**Міністерство освіти і науки України**
**Луцький національний технічний університет**
**Факультет комп'ютерних та інформаційних технологій**
**Кафедра інженерії програмного забезпечення**

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

# РОЗРОБКА СЕРВІСУ ДЛЯ ЗВЕРНЕННЯ СТУДЕНТІВ З ВИКОРИСТАННЯМ МОВИ PYTHON

# DEVELOPMENT OF A SERVICE FOR APPLYING STUDENTS USING THE PYTHON LANGUAGE

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-42
Потапюк Валентин Валерійович
_____
(підпис)

Керівник:
Ліщина Наталія Миколаївна
_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«8» 06 2024 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна
_____

Луцьк – 2024 року

**ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *інженерії програмного забезпечення*
Ступінь вищої освіти *бакалавр*
Галузь знань: *12 «Інформаційні технології»*
Спеціальність: *121 «Інженерія програмного забезпечення»*
Освітня програма: «*Інженерія програмного забезпечення*»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Н. Ліщина
«2» 02 2024 р.

**ЗАВДАННЯ**
**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ**

Потапюку Валентину Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка сервісу для звернення студентів з використанням мови Python»
Керівник роботи: к.т.н., доцент Ліщина Н. М.
затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *«5» червня 2024 р.*

3. Вихідні дані до роботи:

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
*Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.*

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| *Аналіз предметної області* | *Ліщина Н. М.* | | |
| *Специфікація вимог до розробленої системи* | *Ліщина Н. М.* | | |
| *Розробка об'єкта проектування* | *Ліщина Н. М.* | | |
| *Нормоконтроль* | *Повстяна Ю. С.* | | |
| *Гарант ОП* | *Ліщина Н. М.* | | |
| *Показник запозичень тексту* | 4,8% | | |
| *Академічна доброчесність* | *Ящук А. А.* | | |

7. Дата видачі завдання *«2» лютого 2024 р.*

| № | Назва етапів кваліфікаційної роботи бакалавра | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | *Огляд літературних джерел по темі кваліфікаційної роботи бакалавра* | *до 16.02.2024 р.* | вик. |
| 2 | *Аналіз проблеми розробки та впровадження об'єкту проектування* | *до 01.03.2024 р.* | вик. |
| 3 | *Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання* | *до 15.03.2024 р.* | вик. |
| 4 | *Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних* | *до 29.03.2024 р.* | вик. |
| 5 | *Практична реалізація об'єкта проектування та розробка бази даних* | *до 26.04.2024 р.* | вик. |
| 6 | *Тестування та налагодження об'єкта проектування* | *до 03.05.2024 р.* | вик. |
| 7 | *Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру* | *до 5.06.2024 р.* | вик. |

Здобувач вищої освіти _____ Потапчук В.В
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ К. Ліщина
(підпис) (прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

**Рецензія**

**на кваліфікаційну роботу**

**Здобувач вищої освіти:** Потапюк Валентин Валерійович

**Тема: «***Розробка сервісу для звернення студентів з використанням мови Python»*

**Коротка характеристика кваліфікаційної роботи.** *Кваліфікаційна робота представляє собою добре структуровану, детально обґрунтовану та професійно виконану роботу, яка орієнтована на розробку сервісу для звернення студентів з використанням мови Python.*

**Самостійні розробки і пропозиції автора.** *Результатом кваліфікаційної роботи став сервіс, який підключає чат-бот у Telegram для збору звернень від студентів та веб-сервіс для опрацювання звернень членами студентської ради. В ході роботи були проаналізовані існуючі рішення, можливі підходи до побудови архітектури сервісу та інструменти розробки.*

**Недоліки.** *Істотних недоліків в роботі не виявлено.*

**Загальний висновок.** *Робота виконана на високому фаховому рівні, має логічну та послідовну структуру, відповідає встановленим вимогам, може бути представлена до захисту екзаменаційній комісії та заслуговує позитивної оцінки.*

Рецензент: Кабак Віталій Васильович, зав. каф. УОТ
(прізвище, ім'я, по-батькові, посада)

**Рецензент кваліфікаційної роботи** _____
(підпис)

«7» 06 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет

**Факультет комп'ютерних та інформаційних технологій**
**Кафедра інженерії програмного забезпечення**

# ВІДГУК
# КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач вищої освіти *Потапюк Валентин Валерійович*
група *ІПЗ-42*

**Тема кваліфікаційної роботи:** *Розробка сервісу для звернення студентів з використанням мови Python*

Керівник: Ліщина Наталія Миколаївна

**Актуальність теми.** Виконання запланованих в роботі завдань дозволило створити ефективний та зручний сервіс для звернень студентів до студради навчального закладу. Новий сервіс сприяє покращенню комунікації між студентами та адміністрацією, підвищенню ефективності обробки студентських запитів та загальному підвищенню якості обслуговування студентів. Це, в свою чергу, допоможе створити більш комфортні умови для навчання та взаємодії студентів з адміністрацією університету.

**Об'єктом дослідження** є взаємодія студентів з студрадою навчального закладу, зокрема, механізми подачі та обробки звернень студентів до різних підрозділів університету.

**Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи.** Результатом кваліфікаційної роботи став сервіс, що включає чат-бот у Telegram для збору звернень від студентів і веб-сервіс для їх опрацювання членами студентської ради. Під час роботи було проаналізовано існуючі рішення, можливі підходи до побудови архітектури сервісу та інструменти розробки.

**Зауваження та недоліки:** істотних недоліків не виявлено.

**Загальний висновок** робота виконана на високому рівні та заслуговує позитивної оцінки _за умови успішного захисту.

**Керівник** _____ (Н. Ліщина)
(підпис) (прізвище, ім'я, по батькові)

«7» 06 2024 р.

# АНОТАЦІЯ

Потапюк В. В. Розробка сервісу для звернення студентів з використанням мови Python. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків, списку використаних джерел. В роботі розглядаються існуючі рішення в різних підрозділах студентської ради і їх ефективність по відношенню до розробленого сервісу, аналізуються можливі архітектурні підходи та обрані рішення, а також обрані технології, розглядаються основні функції сервісу та інтерфейсу з поясненнями та ілюстраціями.

Ключові слова: студентська рада, студентські звернення, Python, програмне забезпечення для відстеження звернень.

# ABSTRACT

Potapyuk V. V. Development of a service for student appeals using the Python language. Manuscript.

Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualification work consists of an introduction, three sections, conclusions, and a list of used sources. The work examines existing solutions in various divisions of the student council and their effectiveness in relation to the developed service, analyzes possible architectural approaches and selected solutions, as well as selected technologies, examines the main functions of the service and interface with explanations and illustrations.

Keywords: student council, student referrals, Python, referral tracking software.

**ЗМІСТ**

# ВСТУП

Актуальність теми. Щодня величезна кількість студентів по всьому світу стикаються з проблемами різного характеру, які так чи інакше впливають на навчальний процес. В кожному сучасному закладі освіти є керівні органи, в компетенцію яких входить допомога студентам у вирішенні подібних завдань. Співробітники цих керівних організацій змушені обробляти велику кількість різноманітних запитів від студентів, які, до того ж, адресовані різним структурам.

Проблема організації збору запитів від студентів не оминула і ЛНТУ. У цей час збір запитів здійснюється за допомогою Google Forms, а також через неофіційні звернення студентів через особисті повідомлення до кураторів, повідомлення в студентських групах, в групах студентських рад факультетів або безпосередньо до студентів – учасників цих спільнот у соціальних мережах.

Аналіз такого підходу дозволив виділити наступні проблеми:

– зберігання та обробка запитів студентів децентралізована, що знижує ефективність роботи над ситуаціями, іншими словами, запити можуть бути втрачені або оброблені тривалий час;

– відсутнє єдине бачення ситуації при контролі виконання завдань окремими частинами студентської ради та прозорості в роботі між ними, тому ведення статистики та розробка системних рішень для кейсів викликає труднощі.

Для вирішення цієї проблеми було запропоновано розробити єдину систему збору та агрегування звернень учнів за загальною тематикою. Ця система дає можливість систематизувати процес роботи зі зверненнями, а також підвищує ефективність роботи комітетів та інших підрозділів в рамках взаємодії зі студентами. Система являє собою зв’язку чат-бота для подання звернень студентами з веб-сервісом для опрацювання цих самих запитів членами відділів студентської ради. Перш за все, і веб-сервіс, і чат-бот легко інтегруються в робочі

процеси і немає необхідності витрачати сторонні ресурси на установку, наприклад, мобільного додатку.

Запропонована система усуває практично всі перераховані вище проблеми, так як всі заявки збираються в одному місці, що полегшує і оптимізує роботу співробітників, задіяних в їх обробці.

Ще однією важливою перевагою системи є база запитів, яка дозволяє не тільки аналізувати статистику прийнятих рішень на основі запитів, а й, звичайно ж, стає цінним ресурсом для обробки запитів у майбутньому.

Важливим аспектом є те, що, згідно з дослідженнями, більшість студентів віддають перевагу знеособленій адресі, особливо анонімній, перед особистою адресою конкретної, але невідомої особи.

В результаті цієї розробки планується впровадження чат-боту у Telegram для збору запитів від студентів та веб-сервісу, який об’єднує всіх студентів і студентську раду, а також розробка функціоналу для реагування на запити студентів. Послуга буде впроваджена в реальну роботу студентської ради, що підвищить ефективність її функціонування. Крім того, планується масштабна піар-кампанія сервісу для більш тісної та продуктивної співпраці студентів зі студентською радою.

Мета роботи – розробити сервіс, який дозволить студентам зручно подавати звернення до студради закладу освіти, забезпечити зручний інтерфейс та швидку обробку заявок.

Об’єктом дослідження є взаємодія студентів з студрадою навчального закладу, зокрема, механізми подачі та обробки звернень студентів до різних підрозділів університету.

Предметом дослідження є процеси подання звернень студентів до студентської ради та їх подальша обробка та вирішення. Особлива увага приділяється використанню мови програмування Python та відповідних

фреймворків для створення цього сервісу, а також оцінці його ефективності у покращенні комунікації та обробки студентських запитів.

Завдання роботи:

– проаналізувати існуючі сервіси для звернень студентів;

– визначити вимоги до функціоналу сервісу;

– розробити архітектуру сервісу;

– реалізувати сервіс на мові Python з використанням відповідних фреймворків.

# РОЗДІЛ 1
# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз сучасного стану проблеми

Предметною областю даної роботи є супровід освітнього процесу. Студенти віддають перевагу безособовому спілкуванню, оскільки воно дозволяє їм отримати необхідну допомогу, не турбуючись про будь-яку потенційну упередженість чи судження з боку конкретної людини. Також незаперечним є той факт, що отримати допомогу анонімно набагато простіше, оскільки, в цьому випадку студенти почуваються комфортніше, ставлячи запитання і не соромлячись чи сорому за те, що почуваються некомпетентними. Крім того, знеособлене рішення проблеми в сукупності з систематичним збором інформації дозволяє студенту швидко отримати допомогу, так як немає необхідності чекати, поки конкретна людина буде доступна для спілкування. Нарешті, знеособлене поводження виключає можливість можливих непорозумінь, які можуть виникнути при особистій взаємодії з кимось. Такий підхід особливо корисний, практично унікальний, в ситуаціях, коли проблема складна і делікатна.

Розглянемо аналоги сервісу звернення студентів. Існує кілька аналогів сервісів, які забезпечують подібну функціональність для звернень студентів до адміністрації та інших підрозділів університетів. Розглянемо деякі з них.

Google Forms широко використовується для збору даних, включаючи звернення студентів. Google Forms – це проста у використанні платформа для створення форм і опитувань.

Microsoft Forms аналог Google Forms, інтегрований з іншими сервісами Microsoft 365. Пропонує подібні функції для створення та обробки форм.

Ticketing Systems (наприклад, Zendesk, Freshdesk) використовуються для управління зверненнями та запитами, включаючи звернення від студентів. Пропонують розширені можливості для організації та обробки запитів, включаючи автоматизацію та аналітику [1].

Canvas LMS це популярна система управління навчанням, яка має функціонал для комунікації між студентами та адміністрацією. Пропонує інструменти для обробки звернень та запитів, а також інтеграцію з іншими навчальними сервісами [2].

Moodle – відкрита платформа для управління навчанням з великою кількістю плагінів, включаючи ті, що дозволяють обробляти звернення студентів. Надає можливість кастомізації під специфічні потреби навчального закладу [3].

Custom University Portals. Деякі університети розробляють власні портали для комунікації зі студентами, які включають функціонал для подачі та обробки звернень. Пропонують спеціалізовані рішення, адаптовані під потреби конкретного навчального закладу [4].

Чат-боти в месенджерах (наприклад, Telegram, Facebook Messenger). Боти, розроблені для автоматизації процесів звернення та комунікації зі студентами. Забезпечують швидку та зручну подачу запитів через популярні платформи.

Кожен з цих сервісів має свої переваги та недоліки, і вибір конкретного рішення залежить від потреб та ресурсів навчального закладу.

Розглянемо детальніше переваги та недоліки кожного з сервісів.

Переваги Google Forms:

– простота використання. Інтуїтивно зрозумілий інтерфейс для створення та заповнення форм;

– безкоштовність. Безкоштовний для користувачів з обліковими записами Google;

– інтеграція з Google Sheets. Легко експортувати дані до таблиць для подальшого аналізу;

– доступність. Доступний з будь-якого пристрою з доступом до інтернету.

Недоліки:

– обмежені можливості кастомізації. Мало можливостей для налаштування зовнішнього вигляду та функціоналу форм;

– відсутність автоматизації. Обробка запитів вимагає ручного втручання;

– безпека даних. Обмежені можливості контролю доступу та захисту конфіденційності.

Переваги Microsoft Forms:

– інтеграція з Office 365. Легко інтегрується з іншими сервісами Microsoft, такими як Excel та SharePoint;

– простота використання. Інтуїтивно зрозумілий інтерфейс для створення форм;

– аналітика. Можливість перегляду та аналізу відповідей в реальному часі.

Недоліки:

– обмежені можливості кастомізації. Подібно до Google Forms, має обмежені можливості для налаштування;

– відсутність автоматизації. Вимагає ручної обробки запитів;

– обмеження для безкоштовних користувачів. Платні функції доступні тільки для підписників Office 365.

Переваги Ticketing Systems:

– розширені можливості. Підтримка автоматизації процесів, аналітики, та управління запитами;

– система пріоритизації. Можливість встановлення пріоритетів для обробки запитів;

– масштабованість. Підходить для великих організацій з великою кількістю запитів.

Недоліки:

– вартість. Висока вартість підписки та впровадження;

– складність налаштування. Вимагає часу та ресурсів для налаштування та інтеграції;

– високий поріг входу. Може бути складним для нових користувачів.

Переваги Canvas LMS:

– інтеграція з навчальним процесом. Підтримка комунікації між студентами та викладачами;

– гнучкість. Велика кількість вбудованих інструментів та можливостей для налаштування;

– широке використання. Популярність серед університетів та коледжів.

Недоліки:

– основний фокус на навчанні. Обмежені можливості для управління зверненнями, які не стосуються навчального процесу;

– вартість. Може бути дорогим для навчальних закладів з обмеженим бюджетом;

– складність у використанні. Потребує навчання для ефективного використання всіх функцій.

Переваги Moodle:

– відкритий код. Можливість повної кастомізації та адаптації під конкретні потреби;

– широке співтовариство. Велика кількість плагінів та додаткових модулів;

– безкоштовність. Можливість використання без ліцензійних зборів.

Недоліки:

– складність налаштування. Вимагає значних технічних знань та ресурсів для налаштування та підтримки;

– підтримка. Відсутність офіційної підтримки, залежність від спільноти користувачів;

– потреба в серверах. Вимагає наявності власних серверів для розгортання.

Переваги Custom University Portals:

– спеціалізовані рішення. Повністю адаптовані під потреби конкретного навчального закладу;

– інтеграція з внутрішніми системами. Можливість інтеграції з існуючими внутрішніми системами університету;

– контроль даних. Повний контроль над даними та процесами.

Недоліки:

– висока вартість розробки. Значні витрати на розробку та підтримку;

– тривалі терміни впровадження. Вимагає багато часу для розробки та тестування;

– залежність від розробників. Потреба в технічній підтримці та оновленнях.

Переваги Чат-ботів в месенджерах:

– зручність. Студенти можуть подавати запити безпосередньо з мобільних пристроїв;

– швидкість обробки. Автоматизоване оброблення запитів та швидкі відповіді;

– інтерактивність. Можливість створення діалогових сценаріїв для збору необхідної інформації.

Недоліки:

– обмеженість функціоналу. Може бути складно реалізувати складні функції;

– потреба в інтеграції. Вимагає інтеграції з іншими системами для повної обробки запитів;

– безпека. Необхідність забезпечення безпеки даних, переданих через месенджери.

Кожен з цих сервісів має свої унікальні переваги та недоліки, і вибір найбільш підходящого рішення залежить від конкретних потреб та ресурсів навчального закладу.

У Луцькому національному технічному університеті (ЛНТУ) створено чат-бот, який допоможе оптимізувати доступ студентів до інформації та зворотного зв’язку з викладачами, що має позитивно вплинути на якість засвоєння навчального матеріалу. Користувач починає роботу з ботом з реєстрації, активуючи його за допомогою кнопки «розпочати» (рис. 1.1) або ввівши команду «/start» у відповідне поле.


Рисунок 1.1 – Кнопка активації бота

Через API чат-бот отримує унікальний ідентифікатор користувача, закріплений за ним на платформі, і за допомогою цього ідентифікатора отримує ім'я, яке користувач ввів для спілкування у месенджері. Взаємодія з користувачем здійснюється через клавіатуру з кнопками, що пропонують різні варіанти дій, на які чат-бот реагує як на окремі команди.

Після реєстрації користувачу відображається повідомлення та надається набір доступних дій, передбачених початковою клавіатурою. Для вибору певного пункту меню потрібно натиснути відповідну кнопку. Після цього бот змінює набір доступних кнопок відповідно до вибору користувача та інформує його про зроблений вибір. Користувач може в будь-який момент повернутися до попередніх блоків меню, скориставшись кнопкою «назад», яка розміщена останньою в кожному блоці меню, крім головного.

Крім того, користувач може поставити запитання, на яке він не знайшов відповіді під час спілкування з чат-ботом (рис. 1.2), або отримати контактні дані відповідальних осіб для прямого спілкування з ними.

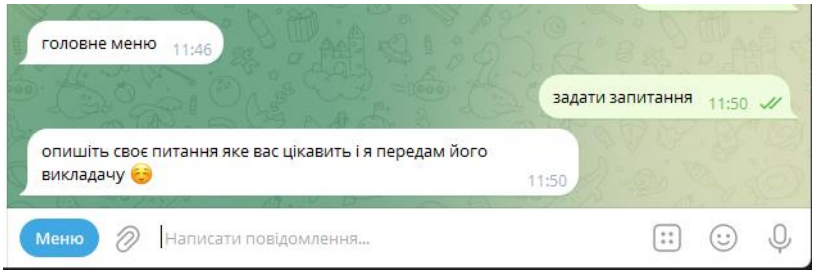


Рисунок 1.2 – Зразок діалогу при не типовому запитанні

Користувач також може отримати доступ до соціальних мереж закладу освіти через посилання, які бот надасть.

У адміністратора бота доступний розширений функціонал. Коли користувач, доданий до групи адміністраторів, авторизується в Telegram-боті, він бачить початковий набір кнопок клавіатури з додатковими адміністративними можливостями. Наприклад, адміністратор має кнопку для перегляду поточної кількості підписників бота.

Крім того, у нього є функціональне меню для надсилання повідомлень підписникам, де можна вибрати тип і зміст повідомлення.

Ще одна привілея адміністратора – можливість редагувати заздалегідь підготовлені відповіді бота, тобто ті повідомлення, які бот надсилає у відповідь на певні запитання користувачів (рис. 1.3).

Оскільки бот розміщується на одному з серверів університету, було вирішено додати до базового набору клавіатури адміністратора додаткові кнопки для керування сервером. Це дозволяє адміністраторам перезапускати ресурсні пули та очищувати кеш тимчасових файлів, що сприяє оптимізації роботи сервера.

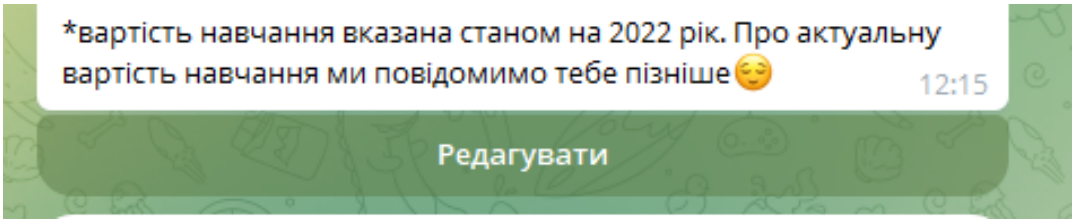


Рисунок 1.3 – Клавіатура редагування повідомлень для адміністратора

Також створено чат-бот для НПП університету, який спрощує викладачам пошук інформації та швидко надає відповіді на їхні запитання. Спочатку викладачеві потрібно обрати факультет (рис. 1.4).

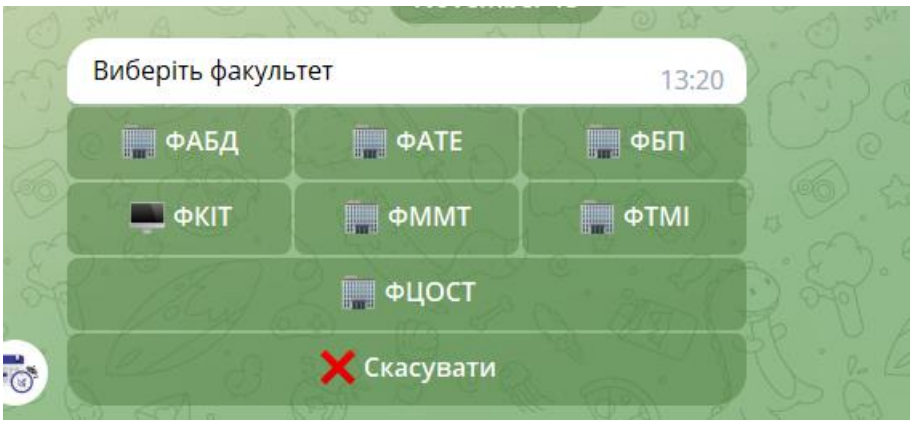


Рисунок 1.4 – Вибір факультету

Вибравши факультет та кафедру з’являється випадаючий список НПП кафедри (рис. 1.5).


Рисунок 1.5 – Вибір викладача

Розклад викладача з’являється в доступному форматі (рис. 1.6).

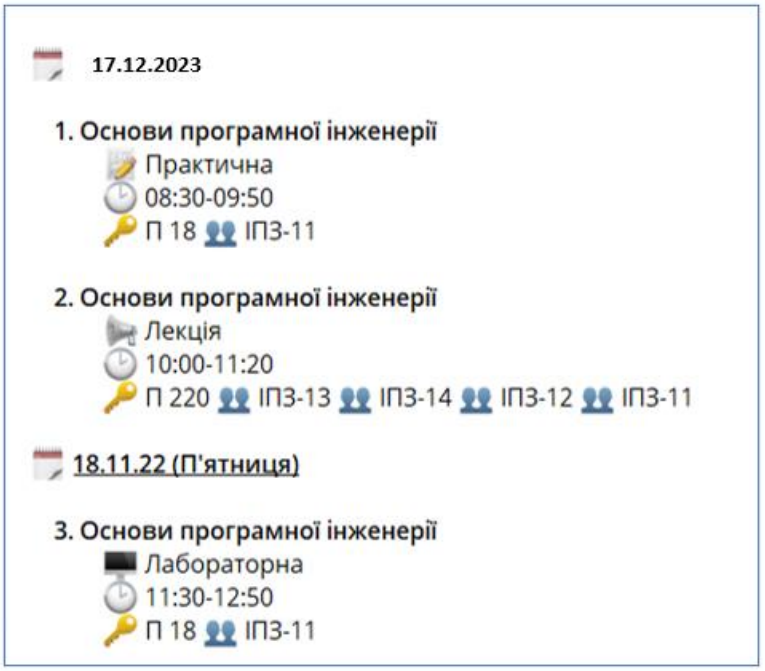


Рисунок 1.6 – Розклад викладача

Крім того функціонує в ЗВО електронний кабінет студента, де формується індивідуальний план студента (рис. 1.7).

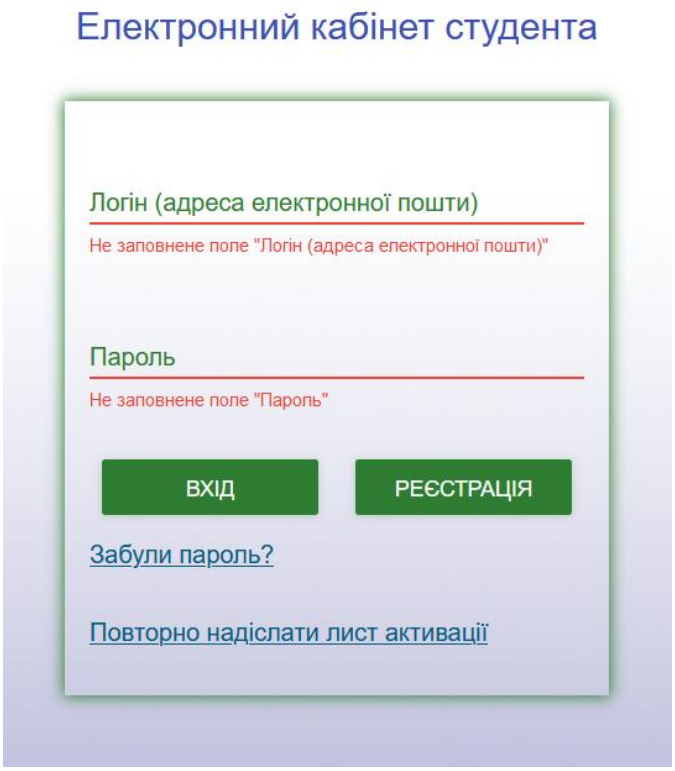


Рисунок 1.7 – Електронний кабінет студента

Опитування викладачів та студентів проводяться на окремому порталі опитувань. Доступ до опитувань здобувачі здійснюють через особистий кабінет студента (рис. 1.8).

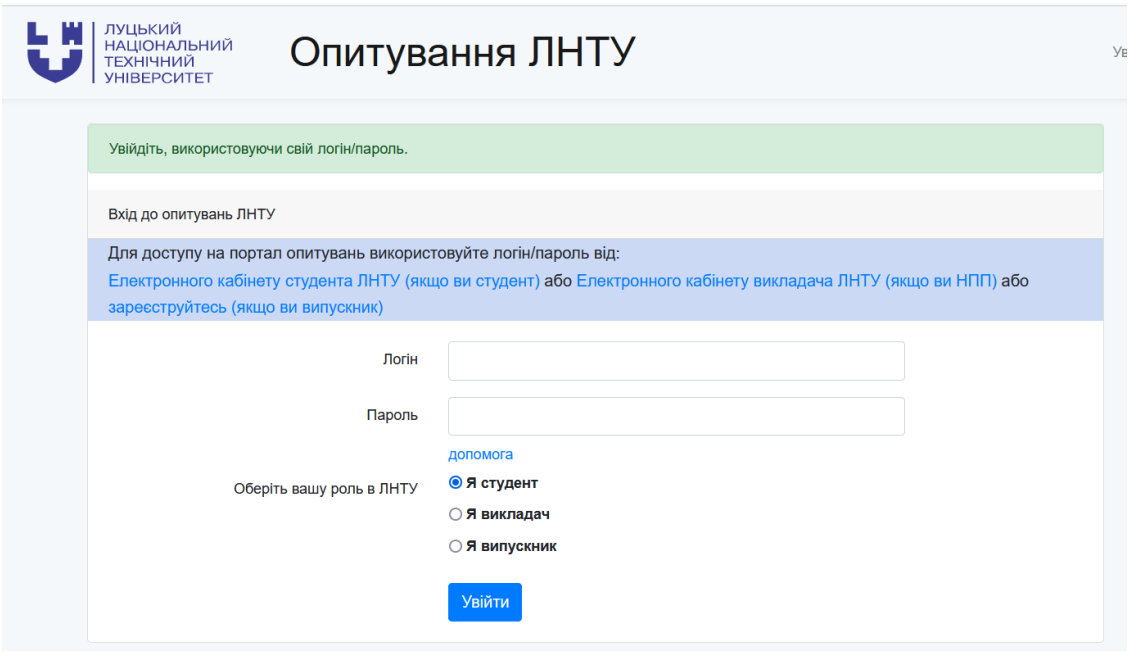


Рисунок 1.8 – Електронний кабінет студента

Варто зазначити, що розглядалися лише ті послуги, які існують на даний момент та використовуються в університеті. Такий вибір обумовлений необхідністю виключення можливості повного дублювання функціоналу, наявністю існуючих сервісів для вивчення їх можливостей, ретельним вивченням внутрішньої структури ЗВО, і, як наслідок, реалізацією затребуваних можливостей сервісу, а також аналізом запитів студентів.

Однак, потреба розробки окремого сервісу для звернень студентів ґрунтується на ряді причин:

– специфічні потреби навчальних закладів. Існуючі універсальні сервіси не завжди можуть задовольнити всі специфічні потреби навчальних закладів. Студенти часто звертаються з різними запитами, які вимагають спеціальної обробки, інтеграції з іншими системами та надання специфічної інформації. Окремий сервіс може бути адаптований до конкретних потреб університету, включаючи унікальні процеси та процедури;

– підвищення ефективності комунікації. Окремий сервіс дозволяє централізувати та стандартизувати процеси подання та обробки звернень. Це забезпечує:

1) швидкість обробки запитів. Автоматизація та стандартизація процесів зменшує час, необхідний для обробки кожного запиту;

2) прозорість. Студенти можуть відстежувати статус своїх звернень в реальному часі;

3) зниження навантаження на персонал. Автоматизовані відповіді на типові запити зменшують навантаження на адміністративний персонал;

– поліпшення обслуговування студентів. Забезпечення більш високого рівня обслуговування студентів за рахунок:

1) зручності. Інтуїтивно зрозумілий інтерфейс, доступний з різних пристроїв (комп’ютери, смартфони, планшети);

2) доступності. Можливість подавати запити в будь-який час, незалежно від робочих годин адміністрації;

– інтеграція з іншими системами університету. Окремий сервіс може бути інтегрований з іншими інформаційними системами університету, такими як:

1) системи управління навчальним процесом (LMS). Автоматичне отримання інформації про студента, його успішність та інші дані;

2) системи електронної пошти та сповіщень. Автоматичне відправлення повідомлень про статус запиту;

3) бази даних студентів. Легкий доступ до необхідної інформації для швидкої обробки запитів;

– забезпечення безпеки та конфіденційності. Окремий сервіс дозволяє краще контролювати безпеку та конфіденційність студентських даних. Це включає:

4) захист даних. Впровадження сучасних методів шифрування та контролю доступу;

5) відповідність нормативним вимогам. Забезпечення відповідності локальним та міжнародним нормам захисту даних;

– збирання та аналіз даних. Окремий сервіс може забезпечити ефективний збір та аналіз даних щодо звернень студентів, що дозволяє:

1) аналіз тенденцій. Виявлення повторюваних проблем та звернень для їх системного вирішення;

2) поліпшення процесів. Використання аналітики для оптимізації процесів обробки запитів;

3) прийняття обґрунтованих рішень. Використання даних для прийняття управлінських рішень, спрямованих на поліпшення обслуговування студентів.

Отже, розробка окремого сервісу для звернень студентів дозволяє краще задовольняти специфічні потреби навчального закладу, підвищувати ефективність комунікації та обслуговування, забезпечувати інтеграцію з іншими системами, підвищувати безпеку даних та здійснювати ефективний збір та аналіз даних. Це в кінцевому підсумку покращує досвід студентів та сприяє більш ефективному функціонуванню університету.

## 1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Виходячи із мети роботи сформулюємо такі завдання дослідження:

– проаналізувати існуючі сервіси для звернень студентів. Провести огляд сучасних сервісів для звернень студентів, які використовуються в університетах та інших навчальних закладах. Визначити функціональні можливості та характеристики цих сервісів;

– визначити вимоги до функціоналу сервісу. Визначити основні функціональні можливості сервісу, Визначити нефункціональні вимоги;

– розробити архітектуру сервісу. Визначити загальну архітектуру сервісу, включаючи клієнтську та серверну частини. Визначити технології та фреймворки, які будуть використовуватись для реалізації сервісу;

– реалізувати сервіс на мові Python з використанням відповідних фреймворків. Здійснити реалізацію веб-сервісу для студради. Розробити інтерфейс для обробки звернень студрадою;

– виконати тестування та відлагодження. Тестування з реальними користувачами для збору зворотного зв’язку та виявлення можливих проблем.

Виконання зазначених завдань дозволить створити ефективний та зручний сервіс для звернень студентів до студради закладу, що сприятиме покращенню комунікації та підвищенню комфортності навчання студентів.

# РОЗДІЛ 2
# СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

## 2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Сервіс складається з чотирьох частин: чат-бота в Telegram для студентів, веб-сервісу для обробки запитів від членів студентської ради, бази даних, в якій зберігаються всі дані, та сховища BLOB-об’єктів (Binary Large Object) для зберігання фотографій. Через чат-бот студенти можуть подавати заявки, які містять таку інформацію:

- факультет/спеціальність, де виявлено проблему;
- ім'я/анонімний запит;
- текстове повідомлення/фото.

Після відправки запиту веб-сервіс додає до кожного з них інформацію зі статусом запиту («Новий запит», «Виконується», «Закрито»), а також всі повідомлення з діалогу. У базі даних зберігаються всі запити студентів. Нижче наведена загальна архітектура сервісу (рис. 2.1).

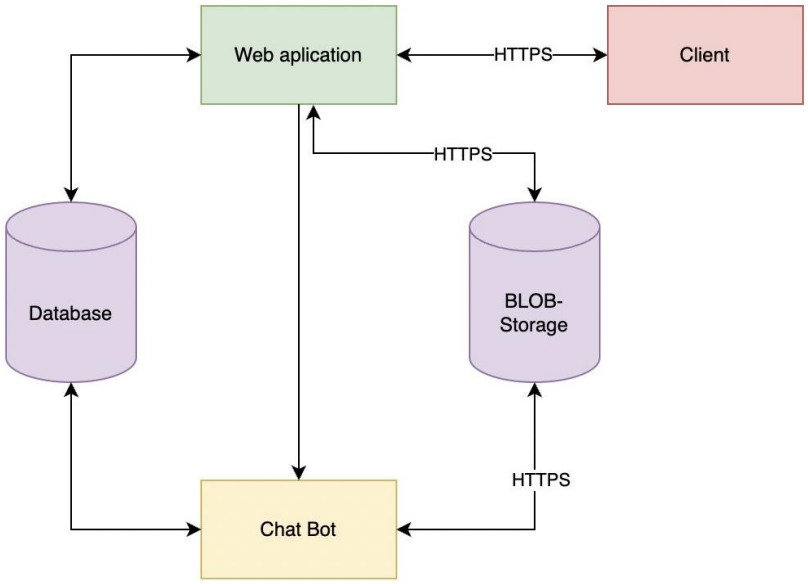


Рисунок 2.1 – Архітектура сервісу

У даній роботі був обраний патерн проектування архітектури MVT (Model-View-Template) [5]. Цей шаблон використовується у поєднанні з фреймворком Django. Модель відображає бізнес-логіку програми, а вид відображає інтерфейс і дані. Шаблон відповідає за те, як дані будуть відображатися на сторінці. Цей шаблон оформлення використовується для створення HTML для сторінки, яка містить дані, отримані з моделі та представлення даних. Основна відмінність цього фреймворку полягає в тому, що контролер відсутній, замість нього представлення виступає в ролі контролера. Він отримує запит від користувача, обробляє його, витягує дані з моделі та передає їх шаблону для відображення (рис. 2.2).

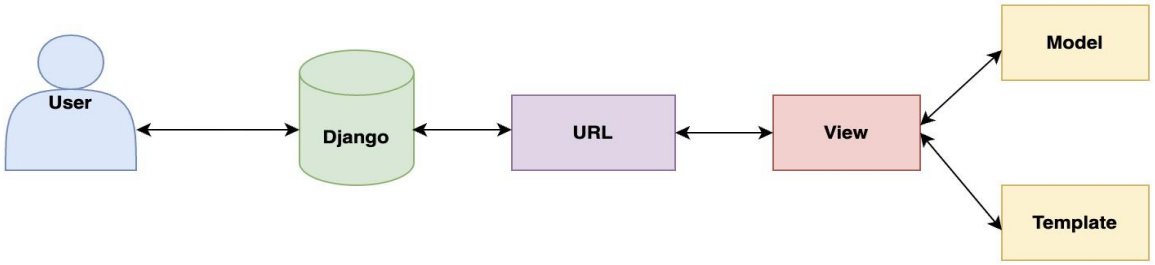


Рисунок 2.2 – Шаблон проектування архітектури MVT [5]

Для більш чіткого розуміння структури та ієрархії сервісу [6] була побудована схема пакетів за допомогою Unified Modeling Language (UML) (рис. 2.3). Прямокутники представляють пакети та зв’язки між ними. Верхній рівень схеми пакетів відображає основні компоненти програмної системи, а більш детальні рівні відображають компоненти нижчого рівня.

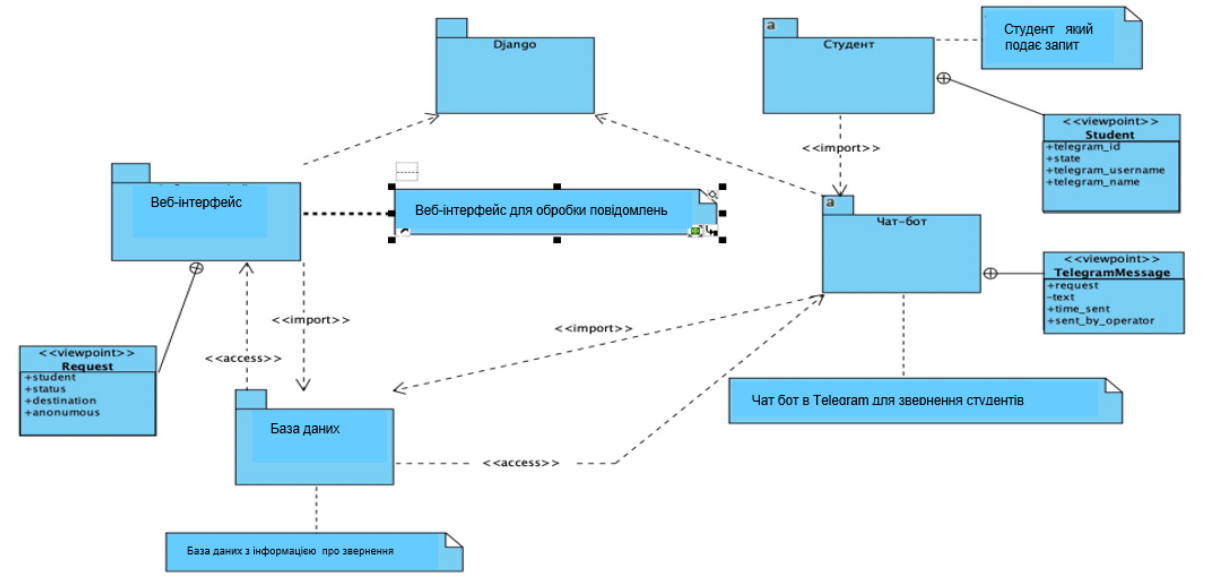


Рисунок 2.3 – Схема пакетів

Схема бази даних була побудована з трьох таблиць: Student, Request і TelegramMessage (рис. 2.4).

У таблицях відображаються такі стовпці:

– вступ студента – студент:

1) telegram_id;
2) стан;
3) telegram_username;
4) telegram_name;

– запит – параметри запиту від студента:

1) студент;
2) статус;
3) призначення;
4) безіменний;

– Telegram-повідомлення – текстова частина звернення студента або члена студентської ради:

1) прохання;
2) текст;
3) time_sent;
4) sent_by_operator.

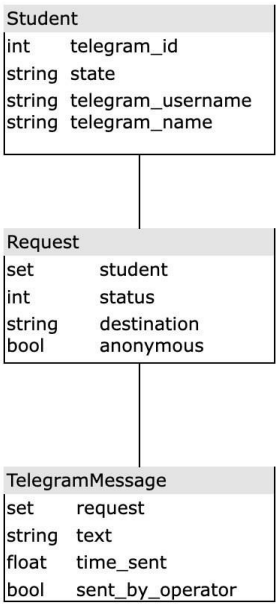


Рисунок 2.4 – Структура бази даних

Діаграма випадків використання застовується для опису функціональних вимог системи. Він складається з овалів (прецедентів), людської фігури (акторів) і ліній, що з’єднують акторів і прецедентів. Випадки використання описують, що має робити система, а суб’єктами є користувачі або інші системи, які взаємодіють із системою (рис. 2.5).

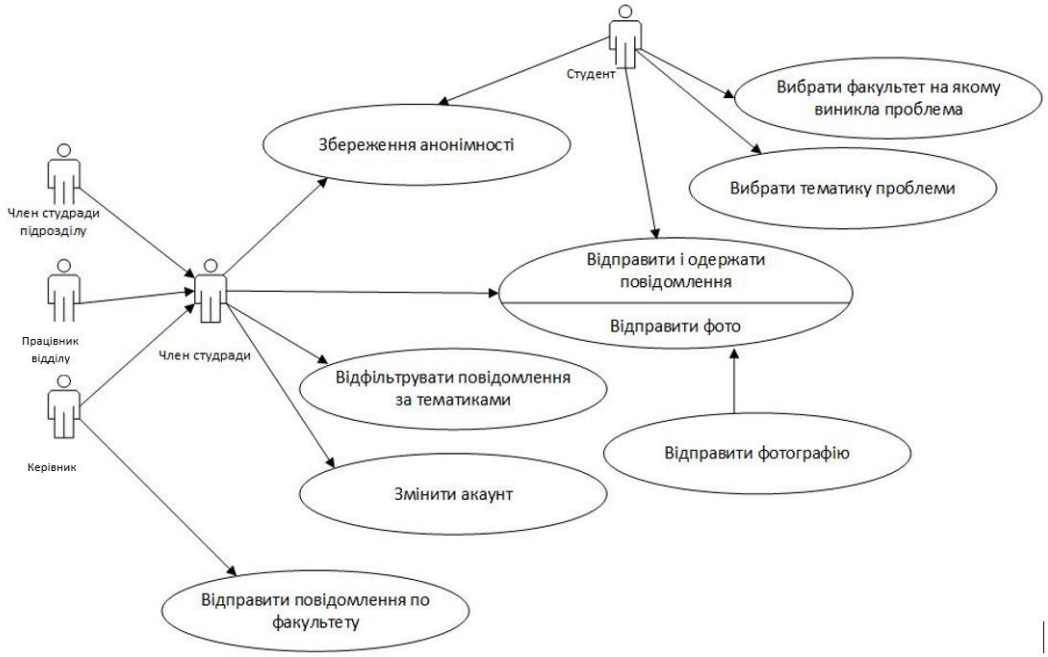


Рисунок 2.5 – Діаграма прикладів використання

## 2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для реалізації веб-сервісу та чат-бота використано мову програмування Python. Вибір Python ґрунтується на таких перевагах:

– широкий спектр фреймворків, таких як Django, Flask, Pyramid, CherryPy та багато інших, які надають інструменти для розробки масштабованих та продуктивних веб-додатків. Дізнайтеся більше про вибір фреймворку нижче;

– простота та читабельність. Python – це мова програмування з простим і легким для читання синтаксисом, що робить її доступною для розробників-початківців. Код, написаний на Python, часто здається більш лаконічним і простим, ніж код, написаний на JavaScript. Це допоможе тим членам Студентської ради, які підтримуватимуть сервіс у майбутньому, швидше розібратися в коді;

– універсальність. Python можна використовувати як для front-end, так і для back-end розробки. У той час як JavaScript використовується в основному для front-end розробки, Python можна використовувати для сценаріїв на стороні сервера, автоматизації завдань та інших функцій, пов’язаних з Інтернетом [7-9].

Крім того, для веб-сервісу використовувався фреймворк Django, оскільки він є одним з найпопулярніших фреймворків для веб-розробки на мові Python. Вибір був зроблений виходячи з наступних переваг:

– швидке прототипування та розробка веб-додатків. Цей фреймворк пропонує безліч готових компонентів, таких як аутентифікація, адміністративний інтерфейс та багато іншого, що дозволяє швидко створювати прототипи та розробляти веб-додатки:

– потужний ORM. Фреймворк включає в себе Object-Relational Mapping (ORM), що дозволяє працювати з базою даних в об’єктно-орієнтованому стилі. Це робить роботу з базою даних простішою та простішою, а також зменшує обсяг коду, необхідного для виконання операцій з базою даних;

– важливою перевагою є наявність шаблонізатора – успадкування шаблонів, що допомагає швидко створити основу для подальшої розробки веб-сторінки.

Чат-бот і веб-сервіс розгортаються за допомогою інструменту в Docker – Swarm. Цей інструмент дозволяє створювати та керувати кластером із кількох хостів, які запускають контейнери Docker. Кластер Swarm складається з кількох вузлів, які можуть бути фізичними або віртуальними машинами, об'єднаними в мережу та керованими одним API.

База даних базується на PostgreSQL. Оскільки фреймворк вже був обраний, необхідно було підібрати систему управління базами даних, яка б відповідала його вимогам. Для роботи з об'єктно-реляційним відображенням (ORM) можна використовувати тільки наступні варіанти: PostgreSQL, SQlite, MySQL. SQlite не підійшов, так як ця система лежить єдиним файлом всередині контейнера Docker і, якщо його перезапустити, то всі дані будуть очищені, чого допускати не варто, MySQL досить застаріла система, хоча і підходить, тому з трьох варіантів система PostgreSQL стала найоптимальнішою [10].

Minio було обрано як сховище. Дане сховище має високу продуктивність, що дозволяє обробляти великі обсяги даних і забезпечувати високу швидкість читання і запису. Важливим фактором при виборі стала наявність простого інтерфейсу та API, які дозволяють легко керувати сховищем та працювати з даними.

Бібліотека Telebot була використана для розробки чат-бота Telegram. Це бібліотека Python для роботи з Telegram API. Вона надає простий і зручний інтерфейс для створення та керування ботами Telegram. За допомогою цієї бібліотеки можна створювати ботів, які можуть надсилати та отримувати повідомлення, зображення, відео, аудіо та інші типи файлів.

Протокол HTTPS використовувався для обміну даними між клієнтською та серверною частинами веб-сервісу.

Для роботи з мовою програмування Python використовувалося середовище розробки JetBrains PyCharm [11].

# РОЗДІЛ 3
# РОЗРОБКА СЕРВІСУ ДЛЯ ЗВЕРНЕННЯ СТУДЕНТІВ

## 3.1 Практична реалізація об’єкта проектування

Опишемо основний функціонал чат-бота.

– вибір обсягу завдання:

1) проблема на факультеті;
2) проблема не в факультеті;

– можливість вибору факультету;

– анонімність:

1) так (у веб-сервісі вказано «анонімний запит»);
2) ні (логін або ім’я користувача вказано у веб-сервісі);

– можливість описати проблему за допомогою текстового повідомлення або прикріпивши фото;

– можливість завершити звернення фразою: «Мою проблему вирішено, дякую!»;

– можливість вибору наступних питань, не пов’язаних з факультетом:

1) проблема, пов’язана з навчальним корпусом;
2) проблема в гуртожитку;
3) проблема, пов’язана з навчально-виховним процесом;
4) я іноземний студент і мені потрібна допомога з навчальним процесом;
5) тут немає варіанту, який вам потрібен;

– можливість робити крок назад від кожної ітерації.

Основний функціонал веб-сервісу.

– авторизація за логіном та паролем;

– форма звернення, яка містить такі поля:

1) номер заявки;
2) ПІБ автора запиту (або словосполучення «анонімний запит», якщо студент бажає залишитися анонімним);

3) останнє повідомлення в діалоговому вікні;

4) тема проблеми, обрана учнем;

– після натискання на кнопку «Прийняти запит на обробку» статус зміниться на «Прийнято»;

– після натискання на кнопку «Закрити справу» статус зміниться на «Оброблено»;

– можливість бачити всі повідомлення в запиті;

– можливість надсилання текстових повідомлень у відповідь на запит;

– можливість прикріплювати фотографії;

– можливість фільтрації за тематикою запитів.

Профіль:

1) найменування підрозділу студентської ради;

2) логін (e-mail відділу);

3) пароль;

– можливість виходу з облікового запису. У веб-сервісі всього 3 групи ролей:

– DIO має доступ до всіх запитів у системі;

– LSS доступ тільки до запитів свого підрозділу;

– відділ має доступ лише до звернень за своєю тематикою:

1) проблема навчального корпусу – господарський відділ;

2) проблема в гуртожитку – деканат;

3) проблема, пов’язана з освітнім процесом – навчальний відділ;

4) я іноземний студент і мені потрібна допомога з навчальним процесом – міжнародний відділ;

5) тут немає потрібної опції – загальний список запитів.

Взаємодія з чат-ботом. При вході в систему студент бачить стартовий екран (рис. 3.1), а після активації діалогового вікна за допомогою кнопки «Пуск» надсилається вітальне повідомлення (рис. 3.2). Після натискання на кнопку «Подати апеляцію» потрібно вибрати, де виникла проблема (на факультеті/не на факультеті) (рис. 3.3).

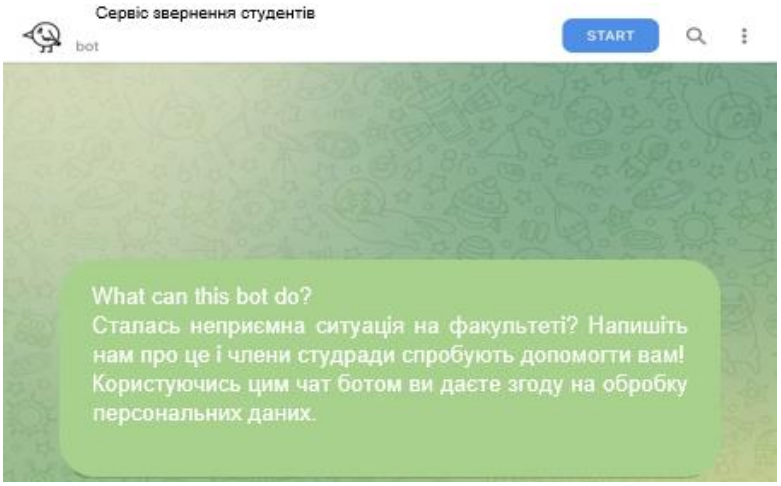


Рисунок 3.1 – Стартовий екран чат-бота

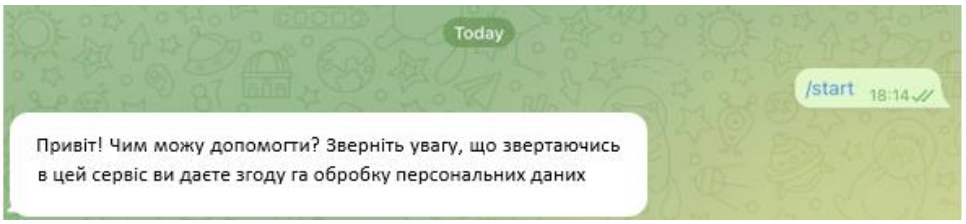


Рисунок 3.2 – Вітальне повідомлення

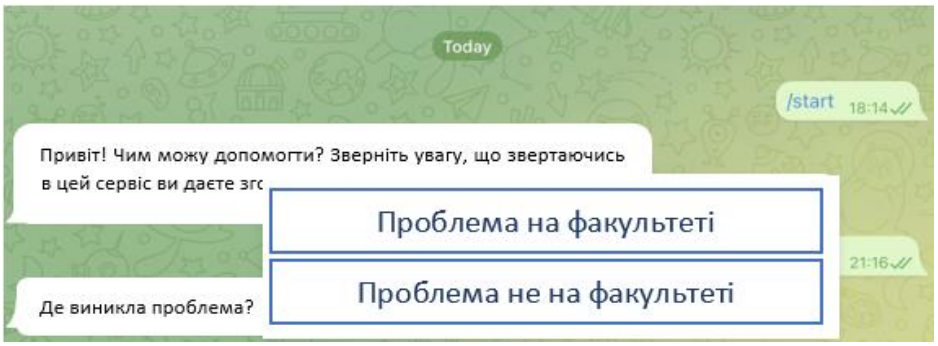


Рисунок 3.3 – Вибір місця виникнення проблеми

Подача заявки студентом. Якщо була обрана задача на факультеті, студенту пропонується вибрати факультет (рис. 3.4). А потім вибрати загальну тему звернення (рис. 3.5).

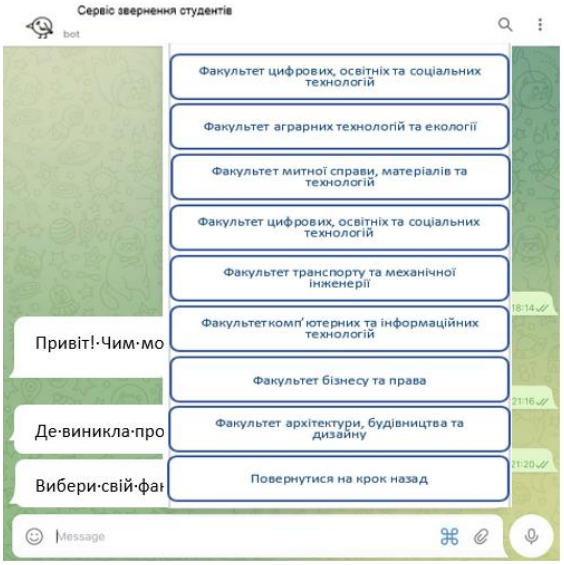


Рисунок 3.4 – Вибір факультету, на якому виникла проблема

Якщо задача була обрана за межами факультету, студенту відразу пропонується вибрати одну із запропонованих задач (рис. 3.5).

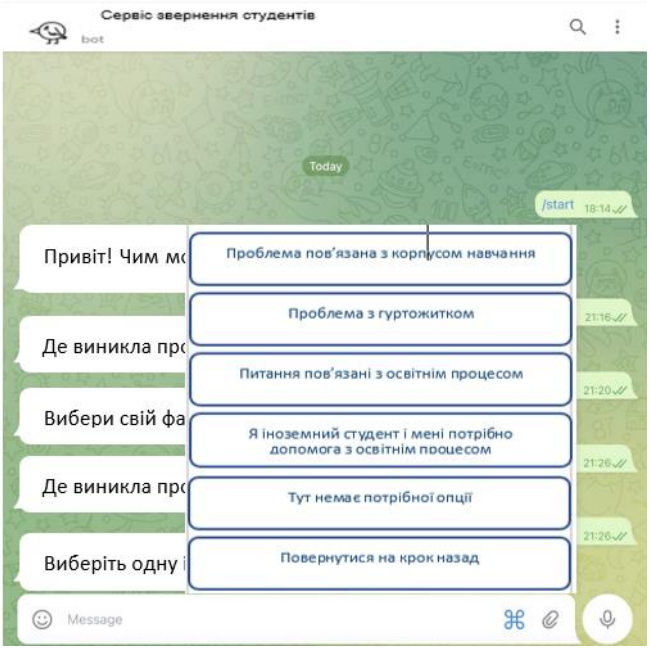


Рисунок 3.5 –Вибір проблемної ділянки не на факультеті

Наступним кроком студент має вибрати, чи залишатися йому анонімним, студент може вибрати, залишатися анонімним чи не залишатися анонімним (рис. 3.6).

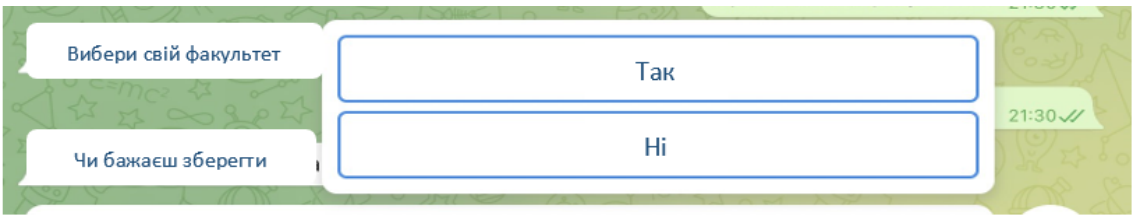


Рисунок 3.6 – Збереження анонімності

Останнім кроком у зверненні є текстовий опис проблеми з можливістю надіслати фото, у відповідь член студентської ради може надіслати повідомлення за допомогою веб-сервісу (рис. 3.7).

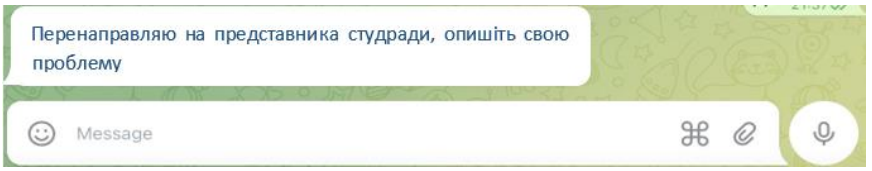


Рисунок 3.7 – Крок опису задачі студента

Студент може закрити справу, натиснувши на кнопку «Мою проблему вирішено, дякую!». (рис. 3.8). Після цього статус заявки змінюється на «Оброблено» і студент отримує повідомлення всередині чат-бота. Аналогічна процедура, якщо заяву закриває член студентської ради.

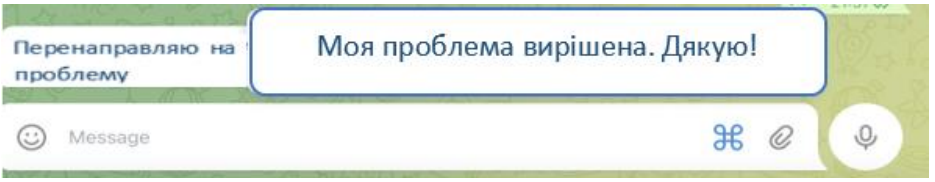


Рисунок 3.8 – Можливість вирішити справу студентом

### 3.2 Взаємодія з веб-сервісом

Опишемо авторизацію та зміну облікового запису в системі. Для входу в систему необхідно ввести логін і пароль (рис. 3.9).


Рисунок 3.9 – Вікно авторизації у веб-сервісі

Всі запити студентів доступні для перегляду тільки з облікового запису.

Звернення студентів доступні для перегляду для відділів, залежно від інформації, наданої студентом у чат-боті. Іншими словами, веб-сервіс реалізує сценарій автоматичного підбору адресата в залежності від набору параметрів запиту студента. Тобто, якщо студент вказав на проблему на факультеті, то звернення буде направлено до студентської ради факультету.

Щоб змінити обліковий запис, потрібно натиснути на кнопку «Вийти з облікового запису».

Розглянемо роботу із запитами. Далі член студентської ради потрапляє на головний екран зі зверненнями, при натисканні на кнопку «Переглянути» всередині інтерфейсу відкривається діалог зі студентом, і якщо студент вирішить залишитися анонімним, то буде вказано «Анонімний запит», в іншому випадку ім'я з облікового запису Telegram.

Якщо звернення не прийнято або член студентської ради ще не написав жодного повідомлення студенту у відповідь, то воно підсвічується червоним кольором і має статус «Нова заявка», після натискання на кнопку «Взяти в роботу» статус змінюється на «Обробка». Після цього член студентської ради може написати студенту відповідне повідомлення і він отримає його в чат-боті, при цьому червоне підсвічування заявки зникне (рис. 3.10).

При натисканні на кнопку «Закрити запит» статус змінюється на «Оброблений» і студенту надсилається повідомлення у відповідь.


Рисунок 3.10 – Головний екран веб-сервісу

Після натискання кнопки «Прийняти заявку» статус змінюється на «Прийнято» і член студентської ради може надіслати повідомлення студенту.

Член студентської ради може фільтрувати звернення за однією або декількома загальними темами.

### 3.3 Тестування веб-сервісу

Мета тестування – перевірити функціональність, безпеку та ефективність сервісу для подання звернень студентів до студентської ради з метою забезпечення його коректної та надійної роботи [12].

Розглянемо такі сценарії тестування:

– подача звернення. При цьому користувач відкриває інтерфейс для подачі звернення, доступний через веб-сайт. На формі подачі звернення користувач заповнює всі обов’язкові поля, такі як заголовок звернення, опис проблеми та контактні дані. Після заповнення користувач натискає кнопку «Надіслати». Після надсилання звернення система перевіряє правильність заповнення форми та зберігає інформацію про звернення в базі даних.

Користувач отримує підтвердження успішної подачі звернення і унікальний ідентифікатор для відстеження статусу звернення;

– відстеження статусу звернення. Користувач відкриває інтерфейс для відстеження статусу свого звернення, який доступний на веб-сайті. Після цього він вводить у форму відстеження унікальний ідентифікатор звернення, отриманий при подачі звернення. Після введення ідентифікатора система перевіряє базу даних та відображає поточний статус звернення. Користувач перевіряє, що статус відображається коректно, що дає йому можливість відстежувати процес обробки свого звернення;

– авторизація та автентифікація. Система перевіряє ідентифікаційні дані користувача, такі як логін та пароль, для забезпечення доступу до функціоналу сервісу. У випадку спроби доступу до функціоналу подачі звернення без авторизації, система блокує доступ та повідомляє користувача про необхідність увійти в свій обліковий запис. Також, при спробі отримати доступ до адміністративного функціоналу без відповідних прав, система відхиляє запит та повідомляє користувача про недостатність прав для здійснення цієї операції. Ці механізми забезпечують безпеку та конфіденційність даних, а також обмежують доступ до функціоналу лише авторизованим користувачам з відповідними правами;

– перегляд звернень для адміністратора. Адміністратор увіходить в систему за допомогою своїх облікових даних. Після входу в систему, адміністратор перевіряє, що звернення відображаються коректно, і усі необхідні дані про звернення присутні. Він також перевіряє можливість обробки та зміни статусу кожного звернення, щоб забезпечити ефективну роботу з ними;

– тестування з використанням фокус-групи. Організовується зустріч зі студентами, які мають намір скористатися сервісом. Учасникам групи надається можливість використати сервіс та подати звернення через нього. Після використання сервісу учасники фокус-групи обговорюють свої враження та досвід використання. Збирається зворотний зв’язок та оцінки щодо зручності використання, якості обробки звернень та загального враження від сервісу. Цей

підхід дозволяє здійснити оцінку сервісу з боку реальних користувачів і зрозуміти їхні потреби та очікування.

Проведення тестування за вищезазначеними сценаріями дозволило переконатися у коректній роботі та безпеці сервісу для подання звернень студентів до студентської ради. Результати тестування допоможуть виявити можливі проблеми та вдосконалити функціональність та ефективність сервісу.

# ВИСНОВКИ

Результатом цієї кваліфікаційної роботи став сервіс, який підключає чат-бот у Telegram для збору звернень від студентів та веб-сервіс для опрацювання звернень членами студентської ради. В ході роботи були проаналізовані існуючі рішення, можливі підходи до побудови архітектури сервісу та інструменти розробки.

Зокрема, було вирішено такі завдання:

– проведено огляд сучасних сервісів для звернень студентів, які використовуються в університетах та інших навчальних закладах. Визначено функціональні можливості та характеристики цих сервісів. Виявлено сильні та слабкі сторони існуючих рішень, що допомогло сформувати вимоги до нашого сервісу;

– визначено основні функціональні можливості сервісу, такі як можливість подачі звернень через чат-бот у Telegram, відстеження статусу звернення, автоматичне перенаправлення звернень до відповідних підрозділів та сповіщення про зміни статусу звернень. Визначено нефункціональні вимоги, включаючи безпеку, продуктивність, масштабованість та зручність використання;

– визначено загальну архітектуру сервісу, включаючи клієнтську та серверну частини;

– здійснено реалізацію чат-бота у Telegram з використанням бібліотеки python-telegram-bot, який дозволяє студентам подавати звернення та відстежувати їх статус; виконання тестування та відлагодження. Проведено інтеграційне тестування для перевірки коректної взаємодії всіх компонентів сервісу. Виконано тестування з реальними користувачами для збору зворотного зв’язку та виявлення можливих проблем, що дозволило вдосконалити сервіс.

Виконання зазначених завдань дозволило створити ефективний та зручний сервіс для звернень студентів до студради навчального закладу. Новий сервіс сприяє покращенню комунікації між студентами та адміністрацією,

підвищенню ефективності обробки студентських запитів та загальному підвищенню якості обслуговування студентів. Це, в свою чергу, допоможе створити більш комфортні умови для навчання та взаємодії студентів з адміністрацією університету.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is a ticketing system? URL: https://www.zendesk.com/blog/ticketing-system/ (дата звернення: 10.05.2024).
2. Canvas LMS. URL: https://www.instructure.com/canvas (дата звернення: 10.05.2024).
3. Moodle. URL: https://moodle.org/?lang=uk (дата звернення: 10.05.2024).
4. Custom University Portals. URL: https://budibase.com/portals/templates/university-portal-app/ (дата звернення: 10.05.2024).
5. MVT (Model-View-Template). URL: https://angelogentileiii.medium.com/basics-of-django-model-view-template-mvt-architecture-8585aecffbf6 (дата звернення: 15.05.2024).
6. Unified Modeling Language (UML). URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ (дата звернення: 15.05.2024).
7. Python. URL: http://pythonguide.rozh2sch.org.ua/ (дата звернення: 20.05.2024).
8. Олексій Васильєв. Програмування в PYTHON. Теорія і практика. Видавництво: Ліра-К., 2023. 462 с.
9. Пол Беррі. Книга Head First. Python. Видавництво «Фабула», 2021. 607 с.
10. PostgreSQL. URL: https://www.postgresql.org/ (дата звернення: 20.05.2024).
11. JetBrains PyCharm. URL: https://www.jetbrains.com/pycharm/ (дата звернення: 20.05.2024).
12. Що таке тестування ПЗ, його етапи, види, інструменти. https://university.sigma.software/what-is-software-testing/ (дата звернення: 20.05.2024).