделі;
- інформація з бази знань;
- результати генерації відповідей;
- адміністративні дані.

У процесі функціонування системи відбувається постійна взаємодія між AI-модулем та базою знань університету. Для забезпечення актуальності відповідей GPT-чатбот повинен використовувати інформацію щодо:

- правил вступу;
- освітніх програм;
- спеціальностей;
- термінів вступної кампанії;
- вартості навчання.

З метою покращення якості відповідей доцільно реалізувати Retrieval-підхід, за якого перед генерацією відповіді система виконує пошук релевантної інформації в базі знань. Це дозволяє зменшити ризик генерації помилкової або застарілої інформації.

Таким чином, моделювання функціонування GPT-чатбота дозволило формалізувати процеси взаємодії користувачів із системою, визначити основні функціональні модулі та описати логіку роботи інформаційної системи.

2.5 Проєктування архітектури GPT-чатбота

Одним із ключових етапів розробки інформаційної системи є проєктування її архітектури. Архітектура системи визначає структуру програмного продукту,

взаємодію між окремими компонентами, механізми обробки даних і принципи організації програмних модулів. Для GPT-чатбота підтримки вступників правильне проєктування архітектури має особливе значення, оскільки система повинна забезпечувати інтеграцію AI-моделі, роботу з базою знань, підтримку користувачів у режимі реального часу та можливість масштабування.

Основною метою проєктування архітектури є визначення структури системи; опис основних модулів; організація взаємодії компонентів; забезпечення масштабованості та надійності; спрощення подальшої розробки та супроводу системи.

Під час проєктування GPT-чатбота доцільно використовувати модульний підхід, при якому система складається з окремих функціональних компонентів. Такий підхід забезпечує гнучкість системи, можливість масштабування, спрощення тестування, незалежне оновлення окремих модулів та зменшення складності супроводу.

Архітектура GPT-чатбота підтримки вступників включає такі основні компоненти:

- користувацький інтерфейс;
- Telegram Bot API;
- backend-сервер;
- AI-модуль;
- база знань;
- база даних;
- адміністративний модуль.

Загальну архітектуру системи доцільно подати у вигляді структурної схеми на рисунку 2.4.

Основним каналом взаємодії користувачів із системою є Telegram-бот. Користувач надсилає повідомлення через Telegram, після чого Telegram Bot API передає запит до серверної частини системи. Backend-сервер здійснює обробку повідомлення, виконує взаємодію з базою знань та AI-модулем, після чого формує відповідь і передає її користувачу.

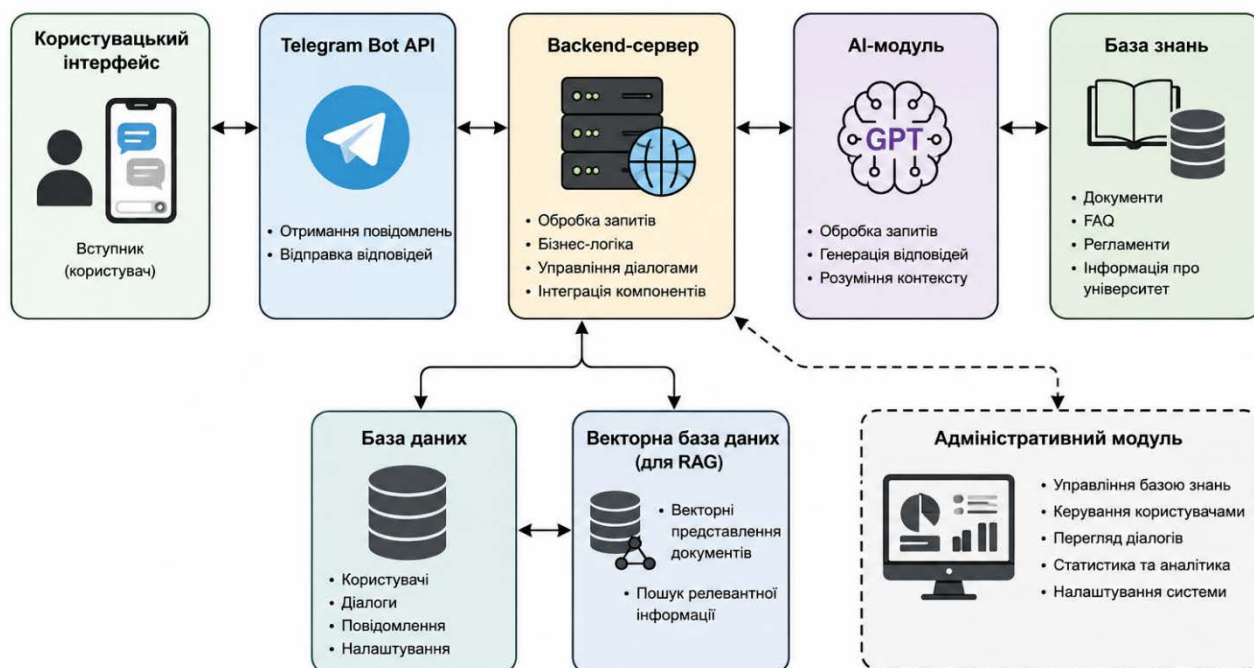


Рисунок 2.4 – Загальна архітектура системи

Telegram-бот виконує функції інтерфейсу взаємодії з користувачем. Основними задачами даного компонента є:

- прийом повідомлень;
- передача запитів до backend-сервера;
- відображення відповідей чатбота;
- підтримка меню навігації;
- реалізація команд користувача.

Використання Telegram як платформи взаємодії пояснюється його популярністю серед молоді, доступністю та зручністю. Крім того, Telegram Bot API забезпечує просту інтеграцію із серверною частиною системи.

Центральним елементом архітектури є backend-сервер, який забезпечує координацію роботи всіх компонентів системи. Backend реалізується за допомогою мови програмування Python та фреймворку FastAPI.

Основними функціями backend-сервера є:

- обробка запитів користувачів;
- взаємодія з Telegram API;
- інтеграція з OpenAI API;

- робота з базою знань;
- логування подій;
- управління бізнес-логікою системи;
- адміністрування.

Backend-сервер виконує роль проміжного рівня між користувачем, AI-моделлю та базою даних. Саме backend забезпечує обробку логіки роботи системи та контроль усіх інформаційних потоків.

Одним із найважливіших компонентів системи є AI-модуль. Даний модуль реалізує взаємодію з GPT-моделлю через OpenAI API та відповідає за генерацію текстових відповідей користувачам.

AI-модуль виконує такі функції:

- формування запиту до GPT-моделі;
- передача контексту діалогу;
- обробка результатів генерації;
- фільтрація відповідей;
- контроль формату відповідей.

Важливою особливістю AI-модуля є підтримка контексту діалогу. Це дозволяє GPT-чатботу враховувати попередні повідомлення користувача та забезпечувати більш природну взаємодію.

Для підвищення якості відповідей у системі доцільно використовувати Retrieval-Augmented Generation (RAG). Даний підхід передбачає поєднання GPT-моделі з механізмом пошуку інформації у внутрішній базі знань університету.

База знань є одним із ключових компонентів системи та містить:

- правила вступу;
- інформацію про спеціальності;
- освітні програми;
- FAQ;
- контакти приймальної комісії;
- терміни вступної кампанії;
- вартість навчання.

Перед генерацією відповіді система виконує пошук релевантної інформації у базі знань та передає знайдені дані до GPT-моделі. Це дозволяє:

- підвищити точність відповідей;
- забезпечити актуальність інформації;
- зменшити ризик генерації помилкових даних;
- адаптувати відповіді до особливостей університету.

Для зберігання структурованих даних використовується база даних PostgreSQL. У базі даних можуть зберігатися:

- інформація про користувачів;
- історія звернень;
- журнали подій;
- адміністративні налаштування;
- контент бази знань.

PostgreSQL забезпечує:

- надійність зберігання даних;
- підтримку SQL-запитів;
- масштабованість;
- захист інформації;
- інтеграцію з Python.

Для підтримки роботи системи передбачається використання адміністративного модуля. Даний модуль призначений для управління інформаційним наповненням чатбота та контролю його функціонування.

Основними функціями адміністративного модуля є:

- оновлення бази знань;
- редагування інформації;
- перегляд статистики використання;
- моніторинг роботи системи;
- аналіз помилок;
- управління налаштуваннями чатбота.

Адміністративний інтерфейс може бути реалізований у вигляді вебпанелі з використанням HTML, CSS та JavaScript.

Архітектура системи також повинна враховувати питання безпеки та стабільності роботи. Для цього необхідно реалізувати:

- механізми авторизації адміністратора;
- захист API;
- логування подій;
- обробку помилок;
- резервування сервісів;
- контроль доступу до бази даних.

Особливу увагу необхідно приділити захисту персональних даних користувачів. Система не повинна зберігати конфіденційну інформацію без необхідності, а передача даних має здійснюватися через захищені канали зв'язку.

З огляду на можливе збільшення кількості користувачів у період вступної кампанії архітектура системи повинна забезпечувати масштабованість. Модульний підхід дозволяє:

- збільшувати продуктивність окремих компонентів;
- масштабувати backend-сервер;
- оптимізувати обробку запитів;
- підключати додаткові сервіси.

Для забезпечення стабільної роботи системи також необхідно передбачити механізми моніторингу та логування. Це дозволить:

- виявляти помилки;
- аналізувати навантаження;
- контролювати продуктивність;
- оцінювати ефективність роботи чатбота.

Під час проектування архітектури важливим є врахування можливості подальшого розвитку системи. У перспективі GPT-чатбот може бути доповнений:

- голосовою взаємодією;

- інтеграцією з вебсайтом університету;
- підтримкою кількох мов;
- рекомендаційною системою;
- інтеграцією з електронним кабінетом вступника.

Перевагами запропонованої архітектури є:

- модульність;
- масштабованість;
- підтримка AI-технологій;
- гнучкість інтеграції;
- можливість розширення функціоналу;
- зручність супроводу системи.

Таким чином, у результаті проєктування архітектури GPT-чатбота було визначено основні компоненти системи, принципи їх взаємодії та механізми обробки інформації. Запропонована архітектура забезпечує підтримку AI-функціоналу, інтеграцію з Telegram, роботу з базою знань та можливість масштабування системи відповідно до потреб закладу вищої освіти.

2.6 Висновки до розділу 2

У другому розділі було виконано дослідження та проєктування GPT-чатбота підтримки вступників університету. Сформульовано постановку задачі розробки системи та визначено основні проблеми інформаційної підтримки вступників, пов'язані з великим обсягом інформації, високим навантаженням на приймальні комісії та необхідністю забезпечення оперативного доступу до актуальних даних.

Проведено аналіз функціональних і нефункціональних вимог до системи, що дозволило визначити перелік основних функцій чатбота, вимоги до швидкодії, доступності, безпеки та масштабованості. Для опису потреб

користувачів сформовано набір user stories, які стали основою для подальшого проєктування системи.

Виконано аналіз та обґрунтування вибору технологій реалізації GPT-чатбота. Визначено доцільність використання Telegram Bot API, FastAPI, PostgreSQL, OpenAI API та технології Retrieval-Augmented Generation для забезпечення актуальності відповідей. Розроблено функціональну модель роботи системи та запропоновано архітектуру, що включає користувацький інтерфейс, Telegram-бота, серверну частину, AI-модуль, базу знань, базу даних та адміністративний модуль.

Також проведено аналіз основних ризиків і обмежень реалізації системи, пов'язаних із точністю відповідей штучного інтелекту, актуальністю інформації, інформаційною безпекою та залежністю від зовнішніх сервісів. Отримані результати створюють основу для подальшого планування та управління проєктом розробки GPT-чатбота.

РОЗДІЛ 3 УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ GPT-ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ

3.1 Ініціація та характеристика проєкту

Ініціація є початковим етапом життєвого циклу проєкту, у межах якого визначаються основні цілі, характеристики, обмеження та очікувані результати реалізації проєкту. На даному етапі формується загальне бачення продукту проєкту, визначаються ключові зацікавлені сторони, оцінюються потреби замовника та обґрунтовується доцільність реалізації проєкту.

Проєкт розробки GPT-чатбота підтримки вступників університету спрямований на автоматизацію процесу інформаційного консультування вступників та підвищення ефективності цифрової взаємодії між університетом і абітурієнтами. Актуальність проєкту обумовлена необхідністю цифровізації освітніх сервісів, зростанням кількості онлайн-комунікацій та потребою у швидкому наданні актуальної інформації під час вступної кампанії.

Основною проблемою, яку вирішує проєкт, є значне навантаження на приймальну комісію університету у період вступної кампанії. Велика кількість однотипних звернень щодо правил вступу, спеціальностей, вартості навчання та термінів подачі документів ускладнює оперативне надання консультацій та потребує залучення значних людських ресурсів.

Використання GPT-чатбота дозволить автоматизувати значну частину консультаційної підтримки вступників, забезпечити цілодобовий доступ до інформації та покращити якість цифрових сервісів університету.

Метою проєкту є розробка та впровадження GPT-чатбота підтримки вступників університету для автоматизації процесу інформаційного консультування та підвищення ефективності взаємодії зі вступниками.

Основним продуктом проєкту є AI-орієнтована інформаційна система у вигляді GPT-чатбота, інтегрованого з Telegram та базою знань університету.

До основних функцій продукту належать:

- підтримка текстового діалогу природною мовою;
- консультації щодо вступної кампанії;
- надання інформації про спеціальності;
- відповіді на типові запитання;
- пошук інформації у базі знань;
- підтримка Telegram-взаємодії;
- адміністрування інформаційного наповнення системи.

Очікуваними результатами реалізації проєкту є:

- автоматизація консультаційної підтримки вступників;
- зменшення навантаження на працівників приймальної комісії;
- скорочення часу обробки запитів;
- підвищення доступності інформації;
- покращення цифрової інфраструктури університету;
- підвищення рівня задоволеності користувачів.

Для успішної реалізації проєкту необхідно визначити основні зацікавлені сторони (stakeholders), які можуть впливати на проєкт або бути зацікавленими у його результатах.

Основні стейкхолдери проєкту наведено у таблиці 3.1.

Таблиця 3.1 – Основні стейкхолдери проєкту

Стейкхолдер	Інтерес до проєкту
Адміністрація університету	Підвищення рівня цифровізації
Приймальна комісія	Зменшення навантаження
Вступники	Швидкий доступ до інформації
Батьки вступників	Отримання консультацій
Команда розробки	Реалізація програмного продукту
Адміністратор системи	Підтримка функціонування чатбота

Одним із важливих елементів ініціації проєкту є визначення критеріїв успіху. Для проєкту GPT-чатбота основними критеріями успішності є:

- стабільна робота системи;
- коректність відповідей чатбота;
- підтримка актуальної інформації;

- інтеграція з Telegram;
- відповідність функціональних можливостей вимогам користувачів;
- дотримання термінів та бюджету проєкту.

У процесі ініціації також необхідно визначити основні обмеження проєкту.

До основних обмежень належать:

- обмежений бюджет;
- залежність від OpenAI API;
- необхідність регулярного оновлення бази знань;
- обмежені часові рамки реалізації;
- потреба у стабільному інтернет-з'єднанні.

Крім того, необхідно враховувати припущення проєкту, які приймаються як умови його реалізації:

- доступність OpenAI API;
- наявність необхідних технічних ресурсів;
- стабільна робота Telegram API;
- можливість отримання актуальної інформації від університету;
- наявність команди розробки відповідної кваліфікації.

Для проєкту розробки GPT-чатбота доцільно використовувати ітеративний життєвий цикл із застосуванням Agile-підходу та Scrum-методології. Це пояснюється високим рівнем невизначеності AI-проєктів, необхідністю поступового вдосконалення функціоналу та потребою у регулярному тестуванні системи.

Життєвий цикл проєкту включає такі основні етапи:

1. Ініціація проєкту.
2. Аналіз вимог.
3. Проєктування системи.
4. Розробка backend та Telegram-бота.
5. Інтеграція GPT API.
6. Формування бази знань.
7. Тестування системи.

8. Впровадження та запуск.

9. Супровід і підтримка.

Для організації управління проєктом формується команда, до складу якої входять:

- Project Manager;
- Backend Developer;
- AI Engineer;
- Business Analyst;
- QA Engineer.

Project Manager відповідає за планування, координацію та контроль виконання проєкту. Backend Developer реалізує серверну частину системи та інтеграцію компонентів. AI Engineer забезпечує інтеграцію GPT-моделі та налаштування AI-функціоналу. Business Analyst формує вимоги до системи та взаємодіє із замовником. QA Engineer відповідає за тестування системи та контроль якості програмного продукту.

Важливим аспектом ініціації є визначення очікуваних переваг від впровадження системи. Реалізація GPT-чатбота дозволить:

- скоротити навантаження на персонал приймальної комісії;
- забезпечити цілодобову підтримку вступників;
- підвищити швидкість надання консультацій;
- покращити цифровий імідж університету;
- автоматизувати обробку типових запитів.

Таким чином, у межах етапу ініціації було визначено основні характеристики проєкту розробки GPT-чатбота підтримки вступників університету, сформульовано мету, визначено зацікавлені сторони, критерії успіху, обмеження та очікувані результати проєкту. Отримані результати створюють основу для подальшого планування структури робіт, ресурсів, розкладу та управління реалізацією проєкту.

3.2 Формування структури робіт проєкту

Одним із ключових етапів планування IT-проєкту є формування структури робіт проєкту (Work Breakdown Structure, WBS). WBS являє собою ієрархічну декомпозицію проєкту на окремі роботи та задачі, необхідні для досягнення поставлених цілей. Формування структури робіт дозволяє деталізувати процес реалізації проєкту, визначити обсяг робіт, розподілити відповідальність між учасниками команди та забезпечити основу для подальшого планування розкладу, ресурсів і бюджету.

Для проєкту розробки GPT-чатбота підтримки вступників університету використовується продуктово-орієнтований підхід до побудови WBS. Це дозволяє структурувати роботи відповідно до функціональних компонентів системи та етапів реалізації програмного продукту.

Процес декомпозиції проєкту здійснюється поетапно – від загальних етапів до конкретних задач. На верхньому рівні WBS проєкт поділяється на основні фази життєвого циклу: Ініціація проєкту, Аналіз та проєктування, Розробка системи, Інтеграція AI-функціоналу, Формування бази знань, Тестування, Впровадження, Завершення проєкту.

Кожен етап містить набір детальніших робіт, необхідних для реалізації функціональних можливостей GPT-чатбота.

На етапі ініціації проєкту виконуються роботи, пов'язані з визначенням цілей, формуванням команди та організацією управління проєктом. До складу робіт даного етапу входять: визначення цілей проєкту; формування команди; визначення стейкхолдерів; підготовка Project Charter; визначення обмежень та ризиків.

Етап аналізу та проєктування включає роботи, пов'язані зі збором вимог та розробкою проєктних рішень. Основними задачами даного етапу є: аналіз вимог користувачів; формування функціональних вимог; формування нефункціональних вимог; моделювання бізнес-процесів; проєктування архітектури системи; вибір технологій реалізації.

Одним із найбільш трудомістких етапів є розробка системи. Даний етап включає реалізацію серверної частини, Telegram-бота та інтеграцію окремих компонентів системи.

До складу робіт етапу розробки входять: налаштування середовища розробки; створення backend-сервера; реалізація REST API; розробка Telegram-бота; інтеграція Telegram API; створення адміністративного модуля; реалізація авторизації; реалізація логування подій.

Окремий етап присвячений інтеграції AI-функціоналу. Основними задачами є: інтеграція OpenAI API; налаштування GPT-моделі; реалізація обробки запитів; підтримка контексту діалогу; реалізація механізму генерації відповідей; інтеграція RAG-підходу.

Для забезпечення актуальності інформації необхідно сформувати базу знань системи. Роботи даного етапу включають: збір інформації щодо вступної кампанії; формування структури бази знань; наповнення FAQ; підготовку інформації про спеціальності; формування даних щодо вартості навчання; оновлення правил вступу.

Важливим етапом є тестування системи, яке дозволяє перевірити працездатність чатбота та виявити можливі помилки. До основних робіт даного етапу належать: функціональне тестування; тестування Telegram-інтеграції; тестування GPT-відповідей; перевірка роботи бази знань; навантажувальне тестування; виправлення помилок.

Етап впровадження передбачає: розгортання системи; налаштування серверного середовища; запуск чатбота; тестову експлуатацію; підготовку інструкцій користувача; навчання адміністратора системи.

На завершальному етапі проєкту виконуються: оцінка результатів; аналіз досягнення цілей; підготовка звітної документації; завершення проєкту.

У табличному вигляді структуру робіт проєкту наведено в додатку А.

Для управління виконанням робіт у межах Agile-підходу формується Product Backlog проєкту. Product Backlog являє собою перелік функціональних можливостей системи, реалізація яких планується у межах проєкту.

До основних елементів backlog належать:

- авторизація адміністратора;
- підтримка Telegram;
- GPT-генерація відповідей;
- підтримка FAQ;
- база знань;
- історія повідомлень;
- статистика використання;
- адміністрування контенту.

На основі Product Backlog у процесі реалізації проєкту формуються Sprint Backlog для окремих спринтів Scrum.

Перевагою побудованої структури робіт є її модульність та відповідність архітектурі системи. Це дозволяє організувати паралельне виконання окремих задач та підвищити ефективність управління проєктом.

Таким чином, у результаті формування структури робіт проєкту було виконано декомпозицію процесу розробки GPT-чатбота підтримки вступників на окремі етапи та задачі. Побудована WBS створює основу для подальшого планування ресурсів, розкладу, бюджету та організації управління реалізацією проєкту.

Ієрархічна структура робіт проєкту використовується для формалізації загального обсягу робіт та визначення основних deliverables проєкту. Водночас безпосередня організація процесу реалізації GPT-чатбота здійснюється із застосуванням Scrum-підходу.

На основі сформованої WBS формується Product Backlog проєкту, який містить user stories та функціональні вимоги системи. Подальша реалізація функціоналу виконується ітеративно у межах Scrum-спринтів.

Такий підхід дозволяє поєднати переваги структурованого планування проєкту з гнучкістю Agile-методології та забезпечити ефективне управління AI-орієнтованим IT-проєктом.

3.3 Планування команди та ресурсів проєкту

Одним із важливих етапів управління IT-проєктом є планування команди та ресурсного забезпечення. Ефективна організація людських, технічних та програмних ресурсів безпосередньо впливає на терміни реалізації проєкту, якість програмного продукту та досягнення поставлених цілей. Для проєкту розробки GPT-чатбота підтримки вступників університету особливого значення набуває формування команди, здатної забезпечити реалізацію AI-функціоналу, інтеграцію компонентів системи та підтримку сучасних підходів до управління проєктами.

Планування ресурсів передбачає визначення складу команди, розподіл ролей та відповідальності, оцінку потреб у технічних ресурсах, визначення програмного забезпечення, організацію взаємодії учасників проєкту.

Для реалізації GPT-чатбота формується невелика Scrum-команда, до складу якої входять Project Manager, Business Analyst, Backend Developer, AI Engineer та QA Engineer.

Невеликий склад команди пояснюється обмеженням масштабом проєкту та використанням сучасних AI-сервісів, які дозволяють скоротити обсяг ручної розробки окремих функціональних компонентів.

Project Manager відповідає за організацію управління проєктом, планування робіт, координацію команди та контроль виконання завдань. Основними функціями Project Manager є:

- формування плану проєкту;
- контроль термінів виконання;
- управління ризиками;
- організація Scrum-процесів;
- взаємодія зі стейкхолдерами;
- контроль виконання backlog.

Business Analyst здійснює аналіз вимог користувачів та формує функціональні вимоги до системи. До основних задач аналітика належать:

- збір вимог;
- формування user stories;
- опис бізнес-процесів;
- моделювання функціоналу системи;
- взаємодія із замовником.

Backend Developer відповідає за реалізацію серверної частини системи та інтеграцію її компонентів. Основними задачами розробника є:

- створення backend-сервера;
- реалізація REST API;
- інтеграція Telegram API;
- робота з базою даних;
- реалізація бізнес-логіки системи.

AI Engineer забезпечує реалізацію AI-функціоналу системи. До його задач належать:

- інтеграція OpenAI API;
- налаштування GPT-моделі;
- реалізація RAG-підходу;
- оптимізація AI-відповідей;
- тестування AI-функціоналу.

QA Engineer відповідає за контроль якості програмного продукту та тестування системи. Основними задачами тестувальника є:

- функціональне тестування;
- перевірка AI-відповідей;
- тестування Telegram-бота;
- навантажувальне тестування;
- фіксація помилок.

Склад команди проєкту наведено у таблиці 3.2.

Для ефективного управління відповідальністю між учасниками проєкту доцільно використовувати матрицю RACI. RACI дозволяє визначити відповідального за виконання роботи (Responsible); особу, яка приймає рішення

(Accountable); консультантів (Consulted); осіб, яких інформують (Informed). Фрагмент RACI-матриці наведено у таблиці 3.3.

Таблиця 3.2 – Команда проєкту

Роль	Основні функції
Project Manager	Управління проєктом та Scrum-процесами
Business Analyst	Аналіз вимог та user stories
Backend Developer	Реалізація backend та API
AI Engineer	Інтеграція GPT та AI-функціоналу
QA Engineer	Тестування та контроль якості

Таблиця 3.3 – Фрагмент RACI-матриці проєкту

Робота	PM	BA	Backend Dev	AI Engineer	QA
Аналіз вимог	A	R	I	I	I
Проектування системи	A	R	C	C	I
Розробка backend	I	I	R	C	I
Інтеграція GPT API	I	I	C	R	I
Тестування системи	I	I	C	C	R
Управління проєктом	R	I	I	I	I

Для реалізації GPT-чатбота необхідно також визначити технічні ресурси проєкту. До основних технічних ресурсів належать:

- серверне середовище;
- персональні комп'ютери розробників;
- доступ до OpenAI API;
- доступ до Telegram Bot API;
- система управління базами даних;
- хмарні сервіси.

Для серверної частини системи доцільно використовувати хмарний VPS-сервер, який забезпечить:

- постійний доступ до системи;
- підтримку backend-сервера;
- роботу Telegram-бота;
- інтеграцію з базою даних.

До програмних ресурсів проєкту належать:

- Python;
- FastAPI;
- PostgreSQL;
- OpenAI API;
- LangChain;
- GitHub;
- Jira;
- Project Libre.

Python використовується як основна мова програмування серверної частини системи. FastAPI застосовується для створення backend API. PostgreSQL використовується як система управління базами даних. OpenAI API забезпечує AI-функціонал системи. LangChain використовується для реалізації RAG-підходу та інтеграції бази знань із GPT-моделлю.

Для організації командної роботи та управління задачами доцільно використовувати Jira або Trello. Дані системи дозволяють формувати backlog; управляти спринтами; відстежувати виконання задач; контролювати прогрес проєкту.

Для планування розкладу та ресурсів проєкту використовується Microsoft Project. Технічні та програмні ресурси проєкту наведено у таблиці 3.4.

Таблиця 3.4 – Ресурси проєкту

Ресурс	Призначення
Python	Backend-розробка
FastAPI	Реалізація API
PostgreSQL	База даних
OpenAI API	AI-функціонал
Telegram Bot API	Взаємодія з користувачами
LangChain	Реалізація RAG
GitHub	Контроль версій
Jira/Trello	Управління задачами
Project Libre	Планування проєкту

Важливим елементом планування ресурсів є оцінка завантаження команди. Оскільки проєкт реалізується за Scrum-підходом, робота організовується у межах

спринтів. Для кожного спринту формується Sprint Backlog та визначається перелік задач для окремих учасників команди.

Використання Scrum дозволяє гнучко розподіляти задачі, оперативно реагувати на зміни, контролювати навантаження та забезпечувати регулярний моніторинг прогресу.

Під час планування ресурсів також необхідно враховувати ризики, зокрема недостатню кваліфікацію учасників, перевантаження команди, затримки виконання задач та технічні проблеми інтеграції.

Для мінімізації ризиків доцільно використовувати Agile-підхід, організовувати регулярні Scrum-зустрічі, проводити тестування функціоналу та здійснювати моніторинг виконання задач.

Таким чином, у результаті планування команди та ресурсів проєкту було визначено склад Scrum-команди, ролі учасників, технічні та програмні ресурси, необхідні для реалізації GPT-чатбота підтримки вступників університету. Сформована структура ресурсного забезпечення створює основу для подальшого планування розкладу, бюджету та організації процесу управління проєктом.

3.4 Планування розкладу проєкту

Одним із ключових процесів управління IT-проєктом є планування розкладу. Даний процес передбачає визначення послідовності виконання робіт, оцінку тривалості задач, формування календарного плану та встановлення контрольних точок проєкту. Якісне планування розкладу дозволяє забезпечити своєчасне виконання робіт, ефективне використання ресурсів та контроль реалізації проєкту.

Для проєкту розробки GPT-чатбота підтримки вступників університету планування розкладу здійснюється на основі сформованої структури робіт (WBS) та з урахуванням Scrum-підходу до організації процесу розробки.

Основними задачами планування розкладу є:

- визначення послідовності виконання робіт;

- оцінка тривалості задач;
- встановлення залежностей між роботами;
- формування календарного плану;
- визначення контрольних точок (milestones);
- побудова мережевого графіка та Gantt chart.

Для оцінки тривалості робіт використовується експертний підхід із врахуванням складності реалізації функціоналу системи та досвіду команди розробки. Проєкт реалізується у межах ітеративного життєвого циклу із застосуванням Scrum-методології.

Загальна тривалість проєкту становить 12 тижнів. Робота команди організовується у межах чотирьох Scrum-спринтів тривалістю 3 тижні кожен.

На початковому етапі проєкту виконуються роботи, пов'язані з аналізом вимог та проектуванням системи. Після завершення етапу аналізу здійснюється розробка backend-сервера, Telegram-бота та інтеграція GPT API. Завершальні етапи включають тестування, впровадження та запуск системи.

Для визначення послідовності виконання робіт було сформовано перелік основних задач проєкту та їх тривалість. Роботи проєкту з оцінками тривалості наведено у таблиці А.1.

Модель проєкту побудовано в середовищі Project Libre. На рисунках 3.1 та 3.2 представлено фрагменти діаграми Ганта та сітьового графіку проєкту.

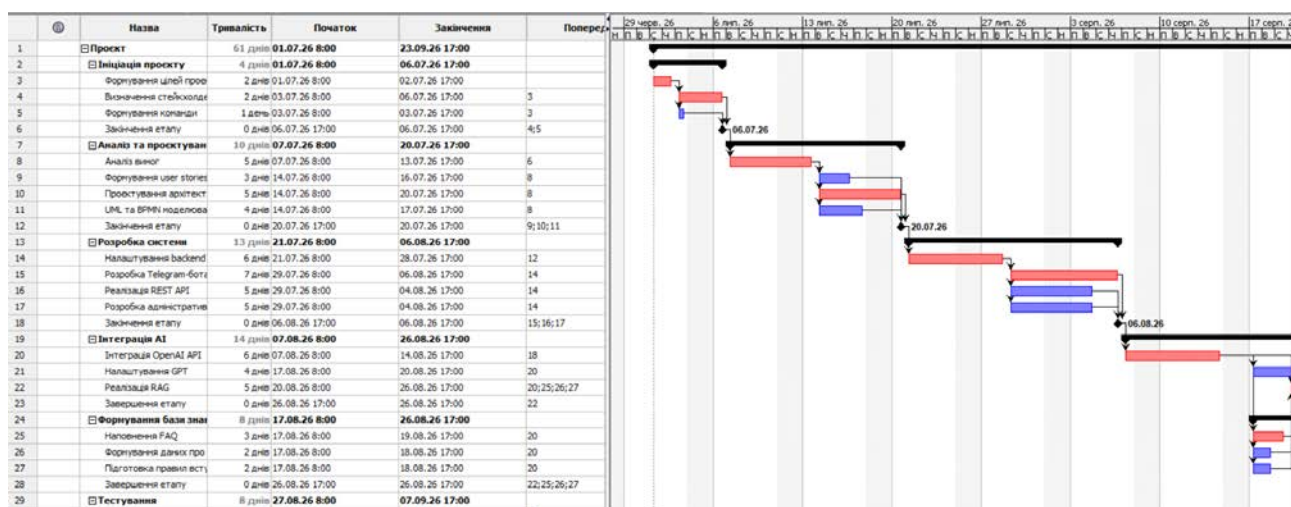


Рисунок 3.1 – Фрагмент діаграми Ганта проєкту

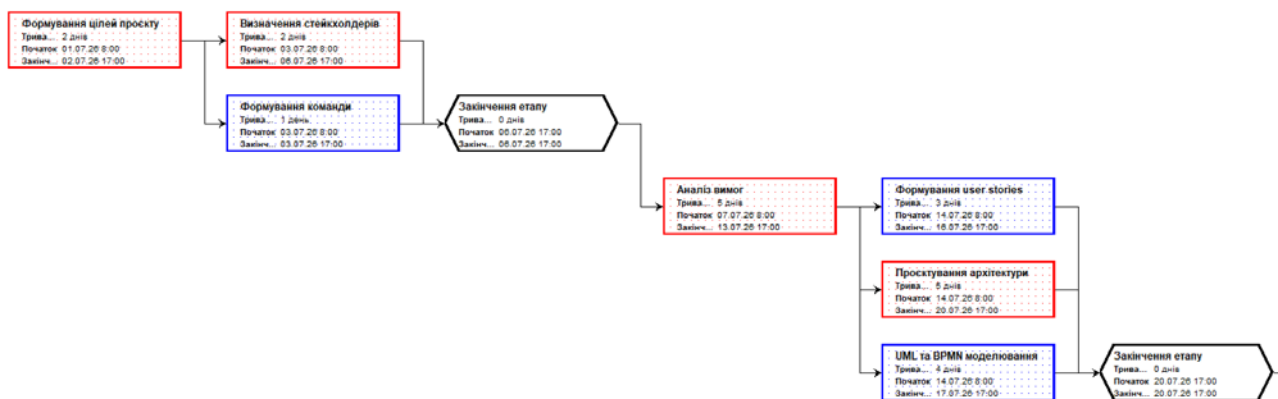


Рисунок 3.2 – Фрагмент сітьового графіку проєкту

Для контролю виконання проєкту визначаються контрольні точки (milestones). Основні milestones проєкту наведено у таблиці 3.5.

Таблиця 3.5 – Контрольні точки проєкту

Контрольна точка	Очікуваний результат
Завершення аналізу вимог	Сформовані user stories
Завершення проєктування	Архітектура системи
Завершення backend-розробки	Реалізований API
Інтеграція GPT API	Робочий AI-модуль
Завершення тестування	Готовність системи до запуску
Впровадження системи	Запуск чатбота

Таким чином, у результаті планування розкладу проєкту було визначено тривалість основних робіт, сформовано послідовність виконання задач, побудовано структуру Scrum-спринтів та визначено контрольні точки реалізації GPT-чатбота підтримки вступників університету. Сформований календарний план створює основу для подальшого планування бюджету та контролю виконання проєкту.

3.5 Планування вартості проєкту

Планування вартості є одним із ключових процесів управління проєктом, який передбачає оцінку необхідних фінансових ресурсів для реалізації запланованих робіт. Основною метою планування вартості є визначення бюджету

проєкту, оцінка витрат на ресурси та забезпечення фінансового контролю під час реалізації проєкту.

Для ІТ-проєктів основними статтями витрат є:

- оплата праці команди;
- програмне забезпечення;
- серверна інфраструктура;
- використання зовнішніх сервісів;
- тестування та супровід системи.

Для проєкту розробки GPT-чатбота підтримки вступників університету оцінка вартості виконується на основі структури робіт проєкту (WBS), тривалості виконання задач, складу команди, технічних та програмних ресурсів.

Основною складовою бюджету ІТ-проєкту є витрати на оплату праці команди розробки. Для оцінки вартості використовується погодинний підхід, при якому загальна вартість визначається на основі трудомісткості робіт та погодинної ставки учасників команди.

Оцінка трудомісткості робіт здійснюється відповідно до календарного плану проєкту. Загальна тривалість реалізації проєкту становить 12 тижнів.

Середні погодинні ставки учасників команди наведено у таблиці 3.6. Перелік робіт з оцінкою вартості їх виконання представлено в таблиці А.1.

Таблиця 3.6– Погодинні ставки учасників проєкту

Роль	Погодинна ставка, грн
Project Manager	500
Business Analyst	400
Backend Developer	550
AI Engineer	650
QA Engineer	350

Оцінка витрат на оплату праці наведена у таблиці 3.7.

Отже, витрати на команду проєкту становлять 502 000 грн.

Крім оплати праці, необхідно враховувати витрати на використання зовнішніх сервісів та програмного забезпечення. Одним із ключових ресурсів

проєкту є OpenAI API, яке використовується для реалізації AI-функціоналу чатбота.

Вартість використання OpenAI API залежить від кількості запитів, обсягу тексту, обраної GPT-моделі та інтенсивності використання системи.

Таблиця 3.7 – Витрати на оплату праці

Роль	Кількість годин	Ставка, грн	Вартість, грн
Project Manager	160	500	80 000
Business Analyst	120	400	48 000
Backend Developer	320	550	176 000
AI Engineer	240	650	156 000
QA Engineer	120	350	42 000

Витрати на використання OpenAI API залежать від обраної моделі, інтенсивності використання системи та кількості оброблених запитів. Для цілей планування проєкту витрати на використання сервісів штучного інтелекту приймаються експертним методом та включаються до складу витрат на технічну інфраструктуру. Для дипломного проєкту приймається умовна оцінка витрат на використання OpenAI API у розмірі 5 000 грн на етап розробки та тестування системи.

Також необхідно врахувати витрати на VPS-хостинг, доменне ім'я, програмні сервіси, резервне копіювання даних.

Витрати на технічну інфраструктуру наведено у таблиці 3.8.

Таблиця 3.8 – Витрати на технічні ресурси

Ресурс	Вартість, грн
OpenAI API	5 000
VPS-хостинг	4 000
Доменне ім'я	800
GitHub / Jira	Використовуються безкоштовні редакції GitHub та Jira
Резервне копіювання	1 200

Таким чином, витрати на технічну інфраструктуру становлять 11 000 грн.

Загальний бюджет проєкту наведено у таблиці 3.9.

Таблиця 3.9 – Загальний бюджет проєкту

Стаття витрат	Вартість, грн
Оплата праці команди	502 000
Технічна інфраструктура	11 000
Усього	564500

Таким чином, базовий бюджет проєкту розробки GPT-чатбота підтримки вступників університету становить 564 500 грн.

3.6 Управління ризиками проєкту

Управління ризиками є одним із ключових процесів управління IT-проєктами, метою якого є виявлення потенційних загроз для реалізації проєкту, оцінка їх впливу та розробка заходів щодо мінімізації негативних наслідків. Для AI-орієнтованих IT-проєктів, зокрема проєктів розробки GPT-чатботів, управління ризиками має особливе значення через високий рівень технологічної невизначеності, залежність від зовнішніх сервісів та складність інтеграції AI-компонентів.

Основні ризики проєкту наведено у таблиці 3.10.

Таблиця 3.10 – Реєстр ризиків проєкту

Ризик	Ймовірність	Вплив	Рівень ризику	Спосіб реагування
Недоступність OpenAI API	Середня	Високий	Високий	Кешування та резервні сценарії
Неточність AI-відповідей	Висока	Високий	Високий	Retrieval та тестування
Зміна вимог	Висока	Середній	Середній	Agile/Scrum
Перевищення бюджету	Середня	Середній	Середній	Контроль витрат
Перевантаження системи	Середня	Високий	Високий	Масштабування
Витік даних	Низька	Високий	Середній	Захист доступу
Помилки інтеграції	Середня	Середній	Середній	Поетапне тестування

3.7 Організація Agile/Scrum управління проєктом

Сучасні IT-проєкти характеризуються високим рівнем невизначеності, постійною зміною вимог та необхідністю швидкої адаптації до змін. Особливо це стосується AI-орієнтованих систем, у яких результати роботи моделі можуть уточнюватися або змінюватися під час реалізації проєкту. У таких умовах традиційні каскадні підходи до управління проєктами не завжди забезпечують достатню гнучкість та ефективність управління. Тому для проєкту розробки GPT-чатбота підтримки вступників університету доцільно використовувати Agile-підхід із застосуванням Scrum-методології.

У Scrum-підході проєкт поділяється на серію коротких спринтів тривалістю 2–4 тижні. Для проєкту GPT-чатбота передбачається використання чотирьох спринтів тривалістю 3 тижні.

Структуру Scrum-спринтів наведено у таблиці 3.11.

Таблиця 3.11 – Scrum-спринти проєкту

Спринт	Основні задачі
Sprint 1	Аналіз вимог, user stories, проєктування системи
Sprint 2	Розробка backend та Telegram-бота
Sprint 3	Інтеграція GPT API та реалізація RAG
Sprint 4	Тестування, впровадження та документація

Кожен спринт завершується формуванням робочого інкременту системи, який може бути протестований або продемонстрований замовнику.

Організація Scrum-процесу передбачає наявність визначених ролей – Product Owner, Scrum Master та Scrum Team.

У межах проєкту GPT-чатбота роль Product Owner виконує представник замовника або Business Analyst, який відповідає за:

- формування Product Backlog;
- пріоритезацію вимог;
- визначення функціональних потреб системи;
- взаємодію із зацікавленими сторонами.

Scrum Master відповідає за організацію Scrum-процесів та забезпечення ефективної роботи команди. Основними функціями Scrum Master є:

- організація Scrum-зустрічей;
- контроль дотримання Scrum-підходу;
- усунення перешкод;
- координація взаємодії команди.

У межах даного проєкту роль Scrum Master виконує Project Manager.

Scrum Team включає:

- Backend Developer;
- AI Engineer;
- QA Engineer;
- Business Analyst.

Команда є самоорганізованою та спільно відповідає за реалізацію функціоналу системи.

Основним артефактом Scrum є Product Backlog – перелік функціональних можливостей системи, які необхідно реалізувати у межах проєкту. Елементи backlog формуються у вигляді user stories (табл. 2.3).

На початку кожного спринту проводиться Sprint Planning – зустріч команди, під час якої визначаються задачі спринту, оцінюється трудомісткість, формується Sprint Backlog, визначаються цілі ітерації.

Sprint Backlog являє собою перелік задач, які команда повинна реалізувати протягом спринту.

У процесі реалізації проєкту проводяться щоденні Scrum-зустрічі (Daily Scrum). Тривалість зустрічі становить приблизно 15 хвилин. Під час Daily Scrum учасники відповідають на три основні питання: Яку роботу було виконано? Яка робота планується? Які існують перешкоди?

Daily Scrum дозволяє контролювати прогрес, виявляти проблеми, координувати роботу команди та забезпечувати прозорість процесу розробки.

Після завершення спринту проводиться Sprint Review, під час якого команда демонструє реалізований функціонал та отримує зворотний зв'язок.

Для проєкту GPT-чатбота результатами Sprint Review можуть бути:

- демонстрація Telegram-бота;
- тестування GPT-відповідей;
- перевірка роботи бази знань;
- оцінка функціоналу адміністрування.

Після Sprint Review команда проводить Sprint Retrospective – зустріч для аналізу ефективності роботи спринту та визначення можливостей покращення процесів.

Retrospective дозволяє виявляти проблеми взаємодії, аналізувати ефективність Scrum, покращувати організацію роботи та підвищувати продуктивність команди.

Для управління задачами та backlog доцільно використовувати Jira або Trello.

Важливою перевагою Agile/Scrum є можливість реалізації MVP (Minimum Viable Product). У межах перших спринтів команда може створити базову версію чатбота, після чого поступово розширювати функціональні можливості системи.

Таким чином, організація управління проєктом із використанням Agile/Scrum забезпечує гнучкість реалізації GPT-чатбота підтримки вступників університету, дозволяє ефективно управляти змінами, поступово вдосконалювати AI-функціонал та забезпечувати постійний контроль якості програмного продукту.

3.8 Висновки до розділу 3

У третьому розділі було виконано комплексне планування проєкту розробки GPT-чатбота підтримки вступників університету. Визначено основні характеристики проєкту, його цілі, результати та зацікавлені сторони. Сформовано ієрархічну структуру робіт, яка дозволила деталізувати зміст проєкту та встановити взаємозв'язки між окремими роботами.

Виконано планування команди та ресурсів проєкту, визначено ролі учасників і необхідні технічні ресурси для реалізації системи. На основі структури робіт сформовано календарний план проєкту, визначено тривалість виконання робіт, контрольні точки та організацію робіт у межах Scrum-спринтів.

Проведено оцінювання вартості проєкту та сформовано бюджет, що враховує витрати на оплату праці команди, використання програмних сервісів і резерв на управління ризиками. Також виконано ідентифікацію основних ризиків проєкту та розроблено заходи щодо їх мінімізації.

Особливу увагу приділено організації Agile/Scrum управління проєктом, що дозволяє забезпечити гнучкість процесу розробки, адаптацію до змін вимог та поступове вдосконалення функціональних можливостей системи. Отримані результати підтверджують доцільність використання сучасних підходів до управління AI-орієнтованими IT-проєктами.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне завдання управління проєктом розробки GPT-чатбота підтримки вступників університету в умовах цифрової трансформації закладів вищої освіти.

У процесі виконання роботи було проаналізовано особливості цифровізації університетів, сучасні інформаційні системи підтримки вступників, технології штучного інтелекту та підходи до управління IT-проєктами. Встановлено, що використання GPT-технологій є перспективним напрямом автоматизації консультаційної підтримки вступників та розвитку цифрових сервісів університету.

У результаті дослідження сформовано функціональні та нефункціональні вимоги до системи, визначено основні сценарії взаємодії користувачів із чатботом та обґрунтовано вибір технологій реалізації. Розроблено архітектуру GPT-чатбота, яка забезпечує інтеграцію AI-модуля, бази знань, Telegram-платформи та серверної інфраструктури.

У роботі виконано планування проєкту розробки системи, сформовано структуру робіт, календарний план, склад команди та бюджет проєкту. Проведено аналіз ризиків і розроблено заходи щодо їх мінімізації. Для організації процесу розробки обґрунтовано використання Agile/Scrum-підходу, який забезпечує гнучкість управління та адаптацію до змін вимог.

Практичним результатом роботи є розроблений проєкт створення GPT-чатбота підтримки вступників університету, який може бути використаний як основа для реалізації реальної інформаційної системи. Впровадження такого рішення дозволить підвищити якість консультаційної підтримки вступників, забезпечити оперативний доступ до актуальної інформації та зменшити навантаження на працівників приймальної комісії.

Отже, поставлену мету роботи досягнуто, а всі визначені завдання виконано в повному обсязі.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Schwab K. The Fourth Industrial Revolution. Geneva : World Economic Forum, 2016. 172 p.
2. Vial G. Understanding digital transformation: A review and a research agenda. The Journal of Strategic Information Systems. 2019. Vol. 28, № 2. P. 118–144.
3. Westerman G., Bonnet D., McAfee A. Leading Digital: Turning Technology into Business Transformation. Boston : Harvard Business Review Press, 2014. 256 p.
4. Benavides L. M. C., Arias J. A. T., Serna M. D. A., Bedoya J. W. B., Burgos D. Digital Transformation in Higher Education Institutions: A Systematic Literature Review. Sensors. 2020. Vol. 20, № 11. Article 3291. DOI: 10.3390/s20113291.
5. Биков В. Ю., Лещенко М. П. Цифрова гуманістична педагогіка відкритої освіти. Теорія і практика управління соціальними системами. 2016. № 4. С. 115–130.
6. Про вищу освіту : Закон України від 01.07.2014 № 1556-VII. URL: <https://zakon.rada.gov.ua/laws/show/1556-18> (дата звернення: 13.06.2026).
7. Про освіту : Закон України від 05.09.2017 № 2145-VIII. URL: <https://zakon.rada.gov.ua/laws/show/2145-19> (дата звернення: 13.06.2026).
8. Концепція розвитку цифрової економіки та суспільства України на 2018–2020 роки : розпорядження Кабінету Міністрів України від 17.01.2018 № 67-р. URL: <https://zakon.rada.gov.ua/laws/show/67-2018-%D1%80> (дата звернення: 13.06.2026).
9. Про Цілі сталого розвитку України на період до 2030 року : Указ Президента України від 30.09.2019 № 722/2019. URL: <https://zakon.rada.gov.ua/laws/show/722/2019> (дата звернення: 13.06.2026).
10. Стратегія розвитку вищої освіти в Україні на 2022–2032 роки : розпорядження Кабінету Міністрів України від 23.02.2022 № 286-р. URL: <https://zakon.rada.gov.ua/laws/show/286-2022-%D1%80> (дата звернення: 13.06.2026).

11. Digital Education Action Plan (2021–2027). European Commission. Brussels, 2020. URL: <https://education.ec.europa.eu/focus-topics/digital-education/action-plan> (дата звернення: 13.06.2026).

12. Alfrević A. M., Mabić M., Alfrević N. Evaluating Digital Maturity in Higher Education Institutions: A Preliminary Empirical Study in the Western Balkans. World. 2025. Vol. 6, № 4. Article 130. DOI: 10.3390/world6040130.

13. Connected Student Report: Insights on Student Expectations. Salesforce. San Francisco, 2021. 28 p.

14. Вступ.ОСВІТА.UA : інформаційний портал для вступників. URL: <https://vstup.osvita.ua> (дата звернення: 13.06.2026).

15. Освіта в умовах воєнного стану в Україні: інформаційно-аналітичний збірник. Київ : Міністерство освіти і науки України, 2023. URL: <https://mon.gov.ua/static-objects/mon/sites/1/zagalna%20serednya/serpneva-konferencia/2023/22.08.2023/Inform-analytic.zbirn-Osvita.v.umovah.voyennogo.stanu-vykl.rozv.povoyen.perspekt.22.08.2023.pdf> (дата звернення: 13.06.2026).

16. Manifesto for Agile Software Development. Agile Alliance. URL: <https://agilemanifesto.org> (дата звернення: 13.06.2026).

17. The Scrum Guide. Schwaber K., Sutherland J. URL: <https://scrumguides.org> (дата звернення: 13.06.2026).

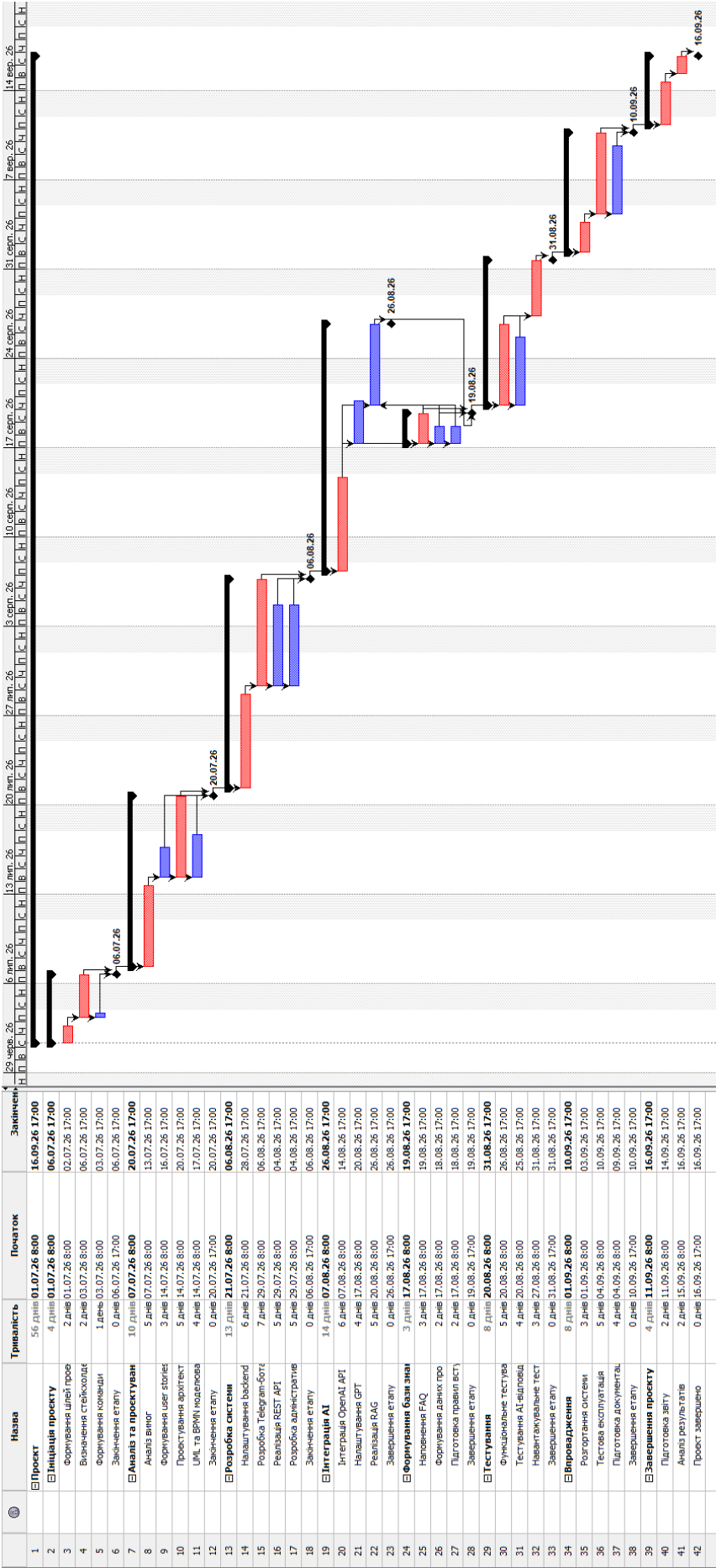
18. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 7th ed. Newtown Square, PA : Project Management Institute, 2021. 370 p.

Додаток А Структура робіт проєкту GPT-чатбота

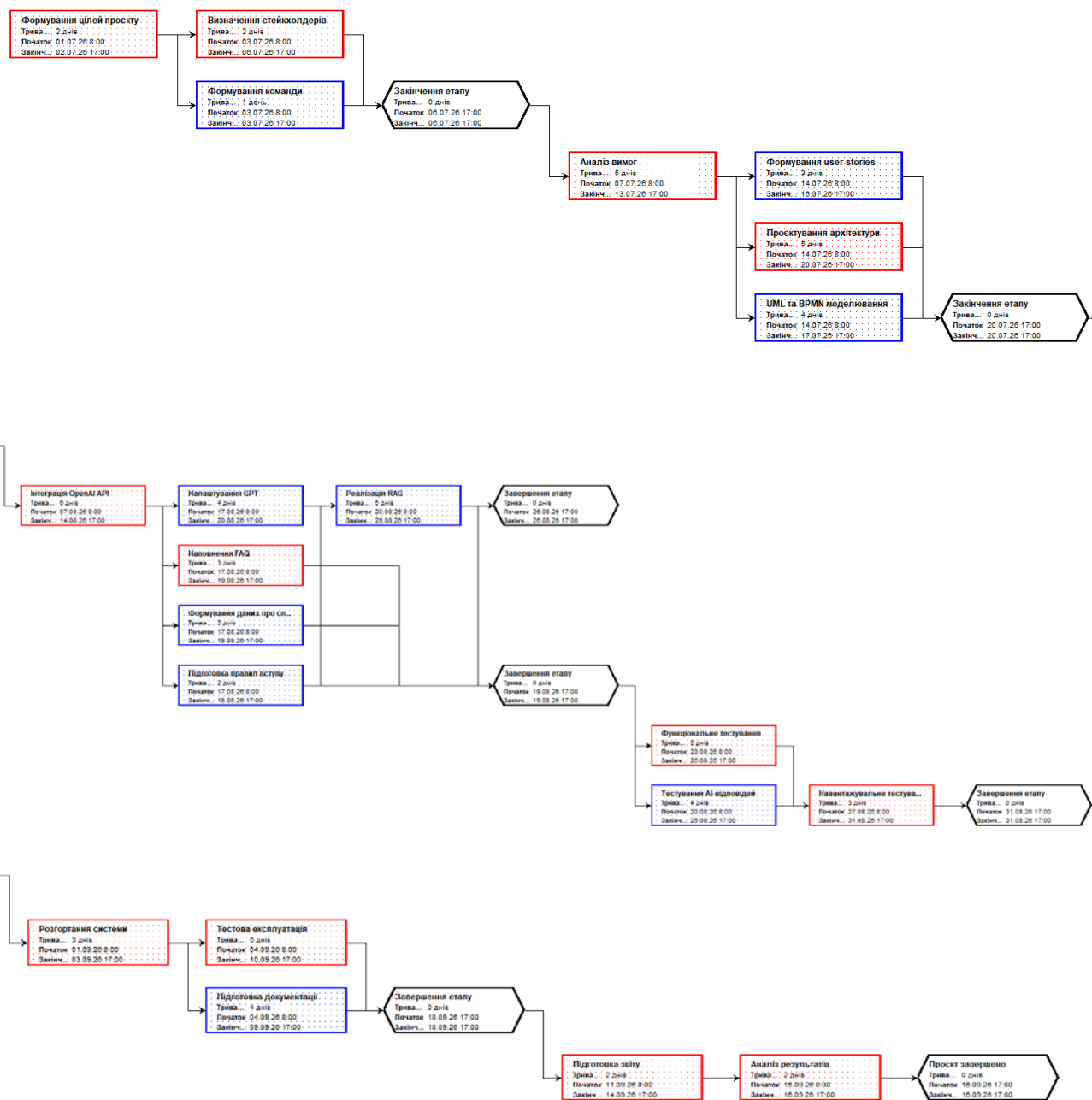
Таблиця А.1 – Структура робіт проєкту GPT-чатбота

Код WBS	Назва роботи	Тривалість, днів	Виконавець	Вартість, грн
1.1	Формування цілей проєкту	2	PM, BA	14 400
1.2	Визначення стейкхолдерів	2	PM, BA	14 400
1.3	Формування команди	1	PM	4 000
2.1	Аналіз вимог	5	BA	16 000
2.2	Формування User Stories	3	BA	9 600
2.3	Проектування архітектури	5	BA, Backend Dev, AI Engineer	64 000
2.4	UML та BPMN моделювання	4	BA	12 800
3.1	Налаштування backend	6	Backend Dev	26 400
3.2	Розробка Telegram-бота	7	Backend Dev	30 800
3.3	Реалізація REST API	5	Backend Dev	22 000
3.4	Розробка адміністративного модуля	5	Backend Dev	22 000
4.1	Інтеграція OpenAI API	6	AI Engineer	31 200
4.2	Налаштування GPT	4	AI Engineer	20 800
4.3	Реалізація RAG	5	AI Engineer	26 000
5.1	Наповнення FAQ	3	BA	9 600
5.2	Формування даних про спеціальності	2	BA	6 400
5.3	Підготовка правил вступу	2	BA	6 400
6.1	Функціональне тестування	5	QA	14 000
6.2	Тестування AI-відповідей	4	QA, AI Engineer	32 000
6.3	Навантажувальне тестування	3	QA	8 400
7.1	Розгортання системи	3	Backend Dev	13 200
7.2	Тестова експлуатація	5	QA, Backend Dev	36 000
7.3	Підготовка документації	4	PM, BA	28 800
8.1	Підготовка звіту про проєкт	2	PM	8 000
8.2	Аналіз результатів	2	PM, BA	14 400

Додаток Б Діаграма Ганта



Додаток В Сітьовий графік



Додаток Г Графічні матеріали

Харківський національний університет міського господарства
імені О.М. Бекетова
ННІ енергетичної, інформаційної та транспортної інфраструктури
кафедра управління проектами в міському господарстві і будівництві

Графічні матеріали до кваліфікаційної роботи

на тему
УПРАВЛІННЯ ПРОЄКТОМ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО
ЧАТБОТА ПІДТРИМКИ ВСТУПНИКІВ УНІВЕРСИТЕТУ

Виконав: студент 4 курсу групи гр. УІІІ(4) 2022-1
Спеціальність 122 Комп'ютерні науки
Освітня програма «Комп'ютерні науки. Управління проектами»
Будніченко Максим Юрійович
Керівник Гусєва Ю.Ю.

Мета і задачі дослідження

Метою роботи є управління проектом розробки GPT-чатбота підтримки вступників університету на основі сучасних технологій штучного інтелекту та гнучких методологій управління IT-проектами.

Для досягнення поставленої мети визначено наступні задачі:

- проаналізувати особливості цифровізації закладів вищої освіти;
- дослідити сучасні інформаційні системи підтримки вступників;
- проаналізувати технології штучного інтелекту GPT-моделі для створення чатботів;
- дослідити сучасні підходи до управління IT-проектами;
- сформулювати функціональні та нефункціональні вимоги до системи;
- розробити архітектуру GPT-чатбота;
- спроектувати функціонування інформаційної системи;
- виконати планування команди, ресурсів, розкладу та бюджету проекту;
- розробити систему управління ризиками та організацію Agile/Scrum управління проектом.



Ідеяпроєкту



У процесі вступу абітурієнти стикаються з необхідністю отримання значного обсягу інформації.

Одним із перспективних напрямів вирішення цієї задачі є використання технологій штучного інтелекту та GPT-орієнтованих чатботів.

Такі системи дозволяють автоматизувати процес інформаційної підтримки користувачів, забезпечити цілодобову доступність сервісу та значно підвищити швидкість обробки запитів.

3

Функціональні та нефункціональні вимоги системи



4

UserStories

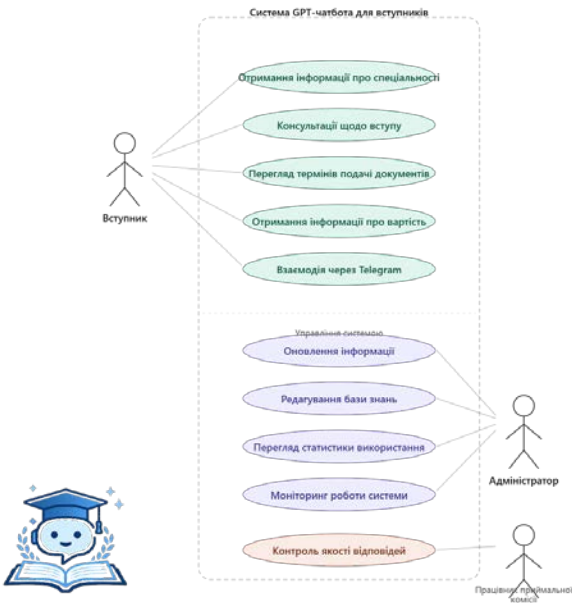
Фрагмент переліку User Stories проекту

ID	User Story	Пріоритет
US-01	Як вступник, я хочу отримувати інформацію про спеціальності університету, щоб обрати напрям навчання	Високий
US-02	Як вступник, я хочу дізнаватися про умови вступу, щоб правильно підготувати документи	Високий
US-03	Як вступник, я хочу отримувати інформацію про терміни подачі заяв, щоб не пропустити дедлайни	Високий
US-04	Як вступник, я хочу ставити запитання природною мовою, щоб швидко отримувати зрозумілі відповіді	Високий
US-05	Як вступник, я хочу користуватися чатботом через Telegram, щоб мати швидкий доступ до сервісу	Високий
US-06	Як вступник, я хочу отримувати інформацію про вартість навчання, щоб оцінити можливість вступу	Середній
US-07	Як вступник, я хочу отримувати рекомендації щодо спеціальностей, щоб обрати найбільш відповідну освітню програму	Середній
US-08	Як вступник, я хочу отримувати контакти приймальної комісії, щоб мати можливість звернутися до представників університету	Середній
US-09	Як вступник, я хочу отримувати інформацію про бюджетні місця, щоб оцінити свої шанси на вступ	Середній
US-10	Як вступник, я хочу отримувати відповіді цілодобово, щоб користуватися сервісом у будь-який час	Високий

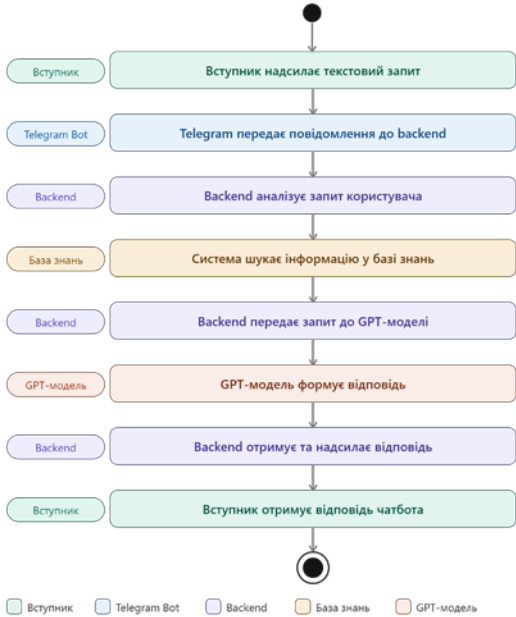
User Stories –це короткий опис функціональних потреб користувача, сформульований з його точки зору, який допомагає визначити вимоги до системи та пріоритети розробки.



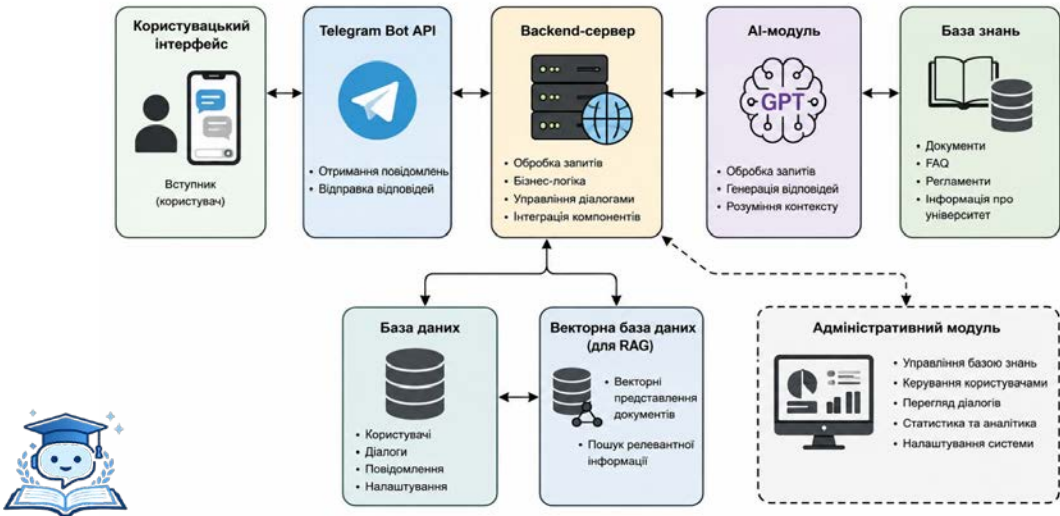
UML Use Case Diagram системи



Activity Diagram процесу взаємодії користувача із чатботом

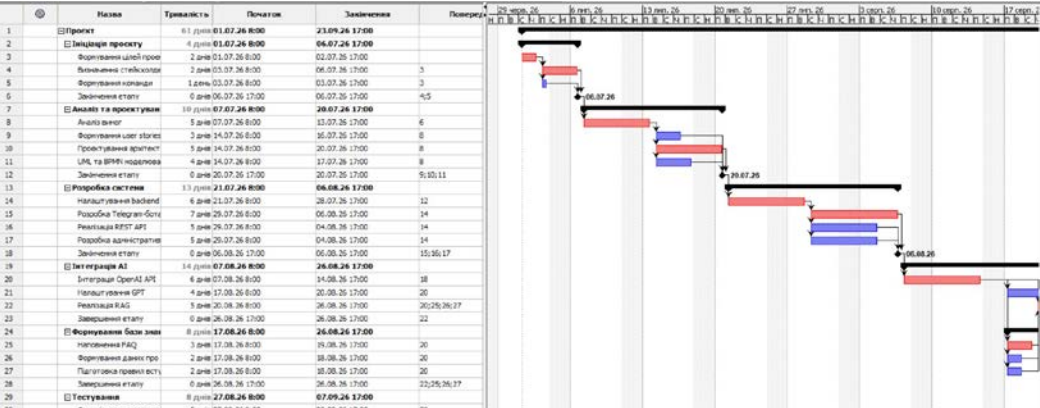


Загальна архітектура системи



7

Управління розкладом



Загальна тривалість проекту складає 61 день, початок – 01.07.26, закінчення – 23.09.26.



7

Управління вартістю

Роль	Кількість годин	Ставка, грн	Вартість, грн
Project Manager	160	500	80 000
Business Analyst	120	400	48 000
Backend Developer	320	550	176 000
AI Engineer	240	650	156 000
QA Engineer	120	350	42 000

Загальна базова вартість проекту складає 564 500грн.

Ресурс	Вартість, грн
OpenAI API	5 000
VPS-хостинг	4 000
Доменне ім'я	800
GitHub / Jira	Використовуються безкоштовні редакції GitHub та Jira
Резервне копіювання	1 200

Стаття витрат	Вартість, грн
Оплата праці команди	502 000
Технічна інфраструктура	11 000
Усього	564500



9

Управління командою проекту

Команда проекту

Роль	Основні функції
Project Manager	Управління проектом та Scrum-процесами
Business Analyst	Аналіз вимог та user stories
Backend Developer	Реалізація backend та API
AI Engineer	Інтеграція GPT та AI-функціоналу
QA Engineer	Тестування та контроль якості

Фрагмент RACI-матриці проекту

Робота	PM	BA	Backend Dev	AI Engineer	QA
Аналіз вимог	A	R	I	I	I
Проектування системи	A	R	C	C	I
Розробка backend	I	I	R	C	I
Інтеграція GPT API	I	I	C	R	I
Тестування системи	I	I	C	C	R
Управління проектом	R	I	I	I	I

R

Responsible
Виконує роботу

A

Accountable
Несе відповідальність за результат

C

Consulted
Консультується

I

Informed
Інформується



Чіткий розподіл ролей та відповідальності забезпечує ефективну взаємодію в команді та досягнення цілей проекту



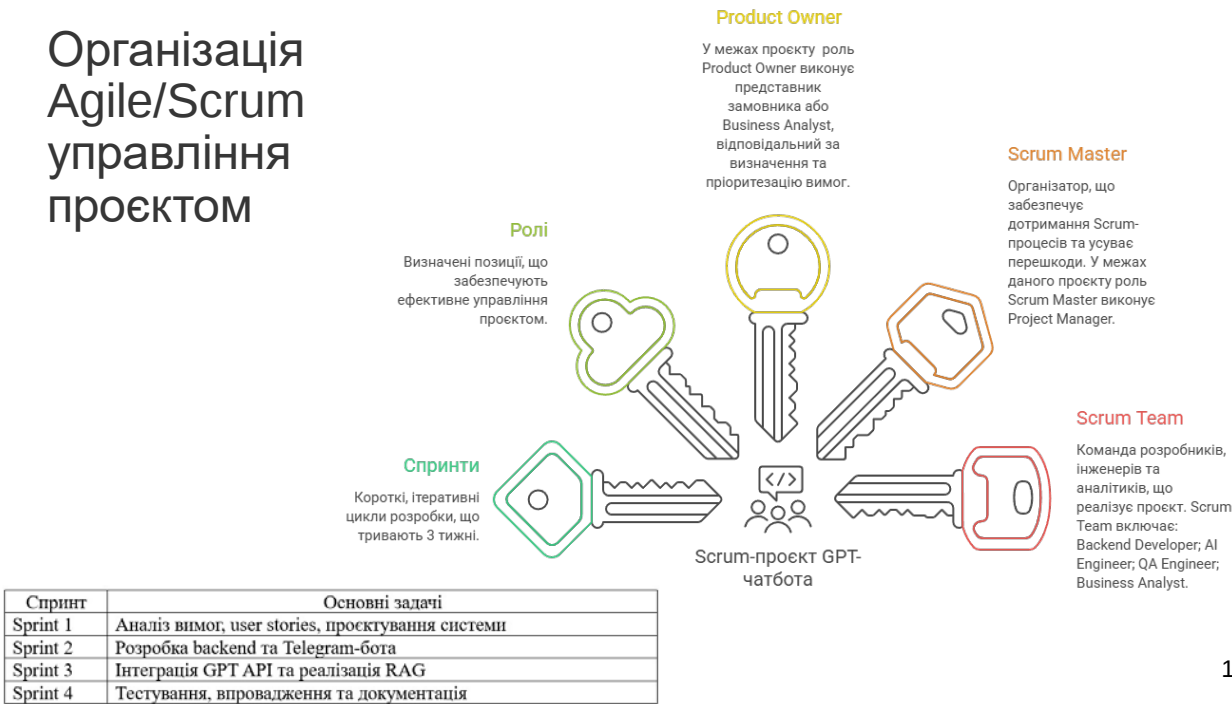
10

Реєстр ризиків

№	Ризик	Причина / Опис	Ймовірність (1–5)	Вплив (1–5)	Рівень ризику	Заходи реагування
1	 Недоступність API OpenAI	Збій або обмеження доступу до API OpenAI	3	5	Високий	Використання кешування відповідей; резервна модель; моніторинг статусу API
2	 Помилки даних бази знань	Некоректна або неактуальна інформація у БЗ	3	4	Високий	Регулярне оновлення БЗ; верифікація інформації; процес контролю якості
3	 Низька якість відповідей бота	Недостатнє навчання моделі або неточні налаштування	4	3	Середній	Тестування сценаріїв; оптимізація промптів; збір зворотного зв'язку
4	 Проблеми безпеки	Витік даних або несанкціонований доступ	2	5	Високий	Шифрування даних; контроль доступу; регулярні аудити
5	 Затримка виконання проєкту	Недооцінка обсягу робіт або зміни вимог	3	3	Середній	Гнучке планування; регулярні стандарти; контроль прогресу
6	 Перевищення бюджету	Зростання вартості API, інфраструктури або ресурсів	2	4	Середній	Моніторинг витрат; резервний бюджет; оптимізація ресурсів

11

Організація Agile/Scrum управління проєктом



12

Висновки

- Проведено аналіз цифровізації ЗВО та сучасних технологій GPT-чатботів.
- Сформовано вимоги та розроблено архітектуру системи підтримки вступників.
- Виконано планування проєкту: WBS, команда, розклад, бюджет і ризики.
- Обґрунтовано використання Agile/Scrum для управління проєктом.
- Запропоноване рішення має практичну цінність для закладів вищої освіти.



13

