ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

ДИПЛОМНА РОБОТА

на тему: «**Розробка Телеграм чат-боту для абітурієнтів Державного університету телекомунікацій»**

Виконав: студент 4 курсу, групи ПД–43

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Єремеєв А.Р.

(прізвище та ініціали)

Керівник Трінітіна Н. А.

(прізвище та ініціали)

Рецензент_____

(прізвище та ініціали)

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрям підготовки - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О.В. Негоденко

“ ” 2023 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Єремеєву Андрію Руслановичу

(прізвище, ім‘я, по батькові)

1. Тема роботи: «Розробка Телеграм чат-боту для абітурієнтів Державного університету телекомунікацій »

Керівник роботи Трінітіна Н. А.

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

Затверджені наказом вищого навчального закладу від “ 24 ” лютого 2023 року №26

2. Строк подання студентом роботи “ 1 ” червня 2023 року

3. Вхідні дані до роботи:

3.1 Науково-технічна література, пов’язана з розробкою телеграм чат – боту.

3.2 Технічна документація баз даних та комунікації між системами.

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

4.1 Аналіз обов’язків розроблюваної системи

4.2 Аналіз та дослідження існуючих аналогів

4.3 Тестування розробленої системи

5. Перелік демонстраційного матеріалу

5.1 Мета, об’єкт та предмет дослідження

5.2 Аналоги

5.3 Технічне завдання

5.4 Стек технологій

5.5 Висновки

6. Дата видачі завдання “ 25 ” лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23-27.02.23	Виконано
2	Аналіз та дослідження існуючих аналогів	28.02.23-10.03.23	Виконано
3	Моделювання, проектування системи	13.03.23-24.03.23	Виконано
4	Розробка основного функціоналу системи	27.03.23-28.04.23	Виконано
5	Тестування системи	01.05.23-05.05.23	Виконано
6	Вступ, реферат, висновки	15.05.23-19.05.23	Виконано
7	Розробка основних демонстраційних матеріалів	22.05.23-26.05.23	Виконано
8	Попередній захист роботи		Виконано
9	Здача роботи	01.06.23	Виконано

Студент

(підпис) (прізвище та ініціали)

Керівник роботи

(підпис) (прізвище та ініціали)

Реферат

Випускна кваліфікаційна робота складається з 81 сторінок, 32 рисунків, 23 таблиць, 28 джерел, 3 додатків.

Ключові слова: телеграм-бот, дизайн, розробка, веб, Telegram, API.

Об’єктом дослідження є програмне забезпечення, що розробляється для допомоги абітурієнтам при вступі до державного університету телекомунікацій.

Метою роботи є розробка телеграм бота, а також модуля адміністратора для управління контентом телеграм бота.

У ході дослідження було проведено аналіз вимог до програмного забезпечення телеграм-бота, аналіз архітектур та інструментів для реалізації програмного продукту.

За результатами дослідження було розроблено та розроблено програмне забезпечення телеграм-бота, а також модуль адміністрування.

Ступінь впровадження: на даний момент модуль працює, знаходиться на стадії тестування. Сфера застосування: програмне забезпечення може використовуватися скрізь, де є мережа, і не потребує додаткових пристроїв. Направлено допомагати абітурієнтам вступати до державного університету телекомунікацій. Основна аудиторія, яка користуватиметься цим програмним забезпеченням, це абітурієнти державного університету телекомунікацій, а також ті, хто обирає ВНЗ для вступу та потребує короткої інформації про ВНЗ.

Економічна ефективність/цінність роботи: керуючись важливістю економії часу в процесі розробки програмного забезпечення.

Функціональність програмного забезпечення можна розширити шляхом додавання додаткових модулів або підключення інтелектуальних систем. Також в майбутньому планується опублікувати результат роботи у відкритому доступі на GitHub.

ЗМІСТ

ВИЗНАЧЕННЯ, ПОЗНАЧЕННЯ, СКОРОЧЕННЯ

ПП – програмний продукт

ПЗ – програмне забезпечення

API – Application Programming Interface

ІС – Інформаційна система.

UML – Unified modeling language

JSON – JavaScript object notation

Модуль – це окрема частина програмного коду, фрагмент програми, що повністю самостійно виконує своє завдання

ВСТУП

Державний університет телекомунікації є одним з ключових навчальних центрів у сфері інформаційних технологій, економіки та телекомунікацій. Вибір напрямів підготовки та освітніх програм досить масштабний, у зв'язку з чим абітурієнти змушені витрачати години, дні і тижні на те, щоб з'ясувати цікаві для них подробиці про кожного великого представника в сфері освітніх послуг.

Актуальність роботи продиктована ситуацією на ринку інформаційних технологій. Щороку він поповнюється новими програмними продуктами, веб-сервісами та мобільними додатками, за допомогою яких можна будувати не лише комунікації, а й сприяти зміцненню бренду університету як освітнього центру.

Метою даної бакалаврської роботи є розробка програмного забезпечення Applicant Bot, яке дозволяє автоматизувати завдання, пов’язані з оповіщенням абітурієнтів та взаємодією з ними в приймальній комісії.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– Визначення вимог, які має мати програмне забезпечення;

– Визначення вимог, яким має відповідати програмне забезпечення для керування вмістом цього програмного забезпечення;

– Проектування архітектури програмного забезпечення;

– Розробка зручного інтерфейсу адміністративного модуля для телеграм-ботів;

– Впровадження розробленого програмного забезпечення.

Головним привілеєм кожного закладу вищої освіти є випуск компетентних спеціалістів, затребуваних на ринку праці. В умовах високої конкуренції серед вищих навчальних закладів необхідно підтримувати комунікацію з абітурієнтами.

Таким чином, рішення про створення чат-бота було продиктоване бажанням оптимізувати діяльність співробітників і студентів, які проходять практику в приймальній комісії університету, які змушені витрачати багато часу на трансляцію інформації, що міститься у відкритих джерелах.

Практичне значення розробленого чат-бота надається конкретним користувачам – вступникам державного університету телекомунікацій.

Як джерело інформації використано офіційну документацію для розробників PHP, накази та інші документи, що містять інформацію про кількість бюджетних місць, форми навчання та напрями підготовки.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Аналіз методів взаємодії університету зі студентами

Повсюдно шляхи взаємодії приймальної комісії університету з абітурієнтами можна виділити двома способами: очний і дистанційний. Під внутрішніми методами мається на увазі:

1. Консультування членів приймальної комісії в університеті;
2. Перебування заявника на Дні відкритих дверей;
3. Залучення здобувача до наукових заходів, які проводить університет (олімпіади, конференції тощо
4. Залучення здобувача до творчих заходів, які проводить університет («День відкритих дверей», конкурси тощо).

Дистанційні методи – це:

1. Телефонне консультування абітурієнта працівником приймальної комісії;
2. Консультування вступника поштою працівником приймальної комісії;
3. Консультування по електронній пошті вступника працівником приймальної комісії;
4. Консультування вступника працівником приймальної комісії на сайті приймальної комісії; (Рис.1.1)
5. Консультування абітурієнта працівником приймальної комісії в соціальних мережах («Instagram», «Facebook» тощо); (Рис.1.2)
6. Консультування вступника працівником приймальної комісії в месенджері.

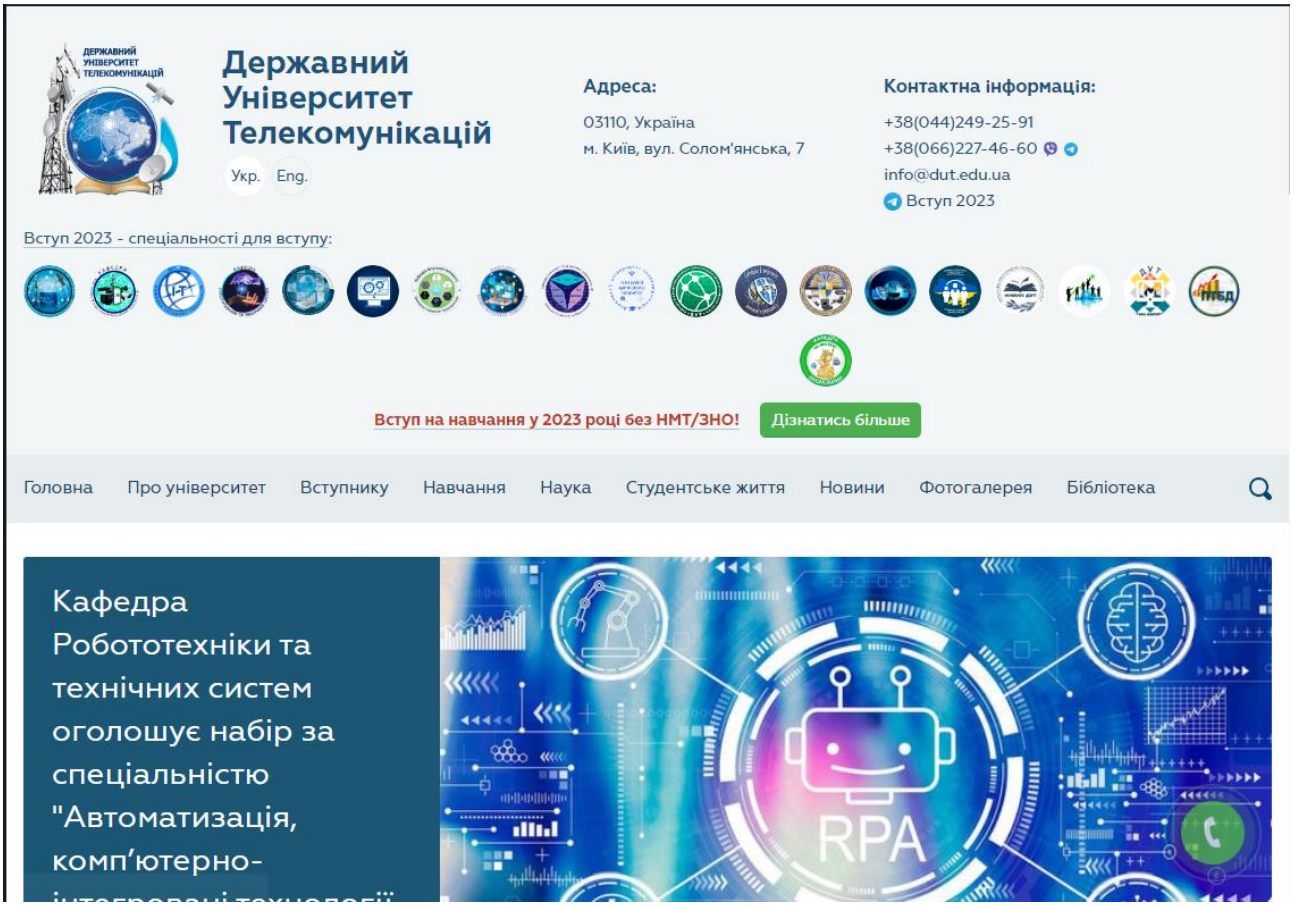


Рисунок 1.1 – Сторінка приймальної комісії на офіційному сайті університету

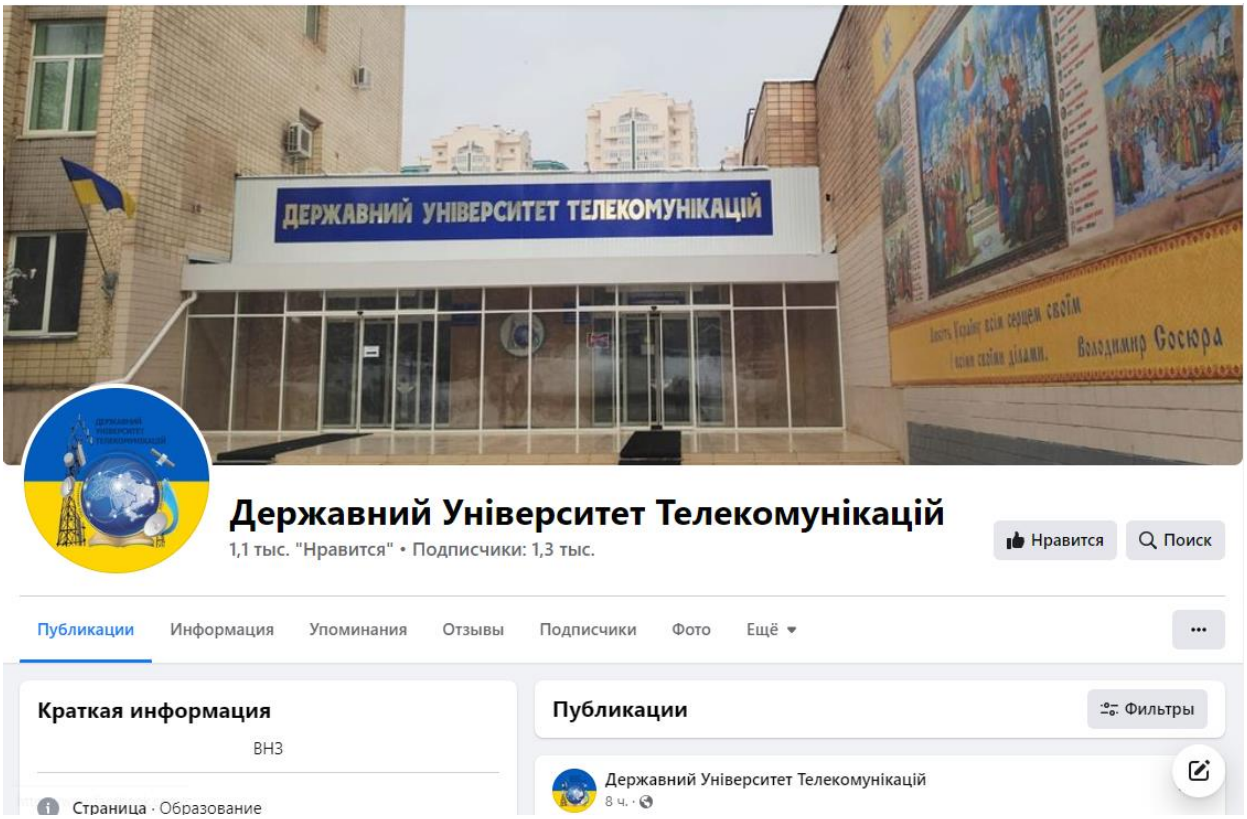


Рисунок 1.2 – Офіційна сторінка університету в соціальній мережі «Facebook»

На поточний період приймальна комісія динамічно освоює дистанційні способи взаємодії з абітурієнтами. Процедура збереження «старих», очних методів і засобів взаємодії наближається різко і досить довго, незважаючи на очевидні переваги дистанційних методів взаємодії, серед яких: загальнодоступність, низька вартість, практичність абсолютно для всіх. учасники взаємодії, зрозумілість, ймовірність не втратити історію листування та ін.

Месенджери - до сьогоднішнього дня вважалися не охопленими платформою, однак ця випускна кваліфікаційна робота покликана усунути та компенсувати це упущення. Розберемо більш детально ключові недоліки методів очної взаємодії:

1. Обмеження за часом. Приймальна комісія має чіткий режим роботи, обмежений буднями з дня на день. Двері приймальної комісії зачиняються о 16:00, працівники залишають свої робочі місця і повертаються лише на 4 години в суботу. Досить часто університет втрачає свій контингент через обмежений термін консультування та функціонування.

2. Незадовільна «прозорість» спілкування. При спілкуванні з працівником приймальної комісії за допомогою Інтернету – наприклад, соціальних мереж, електронної пошти чи месенджерів, історія листування зберігається. До нього завжди можна повернутися, якщо необхідно завершити процедуру оскарження.

3. Великі часові витрати. Організація спеціалізованих заходів в інтересах абітурієнтів, які мають усі шанси послужити «майданчиком для спілкування» абсолютно для всіх зацікавлених сторін (абітурієнтів, батьків, співробітників університету), займає величезну кількість часу, який дуже рідко «оплачується» за кількістю присутніх. Підводячи підсумки короткого огляду, можна зробити висновок, що настав час зробити основну ставку на дистанційні форми взаємодії з абітурієнтами, зокрема, за допомогою мережі Інтернет. А безпосередньо, консультування абітурієнтів за допомогою:

1. Офіційний сайт;
2. Сторінки та акаунти в соціальних мережах;
3. Посланники.

Водночас месенджерів все ще вважаються незахищеною платформою, яка, ймовірно, є більш комфортним засобом встановлення контакту між абітурієнтами та приймальною комісією.

1.2. Аналіз аудиторії користувачів месенджерів

Месенджери стають без винятку найпоширенішим каналом спілкування, в тому числі публічним простором з підтримкою чатів і каналів. Вже зараз можна говорити про те, що канали в месенджерах стали зручною базою для використання контенту для багатьох користувачів соціальних мереж. При споживанні новинного контенту користувачі Telegram вибирають колекції з анонсами новин від агрегаторів і ці канали, які публікують медіаконтент 1-2 рази на день – їхня аудиторія набагато більша, ніж, наприклад, ЗМІ, що бомбардують передплатників новинами протягом дня. Месенджери використовуються не тільки для

індивідуального спілкування, а й для спілкування в публічних чатах і читання публічних каналів, що є одним із трендів 2019 року.

Чати та канали перетворюють месенджери на соціальні мережі та залучають нових користувачів. Законодавцем трендів тут, безсумнівно, є Telegram, який, мабуть, і орієнтує месенджер на розширення з такими вражаючими темпами. Підводячи підсумок, залишається лише відзначити, що тенденція зростання інтересу до месенджерів продовжується. На кумулятивному тлі подвійного збільшення обсягу згадок Telegram став лідером зростання з чотириразовим темпом. Зростання інтересу до Telegram пов'язане з активним використанням публічних каналів - за рік кількість користувачів україномовних каналів зросла в рази - від сотень до 10 тисяч. Канали з рідкісним авторським контентом – розважальними, освітніми, а також канали з важливими інформативними програмами та політикою [2].

Відповідно до аналізу, проведеного незалежним виданням «Telegram-store», вік і соціальний статус користувачів в україномовному секторі поділяються наступним чином:

– вік від 18 до 35 років;

– отримує або має вищу освіту;

– має інтелектуальну працю (значна частка it-експертів).

Функціонал месенджерів, зокрема Telegram, орієнтований насамперед на мобільну аудиторію, яка швидко використовує телефони для роботи чи відпочинку: користувачеві зовсім не потрібно заходити на сайт, достатньо надіслати повідомлення боту. Як висновок за розділом можна зазначити такі твердження:

1. Конкуренція з боку закладів, що надають освітні послуги, величезна як ніколи.

2. Загальна кількість вищих навчальних закладів України перевищує 2 тисячі.

3. У розділі здійснено класифікацію способів взаємодії абітурієнтів із приймальною комісією ВНЗ. Виявлено два фундаментальних методи.

4. Зроблено висновок, що дистанційний спосіб взаємодії вважається більш ефективним.

5. Визначено низку видів дистанційного способу взаємодії, з яких максимальний інтерес у рамках роботи представляє використання месенджерів як платформи для взаємодії між абітурієнтами та приймальною комісією.

6. В результаті аналізу цільової аудиторії месенджерів можна зробити висновок, що месенджер Telegram вважається більш захоплюючою платформою для спілкування.

1.3. Огляд існуючих рішень

Так як додаток розроблялося для університету, бажано розглянути конкурентів. На сьогоднішній день в університетах є 2 більш-менш придатних для порівняння рішення, також найближчих за необхідним функціоналом: «Абітурієнт ТПУ» і «Хочу в КДУ».

«Абітурієнт ТПУ» — мобільний додаток для абітурієнтів, які вступають до Телекомунікаційного політехнічного університету [3]. Додаток має досить приємний інтерфейс, з можливістю входу в особистий кабінет і редагування персональних даних, також в додатку є можливість відповідати на питання вікторини, тим самим заробляючи бали з можливістю виграти один з головних призів.

«Хочу в КДУ» — мобільний додаток для абітурієнтів Київського державного університету [4]. У програмі багато функцій, але коли ви натискаєте на них, половина з них перенаправляє на веб-сайт, таким чином втрачаючи автономність програми. Також реалізовано функціонал «Калькулятора ЄДІ», сенс якого полягає у виборі предметів, що складаються на ЄДІ, на основі яких складається ймовірність вступу абітурієнта на певні напрями.

Згідно з даними впровадження програмного забезпечення, всі без винятку концепти є завершеними програмними продуктами і забезпечують тією чи іншою

мірою наявність основних функцій для комфортної роботи абітурієнтів. Для наочного уявлення про переваги та недоліки систем складено таблицю 1.1.

Таблиця 1.1 - Порівняння функціональних можливостей систем

Параметри	Додатки		
	«Телеграм бот ДУТ»	«Абітурієнт ТПУ»	«Хочу в КДУ»
Окреме місце у телефоні для програми	+	+	+
Можливість перегляду інформації про напрямки вишу	+	+	+
Можливість перегляду новин та заходів	+	+	+
Дизайн	+	-	-
Кросплатформеність	+	+	+
Автономність	+	-	-
Зручний та чуйний інтерфейс	+	-	-
Отримання інформації на пошту	+	-	-

Виходячи з таблиці 1.1 видно, що найбільш повно включає характеристики еталонного програмного продукту для абітурієнтів ДУТ.

Ви також можете помітити, що ця конкретна програма найбільш проста у використанні.

Виходячи з цих факторів, було вирішено розпочати розробку програмного забезпечення Telegram Bot.

Перш за все, ніж в роботі буде наведено покроковий алгоритм розробки чат-бота, слід повністю визначитися з його основним призначенням. По-перше, взаємодія через месенджер породжує значний рівень залучення. По-друге, можна відзначити швидкість взаємодії з чат-ботом. Електронна пошта та соціальні мережі не завжди є найкращим рішенням. Тому що користувач не завжди уявляє, хто саме і коли дасть йому відповідь. Включно з найбільшими групами в соціальних мережах, вони жодним чином не гарантують надання миттєвої відповіді, яка необхідна заявнику або користувачеві в даний момент. Користувач пише і йде, чекаючи (або не чекаючи) відповіді, через те, що сама модель спілкування жодним чином не враховує миттєвий інтерактив.

З чат-ботом все інакше без винятку – користувач має змогу миттєво отримувати відповідь на свої запитання. Близькість, швидкість і перелік можливостей - ось на чому розкручуються месенджери. Компанії просто повинні бути там, де є їхні користувачі, спілкуватися з ними на одному рівні. І з цієї точки зору такий месенджер, як Telegram, є чудовою платформою для взаємодії з суспільством. Як ззовні університету, так і зсередини [5]. Споживачі у віці 18-35 років є найбільш складною і цікавою цільовою аудиторією для рекламодавців. Звичайні соціальні мережі вже не готові тримати їх інтерес дуже довго.

Підводячи підсумок, можна відзначити наступні переваги використання месенджера приймальної комісії університету Telegram:

– Розваги – Чат-бот можна розширити за допомогою цікавого списку функцій, таких як ігри, опитування та інші форми взаємодії, призначені для «залучення» користувача до розмови.

– Мобільність і популярність за платформами - Telegram вважається кросплатформенним додатком, до якого легко отримати доступ не тільки з мобільного телефону, а й з персонального ПК

– Миттєвий і безпечний - розробники Telegram неодноразово говорили про його безпеку, яка здійснюється за рахунок використання оригінального алгоритму шифрування MTProto.

– Доступність і глобальність - обмежень практично немає.

Аналіз, присвячений визначенню відповідного програмного продукту для спрощення абітурієнтами процесу вибору напряму навчання, показав, що існуючі програмні продукти не повністю вирішують проблему [6].

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

Після проведеного аналізу та обґрунтування необхідності виконання цієї роботи необхідно спроектувати реалізований програмний продукт.

2.1. Концептуальна модель програмного продукту

Для опису роботи програмного забезпечення необхідно побудувати концептуальну модель системи, що розробляється. Така модель повинна бути адекватною предметній області; тому він повинен містити знання всіх учасників процесів системи.

Спочатку виберемо акторів, які будуть взаємодіяти з програмним продуктом:

– адміністратор програми - спеціаліст, який буде керувати всім контентом в системі, а також він матиме повний доступ до всіх функцій;

– користувач - в основному абітурієнти, які вступають до ВНЗ, але в подальшому додатком може користуватися і студент.

Також виділимо основні прецеденти:

Переглядайте новини, контакти, інформацію про вступ до Державного університету телекомунікацій.

Виконавець: користувач. Виробляється в додатку telegram. Опис кроків:

– запустити бота;

– пройти опитування;

– перейти в модуль «Новини», «Контакти», «Як вступити до Державного університету телекомунікацій» - за допомогою кнопок навігації;

– виберіть пункт у меню, що з’явиться.

– за допомогою навігаційних кнопок перейдіть до модуля «Меню», щоб побачити навігаційне меню. Додавання новин, контактів, інформації про вступ до Державного університету телекомунікацій.

Виконавець: адміністратор. Виробляється в адміністративній панелі веб-додатку. Опис кроків:

– авторизуватися в системі;

– зайти в модуль «Новини», «Контакти», «Як вступити до Державного університету телекомунікацій» - за допомогою навігаційного меню;

– перейти до додавання або редагування новин, контактів чи іншої інформації;

– розташувати новини, контакти чи школу за потреби та додайте.

Взаємодія акторів з програмним продуктом та опис варіантів використання графічно представлені на рисунку 2.1 як діаграма варіантів використання.

Діаграма варіантів використання в UML — це діаграма, яка відображає зв’язки між акторами та варіантами використання та є складовою частиною моделі варіантів використання, що дозволяє рівню показати концепцію на концептуальному рівні [7].

Прецедент - можливість змодельованої концепції (частина її функціональності), на основі якої користувач може отримати конкретний, вимірний і бажаний результат. Обставина відповідає індивідуальній службі концепту, визначає єдину альтернативу його використання та являє собою традиційний спосіб взаємодії користувача з концептом. Варіанти використання однаково застосовуються з метою визначення зовнішніх умов концепції.

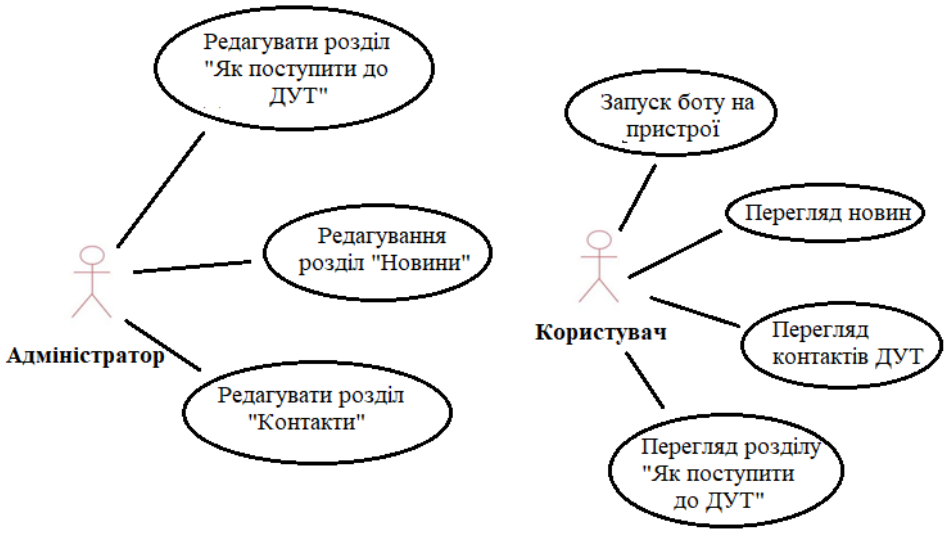


Рисунок 2.1 – Діаграма варіантів використання програмного продукту

Процес відповіді користувача на надсилання запиту можна представити у вигляді діаграми активності (рис. 2.2).

Діаграма діяльності – UML-діаграма, яка показує дії в станах, описані на діаграмі станів. Під діяльністю розуміється класифікація виконуваної дії в послідовній і синхронній скоординованій формі виконання підлеглих компонентів: вкладених дій і одиничних дій англійською мовою, дій, пов'язаних потоками, що проходять від виходів 1-го вузла до входів другого [7]. Діаграми діяльності використовуються для моделювання бізнес-процесів, науково-технічних процесів, паралельних і паралельних концепцій. Діаграми діяльності складаються з невеликої кількості фігур, з’єднаних стрілками.

Ключові форми:

– вигнуті прямокутники – дії;

– ромби – розв’язок;

– великі смуги – початок (гілка) і кінець гілки дії;

– темний діапазон – це початок процесу;

– темна гама зі штрихом вважається завершенням процесу.

Стрілки йдуть від стовбура до кінця процесу та показують потоки утилізації або об’єкта.

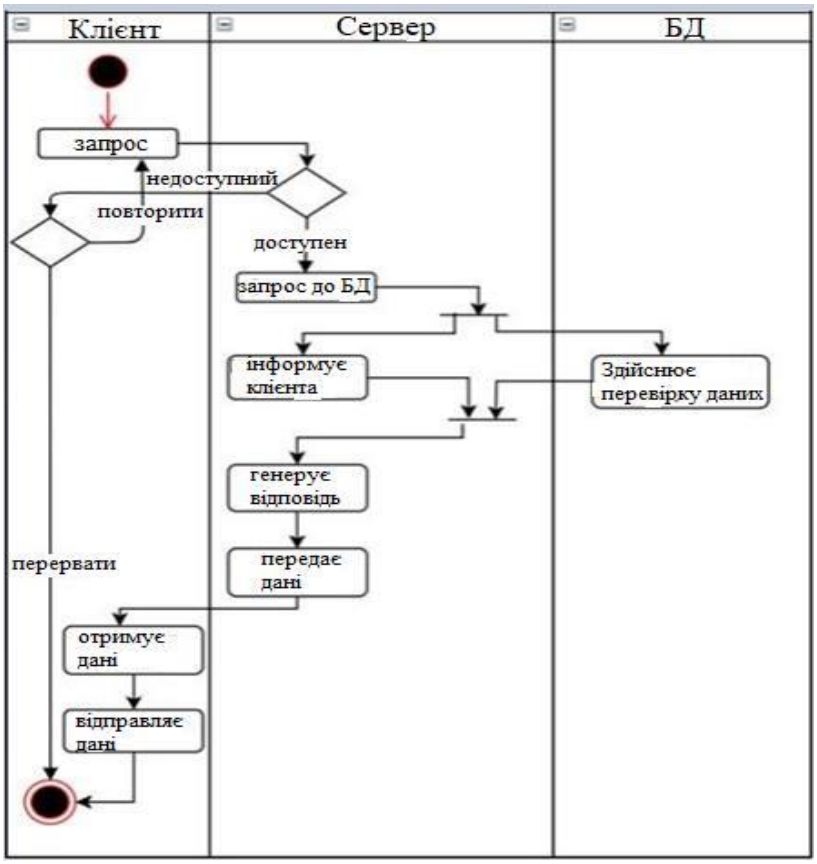


Рисунок 2.2 – Діаграма діяльності процесу «Пройдіть опитування»

2.2. Компонентна архітектура

Компонентна архітектура описує підхід до проектування та розробки систем з використанням методів розробки програмного забезпечення. Основна увага в цьому випадку зосереджена на декомпозиції дизайну на окремі функціональні або логічні компоненти, які забезпечують чітко визначені інтерфейси, які містять методи, події та властивості. У цьому випадку забезпечується більш високий рівень абстракції, ніж при об’єктно-орієнтованій розробці, і увага не зосереджується на таких питаннях, як протоколи зв’язку чи загальний стан [8].

Є два підходи до впровадження ПЗ для абітурієнтів Державного університету телекомунікацій:

– однокомпонентна архітектура;
– багатокомпонентна архітектура.

2.2.1. Однокомпонентна архітектура

В однокомпонентній архітектурі програма встановлюється з магазину програм із уже завантаженим вмістом. Вся логіка представлена в одному додатку, індивідуально для кожної платформи.

Переваги такої архітектури:

– менша трудомісткість розробки;

– незалежність підключення до Інтернету;

– безпека передачі даних.

Недоліки цієї архітектури:

– тривалий час для оновлення контенту;

– не дозволяє використовувати соціальну взаємодію між користувачами;

– труднощі при переході на нову платформу.

2.2.2. Багатокомпонентна архітектура

Багатокомпонентна або клієнт-серверна архітектура — це програма графічного інтерфейсу користувача, яка взаємодіє з сервером бази даних, де зберігається постійно оновлюваний вміст і основна частина бізнес-логіки. Як правило, цей стиль описує відносини між клієнтом і сервером (або серверами), де їхній клієнт виконує послідовність дій, запит очікує відповіді - обробка після отримання. Сервер, у свою чергу, дозволяє користувачеві виконати необхідну для отримання результату обробку - передає результат.

Основними перевагами багатокомпонентної архітектури є:

– Простота підтримки. Рівні незалежні один від одного, що дозволяє вносити оновлення або зміни, не впливаючи на програму в цілому.

– Масштабованість. Рівні організовані на основі розширення шарів, тому ви можете легко масштабувати програму.

– Гнучкість. Кожним рівнем можна керувати та масштабувати незалежно для більшої гнучкості.

– Доступність. Додатки можуть використовувати модульну архітектуру, що дозволяє системі використовувати легко масштабовані компоненти, що збільшується.

Порівняння доступності

Переваги та недоліки розглянутих архітектур, було зроблено висновок, що найбільш підходящою архітектурою для реалізації прикладного посібника є багатокомпонентна архітектура.

В якості архітектурного рішення використовується один з видів багатокомпонентної архітектури - трирівнева архітектура.

Ця архітектура дозволяє інтелектуально розподіляти модулі обробки даних, які передаються на один або декілька окремих серверів. Однією з очевидних переваг цієї архітектури є те, що сервери можуть взаємодіяти один з одним, це дозволить розділити систему на більш детальні функціональні блоки з певними ролями. Ця архітектура графічно представлена на рисунку 2.3.

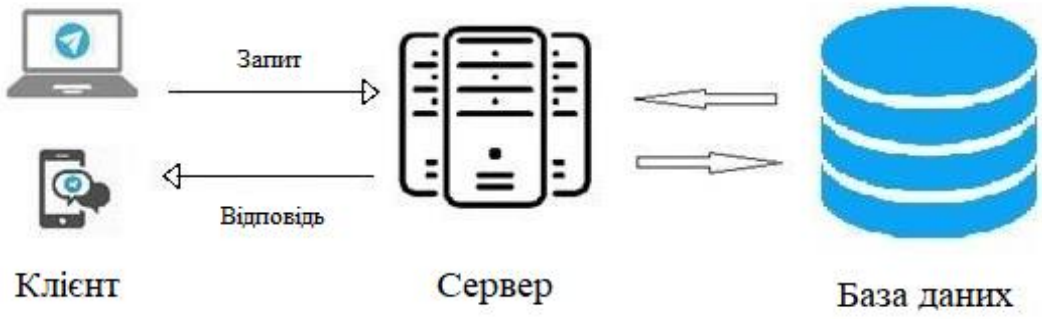


Рисунок 2.3 - Трирівнева компонентна архітектура

Трирівнева компонентна архітектура має три основні рівні:

Рівень клієнта

Це графічний інтерфейс, який забезпечує функції введення та відображення даних. Цей рівень зберігає просту бізнес-логіку та містить стан програми. У розробленій системі на цьому рівні будуть розташовані:

– бізнес-логіка для відправки запиту на сервер, отримання та обробки отриманих даних;

– бізнес-логіка взаємозв’язку робочих компонентів через сервіси та провайдерів;

– взаємодія елементів GUI.

– рівень сервера додатків

Цей рівень є сполучною ланкою між клієнтськими рівнями та базою даних, де зберігається більша частина бізнес-логіки. На цьому рівні розроблена система буде містити:

– бізнес-логіка прийому та обробки запитів від клієнта;

– створення колекцій для бази даних;

– бізнес-логіка взаємодії з базою даних.

Рівень бази даних

Цей рівень зберігатиме та оброблятиме дані та взаємодіятиме виключно з рівнем сервера додатків.

Щоб відобразити зв’язок між логічними та фізичними модулями, потрібно створити діаграму компонентів. Діаграма компонентів UML графічно зображена на рисунку 2.4.

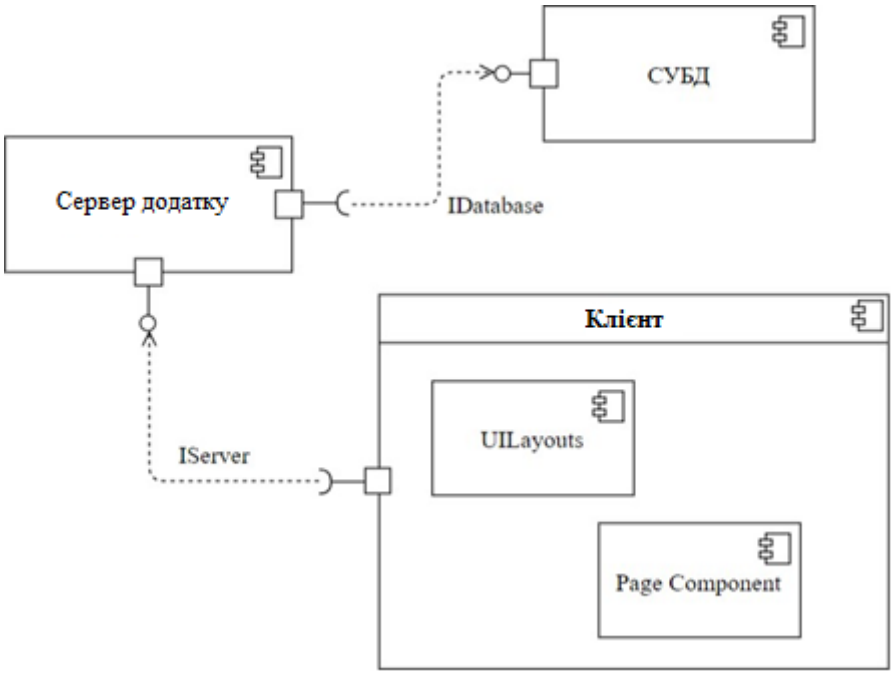


Рисунок 2.4 - Діаграма компонентів UML

Компонент «Сервер додатків» - реалізує бізнес-логіку для отримання та обробки запитів від клієнта, надає інтерфейс IServer і використовує інтерфейс IDatabase.

Компонент «СУБД» - реалізує доступ до даних та забезпечує інтерфейс IDatabase.

Компонент «Клієнт» використовує інтерфейс IServer і складається з двох компонентів:

– UILayouts – складова частина, що містить елементи інтерфейсу користувача;

– Page Component – складова частина, призначена для відображення нової інформації на екрані та пунктів меню [9].

2.3. Визначення загальних вимог до програмного продукту

У цьому розділі будуть визначені вимоги до програмного продукту.

Вимоги до адміністратора програми:

– програмне забезпечення має дозволяти додавати нові розділи до навігаційного меню;

– програмне забезпечення має дозволяти працювати з різними даними;

– усі зміни необхідно вносити через панель адміністратора;

– програмне забезпечення повинен зберігати всі дані в базі даних;

– програмне забезпечення має бути кросплатформним.

Визначимо вимоги до користувача:

– усі зміни мають бути автоматично доступні програмному забезпеченню;

– програмне забезпечення повинно дозволяти переглядати список шкіл;

– програмне забезпечення має дозволяти переглядати новини;

– програмне забезпечення має дозволяти перегляд контактів;

– програмне забезпечення має дозволяти перегляд розділу «Як увійти в Державного університету телекомунікацій?»;

– програмне забезпечення повинно дозволяти відповідати на питання для заповнення персональних даних.

Функціональні вимоги до програмного продукту

Виходячи з вищевикладених вимог, визначимо перелік функцій розробленої програми:

– можливість створювати новини;

– можливість редагувати школи;

– можливість редагувати контакти;

– можливість редагувати розділи в програмному забезпеченні;

– можливість редагувати користувачів;

– можливість видалення користувачів;

– перевірка правильності введеної інформації.

2.4. Архітектура програмного забезпечення

Обрана трирівнева архітектура буде розроблена з використанням сервіс-орієнтованого архітектурного стилю. Ця архітектура дозволяє створювати розподілене програмне забезпечення, що складається з набору незалежних сервісів.

Трирівнева архітектура буде розроблена з використанням сервіс-орієнтованого архітектурного стилю. Сервісно-орієнтована архітектура дозволяє створювати розподілене програмне забезпечення, що складається з набору незалежних сервісів [10].

Ця архітектура має ряд переваг, і одна з них - здатність швидко адаптуватися до мінливих вимог і викликів інформаційного світу.

На даний момент найбільш затребуваними є веб-сервіси SOAP і REST. Основні відмінності між підходами SOAP і REST [10]:

Пакети запитів і відповідей REST набагато менші за відповідні пакети SOAP.

SOAP розглядає рівень даних HTTP як пасивного спостерігача, а REST як активну взаємодію з використанням існуючих методів HTTP, таких як GET, POST, PUT, DELETE.

Модель SOAP підтримує певний ступінь самоаналізу, дозволяючи розробникам послуг описувати свій API у файлі WSDL (мова опису мови). У свою чергу, модель REST уникає складнощів WSDL і використовує чистіший інтерфейс на основі стандартних методів HTTP.

REST обмежується операціями CRUD з відповідними методами HTTP. SOAP може містити майже необмежену кількість методів. Порівнюючи переваги та недоліки двох підходів до реалізації веб-сервісів, можна побачити, що веб-сервіси REST є найбільш зрозумілими та простими у реалізації, мають найменші розміри пакетів.

RESTful API також можна кешувати. Це означає, що клієнт має можливість зберігати всі відповіді в кеші. В результаті API покращив продуктивність.

API RESTful зазвичай називають веб-службами RESTful, оскільки вони реалізують принципи REST, а також протоколи HTTP.

По суті, він охоплює багато ресурсів за допомогою методів HTTP, а потім представлений у стандартному форматі, яким зазвичай є XML. Поки гіпертекст є

стандартним, він працює для дійсних типів Інтернет-медіа. Деякі приклади показані на рисунку 2.5.

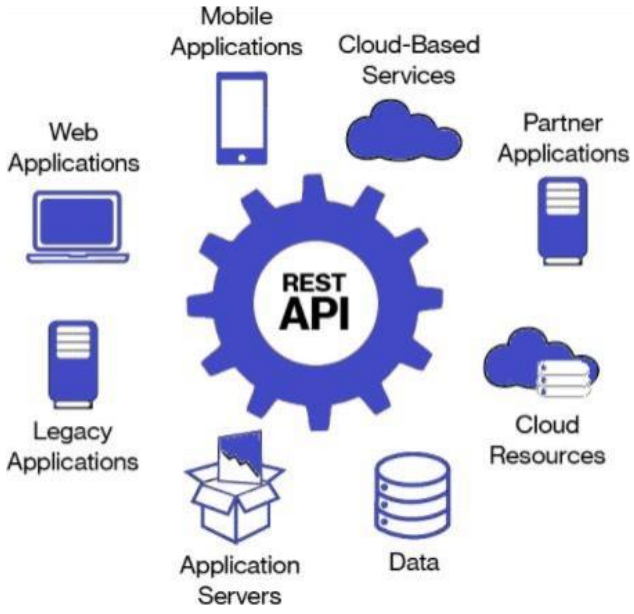


Рисунок 2.5 – Візуальне представлення REST

REST використовує 4 основні методи HTTP: GET, POST, PUT, DELETE. У більшості випадків кожен із методів використовується для виконання попередньо визначеної дії з CRUD (Створення, Читання, Оновлення, Видалення).

POST - створити, GET - прочитати, PUT - оновити, DELETE - видалити. Приклад реалізації веб-сервісу REST графічно показано на рисунку 2.6.

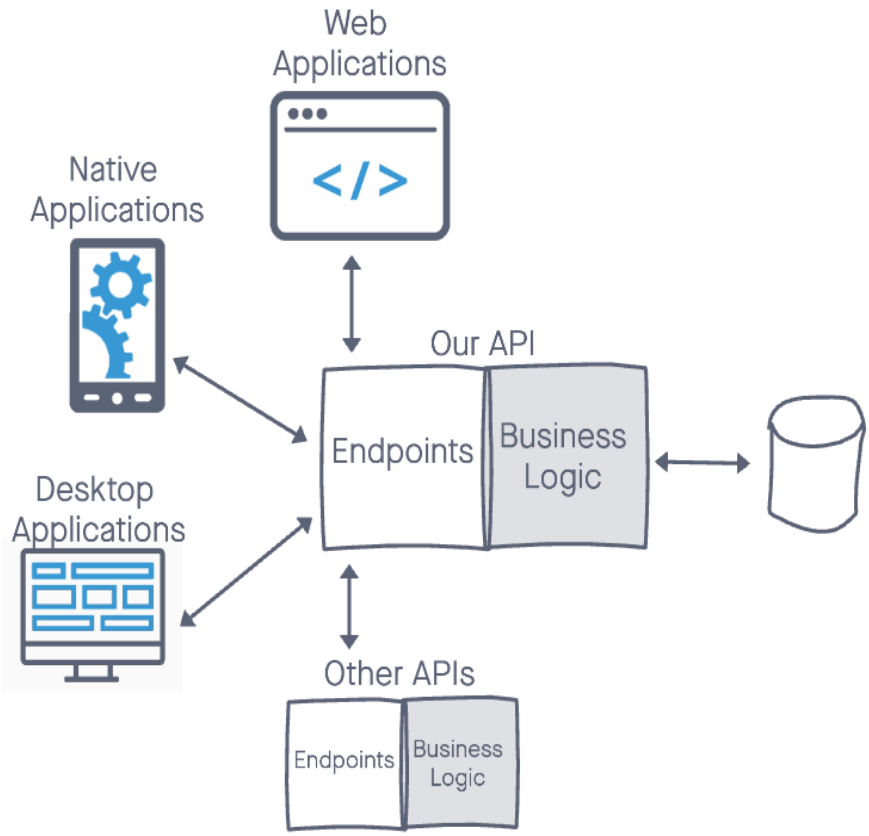


Рисунок 2.6 - Веб-сервіс REST

На рисунку 2.6 показана початкова форма, REST API, створена за допомогою Python для взаємодії з базою даних MySQL.

Цей програмний продукт складається з двох частин. Це зручний інтерфейс, який надає користувачам новини, шкільну інформацію тощо. Система керування контентом Telegram-бот також має інтерфейс адміністратора, до якого адміністратори входять для оновлення вмісту та керування програмним забезпеченням.

У цьому випадку архітектура двох додатків буде розглядатися окремо:

– орієнтований на користувача інтерфейс,
– інтерактивний непублічний адміністративний інтерфейс

2.5. Інтерфейс адміністративної панелі на Python

Структура веб-сайту представлена на зображенні нижче (рис. 2.7).

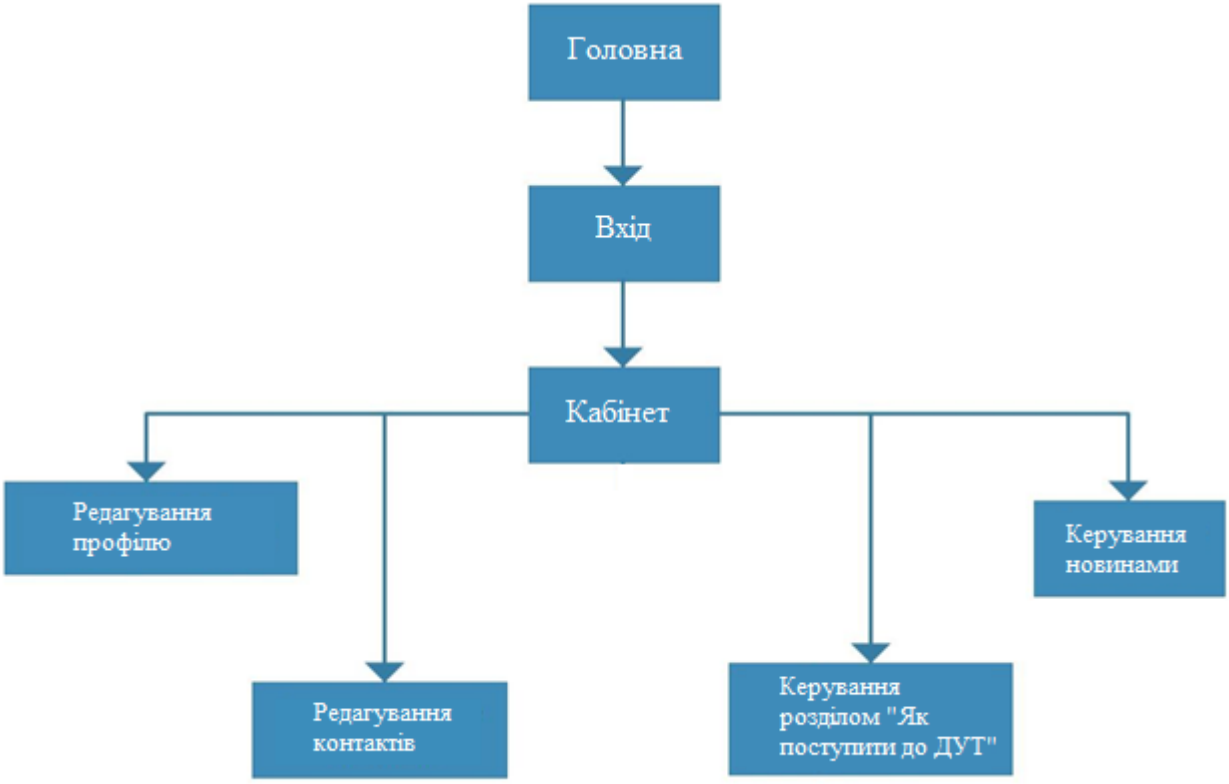


Рисунок 2.7 – Структура адміністративної панелі

Для керування контентом відображений користувачеві було розроблено телеграм-бот на Python і основна структура програми можна бачити на рисунку 2.7.

2.6. Інтерфейс телеграм бота

Основні елементи інтерфейсу чат-ботів універсальні та властиві кожному додатку обміну повідомленнями. Як зазначено в документації TelegramAPI, Telegrambot може спілкуватися з серверами двома способами:

1. getUpdates - pull: бот постійно звертається до сервера Telegram і перевіряє, чи є нові повідомлення;

2. setWebhook - push: у міру надходження нових повідомлень сервер Telegram відправляє їх боту.

На рисунку 2.8 показано різницю між цими двома методами.

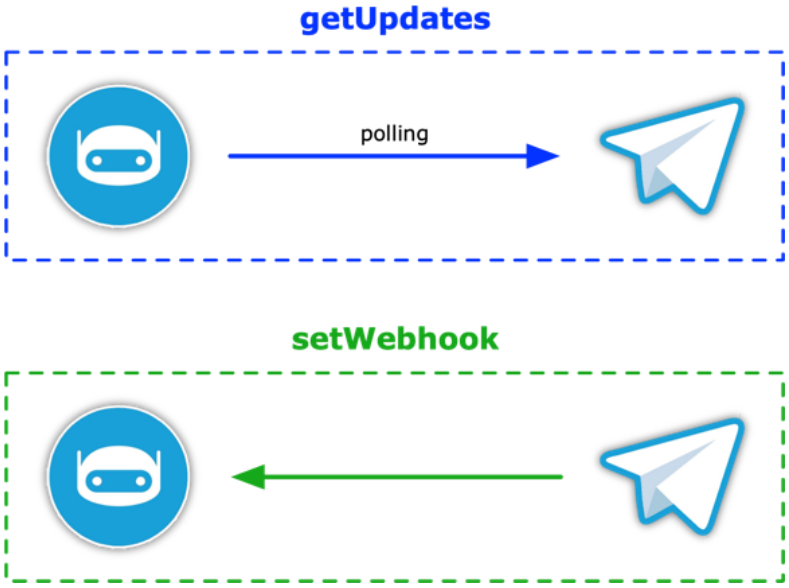


Рисунок 2.8 - Способи взаємодії з ботом

Очевидно, що другий спосіб (setWebhook) більш раціональний для всіх учасників процесу. Однак у цьому є неявна складність: хтось повинен отримувати повідомлення від Telegram на стороні бота, тобто потрібен веб-сервер або еквівалент. Сертифікат SSL також потрібен для зв'язку через безпечний канал HTTPS. Це необхідна та важлива частина взаємодії вебхука.

Програмне забезпечення включає веб-сервер і Telegrambot. Основні ідеї, які були реалізовані в додатку, представлені нижче:

1. Обробка повідомлень користувача в режимі реального часу;
2. Миттєве представлення можливих відповідей користувачеві;
3. Обробка рішення користувача в реальному часі та пересилання відповіді;

Веб-сервер, у свою чергу, складається з серверного програмного забезпечення бота Telegram. Тобто спілкування з базою даних, виконання команд, які запитує користувач.

2.7. Гібридна архітектура програмного продукту

В результаті було отримано два окремо працюючих додатки з використанням REST API. Цілісність архітектури з одним REST API, що взаємодіє з двома клієнтськими програмами, показано на рисунку 2.9.

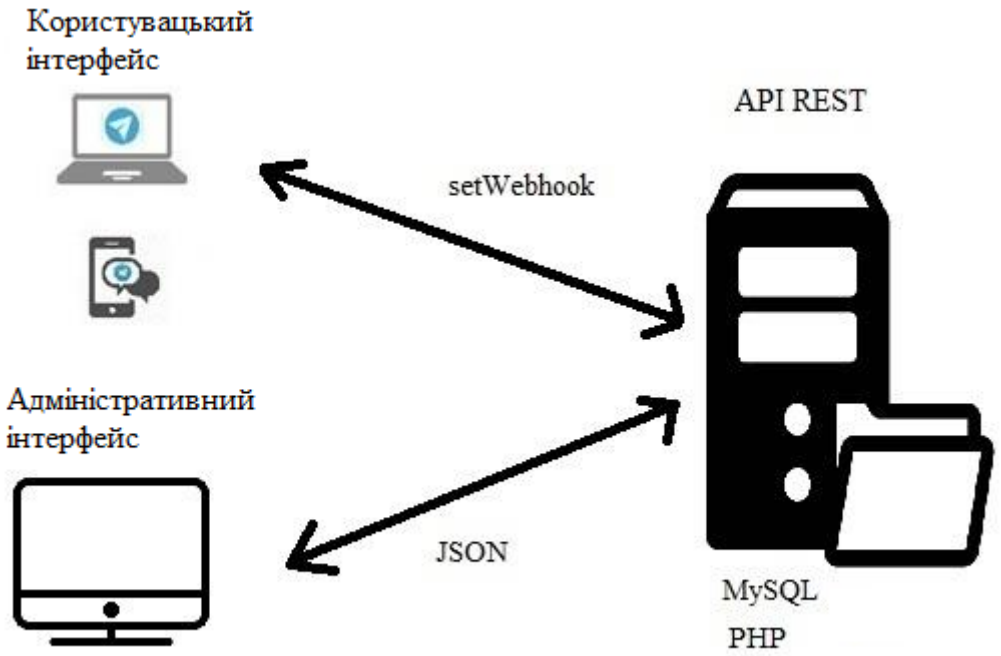


Рисунок 2.9 – Гібридна архітектура набору прикладних програм MEAN

Як показано на рисунку 2.9, щоб забезпечити найбільш відповідне рішення, один REST API надає дані двом додаткам, орієнтованим на взаємодію з користувачем, на основі різних частин технологічного набору MEAN.

2.8. Захист даних, що передаються

Щоб виключити можливість перехоплення даних сторонніми додатками, при передачі даних від сервера до клієнта за протоколами HTTP необхідно забезпечити захист даних, що передаються.

Для захисту даних, що передаються, організовані запити HTTPS, а також ідентифікація та авторизація клієнта на сервері. Для цього при виконанні запиту

передається токен. Веб-токен JSON складається з трьох частин, розділених крапками:

– HEADER
– PAYLOAD
– SIGNATURE

Вихід складається з трьох розділених крапками рядків Base64-URL, які можна легко передати в середовищах HTML і HTTP, будучи більш компактними, ніж стандарти на основі XML, такі як SAML.

Рисунок 2.10 ілюструє автентифікацію клієнт-сервер за допомогою діаграми.

3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Вибрані технології розробки

3.1.1. Аналіз платформи для розробки

Є багато мов програмування, спеціалізованих на користь виконання різних завдань. Кожен з них характеризується унікальним набором операторів і певним синтаксисом. Розглянемо 3 більш відомі мови, що використовуються у веб-розробці: PHP, Ruby та Python. Проаналізувавши плюси і мінуси виявимо, наскільки вони видаються, чим виділяються між собою, якою мірою серед тієї чи іншої аудиторії вони потрібні. PHP – мова програмування, що виконується на стороні сервера, сконструйований Расмусом Лердорфом (Rasmus Lerdorf) як інструментарій для формування динамічних та інтерактивних інтернет-сайтів.

Ця мова показала себе досить гнучкою і потужною, тому набула значної популярності і застосовується в проектах різного масштабу: зі звичайного блогу аж до найбільших веб-додатків в Мережі інтернет. Ruby – динамічна об'єктно-орієнтована мова програмування, створена Юкіхіро Матсумото. Ruby був заснований під впливом аналогічних мов, як Perl, Eiffel і Smalltalk. Мова Ruby застосовується у веб-розробці у складі відкритого веб-фреймворку Rails, що більше називається Ruby on Rails (RoR). Python широко використовується як інтерпретована мова на користь скриптів різноманітного напряму (незважаючи на те, що є і транслятори мови Python).

Концепція формування цієї мови зародилася на завершення 1980-х та розробленого Гвідо ван Россумом. Python – мультипарадигмальна мова програмування: дає можливість поєднувати процедурний стиль написання програмного коду з об'єктно-орієнтованим та функціональним.

Для чіткого уявлення про серверні мови програмування PHP, Python або Ruby важливо розуміти їхні переваги та недоліки. У кожної мови є плюси та мінуси. Все залежить від того, які потреби та очікування пов'язує з цими мовами. Можна

вибрати мову, що найбільше відповідає потребам та потребам діяльності розробника. Далі наведено плюси та мінуси кожної з трьох мов програмування.

Таблиця 3.1 PHP, Python та Ruby: переваги

PHP	**Ruby**	**Python**
Безкоштовне програмне забезпечення за ліцензією PHP	Кросплатформенність та відкритий вихідний код	Легкий та швидкий у вивченні
Легкий у освоєнні (висока швидкість навчання)	Може бути вбудований у мову розмітки гіпертексту	Підтримується безліччю платформ та операційних систем
Велика спільнота користувачів та розробників	Мова програмування надвисокого рівня (VHLL)	Читабельний та організований синтаксис
Розширена підтримка баз даних	Простий і зрозумілий синтаксис, що дозволяє розробнику-початківцю дуже швидко вивчити мову	Забезпечення швидкого прототипування та динамічних семантичних властивостей
Надає велику кількість доступних розширень та вихідних кодів	Просте підключення до баз даних DB2, MySQL, Oracle та Sybase	Величезна спільнота підтримки
Дозволяє виконання коду в обмежених середовищах виконання	Створені на Ruby великі масштабовані програми прості у супроводі	Проста побудова додатків шляхом тестування та імпорту необхідних функцій

Надається можливість управління нативними сесіями та розширення API	Наявність вбудованого відладчика та гнучкого синтаксису	Реюзабіліті (можливість повторного використання) за рахунок ретельної розробки пакетів та модулів
Хороша альтернатива конкурентам, таким як ASP (Active Server Pages) від Microsoft	Можливість написання зовнішніх бібліотек на Ruby або C	Об'єктно-орієнтований підхід до програмування

Таблиця 3.2 PHP, Python та Ruby: недоліки

PHP	**Ruby**	**Python**
Не підходить для розробки настільних програм	Можливі труднощі у вивченні	Недостатньо ефективна робота з багатоядерними та багатопроцесорними обчислювальними системами
Традиційно мізерний функціонал для обробки помилок	Нестача інформаційних ресурсів	Обмежений рівень доступу до баз даних
Глобальні параметри конфігурації можуть змінювати семантику мови, ускладнюючи процеси впровадження та сумісності	Великі витрати процесорного часу (CPU time) у порівнянні з іншими мовами	Відсутність комерційної підтримки навіть для Open Source проектів (проте ця ситуація починає змінюватися)

Звернення до об'єктів за замовчуванням здійснюється методом «виклику за значенням» (CallByValue), що суперечить аналогічним операціям для більшості мов і застає багатьох програмістів зненацька	Порівняно повільна розробка оновлень	Невелика кількість розробників Python у порівнянні з іншими мовами, наприклад Java
Загалом вважається менш захищеним у порівнянні з іншими мовами програмування		За відгуками повільніший у порівнянні з мовами типу Java

Проаналізувавши всі недоліки та переваги вибір, зупиняється мовою програмування Python.

– PHP – найкраща мова для того, щоб формувати динамічні веб-сторінок та веб-додатків, з великою кількістю фреймворків та бібліотек, які щороку оновлюються і отже не становить ніякої труднощі та сприятливіше для роботи розробнику;

– Python – різностороння мова програмування, завдяки якій є можливість виконувати всілякі програми у спектрі з інтернет-сайтів та десктопних додатків аж до роботів та системних сервісів;

– Ruby – більш високорівнева мова, що надає витрачати меншою мірою уваги складовим частинам інтерфейсу та організації збереження даних, щоб сконцентруватися на практичнішому завданні.

3.1.2. Аналіз СУБД

Система управління базами даних стала невід'ємною частиною розробки динамічних веб-продуктів. З її допомогою можна систематизувати весь масив потрібних файлів. Все це потрібно для швидкого доступу та оптимізації роботи програми або сайту. Для аналізу було обрано такі популярні СУБД як: MariaDB [13], MySQL та PostgreSQL.

СУБД – це таблиця, і інший структури може бути в даної системи. Зате дані в таблиці можуть бути різного типу. Деякі СУБД підтримують не так багато типів, деякі запроваджують навіть нові, для інформаційних технологій. Це і булеви, і рядкові, і дані з точкою, що плаває, і багато інших. І всі ці дані пов'язані між собою згідно з реляційною моделлю.

Проте, щоб прописати дані у рядок, їй потрібно задати тип даних. Дуже схожу процедуру можна робити з осередками в Excel та подібних програмах. Коли йдеться про веб-сайти та додатки, то в осередках виявляються дані про користувачів ресурсу, сам вміст, продукція і будь-що. Мета бази даних тут у тому, щоб оперативно надавати цю інформацію на запит користувача: серверний скрипт, в обмін на клієнтський.

Переваги та недоліки СУБД

Якщо провести порівняння файлової системи та побудованому на ній сайті, то можна спостерігати, наскільки більш плавно та ефективно працює система управління базами даних. Серед недоліків, що часто обговорюються, сучасних систем управління базами даних:

– нелегко освоїти. Розуміти принципи роботи MySQL означає розбиратися в базах даних.

– вартість. Сама система управління базами даних може коштувати дуже багато, але обслуговування бази даних, придбання стороннього ПЗ та інші. Навіть розмістити велику базу даних хорошому хостингу коштує певних витрат.

– вага. Одні тільки файли для функціональної програми будуть важити багато. А якщо обернути їх у базу даних, обсяг суттєво зросте, адже тепер файли мають деякі функції, перебувають у логічному зв'язку та викликаються скриптами. Все це коштує не лише грошей, а й пам'яті на сервері. Це особливо відчутно, якщо сервер служить персональний комп'ютер розробника.

– централізоване розміщення. Тільки в останні роки розробники почали використовувати розподілений реєстр для зберігання файлів. Коли файли знаходяться в межах однієї бази даних, вони вразливі.

Добре, що більшість цих недоліків перекриваються перевагами СУБД. Система відкриває безліч можливостей, які недоступні для жодної файлової системи.

По-перше, це економія пам'яті. Хоча і сама СУБД займає певне місце, вона не дає розмножуватися зайвої інформації. Ніякого надлишку, жодних файлів-дублів. У той самий час, та інформація, яка має зберігатися надміру, зберігається саме в такий спосіб.

Важливо, що СУБД дозволяє уникнути подвійних істин. Система самостійно відбирає елементи, які дають інформацію одного роду та стежить за тим, щоб вони не суперечили один одному. Виходить, що при тому ж обсязі продукт отримує значно більший обсяг інформації.

Велику роль грають системи управління базами даних під час спільної розробки. Забезпечити груповий доступ до дерева файлів досить важко, дотримуючись усіх запобіжних заходів. Але можна забезпечити санкціонований доступ до СУБД обмеженому колу осіб, без втрат для системи безпеки.

Сьогодні важко реалізувати зберігання даних краще, ніж із сучасною СУБД. Використання АБД дозволяє визначити необхідні заходи безпеки. До того ж нові інструменти для захисту бази даних виходять щодня. Доступ, як правило, здійснюється через форму запилення, але при достатніх навичках, є можливість реалізувати все: від антропометрії до двофакторної аутентифікації. Особливо це можна застосувати до open-source СУБД, якою є MariaDB.

MySQL – це, однозначно, найпопулярніша з усіх існуючих СУБД. На ній будують не лише веб-додатки та складне програмне забезпечення. Функціональність цієї системи змушує конкурентів вигадувати нові і нові рішення. Але й самі розробники не відстають: остання версія програмного забезпечення вийшла зовсім недавно. Свою періодичність вони не переривають уже протягом більш як двадцяти років.

Раніше цією розробкою володіла компанія Sun Microsystems, яка випустила Java та багато інших інструментів розробки. У 2010 році всі продукти, разом з MySQL, перейшли компанії Oracle. Вона здійснює підтримку СУБД і сьогодні.

Спочатку ця система була розроблена однойменною компанією у 1995 році. Творці використовували найшвидші мови програмування: C, C++ і HTML. Таким чином, розробники отримали у розпорядження стабільну та швидку СУБД із постійною підтримкою. Сьогодні MySQL входить до складу так званих «джентельменських наборів», які складаються із сервера, бази даних та скриптової мови програмування.

Однозначною перевагою MySQL перед конкурентами можна назвати використання. Як завжди, чим популярніше ПЗ, тим легше з ним працювати. Всі баги виявляються швидко, так само швидко виправляються. Не варто забувати і про те, що це програмний продукт для програмістів і розробників, який розвивається швидко завдяки спільноті. Постійно з'являються нові плагіни та різні розширення для MySQL.

Встановлювати MySQL дуже просто. Завдяки наявності GUI - графічного інтерфейсу користувача, це перетворюється на звичайну установку ПЗ. Те саме стосується й інсталяції доповнень до СУБД.

Не можна не згадати про те, що MySQL - одна з найбільш кросплатформових СУБД. Якщо згадувати про масштабованість: майже всі найбільші ресурси, з якими розробник працює в мережі, побудовані на основі MySQL. Хоча існують і професійніші варіанти.

Як і у всіх open-source проектів, у MySQL трапилося вдале відгалуження, яке отримало назву MariaDB. Метою розробників «Марії» було створити продукт,

повністю сумісний із MySQL, але значно покращений. Наприклад, двигуном для зберігання даних у

MySQL була MyISAM. У Марії – це Aria, яка подарувала СУБД велику продуктивність у порівнянні з основним проектом. І хоча MariaDB заснована на MySQL, останні версії містять не більше ніж 25% оригінального коду.

Марія може похвалитися вищою продуктивністю загалом. Особливо це стосується перекодування символів. На високих обсягах інформації коефіцієнт досягає більш як 2%. Налагоджувальний код також оптимізований, порівняно з MySQL. Загалом розробники відзначають вищу швидкість розробки, ніж міг видати «батько». Спільнота, яка працює над MariaDB обіцяє ще більші покращення.

Крім того, сам користувач може покращувати та оптимізувати роботу Марії. Що відрізняє цю СУБД від решти, так це повноцінний open-source: ніяких закритих елементів або модулів, все в доступі. Правити під свій продукт код можна необмежено, як і робити пропозиції щодо зміни спільноти, яка розробляє MariaDB.

PostgreSQL – це ще одна система управління базами даних, тільки вже не реляційна, об'єктно-реляційна. Це означає, що користувач може створювати об'єкти для операцій, куди можуть входити різні дані. Вона повністю безкоштовна і найбільш гнучка. Деякі розробники називають PostgreSQL найпрофесійнішим рішенням, з усіх, що існують на ринку. Ця СУБД з'явилася з некомерційного університетського проекту, створеного у Берклі, що називалася Postgres. Ця система розроблялася довгі вісім років і підтримується до цього дня.

Як буває з такими продуктами, вона вийшла "від програмістів для програмістів" - неймовірно функціональною, але надто складною в освоєнні для звичайного розробника. Спочатку СУБД навіть мала власну мову запитів, але згодом від цієї ідеї відмовилися, залишивши тривіальний «сіквел». Незважаючи на ентузіазм незалежних розробників,

PostgreSQL не така гарна, якою її люблять називати. До цього часу у вихідному коді знаходять проблемні місця.

За масштабованістю PostgreSQL не поступається, якщо порівнювати з MySQL та MariaDB. На основі цього програмного забезпечення будуються масивні проекти з обробки Big Data, оскільки її стабільності довіряють розробники. Декілька варіантів інтерфейсу роблять продукт доступним для персоналізації.

Але до масового продукту PostgreSQL ще далеко. Справа в тому, що ця система дуже складна для простого розробника. Навіть якщо взяти до рук документацію, не завжди можна отримати відповіді на всі запитання, включаючи найлогічніші. Також, бентежить швидкість виконання запиту в PostgreSQL.

Ця СУБД чудово підходить для корпоративних рішень. Наприклад, база даних для IT-компанії, де її підтримкою може зайнятися кожен із розробників. Тим більше, що PostgreSQL повністю безкоштовна.

Проаналізувавши ці СУБД приходимо до висновку, що за використанням – це MySQL, за розширюваністю – MariaDB та PostgreSQL. Під час обговорення з науковим керівником було вирішено використати СУБД MySQL.

3.1.3. Фреймворки

У створенні програми задіяні два фреймворки – Yii та Bootstrap. Yii – це продуктивний компонентний PHP фреймворк, спеціалізований на користь активної розробки нинішніх веб-додатків. Внаслідок його компонентної структури і чудової підтримки кешування, фреймворк особливо підійде для розробки таких серйозних проектів так само як портали, форуми, CMS, торгові майданчики і.т.д. Для організації програмного коду Yii застосовує архітектурний патерн MVC (Model-View-Controller). Yii керується філософією легкого і вишуканого програмного коду ніяк не намагаючись ускладнювати проектування тільки дотримуючись тих чи інших стандартів проектування. Yii чудово розширюємо. Є можливість налаштувати чи змінити майже будь-яку частину корінного програмного коду [11]. Використовуючи архітектуру розширень легко обмінюватися кодом або використовувати код спільноти. Підтримується та розвивається високо кваліфікованою командою та величезною спільнотою розробників. Розробники фреймворку дивляться за тенденціями веб розробки та формуванням інших

проектів. Найбільш оптимальні можливості та найкращі практики постійно вводяться у фреймворк у різновиді звичайних та стильних інтерфейсів. Bootstrap – найпопулярніший фреймворк для розробки адаптивних та мобільних web-проектів. Bootstrap використовує найсучасніші технології CSS та HTML. Даний фреймворк дозволяє використовувати вже готові класи 25 стилів для створення приємного на вигляд і відповідного під різні дозволи екрану сайту.

3.1.4. Телеграм Bot API

Було проведено дослідження бібліотек, з метою підбору придатної для застосування у розробці. Через дослідження бібліотек для Телеграм було прийнято рішення використовувати бібліотеку Telegram Bot API. Перевага була аргументована тим, що, вона ініціативніше підтримується розробниками і має велику кількість користувачів, що дає можливість визначити рішення в виникаючі завдання. Бібліотека Telegram Bot API набуває в собі всі без винятку аспекти відправлення та отримання запитів, дозволяючи зосередитися безпосередньо в логікі. Інтеграція бібліотеки є максимально елементарною, для цього достатньо за допомогою консолі виконати команду: php composer.phar require telegram-bot/api Bot API є HTTP-інтерфейсом для роботи з ботами в Telegram. Кожен бот – це спеціальний обліковий запис, створений для автоматичного оброблення та відправлення повідомлень. Існує два протилежні за логікою способи отримання оновлень від бота:

– long pulling – програма автоматично опитує сервера Телеграмна наявність будь-яких оновлень для бота. За замовчуванням 100мс;

– webhook - сервера Телеграм самі сповіщають додаток на сервері як тільки з'являться будь-які оновлення [12].

Вхідні оновлення зберігатимуться на сервері, доки їх не оброблять, але не довше 24 годин. Незалежно від способу отримання оновлень, у відповідь надсилається об'єкт Update, серіалізований у JSON.

Всі запити до Телеграм Bot API повинні здійснюватися через HTTPS у наступному вигляді: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ. Принцип роботи взаємодії чат-бота та користувача зображено на рисунку 3.1.

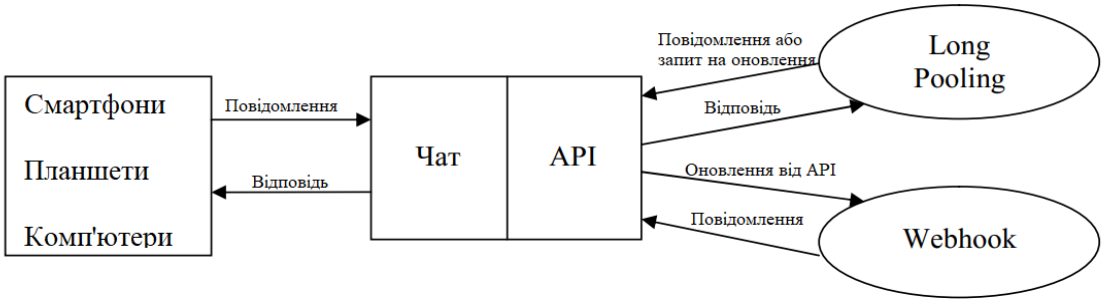


Рисунок 3.1 – Принцип роботи чат-бота на платформі Telegram

Щоб отримати token необхідно написати спеціальному боту @BotFather. Приклади доступних для API методів описані нижче:

– getUpdates – цей метод використовується для отримання оновлень за технологією long polling;

– setWebhook – метод прив'язує до бота URL домену, де міститься запущений бот;

– sendMessage – метод надсилає текстове повідомлення клієнт Telegram;

– sendLocation – метод відправляє повідомлення з координатами до клієнта Telegram;

– getFile – метод повертає завантажений файл на ім'я та ін.

Допускаються POST і GET запити. Для передачі параметрів у Bot API є 4 способи:

– Запит в URL

– application/x-www-form-urlencoded

– application/json (не прийнятний для завантаження файлів)

– multipart/form-data (для завантаження файлів)

3.1.5. Вибір хостингу

Open Server — портативний локальний WAMP/WNMP-сервер з багатофункціональною програмою керування та великим вибором плагінів. Представлений програмний пакет - це не чергова аматорська збірка, зібрана «на коліні», це перший повноцінний професійний інструмент, створений спеціально для веб-розробників з урахуванням їх рекомендацій і побажань (рисунок 3.2).

Рисунок 3.2 – Open Server

Можливості

Open Server - це повністю і повністю портативний сервер. Відсутність системних служб, купи сміття в реєстрі і system32. Ви можете носити його всюди з собою на флешці (бажано на швидкісній), запускати на робочій / домашній машині, не боячись, що у вас щось не вийде.

Якщо на комп'ютері немає необхідних системних компонентів, Open Server встановить їх самостійно, просто виберіть у меню [Інструменти — Перший запуск], якщо сервер запускається на комп'ютері вперше.

За допомогою Open Server ви можете запустити / зупинити сервер або відкрити потрібний домен. Можливості Open Server:

– детальний перегляд журналів всіх компонентів в реальному часі;
– вибір модулів HTTP, СУБД і PHP у будь-якій комбінації;
– підтримка SSL і кириличних доменів з коробки;
– підтримка псевдонімів або інших доменних покажчиків, а також зручна форма для їх налаштування;

– створення локального піддомену без втрати видимості основного домену в Інтернеті;

– доступ до доменів (в один клік) і швидкий доступ до шаблонів конфігурації модулів;

– багатомовний інтерфейс;

Програма постійно вдосконалюється, всі адекватні запити користувачів Open Server детально вивчаються і більшість з них реалізуються.

Для роботи з таблицями даних використовується phpMyAdmin.

phpMyAdmin – це безкоштовний програмний інструмент, призначений для адміністрування СУБД MySQL через браузер (рисунок 3.3). phpMyAdmin підтримує широкий спектр операцій на MySQL та MariaDB. phpMyAdmin дозволяє управляти базами даних, таблицями, відносинами, індексами, користувачами, дозволами тощо [15].

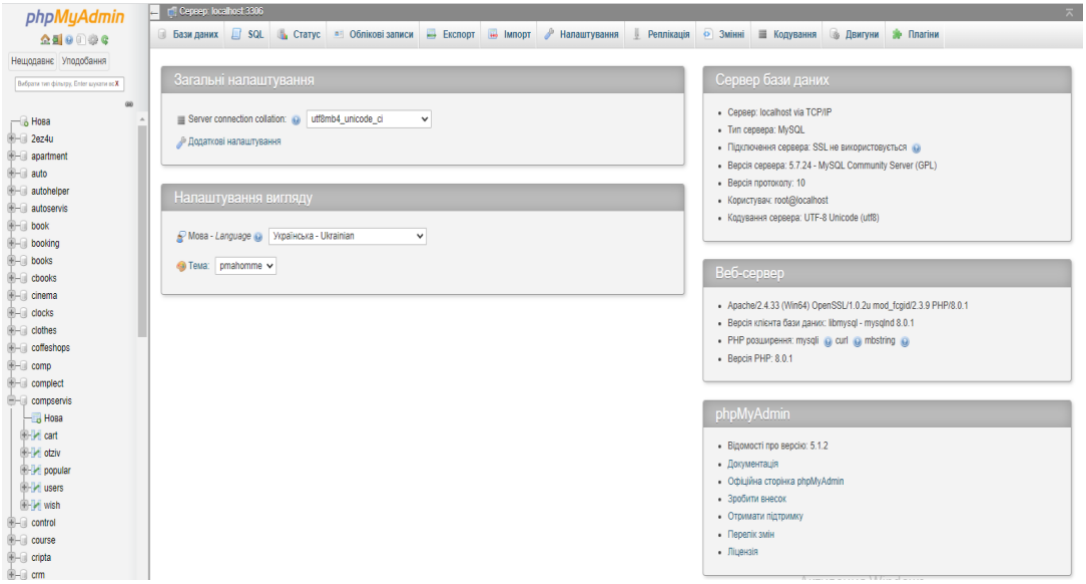


Рисунок 3.3 – phpMyAdmin

3.1.6. Середовище розробки

Для реалізації платформи для розробки Telegram-бота для забезпечення інформаційної підтримки розпізнавання тексту було обрано середовище розробки PyCharm.

PyCharm – це найстабільніша інтелектуальна Python IDE з повним набором засобів для ефективної розробки застосунків на мові програмування Python. Випускається в двох варіантах – безкоштовна версія PyCharm Community Edition, та розширена версія додатку за підпискою, що підтримує більший набір можливостей PyCharm Professional Edition (рис. 3.4). PyCharm виконує інспекцію коду на льоту, автодоповнення, в тому числі ґрунтуючись на інформації, що була отримана під час виконання коду, навігацію по коду, забезпечує безліч рефакторингів, тощо [17].

PyCharm також являє собою інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний відладчик, інструмент для запуску юніт-тестів та підтримує веб-розробку на Django. PyCharm розроблена компанією JetBrains на основі IntelliJ IDEA. Більше 10 років розробки платформи IntelliJ надає PyCharm більше 50 найрізноманітніших плагінів, включаючи підтримку додаткових VCS, інтеграції з різними інструментами та фреймворками, редактором оновлень, таким як емуляція Vim.

Рисунок 3.4 — PyCharm

Ключові можливості середовища розробки:

– потужний та функціональний редактор коду з підсвічуванням синтаксису, авто-форматуванням та авто-відступами для підтримуваних мов;

– проста й потужна навігація в коді;

– допомога при написанні коду, що включає в себе автодоповнення, авто-імпорт, шаблони коду, перевірка на сумісність версії інтерпретатора мови, та багато іншого;

– швидкий перегляд документації для будь-якого елемента прямо у вікні редактора, перегляд зовнішньої документації через браузер, підтримка docstring – генерація, підсвічування, автодоповнення та багато іншого;

– велика кількість інспекцій коду;

– потужний рефакторинг коду, який надає широкі можливості щодо виконання швидких глобальних змін у проекті [17].

3.2. Серверна сторона програмного продукту

3.2.1. Створення необхідних таблиць у MySQL

У процесі розробки серверної сторони програмного продукту використовувалася мова PHP. Як платформу визначальну структуру програмної системи використовувався фреймворк Yii2. Як сховища даних послужила об'єктно-орієнтована база даних (БД) MySQL.

Для створення таблиць необхідно встановити MySQL. Уся інформація про встановлення MySQL на Windows була взята з офіційної документації щодо встановлення.

Після того, як MySQL був встановлений, можна приступити до створення таблиць для подальшого зберігання даних (рис 3.5).

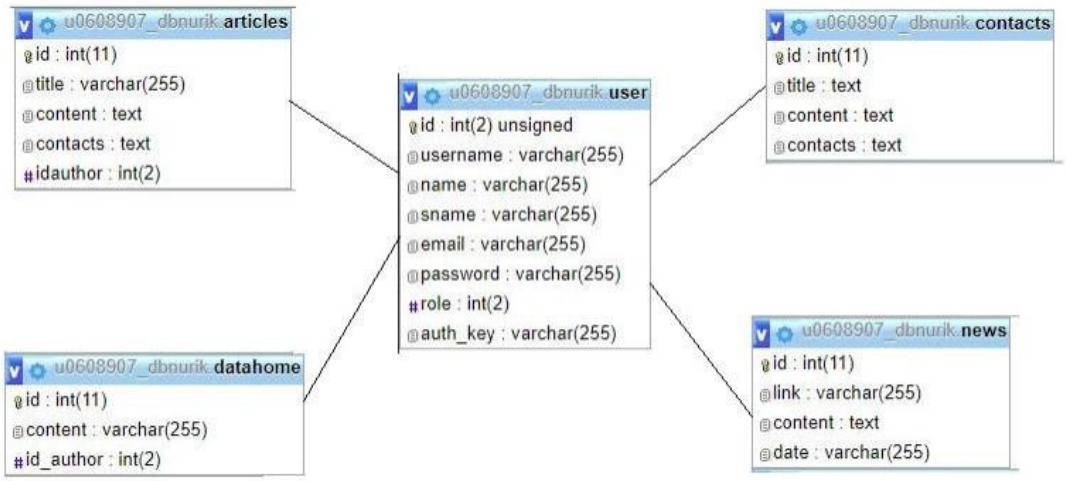


Рисунок 3.5 – Модель бази даних

3.2.2. Реєстрація чат-бота Телеграм

Першим кроком розробки програми є реєстрація у спеціального чат-бота BotFather. Реєстрація починається з команди "/newbot", після чого пропонується ввести назву чат-бота з обов'язковою умовою: в кінці назва має бути вказана "Bot" або "_bot". Якщо всі умови були задоволені, то BotFather видає токен (спеціальний набір символів для доступу до HTTP API ТелеграмBot) та URL-адресу для доступу з чат-боту. Приклад реєстрації чат-бота представлений рисунку 3.6.

Рисунок 3.6 – Приклад реєстрації чат-бота

Для встановлення додаткових параметрів, таких як іконка чат-бота, вітальне повідомлення, опис чат-бота, а також видалення наявних чат-ботів, існують наступні команди (таблиця 3.3).

Таблиця 3.3 – Доступні команди для зміни чат-ботів

Назва команди	**Опис**
/setname	Змінює існуючу назву
/setdescription	Присвоює текст, що буде відображатись при першому запуску бота
/setabouttext	Присвоює текст в полі “Про чат-бот”
/setcommands	Дозволяє створити список доступних команд
/deletebot	Видаляє обраного чат-бота

Крім команд для зміни основних параметрів Telegram-бота існує ряд команд [29], які дозволяють виводити незмінні параметри (наприклад, токен), а також присвоювати значення, які представлені в таблиці 3.4.

Таблиця 3.4 – Доступні команди для додаткового налаштування Telegram-боту

Назва команди	**Опис**
/token	Повертає отриманий раніше токен у обраного бота
/revoke	Анулює отриманий токен доступу до бота
/setinline	Вмикає або вимикає можливість викликати бота з інших чатів
/setinlinegeo	Вмикає або вимикає можливість передавати розташування бота з іншого чату

/setinlinefeedback	Дозволяє отримувати інформацію про кількість обраних користувачами команд
/setjoingroup	Визначає чи може бути доданий бот до групових діалогів
/setprivacy	Включає режим конфіденційності. В цьому режимі бот отримує, обробляє і відсилає назад інформацію окремо для кожного користувача в чаті

Після всіх необхідних налаштувань платформи з боку платформи telegram, яка є базисною для розробки Telegram-бота для забезпечення інформаційної підтримки, можна завершити процес створення таких ботів.

3.2.3. Реалізація клієнтської частини чат-бота

Першим кроком розробки платформи є вхід у спеціальному чат-боті, який має назву @duikt_bot. Перейшовши за посиланням https://t.me/duikt_bot, перед нами відкриється розроблений нами телеграм бот (рис. 3.6)

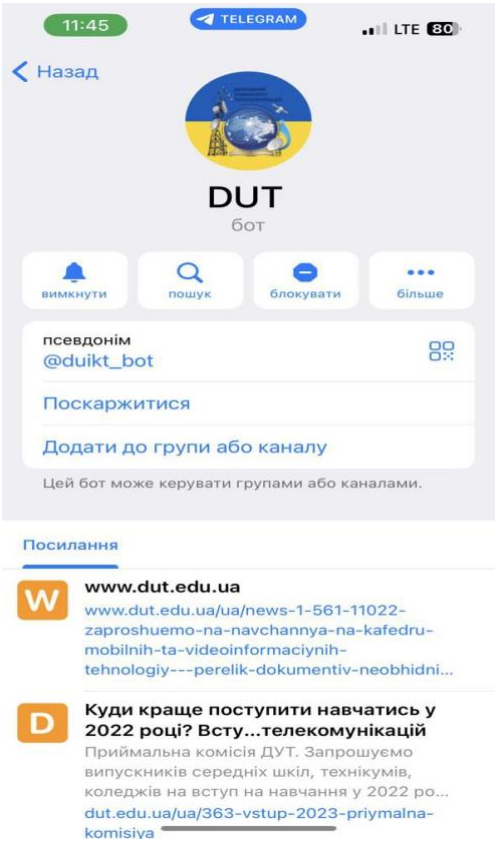


Рисунок 3.7 – Запуск Telegram-бота

Вхід починається з введення користувачем команди /start (рис. 3.8).

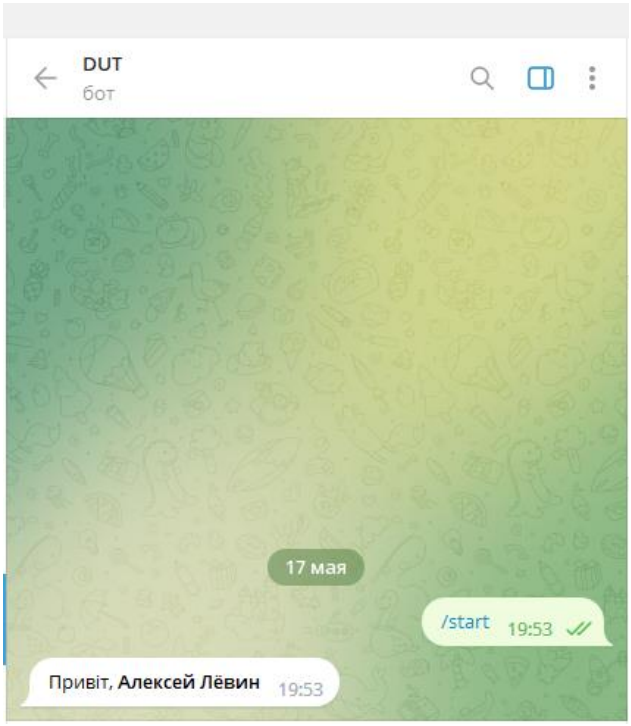


Рисунок 3.8 – запуск чату телеграм боту

Ввівши команду /help ми отримаємо наступну картину (рис. 3.9)

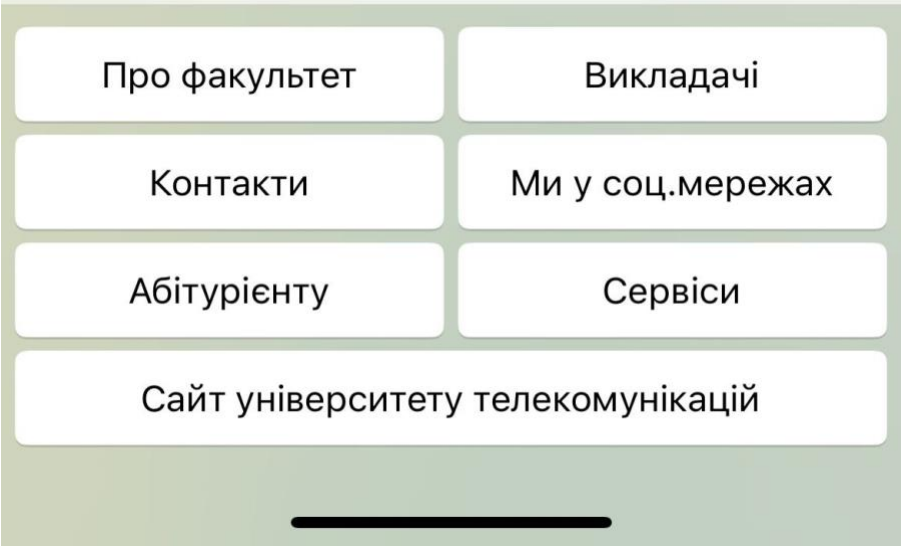


Рисунок 3.9 – меню чат-бота

Вибравши один із пунктів меню, користувач потрапить у під меню. Для прикладу розглянемо пункт меню «Про факультет» (рис. 3.10)

3.2.4. Реалізація серверної частини чат-бота

Користувачі Телеграм можуть взаємодіяти з чат-ботами двома способами: команди («/start», «/help» та інші) з параметрами, або вбудовані клавіатури (inline keyboards). Для зручності користувачів було вирішено зробити інтерфейс із вбудованою клавіатурою.

Для реалізації поставлених цілей потрібно п'ять різних меню:

– меню вибору шкіл;

- основне меню;

– меню перегляду новин;

– меню перегляду розділу

– меню перегляду контактів;

У Месенджер Телеграм кожна клавіатура реалізована як об'єкт, а її кнопки як JSON рядок. У документації ТелеграмBot API сказано, що кожна клавіатура повинна мати один обов'язковий параметр - ім'я кнопки (text), і шість необов'язкових - посилання (url), зворотні дані (callback_data), можливість

вбудованого запиту (switch_inline_query), можливість виведення клавіатури switch_inline_query_current_chat), виклик опису запущеної гри (callback_game) та кнопка з можливістю купівлі (pay).

Приклад реалізації клавіатури однієї з меню чат-бота зображено рисунку 3.11.

```
13  def help(message):
14      markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
15      about = types.KeyboardButton('Про факультет')
16      teachers = types.KeyboardButton('Викладачі')
17      contacts = types.KeyboardButton('Контакти')
18      social = types.KeyboardButton('Ми у соц.мережах')
19      abiturient = types.KeyboardButton('Абітурієнту')
20      servis = types.KeyboardButton('Сервіси')
21      site = types.KeyboardButton('Університет')
22      markup.add(about, teachers, contacts, social, abiturient, servis,site)
23      bot.send_message(message.chat.id,'Menu',reply_markup=markup)
```

Рисунок 3.11 – Реалізована клавіатура головного меню

3.3. Адміністративна панель чат бота

3.3.1 Розробка компонентів структури веб-додатку

Тестування телеграм бота проводилося вручну. Після кожного етапу розробки проводилося тестування продуктивності на заздалегідь підготовлених тест-кейсах, які складалися для визначення цілей розробки та використовуваних інструментів.

Робота з Telegram-ботом:

а) швидкість реагування бота на повідомлення про початок роботи;

б) правильна обробка повідомлень різного типу від користувача.

в) правильне відображення діалогу;

г) обробка натискання клавіші відправити повідомлення;

д) швидкість відповіді після того, як користувач написав повідомлення;

е) швидкість роботи бази даних;

є) оцінювання правильності відповідей.

Бот був протестований на мобільному пристрої Apple Iphone X з наступними характеристиками.

а) екран: 5,5", IPS LCD, 1920x1080, мультитач;

б) процесор: Apple A10 Fusion, 4x1;

в) операційна система: iOS 14.1.4;

г) ОЗУ: 3 ГБ;

д) вбудована пам'ять: 128 ГБ;

е) навігація: GPS;

g) Telegram: Telegram v 8.

Нижче наведено декілька прикладів тестування роботи розробленого телеграм бота @PhisicsAndMathFaculty

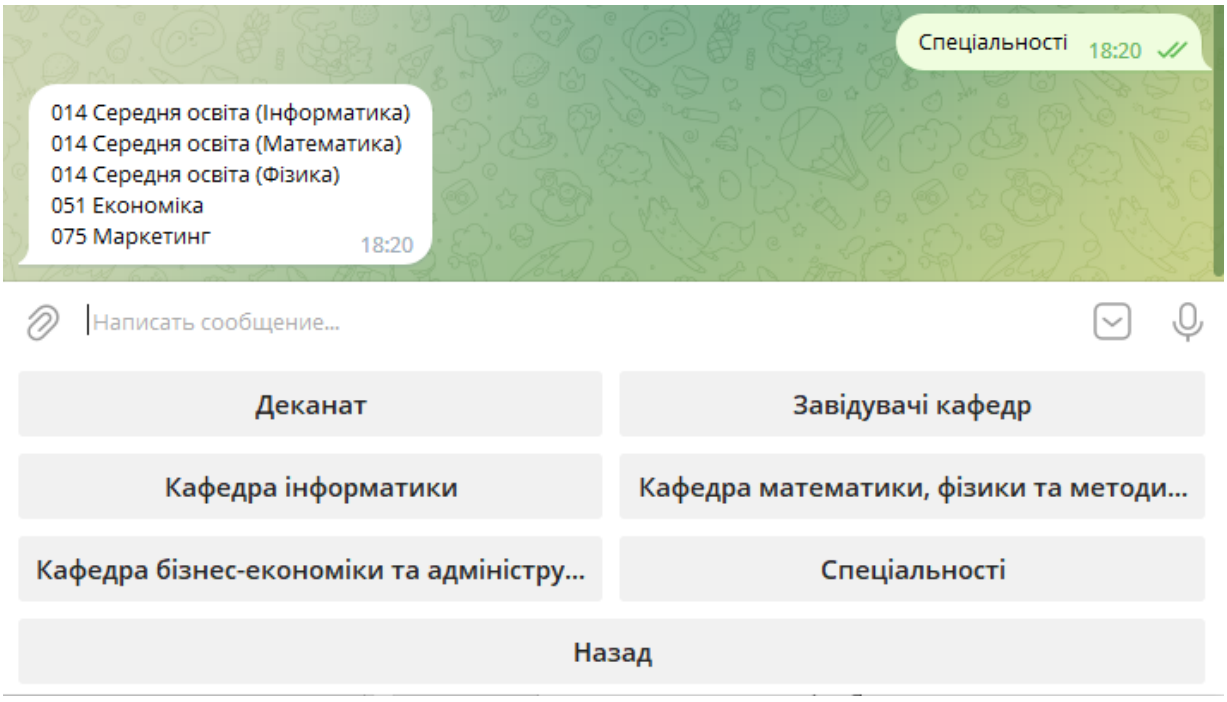


Рисунок 3.12 – Робота телеграм бота (спеціальності)

Рисунок 3.13 – Робота телеграм бота (ми у соц. мережах)

Рисунок 3.14 – Робота телеграм бота (Абітурієнту->Вступна компанія)

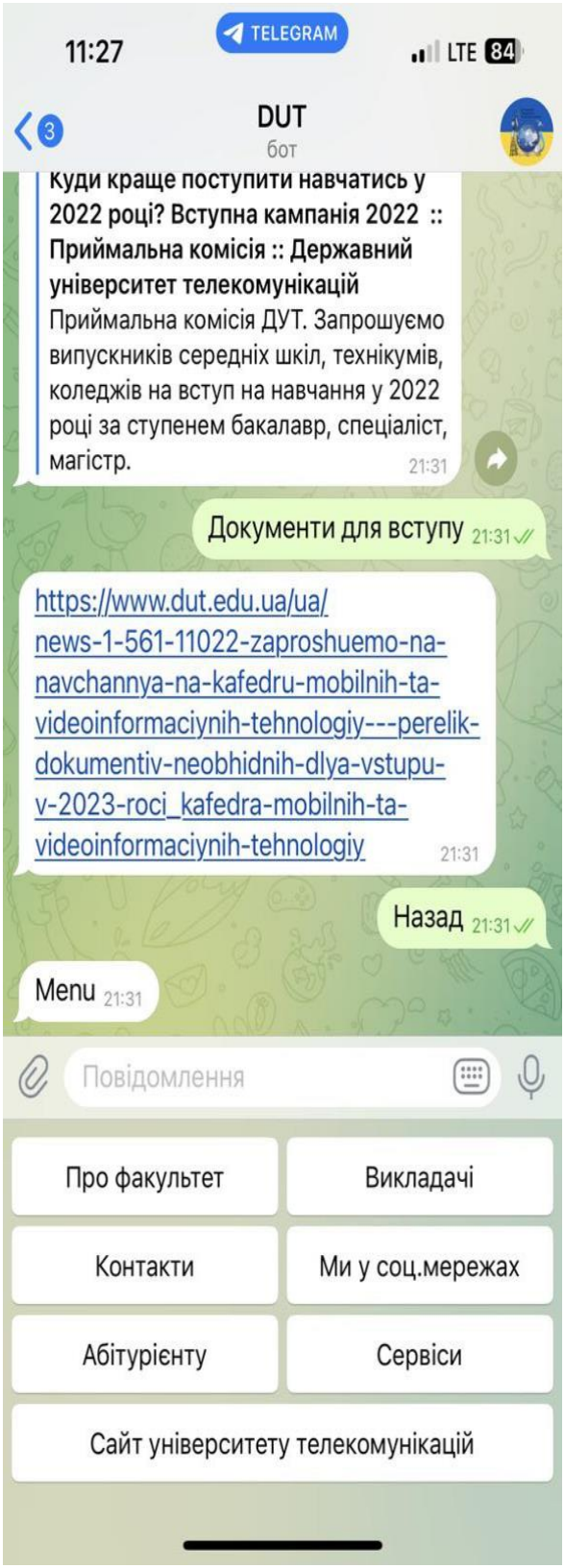


Рисунок 3.15 – Робота телеграм бота (Абітурієнту->Правила прийому)

ВИСНОВКИ

Під час виконання дипломної роботи було розглянуто історію створення та розвитку месенджерів як модерного засобу обміну інформацією та спілкування. Також детально розглянуто приклади різноманітних ботів, що спрощують виконання рутинних задач та дозволяють взаємодіяти з користувачами в автоматичному режимі. Детально описано принципи розробки подібного програмного забезпечення та висунуто основні задачі, які потрібно врахувати при розробці платформи для розробки Telegram-бота для забезпечення розпізнавання тексту з картинки. Слід зазначити, що акцент зроблено на підборі оптимальних інструментів, що є необхідними для коректної роботи зазначеної платформи.

Для створення платформи для розробки Telegram-бота для забезпечення інформаційної підтримки факультету було обрано найбільш стабільні засоби та інструменти розробника, а саме: мову програмування Python разом з набором корисних бібліотек, що використовувались при реалізації програмного коду в багатофункціональному середовищі розробки PyCharm, використано бібліотеку Telebot, яка реалізує безпосередньо Telegram Bot API, а також базу даних на основі SQLAlchemy.

Проаналізовано вже існуючі аналоги до розроблюваного програмного продукту та наведено приклади їх використання, основні переваги та недоліки. В результаті, було виявлено необхідність створення графічного інтерфейсу для взаємодії з користувачами, а не обмежуватись стандартним набором команд для керування ботом у Telegram. Крім того, було систематизовано та формалізовано алгоритм вибору інструментів розробки.

Як результат кваліфікаційної роботи, на базі вищеописаних принципів та із використанням зазначених технологій було створено платформу для розробки Telegram-бот для фізико-математичного факультету. Даний програмний продукт є основним доказом концепції автоматизації інформації про факультет за допомогою ботів для месенджерів (зокрема, у даному випадку для Telegram) та ілюструє можливу швидкість та гнучкість розробки.

В даний час популярність месенджерів як засобів спілкування незмінно зростає. Компанії, сім'ї, друзі щодня користуються можливостями обміну повідомленнями та медіаконтентом на відстані. Також варто відзначити зростання популярності такого виду програмних продуктів як чат-боти, які працюють на платформах месенджерів.

Цілодобова служба підтримки користувачів, конвертування документів та медіафайлів, замовлення таксі, пошук необхідних даних та багато іншого в даний час може бути реалізовано в рамках лише одного месенджера. Користувачам не доведеться завантажувати безліч додатків для вирішення вузьконаправлених завдань, тому що достатньо мати лише месенджер і необхідний набір чат-ботів, які не займають місце в пам'яті смартфона.

У рамках випускної кваліфікаційної роботи було виконано поставлені завдання. По-перше, були вивчені месенджери. Було проведено порівняння та аналіз переваг та недоліків, внаслідок чого був обраний месенджер Telegram як найзручніший та найдоступніший у плані документації Telegram Bot API.

По-друге, були вивчені наявні аналоги, а також виявлено їх переваги, недоліки та цікаві рішення. На основі цього були виявлені вимоги для розробки авторського чат-бота для допомоги абітурієнтам під час вступу до університету.

Рішення розробити чат-бот було продиктовано бажанням оптимізувати діяльність співробітників та студентів, які проходять практику в приймальній комісії університету, змушених витрачати багато часу на трансляцію інформації, що міститься у відкритих джерелах.

Месенджери – до сьогодні були неохопленою платформою, але дана випускна кваліфікаційна робота усунула та заповнила цю прогалину.

Основними перевагами роботів перед іншими формами взаємодії є зручність, надійність і доступність.

Чат-бот не йде на вихідні, на відміну від співробітників приймальної комісії, позбавлений можливості припуститися помилки через так званий людський фактор і захищений за допомогою спеціального алгоритму.

Функціонал месенджерів, зокрема, Telegram, націлений насамперед на мобільну аудиторію, яка активно використовує смартфони для роботи та розваг.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Gartner Inc. // [Електронний ресурс]. URL: http://www.gartner.com/newsroom/id/3215217/ (дата звернення: 21.04.2023);

2. Інтернет 2017-2018 у світі: Статистика та тренди // [Електронний ресурс]: URL: https://www.web-canape.ru/business/internet-2017-2018-v-mire-statistika-i-trendy/ (дата звернення: 21.04.2023)

3. Офіційна сторінка мобільного додатка «Абітурієнт ТПУ» // [Електронний ресурс]. URL: https://www.mobile.tpu.ru/ (дата звернення: 21.04.2023)

4. Новина про запуск програми мобільного додатка «Хочу в ТДУ» // [Електронний ресурс]. URL: http://www.tsu/news/tgu-zapustil-mobilnoe- prilozhenie-dlya-abituriento/ (дата звернення: 21.04.2023)

5. Моделювання UML. Загальні діаграми // [Електронний ресурс]: Моделювання UML. URL: http://book.uml3.ru/sec_1_5#p7 (дата звернення: 21.04.2023).

6. Використання компонентної архітектури у веб-додатках // [Електронний ресурс]: URL: https://fwdays.com/en/event/js-frameworks-day-2015/review/komponentnaia-arkhitektura-v-web-prilozheniiakn (дата звернення: 21.04.2023).

7. Архаков, Д. PHP: Робимо кнопки в Telegram API (inline-keyboards) // [Електронний ресурс]: URL: https://archakov.im/post/nodejs-make-buttons-on-telegram-api.html (Дата звернення: 21.04.2023).

8. Банокін П.І. Методи та засоби проектування інформаційних систем та технологій: навчальний посібник / П.І. Банокін; Київський політехнічний університет. -Київ: Вид-во киїівського політехнічного університету, 2015. -92 С.

9. Матвєєва Н. Ю., Технології створення та застосування чат-ботів [Електронний ресурс] / Н. Ю. Матвєєва, А. В Золотарюк. // Наукові записки молодих дослідників. - 2018. - №1. - С. 28-30. // [Електронний ресурс]: URL: https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov (дата звернення: 21.04.2023).

10. Офіційний сайт Telegtam API // [Електронний ресурс]: URL: https://core.telegram.org/api. (Дата звернення: 21.04.2023).

11. Комп'ютерна система PLATO [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/PLATO.

12. Talkomatic – Just think of it [Електронний ресурс] – Режим доступу: https://just.thinkofit.com/talkomatic/.

13. AIM messenger changed the way we communicate [Електронний ресурс] –Режим доступу: https://www.vox.com/culture/2017/12/15/16780418/aim-aol-instant-messenger-shutdowncultural-impact.

14. History of Friendster [Електронний ресурс] – Режим доступу: https://socialnetworking.lovetoknow.com/history-friendster.

15. Facebook [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Facebook.

16. Хто та як використовує месенджери в Україні? [Електронний ресурс] – Режим доступу : https://minfin.com.ua/2018/04/14/33167841/.

17. Рейтинг мобільних додатків за березень 2020 [Електронний ресурс] – Режим доступу : https://tns-ua.com/news/reyting-mobilnih-dodatkiv-za-berezen-2020.

18. Messina, C. 2016 will be the year of conversational commerce [Електронний ресурс] / C. Messina // Medium. – 2016. – Режим доступу: https://medium.com/chris-messina/2016-will-be-the-year-of-conversationalcommerce-1586e85e3991.

19. MTPoto Mobile Protocol [Електронний ресурс] – Режим доступу: https://core.telegram.org/mtproto. Дата звернення – травень 2020 р.

20. Козлов А. А. Телеграм-бот як простий та зручний спосіб отримання інформації [Електронний ресурс] / А. А. Козлов, А. В. Батищев // Територія науки. – 2017. – №5. – с. 55-64.

21. Ebot one – редактор ботів для Telegram [Електронний ресурс] – Режим доступу: https://ebot.one/.

22. Manybot – платформа для створення ботів [Електронний ресурс] – Режим доступу: https://manybot.io/.

23. How to make a responsive telegram bot [Електронний ресурс] – Режим доступу: https://www.sohamkamani.com/blog/2016/09/21/making-atelegram-bot/.

24. Васильєв О. Python на прикладах. Практичний курс по програмуванню / Васильєв Олексій., 2016. – С. 432.

25. Введення до модулю os [Електронний ресурс] – Режим доступу : https://docs.python.org/3/library/os.html.

26. Використання модулю subprocess [Електронний ресурс] – Режим доступу: https://docs.python.org/3/library/subprocess.html?highlight=subprocess#module-subprocess.

27. Використання пакету logging [Електронний ресурс] – Режим доступу: https://docs.python.org/3/library/logging.html?highlight=logging#module-logging..

28. The Python SQL Toolkit and Object Relational Mapper [Електронний ресурс] – Режим доступу : https://www.sqlalchemy.org/.

29. Приклади ботів для Telegram [Електронний ресурс] – Режим доступу: https://github.com/TelegramBots/telegram.bot.examples.

ДОДАТКИ

Додаток А (main.py)

```
import telebot
from telebot import types

bot = telebot.TeleBot('5472102192:AAF-23G5vdfos8cV1Wqcrt5MpLe8Z9vXlhQ')

@bot.message_handler(commands=['start'])
def start(message):
    mess = f'Привіт, <b>{message.from_user.first_name}
{message.from_user.last_name}</b>'
    bot.send_message(message.chat.id,mess, parse_mode='html')



@bot.message_handler(commands=['help'])
def help(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    about = types.KeyboardButton('Про факультет')
    teachers = types.KeyboardButton('Викладачі')
    contacts = types.KeyboardButton('Контакти')
    social = types.KeyboardButton('Ми у соц.мережах')
    abiturient = types.KeyboardButton('Абітурієнту')
    servis = types.KeyboardButton('Сервіси')
    site = types.KeyboardButton('Університет')
    markup.add(about, teachers, contacts, social, abiturient, servis,site)
    bot.send_message(message.chat.id,'Menu',reply_markup=markup)

@bot.message_handler(content_types=['text'])
def message(message):
```

```
    if message.text == 'Про факультет':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        decanat = types.KeyboardButton('Деканат')
        zavcaf = types.KeyboardButton('Завідувачі кафедр')
        kaf_info = types.KeyboardButton('Кафедра інформатики')
        kaf_mat = types.KeyboardButton('Кафедра математики, фізики та методик їх
навчання')
        kaf_busness = types.KeyboardButton('Кафедра бізнес-економіки та
адміністрування')
        specialist = types.KeyboardButton('Спеціальності')
        cancel = types.KeyboardButton('Назад')
        markup.add( decanat,zavcaf, kaf_info, kaf_mat, kaf_busness, specialist, cancel)
        bot.send_message(message.chat.id, 'Про факультет', reply_markup=markup)
    elif message.text == 'Деканат':
        mess = 'ПІБ, науковий ступінь, вчене звання, посада, фото – про кожного'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Завідувачі кафедр':
        mess = 'ПІБ, науковий ступінь, вчене звання, посада, фото – про кожного'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Кафедра інформатики':
        mess = 'Інформація про кафедру, логотип, …'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Кафедра математики, фізики та методик їх навчання':
        mess = 'Інформація про кафедру, логотип, …'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Кафедра бізнес-економіки та адміністрування':
        mess = 'Інформація про кафедру, логотип, …'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Спеціальності':
        mess = '014 Середня освіта (Інформатика)\n014 Середня освіта
```

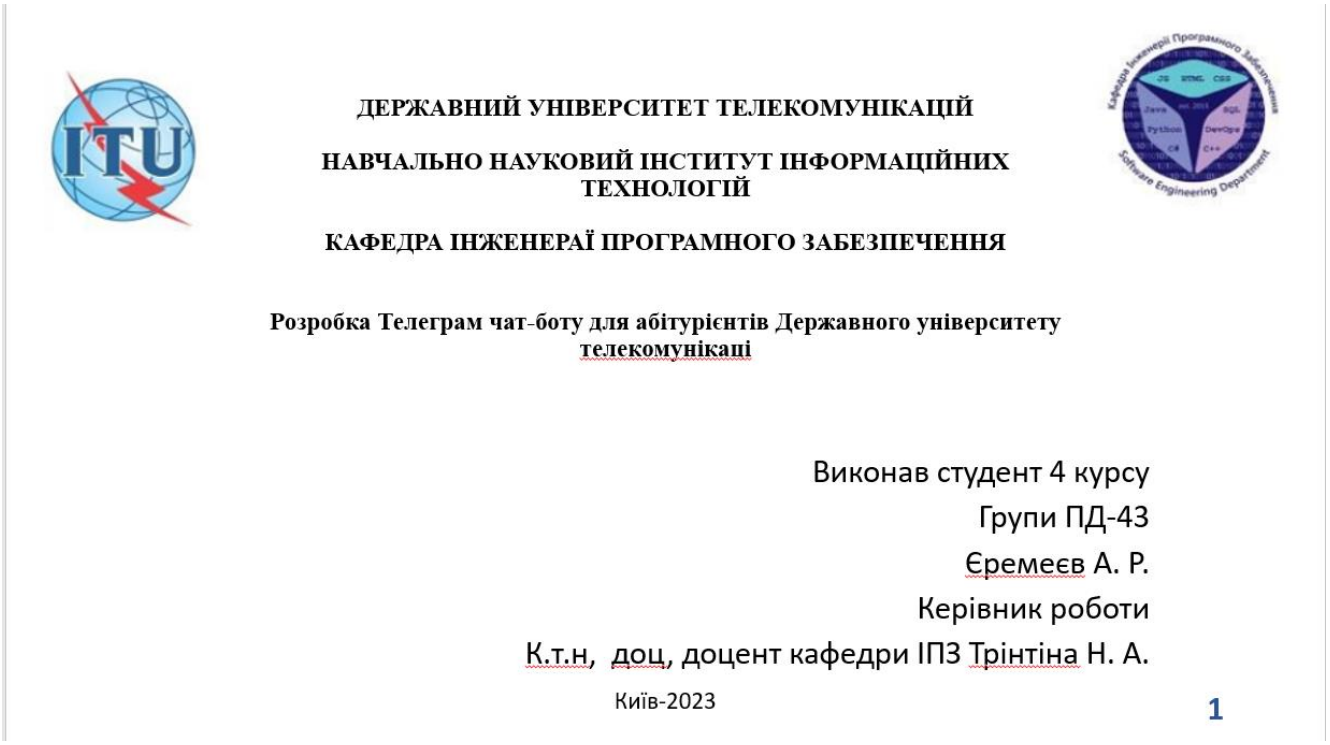
```
(Математика)\n014 Середня освіта (Фізика)\n051 Економіка\n075 Маркетинг'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Викладачі':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        teachers = types.KeyboardButton('Перелік усіх викладачів за алфавітом.')
        cancel = types.KeyboardButton('Назад')
        markup.add( teachers, cancel)
        bot.send_message(message.chat.id, 'Викладачі', reply_markup=markup)
    elif message.text == 'Перелік усіх викладачів за алфавітом.':
        mess = 'ПІБ, науковий ступінь, вчене звання, посада, фото, кафедра, контакти
(тел, мейл за бажанням)'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Контакти':
        mess = 'Адреса, тел, сайт, мейл,'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Ми у соц.мережах':
        mess = 'Фізико-математичний факультет:\n'
        mess +='Instagram: https://www.instagram.com/dut_official/\n'
        mess +='Facebook: https://www.facebook.com/dut.edu.ua/\n'
        mess +='YouTube:
https://www.youtube.com/channel/UCrkaw_RZIE7scOoOAZy_j8g\n'
        mess +='(https://www.youtube.com/channel/UCrkaw_RZIE7scOoOAZy_j8g)'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Абітурієнту':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        priyom = types.KeyboardButton('Приймальна комісія')
        vstup = types.KeyboardButton('Вступна кампанія')
        pravilo = types.KeyboardButton('Правила прийому')
        document = types.KeyboardButton('Документи для вступу')
        cancel = types.KeyboardButton('Назад')
```

```
        markup.add( priyom, vstup, pravilo, document, cancel)
        bot.send_message(message.chat.id, 'Абітурієнту', reply_markup=markup)
    elif message.text == 'Приймальна комісія':
        mess = 'Приймальна комісія'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Вступна кампанія':
        mess = 'https://www.dut.edu.ua/ua/363-vstup-2023-priymalna-komisiya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Правила прийому':
        mess = 'https://dut.edu.ua/ua/108-pravila-priyomu-priymalna-komisiya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Документи для вступу':
        mess = 'https://www.dut.edu.ua/ua/news-1-561-11022-zaproshuemo-na-
navchannya-na-kafedru-mobilnih-ta-videoinformaciynih-tehnologiy---perelik-
dokumentiv-neobhidnih-dlya-vstupu-v-2023-roci_kafedra-mobilnih-ta-
videoinformaciynih-tehnologiy'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Сервіси':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        elec = types.KeyboardButton('Електронний розклад')
        dist = types.KeyboardButton('Дистанційне навчання')
        kab_stud = types.KeyboardButton('Кабінет студента')
        kab_vik = types.KeyboardButton('Кабінет викладача')
        cancel = types.KeyboardButton('Назад')
        markup.add( elec, dist, kab_stud, kab_vik, cancel)
        bot.send_message(message.chat.id, 'Сервіси', reply_markup=markup)
    elif message.text == 'Електронний розклад':
        mess = 'https://www.dut.edu.ua/ua/140-rozklad-zanyat-asu-navchannya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Вартість навчання':
```

```
        mess = 'https://www.dut.edu.ua/ua/430-vartist-navchannya-priymalna-komisiya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Підготовче відділення':
        mess = 'https://www.dut.edu.ua/ua/459-zagalna-informaciya-pidgotovche-
viddilennya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Міжнародна діяльність':
        mess = 'https://www.dut.edu.ua/ua/331-zaproshuemo-na-navchannya-navchannya'
        bot.send_message(message.chat.id, mess, parse_mode='html')
    elif message.text == 'Військова кафедра':
        mess = 'https://www.dut.edu.ua/ua/213-zagalna-informaciya'
        bot.send_message(message.chat.id, mess, parse_mode='html')

    elif message.text == 'Назад':
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
        about = types.KeyboardButton('Про факультет')
        teachers = types.KeyboardButton('Викладачі')
        contacts = types.KeyboardButton('Контакти')
        social = types.KeyboardButton('Ми у соц.мережах')
        abiturient = types.KeyboardButton('Абітурієнту')
        servis = types.KeyboardButton('Сервіси')
        site = types.KeyboardButton('Сайт університету телекомунікацій')
        markup.add(about, teachers, contacts, social, abiturient, servis, site)
        bot.send_message(message.chat.id, 'Menu', reply_markup=markup)


    elif (message.text=="Hello" or message.text=="привіт" or message.text=="Привіт"
or message.text=="hello"):
        bot.send_message(message.chat.id, 'І тобі привіт', parse_mode='html')
    elif message.text=="id":
```

```
    bot.send_message(message.chat.id, f"Твій ID: {message.from_user.id}",
parse_mode='html')
  else:
    bot.send_message(message.chat.id, "Я тебе не розумію", parse_mode='html')


bot.polling(none_stop=True)
```

Додаток В (презентація)



Мета, об’єкт та предмет дослідження

- **Мета роботи** - спрощення та підвищення ефективності процесу подання документів до ВНЗ за допомогою Телеграм чат-боту.
- **Об’єкт дослідження** – програмне забезпечення, що розробляється для допомоги абітурієнтам при вступі до державного університету телекомунікацій.
- **Предмет дослідження** – Телграм чат – боти інших закладів «Абітурієнт ТПУ» і «Хочу в КДУ».

2

Задачі до димпломної роботи

1. ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ
2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ
3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3

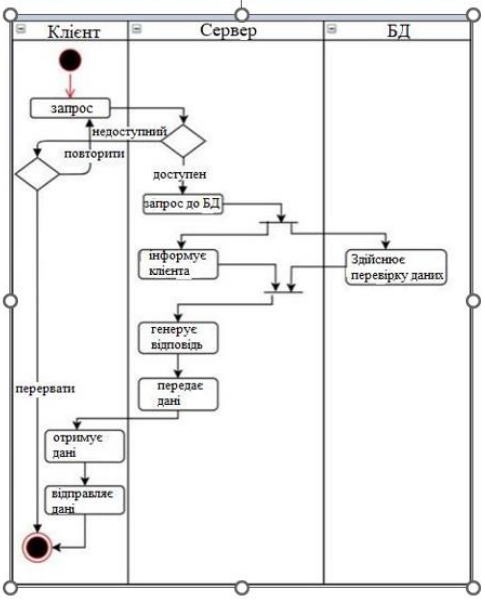
Параметри	Додатки		
	Телеграм бот дут	Абітуріент ТПУ	Хочу в КДУ
Окреме місце для телефона в программі	+	+	+
Можливість перегляду інформації про напрямки ВНЗ	+	+	+
Можливість перегляду про новини та заходи	+	+	+
Кросплатформенність	+	+	+
Автономність	+	-	-
Дизайн	+	-	-
Отримання інформації на пошту	+	-	-

4

Програмні засоби реалізації

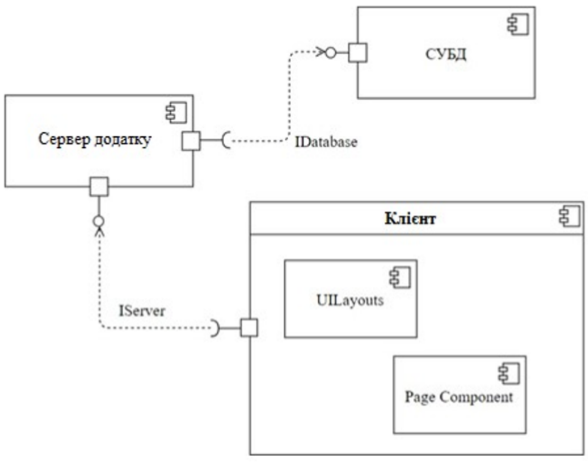
5

Діаграма діяльності процесу



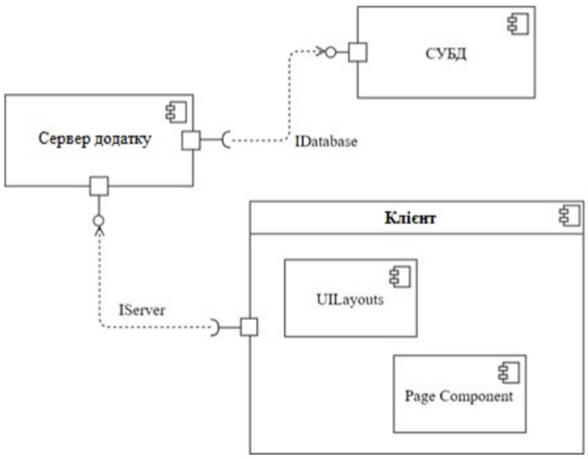
6

Архітектура Сервісу



7

Архітектура Сервісу

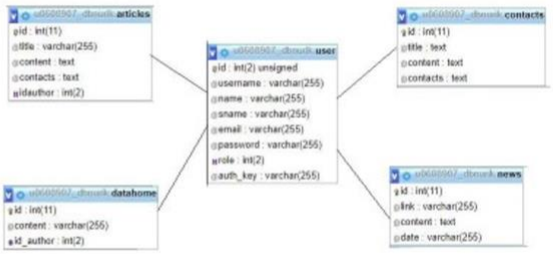


7

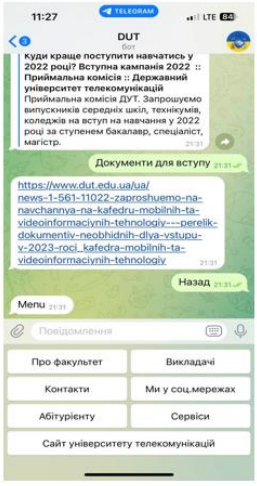
Діаграма варіантів використання

9

Схема бази данних



Екрані форми



Апробація

- Єремеєв А.Р. Розробка Телеграм чат-боту для абітурієнтів Державного університету телекомунікацій. Матеріали наукової конференції «Період трансформаційних процесів в світовій науці: задачі та виклики». Збірник тез. – 02.06.2023, м.Одеса

Висновки

1. Проведено аналіз предметної області.
2. Проведено дослідження інструментів та програмних засобів реалізації.
3. Визначено концепцію, вимоги, створено UML діаграми.
4. Розроблено застосунок під платформу телеграм чат - боту із користувацьким сервісом, додаванням учасників.
5. Розроблено та закладено можливість для подальшої модифікації для покращення функціоналу та розширення.

Дякую за увагу!