

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Ганьба Яна Олегівна

УДК 68.12.02:37.016.1

КВАЛІФІКАЦІЙНА РОБОТА

**Телеграм-бот для вступної кампанії Поліського національного університету
126 «Інформаційні системи та технології»**

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Ковальчук Майя Олегівна,
Кандидат педагогічних наук, доцент кафедри КТіМС

Житомир – 2024

Висновок кафедри _____
за результатами попереднього захисту: _____

Протокол засідання кафедри _____
№ _____ від « _____ » _____ 20 _____ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)
« _____ » _____ 20 _____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

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

АНОТАЦІЯ

Ганьба Я.О. Телеграм-бот для вступної кампанії Поліського національного університету – кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 126 – Інформаційні системи та технології. – Поліський національний університет, Житомир, 2024.

Кваліфікаційна робота присвячена темі реалізації телеграм-бота для вступної кампанії Поліського національного університету. Для досягнення даної мети у теоретичній частині роботи було здійснено: аналіз предметної області, вибір технологічних засобів та визначення функціоналу бота. У практичній частині було розроблено сам телеграм-бот та його основні функції, також розроблено діаграми IDEF0, IDEF 3, проведено тестування та розгортання бота. Також створено керівництво користувачу для комфортного користування.

Ключові слова: Telegram, API, чат-бот, бот, штучний інтелект, телеграм-бот, вступна кампанія, абітурієнт.

SUMMARY

Hanba Y.O. Telegram bot for the admission campaign of Polissia National University - Qualification work as a manuscript.

Qualification work for the degree of Bachelor in the specialty 126 – Information Systems and Technologies. – Polissia National University, Zhytomyr, 2024.

The qualification work is dedicated to the implementation of a Telegram bot for the admission campaign of Polissia National University. To achieve this goal, the theoretical part of the work included an analysis of the subject area, analysis of needs and expectations, selection of technological tools, and determination of the bot's functionality. In the practical part, the Telegram bot itself and its main functions were developed, as well as IDEF0, IDEF 3, and UML diagrams were created, testing and deployment of the bot were carried out. Also, a user guide has been created for comfortable use.

Keywords: Telegram, API, chat-bot, bot, artificial intelligence, Telegram bot, admission campaign, applicant.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ	12
1.1. Аналіз інформаційних потреб і визначення предметної області дослідження	Error! Bookmark not defined.
1.2. Функціональне моделювання розробки телеграм-боту	14
1.3. Структура системи за напрямом розробки телеграм-боту для вступної кампанії Поліського національного університету	16
Висновки до першого розділу	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТУ	18
2.1. Реалізація математичної моделі персоналізації відповідей	18
2.2. Формалізація моделі системи телеграм-боту для вступної кампанії Поліського національного університету	20
2.3. Проектування інтерфейсу та реалізація функціоналу	21
Висновки до другого розділу	25
РОЗДІЛ 3. ТЕСТУВАННЯ ГРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ	26
3.1. Інструкція користувачу телеграм-боту для вступної кампанії Поліського національного університету.....	26
3.2. Тестування телеграм-боту та аналіз результатів	27
Висновки до третього розділу	30
ВИСНОВОК.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDEF – Integrated DEFinition;

DFD – Data Flow Diagram;

ІІІ – штучний інтелект;

CF – collaborative filtering;

NLP – Natural language processing;

API – Application Programming Interface;

UML – Unified Modeling Language.

ВСТУП

Питання комунікації закладів вищої освіти з абітурієнтами, студентами та батьками регулюється рядом законодавчих актів, зокрема – Закон України «Про освіту» у Статті 50 зазначається, що заклади освіти повинні забезпечувати вільний доступ до інформації про свою діяльність, у тому числі: правила прийому на навчання; умови проживання в гуртожитках; контактні дані керівництва та інших підрозділів тощо. А у 51 статті цього закону, чітко окреслено, що заклади освіти повинні використовувати різні форми та методи комунікації з абітурієнтами, студентами та батьками. У той же час Закон України «Про інформацію» (Стаття 6) описано право доступу до інформації про діяльність органів влади та інших суб'єктів влади, у тому числі й закладів вищої освіти.

Після повномасштабного вторгнення, частина громадян України має обмежену можливість шляхів доступу до інформації, саме тому сучасні ЗВО з метою забезпечення якісної комунікації та виконання норм законодавства мають сприяти використанню різноманітних каналів комунікації. Одним із таких каналів є використання Телеграм-ботів.

Дослідження показують, що використання чат-ботів зростає. Згідно з дослідженнями Gartner, до 2022 року 70% взаємодій з клієнтами здійснювалося за допомогою нових технологій, таких як чат-боти та віртуальні помічники. Чат-боти вже успішно використовуються іншими університетами. Наприклад, Університет Каліфорнії в Берклі використовує чат-бот для надання інформації про вступ, відповідей на запитання та реєстрації на заходи. Впровадження телеграм-бота не потребує значних інвестицій. Вищевикладене чітко свідчить про те, що тема є актуальною.

Мета кваліфікаційної роботи полягає в розробці чат-боту в месенджері Telegram для вступної кампанії Поліського національного університету для ефективності комунікації з абітурієнтами.

Виходячи, з вище означеної мети, перед нами постає ряд **завдань**:

- 1) проаналізувати предметну область дослідження;

- 2) визначити вимоги до розробок такого типу;
- 3) розробити інтерфейс та функціонал бота;
- 4) спроектувати Telegram-бот для вступної кампанії Поліського національного університету;
- 5) протестувати розроблений бот.

Предметом дослідження є методи і засоби проектування інформаційної системи в месенджерах для ефективної комунікації з абітурієнтами.

Об'єктом дослідження є процес розробки Telegram-боту для вступної кампанії Поліського національного університету.

У процесі дослідження використані наступні методи: *аналізу* (аналіз наукової та науково-методичної літератури з області дослідження); *порівняння* (порівняння аналогів чат-ботів на платформі Telegram; різних методів відповіді на запитання користувачів, зокрема, базовий шаблонний відгук проти відповідей, згенерованих за допомогою штучного інтелекту); *машинного навчання* (для покращення процесу розпізнавання та обробки запитів користувачів, таких як відповіді на запитання або аналіз текстової інформації); *автоматизації* (для процесів обробки запитів користувачів та надання відповідей, що допомагає забезпечити швидку та ефективну відповідь на запитання або запити); *веб-скрапінгу* (для отримання актуальної інформації про вступну кампанію з офіційних ресурсів університету); *тестування й аналітики* (дослідження працездатності розробки; для збору та аналізу даних про взаємодію користувачів з ботом та оцінки ефективності його роботи).

Практичне значення отриманих результатів. Результати дослідження можуть бути використані університетом для покращення комунікації з абітурієнтами, зробити інформацію про вступну кампанію більш доступною, персоналізувати спілкування з абітурієнтами, збирати зворотний зв'язок та знизити витрати на вступну кампанію.

Наукова новизна полягає у інтеграції штучного інтелекту для покращення взаємодії з користувачами. Ця інформація може бути корисною у процесі дослідження можливостей використання штучного інтелекту для

покращення взаємодії Telegram-бота з користувачами. **Унікальністю** кваліфікаційної роботи є використання передових технологій у розробці чат-бота: застосування штучного інтелекту та машинного навчання для покращення функціональності чат-бота, зокрема: розпізнавання природної мови для розуміння запитів абітурієнтів та персоналізація відповідей на основі контексту спілкування.

Апробація результатів дослідження здійснювалася під час Міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених «Безпека, технології, інновації: нові горизонти» (м. Житомир, 2023 р.) та Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем» (м. Житомир, 2024 р.), за результатами яких опубліковано тези у збірниках конференцій.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 24 сторінки.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ДОСЛІДЖЕННЯ

1.1. Аналіз інформаційних потреб і визначення предметної області дослідження

В умовах динамічного розвитку інформаційних технологій та зростання популярності онлайн-комунікацій, заклади вищої освіти змушені шукати нові та ефективні способи взаємодії з абітурієнтами.

Вступна кампанія – це час критично важливих рішень для абітурієнтів. Кожен майбутній студент має потребу в швидкому доступі до чіткої та доступної інформації. Існуючі проблеми та виклики, з якими стикаються абітурієнти, пов'язані не лише з великою кількістю інформації, але й з її фрагментарністю та неоднорідністю.

У світлі сучасних технологій та зростаючого інтересу до корисних та ефективних інформаційних інструментів, аналіз інформаційних потреб є достатньо важливим кроком. Абітурієнти хочуть отримувати не лише базову інформацію про університет, а й чіткі відповіді на свої запитання. Такий телеграм-бот може стати ідеальним рішенням для полегшення доступу до інформації та надання швидких відповідей на індивідуальні запитання.

Окрім технологій месенджерів, які слугують наочною відповіддю на виклики сучасного комунікаційного простору, ще одна революційна технологія – штучний інтелект (ШІ) – активно трансформує наше розуміння світу. Його здатність аналізувати величезні обсяги даних, виконувати завдання, які раніше виконували люди, та прогнозувати майбутнє у різних сферах життя вже знайшла практичне застосування в багатьох галузях. Об'єднання можливостей Telegram-боту та ШІ відкриває нові, захоплюючі перспективи для суспільства, бізнесу, науки та багатьох інших сфер.

Під час визначення цільової аудиторії було встановлено, що різні групи користувачів мають різні потреби та очікування. Враховуючи, що до боту буде інтегровано ШІ, це допоможе створити персоналізоване середовище взаємодії та залучити більше абітурієнтів, а також їх батьків.

Однією з ключових переваг телеграм-бота є його здатність забезпечити інформаційний доступ в одному місці. Замість того, щоб переглядати кілька

ресурсів, абітурієнт може отримати всю необхідну інформацію в одному додатку. Це особливо важливо в умовах інформаційного перенасичення та швидкого ритму сучасного життя.

Такий телеграм-бот може не лише вирішувати існуючі проблеми, але й вносити нові можливості в процес вступу. Забезпечуючи оперативний доступ до інформації та надаючи інтерактивні можливості для спілкування, він стає невід'ємною частиною інформаційного простору вступної кампанії університету.

Наразі Поліський національний університет не має власного телеграм-бота, що контрастує з досвідом інших українських навчальних закладів, які вже впровадили цю інноваційну технологію. В додатку А табл. 1 представлено порівняльний аналіз телеграм-ботів для вищих навчальних закладів, а саме: Луцького національного технічного університету, Київського політехнічного інституту ім. І. Сікорського та Одеського національного університету. У якості критеріїв аналізу телеграм-ботів ЗВО було обрано: *інтерфейс* (наявність кнопок та зображень; зручність навігації та пошуку інформації; естетична привабливість інтерфейсу); *зручність* (простота використання бота; швидкість отримання відповідей; відсутність технічних проблем); *функціональність* (наявність інформації про університет; можливість отримати відповіді на запитання користувачів; наявність додаткових функцій – реєстрація на події, подання документів на вступ); *підтримка користувачів* (наявність можливості зв'язатися з оператором або отримати допомогу; швидкість та якість відповідей на звернення користувачів); *відгуки* (можливість користувачам залишати відгуки про роботу бота; врахування відгуків користувачів при вдосконаленні бота).

Найбільш зручним та функціональним серед розглянутих телеграм-ботів є чат-бот Одеського національного університету. Він має інтуїтивно зрозумілий інтерфейс з кнопками та зображеннями, надає швидкі та актуальні відповіді, а також корисну інформацію про університет. Чат-боти Луцького національного технічного університету та Київського політехнічного інституту ім. І. Сікорського також мають свої переваги, але їм не вистачає деяких функцій та зручності.

Таким чином, під час розробки телеграм-боту для вступної кампанії Поліського національного університету варто врахувати наступні рекомендації:

використовувати інтуїтивно зрозумілий інтерфейс з кнопками та зображеннями; забезпечувати швидкі та актуальні відповіді на запити користувачів інтегрувавши ІІІ; надавати корисну та повну інформацію про університет; впровадити систему підтримки користувачів.

Впровадження цих рекомендацій дозволить покращити зручність та функціональність телеграм-ботів ВНЗ, що зробить їх більш корисними та затребуваними серед абітурієнтів, студентів та інших користувачів.

У цьому контексті створення телеграм-бота для вступної кампанії університету є кроком вперед у використанні технологій для покращення взаємодії між університетом та абітурієнтами.

1.2 Функціональне моделювання розробки телеграм-боту

Для визначення основних функцій та можливостей бота, які відповідатимуть потребам користувачів та цілям університету, а також для розробки структури та інтерфейсу бота, які забезпечать зручність та простоту його використання необхідна чітка та детальна специфікація телеграм-бота, яка буде слугувати основою для його подальшої розробки. Саме тому, скористаємося IDEF0 – методологія функціонального моделювання, яка використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв'язують ці функції.[2]

Основні складові IDEF0-моделі охоплюють різні рівні декомпозиції, сприяючи більш детальному вивченню функцій та взаємодій між ними. Розбиття на два рівні декомпозиції дає змогу провести системний аналіз на більш глибокому рівні функціональної моделі, охоплюючи як загальні, так і деталізовані аспекти системи.

Перший рівень декомпозиції сприяє детальному аналізу всього процесу створення телеграм-бота, шляхом розбиття основного блоку на п'ять під-блоків (див. рис. 1.2.1. дод. Б.). Кожен з блоків описує окремий процес: від аналізу вимог до впровадження, а разом, при послідовному виконанні, вони утворюють один загальний процес – створення бота. Механізми та керування розподілені по кожному блоку, які було наведено в контекстній діаграмі. Блоки пов'язані між

собою стрілками, які є результатом для одного блоку і вхідними даними для іншого. Наприклад: проаналізувавши всі вимоги, можна проєктувати бота, розробляти, тестувати та в кінцевому результаті, впроваджувати у вступну кампанію.

Раніше було зазначено, що декомпозиція першого рівня складається з п'яти блоків, першим з яких є «Аналіз вимог». Це важливий крок перед розробкою бота, оскільки він дозволяє зрозуміти потреби користувачів і закладає основу для успішного проєктування, що в свою чергу сприяє ефективній розробці, уникає непотрібних зусиль і ресурсів, забезпечує точність втілення функцій, які відповідають очікуванням і вимогам конкретних зацікавлених сторін.

Другим блоком є «Проектування боту». Правильне проектування забезпечує зручність взаємодії, ефективність та позитивний досвід користувачів, що сприяє успіху проєкту в цілому.

Одним з найважливіших є блок «Розробка боту». Це відповідальний процес створення програмного агента, який автоматизує комунікацію з користувачами через текстовий або кнопочковий інтерфейс. Розробка потребує значної уважності, практичних та теоритичних знань.

Після етапу розробки, йде «Тестування». Цей етап націлений на виявлення будь-яких помилок. Якщо помилки відсутні, наступним етапом є «Впровадження боту»

Для більш деталізованого опису такого важливого етапу як «Проектування», була створена декомпозиція другого рівня (див. рис. 1.2.2 Дод. Б.) Декомпозиція має чотири блоки, на відміну від минулої, та дає змогу детальніше проаналізувати процес самого проектування.

Перший блок – «Визначення архітектури». Правильно обрана архітектура допомагає досягти оптимальної продуктивності, масштабованості та ефективності бота. Узгоджена архітектура забезпечує стабільність та високу якість роботи телеграм-бота в процесі його експлуатації.

Другий блок – «Вибір технологій». Це важливий крок, оскільки він визначає інструменти та ресурси, які будуть використані для розробки. Вибір мови програмування, фреймворків та інших інструментів повинен враховувати

якість підтримки та актуальність. Оптимально підібрані технології сприяють ефективності розробки, підтримці та майбутньому розвитку телеграм-бота.

Третім блоком є «Проектування інтерфейсу». Грамотно спроектований інтерфейс допомагає користувачеві легко отримувати необхідну інформацію та взаємодіяти з ботом.

Ключовим блоком є «Безпека». Аспекти безпеки є критично важливими у розробці телеграм-бота, оскільки вони впливають на захист конфіденційності, цілісності та доступності даних. Важливо використовувати шифрування для захисту передачі та зберігання чутливої інформації, такої як особисті дані користувачів.

1.3 Структура системи за напрямом розробки телеграм-боту для вступної кампанії Поліського національного університету

Розглянути структуру системи за напрямом предметної області допоможе діаграма IDEF 3. Діаграма IDEF 3, яка використовується для опису взаємодії між об'єктами даних у контексті інформаційної системи. Це може бути корисно при проектуванні, розробці або вдосконаленні систем обробки інформації, таких як бази даних, програмні системи тощо.

Діаграма IDEF 3 використовує стрілки та блоки для зображення потоків даних та операцій, що відбуваються з цими даними. Це дозволяє аналітикам та розробникам легше розібратися у тому, як дані обробляються та взаємодіють між собою у межах системи.

Діаграму IDEF 3 по процесу вступної кампанії Поліського університету продемонстровано в дод. В.

Перша дія – подання заявок на вступ, де абітурієнт повинен створити заявку для вступу в університет, прикріпивши всі необхідні для цього документи. Далі йде процес оцінки заявки вступною комісією, опанування якого розбивається з'єднанням «АБО», яке розгортає цей потік на два блоки: достатня кількість балів та недостатня кількість балів. Також слід зауважити, що при оцінці заявок враховується пристуність або відстуність пільг. Достатня кількість балів включає

в собі два можливі сценарії: або бюджетна форма навчання, або контрактна. Якщо заявка набирає достатню кількість балів для бюджетної форми навчання, абітурієнт зараховується. Якщо заявка набирає кількість балів лише для контрактної форми навчання, абітурієнт може або скасувати її, або зарахуватись. Якщо заявка не набирає мінімальну кількість балів, абітурієнту або пропонується інша спеціальність з меншим прохідним балом, або заявка скасовується взагалі.

Ще однією діаграмою, за допомогою якої можна детальніше розглянути структурну систему за напрямом предметної області, є DFD-діаграма (див. Рис. 1.3.2, дод. В). DFD дозволяє легко дослідити, як дані обробляються та переміщуються в системі, що робить її ефективним інструментом для аналізу та проектування процесів в інформаційних системах.

Діаграма містить:

- Процеси: запуск бота через посилання на нього, запит необхідної інформації, ознайомлення з необхідною інформацією, оновлення інформації шляхом донавчання штучного інтелекту(ШІ);
- Зовнішні сутності: користувач, адміністратор;
- Сховище даних: нейронна мережа, яка підключена до телеграм-бота.

Висновки до першого розділу

Аналіз інформаційних потреб та визначення предметної області дослідження продемонстрували, що сьогодні існують значні можливості для використання телеграм-ботів у вступній кампанії університетів. Запропоноване функціональне моделювання розробки телеграм-бота дає чітке уявлення про основні етапи та кроки, які необхідно виконати для створення ефективного та зручного бота. Структура системи за напрямом розробки телеграм-боту, описується діаграмами IDEF0, IDEF3 та DFD, дозволяє детально проаналізувати процеси та потоки даних, що відбуваються в системі.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТУ

2.1. Реалізація математичної моделі для персоналізації відповідей

Математична модель для персоналізації відповідей абітурієнтів Поліського національного університету буде ґрунтуватися на системі рекомендацій, яка використовує метод collaborative filtering (CF) для прогнозування того, які матеріали будуть цікавими для конкретного абітурієнта (підбір матеріалу здійснюється з врахуванням – моделі аналізу якості контенту див. дод. Г). Ця система ґрунтується на принципі, що користувачам, які мають схожі інтереси, ймовірно, сподобаються схожі продукти або послуги.

Нехай U - це множина абітурієнтів, а I - множина матеріалів (відповідей, статей, відео тощо).

Нехай $R(u, i)$ - це значення, яке відображає, наскільки користувач u зацікавлений у матеріалі i .

Прогнозована оцінка (R_{ui}) – представляє собою середнє значення зацікавленості матеріалами інших користувачів, які схожі за інтересами на поточного користувач (див. дод. Д):

$$R_{ui} = \sum_{U_{sim}(u)} \frac{R_v(i)}{|U_{sim}(u)|}, \text{ де}$$

- R_{ui} - прогнозована оцінка того, наскільки користувач u зацікавлений у матеріалі i .
- $U_{sim}(u)$ - множина користувачів, які схожі на u за своїми інтересами.
- $R_v(i)$ - оцінка користувача v матеріалу i .
- $|U_{sim}(u)|$ - кардинальність множини $U_{sim}(u)$.

Подібність користувачів ($sim(u, v)$) – вимірює ступінь схожості інтересів між двома користувачами:

$$sim(u, v) = \sum_i \frac{R_u(i) * R_v(i)}{||R_u|| * ||R_v||},$$

де:

- $\text{sim}(u, v)$ - міра схожості користувачів u та v .
- $R_u(i)$ - оцінка користувача u матеріалу i .
- $R_v(i)$ - оцінка користувача v матеріалу i .
- $\|R_u\|$ - норма вектора оцінок користувача u .
- $\|R_v\|$ - норма вектора оцінок користувача v .

Ці прогнози потім будуть використовуватися для персоналізації взаємодії телеграм-бота з абітурієнтом, оскільки на основі прогнозованих оцінок система може рекомендувати користувачеві u елементи i , для яких $R(u, i)$ є найбільшим.

Запропонована модель дозволяє здійснювати аналітичну та візуальну оцінку зацікавленості матеріалами користувачів (див.рис. 2.1.8. Дод. Д), а також побудувати матрицю схожості між користувачами (див. рис. 2.1.9 . Дод. Д).

На основі цих прогнозів бот може рекомендувати абітурієнтам матеріали, які, ймовірно, їм сподобаються. Бот також може використовувати матрицю схожості між користувачами, щоб знаходити абітурієнтів з схожими інтересами.

Цю інформацію можна використовувати для створення персоналізованих рекомендацій або для формування груп для спільної підготовки.

За допомогою інтегрованого штучного інтелекту та розробленої математичної моделі телеграм-бот може персоналізувати відповіді на запитання абітурієнтів на основі їх інтересів та потреб кількома способами:

1. Аналізу – інтеграція ШІ, дозволяє боту аналізувати текст запитань абітурієнтів, щоб виявити ключові слова та фрази, які свідчать про їхні інтереси та потреби. Наприклад, якщо абітурієнт запитує про «спеціальності, пов'язані з комп'ютерними науками», бот може зрозуміти, що він цікавиться вступом на програму комп'ютерних наук.

2. Збір даних – бот може використовувати модель, щоб прогнозувати, які теми та інформація будуть найбільш цікавими для конкретного абітурієнта. На основі цих прогнозів він може надсилати абітурієнту персоналізовані матеріали.

3. Відповіді – бот може використовувати модель, щоб генерувати динамічні відповіді на запитання абітурієнтів, які враховують контекст розмови та поточні потреби абітурієнта.

2.2. Формалізація моделі системи телеграм-боту для вступної кампанії Поліського національного університету

Вступна кампанія є важливим етапом для будь-якого університету. Від успішності цієї кампанії залежить кількість абітурієнтів, які вступлять на навчання, а також репутація університету. Тому важливо, щоб ЗВО використовував усі доступні методи для залучення абітурієнтів та надання їм необхідної інформації.

Поліський національний університет, як один із провідних університетів України, завжди прагне до вдосконалення процесу вступу. Одним із інноваційних методів, які використовує навчальний заклад для покращення комунікації з абітурієнтами, є телеграм-бот. Це, своєрідний віртуальний помічник, який може надавати абітурієнтам інформацію про університет, вступну кампанію, факультети, спеціальності, графіки вступу, правила прийому, події тощо. Він також може відповідати на запитання абітурієнтів та допомагати їм з реєстрацією.

Для того, щоб зробити роботу телеграм-бота більш ефективною та результативною, важливо формалізувати його модель. Це означає чітко описати компоненти та функції системи, а також алгоритми, за якими вона працює.

Система телеграм-бота для вступної кампанії Поліського національного університету – це інструмент, який допомагає абітурієнтам отримати інформацію про університет, вступну кампанію, факультети, спеціальності, графіки вступу, правила прийому, події тощо. Бот також може відповідати на запитання абітурієнтів та допомагати їм з реєстрацією.

Модель складається з наступних компонентів:

Модуль обробки природної мови (NLP) – аналізує запити користувачів, щоб зрозуміти їхні наміри. Використовує машинне навчання для класифікації запитів та виявлення ключових слів. Може бути налаштована для розуміння української мови та сленгу.

Модуль діалогу – генерує відповіді на запити користувачів на основі інформації з бази даних. Використовує шаблони відповідей, правила та машинне

навчання для персоналізації відповідей. Може підтримувати діалог з користувачем, відповідаючи на уточнюючі питання.

Модуль рекомендацій – рекомендує абітурієнтам факультети та спеціальності на основі їхніх інтересів, оцінок та іншої інформації. Використовує алгоритми колаборативної фільтрації та машинне навчання для персоналізації рекомендацій. Може враховувати конкурентність факультетів та спеціальностей.

Інтерфейс користувача – забезпечує зручний інтерфейс для взаємодії користувачів з ботом. Дозволяє користувачам ставити запитання, отримувати інформацію, реєструватися на події та подавати заявки на вступ. Може бути інтегрований з веб-сайтом Поліського національного університету та соціальними мережами.

Запропонована модель буде використовуватися для: автоматизації завдань, таких як відповіді на поширені запитання, що звільнить час для операторів бота, щоб вони могли зосередитися на більш складних завданнях; персоналізації досвіду абітурієнтів, надаючи їм інформацію, яка відповідає їхнім інтересам; збору зворотного зв'язку від абітурієнтів, щоб покращити роботу бота.

2.3. Проєктування інтерфейсу та реалізація функціоналу

Проєктування – це процес створення концепції, планування та розробки продукту, системи або процесу з метою досягнення певних цілей або вирішення певних проблем, який включає в себе аналіз, проєктування та імплементацію рішень, що оптимізують функціональність, продуктивність та ефективність системи або продукту. Основна ціль – це створення продукту або системи, що відповідає вимогам та задовольняє потреби користувачів.

Ключовими завданнями проєктування є: визначення функціональності бота, створення зручного інтерфейсу для користувачів, реалізація нейронної мережі для взаємодії з користувачами, тестування та вдосконалення продукту.

Не менш важливим кроком під час розробки бота є аналіз користувачів та їх очікування (див. рис. 2.3.1 Дод. Е). Це зручно зобразити за допомогою UML діаграми прецедентів та UML діаграми станів. За допомогою UML діаграми станів можна побачити в яких станах може перебувати система(бот) під час свого робочого циклу (див. рис. 2.3.2 Дод. Е).

Під час всього робочого циклу, спроектований бот може перебувати у наступних станах: *очікування команди від користувача* (початковий стан бота, у цьому стані бот чекає на повідомлення від користувача); *обробка команди* (коли користувач надсилає повідомлення, бот переходить у цей стан – у якому аналізує повідомлення користувача, щоб визначити, що користувач хоче зробити); *надсилання інформації* (якщо користувач просить бота надати інформацію, бот переходить у цей стан – у якому готує та надсилає користувачеві відповідь); *посилання надіслане* (після того, як бот надіслав користувачеві посилання, він переходить у цей стан – у якому бот чекає на подальші дії користувача).

Припустимо, що користувач хоче дізнатися про терміни вступної кампанії. Користувач може надіслати боту повідомлення "Коли вступна кампанія?". Бот перейде зі стану "Очікування команди від користувача" до стану "Обробка команди". Бот аналізує повідомлення користувача та розуміє, що користувач хоче дізнатися про терміни вступної кампанії. Бот готує відповідь та надсилає її користувачеві. Користувач отримує відповідь та задоволений. Бот переходить у стан "Очікування команди від користувача".

У першому розділі, під час аналізу предметної області було виявлено, що користувачами такого бота у переважній більшості є випускники ЗОШ та коледжів, які цікавляться темою вступу до університету та їх батьки. Основним очікуванням яких є швидкий та зручний доступ до актуальної інформації про вступ і університет в цілому, а також можливість отримання відповідей на запитання через текстовий (або кнопочковий) чат з ботом.

Саме тому, головною вимогою до інтерфейсу є простота використання, тобто це має бути інтуїтивно зрозумілий інтерфейс для взаємодії з користувачами.

Структура інтерфейсу:

1) Головне меню: кнопки для доступу до основних функцій бота ("Про університет", "Про факультети", "Про спеціальності", "Про вступну кампанію", "Контактна інформація", "Важливі дати"). На рис. 2.3.3 Додатку Ж зображено головне меню з кнопками телеграм-бота для вступної кампанії Поліського національного університету.

2) Чат-вікно: де користувач може задавати питання та отримувати відповіді від бота (див. рис. 2.3.4 Дод. Ж)

Елементами інтерфейсу є кнопки головного меню для навігації та вибору функцій, а також текстове поле чату для введення запитань та отримання відповідей на них.

Функціонал боту:

1) Надання інформації про вступну кампанію: перелік важливих дат, перелік документів, які необхідні для вступу, контакти приймальної комісії та ін.

2) Взаємодія з нейронною мережею: обробка та аналіз текстових запитань користувачів, надання відповідей на запитання та особистих рекомендацій.

Реалізація функціоналу включає в себе декілька ключових кроків, які необхідно виконати для успішного створення та роботи телеграм-бота:

1) Налаштування та розгортання телеграм-бота з використанням Telegram Bot API: використання Telegram Bot API дозволяє створити бота та надати йому необхідні функції та можливості для взаємодії з користувачами. Налаштування бота включає в себе встановлення параметрів, таких як ім'я, опис, аватарка, а також налаштування доступу та прав користувачів.

2) Підключення штучного інтелекту з використанням ChatGPT API: використання ChatGPT API дозволяє інтегрувати нейронну мережу для текстового спілкування з користувачами. Підключення штучного інтелекту включає в себе навчання моделі на основі великої кількості текстових даних, а

також реалізацію алгоритмів для обробки та аналізу текстових запитань користувачів та надання відповідей на них.

Підключення штучного інтелекту до телеграм-бота може бути реалізоване за допомогою API OpenAI та інтеграції цього API в код бота.

Першим кроком є отримання доступу до OpenAI API. Спочатку потрібно отримати доступ до API OpenAI. Для цього потрібно зареєструватися на їхньому веб-сайті і отримати ключ API (див. Рисунок 2.3.5).

```
API_TOKEN = '6772697615:AAFqtseobjQLYIZluwl_KNVuup6ti_90uj8'
OPENAI_TOKEN = 'sk-proj-6FkcAvWFdx7tFX7Nw75WT3BlbkFJ5VjsjdQDwQg1G9ODQY4N'
```

Рисунок 2.3.5 – API ключ

Після отримання ключа API потрібно встановити бібліотеку OpenAI для роботи з API. Для Python це можна зробити, використовуючи `pip`:

```
pip install openai
```

Потрібно написати код, який буде обробляти вхідні повідомлення від користувачів та відправляти їх до API OpenAI для обробки (див. рисунок 2.3.6).

```
from aiogram import types
from bot import dp
from config import OPENAI_TOKEN
from openai import AsyncOpenAI

client = AsyncOpenAI(api_key=OPENAI_TOKEN)

@dp.message_handler()
async def echo(message: types.Message):
    try:
        messages = [
            {"role": "user", "content": message.text}
        ]
        response = await client.chat.completions.create(
            messages=messages,
            model="gpt-3.5-turbo",
        )
        reply_text = response.choices[0].message.content.strip()
        await message.answer(reply_text)
    except Exception as e:
        error_message = str(e)
        if "insufficient_quota" in error_message:
            await message.answer("На жаль, ви використали всю нашу квоту на використання AI. Спробуйте пізніше.")
        else:
            await message.answer(f"Виникла помилка: {error_message}")
```

Рисунок 2.3.6 – обробка повідомлень

Використовуючи ключ API OpenAI, можна створити запит до сервісу з бота, передаючи текст повідомлення в якості вхідних даних для генерації відповіді.

Після отримання відповіді від API OpenAI бот повинен обробити отриманий текст та відправити його користувачеві через Telegram API.

Не менш вижаливим кроком є перевірка справності роботи бота. Це можна зробити, надіславши йому різні повідомлення.

З використанням ChatGPT Prompts потрібно налаштувати параметри генерації тексту, такі як кількість варіантів відповіді, максимальна довжина відповіді тощо, щоб забезпечити оптимальну роботу бота.

Висновки до розділу 2:

У другому розділі було описано проєктування інтерфейсу та реалізацію функціоналу телеграм-бота для вступної кампанії Поліського національного університету. Охарактеризовано процес підключення штучного інтелекту з використанням ChatGPT API. Оскільки бот використовує машинне навчання та методи аналізу даних для персоналізації відповідей на запитання абітурієнтів. Для реалізації запропонованої математичної моделі для персоналізації відповідей, використовується система рекомендацій на основі методу collaborative filtering. З її допомогою прогнозується зацікавленість абітурієнта в певних матеріалах, на основі прогнозів йому надаються персоналізовані рекомендації.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Інструкція користувачу телеграм-боту для вступної кампанії Поліського національного університету

Щоб розпочати роботу з ботом потрібно відкрити застосунок “Telegram” у своєму смартфоні чи комп’ютері. Далі користувач може знайти бот через пошук або перейти за Qr-кодом. Детальну інструкційну картку представлено у Дод. 3.

Після переходу за посиланням на телеграм-бот Поліського національного університету, користувач побачить привітання від бота з ним.

Далі потрібно натиснути кнопку “Розпочати” або ввести команду “/start” для запуску бота та надіслати запит через текстове поле або головне меню. (див. Рисунок 3.1.1 та рис. 3.1.2):

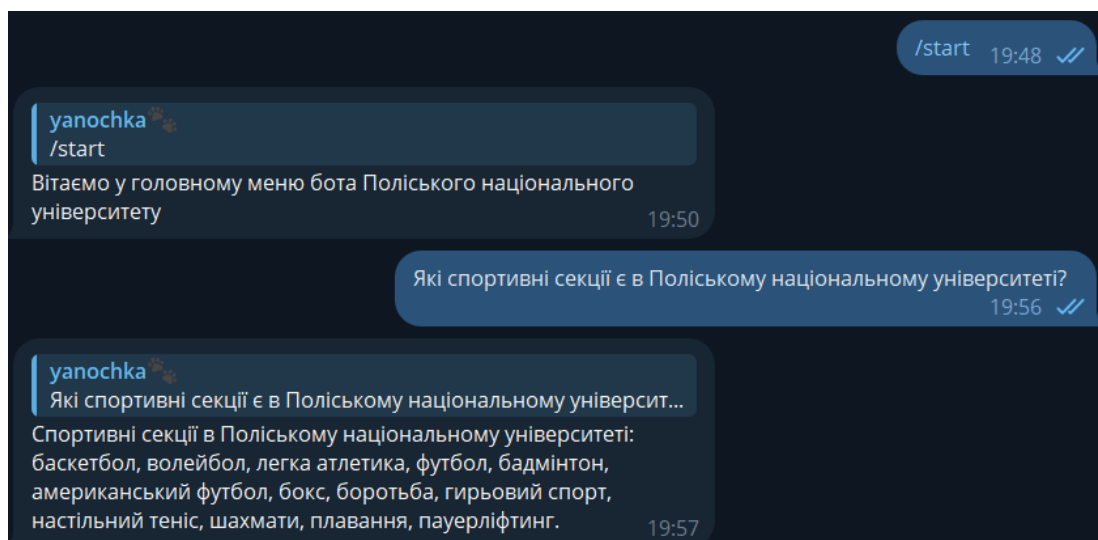


Рисунок 3.1.1 – запит через текст

Після цього буде представлено головне меню у вигляді кнопок. Кожна кнопка відповідає за відповідну функцію. Наприклад, якщо користувач натисне по кнопці з назвою «Факультети», бот надішле йому всю необхідну інформацію про факультети університету. Далі користувач має змогу повернутись до головного меню та ознайомитись з іншою інформацією, яка його цікавить або завершити користування ботом.

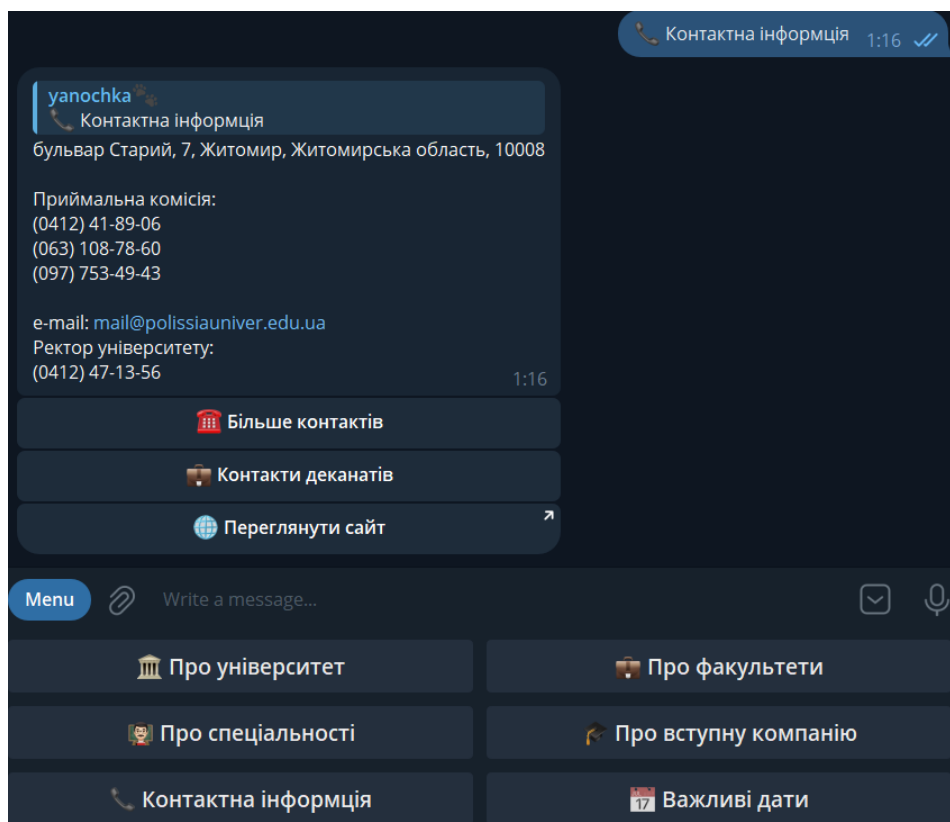


Рисунок 3.1.2 – запит через кнопку

Отже, кожен користувач має змогу індивідуально обрати найзручніший варіант пошуку необхідної інформації: через текстове спілкування з ботом або через кнопки головного меню.

3.2. Тестування телеграм-боту та аналіз результатів

Тестування є важливим кроком для забезпечення якості та надійності телеграм-бота. Перш за все, воно допомагає виявити помилки, дефекти та несправності, які можуть призвести до некоректної роботи бота або недоліків в його функціоналі. Шляхом проведення тестів можна перевірити, чи відповідає реальна реалізація бота вимогам та очікуванням користувачів та розробника.

Тестування допомагає впевнитися, що код написаний належним чином, дотримуючись найкращих практик програмування та стандартів якості. Також дозволяє перевірити, чи працюють усі функції бота так, як очікується, і чи забезпечується коректна взаємодія з користувачем.

Тестування допомагає виявити потенційні проблеми безпеки, такі як уразливості та вразливі точки, що можуть бути використані для атак або неправильної обробки даних. Після успішного проходження тестів розробник може мати впевненість у випуску бота, забезпечуючи його стабільну та надійну роботу.

Отже, тестування є важливою складовою частиною процесу розробки телеграм-бота і допомагає забезпечити високу якість та надійність продукту.

Для перевірки працездатності і якості розробленого продукту, нами застосовано *функціональне тестування* (перевіряє, чи працюють функції бота відповідно до очікувань та специфікацій; виконуються тести на кожну функцію бота, щоб переконатися, що вона працює належним чином) та *тестування продуктивності* оцінює швидкість та продуктивність бота під час обробки запитів та взаємодії з користувачами. Результат функціонального тестування та продуктивності буде зображено нижче на рисунку 3.2.1:

Функція	Швидкість відповіді
/start	2 сек
Натиск на 1 кнопку	1 сек
Натиск на 2 кнопку	1 сек
Натиск на 3 кнопку	1 сек
Натиск на 4 кнопку	1 сек
Натиск на 5 кнопку	1 сек
Натиск на 6 кнопку	1 сек
Текстовий запит	3 сек
Текстовий запит 2	3 сек
Запит англійською	6 сек

Рисунок 3.2.1 – результат тестування

В результаті тестування можна зробити висновок, що бот відповідає очікуванням та чітко виконує функції: відповідає на текстові запитання та надає необхідну інформацію після натиску на кнопку. Слід зауважити, що відповідь надходить за лічені секунди, отже працює стабільно та швидко.

Наступним було *тестування usability*, що допомогло оцінити зручність та простоту використання бота для кінцевих користувачів. Бот тестувався на користувачах різних вікових категорій. Користувачі відмітили, що бот зручний у

використанні, інтерфейс інтуїтивно зрозумілий. Отже, бот повністю задовільняє очікування користувачів.

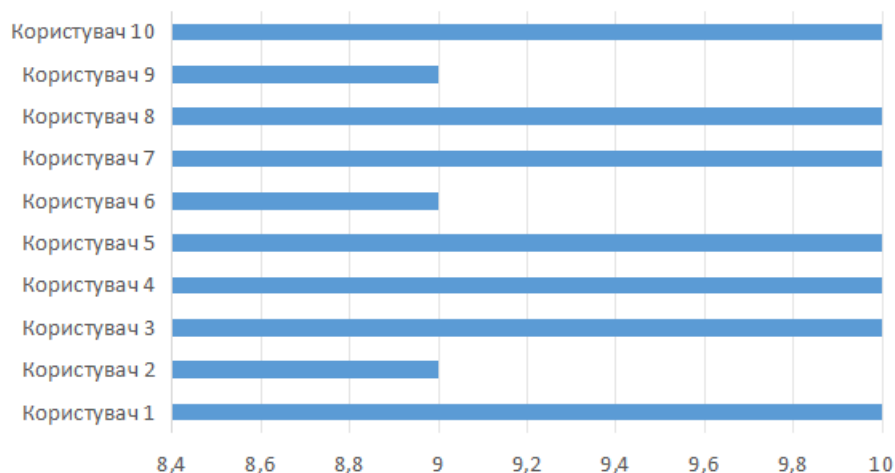


Рисунок 3.2.2 – оцінка роботи бота користувачів

Останнім було *тестування локалізації*, яке оцінює, як бот працює з різними мовами та локалізаціями. Воно включає перевірку коректності відображення тексту, форматування дати та часу, розуміння місцевих культурних особливостей тощо. Результат, тестування на розуміння запитів англійською мовою – позитивний, бот надав чітку, коректну відповідь (див. рисунок 3.2.3).

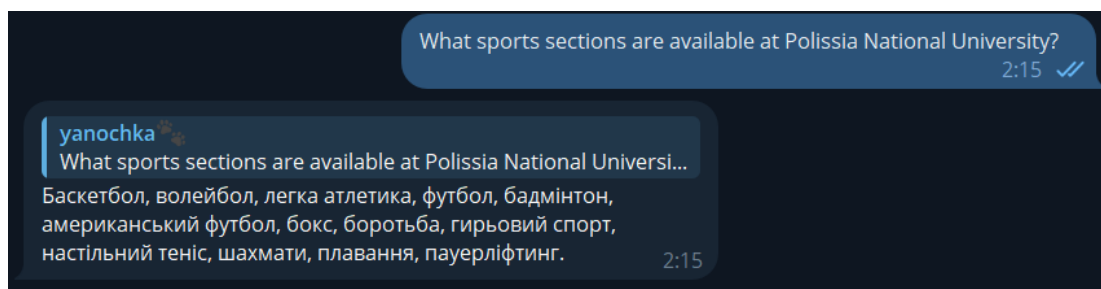


Рисунок 3.2.3 – результат тестування

Наступний етап перевірки працездатності й можливостей бота включав *ручне тестування* (користувачами виконувались різні дії, спроби, та перевірялись відповіді бота на різні запити, що дозволило виявити недоліки, а також оцінити коректність та зручність інтерфейсу). Користувачі, які мали різний досвід використання ботів, виконували різнотипні завдання та давали зворотний зв'язок щодо зручності та ефективності бота. Цей метод дозволяв

отримати реальний відгук від користувачів та виявити потенційні проблеми, які можуть бути непоміченими під час внутрішнього тестування.

Автоматизоване тестування використовувалось для автоматизації тестування різних аспектів телеграм-бота. Були написані автоматизовані скрипти тестування (див. дод. II), які виконували різні сценарії взаємодії з ботом та перевіряли правильність його реакції на різні вхідні дані. Це дозволяло забезпечити швидке та повторюване тестування певних функцій бота під час розробки та змін.

Отже, телеграм-бот для вступної кампанії Поліського національного університету успішно пройшов всі види та методи тестувань і готовий до розгортання та використання в продуктивному середовищі.

Висновки до розділу 3:

В розділі 3 було описано основні види тестування телеграм-бота, такі як: тестування продуктивності; тестування безпеки; тестування юзабіліті; тестування локалізації, а також основні методи тестування: ручне тестування; автоматизоване тестування; тестування з реальними користувачамручне. Після проведення тестування було проаналізовано отримані результати.

ВИСНОВОК

У ході виконання кваліфікаційної роботи були виконані всі поставлені завдання, спрямовані на розробку та впровадження телеграм-бота для вступної кампанії Поліського національного університету.

Був проведений аналіз предметної області, який дозволив детально вивчити особливості вступної кампанії, включаючи етапи, терміни та вимоги до абітурієнтів, аналіз потреб та очікувань абітурієнтів, який був проведений через опитування, що дозволило зрозуміти їхні потреби та очікування від вступної кампанії.

Виявлено, що визначена функціональність бота відповідає вимогам та потребам абітурієнтів, забезпечуючи зручний та ефективний інструмент для отримання інформації про вступ.

Вибір технологічних засобів був здійснений з урахуванням вимог та потреб університету, забезпечуючи ефективну інтеграцію та підтримку бота.

Розроблений інтерфейс та функціонал бота відповідає вимогам та очікуванням користувачів, забезпечуючи зручне та інтуїтивно зрозуміле використання.

Проведене тестування бота дозволило підтвердити його працездатність та відповідність вимогам, забезпечуючи якісну роботу інструменту.

Впровадження та технічна підтримка бота забезпечують його успішне функціонування під час вступної кампанії, забезпечуючи абітурієнтам доступ до необхідної інформації.

Отже, розроблений телеграм-бот відповідає поставленим завданням та вимогам, що дозволяє ефективно використовувати його в рамках вступної кампанії університету.

14. Методичні рекомендації щодо виконання та захисту кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 126 “Інформаційні системи та технології”/ Ю.Б. Бродський, А.В. Лапін, О.В. Маєвський. – Житомир : Поліський нац. ун-т, 2021. – 38 с.

ДОДАТКИ

Додаток А

Характеристика аналогів телеграм-ботів для вищих навчальних закладів

Назва ЗВО	Форма використання	Інтерфейс користувача	Зручність використання	Функціональність	Швидкість відповіді	Актуальність	Технічна стабільність	Технічна підтримка	Відповідність завданням
Чат-бот ЛНУ	Безкоштовна	Текстовий, кноповий	Достатня	Перелік факультетів, форм навчання, мін.конкурсний бал, контактна інформація, посилання на Facebook	Висока	Інформація актуальна	Бот працює стабільно, технічні проблеми відсутні	Відсутня	Відповідає
Чат-бот КП	Безкоштовна	Текстовий	Низька через відсутність кнопок	Відповіді на текстові запити	Дуже низька	Інформація відсутня	Бот працює стабільно	Присутня	Недостатньо відповідає
Чат-бот ОНУ	Безкоштовна	Текстовий, кноповий, зображення	Висока	Перелік факультетів, форм навчання, контактна інформація	Висока	Інформація актуальна	Бот працює стабільно, технічні проблеми відсутні	Відсутня	Відповідає

Додаток Б

Функціональне моделювання розробки телеграм-боту

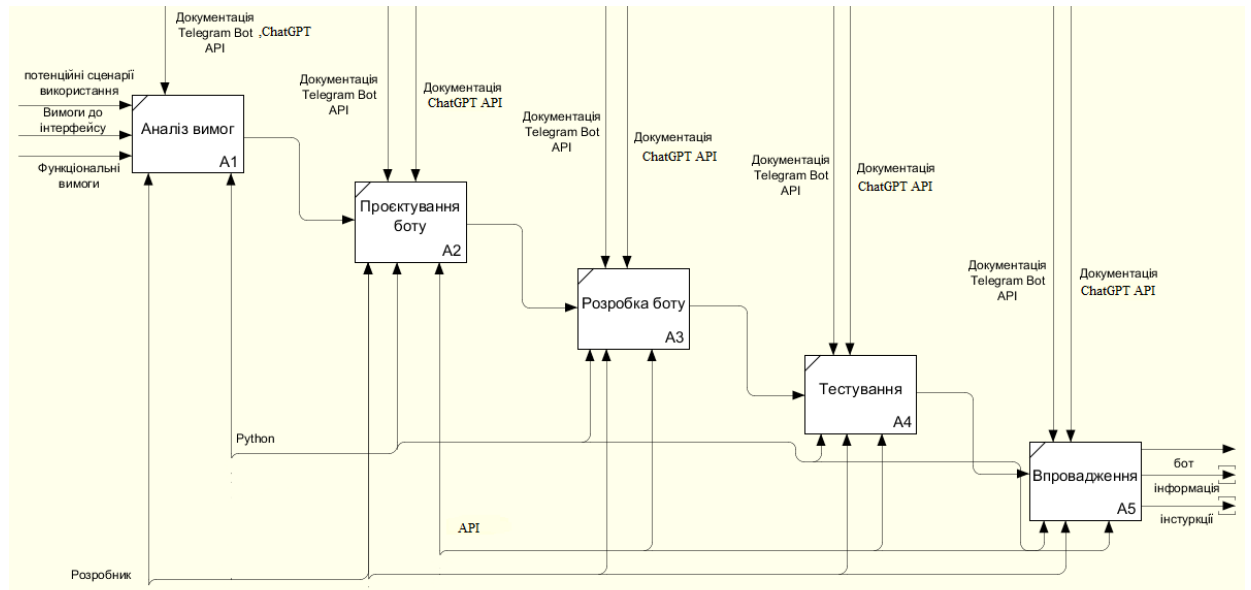


Рисунок 1.2.1 – декомпозиція першого рівня контекстної діаграми IDEF0

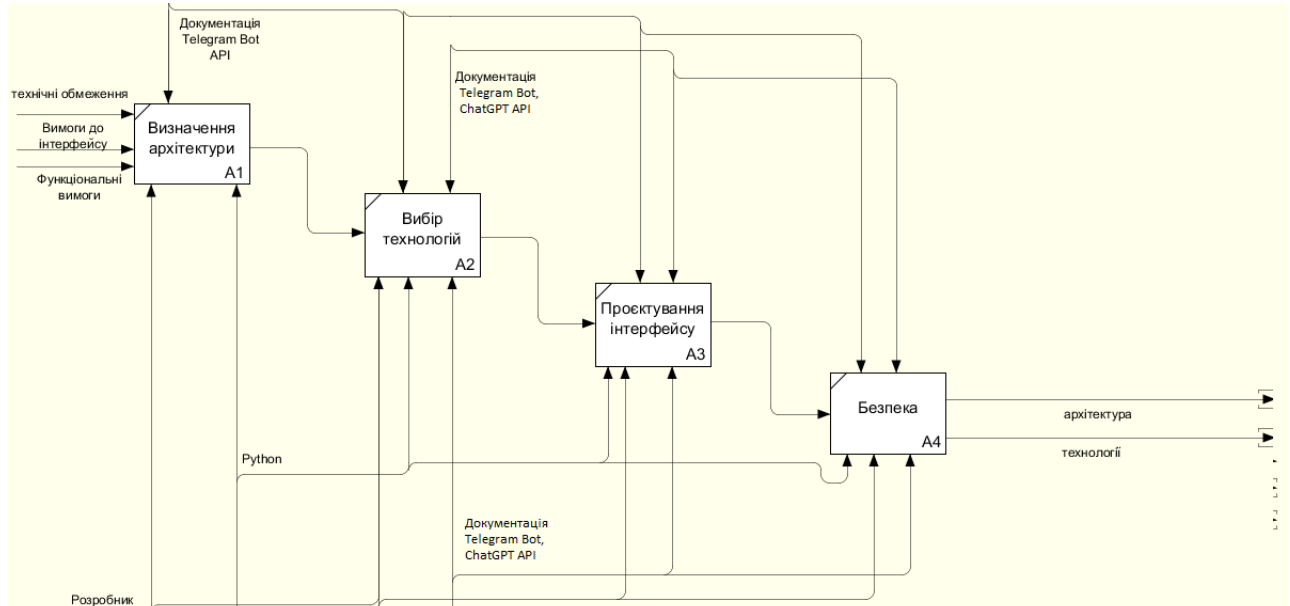


Рисунок 1.2.2 – Декомпозиція другого рівня контекстної діаграми IDEF0 блоку «Проєктування боту»

Додаток В

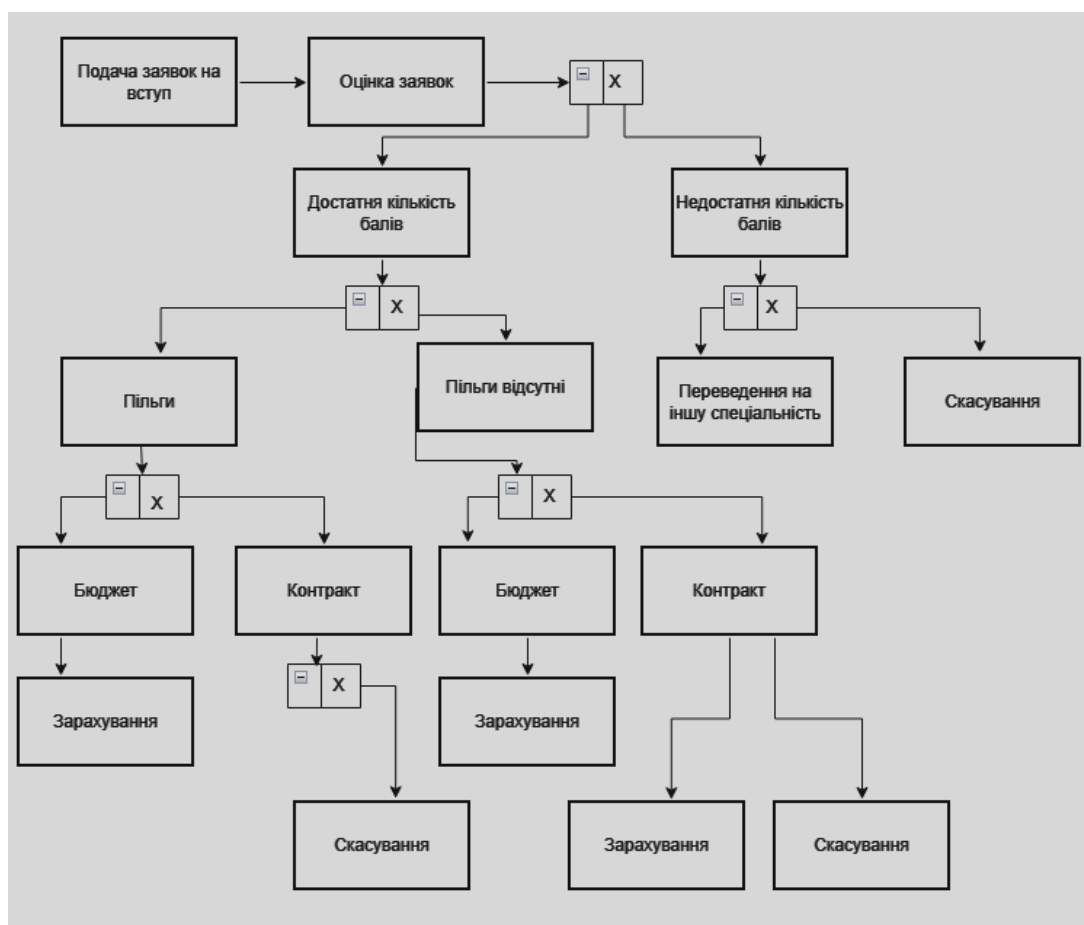


Рисунок 1.3.1 – IDEF3 процесу вступної кампанії університету

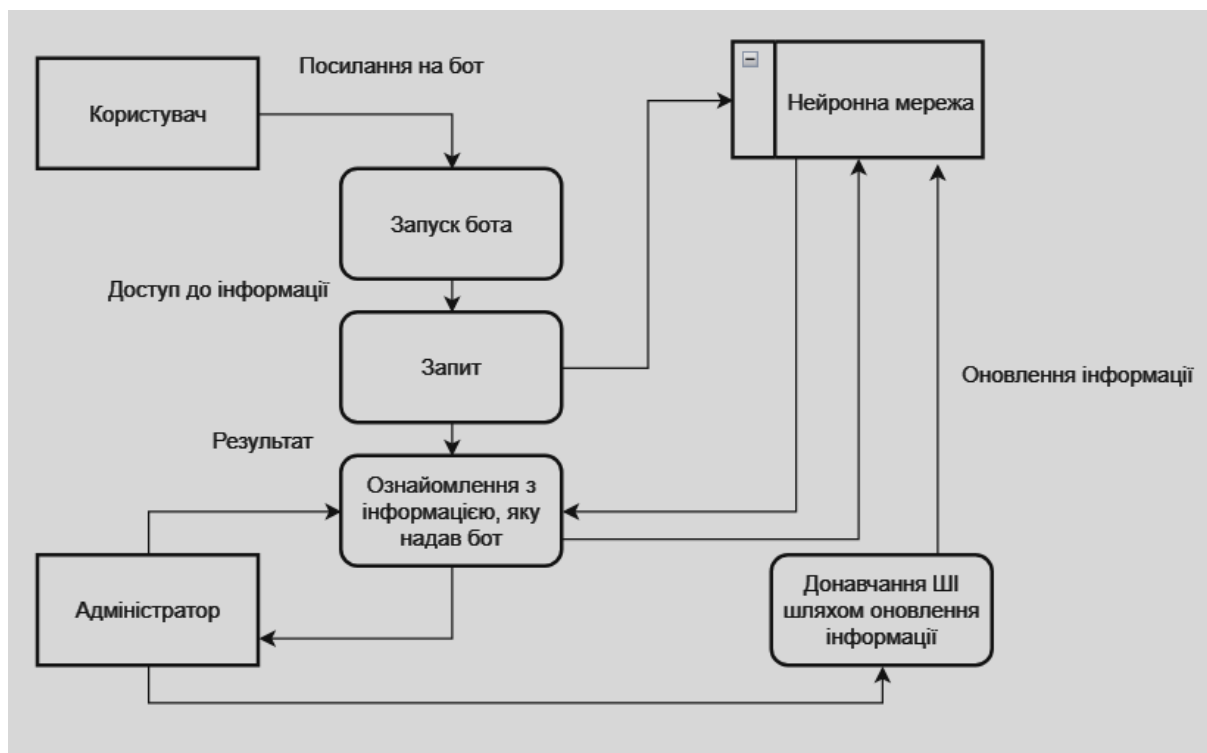


Рисунок 1.3.2 – DFD-діаграма

Додаток Г

Модель аналізу якості контенту

Реалізація моделі була розроблена в редакторі коду Visual Studio, з використанням інтерпретованої об'єктно-орієнтованої мови програмування Python.

Для початку задамо параметри математичної моделі a , b , g , p (рис. 2.1.1)., де a - коефіцієнт, що характеризує вплив якості контенту на приріст кількості користувачів; b - коефіцієнт, що характеризує втрату користувачів з часом; g - початкова кількість користувачів; p - початкова якість контенту.

```
# Задані параметри моделі
a = 0.1
b = 0.05
g = 0.02
p = 0.1
```

Рисунок 2.1.1 – задання параметрів моделі

Опишемо саму систему диференціальних рівнянь (рис. 2.1.2)., де рівняння для кількості користувачів $N(t)$: $dN/dt = a * N * p - b * N$. Рівняння для якості контенту $C(t)$: $dC/dt = 0$

```
# Система диференціальних рівнянь
def model(N, C, t):
    dNdt = a * N * p - b * N
    dCdt = g * N + p
    return dNdt, dCdt
```

Рисунок 2.1.2 – система диференціальних рівнянь

Задамо початкові умови для N_0 , C_0 , y_0 та заданий діапазон часу (рис. 2.1.3)

```
# Початкові умови та момент часу
N0 = 1.0
C0 = 0.5
y0 = [N0, C0]
t = np.linspace(0, 10, 100) # діапазон часу
```

Рисунок 2.1.3 – початкові умови та момент часу

Здійснимо чисельне інтегрування з використанням методу Ейлера (рис. 2.1.4).

Метод Ейлера – це такий чисельний метод, за допомогою якого можна наближено розв'язати звичайні диференціальні рівняння (ЗДР). Він ґрунтується на ідеї апроксимації градієнту функції за допомогою її лінійного наближення. Задача поділяється на кроки і на кожному з них визначається нове значення, використовуючи поточне значення та значення похідної. Хоча метод Ейлера є простим, він може бути не таким точним для деяких видів ЗДР, інші більш складні методи, такі як метод Рунге-Кутта, можуть надавати більш точні результати за рахунок використання додаткових кроків та інтерполяції.

```
# Чисельне інтегрування методом Ейлера
N_t = np.zeros_like(t)
C_t = np.zeros_like(t)

N_t[0], C_t[0] = y0

for i in range(1, len(t)):
    dt = t[i] - t[i-1]
    dNdt, dCdt = model(N_t[i-1], C_t[i-1], t[i-1])
    N_t[i] = N_t[i-1] + dNdt * dt
    C_t[i] = C_t[i-1] + dCdt * dt
```

Рисунок 2.1.4 – чисельне інтегрування методом Ейлера

Останнім кроком було виведення графіків згідно розв’язку рівнянь (рис. 2.1.5). Нижче наведено самі графіки (рис. 2.1.6).

```
# Вивід графіків
plt.figure(figsize=(10, 6))
plt.subplot(2, 1, 1)
plt.plot(t, N_t, label='N(t)')
plt.xlabel('Час')
plt.ylabel('Кількість користувачів')
plt.legend()

plt.subplot(2, 1, 2)
plt.plot(t, C_t, label='C(t)')
plt.xlabel('Час')
plt.ylabel('Рівень якості контенту')
plt.legend()
```

Рисунок 2.1.5 – вивід графіків

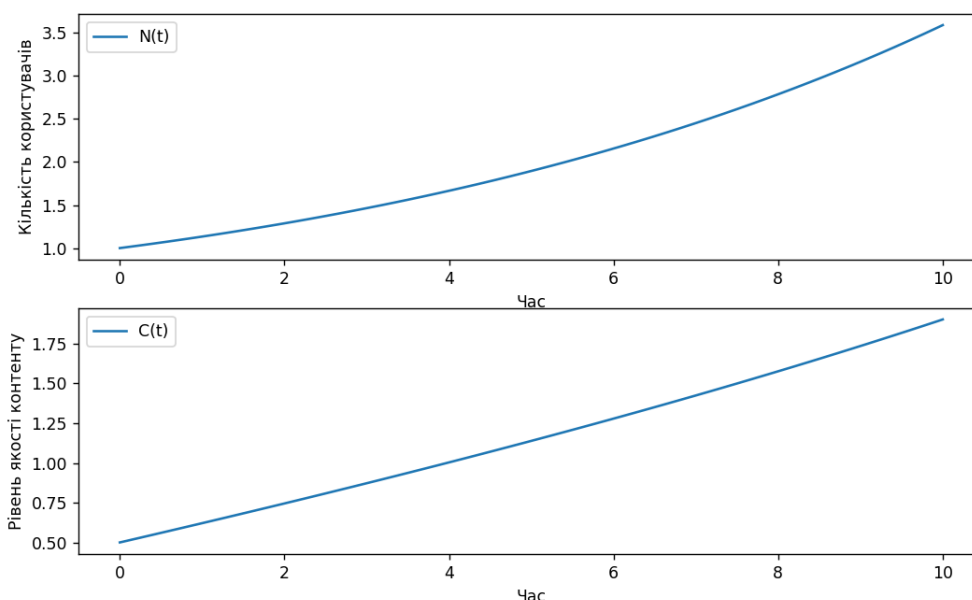


Рисунок 2.1.6 – отримані графіки

Перший графік $N(t)$ показує, як змінюється кількість користувачів з часом. А другий – $C(t)$ показує, як змінюється якість контенту з часом. Аналіз графіків дозволяє зробити висновок, що при зростанні якості контенту спостерігається зростання кількості користувачів.

Ця математична модель дає уявлення про те, як якість контенту може впливати на динаміку зростання кількості користувачів телеграм-бота.

Телеграм-боти стають все більш популярними завдяки своїй простоті використання та доступності. Вони можуть бути ефективним способом надати абітурієнтам інформацію про університет, вступну кампанію, факультети, спеціальності тощо. Чат-боти займають важливе місце серед сучасних технологій. Їх поєднання зручності у використанні, універсальності та взаємодії з користувачами робить їх незамінними інструментами для різних галузей, таких як бізнес, освіта, розваги та інші. Успіх таких ботів - це не тільки успіх

технологій, але й ілюстрація того, як інновації можуть поліпшити наше життя, зробивши його більш ефективним та комфортним.

Успішність боту проявляється в кількох ключових аспектах. По-перше, це зручність та легкість у використанні. Чат-боти дають змогу користувачеві отримати інформацію чи виконати завдання, навіть не виходячи з месенджера. Це значно заощаджує час та ресурси, забезпечуючи швидкий та зручний доступ до необхідної інформації.

По-друге, величезний вибір функцій та сценаріїв роботи робить чат-боти вкрай адаптивними під різноманітні потреби користувачів. Це дає змогу їм успішно функціонувати в різних галузях та вирішувати різноманітні завдання.

Третім аспектом є взаємодія та залучення аудиторії. Такі боти можуть надавати персоналізовані послуги, враховуючи індивідуальні потреби користувачів. Це сприяє формуванню позитивного сприйняття бренду та підвищенню лояльності аудиторії. Також, маючи змогу аналізувати поведінку користувачів, боти мають змогу оптимізувати свою роботу, пропонуючи корисні поради або товари.

Також слід додати, що успішність визначається їхньою здатністю взаємодіяти з іншими технологіями. Інтеграція з іншими сервісами чи платформами дозволяє розширити їх функціональні можливості та забезпечити конкурентну перевагу.

Не менш важливою є якість контенту, який бот пропонує користувачеві. Чим цікавіший та інформативніший буде контент, тим більше буде зацікавленою аудиторія.

Тому, розглянемо формулу оцінки успішності телеграм-боту для вступної кампанії Поліського національного університету.

Для опису динаміки або еволюції телеграм боту, розглянуто його функціонування в просторі таких показників: кількість користувачів та якість контенту.

Зміна кількості користувачів залежить від поточної кількості користувачів та від рівня якості контенту – чим більше користувачів вже є, відповідно тим більше нових користувачів залучає телеграм канал, чим цікавіший контент в каналі – тим більше користувачів.

Під час формалізації моделі використано підхід позитивного зворотнього зв'язку – тобто, чим більше користувачів, тим цікавішим має бути контент. Також слід зауважити, що рівень контенту залежить від професійного рівня людини, яка цей контент створює та публікує.

Тоді функціонування інформаційної системи «телеграм - бот» описується такою моделлю (формула 2.1):

$$\begin{aligned}\frac{dN}{dt} &= aN + bC \\ \frac{dC}{dt} &= gN + P\end{aligned}$$

(2.1)

Де:

$N(t)$ – кількість користувачів телеграм - боту;

$C(t)$ – рівень якості контенту в відповідному телеграм - боті;

P – рівень професіоналізму людини, яка створює та публікує контент;

a, b, g – параметри моделі.

Розв'язок рівняння представлено у такому вигляді (формула 2.2):

(2.2)

$$N(t) = e^{at+bC+K_1}$$

$$C(t) = e^{gt+Pt+K_2}$$

Де K_1, K_2 – вільні члени розв'язку.

Проте, для дослідження доцільно розуміти загальну динаміку системи за різних початкових умов, для цього на основі системи рівнянь створено фазовий портрет системи (див. рисунок 1.7).

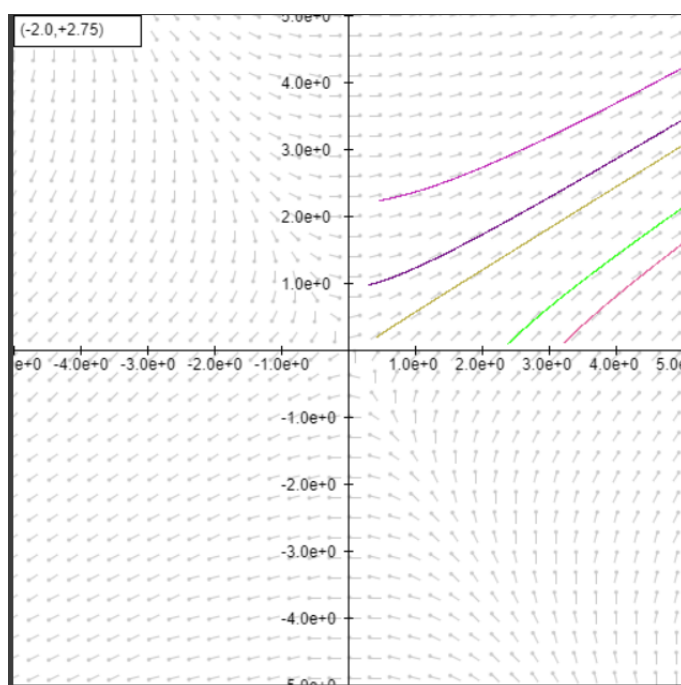


Рисунок 2.1.7 – фазовий портрет системи

Фазовий портрет - зображення траєкторій динамічної системи у фазовому просторі. Кожен стан системи відповідає певній точці на фазовому портреті. Фазові портрети служать для наочного відображення особливостей еволюції динамічної системи: стаціонарних точок, циклів, басейнів притягання.[3]

Додаток Д

Розрахунок подібності користувачів та прогнозованої оцінки

```

1  import numpy as np
2
3  # Оцінки користувачів
4  R = np.array([
5      [5, 4, 0, 0, 3],
6      [4, 0, 0, 0, 1],
7      [1, 1, 0, 0, 4],
8      [0, 0, 5, 4, 0]
9  ])
10
11 # Розрахунок подібності користувачів
12 def similarity(u, v):
13     numerator = np.sum(R[u] * R[v])
14     denominator = np.linalg.norm(R[u]) * np.linalg.norm(R[v])
15     return numerator / denominator
16
17 # Розрахунок прогнозованої оцінки
18 def predicted_rating(u, i):
19     similar_users = [v for v in range(len(R)) if v != u and R[v][i] != 0]
20     numerator = sum(R[v][i] * similarity(u, v) for v in similar_users)
21     denominator = len(similar_users)
22     return numerator / denominator
23
24 # Приклад використання
25 user = 0 # Перший користувач
26 item = 2 # Третій матеріал
27 predicted_rating_value = predicted_rating(user, item)
28 print("Прогнозована оцінка користувача {} для матеріалу {}: {:.2f}".format(user, item,
29     predicted_rating_value))

```

Рис. 2.1.8 - Код розрахунку подібності користувачів та прогнозованої оцінки за допомогою математичних моделей.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Оцінки користувачів
5  R = np.array([[5, 4, 0, 0, 3], [4, 0, 0, 0, 1], [1, 1, 0, 0, 4],
6              [0, 0, 5, 4, 0]])
7
8
9  # Розрахунок подібності користувачів
10 def similarity(u, v):
11     numerator = np.sum(R[u] * R[v])
12     denominator = np.linalg.norm(R[u]) * np.linalg.norm(R[v])
13     if denominator == 0: # Уникаємо ділення на 0
14         return 0
15     else:
16         return numerator / denominator
17
18
19 # Розрахунок прогнозованої оцінки
20 def predicted_rating(u, i):
21     similar_users = [v for v in range(len(R)) if v != u and R[v][i] != 0]
22     if not similar_users: # Перевірка на наявність схожих користувачів
23         return 0
24     else:
25         numerator = sum(R[v][i] * similarity(u, v) for v in similar_users)
26         denominator = len(similar_users)
27         return numerator / denominator

```

```

28
29
30 # Приклад використання
31 user = 0 # Перший користувач
32 item = 2 # Третій матеріал
33 predicted_rating_value = predicted_rating(user, item)
34 print("Прогнозована оцінка користувача {} для матеріалу {}: {:.2f}".format(
35     user, item, predicted_rating_value))
36
37 # Побудова двох окремих графіків
38 plt.figure(figsize=(10, 5))
39
40 # Перший графік - оцінки користувачів для різних матеріалів
41 plt.subplot(1, 2, 1)
42 for user in range(len(R)):
43     plt.plot(range(len(R[user])), R[user], label='Користувач {}'.format(user))
44 plt.xlabel('Номер матеріалу')
45 plt.ylabel('Оцінка')
46 plt.title('Оцінки користувачів для різних матеріалів')
47 plt.legend()
48 plt.grid(True)
49
50 # Другий графік - схожість між користувачами
51 plt.subplot(1, 2, 2)
52 similarities = np.zeros((len(R), len(R)))
53 for u in range(len(R)):
54     for v in range(len(R)):
55         similarities[u, v] = similarity(u, v)
56 plt.imshow(similarities, cmap='viridis', interpolation='nearest')
57 plt.colorbar()
58 plt.title('Схожість між користувачами')
59 plt.xlabel('Користувачі')
60 plt.ylabel('Користувачі')
61
62 plt.tight_layout()
63 plt.show()
64

```

Рис. 2.1.9 - Код побудови графіка оцінок користувачів для різних матеріалів та матриці схожості між користувачами.

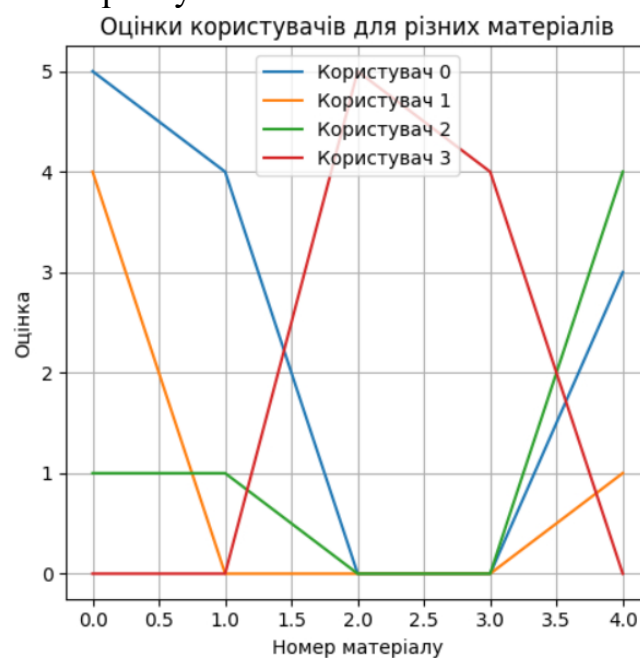


Рис. - 2.1.10 Оцінка користувачів для різних матеріалів

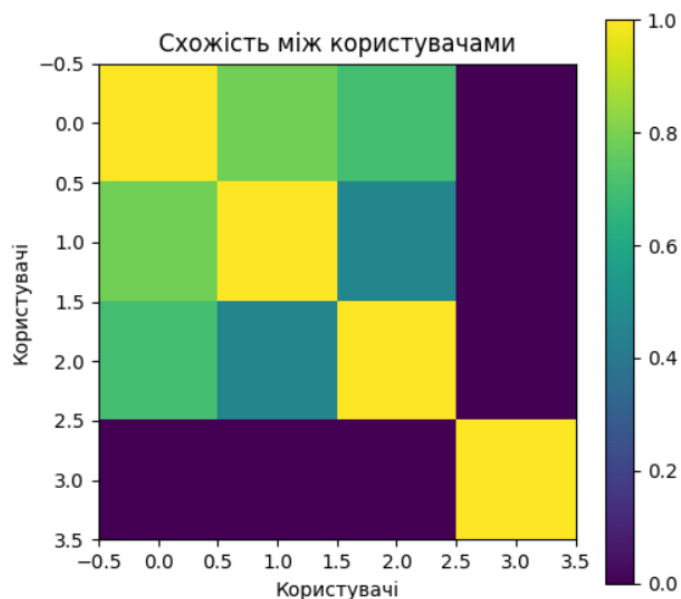


Рис. 2.1.11 - Матриця схожості між користувачами

Рисунки свідчать про те, що існує зв'язок між оцінками користувачів та їх схожістю. Цю інформацію можна використовувати для покращення роботи телеграм-бота для вступної кампанії Поліського національного університету.

Наприклад, бот може рекомендувати матеріали абітурієнтам на основі їх схожості з іншими абітурієнтами, які мають схожі оцінки.

Бот також може генерувати персоналізовані відповіді на запитання абітурієнтів на основі їх інтересів та схожості з іншими абітурієнтами.

Усі коди розміщено за посиланням <https://replit.com/join/wzpahwroin-mk4099265>

Додаток Е

UML діаграми

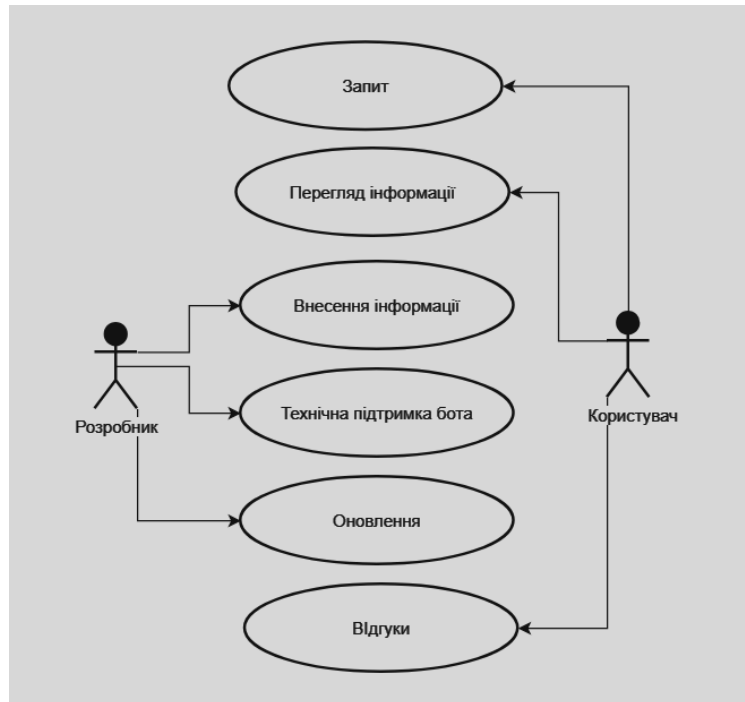


Рисунок 2.3.1 – UML діаграма прецендентів

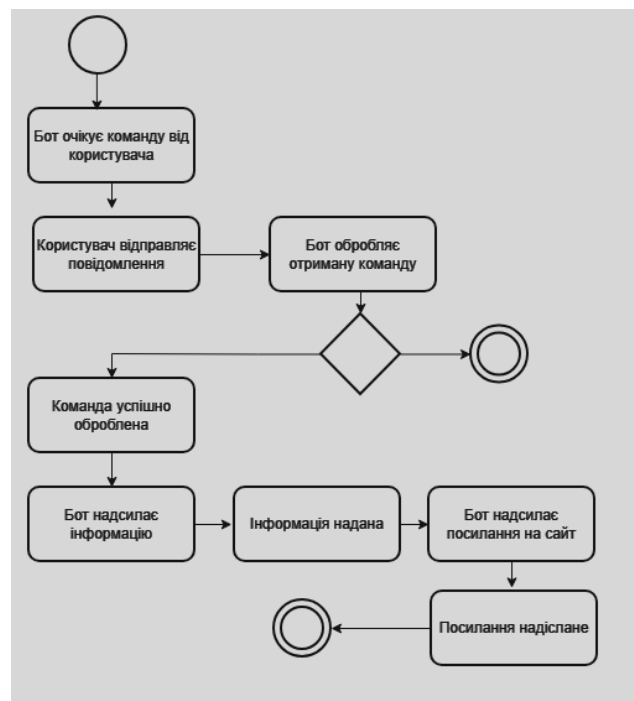


Рисунок 2.3.2 – UML діаграма станів

Додаток Ж

Інтерфейс чат-боту для вступної кампанії

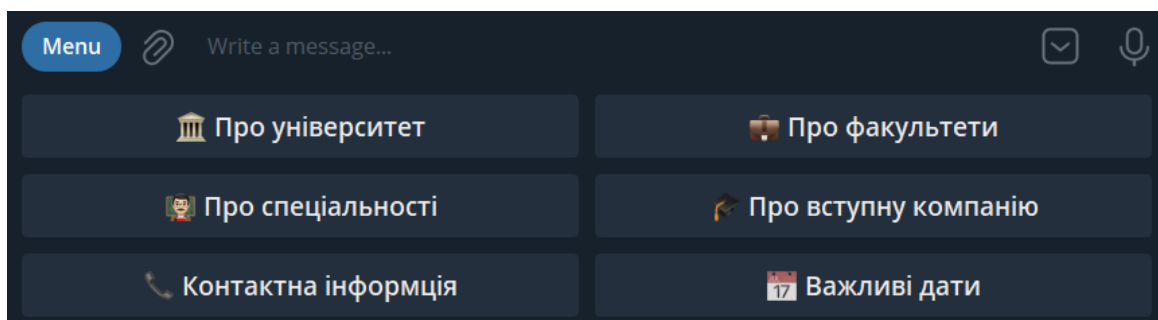


Рисунок 2.3.3 – головне меню

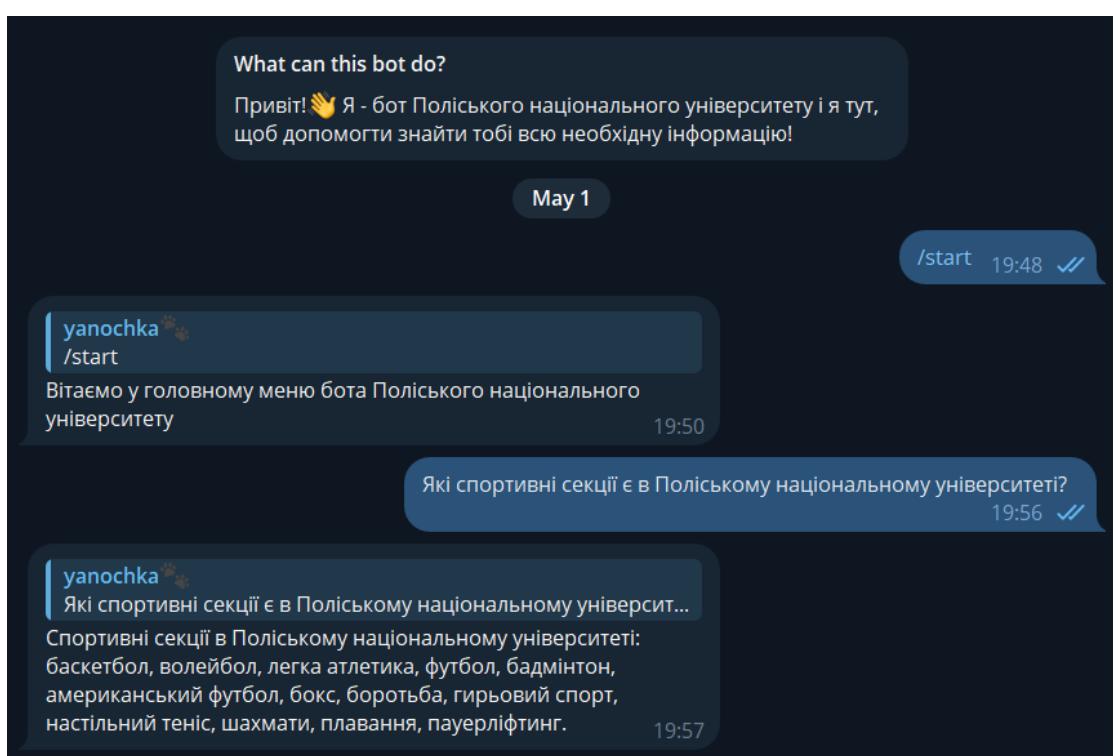


Рисунок 2.3.4 – текстове спілкування з ботом

Додаток 3

Інструкція користувача телеграм-боту для вступної кампанії Поліського національного університету

Вітаємо! Ви перебуваєте на порозі нового та захоплюючого етапу життя – вступу до університету. Поліський національний університет – один із найпрестижніших навчальних закладів України, який пропонує широкий спектр освітніх програм та можливостей для розвитку своїх студентів.

Щоб допомогти вам у цій важливій справі, ми створили телеграм-бот для вступної кампанії. Цей бот стане вашим віртуальним помічником, який надасть вам всю необхідну інформацію про університет, вступну кампанію, факультети, спеціальності, графіки вступу, правила прийому, події та багато іншого.

За допомогою телеграм-бота ви зможете:

- Отримати інформацію про університет, його історію, структуру, факультети та спеціальності.
- Дізнатися про вступну кампанію, графіки вступу, правила прийому та необхідні документи.
- Знайти свою спеціальність та ознайомитися з її навчальним планом, вимогами до вступу та перспективами працевлаштування.
- Задати будь-які запитання про вступ та отримати компетентну відповідь.
- Зареєструватися на події, які проводяться університетом.
- Отримати персоналізовані рекомендації щодо вступу.

Цей телеграм-бот створений для того, щоб зробити процес вступу до Поліського національного університету максимально простим та комфортним. Ми сподіваємося, що він стане вашим цінним помічником на цьому шляху.

Щоб розпочати роботу з ботом, виконайте наступні дії:

1. Відкрийте Telegram на своєму мобільному пристрої.
2. Знайдіть користувача "@PolissyaNationalUniversity_bot" або перейдіть за QR-кодом
3. Натисніть кнопку "Почати".
4. У головному меню бота виберіть пункт, який вас цікавить.
5. Дотримуйтеся інструкцій бота, щоб отримати необхідну інформацію.
6. Якщо у вас виникли запитання, ви можете задати їх боту, надіславши йому повідомлення.
7. Дотримуйтеся інструкцій бота.



Важливо!

- Бот доступний цілодобово 7 днів на тиждень.
- Ви можете задавати боту будь-які питання, пов'язані з вступною кампанією до Поліського національного університету.
- Якщо у вас виникнуть проблеми з роботою бота, зверніться до приймальної комісії університету.
- Бот не замінює реальної комунікації з представниками приймальної комісії.

Ми щиро сподіваємося, що цей бот допоможе вам зробити правильний вибір та стати студентом Поліського національного університету!

Додаток И

ТЕСТУВАННЯ ЧАТ-БОТУ

прикладі скриптів

Тестування функціональності:

```

1  from telethon.sync import TelegramClient
2  from telethon.tl.types import InputPeerUser
3
4  # Змінні, які потрібно заповнити перед запуском
5  api_id = '6772697615:AAFqtseobjQLYIZluwl_KNVuup6ti_90uj8'
6  username = 'PolissyaNationalUniversity_bot'
7
8  1 usage
9  async def main():
10     # Підключення до Telegram
11     client = TelegramClient(username, api_id)
12     await client.start()
13
14     # Отримання списку діалогів
15     dialogs = await client.get_dialogs()
16
17     # Пошук діалогу з ботом за його ім'ям користувача
18     for dialog in dialogs:
19         if dialog.name == username:
20             bot = dialog.entity
21             break
22         else:
23             print("Бот не знайдений у списку діалогів.")
24             return
25
26     # Надсилання тестового повідомлення
27     try:
28         await client.send_message(bot, 'Тестове повідомлення для перевірки функціональності.')
29         print("Тестове повідомлення надіслано успішно.")
30     except Exception as e:
31         print("Помилка при надсиланні тестового повідомлення:", e)
32
33     # Отримання відповіді від бота
34     @client.on(events.NewMessage(from_users=bot))
35     async def handle_response(event):
36         print("Отримано відповідь від бота:", event.message.message)
37
38     # Очікування нових повідомлень
39     print("Очікування відповіді від бота...")
40     await client.run_until_disconnected()
41
42 # Запуск основної функції
43 with TelegramClient('anon', api_id, api_hash) as client:
44     client.loop.run_until_complete(main())

```