

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка програмних засобів підтримки онлайн-
чату на сайті університету»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

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

Володимир СТАЦЕНКО

(підпис)

Виконав: здобувач вищої освіти групи ПД-44

Володимир СТАЦЕНКО

Керівник: _____

Богдан ХУДІК

доктор філософії (PhD) _____

Рецензент: _____

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Стаценку Володимиру Костянтиновичу

1. Тема кваліфікаційної роботи: «Розробка програмних засобів підтримки онлайн-чату на сайті університету»

керівник кваліфікаційної роботи доктор філософії (PhD) Богдан ХУДІК,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56.

2. Строк подання кваліфікаційної роботи «02» червня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про процес надання інформації абітурієнтам та їх батькам про університет, аналіз існуючих рішень у вигляді чат-ботів, проєктувальні матеріали чат-боту, опис процесу реалізації чат-боту, результати тестування чат-боту.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд та аналіз існуючих методів процесу надання інформації абітурієнтам та їх батькам про університет. та технологій для реалізації чат-боту.

2. Проектування чат-боту, шляхом створення діаграми варіантів використання та діаграм структури бази даних.
3. Опис програної реалізації та сценаріїв діалогу розробленого чат-боту.
4. Тестування чат-боту.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до чат-боту.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Структура бази даних.
6. Архітектура чат-боту.
7. Діаграма класів основної логіки чат-боту.
8. Діаграма класів чатінгу.
9. Екранні форми.
10. Апробація результатів дослідження.

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	25.02.-04.03.2025	
2	Аналіз та дослідження існуючих аналогів	05.03-13.03.2025	
3	Огляд існуючих методів інформування абітурієнта про університет та існуючі аналоги чат-ботів	14.03-18.03.2025	
4	Проектування чат-боту для інформування абітурієнта про університет.	19.03-25.03.2025	
5	Програмна реалізація чат-боту	26.03-24.04.2025	
6	Тестування чат-боту	25.04-28.04.2025	
7	Оформлення роботи: вступ, висновки, реферат	29.04-11.05.2025	
8	Розробка демонстраційних матеріалів	12.05-18.05.2025	
9	Попередній захист роботи	19.05-30.05.2025	

Здобувач вищої освіти

_____ (підпис)

Володимир СТАЦЕНКО

Керівник
кваліфікаційної роботи

_____ (підпис)

Богдан ХУДІК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 61 стор., 5 табл., 45 рис., 20 джерел.

Мета роботи – спрощення процедури ознайомлення абітурієнтів університету та особливостями навчання в ньому.

Об'єкт дослідження – процес надання інформації абітурієнтам та їх батькам про університет.

Предмет дослідження – чат-бот для спілкування з абітурієнтом або його батьками про університет та процедуру вступу.

Короткий зміст роботи: в роботі було проаналізовано методи, які університети використовують для інформування студентів про університет, проаналізовано існуючі рішення у вигляді чат-ботів від трьох університетів (1 український, 2 зарубіжні), спроектовано чат-боту за допомогою діаграм варіантів використання та діаграм структури бази даних, було описано процес розробки чат-боту та описано основні сценарії діалогів чат-боту, було описано процес тестування чат-боту та результати тестування.

Сферою використання застосунку є сайт Державного університету Інформаційно-комунікаційних технологій.

КЛЮЧОВІ СЛОВА: ЧАТ-БОТ, C#, BOT FRAMEWORK, MONGODB, SIGNALR, ВСТУП ДО УНІВЕРСИТЕТУ, ОНЛАЙН ЧАТ, REACT.

ЗМІСТ

ВСТУП	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1. Опис предметної області.....	7
1.2 Аналіз існуючих рішень	9
1.3 Вимоги до програмного забезпечення	24
1.4 Постановка задачі	25
1.5 Технології реалізації	26
2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ ПІДТРИМКИ ОНЛАЙН-ЧАТУ НА САЙТІ УНІВЕРСИТЕТУ	29
2.1 Проєктування архітектури системи	29
2.2 Проєктування бази даних.....	31
2.3 Діаграма варіантів використання	34
3. ОПИС РЕАЛІЗАЦІЇ	36
3.1 Структура системи.....	36
3.2 Діаграма класів.....	38
3.3 Реалізація функціоналу чат-боту.....	40
4. ТЕСТУВАННЯ ЧАТ-БОТУ	49
4.1 Ручне тестування.....	49
4.2 Аналіз результатів тестування.....	52
ВИСНОВОК.....	54
ПЕРЕЛІК ПОСИЛАНЬ.....	56
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	58
ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ.....	67

ВСТУП

Вибір університету школярем після завершення школи є одним із найважливіших етапів життя, який визначає майбутнє людини. Абітурієнти можуть обирати університети, кафедри та спеціальності за різноманітними критеріями. Підлаштовуватися під кожного абітурієнта університет не може, оскільки в кожного абітурієнта є власні смаки і вподобання.

В свою чергу, університет зацікавлений у привертанні уваги у якомога більшої кількості абітурієнтів. Для цього університети ведуть соціальні мережі, роблять дні відкритих дверей, створюють та підтримують сайти кафедр та багато іншого. Тобто, університет намагається розповісти про себе якомога більше і якомога в найкращому ключі.

Але навіть попри різноманітність заходів для ознайомлення, все одно у багатьох абітурієнтів (та особливо у їхніх батьків) можуть виникати якісь «локальні» питання стосовно власної дитини, або ж уточнення щодо отриманої на заходах інформації. Задля цього в університетах є спеціальна приймальна комісія, яка приймає як й електронні листи, так й телефонні дзвінки. Але такий підхід використовує велику кількість людських ресурсів, які, як правило, є обмеженими.

Данна дипломна робота пропонує рішення вищенаведеної проблеми, шляхом створення чат-боту для абітурієнтів та їхніх батьків. Сутність чату боту полягає у допомозі отримання відповідей на питання абітурієнтів або їхніх батьків. Чат бот буде містити відповіді на майже усі можливі питання, пов'язані із вступом до університету, інформацією про кафедру, спеціальності, які існують на кафедрі й так далі. У випадку, якщо користувач не зможе знайти відповідь на власне питання, то він завжди матиме можливість зробити запит у чаті на консультацію із членом приймальної комісії.

На поточний момент, дипломна робота буде розроблюватися для кафедри Інженерії програмного забезпечення, але в майбутньому вона може бути розширена для використання для всіх кафедр університету.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Опис предметної області

Тема даного курсового проєкту – Розробка програмних засобів підтримки онлайн-чату на сайті університету». Предметною областю для обраної мною теми є кафедра Інженерії програмного забезпечення в Державному університеті Інформаційно-комунікаційних технологій.

Для початку, необхідно розглянути такий термін, як кафедра в університеті.

Кафедра[1] – це структурний підрозділ університету, основною задачею якого є організація навчальної та науково-дослідницької діяльності в пов'язаній з кафедрою наукової галузі.

Тобто, кафедра це – об'єднання викладачів, наукових співробітників, аспірантів та студентів, які працюють над вирішенням спільних наукових та навчальних завдань за визначеним напрямком.

Розглянемо основні функції кафедри:

- освітня функція. Сутність цієї функції полягає у розробці освітніх планів, програм та методичних матеріалів задля навчання студентів, які вступили на навчання на даній кафедрі;
- науково-дослідницька функція. Сутність цієї функції полягає у виконанні наукових робіт викладачами та науковими учасникам кафедри;
- організаційна функція. Сутність цієї функції полягає у організації семінарів, наукових заходів;
- кадрова функція. Сутність цієї функції полягає у відборі та підвищенні кваліфікації викладачів, а, також сприяння у розвитку наукових кадрів шляхом навчання на аспірантурі та докторантурі.
- тобто, кафедра займається науковим покращенням усіх її учасників у специфічній для неї наукової галузі.

Однією із найважливіших в очах абітурієнта функцій є освітня функція. Бо абітурієнт хоче знати, що саме він буде вивчати, які інструменти використовувати, чи відповідає навчальна програма його інтересам та вимогам сучасності. У зв'язку з цим необхідно надавати доволі велику увагу до розкриття такої інформації у зрозумілому для абітурієнта форматі.

Подальшим інструментом для аналізу предметної області буде слугувати сайт кафедри Інженерії програмного забезпечення у Державному університеті Інформаційно-комунікаційних технологій[2]. Майбутній чат бот повинен містити у собі усю інформацію, яка знаходиться на сайті, та надавати структуровані та чіткі відповіді на питання користувачів.

Виходячи з аналізу сайту кафедри, розроблений чат-бот повинен надавати доступ до інформації наступного характеру:

- інформацію про 121-у спеціальність(переваги, недоліки, навчальні програми, технології й так далі);
- перелік навчальних дисциплін та короткий їх опис;
- перелік освітніх програм та детальна інформація про них;
- інформація про матеріально-технічне забезпечення;
- перелік партнерів кафедри та інформацію про майбутню допомогу з працевлаштуванням.

Також, крім інформації про саму кафедру також доволі важливою та ключовою інформацією є інформація стосовно вступу в університет. Чат-бот повинен надавати користувачеві наступну інформацію:

- перелік документів для вступу;
- відомості стосовно процедуру вступу до університету, а, також, правила прийому;
- перелік переваг університету;
- контактна інформація вступного центру (телефон, емейл, адреса);
- вартість навчання;
- особливості вступу для абітурієнтів з тимчасово окупованих територій;

- перелік конкурсних предметів для вступу;
- перелік спеціальностей на кафедрі та їх освітньо-професійних програм;

1.2 Аналіз існуючих рішень

На даний момент, офіційні чат-боти, які повноцінно функціонують, знайти складно – навіть якщо мова йде про великі вітчизняні університети. Серед існуючих рішень було знайдено 1 існуючий аналог для Українського університету та 2 аналоги зарубіжних університетів:

- чат-бот від Національного Технічного Університету «Харківський політехнічний університет»[2];
- Університет Мурсії (Іспанія) – чат-бот Lola [3];
- Університет Верони(Італія) – чат бот InfoChat Studenti [4].

Існуючі рішення будемо розглядати за їх порядком.

Першим буде розглянуто чат-бот від Національного Технічного Університету «Харківський політехнічний університет». Чат-бот призначений для підтримки абітурієнтів у виборі їх професійного шляху.

Механіка роботи чат-боту побудована у форматі адаптивного опитувальника з елементами штучного інтелекту – бот задає користувачу серію запитань, причому наступні питання змінюються залежно від попередніх відповідей. Для реалізації такого функціоналу розробники інтегрували AI-модель GPT4-O Mini, яку можна налаштовувати під потреби конкретного користувача.

Таким чином, розроблений чат-бот забезпечує індивідуальний підхід до кожного абітурієнта та фактично імітує розмову з представником університету.[5]

Перехід до чат-боту реалізовано шляхом відкриття нового вікна у браузері. У вікні абітурієнт бачить логотип університету, та вікно чату, де

пояснюється основна мета чат-боту та нижче зображено кнопку «Пройти опитування»;

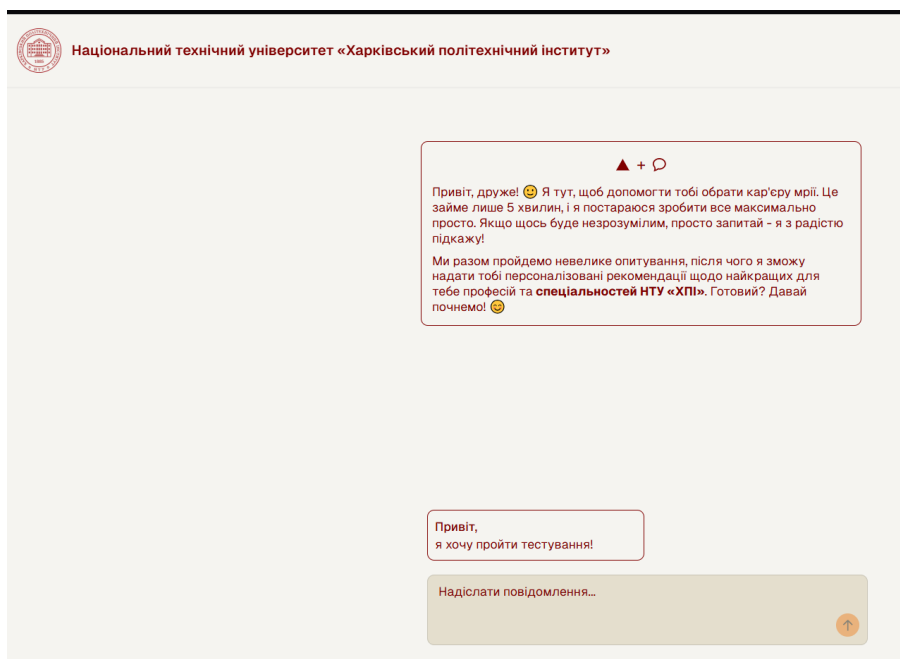


Рис. 1.1 Початкова сторінка чат-боту

Для початку, абітурієнт повинен пройти невеличке опитування, яке допоможе підібрати кафедру та спеціальність, які найбільше підходять для нього. Опитування складається із 6-8 питань.

Питання, які задає чат-бот зображено на рисунках 1.2-1.4.

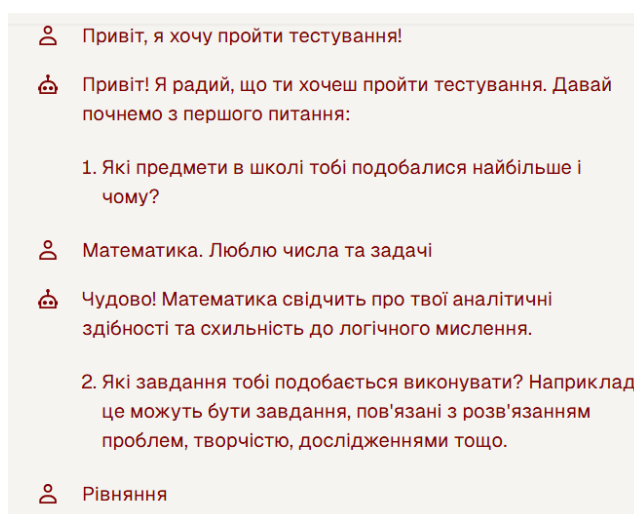


Рис. 1.2 Опитування від чат-боту (частина 1)

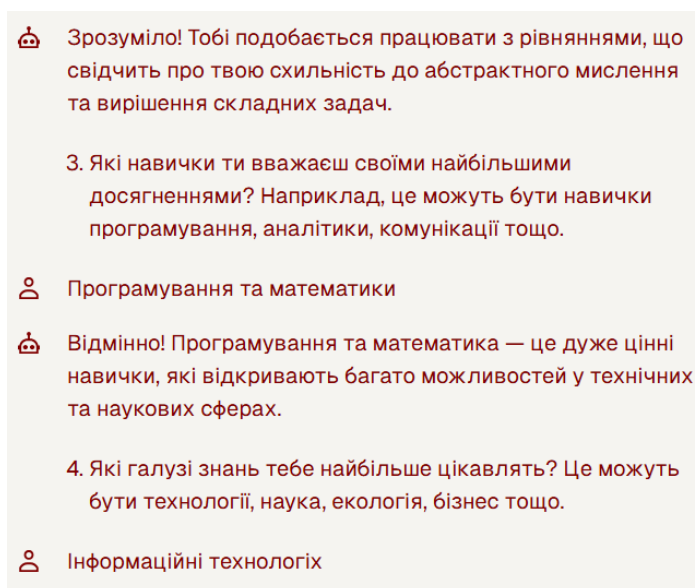


Рис. 1.3 Опитування від чат-боту (частина 2)

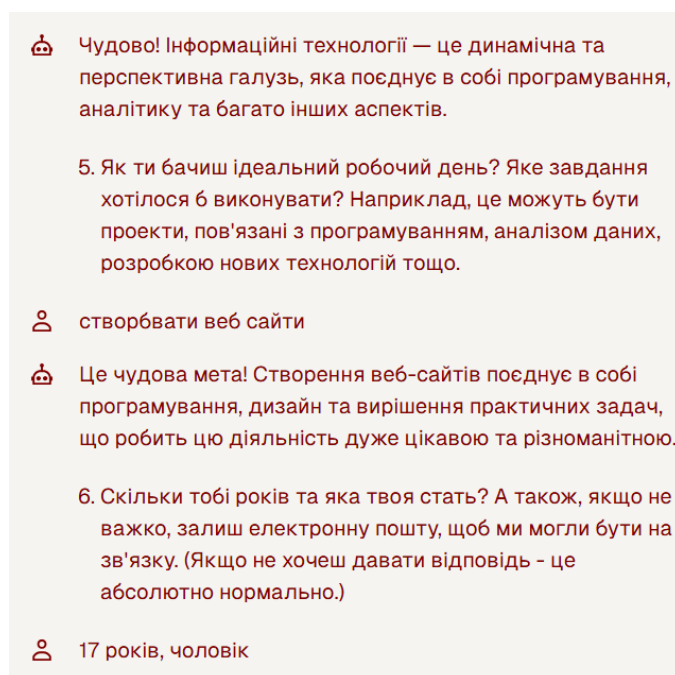


Рис. 1.4 Опитування від чат-боту (частина 3)

Фінальним результатом опитування є сформований портрет користувача, складений перелік рекомендованих професій та перелік рекомендованих спеціальностей НТУ «ХПІ». Сформовану відповідь можна переглянути на рисунках нижче.



Дякую за інформацію! Тепер я можу підсумувати твої відповіді та скласти портрет користувача:

Портрет користувача:

1. Тобі подобається математика, особливо робота з рівняннями, що свідчить про твої аналітичні здібності.
2. Ти маєш навички програмування та цікавишся інформаційними технологіями.
3. Твоя мета — створювати веб-сайти, що поєднує в собі твої інтереси в програмуванні та дизайні.

Тепер давай порівняємо твій портрет з професіями, які пропонує наш університет.

Рис. 1.5 Портрет користувача

На основі твоїх відповідей, найбільше підходять такі професії:

1. **Спеціаліст з комп'ютерних наук та програмування** - ця професія ідеально підходить для тебе, оскільки вона вимагає аналітичного та алгоритмічного мислення, а також навичок програмування, які ти вже маєш.
2. **Спеціаліст з інформаційних систем та технологій** - ця професія також може бути цікавою, оскільки вона передбачає аналіз складних систем та розробку технічних рішень.
3. **Спеціаліст з комп'ютерної інженерії та кібербезпеки** - якщо тебе цікавить безпека інформаційних систем, ця професія може бути для тебе.

Рис. 1.6 Перелік рекомендованих професій

Рекомендовані спеціальності НТУ "ХПІ":

1. F1 Прикладна математика
2. F3 Комп'ютерні науки
3. F2 Інженерія програмного забезпечення
4. F4 Системний аналіз та наука про дані

Ці спеціальності допоможуть тобі розвивати твої навички в програмуванні та інформаційних технологіях, а також підготують до кар'єри в обраній професії.

Якщо у тебе є ще питання або ти хочеш дізнатися більше про конкретні спеціальності, не соромся запитувати!

Рис. 1.7 Перелік рекомендованих спеціальностей

По завершенню опитування, користувач може продовжити вести діалог із чат-бот у вільному форматі, та запитувати будь які запитання пов'язані із діяльністю університету. Наприклад, користувач може запитати більш детальну інформацію про запропоновані йому спеціальності. Чат-бот розділить свою відповідь на таке питання у трьох секціях:

- секція «Опис спеціальності»;
- секція «Перспективи кар'єри»;
- секція «Чому варто обрати цю спеціальність?».

🔗 Спеціальність F2 Інженерія програмного забезпечення в НТУ "ХПІ" зосереджена на розробці, проектуванні, тестуванні та впровадженні програмного забезпечення. Ось кілька ключових аспектів цієї спеціальності:

Опис спеціальності:

1. **Основи програмування:** Студенти вивчають різні мови програмування (таких як Java, C++, Python) та основи алгоритмів і структур даних.
2. **Розробка програмного забезпечення:** Зосередженість на методах і технологіях розробки програмного забезпечення, включаючи Agile, Scrum, DevOps та інші підходи.
3. **Тестування та забезпечення якості:** Вивчення методів тестування програмного забезпечення, автоматизації тестування та забезпечення якості продукту.
4. **Системний аналіз:** Студенти навчаються аналізувати вимоги до програмного забезпечення, проектувати архітектуру системи та взаємодію між компонентами.
5. **Бази даних:** Основи роботи з базами даних, включаючи SQL, NoSQL, проектування баз даних та управління даними.
6. **Веб-розробка:** Вивчення технологій для створення веб-додатків, включаючи HTML, CSS, JavaScript, а також фреймворків для фронтенд та бекенд-розробки.

Рис. 1.8 Секція «Опис спеціальності»

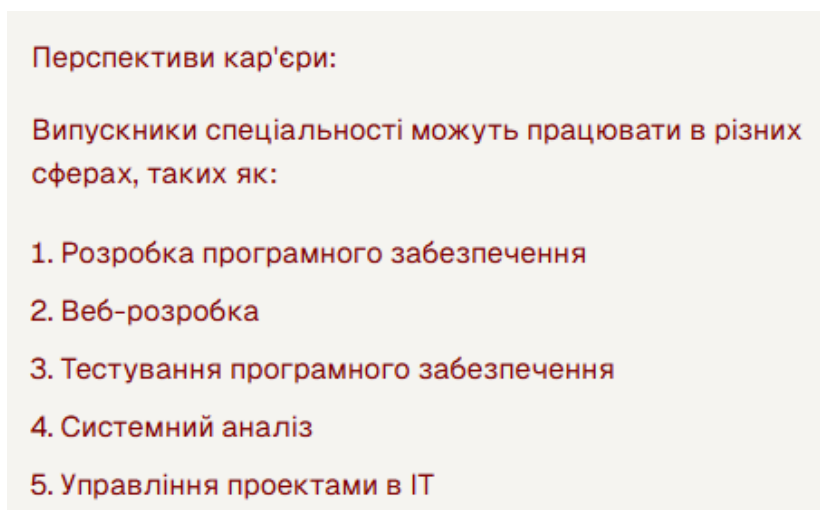


Рис. 1.9 Секція «Перспективи кар'єри»



Рис. 1.10 Секція «Чому варто обрати цю спеціальність?»

Але також, слід зауважити, що інформація, яку надає чат-бот, не завжди є актуальною. Одним із таких прикладів є питання про вступ до університету, на що чат-бот порекомендував портали, які можуть допомогти із підготовкою до ЗНО, хоча на поточний момент для вступу у вищі навчальні заклади необхідно скласти НМТ. Така інформація може ввести в оману абітурієнта та знизити рівень довіри до чат-бота та університету в цілому.

Наступним існуючим рішенням буде розглянуто реалізацію чат-бота Lola від Університету Мурсії (Іспанія).

Чат бот є інтегрованим у систему університету та працює 24/7, щоб підтримувати абітурієнтів. Lola допомагає майбутнім студентам під час передвступної підготовки та процесу зарахування, а саме відповідає на поширені запитання щодо подачі документів, вибору освітніх програм, умов вступу й так далі. Lola здатна вести діалог природною мовою і постійно вдосконалюється, навчаючись на основі запитань користувачів. За перші місяці роботи Lola успішно опрацював тисячі запитів абітурієнтів з приблизною точністю відповідей до 90%.[6]

Щоб розпочати діалог із Lola необхідно перейти на сторінку вступу. На сторінці вступу у правому нижньому кутку буде іконка чат-бота (див. рисунок 1.11) та натиснути на неї. У відкритому вікні необхідно дати згоду на збереження даних чату, як зображено на рисунку 1.12.

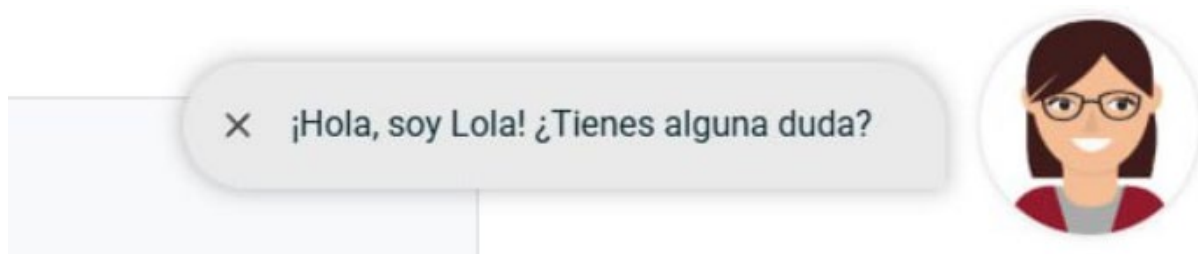


Рис. 1.11 Іконка чат-бота Lola

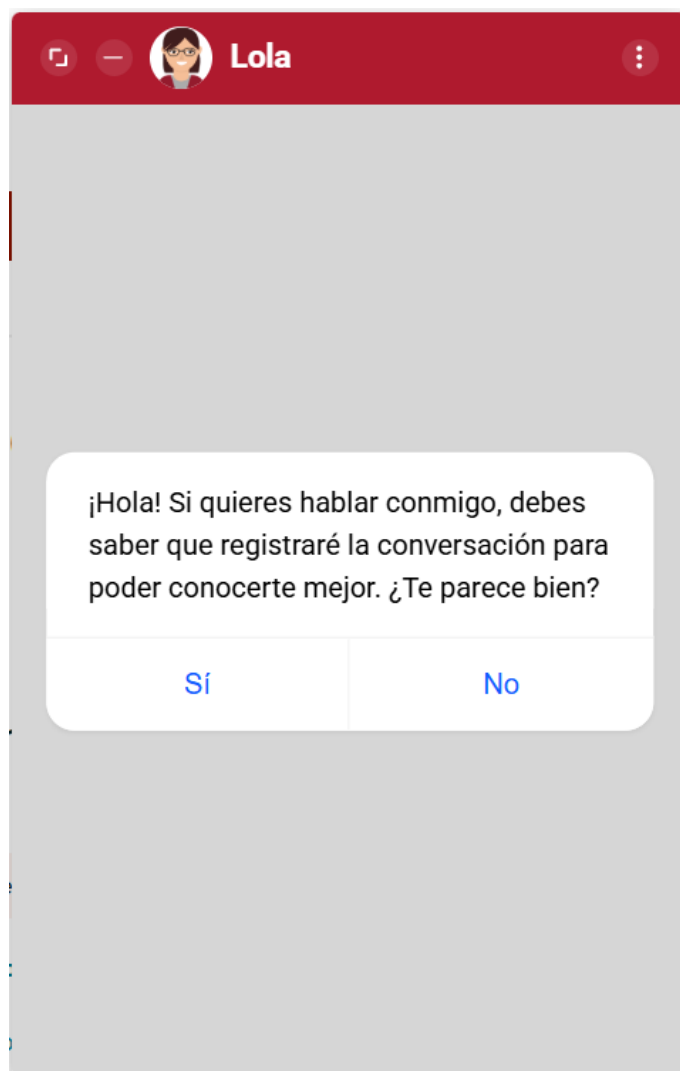


Рис. 1.12 Вікно роботи із чат-ботом

Після згоди на збереження даних, з'являється вікно самого чату (рисунок 1.13), де чат-бот представляється та каже, що окрім неї абітурієнт може запитати університет через пошту чи через телефонний дзвінок.

Також, чат-бот попереджає, що він працює в тестовому режимі і продовжує вдосконалюватися, тому надана ним інформація може бути помилковою. Слід зауважити, що вікно роботи із чат-ботом знаходиться прямо на сайті; також, вікно можна переміщати по інтерфейсу сайту та, за потреби, масштабувати.

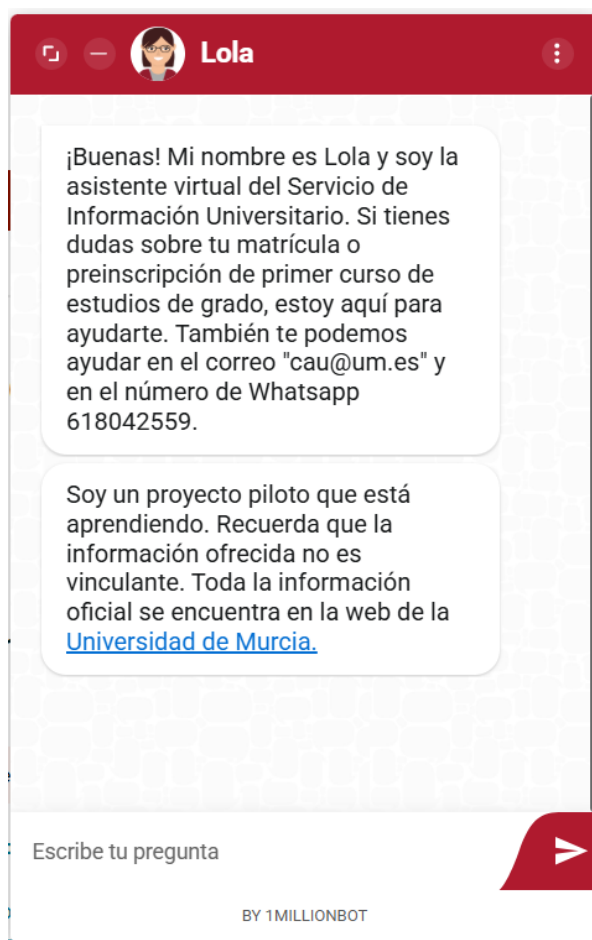


Рис. 1.13 Вікно чату

На більшість запитань, які ставить користувач, чат-бот надає посилання на спеціальні сторінки, на яких він зможе знайти відповідь. На рисунку 1.14 зображено відповідь чат-бота на питання «Перерахуй учбові спеціальності в університеті». У відповіді чат-бот надав дві кнопки, натиснувши на які, користувач переходить за посиланням.

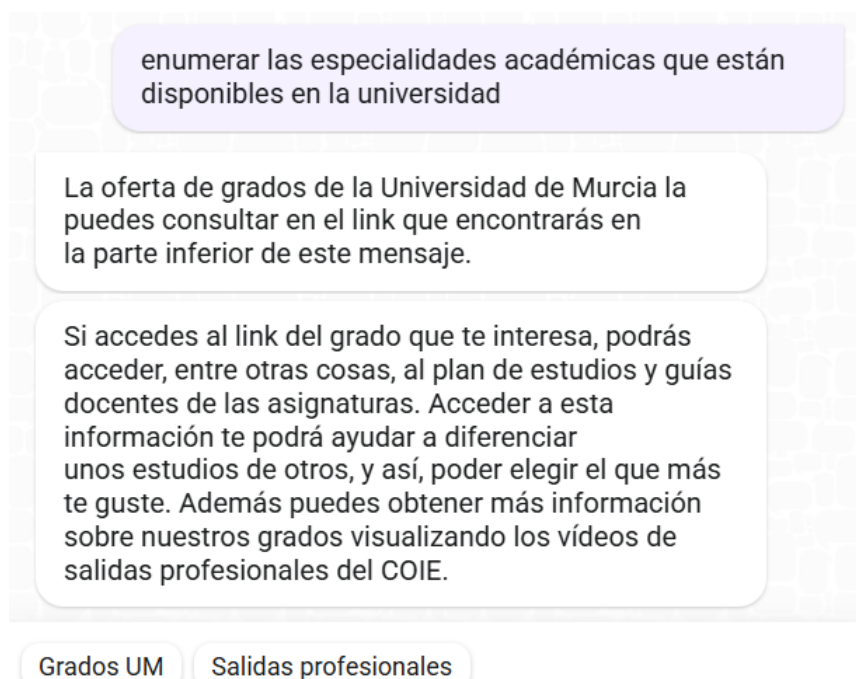


Рис. 1.14 Питання про спеціальності університету

На прохання надати дати подачі документів на вступ до університету, чат-бот надав у якості відповіді документ із датами етапів вступу за минулий рік.

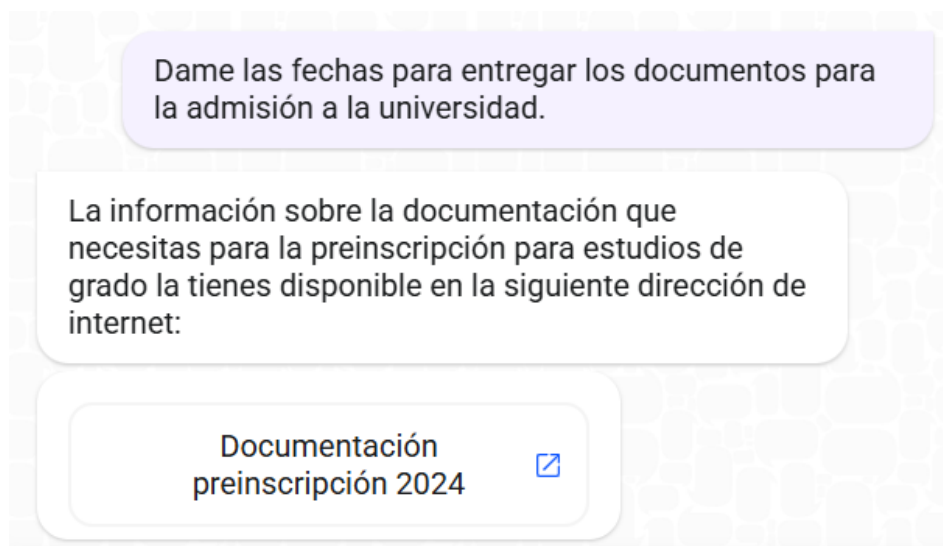


Рис. 1.15 Відповідь на питання про дати вступу

Також чат-бот може спілкуватися не тільки на теми, пов'язані зі вступом саме в цей університет. Одним із запитань до чат боту було «Що робити, якщо я боюсь вступу до університету?», на що чат-бот відповів у жартівливій

манері: «Страх – це боягуз. Запитай його «Чому?» й він одразу зникне.». Переписку зображено на рисунку 1.16.

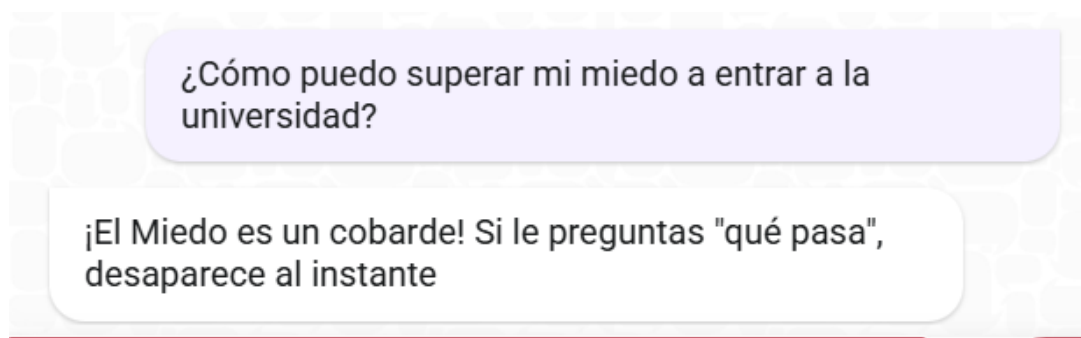


Рис. 1.16 Переписка про страх

Наступним існуючим аналогом було розглянуто чат-бот InfoChat Studenti від університету Верони(Італія).

InfoChat Studenti – чат-бот розроблений на базі технології IBM Watson, яка забезпечує діалог природньою мовою. Робота із чат ботом відбувається через вікно чату, який розташовано у правому нижньому кутку офіційного сайту Університету Верони. Абітурієнт може вести розмову, вибираючи готові варіанти відповідей, які пропонує чат бот, або ставити текстові запитання прямо у чат задля отримання швидкої та прямої відповіді. Чат-бот доступний на всіх профільних сторінках сайту та працює 24/7. Також, слід зауважити, те що чат-бот підтримує діалог двома мовами: англійська та італійська. [7]

Для активації чат-бота, як вже було вище зазначено, необхідно натиснути на спеціальну кнопку у правому нижньому кутку. Кнопка зображена на рисунку 1.17.



Рис. 1.17 Іконка активації чат-бота

Після натискання на кнопку активації, відкриється невеличке вікно роботи із чат-ботом, в якому бот вже надіслав стартове повідомлення. В повідомленні чат-бот представляє себе та одразу пропонує задати якесь питання. Меню роботи із чат-ботом зображено на рисунку 1.18.

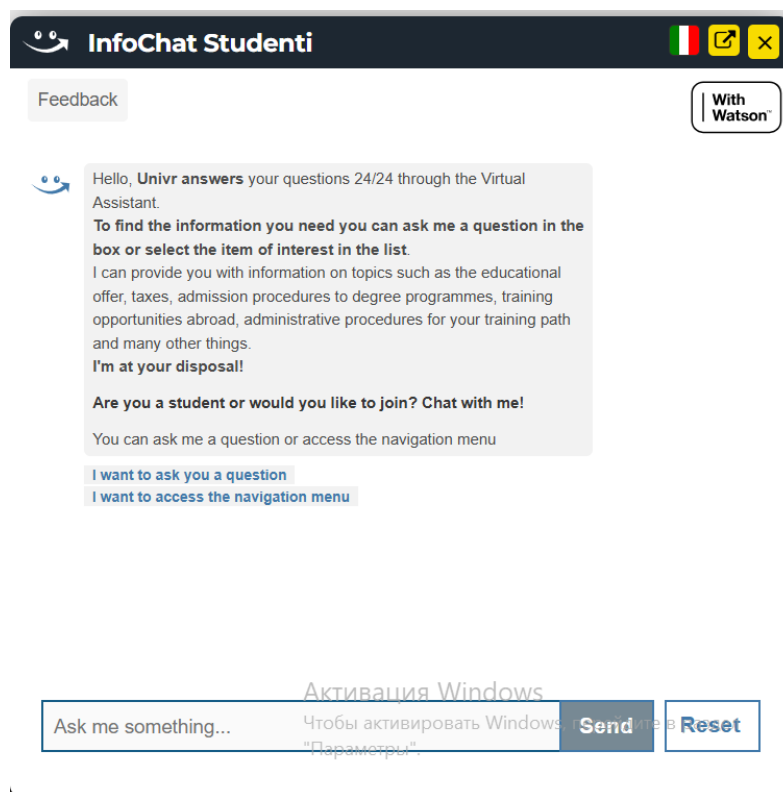


Рис. 1.18 Меню роботи із чат ботом

Чат-бот пропонує 2 варіанти початку діалогу:

- питання від абітурієнта;
- навігаційне меню для пошуку необхідної теми.

Навігаційне меню зображено на рисунку 1.19.

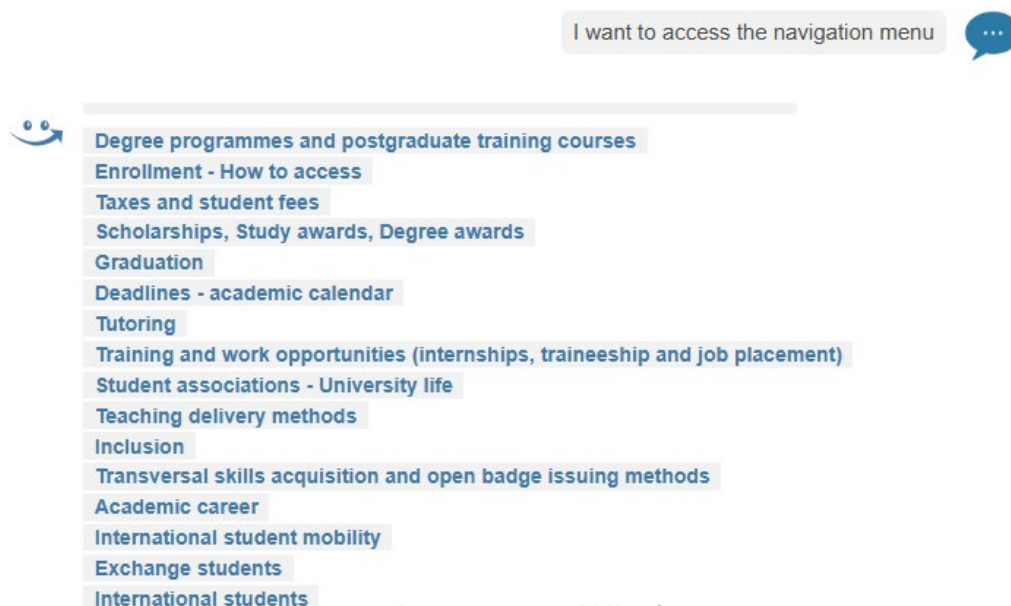


Рис. 1.19 Навігаційне меню

У випадку, якщо користувач обрав пункт, в якому більше немає підпунктів, то чат-бот надає користувачеві посилання на сторінку, де він може знайти необхідну інформацію (рисунок 1.20).

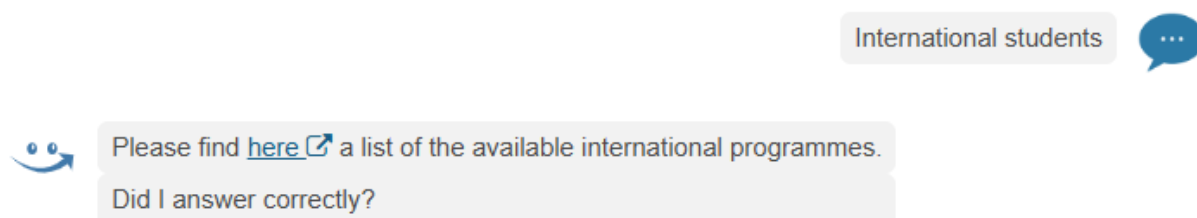


Рис 1.20 Відповідь на питання без підпунктів

У випадку, якщо у обраному пункті меню є ще і підпункти, які, в свою чергу, користувачеві запропонують ще один перелік підпунктів, які відносяться до обраної раніше теми (рисунок 1.21).

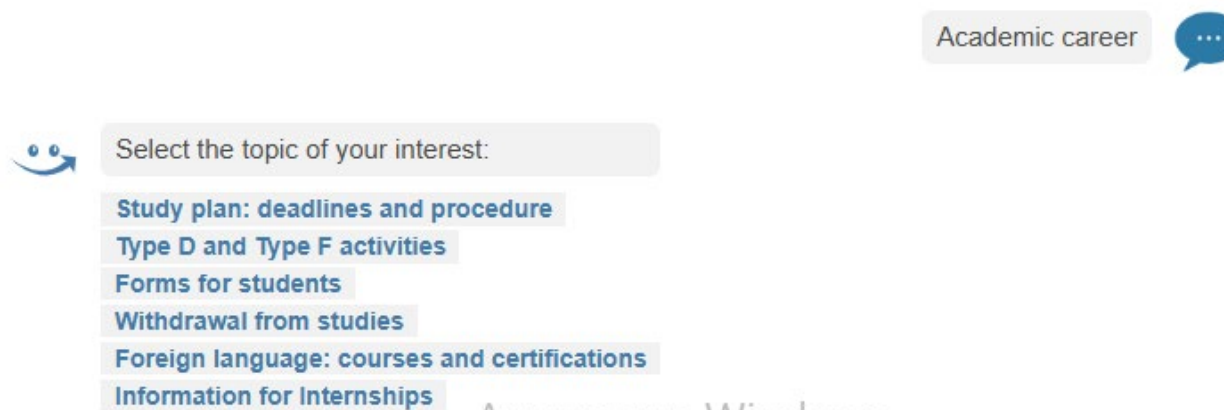


Рис. 1.21 Меню підпунктів, які відносяться до обраної теми

Якщо у самому початку роботи із чат-ботом обрани пункт «Задати питання», то чат-бот буде очікувати текстового питання від абітурієнта (рисунок 1.21).

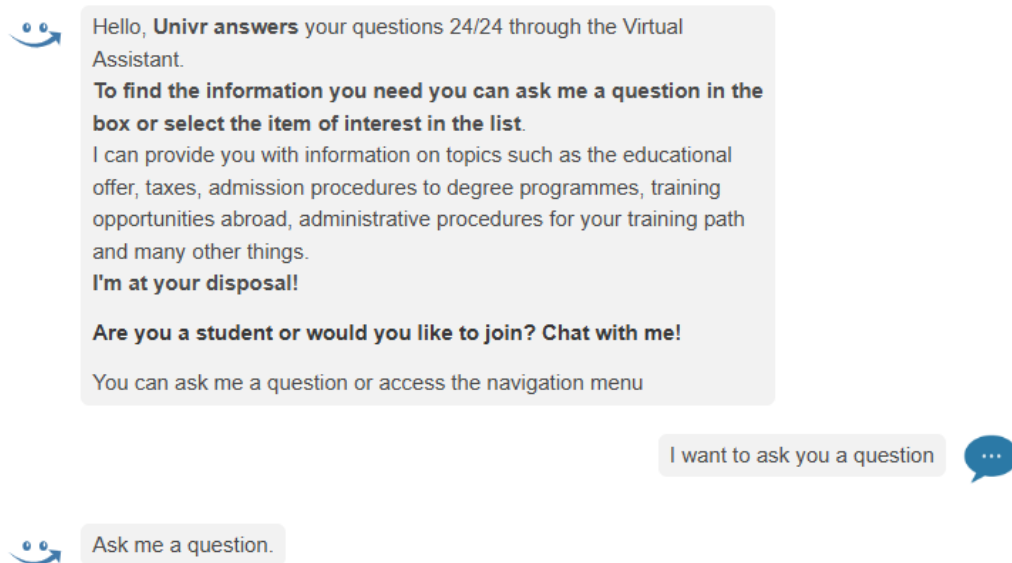


Рис. 1.21 Очікування чат-бота на текстове питання

Коли чат-бот отримує повідомлення, він одразу формулює відповідь, в якій надає пояснення стосовно питання абітурієнта та, за потреби, запитує додаткові питання для уточнення (рисунок 1.22).

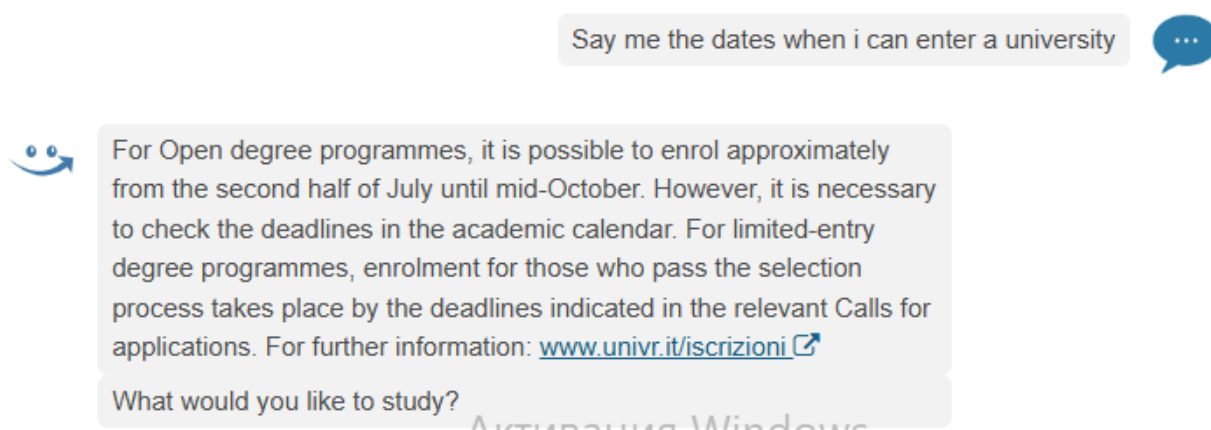


Рис. 1.22 Відповідь чат-бота на текстове запитання

У випадку, якщо чат-бот не зміг зрозуміти запитання абітурієнта, він виводить відповідне повідомлення, де просить перефразувати питання (рисунок 1.23).

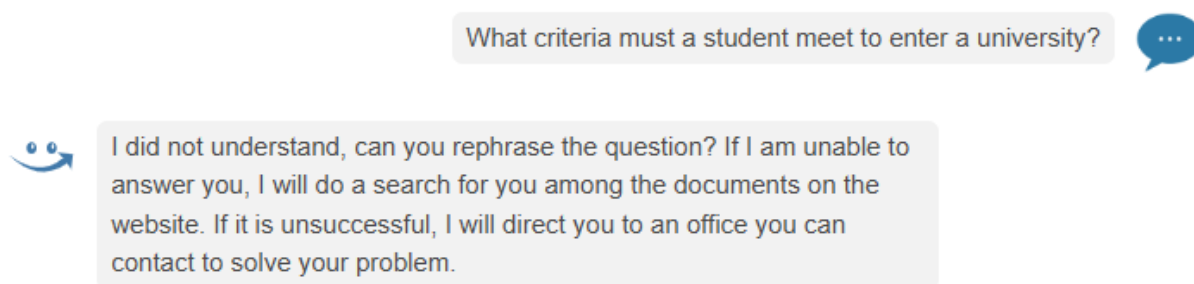


Рис. 1.23 Реакція на не зрозуміле повідомлення

Після проведення аналізу існуючих рішень, було підведено підсумки у вигляді таблиці, в якій порівнюються характеристики існуючих аналогів та розроблюваного чат-боту.

Таблиця 1.1

Порівняння характеристик

Характеристика	НТУ «ХП»	Lola	InfoChat Studenti	ChatBotUni
Пошук за ключовими словами	-	-	-	+
Інтерактивний онлайн-чат	+	+	+	+
Збереження історії діалогу	+	+	-	+
Українська локалізація	+	-	-	+
Меню навігації	-	-	+	+
Інформація про кафедру	+	-	-	+
Інформація про вступ	+	+	+	+
Онлайн чат із представником університету	-	-	-	+

1.3 Вимоги до програмного забезпечення

В цьому розділі буде визначено наступні типи вимог до розроблюваного програмного забезпечення[5]:

- функціональні вимоги;
- не функціональні вимоги;
- вимоги до інтерфейсу.

Задля зручності, функціональні та нефункціональні вимоги будуть зображені у вигляді таблиць зі стовпцями «Вимога» та «Пояснення».

Таблиця 1.2

Функціональні вимоги

Вимога	Пояснення
Чат-бот повинен реагувати на повідомлення	Чат повинен аналізувати отримане повідомлення та робити відповідні дії, пов'язані із повідомленням
Чат-бот повинен зберігати історію чату	Чат між чат ботом та користувачем повинен зберігатися для майбутнього продовження діалогу чи для повторного перегляду користувачем відповідей
Функція пошуку в чат-боті	Чат-бот повинен мати функцію пошуку тем за ключовими словами
Кнопка виклику оператора	У разі, якщо користувачу не зміг допомогти чат-бот, повинна бути кнопка, що викликає працівника кафедри, який допомагає користувачеві отримати відповідь на питання
Чат між оператором та користувачем	Програмне забезпечення повинно містити функціонал онлайн чату у реальному часі між двома людьми
Валідація повідомлень	Чат повинен забезпечувати валідацію повідомлень користувача

Таблиця 1.3

Нефункціональні вимоги

Вимоги	Пояснення
Швидкодія	Швидкість відповіді чат-боту не повинна перевищувати 5 секунд
Доступність	Чат-бот повинен бути доступним для користувача протягом більше як 99% часу
Безпека	Чат-бот повинен містити захист від DDOS атак та від несанкціонованого доступу
Повідомлення про помилку	Чат-бот у разі виникнення помилок, виводить відповідне повідомлення

Вимоги до інтерфейсу:

- інтерфейс повинен бути зручним та інтуїтивно зрозумілим;
- інтерфейс повинен містити усі базові елементи для роботи із чатом;
- наявність перемикача між режимом онлайн-чату з оператором та чат-ботом;
- в локалізації інтерфейсу використовується українська мова.

1.4 Постановка задачі

Розробити веб-сторінку та чат-бота, який буде інтегрований у цю сторінку, функціоналом бота буде допомога із ознайомленням з кафедрою та надання відповідей на поширені питання абітурієнтів і їх батьків в процесі вибору університету для вступу. Також, чат бот повинен мати опцію перемикавання, під час якої абітурієнт може напяму зв'язатися із представником університету у чаті та задати йому питання у режимі реального часу.

Чат-бот повинен реалізовувати наступний функціонал:

- пошук гілок тем за ключовими словами;
- підготовка сценаріїв спілкування із користувачем;
- формування конструктивних відповідей;
- можливість виклику представника університету.

Для оптимальної роботи чат-боту, він повинен володіти наступною інформацією:

- інформація, пов'язана зі вступом (процес вступу, необхідні документи). В контексті даної дипломної роботи, було розглянуто лише особливості вступу на освітній рівень «Бакалавр»;
- інформація, пов'язана із описом кафедри;
- інформація, пов'язана із роботою кафедри;
- інформація, пов'язана із описом спеціальності 121 (які предмети вивчаються, скільки років триває навчання, відображення інформаційної картки спеціальності тощо);
- інформація про процес навчання.

Для більшої універсальності, чат-боту необхідно також додати функцію, яка дозволяє викликати представника університету для особистої розмови. Переписка з представником університету відбувається в тому ж інтерфейсі, що і переписка з ботом. Для представника університету повинно бути реалізоване окреме вікно, де буде приходити сповіщення про отримані питання, на які він зможе відповідати в режимі реального часу.

1.5 Технології реалізації

Провівши аналіз поставленої задачі та вимог до програмного продукту, було проведено аналіз наявних технологій, за допомогою яких можна реалізувати чат-ботів. Також, було наступні технології, які будуть використані під час реалізації програмного продукту:

- ASP.Net Core C#;

- Bot Framework SDK C#;
- SignalR C#;
- React;
- Web Chat SDK;
- MongoDB;
- Azure.

Тепер розглянемо кожну із перелічених технологій та виокремимо причини, чому було обрано саме ці технології.

Asp.Net Core [6] – це кросплатформне веб-середовище для розробки веб-застосунків та служб, створене компанією Microsoft. Основною мовою для роботи із цим середовищем слугує мова програмування C#[7]. В даному дипломному проєкті це середовище використовується для реалізації серверної частини застосунку..

Це веб-середовище було обрано через зручність з інтеграцією з Bot Framework SDK та легкістю у публікації проєкту на Azure. Також Asp.Net Core є потужним та надійним середовищем розробки, що зіграло вагому роль при його виборі.

Bot Framework SDK – це платформа з відкритим кодом, за допомогою якої можна створювати ботів різних типів. Платформа підтримує такі мови програмування, як C#, JavaScript, Java та Python.[8] Ця платформа в проєкті використовується для безпосередньо створення бота.

SignalR – бібліотека з відкритим кодом, яка реалізує виконання функціоналу в реальному часі та дозволяє за допомогою веб-сокетів створити миттєвий зв'язок між клієнтською та серверною частинами застосунку.. Бібліотека використовується для реалізації застосунків з виконанням функцій у реальному часі, в тому числі й месенджерів.[9] Ця бібліотека буде застосовуватися в даному дипломному проєкті для реалізації чату між абітурієнтом та оператором.

React – це гнучка JavaScript-бібліотека, призначенням якої є створення інтерфейсів користувача, за принципом складання компонентів. Компоненти в даному випадку це дрібні частинки коду. Ця бібліотека використовується коли необхідно створити Single page-застосунок. [10] В даному дипломному проєкті цю бібліотеку буде використано для створення інтерфейсу користувача для роботи із чат-ботом.

Web Chat SDK – фронтенд-бібліотека, яка допомагає розробнику інтегрувати чат-бот в інтерфейс веб-застосунку. Використання цієї бібліотеки пришвидшує швидкість розробки за рахунок того, що має в собі деякі заготовлені наперед компоненти.[11]. В контексті даного дипломного проєкту, ця технологія буде використовуватися під час розробки інтерфейсу.

MongoDB – документно-орієнтована база даних, яка відноситься до сімейства NoSQL. Різниця MongoDB та SQL-баз даних полягає у тому, що в MongoDB дані зберігаються колекціями, а не таблицями. Колекції зберігаються в форматі «ключ-значення».[12] В контексті даного дипломного проєкту, ця технологія використовується, як база даних.

Azure – хмарний сервіс обчислень та різноманітних служб. Сервіс допомагає з розгорненням застосунку у хмарі.[13] В контексті даного дипломного проєкту, цей сервіс використовується у якості хостингу.

2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ ПІДТРИМКИ ОНЛАЙН-ЧАТУ НА САЙТІ УНІВЕРСИТЕТУ

2.1 Проєктування архітектури системи

Провівши аналіз більшості популярних архітектур, було обрано так звану «чисту» архітектуру.

Чиста архітектура – підхід до проєктування програмного забезпечення, розроблений Робертом Мартіном, відомим під псевдонімом «Uncle Bob». Ідея архітектури полягає в чіткому розділенні відповідальностей шляхом створення шарів у структурі системи. Такий підхід забезпечує ізоляцію бізнес-логіки від деталей реалізації, що дає змогу легко змінювати зовнішні компоненти, такі як UI, базу даних або фреймворки, без впливу на внутрішню роботу системи.[14]

Для більшого розуміння щодо того, як необхідно правильно реалізовувати чисту архітектуру у системах, необхідно правильно розуміти основні принципи такої архітектури:

- вихідний код зовнішніх рівнів може залежати від коду внутрішніх рівнів – але не навпаки. Тобто, внутрішній шар повинен працювати таким чином, щоб нічого не знати про зовнішній шар;
- архітектура не повинна залежати від фреймворків, які будуть використовуватися у системі;
- внутрішній шар жодним чином не залежить від інтерфейсу користувача;
- внутрішній шар жодним чином не залежить від обраної бази даних для збереження даних;
- внутрішній шар не повинен залежати від зовнішніх бібліотек, сервісів, служб чи API.

Підбиваючи підсумок, найголовнішим принципом є ізолюваність внутрішнього шару системи. В контексті даного дипломного проєкту таким шаром є шар ядра чат-боту. Згідно до принципів чистої архітектури, було спроектовано наступну архітектуру:

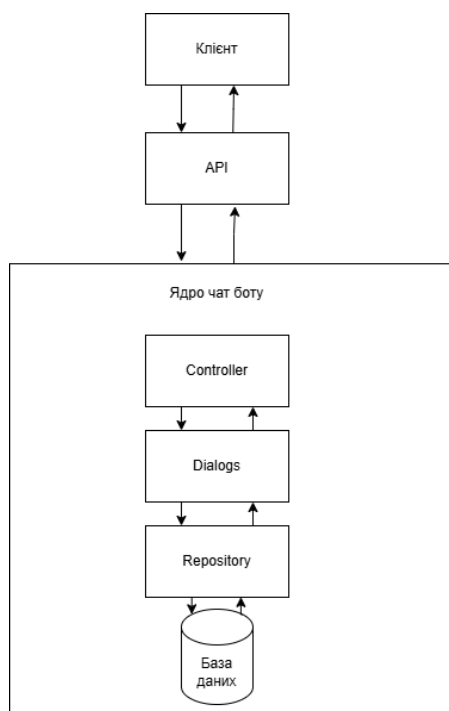


Рис. 2.1 Архітектура системи

Шар «Клієнт» - це зовнішній шар, який відповідає за веб-інтерфейс користувача, в якому користувач робить запити у систему та отримує результат.

Шар «API» - це зовнішній шар, який відповідає за зовнішній API чат-боту. Тобто, цей шар отримує запит та перенаправляє у внутрішній шар.

Шар «Контролер» - це внутрішній шар, який отримує вхідні дані від зовнішнього API, після чого перетворює їх у формат внутрішнього шару та розпочинає сценарій виконання роботи.

Шар «Діалоги» - це шар бізнес-логіки, який відповідає за реалізацію діалогів чат-боту та реалізацію режиму онлайн-чату між абітурієнтом та представником університету через чат-боту.

Шар «Репозиторії» - це шар-посередник між чат-ботом та базою даних. Шар побудований таким чином, щоб не залежати від конкретної бази даних.

Шар «База даних» - найнижчий зовнішній шар системи, який відповідає за базу даних, яка зберігає необхідні дані для роботи системи.

Підсумовуючи, архітектура відповідає принципам чистої архітектури, адже спроектована таким чином, щоб ядро чат-боту не залежало від реалізації інтерфейсу користувача та не залежало від вибору бази даних.

2.2 Проектування бази даних

Для аналізу ефективності чат-боту, необхідно мати базу даних, яка буде зберігати історію чатів користувачів. Також, база даних потрібна для виконання авторизації користувачів. На рисунку 2.2 зображена ER-діаграма бази даних.

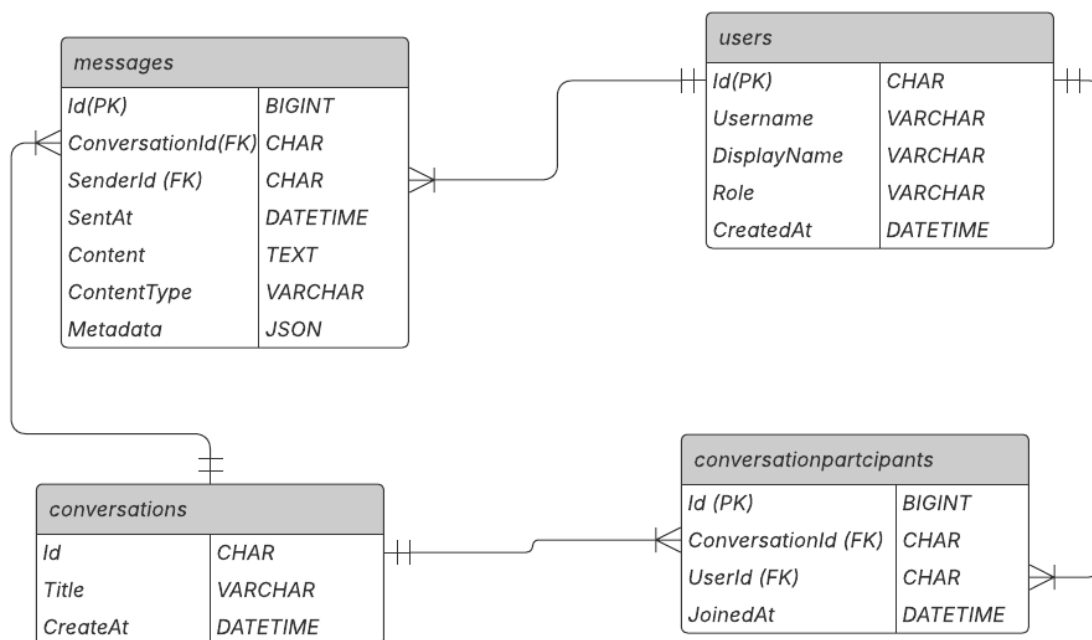


Рис 2.2 ER-діаграма бази даних

Тобто, для повноцінної роботи системи було виділено 4 сутності:

- users для зберігання інформації про користувачів системи;
- conversations для зберігання інформації про бесіду;
- messages для зберігання повідомлень у бесіді;
- conversationparticipants для зберігання інформації про учасників бесіди.

На основі спроектованої ER-діаграми було спроектовано діаграму схеми JSON для майбутньої бази даних у MongoDB.

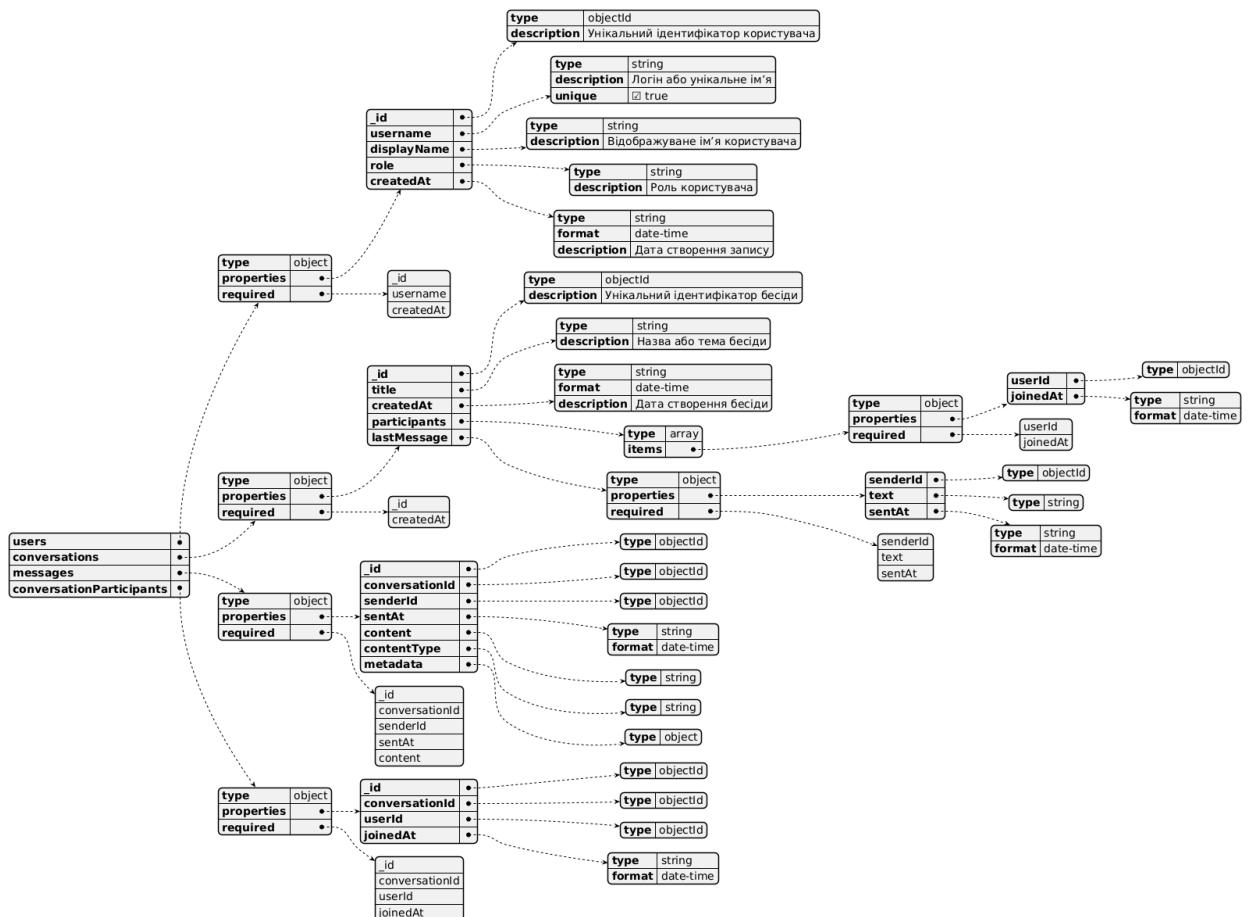


Рис. 2.2 Діаграма схеми JSON

В основі діаграми лежить чотири об'єкти, які є колекціями у MongoDB:

- users;
- conversations;

- messages;
- conversationParticipants.

Кожна колекція є типом «об'єкт» та містить блок полів, для якого вказується перелік обов'язкових для заповнення полів.

Розглянемо колекцію users. Колекція має наступні п'ять полів:

- `_id` – поле, яке відповідає за збереження первинного ключу запису;
- `username` – унікальне поле, що відповідає за збереження логіну користувача;
- `displayName` – поле, яке відповідає за збереження імені та прізвища користувача;
- `role` – поле, яке відповідає за збереження ролі користувача (в контексті даного дипломного проєкту в систему впроваджено 2 ролі – а саме, Адміністратор та Користувач);
- `createdAt` – поле, яке відповідає за збереження дати та часу створення запису.

Розглянемо колекцію conversations. Колекція має наступні 4 поля:

- `_id` – поле, яке відповідає за збереження первинного ключу запису;
- `title` – поле, яке відповідає за збереження назви бесіди;
- `createdAt` – поле, яке відповідає за збереження дати та часу створення запису;
- `participants` – вбудований масив учасників бесіди, в якому як мінімум один учасник;
- `lastMessage` – вбудований об'єкт повідомлення, який відповідає за збереження останнього повідомлення у бесіді.

Розглянемо колекцію messages. Колекція має наступні 7 полів:

- `_id` – поле, яке відповідає за збереження первинного ключу запису;
- `conversationId` – поле, яке відповідає за збереження первинного ключа бесіди;

- `senderId` – поле, яке відповідає за збереження первинного ключа відправника повідомлення (у випадку, якщо відправник бот, то туди буде вкладатися заздалегідь заброньований первинний ключ бота);
- `sentAt` – поле, яке відповідає за збереження дати та часу відправки повідомлення;
- `content` – поле, яке відповідає за вміст повідомлення;
- `contentType` – поле, яке відповідає за тип вмісту повідомлення;
- `metadata` – поле, яке відповідає за додаткові дані, прикріплені до повідомлення (метадані).

Розглянемо колекцію `conversationParticipants`. Колекція має наступні 4 поля:

- `_id` – поле, яке відповідає за збереження первинного ключу запису;
- `conversationId` – поле, яке відповідає за збереження первинного ключа бесіди;
- `userId` – поле, яке відповідає за первинний ключ учасника бесіди;
- `joinedAt` – поле, яке відповідає за збереження дати та часу підключення користувача до бесіди.

В цілому, провівши аналіз колекцій, можна виділити наступні зв'язки між ними:

- 1 `users` до багатьох `messages`;
- 1 `conversations` до багатьох `messages`;
- зв'язок «багато до багатьох» між `users` та `conversations` через колекцію `conversationParticipants`.

2.3 Діаграма варіантів використання

Діаграма варіантів використання (або англійською «Use Case Diagram»)

– це діаграма, яка допомагає моделювати поведінку системи. Така система допомагає відобразити вимоги до програмної системи для різних користувачів (акторів) системи.

В даному дипломному проєкті можна виділити 2 актори:

- адміністратор, (представник університету);
- користувач, (абітурієнт).

Для більш наочного вигляду діаграми варіантів використання, її було розбито на 2 окремих рисунки.

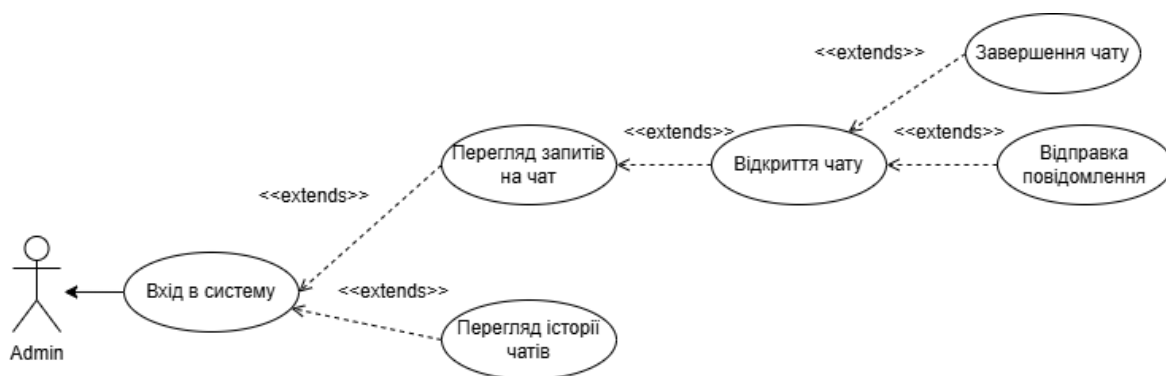


Рис. 2.3 Діаграма варіантів використання для актора «Адміністратор»

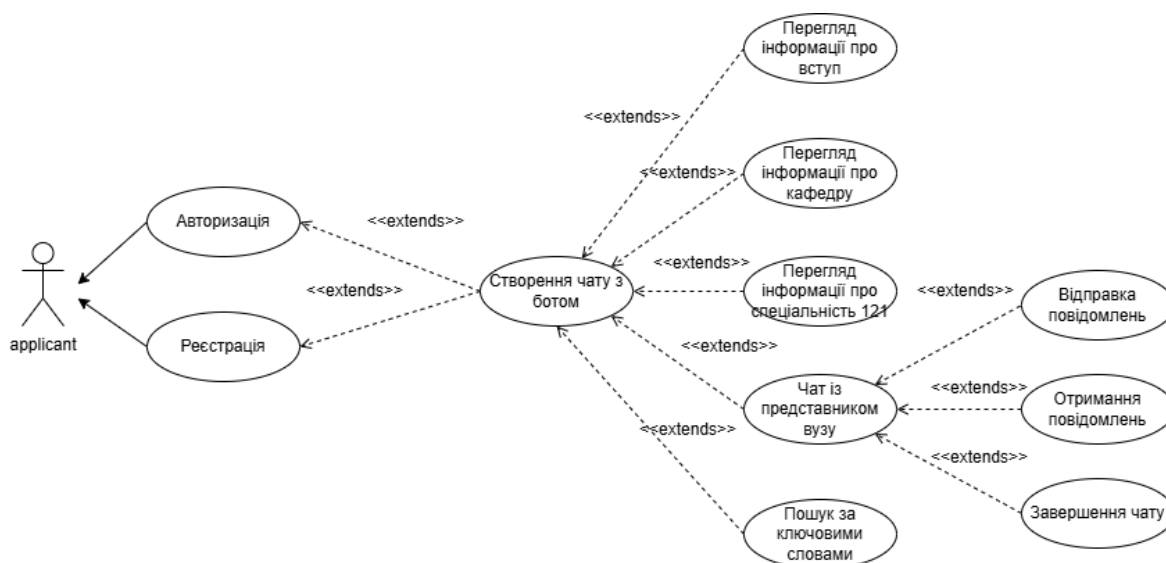


Рис. 2.4 Діаграма варіантів використання для актора «Абітурієнт»

3. ОПИС РЕАЛІЗАЦІЇ

3.1 Структура системи

Для реалізації чат-боту було використано середовище розробки Visual Studio. В цьому середовищі розробки було створено основні класи та підключені основні бібліотеки для роботи з чат ботом.

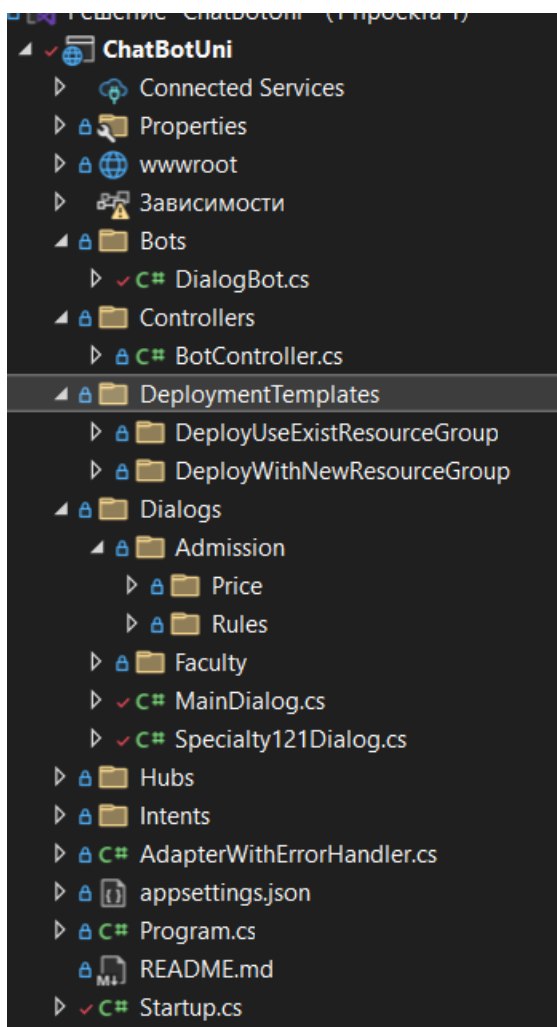


Рис. 3.1 Структура розробленого проєкту

Реалізацію чат-боту можна розподілити на наступні компоненти:

- контроллер, який приймає запити ззовні;
- бот, який отримує запити;

- діалоги, які реалізують взаємодію із користувачем;
- хаб, який відповідає за чат між представником університету та абітурієнтом у реальному часі.

Папка `Connected Services` зберігає в собі налаштування зовнішніх сервісів (по типу `Azure`).

Папка `Properties` зберігає файли конфігурації проєкту.

Папка `wwwroot` зберігає необхідні статичні ресурси для роботи проєкту (фотографії, сторінки й так далі).

Папка `Dependencies` зберігає перелік залежностей, встановлених через пакетний менеджер `NuGet`, та відповідає за управління пакетами під час компіляції проєкту.

Папка `Bots` зберігає класи, що реалізують основу `Bot Framework`. Відповідає за обробку вхідних даних та виклик основного діалогу.

Папка `Controllers` зберігає класи-контролери та відповідає за формування `HTTP-endpoint`, за яким відбувається доставка повідомлення до чат-бота.

Папка `DeploymentTemplates` зберігає `ARM`-шаблони та інші файли, які необхідні для розгортання середовища. Відповідає за автоматичне розгортання бота у хмарних ресурсах.

Папка `Dialogs` зберігає усі файли діалогів, які реалізують логіку спілкування чат-боту та користувача.

Папка `Hubs` містить в собі `SignalR`-хаби, які відповідають за реалізацію чату між представником університету та користувачем у режимі реального часу.

Папка `Intents` містить в собі обробники подій у діалогах, які були винесені із діалогів.

Для реалізації веб-сайту для роботи із чат ботом та спілкування у режимі реального часу, було використано середовище розробки `Visual Studio Code` та бібліотеку `React` для створення інтерфейсу.

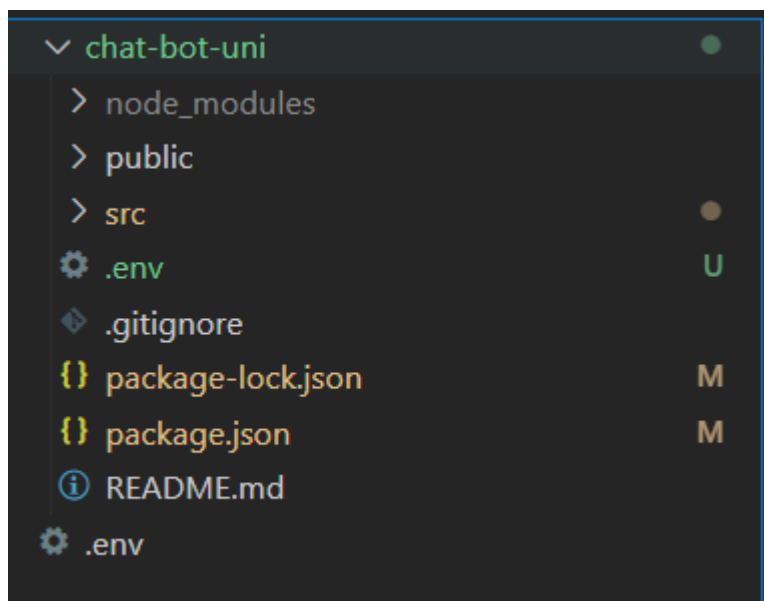


Рис. 3.2 Структура чат-боту

Уся основна взаємодія між чат-ботом та клієнтською стороною виконується за допомогою React та чистого JavaScript, а у якості стилів для шаблону сайту використовується бібліотека bootstrap.

Папка `node_modules` зберігає у собі усі встановлені залежності проєкту та відповідає за надання встановлених бібліотек та фреймворків під час роботи.

Папка `public` зберігає статичні файли, які не оброблюються в ході роботи, а одразу копіюються в кінцеву папку збірки.

Папка `src` містить увесь основний код для роботи фронтенд-частини.

Файл `.env` містить у собі змінні оточення, а саме ключі API.

3.2 Діаграма класів

Діаграма класів – це діаграма, яка візуалізує усі класи у системі та взаємозв'язки між ними. Діаграма класів є однією з найкорисніших діаграм на усіх етапах розробки об'єктно-орієнтованої системи.[16]

Для кращого розуміння системи, було побудовано дві діаграми класів:

- діаграма класів основної бізнес-логіки;

- діаграма класів чатингу.

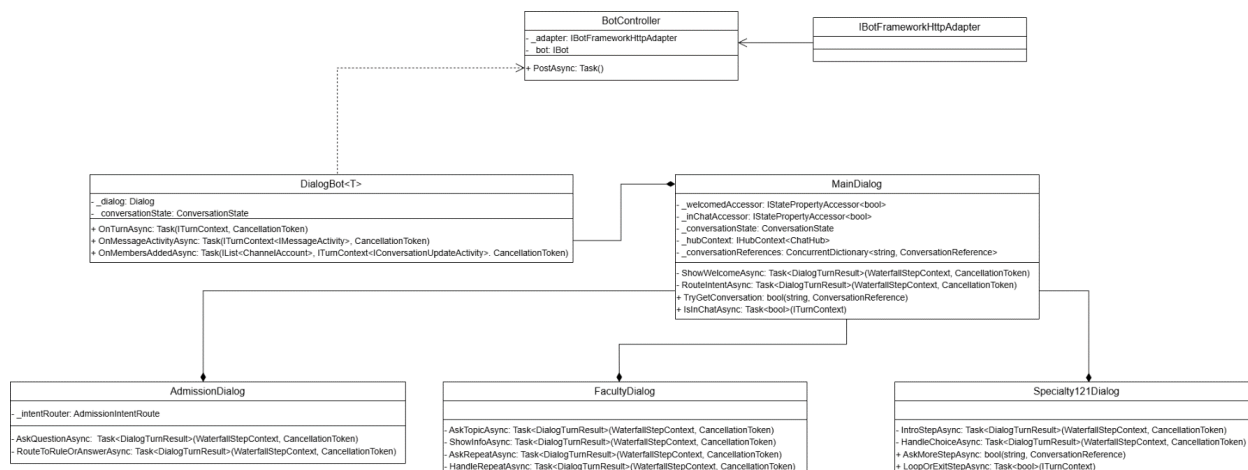


Рис. 3.3 Діаграма класів основної бізнес логіки

На рисунку 3.3 зображено діаграму основних класів системи.

Клас BotController відповідає за отримання вхідного запиту від клієнта у endpoint та подальший виклик бота у класі DialogBot.

Клас DialogBot реалізовує чат-бота. Клас приймає в собі клас MainDialog та викликає його для початку діалогу із користувачем.

Клас MainDialog відповідає за початкове повідомлення для користувача та навігацію по подальшим діалогам. Крім того, у цьому класі реалізовано виклик функціоналу, пов'язаного із онлайн-чатом між користувачем та представником університету.

Класи AdmissionDialog, FacultyDialog та Specialty121Dialog – класи діалогів, які слугують основними класами для розподілу діалогів за категоріями. Класи відповідають за наступні категорії напрямів ведення діалогу:

- інформація про вступ;
- інформація про кафедри університету;
- інформація про спеціальність 121.

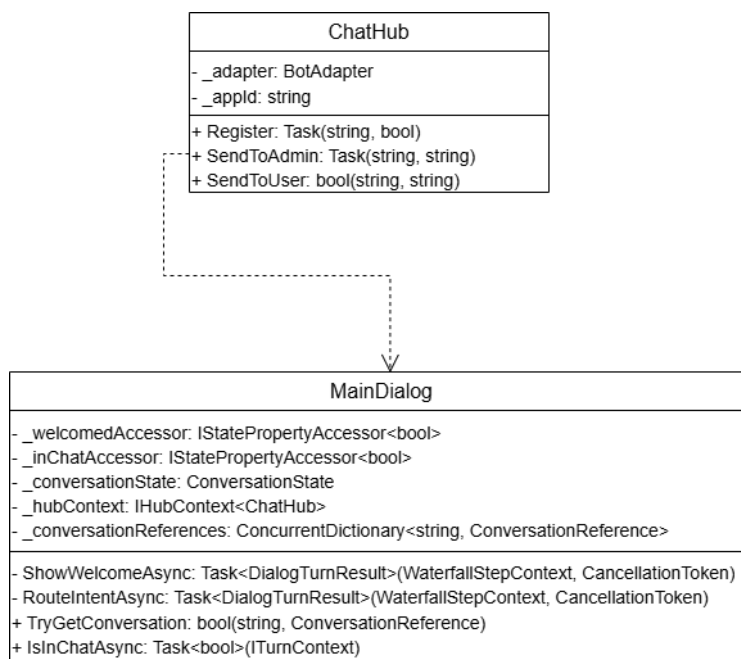


Рис. 3.4 Діаграма класу чатингу

На рисунку 3.4 зображено діаграму основних компонентів, які реалізують онлайн-чат у реальному часі між представником університету та користувачем чат-боту.

Клас ChatHub відповідає за створення сесії та передачу повідомлень між користувачами за допомогою фреймворку SignalR.

Клас MainDialog є класом, який приймає повідомлення від представника університету через ChatHub та відображає їх користувачеві чат-бота. Також, через цей клас виконується запит до адміністратора про чат із клієнтом.

3.3 Реалізація функціоналу чат-боту

В цьому розділі буде продемонстровано основну реалізацію діалог чат-бота та онлайн-чату у реальному часі між представником університету та користувачем. Тобто, буде продемонстровано реалізацію ключових сценаріїв повідомлень та роботу чату між представником університету та користувачем.

Спочатку було реалізовано запуск чат-боту та його початкове повідомлення користувачеві, яке зможе направити його до релевантної для даного користувача теми.

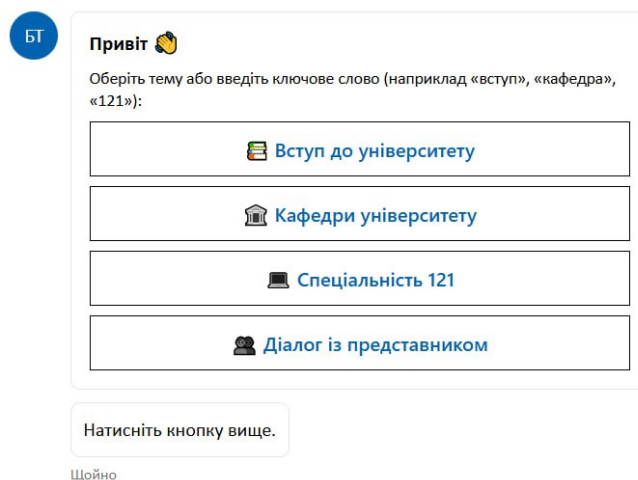


Рис. 3.5 Початкове повідомлення

За відображення цього повідомлення відповідає клас MainDialog. Після отримання повідомлення користувач має 2 варіанти дій: натиснути на кнопку, або ж ввести ключове слово, щоб одразу розпочати діалог за потрібною тематикою.

Якщо користувач хоче переглянути інформацію про спеціальність 121, йому одразу буде виведено нове повідомлення, де йому необхідно натиснути на відповідну кнопку.

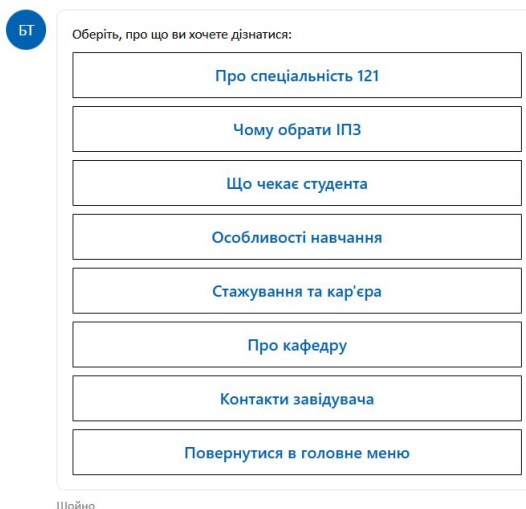


Рис. 3.6 Навігаційне повідомлення для спеціальності 121

Відображення цього повідомлення реалізовано за допомогою класу Speciality121Dialog. Виведене повідомлення містить інформативні кнопки, натиснувши на які, користувач отримає відповідну інформацію, та кнопку навігації «Повернутися на головне меню», яка поверне користувача до головного меню.

Також, якщо користувач натиснув на інформаційну картку, то після цього він отримає повідомлення, де його запитують, чи бажає він ще щось додатково переглянути про 121 спеціальність. Якщо користувач відповів «Ні», то він потрапляє у головне меню, а у випадку «Так» ще раз потрапляє до повідомлення на рисунку 3.6.



Рис. 3.7 Відповідь на питання про контакти завідувача

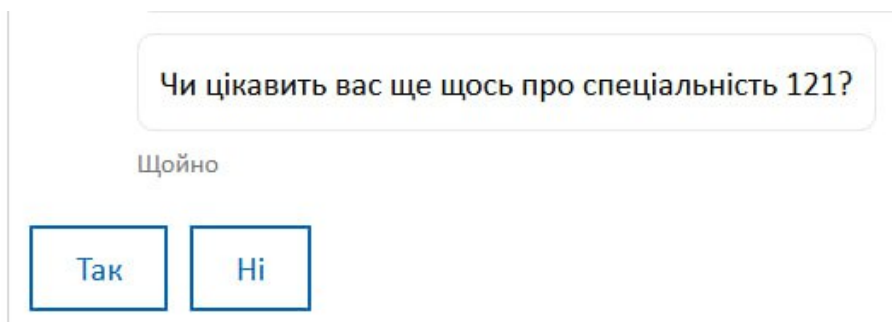


Рис. 3.8 Питання до користувача: чи хоче він ще щось переглянути про 121 спеціальність

Якщо користувача цікавлять правила вступу до університету, то у головному навігаційному повідомленні (рисунок 3.5) користувач повинен натиснути на кнопку «Вступ». Після чого, користувача попросять вказати, з чим саме пов'язане його питання (рисунок 3.9).

Це реалізовано за допомогою класу `AdmissionDialog`, в якому аналізується питання Користувача, після чого він переадресовується у потрібну гілку діалогу.



Рис. 3.9 Діалог питання щодо вступу

Якщо користувач обрав частину діалогу, пов'язану із правилами вступу, то він потрапляє у діалог «Правила вступу». Цей діалог реалізовано за допомогою `AdmissionRulesDialog`, який аналізує питання та відправляє його у `Handlers`, які в свою чергу генерують повідомлення-відповідь.

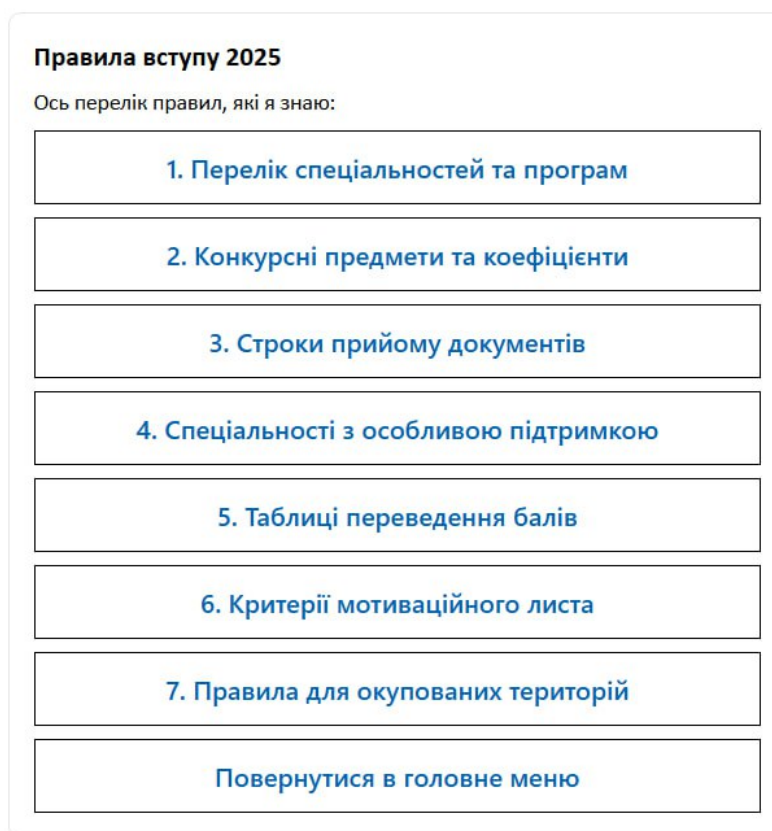


Рис. 3.10 Меню з правилами вступу

Розглянемо сценарій, в якому абітурієнт цікавиться датами подачі документів для вступу. Перше, що запитає в нього чат-бот, це на яку форму навчання розраховує абітурієнт – на бюджетну чи на контрактну (рисунок 3.11).

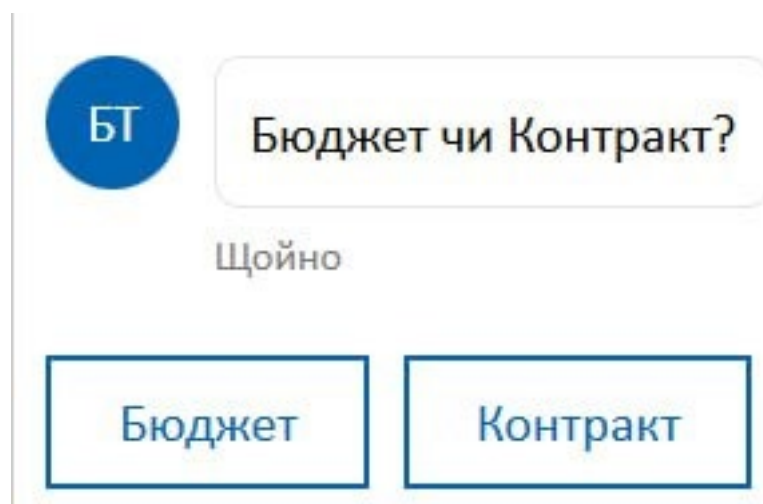


Рис. 3.11 Меню вибору «Контракт» чи «Бюджет»

Розглянемо сценарій, в якому користувач обрав варіант «Бюджет» (рисунок 3.12).

The image shows a chatbot interface. At the top left is a blue circular icon with the letters 'БТ'. To its right is a white rounded rectangle containing the text 'Етапи вступної кампанії 2025 (Бюджет)'. Below this is a table with two columns: '№ п/п Етап' and 'Термін'.

№ п/п	Етап	Термін
1	Реєстрація електронних кабінетів вступників, завантаження необхідних документів	з 01 липня
2	Реєстрація заяв на участь у співбесідах — на бюджет	з 03 до 18 ^{го} 10 липня
3	Строки проведення співбесід — на бюджет	з 08 до 19 липня
4	Прийом заяв та документів у електронній або паперовій формі	з 19 липня до 18 ^{го} 01 серпня
5	Оприлюднення списку рекомендованих до зарахування за держзамовленням	до 06 серпня
6	Виконання вимог до зарахування (договір про навчання)	до 18 ^{го} 09 серпня
7	Зарахування вступників за держзамовленням	до 11 серпня
8	Надання рекомендацій за кошти фізичних/юридичних осіб	11 серпня
9	Виконання вимог вступниками за контрактом	до 18 ^{го} 14 серпня
10	Зарахування вступників за кошти фізичних/юридичних осіб	до 16 серпня
11	Переведення на вакантні місця держзамовлення	18 серпня

Рис. 3.12 Терміни подачі документів для бюджету

Наступним розглянемо сценарій, коли користувач хоче переглянути інформацію про кафедру. Для початку, користувач повинен у головному навігаційному меню натиснути на одну з кнопок, відображених на рисунку 3.5. Після чого, він побачить повідомлення, де йому буде запропоновано обрати навчально-науковий інститут, який відноситься до його кафедри (рисунок 3.13).



Рис. 3.13 Навігаційне повідомлення із навчально-науковими інститутами

Припустимо, користувач обрав «Навчально-науковий інститут Інформаційних технологій». Після вибору навчально-наукового інституту, користувача попросять обрати кафедру, яка його цікавить (рисунок 3.14).

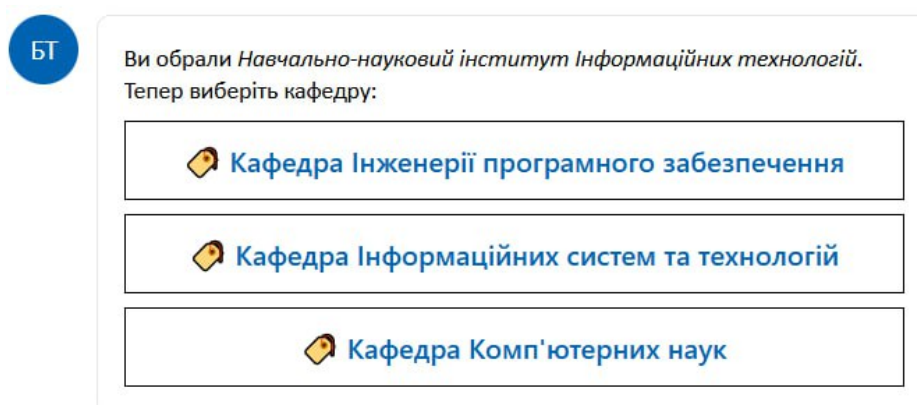


Рис. 3.14 Навігаційне меню кафедр

Обравши «Кафедра Інженерії програмного забезпечення», користувач переходить до нового навігаційного меню, в якому він може переглянути всю інформацію про обрану кафедру (рисунок 3.15).

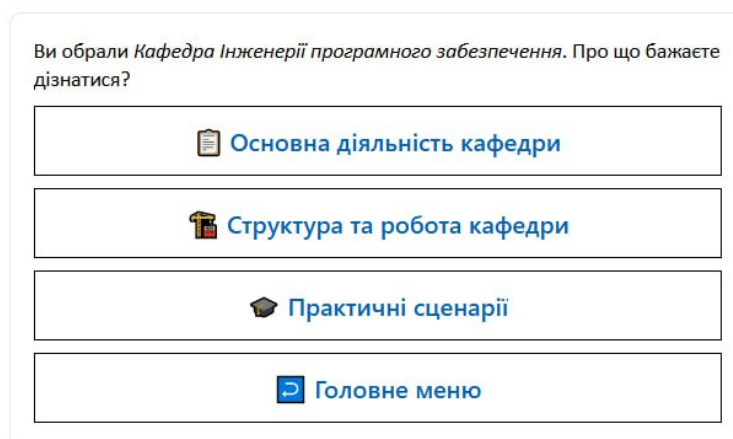


Рис. 3.15 Навігаційне меню інформації про кафедру

В «Основна діяльність кафедри» користувач може ознайомитися із наступною інформацією:

- освітні програми кафедри;
- наукова робота кафедри;
- лабораторії та матеріальна база кафедри.

В «Структура та робота кафедри», користувач може ознайомитися із наступною інформацією:

- структура кафедри;
- керівництво кафедри;
- викладацький склад кафедри;
- співпраця та партнери кафедри.

В «Практичні сценарії», користувач може виконати наступні дії:

- подати заявку на вступ;
- звернутися із запитанням;
- здійснити пошук викладача або загальних контактів кафедри.

Далі, розглянемо реалізацію онлайн-чату між абітурієнтом та представником університету в режимі реального часу.

Щоб потрапити у режим спілкування з представником університету, користувач повинен натиснути на кнопку «Діалог із представником» у головному навігаційному повідомленні (рис. 3.5). Після чого, абітурієнта повідомлять про те, що його запит було прийнято та відправлено представникам університету (рисунок 3.16).



Рис 3.16 Початок діалогу із представником університету

В той же час, адміністратор отримує запит та бачить у переліку запитів прізвище та ім'я абітурієнта (рисунок 3.17). Щоб розпочати чат із абітурієнтом, представник університету повинен натиснути на кнопку «Відкрити чат».

Адмінська консоль

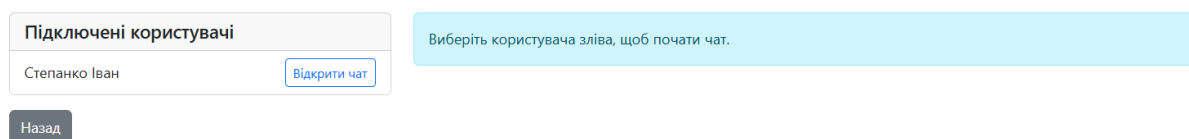


Рис. 3.17 Меню активних запитів абітурієнтів

Після того, як представник університету перейшов до чату користувача, він потрапляє в панель адміністратора (рисунок 3.18).

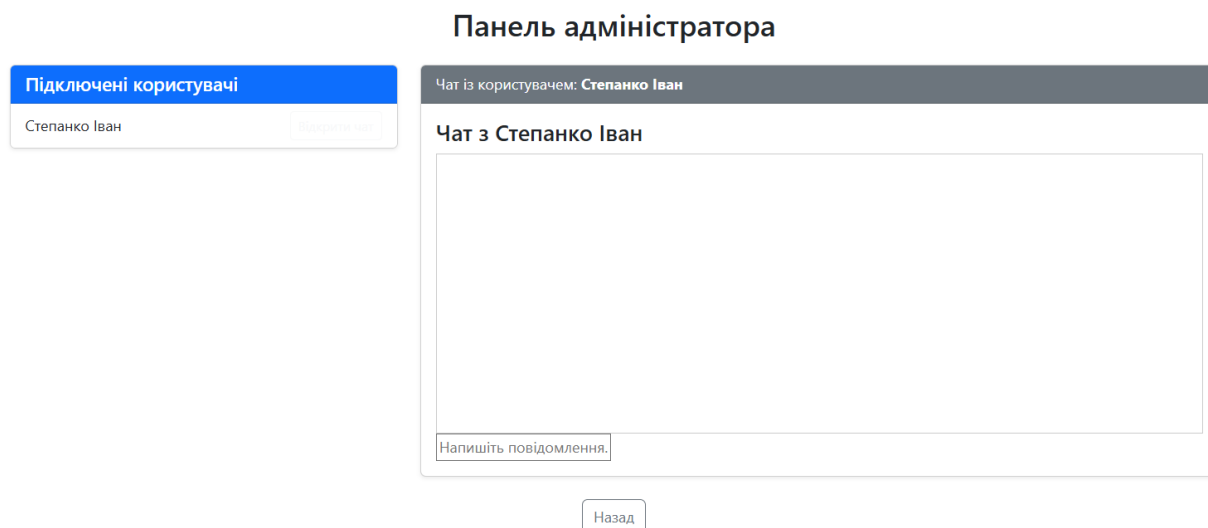


Рис. 3.18 Панель адміністратора

Функціонал чату було реалізовано з використанням фреймворку SignalR, який створює окремий сеанс роботи між представником університету та абітурієнтом.

Сам інтерфейс було розроблено за допомогою фреймворку React, який дозволяє динамічно оновлювати сторінку, що дозволяє бачити нові запити на чат та нові повідомлення у чаті без перезавантаження сторінки.

4. ТЕСТУВАННЯ ЧАТ-БОТУ

4.1 Ручне тестування

Ручне тестування – це тип тестування, в якому тестувальники проводять тестування програмного забезпечення без використання автоматизованих інструментів для тестування. Метою такого типу тестування є визначення дефектів та проблем, які неможливо виявити автоматизованим тестуванням. Також, таке тестування використовується у випадках, коли процес тестування неможливо автоматизувати.

Для тестування чат-боту було обрано саме такий тип тестування у зв'язку з наступними причинами:

- тестування повідомлень чат-боту зручніше та більш ефективно тестувати вручну;
- тестування повідомлень між адміністратором та користувачем зручніше та більш ефективно тестувати вручну;
- автоматизація відправки повідомлень чат-боту та їх обробки може бути некоректно протестованою автоматизованим тестуванням.

[20]

Для тестування системи розроблено таблицю із тестовими сценаріями (таблиця 4.1).

Таблиця 4.1

№	Назва	Кроки	Очікуваний результат
1	Авторизація	1. Перейти на сайт роботи із чат-ботом; 2. Заповнити усі дані для авторизації; 3. Натиснути кнопку «Вхід».	Користувача переадресує на сторінку роботи із чат-ботом
2	Реєстрація	1. Перейти на сайт роботи із чат-ботом; 2. Перейти на сторінку реєстрації; 3. Заповнити дані для реєстрації; 4. Натиснути кнопку «Зареєструвати».	Користувача переадресує на сторінку роботи із чат-ботом
3	Перегляд структури кафедри інженерії програмного забезпечення	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути кнопку «Кафедри університету»; 3. Натиснути кнопку «Навчально-науковий інститут Інформаційних технологій»; 4. Натиснути кнопку «Кафедра інженерії програмного забезпечення»; 5. Натиснути на кнопку «Структура та робота кафедри».	Користувач бачить навігаційне меню, яке містить інформацію про структуру кафедри
4	Перегляд дат подачі документів для бюджету	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Вступ до університету»; 3. Ввести ключове слово «правила»; 4. Натиснути на кнопку «Строки прийому документів»; 5. Обрати «Бюджет».	Користувач бачить таблицю із етапами вступної компанії та датами початку етапу та кінця.

Продовження таблиці 4.1

№	Назва	Кроки	Очікуваний результат
5	Конвертація набраних пунктів у бал НМТ	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Вступ до університету»; 3. Ввести ключове слово «правила»; 4. Натиснути на кнопку «Таблиця переведення балів»; 5. Ввести предмет «Математика»; 6. Ввести від 1 до 45 балів.	Чат бот відправив конвертований бал у діапазоні від 100 до 200
6	Перегляд критеріїв мотиваційного листа	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Вступ до університету»; 3. Ввести ключове слово «правила»; 4. Натиснути на кнопку «Критерії мотиваційного листа».	Чат бот відправив повідомлення з вмістом критеріїв оцінювання мотиваційного листа, розбиті по частинам мотиваційного листа
7	Перегляд вартості навчання	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Вступ до університету»; 3. Ввести ключове слово «вартість»;	Чат бот виводить інформацію про вартість контракту на навчання за рік.
8	Запит чату до адміністратора	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Діалог із представником»; 3. Переглянути запити на чат в панелі адміністратора	У панелі адміністратора з'явився новий запит на чат із адміністратором.

Продовження таблиці 4.1

№	Назва	Кроки	Очікуваний результат
9	Відправка повідомлень адміністратору	1. Перейти на сторінку роботи із чат-ботом; 2. Натиснути на кнопку «Діалог із представником»; 3. Переглянути запити на чат в панелі адміністратора; 4. Натиснути кнопку «Відкрити чат»; 5. Надіслати повідомлення абітурієнту	У абітурієнта в чаті відобразиться відправлене повідомлення

4.2 Аналіз результатів тестування

Після тестування було сформовано відповідну таблицю (таблиця 4.2), в яку було занесено результат тестування кожного тестового сценарію, із таблиці 4.1.

Таблиця 4.2

Результати тестування

№	Назва	Результат	Коментар
1	Авторизація	Пройдено	
2	Реєстрація	Пройдено	
3	Перегляд структури кафедри інженерії програмного забезпечення	Пройдено	Виправити назву кнопки у головному навігаційному меню на більш відповідну

Продовження таблиці 4.2

Результати тестування

№	Назва	Результат	Коментар
4	Перегляд дат подачі документів для бюджету	Пройдено	
5	Конвертація набраних пунктів у бал НМТ	Пройдено	
6	Перегляд критеріїв мотиваційного листа	Пройдено	
7	Перегляд вартості навчання	Пройдено	Додати розподіл спеціальностей по кафедрам
8	Запит чату до адміністратора	Пройдено	Додати відправку повідомлення, коли адміністратор увійшов до чату. Бажано, автоматичне повідомлення, яке представляє адміністратора
9	Відправка повідомлень адміністратору	Пройдено	

ВИСНОВОК

В ході виконання кваліфікаційної бакалаврської роботи було проведено аналіз предметної області, аналіз існуючих рішень, спроектовано чат-бот та онлайн-чат, а також проведено тестування розробленого чат-боту.

Під час аналізу предметної області, було проведено аналіз інформації, яка є важливою для абітурієнтів під час вибору університету.

Під час аналізу існуючих рішень, було розглянуто 3 офіційних чат-боту від університетів. Серед розглянутих аналогів був розглянуто 1 український чат-бот від НТУ «ХПІ» та 2 закордонних чат-боти від італійського університету та від іспанського університету. Було проведено аналіз, як університети надають доступ до чат ботів, та які основні функції виконують чат-боти для університетів.

Під час проєктування чат-бота було спроектовано архітектуру системи та розглянуто кожен шар; спроектовано логічну модель бази даних та діаграму JSON-колекцій бази даних; розглянуто кожен колекцію; побудовано діаграму варіантів використання для двох акторів у системі.

Під час створення опису реалізації чат-боту, було розглянуто та описано структуру проєкту чат-боту та проєкту веб-сайту для роботи із чат-ботом. Було побудовано діаграму класів задля глибшого розуміння структури проєкту та було продемонстровано сценарії повідомлень із чат-ботом та сценарій, як створити чат між абітурієнтом і представником університету.

Під час тестування чат-бота було проведено аналіз та виділено основний тип тестування чат-бота (ручне тестування), а також створено тестові сценарії, які повинні покрити усі основні аспекти роботи чат-бота та чату між представником університету та абітурієнтом. Після побудови тестових сценаріїв, вони були реалізовані. Результати тестування були внесені у спеціальну таблицю. Під час тестування не було виявлено дефектів вище

середнього рівня, але було виявлено 2 дефекти низького рівня, які пов'язані із текстами повідомлень у чат-боті. Усі дефекти було успішно усунено.

Фінальним результатом дипломної роботи є оформлена документація та розроблені 2 проекти для роботи із чат-ботом, а саме:

- веб-сайт для авторизації та підключення до основних функцій чат-бота;
- серверна частина, на якій реалізований чат-бот та функціонал онлайн чату між абітурієнтом та представником університету;

Розроблений проєкт готовий до впровадження на сайт Державного університету Інформаційно-комунікаційних технологій.

Доволі гнучка архітектура чат-боту дозволяє швидко вносити нові сценарії повідомлення та редагувати існуючі. В майбутньому це допоможе без проблем здійснювати підтримку роботи чат-бота.

Робота успішно пройшла апробацію. За її результати було опубліковано наступні тези доповідей:

1. Стаценко В.К. Сучасні технології розробки чат-ботів. Матеріали Всеукраїнської науково-технічної конференції «Сучасний стан та перспективи розвитку IoT». Збірник тез. 24 квітня 2025 року, ДУІКТ, м.Київ (подано до друку);
2. Стаценко В.К., Худік Б.О. Підвищення довіри до кваліфікованості кафедри інформаційних технологій за допомогою чат боту. Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24 квітня 2025 року, ДУІКТ, м.Київ, с. 516-517.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке кафедра в університеті?. *Fast News. Довідник слів*. 02.09.2024. URL: <https://fastnews.com.ua/shho-take-kafedra-v-universyteti.html> (дата звернення: 25.04.2025).
2. Сторінка чат-боту НТУ «ХПІ». URL: <https://khpi-career-chatbot.vercel.app> (дата звернення: 26.04.2025).
3. Сторінка чат-боту університету Мурсії (Іспанія). URL: <https://www.um.es/web/estudios/admision/preinscripcion-grado> (дата звернення: 26.04.2025).
4. Сторінка чат-боту університету Веронія (Італія). URL: <https://www.univr.it/it/> (дата звернення: 26.04.2025).
5. НТУ «ХПІ». Фахівці Харківського Політеху створили інноваційний чат-бот для профорієнтації абітурієнтів. URL: <https://www.kpi.kharkov.ua/ukr/2025/04/29/fahivtsi-harkivskogo-politehu-stvoryly-innovatsijnyj-chatbot-dlya-proforiyentatsiyi-abituriyentiv/> (дата звернення: 26.04.2025).
6. GOOD UNIVERSITIES. The University of Murcia trusts 1millionbot to help its new students. URL: <https://1millionbot.com/en/chatbot-1millionbot-universidad-de-murcia-resolver-dudas-estudiantes/> (дата звернення: 26.04.2025).
7. Документація університету Веронія. Віртуальний помічник Верони. URL: <https://www.univr.it/it/i-nostri-servizi/futuri-studenti/segreterie-studenti/univr-risponde-l-assistente-virtuale-dell-universita-di-verona> (дата звернення: 26.04.2025).
8. Визначення вимог: що це таке і як це застосовувати? | Повне керівництво. URL: https://visuresolutions.com/uk/блог/визначення-вимог/#elementor-toc_heading-anchor-2 (дата звернення: 26.04.2025).
9. What is ASP.Net Core? Microsoft Learn. Learn Asp.Net Core. URL: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core> (дата звернення: 01.05.2025).
10. Bot Framework SDK documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/bot-service/index-bf-sdk?view=azure-bot->

[service-4.0](#) (дата звернення: 02.05.2025).

11. Overview of ASP.NET Core SignalR. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-9.0> (дата звернення: 03.05.2025).

12. Ratnayake B. C# Tutorial — SignalR. *Mediem*. @BasuraRatnayake. 22.01.2024. URL: <https://medium.com/@BasuraRatnayake/c-tutorial-signalr-b7a7b76901be> (дата звернення: 03.05.2025).

13. Посібник: знайомство з React. Офіційна документація. URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html#what-is-react> (дата звернення: 03.05.2025).

14. Chat SDK overview. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/communication-services/concepts/chat/sdk-features> (дата звернення: 04.05.2025).

15. Стівен Т. Що таке MongoDB? вступ, Архітектура, функції та приклад. *GURU99*. 26.09.2024. URL: <https://www.guru99.com/uk/what-is-mongodb.html> (дата звернення: 04.05.2025).

16. Що таке Microsoft Azure? Simpla. URL: <https://simpla.com.ua/blog/shcho-take-microsoft-azure> (дата звернення: 04.05.2025).

17. Мартін Р. Clean Code Blog. The Clean Architecture. URL: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (дата звернення: 05.05.2025).

18. IBM Documentation. Use-case diagrams. URL: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case> (дата звернення: 06.05.2025).

19. IBM Documentation. Class diagrams. URL: <https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams> (дата звернення: 06.05.2025).

20. Гамільтон Т. Що таке ручне тестування? URL: <https://www.guru99.com/uk/manual-testing.html> (дата звернення: 18.05.2025).

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмних засобів підтримки онлайн-чату на сайті університету

Виконав студент 4 курсу
групи ПД-44

Стаценко Володимир Костянтинович
Керівник роботи

доктор філософії, доцент кафедри ІПЗ Худік Богдан Олександрович

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення процедури ознайомлення абітурієнтів університету та особливостями навчання в ньому.

Об'єкт дослідження: процес надання інформації абітурієнтам та їх батькам про університет.

Предмет дослідження: чат-бот для спілкування з абітурієнтом або його батьками про університет та процедуру вступу.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз процесу відбору та ознайомлення абітурієнта з університетом, визначивши основні та найважливіші критерії підбору.
2. Здійснити огляд та аналіз існуючих інструментів, які університети використовують для презентації себе перед абітурієнтом.
3. Провести огляд та визначити технології та інструменти, необхідні для створення чат-боту.
4. Сформулювати вимоги до розробки чат-боту.
5. Спроектувати архітектуру застосунку для роботи із чат-ботом.
6. Розробити функціонал чат-боту та онлайн чату через нього.
7. Провести тестування розробленого чат-боту.

3

АНАЛІЗ АНАЛОГІВ

Характеристика	НТУ «ХПІ»	Lola	InfoChat Studenti	<u>ChatBotUni</u>
Пошук за ключовими словами	-	-	-	+
Інтерактивний онлайн-чат	+	+	+	+
Збереження історії діалогу	+	+	-	+
Українська локалізація	+	-	-	+
Меню навігації	-	-	+	+
Інформація про кафедру	+	-	-	+
Інформація про вступ	+	+	+	+
Онлайн чат із представником університету	-	-	-	+

4

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги:

1. Чат-бот повинен реагувати на повідомлення;
2. Чат-бот повинен зберігати історію чату;
3. Функція пошуку у чат-боті;
4. Чат між оператором та користувачем;
5. Валідація повідомлень.

Нефункціональні вимоги:

1. Чат-бот повинен бути доступним для користувача протягом більше як 99% часу;
2. Чат-бот повинен містити захист від DDOS атак та від несанкціонованого доступу;
3. Чат-бот у разі виникнення помилок, виводить відповідне повідомлення;
4. Швидкість відповіді чат-боту не повинна перевищувати 5 секунд.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Visual Studio 2022



React



Azure

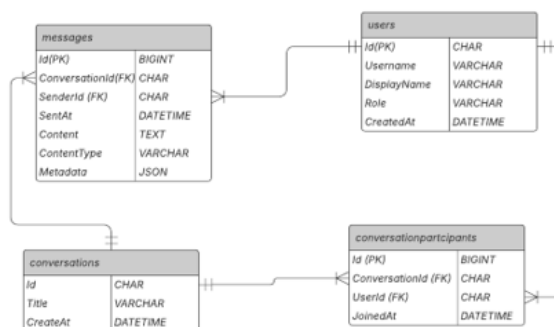
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



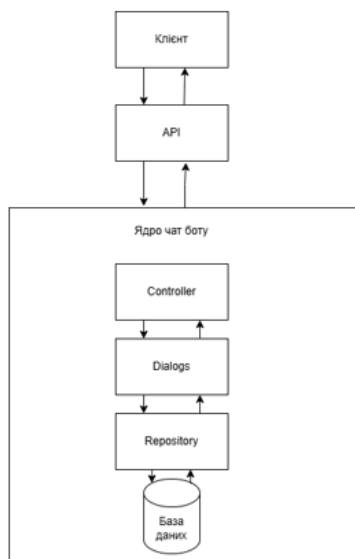
7

СТРУКТУРА БАЗИ ДАНИХ



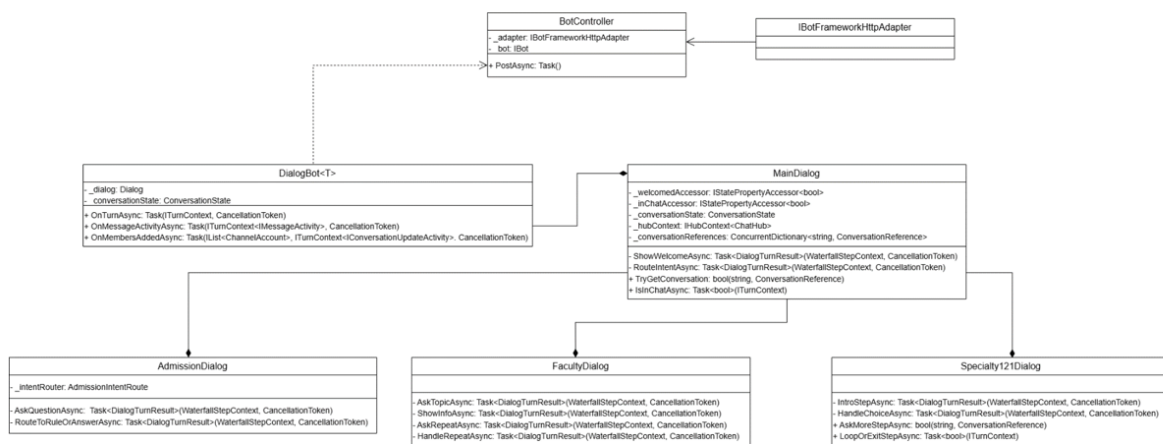
8

АРХІТЕКТУРА ЧАТ-БОТУ



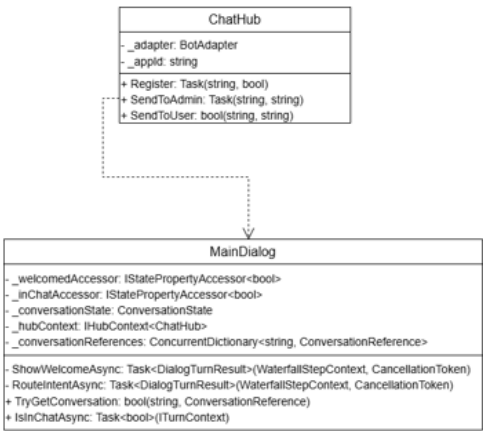
9

ДІАГРАМА КЛАСІВ ОСНОВНОЇ ЛОГІКИ ЧАТ-БОТА



10

ДІАГРАМА КЛАСІВ ЧАТИНГУ



РОЗРОБЛЕНІ ДІАЛОГИ

БТ

Правила вступу 2025

Ось перелічень правил, які я знаю:

1. Перелік спеціальностей та програм

2. Конкурсні предмети та коефіцієнти

3. Строки прийому документів

4. Спеціальності з особливою підтримкою

5. Таблиці переведення балів

6. Критерії мотиваційного листа

7. Правила для окупованих територій

Повернутися в головне меню

Оберіть одне правило

Щойно

Діалог із правилами вступу

БТ

Етапи вступної кампанії 2025 (Бюджет)

№ п/п	Етап	Термін
1	Реєстрація електронних кабінетів вступників, завантаження необхідних документів	з 01 липня
2	Реєстрація заяв на участь у співбесідах — на бюджет	з 03 до 18 ⁰⁰ 10 липня
3	Строки проведення співбесід — на бюджет	з 08 до 19 липня
4	Прийом заяв та документів у електронній або паперовій формі	з 19 липня до 18 ⁰⁰ 01 серпня
5	Оприлюднення списку рекомендованих до зарахування за держзамовленням	до 06 серпня
6	Виконання вимог до зарахування (договір про навчання)	до 18 ⁰⁰ 09 серпня
7	Зарахування вступників за держзамовленням	до 11 серпня
8	Надання рекомендацій за кошти фізичних/юридичних осіб	11 серпня
9	Виконання вимог вступниками за контрактом	до 18 ⁰⁰ 14 серпня
10	Зарахування вступників за кошти фізичних/юридичних осіб	до 16 серпня
11	Переведення на вакантні місця держзамовлення	18 серпня

Перелік дат для етапів вступної компанії

12

РОЗРОБЛЕНІ ДІАЛОГИ

БТ

Привіт! 🤖

Оберіть тему або введіть ключове слово (наприклад «вступ», «кафедра», «121»):

Вступ до університету

Кафедри університету

Спеціальність 121

Діалог із представником

Натисніть кнопку вище.

Щойно

Привітальне повідомлення

БТ

Оберіть, про що ви хочете дізнатися:

Про спеціальність 121

Чому обрати ІПЗ

Що чекає студента

Особливості навчання

Стажування та кар'єра

Про кафедру

Контакти завідувача

Повернутися в головне меню

Щойно

Навігаційне повідомлення про спеціальність 121

13

РОЗРОБЛЕНІ ДІАЛОГИ

БТ

Оберіть, будь ласка, навчально-науковий інститут:

Навчально-науковий інститут Інформаційних техн...

Навчально-науковий інститут Кібербезпеки та зах...

Навчально-науковий інститут Телекомунікацій

Навчально-науковий інститут Менеджменту та під...

Натисніть одну з кнопок вище.

Щойно

Навігаційне повідомлення із
навчально-науковими
інститутами

БТ

Ви обрали *Навчально-науковий інститут Інформаційних технологій*.
Тепер виберіть кафедру:

Кафедра Інженерії програмного забезпечення

Кафедра Інформаційних систем та технологій

Кафедра Комп'ютерних наук

Натисніть кнопку з назвою кафедри.

Щойно

Перелік кафедр за обраним
навчально-науковим
інститутом

14

ЕКРАННІ ФОРМИ

Адмінська консоль

Підключені користувачі

Іван Піддубний

Відкрити чат

Назад

Виберіть користувача зліва, щоб почати чат.

Перегляд запитів на чат

Адмінська консоль

Підключені користувачі

Іван Піддубний

Відкрити чат

Назад

Чат з Іван Піддубний

Іван Піддубний: Доброго дня

Напишіть повідомлення

Чат між абітурієнтом та
представником університету

12

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей:

1. Стаценко В.К. Сучасні технології розробки чат-ботів. Матеріали Всеукраїнської науково-технічної конференції «Сучасний стан та перспективи розвитку IoT». Збірник тез. 24 квітня 2025 року, ДУІКТ, м.Київ (подано до друку);
2. Стаценко В.К., Худік Б.О. Підвищення довіри до кваліфікованості кафедри інформаційних технологій за допомогою чат боту. Матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». Збірник тез. 24 квітня 2025 року, ДУІКТ, м.Київ, с. 516-517.

16

ВИСНОВКИ

1. Проведено аналіз процесу відбору та ознайомлення абітурієнта з університетом, це дало розуміння основних та найважливіших критерії підбору університету абітурієнтом.
2. Здійснено огляд та виконано аналіз існуючих аналогів, це дало розуміння сильних та слабких сторін чат-ботів, та вирішення яких проблем може виконати чат-бот.
3. Проведено огляд технологій та інструментів для створення чат-боту, це дозволило отримати розуміння, які технології будуть використовуватися у розробці чат-боту.
4. Сформовано вимоги до розробки чат-боту, це дало розуміння, який функціонал повинен бути реалізований у чат-боті, та які проблеми вирішує чат-бот.
5. Спроектовано архітектуру застосунку для роботи із чат-ботом, це дало розуміння каркасу майбутнього розроблюваного чат-бота.
6. Розроблено фронтенд та серверну частини роботи із чат-ботом, що дає можливість розпочати тестування чат-боту.
7. Проведено тестування розробленого чат-боту, тестування показало, що чат-бот не має дефектів вище середнього, а усі дефекти рівня нижче середнього були одразу виправлені.

ДОДАТОК Б. ЛІСТИНГ ОСНОВНИХ ПРОГРАМНИХ МОДУЛІВ

```

using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Bot.Builder;
using
Microsoft.Bot.Builder.Integration.AspNet.Core;

namespace ChatBotUni.Controllers
{
    [Route("api/messages")]
    [ApiController]
    public class BotController : ControllerBase
    {
        private readonly
        IBotFrameworkHttpAdapter _adapter;
        private readonly IBot _bot;

        public
        BotController(IBotFrameworkHttpAdapter
        adapter, IBot bot)
        {
            _adapter = adapter;
            _bot = bot;
        }

        [HttpPost]
        [HttpGet]
        public async Task PostAsync()
        {
            await _adapter.ProcessAsync(Request,
            Response, _bot);
        }
    }
}

using Microsoft.Bot.Builder.Dialogs;
using Microsoft.Bot.Builder;
using Microsoft.Bot.Schema;
using System.Collections.Generic;
using System.Threading.Tasks;
using System.Threading;

public class DialogBot<T> : ActivityHandler
where T : Dialog
{
    private readonly Dialog _dialog;
    private readonly ConversationState
    _conversationState;

    public DialogBot(ConversationState
    conversationState, T dialog)
    {
        _conversationState = conversationState;
        _dialog = dialog;
    }

    public override async Task
    OnTurnAsync(ITurnContext turnContext,
    CancellationToken cancellationToken = default)
    {
        await base.OnTurnAsync(turnContext,
        cancellationToken);

        await
        _conversationState.SaveChangesAsync(turnCon
        text, false, cancellationToken);
    }

    protected override async Task
    OnMessageActivityAsync(ITurnContext<IMess
    ageActivity> turnContext, CancellationToken
    cancellationToken)
    {
        await _dialog.RunAsync(
        turnContext,

        _conversationState.CreateProperty<DialogState>
        >("DialogState"),
        cancellationToken);
    }

    protected override async Task
    OnMembersAddedAsync(
        IList<ChannelAccount> membersAdded,

        ITurnContext<IConversationUpdateActivity>
        turnContext,
        CancellationToken cancellationToken)
    {
        foreach (var member in membersAdded)
        {
            if (member.Id !=
            turnContext.Activity.Recipient.Id)
            {
                await _dialog.RunAsync(
                turnContext,

                _conversationState.CreateProperty<DialogState>
                >("DialogState"),
                cancellationToken);
            }
        }
    }
}

```

```

using System;
using System.Collections.Concurrent;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using System.Threading.Tasks;
using ChatBotUni.Dialogs.Faculty;
using ChatBotUni.Hubs;
using Microsoft.AspNetCore.SignalR;
using Microsoft.Bot.Builder;
using Microsoft.Bot.Builder.Dialogs;
using Microsoft.Bot.Schema;

namespace ChatBotUni.Dialogs
{
    public class MainDialog : ComponentDialog
    {
        private const string HasWelcomedKey =
            "HasWelcomed";
        private const string InChatKey = "InChat";

        private readonly
            IStatePropertyAccessor<bool>
            _welcomedAccessor;
        private readonly
            IStatePropertyAccessor<bool>
            _inChatAccessor;
        private readonly ConversationState
            _conversationState;
        private readonly IHubContext<ChatHub>
            _hubContext;

        private static readonly
            ConcurrentDictionary<string,
            ConversationReference>
            _conversationReferences
            = new ConcurrentDictionary<string,
            ConversationReference>();

        public MainDialog(
            IHubContext<ChatHub> hubContext,
            ConversationState conversationState,
            AdmissionDialog admissionDialog,
            AdmissionRulesDialog
            admissionRulesDialog,
            FacultyDialog facultyDialog,
            StructureFacultyDialog
            structureFacultyDialog,
            MainActivityDialog mainActivityDialog,
            Specialty121Dialog specialty121Dialog
        )
            : base(nameof(MainDialog))
        {
            _conversationState = conversationState;

```

```

        _welcomedAccessor = conversationState
            .CreateProperty<bool>(HasWelcomedKey);
        _hubContext = hubContext;
        _welcomedAccessor =
            _conversationState
            .CreateProperty<bool>(HasWelcomedKey);
        _inChatAccessor = _conversationState
            .CreateProperty<bool>(InChatKey);

        AddDialog(admissionDialog);
        AddDialog(admissionRulesDialog);
        AddDialog(facultyDialog);
        AddDialog(structureFacultyDialog);
        AddDialog(mainActivityDialog);
        AddDialog(specialty121Dialog);

        AddDialog(new
            TextPrompt(nameof(TextPrompt)));

        AddDialog(new
            WaterfallDialog(nameof(WaterfallDialog), new
            WaterfallStep[]
            {
                ShowWelcomeAsync,
                RouteIntentAsync
            }));

        InitialDialogId =
            nameof(WaterfallDialog);
    }

    private async Task<DialogTurnResult>
        ShowWelcomeAsync(
            WaterfallStepContext step,
            CancellationToken ct)
    {
        var hasWelcomed = await
            _welcomedAccessor.GetAsync(
                step.Context, () => false, ct);

        if (hasWelcomed)
        {
            return await step.NextAsync(null, ct);
        }

        var welcomeCard = new HeroCard
        {
            Title = "Привіт 🐼",
            Text = "Оберіть тему або введіть
            ключове слово (наприклад «вступ»,
            «кафедра», «121»):",
            Buttons = new List<CardAction>
            {

```

```

        new
        CardAction(ActionTypes.ImBack, "🏠 Вступ до
університету", value: "вступ"),
        new
        CardAction(ActionTypes.ImBack, "🏢 Кафедри
університету", value: "кафедра"),
        new
        CardAction(ActionTypes.ImBack, "📖
Спеціальність 121", value: "спеціальність
121"),
        new
        CardAction(ActionTypes.ImBack, "👤 Діалог із
представником", value: "чат з
адміністратором")
    };

    await step.Context.SendActivityAsync(
MessageFactory.Attachment(welcomeCard.ToA
ttachment()),
    ct);

    await _welcomedAccessor.SetAsync(
step.Context, true, ct);

    return await step.PromptAsync(
        nameof(TextPrompt),
        new PromptOptions
        {
            Prompt =
MessageFactory.Text("Натисніть кнопку
вище.")
        },
        ct);
}

private async Task<DialogTurnResult>
RouteIntentAsync(
    WaterfallStepContext stepContext,
    CancellationToken ct)
{
    var text =
(stepContext.Result?.ToString() ?? "").Trim();
    var lower = text.ToLowerInvariant();

    var inChat = await
_inChatAccessor.GetAsync(
    stepContext.Context, () => false, ct);

    if (inChat)
    {
        var chatMessage =
(stepContext.Context.Activity.Text ??
string.Empty).Trim();

```

```

        if (chatMessage.ToLower() ==
"вийти" || chatMessage.ToLower() ==
"закінчити чат")
        {
            await
_inChatAccessor.SetAsync(stepContext.Context
, false, ct);

            await
_conversationState.SaveChangesAsync(stepCon
text.Context, false, ct);

            await
stepContext.Context.SendActivityAsync(
                MessageFactory.Text("Ви
вийшли з чату з адміністратором. Чим ще
можу допомогти?"),
                ct);

            return await
stepContext.ReplaceDialogAsync(nameof(Water
fallDialog), null, ct);
        }

        var userId =
stepContext.Context.Activity.From.Id;
        await
_inHubContext.Clients.Group("admin")
            .SendAsync("ReceiveFromUser",
userId, chatMessage, DateTime.UtcNow);

        await
stepContext.Context.SendActivityAsync(
            MessageFactory.Text("Ваше
повідомлення надіслано адміністратору."),
            ct);

        return await
stepContext.NextAsync(null, ct);
    }

    switch (lower)
    {
        case "вступ":
            return await
stepContext.BeginDialogAsync(nameof(Admiss
ionDialog), null, ct);

        case "кафедра":
            return await
stepContext.BeginDialogAsync(nameof(Faculty
Dialog), null, ct);

        case "спеціальність 121":
            return await
stepContext.BeginDialogAsync(nameof(Specialt
y121Dialog), null, ct);

```

```

        case "чат з адміністратором":
            var convoRef =
                stepContext.Context.Activity.GetConversationReference();
            var userId =
                stepContext.Context.Activity.From.Id;
            _conversationReferences[userId] =
                convoRef;

            await
                _inChatAccessor.SetAsync(stepContext.Context,
                    true, ct);
            await
                _conversationState.SaveChangesAsync(stepContext.Context,
                    false, ct);

            await
                _hubContext.Clients.Group("admin")
                    .SendAsync("UserJoined",
                        userId);

            await
                stepContext.Context.SendActivityAsync(
                    MessageFactory.Text("З'єднання з
                    представником... Почекайте, будь-ласка."),
                        ct);

            return await
                stepContext.ReplaceDialogAsync(
                    nameof(WaterfallDialog), null,
                        ct);

            default:
                var keywordMap = new
                    Dictionary<string, string[]>
                {

                    [nameof(AdmissionRulesDialog)] = new[] {
                        "вступ", "прийом", "документ", "абітур" },
                    [nameof(FacultyDialog)] = new[]
                        { "кафедр", "факультет", "інженер" },

                    [nameof(StructureFacultyDialog)] = new[] {
                        "структур", "організац", "керівництво" },
                    [nameof(MainActivityDialog)] =
                        new[] { "основн", "програм", "лаборатор",
                            "науков" },
                    [nameof(Specialty121Dialog)] =
                        new[] { "121", "спеціальн", "software", "пз" }
                };

                var matches = keywordMap
                    .Where(kv => kv.Value.Any(kw
                        => lower.Contains(kw)))

```

```

                    .Select(kv => kv.Key)
                    .Distinct()
                    .ToList();

                if (matches.Count == 1)
                {
                    return await
                        stepContext.BeginDialogAsync(matches[0],
                            null, ct);
                }
                else if (matches.Count == 0)
                {
                    await
                        stepContext.Context.SendActivityAsync(
                            MessageFactory.Text("Вибачте, я не зрозумів.
                            Спробуйте інше слово або меню."),
                                ct);
                }
                else
                {
                    await
                        stepContext.Context.SendActivityAsync(
                            MessageFactory.Text(
                                $"Знайдено кілька
                                варіантів за словом «{text}». Будь ласка,
                                уточніть або оберіть меню."
                            ),
                                ct);
                }

                await
                    _welcomedAccessor.SetAsync(stepContext.Context,
                        false, ct);
                return await
                    stepContext.ReplaceDialogAsync(
                        nameof(WaterfallDialog), null,
                            ct);
            }
        }

        public static bool
            TryGetConversation(string userId, out
                ConversationReference reference) =>

            _conversationReferences.TryGetValue(userId,
                out reference);

        public static async Task<bool>
            IsInChatAsync(ITurnContext turnContext)
            {
                var access =
                    turnContext.TurnState.Get<IStatePropertyAccessor<bool>>(InChatKey);

```

```

        return await
        access.GetAsync(turnContext, () => false,
        default);
    }
}

```

```

// ChatHub.cs
using System;
using System.Threading;
using System.Threading.Tasks;
using ChatBotUni.Dialogs;
using Microsoft.AspNetCore.SignalR;
using Microsoft.Bot.Builder;
using Microsoft.Bot.Schema;
using Microsoft.Extensions.Configuration;

```

```

namespace ChatBotUni.Hubs
{
    public class ChatHub : Hub
    {
        private readonly BotAdapter _adapter;
        private readonly string _appId;

        public ChatHub(
            BotAdapter adapter,
            IConfiguration configuration)
        {
            _adapter = adapter;
            // Читаем MicrosoftAppId из
            appsettings.json
            _appId =
            configuration["MicrosoftAppId"] ??
            string.Empty;
        }
    }
}

```

```

        public async Task Register(string userId,
        bool isAdmin)
        {
            if (isAdmin)
            {
                await
                Groups.AddToGroupAsync(Context.Connection
                Id, "admin");
            }
            await
            Groups.AddToGroupAsync(Context.Connection
            Id, userId);
        }
    }
}

```

```

        public async Task SendToAdmin(string
        userId, string message)
        {
            await Clients.Group("admin")

```

```

        .SendAsync("ReceiveFromUser", userId,
        message, DateTime.UtcNow);
        }
    }
}

```

```

        public async Task SendToUser(string
        userId, string message)
        {
            await Clients.Group(userId)

            .SendAsync("ReceiveFromAdmin", userId,
            message, DateTime.UtcNow);

            if
            (MainDialog.TryGetConversation(userId, out
            var convRef))
            {
                await
                _adapter.ContinueConversationAsync(
                _appId,
                convRef,
                async (turnContext, ct) =>
                {
                    var activity =
                    MessageFactory.Text($"{message}");
                    await
                    turnContext.SendActivityAsync(activity, ct);
                },
                CancellationToken.None);
            }
        }
    }
}

```