


Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА
на тему
Телеграм чат-бот для орієнтування абітурієнтів
ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: студент 4 курсу, групи КІ-20мс3
Спеціальності 123 — Комп'ютерна інженерія


Гребінь О.О. 

Керівник к.т.н., доц.


Черняк О. І. 

" 15 " 06 2022 р.

Рецензент к.т.н., доц.

Куперштейн Л.М. 

" 16 " 06 2022 р.

Допущено до захисту
д.т.н., проф. Азаров О.Д. 

" 17 " 06 2022 р.

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

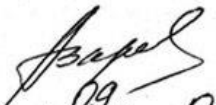
Освітньо—кваліфікаційний рівень бакалавр

Напрямок підготовки 123 — «Комп'ютерна інженерія»
(шифр і назва)

Спеціальність 123 — «Комп'ютерна інженерія»
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

 О.Д. Азаров
" 09 " 02 2022 р.

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ

студенту **Гребеню Олександрю Олександровичу**

1 Тема роботи: Телеграм-чат бот для орієнтування абітурієнтів.

Керівник роботи Черняк Олександр Іванович , к.т.н., доцент, кафедри ОТ
затверджені наказом вищого навчального закладу від 24.03.2022 року №66.

2 Строк подання студентом роботи *17. 06. 2022*

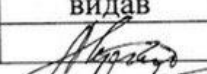

3 Вихідні дані до роботи: технічний опис програмного застосунку, мова програмування Python, цільова платформа — Telegram, технологія розробки — Telebot, середовище розробки Visual Studio Code.

4 Зміст розрахунково-пояснювальної записки: вступ, аналіз сучасних методів розробки, варіантний вибір засобів для розробки програмного забезпечення, програмна реалізація, тестування розробленого програмного застосунку.

5 Перелік графічного матеріалу: схема взаємодії користувача з телеграм ботом, блок-схема алгоритму телеграм чат-бота, схема принципу роботи на платформі Telegram.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Черняк О.І., к.т.н., доц.		

7 Дата видачі завдання 09.03.2022 р.

8 Календарний план приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання	
1	Постановка задачі роботи	09.03.22	12
2	Пошук матеріалів по технологіям розробки чат ботів	10.03.22 — 24.03.22	12
3	Розробка структури програмного застосунку	25.03.22 — 28.03.22	12
4	Обґрунтування та вибір засобів реалізації програмного застосунку	29.03.22 — 04.04.22	12
5	Розробка функцій та методів роботи чат бота	05.04.22 — 28.04.22	12
6	Підготовка матеріалів та розробка алгоритму роботи чат бота	29.04.22 — 26.05.22	12
7	Оформлення пояснювальної записки та ілюстративного матеріалу	27.05.22 — 06.06.22	12
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	07.06.22	12

Студент

Керівник роботи




Гребінь Олександр Олександрович

к.т.н., доц. Черняк Олександр Іванович

АНОТАЦІЯ

Пояснювальна записка містить 78 аркушів, 18 рисунків, 3 лістинга та 18 посилань.

Бакалаврська дипломна робота присвячена розробці чат-бота для орієнтування абітурієнтів з майбутньої спеціальності при вступі до начального закладу.

В роботі виконано аналіз предметної області, здійснено аналіз сучасних методів розробки віртуальних помічників — чат-ботів, проаналізовано та вичено способи створення серверних програм.

Перевірка працездатності чат-бота протестована шляхом практичного тестування системи.

Ключові слова: серверна програма, віртуальний помічник, чат-бот, Telegram.

ABSTRACT

The explanatory note contains 78 sheets, 18 figures, 3 listings and 18 references.

The bachelor's thesis is devoted to the development of a chat-bot to guide entrants in future specialties when entering the primary school.

The analysis of the subject area is performed, the analysis of modern methods of development of virtual assistants — chat-bot is carried out, the methods of creating server programs are analyzed and studied.

Checking the functionality of the chatbot is tested by practical testing of the system.

Keywords: server program, virtual assistant, chat-bot, Telegram.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ЧАТ-БОТІВ	10
1.1 Аналіз предметної області та актуальності розробки	10
1.2 Аналіз технологій створення та застосування чат-ботів.....	12
1.2.1 Огляд месенджера Telegram.....	13
1.3 Огляді аналогів розробки та формулювання вимог розробки.....	14
1.3.1 Чат-бот «Вступник ФМТП».....	15
1.3.2 Чат-бот «Вступник КПП»	16
1.4 Вимоги до створюваного бота	17
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМУ РОБОТИ ЧАТ-БОТА ДЛЯ ОРІЄНТУВАННЯ АБІТУРІЄНТІВ	19
2.1 Проектування функціональних можливостей чат-бота	19
2.2 Розробка блок-схеми алгоритму роботи чат-бота орієнтування абітурієнтів.....	20
2.3 Загальний алгоритм реалізації проєкта.....	22
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЧАТ-БОТА	23
3.1 Варіантний аналіз і обґрунтування вибору програмних засобів.....	23
3.1.1 Python.....	23
3.1.2 Java.....	24
3.1.3 C#	24
3.1.4 JavaScript	25
3.1.5 Обґрунтування вибору мови програмування	25
3.2 Вибір середовища розробки.....	26
3.2.1 Інтегроване середовище PyCharm	26

					08-23.БДР.003.00.000 ПЗ			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Гребінь О.О.			Телеграм чат-бот для орієнтування абітурієнтів Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Черняк О.І.					6	78
<i>Реценз.</i>		Куперштейн Л.М				ВНТУ, гр. КІ-20мсЗ		
<i>Н. Контр.</i>		Швець С.І.						
<i>Затверд.</i>		Азаров О.Д.						

3.2.2 Середовище Visual Studio Code	27
3.3 Розробка функцій чат-бота.....	27
3.3.1 Реєстрація чат-бота для Telegram Bot Api	27
3.3.1. Створення першого повідомлення та інтерфейсу чат-бота.....	32
3.3.2. Розробка головного меню та його складових	34
3.4 Тестування програмного модуля	40
3.5 Інструкція користувача.....	44
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	46
ДОДАТОК А Технічне завдання	47
ДОДАТОК Б Взаємодія користувача та telegram-бота	51
ДОДАТОК В Алгоритм роботи чат-бота.....	52
ДОДАТОК Г Принцип роботи чат-бота на платформі Telegram.....	53
ДОДАТОК Д Лістинг чат-бота	54
ДОДАТОК Е Протокол перевірки навчальної (кваліфікаційної) роботи	78

					08-23.БДР.003.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Чат-боти — сучасний та ефективний засіб, який стає дедалі більш популярним для інформування користувачів з будь-яких питань. Враховуючи, високий показник конверсії відкриттів і переглядів повідомлень, які приходять до месенджерів, чат-боти є значно ефективнішим за будь-який інший інструмент інформування. Для користувачів буде значно простіше отримати необхідну інформацію в автоматичному режимі за допомогою чат-бота, ніж намагатися отримати ту ж саму інформацію за допомогою пошуку в інтернеті, що значно підвищує лояльність. Популярність месенджерів зростає з кожним днем, і зростання популярності даних програм прогнозується мінімум на найближчі 5-10 років вперед.

Основними перевагами для створення чат-бота є:

— підтримка в будь-який час доби, оскільки доступний 24/7 та швидка відповідь на питання користувача, тобто користувач телеграм-боту може вирішити свої питання незалежно від часу доби;

— економія часу користувача, а це в свою чергу швидкий зворотній зв'язок, за рахунок заздалегідь прописаних питань та відповідей для користувача у дереві рішень;

— декілька людей можуть використовувати телеграм-бот в один час, що дає можливість надати необхідну інформацію кожному користувачеві;

— телеграм-бот працює на великій кількості різних платформ, таких як Windows, Linux, Android, iOS та інших, що підтримують Telegram;

— простота інтеграції, оскільки немає необхідності завантажувати додаток або додаткову програму [12-16].

Базуючись на розглянутому, розробка, яка полягає в наданні доступу до інформації абітурієнтам є **актуальною задачею**.

Метою роботи є розробка чат-бота для популярного месенджера Telegram мовою програмування Python в середовищі Visual Studio Code, який допоможе абітурієнтам зорієнтуватися щодо вступу в навчальний заклад.

Задачі дослідження бакалаврської роботи:

- проаналізувати методи та технології розробки чат-ботів;
- проаналізувати існуючі аналоги;
- проаналізувати та обрати засоби реалізації чат-бота;
- виконати моделювання;
- виконати розробку чат-бота за допомогою методів бібліотеки PyTelegramBotAPI мови Python;
- протестувати розроблений чат-бот.

Об'єкт дослідження — розробка чат-бота.

Предмет дослідження — чат-бот, як програмний застосунок, що використовується в месенджерах з метою інформування та охоплення великої кількості користувачів.

Методи дослідження бакалаврської роботи: у роботі використано принципи об'єктно-орієнтованого програмування мовою Python для реалізації запропонованого підходу.

Апробація результатів бакалаврської роботи: зроблено доповідь на LI науково-технічній конференції підрозділів Вінницького національного технічного університету.

Практичне значення отриманих результатів полягає в можливості використання розробленого чат-бота в навчальному закладі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ЧАТ-БОТІВ

Головною ідеєю роботи є створення чат-бота помічника абітурієнта. Для виконання роботи необхідно здійснити аналіз предметної області та обрати технології для реалізації поставленого завдання.

1.1 Аналіз предметної області та актуальності розробки

На сьогоднішній день ринок освітніх послуг в Україні представлений великою кількістю вищих навчальних закладів (ВНЗ). Вибір, перед яким стоять абітурієнти, носить важливу роль і закладає фундамент у професійному становленні майбутнього спеціаліста.

Приймальна комісія — це адміністративний підрозділ ВНЗ, що займається організацією прийому на навчання до навчального закладу. Приймальна комісія створюється з метою організації прийому абітурієнтів та зарахування до освітнього закладу вищої професійної освіти.

Згідно з нормативним актом ВНЗ про правила прийому абітурієнтів, навчальний заклад зобов'язаний ознайомити вступника та (або) його батьків (законних представників) зі своїм статутом, з ліцензією на здійснення освітньої діяльності, зі свідоцтвом про державну акредитацію, з освітніми програмами та іншими документами, що регламентують роботу навчального закладу та здійснення освітньої діяльності, права та обов'язки студентів.

З метою інформування про прийом на навчання ВНЗ розміщує інформацію на офіційному сайті в інформаційно-телекомунікаційній мережі «Інтернет» (далі — офіційний сайт), а також забезпечує вільний доступ у приміщенні ВНЗ до інформації, розміщеної на інформаційному стенді приймальної комісії та (або) в електронній інформаційній системі.

Серед методів взаємодії приймальної комісії ВНЗ з абітурієнтами можна виділити два: очний та заочний.

До очних методів відносять:

- консультація членів приймальної комісії в навчальному закладі;
- присутність абітурієнта на Дні відкритих дверей;
- участь абітурієнта у наукових заходах, що проводяться навчальним закладом;
- участь абітурієнта у творчих заходах, що проводяться навчальним закладом;
- участь абітурієнта в екскурсіях, що проводяться навчальним закладом.

До заочних методів відносять:

- телефонна консультація абітурієнта працівником приймальної комісії;
- консультація поштою абітурієнта працівником приймальної комісії;
- консультація електронною поштою абітурієнта працівником приймальної комісії;
- консультація абітурієнта працівником приймальної комісії на сайті приймальної комісії у розділі «Питання-відповідь»;
- консультація абітурієнта працівником приймальної комісії у соціальних мережах (наприклад, Facebook);
- консультація абітурієнта працівником приймальної комісії у месенджері.

На сьогоднішній день приймальна комісія активно освоює заочні методи взаємодії з абітурієнтами. Процес уникнення «старих», очних методів і засобів взаємодії йде нерівномірно і досить довго, незважаючи на очевидні переваги заочних, серед яких: доступність, низька вартість, зручність для всіх учасників взаємодії, прозорість, можливість зберегти історію листування та ін. Найбільше із запропонованих дистанційних методів використовується інформування абітурієнтів за допомогою офіційного сайту ВНЗ, сторінок та акаунтів у соціальних мережах. При цьому месенджери до сьогодні є неохопленою платформою, яка потенційно може стати найзручнішим засобом налагодження контакту між абітурієнтами та працівниками приймальної комісії.

Месенджер (IM = Instant Messenger) — це програма, мобільний додаток або вебсервіс для миттєвого обміну повідомленнями.

На сьогоднішній день існує велика різноманітність месенджерів, серед яких на особливу увагу заслуговує месенджер Telegram.

1.2 Аналіз технологій створення та застосування чат-ботів

З кожним роком чат-боти стають дедалі популярнішими і користуються великим попитом. Вони є невід'ємною частиною повсякденності і дуже затребувані в багатьох і абсолютно різних сферах — від банківських структур до інтернет-магазинів. Ботів можна зустріти або налаштувати в месенджерах і соцмережах (Telegram, Viber, Facebook), на сайтах, в мобільних додатках, системах управління (Siri), голосових асистентів (Аліса) і так далі.

Розглянемо більш детально, що ж таке чат-боти і як вони функціонують.

При створенні чат-ботів вибір типу має ключове значення. Існує два типи ботів: ускладнені та прості.

Ускладнені чат-боти працюють на основі штучного інтелекту, завдяки чому вони можуть розпізнавати питання будь-якої складності формулювання і на різних мовах.

Прості — функціонують, дотримуючись своїх інструкцій: вони можуть надавати відповідь лише на ті питання, які вже прописані в системі. Оскільки в месенджерах, зокрема Telegram, використовуються прості боти, тому саме цей тип ми будемо розглядати.

Чат-бот — це спеціальна програма, в корені якої прописаний алгоритм, що дозволяє автоматично обробляти повідомлення від користувачів та надавати коректні й логічні відповіді. В основі механізму дії розташовується завчасно прописаний сценарій спілкування, що і дозволяє швидко реагувати на дії співрозмовника. Тобто, в чат-ботах користувач спілкується ніби з роботом, який замінює реальну людину.

1.2.1 Огляд месенджера Telegram

Telegram — це кросплатформна програма, розроблена мовою програмування C++, що дозволяє обмінюватися повідомленнями та файлами більшості форматів. Месенджер використовує спеціально розроблену серверну частину із закритим кодом, що працює на серверах Німеччини та США.

Згідно з дослідженнями, проведеним незалежним виданням «Telegram-store», вік та соціальне становище користувачів у україномовному сегменті розподіляється таким чином:

- вік від 18 до 35 років;
- здобуває або має вищу освіту;
- має роботу інтелектуального характеру (великий відсоток IT-фахівців).

Функціонал месенджерів, зокрема, Telegram, націлений насамперед на мобільну аудиторію, яка активно використовує смартфони для роботи та розваг: користувачеві не потрібно заходити на сайт, достатньо надіслати повідомлення боту.

Telegram має ряд переваг:

- приватність — всі чати зашифровуються, а повідомлення знищуються через зазначений час;
- швидкість — швидкість доставки повідомлень вища, ніж у аналогів;
- розподіл — сервери Telegram розташовані по всьому світу, що підвищує стійкість до відмови;
- відкритість — використання відкритого протоколу MTProto та API, безкоштовні для всіх (рисунок 1.1);
- відсутність підписок та реклами;
- відсутність обмежень на розмір повідомлень та вкладених файлів.

MTPROTO 2.0, part I

Cloud chats (server-client encryption)

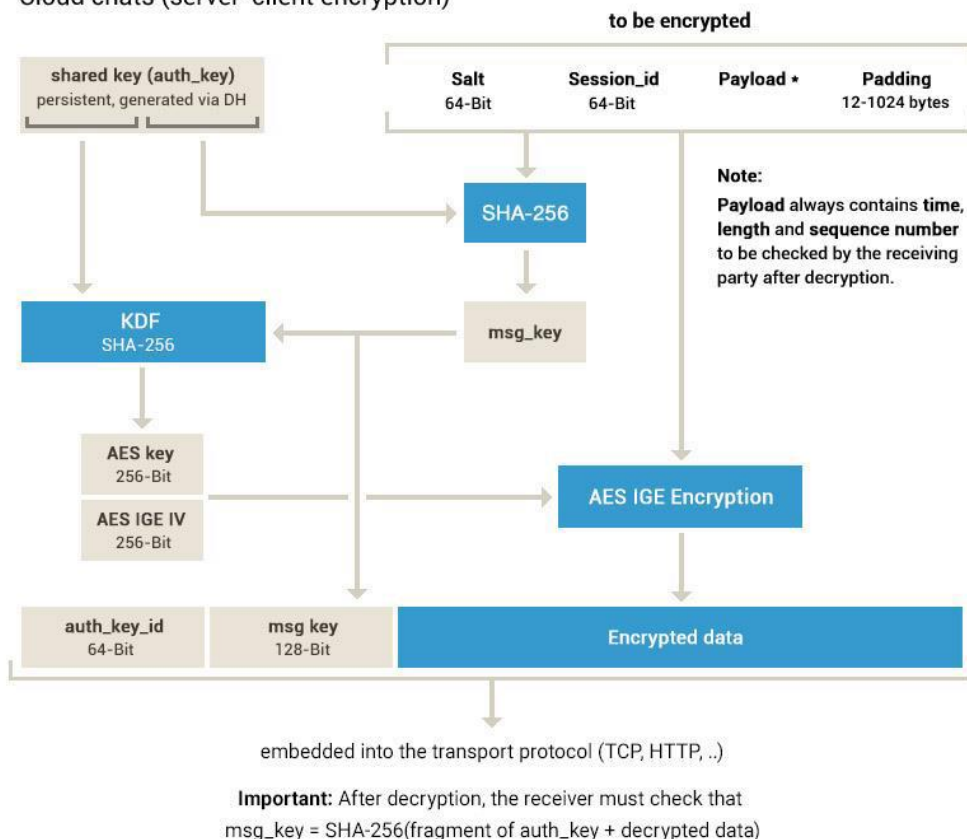


Рисунок 1.1 — Схема роботи протоколу MTPROTO

Секретний чат корисний людям, які потребують найбезпечнішого спілкування. На додаток до використання основного протоколу, усі повідомлення шифруються ключем за принципом "пристрій до пристрою". Ніхто, включаючи працівників Telegram, не може перехопити та розшифрувати повідомлення. Повідомлення з таких чатів не зберігаються на сервері програми, тому їх не можна пересилати. Однак завжди існує ризик витоку абсолютно секретної інформації через скріншоти інтерфейсу Telegram з відкритими секретними функціями чату.

1.3 Огляді аналогів розробки та формулювання вимог розробки

Під час детального вивчення предметної області було переглянуто ряд сайтів навчальних закладів з інформацією для абітурієнта. Серед великої кількості

інформації, представленої на сайтах, дуже важко зорієнтуватися і тим більше знайти необхідну інформацію. На допомогу в такому випадку підійде чат-бот.

Перед розробкою програмної реалізації, необхідно виконати огляд існуючих чат-ботів для уникнення помилок вже створених ботів та зрозуміти загальну концепцію їх використання та створити чат-бот з оригінальним набором функцій. Розроблюваний чат-бот використовується тільки в Telegram.

Бот має надавати відповідь на усі питання, які виникають при виборі майбутньої спеціальності. В чат-боті використовуються інлайнова клавіатура та кнопки.

1.3.1 Чат-бот «Вступник ФМТП»

На факультеті міжнародної торгівлі та права Державного торговельно-економічного університету (КНТЕУ) використовується чат-бот «Вступник ФМТП», який представлений на рисунку 1.2.



Рисунок 1.2 — Бот Вступник ФМТП

Чат-бот «Вступник ФМТП» — це віртуальний консультант про спеціальності факультету, правила, етапи та вимоги вступу. Даний віртуальний помічник забезпечує зв'язок з приймальною комісією та надає всю необхідну інформацію. Чат-бот «Вступник ФМТП» зручний в користуванні, надає відповідну інформацію, яка пропонується користувачу в структурованому вигляді, забезпечує інтерактивне спілкування.

1.3.2 Чат-бот «Вступник КПІ»

Особливої увагу заслуговує чат-бот «Вступник КПІ». На відміну від попереднього чат-бота, «Вступник КПІ» пропонує інформацію про навчання, умови вступу, вартість навчання тощо, у вигляді повідомлень без участі користувача (рисунок 1.3).

Такий чат-бот не є інтерактивним, а виконує роль інформаційного бюлетеня.

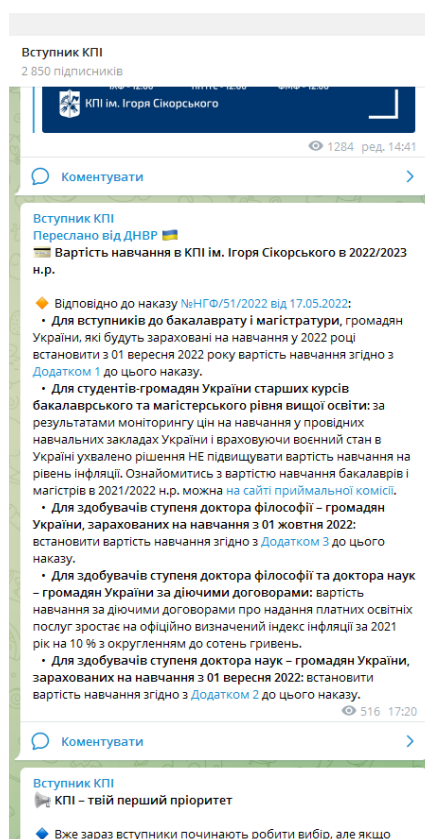


Рисунок 1.3 — Чат-бот «Вступник КПІ»

Чат-бот «Вступник КПІ» є інформаційним чатом, інформація в якому не залежить від вибору користувача.

1.4 Вимоги до створюваного бота

Вимоги до створюваного чат-бота були сформовані після аналізу аналогів чат-ботів для абітурієнтів та врахування основних переваг чат-ботів:

— підтримка в будь-який час доби, оскільки доступний 24/7 та швидка відповідь на питання користувача, тобто користувач телеграм-боту може вирішити свої питання незалежно від часу доби;

— економія часу користувача, а це в свою чергу швидкий зворотній зв'язок, за рахунок заздалегідь прописаних питань та відповідей для користувача у дереві рішень;

— декілька людей можуть використовувати телеграм-бот в один час, що дає можливість надати необхідну інформацію кожному користувачеві;

— телеграм-бот працює на великій кількості різних платформ, таких як Windows, Linux, Android, iOS та інших, що підтримують Telegram;

— простота інтеграції, оскільки немає необхідності завантажувати додаток або додаткову програму [12-16].

Умовним замовником чат-бота є ВіФК НУХТ. Коледж має сайт де можна знайти всю необхідну інформацію та соціальні мережі для розміщення новин, але для швидкого і більш зручного способу пошуку користувачем інформації та взаємодії з нею краще підійде чат-бот.

Тому, під час розробки необхідно, щоб майбутній чат-бот НЗ вирішував наступні задачі:

— забезпечення студентів та майбутніх вступників інформацією про НЗ (сайт НЗ, інформація для абітурієнтів

— можливість перегляду інформації про умови вступу;

— забезпечення зворотного зв'язку;

- забезпечення надійності, безпеки та цілісності даних;
- можливість розсилки через бот інформації та матеріалів;
- можливість проведення опитування серед користувачів;
- завантажування файлів та їх публікація.

Чат-бот повинен мати можливість працювати на всіх версіях Telegram та всіх пристроях, які його підтримують. Повністю відповідати поставленим вимогам, тобто виконувати всі вищезазначені функції. Чат-бот повинен мати зручний інтерфейс, мінімум тексту та найкориснішу інформацію. Користувачі повинні швидко зрозуміти, як користуватися ботом, тобто інтерфейс чат-бота має бути зрозумілим та простим.

Інформація, надана приймальною комісією, повинна бути точною та відповідною.

Для користування ботом необхідний будь-який пристрій, на якому встановлена програма Telegram і наданий доступ до мережі Інтернет.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМУ РОБОТИ ЧАТ-БОТА ДЛЯ ОРІЄНТУВАННЯ АБИТУРІЄНТІВ

2.1 Проектування функціональних можливостей чат-бота

Чат-бот розробляється на платформі Telegram, тому необхідно більш детально розглянути саме TelegramAPI, що являє собою інтерфейс на основі HTTP для розробки ботів на цій платформі. За допомогою цього API додаток зв'язується із сервером. Важливою перевагою TelegramAPI є те, що він має повністю відкритий код і це дає можливість взаємодіяти з платформою ботів. Також, за допомогою TelegramAPI можна керувати створеними ботами у Telegram, наприклад отримувати та відправляти повідомлення

Розроблюваний в даній бакалаврській дипломній роботі чат-бот, має наступні функціональні можливості:

- взаємодія з чат-ботом за допомогою кнопок та навігація за допомогою головного меню;
- відображення інформації необхідної для користувача при виконанні запитів;
- інтуїтивно зрозумілий та приємний інтерфейс;
- робота на всіх пристроях, що мають доступ до Інтернету та підтримують месенджер Telegram;
- швидке надання відповідей на запити користувача.

Робота з чат-ботом починається з команди /start і перше, що побачить користувач — це кнопки головного меню.

Кожний розділ з меню відповідає за різні функції, розглянемо їх:

- пункт «Сайт коледжу» забезпечує перегляд сайту за url;
- пункт «Приймальна комісія» включає в себе перелік підрозділів;
- пункт «Після 9-го класу» надає інформацію про етапи вступу, перелік документів, можливість поставити питання адміністратору та перехід в головне меню;

— пункт «Після 11-го класу» надає інформацію про етапи вступу, перелік документів, можливість поставити питання адміністратору та перехід в головне меню;

— пункт «Уже маю спеціальність» надає інформацію про етапи вступу, перелік документів;

— пункт «Написати Адміністратору» переводить в повідомлення адміністратора;

— перехід в головне меню «Головне меню»;

— пункт «Обрати спеціальність» надає доступ до 9 спеціальностей, з описом про освітню програму, спеціальність та галузь;

— пункт «Дні відкритих дверей» дозволяє завантажити фото та коротку інформацію про вступ;

— пункт «Підготовчі курси для вступників» відображає основні дати проведення підготовчих курсів;

— пункт «Вартість навчання» надає доступ про вартість навчання за спеціальностями.

2.2 Розробка блок-схеми алгоритму роботи чат-бота орієнтування абітурієнтів

Взаємодія з більшістю telegram-ботами виконується за такою схемою: спочатку користувач відправляє певну команду (повідомлення або необхідний запит чат-ботові), далі те, що користувач відправляє, потрапляє до програмного забезпечення, яке працює на сервері розробника. Після отримання відповіді від сервера, повідомлення відображаються на екрані для користувача.

Схема взаємодії користувача та telegram-бота зображена на рисунку 2.1.

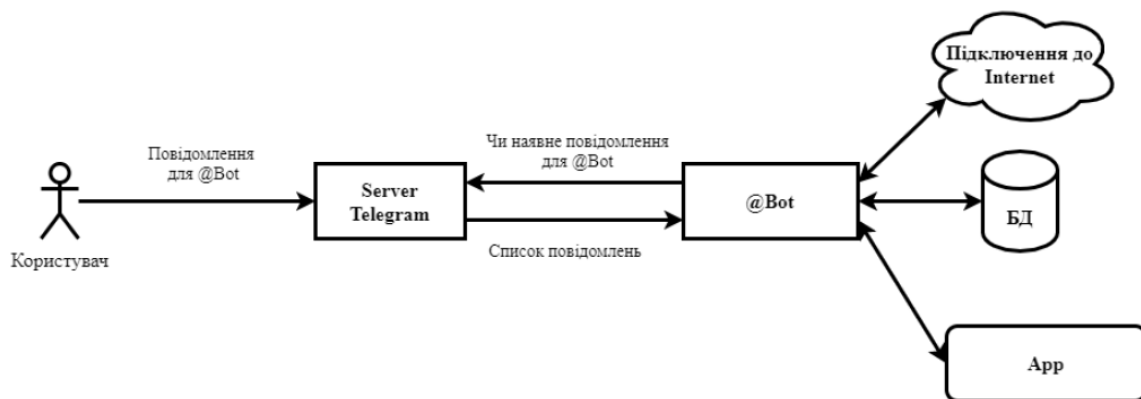


Рисунок 2.1 — Схема взаємодії користувача та telegram-бота

Блок-схема алгоритму роботи чат-бота представлена на рисунку 2.2.

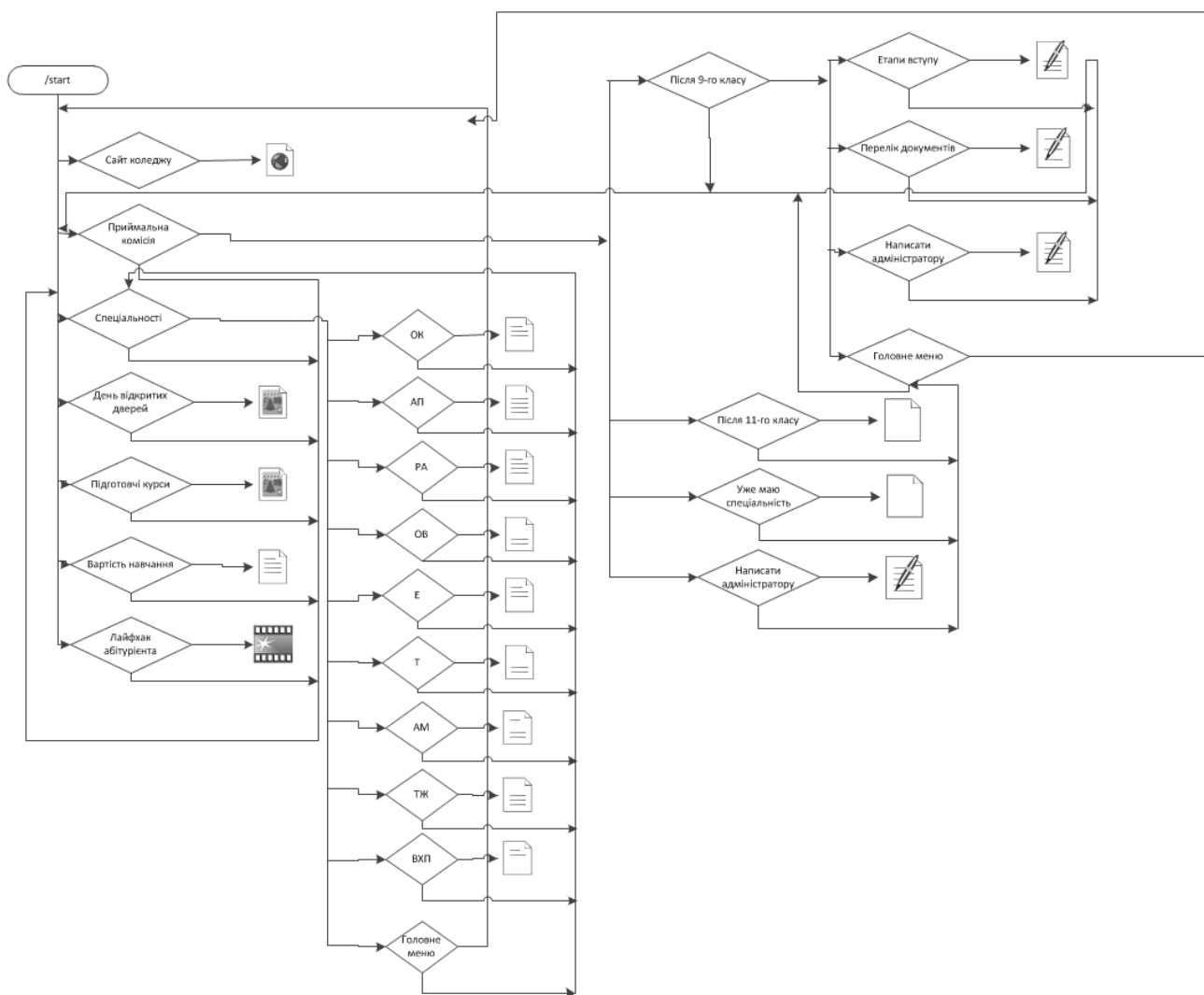


Рисунок 2.2 — Блок-схема алгоритму чат-бота «Помічник вступника»

2.3 Загальний алгоритм реалізації проєкта

Перед написанням коду необхідно вивчити всю наявну документацію по темі, навчальні матеріали та корисні статті. Після навчання слід написати основні вимоги до бота. Далі необхідно здійснити підключення основних функцій бота. При коректній роботі програми починається період тестування функцій і, у разі необхідності, додавання нових. Коли бот буде повністю працездатним і протестованим, розпочинається етап апробації бота. Описаний алгоритм представлено в Таблиці 2.1.

Таблиця 2.1 — Загальний алгоритм реалізації проєкта

Етапи	Дії
1	Вивчення матеріалів. Розробка невеликих тестових додатків для розуміння логіки роботи API.
2	Нарощування функціоналу програми та підготовка до підключення TelegramBotAPI.
3	Створення та налаштування бота всередині Telegram для отримання унікального ключа доступу та початок створення логіки програми для адекватної роботи в рамках TelegramBotAPI.
4	Тестування та нарощування функціоналу для задовільного результату.
5	Доробка дрібних недоліків та тестування працездатності.
6	Апробація чат-бота.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЧАТ-БОТА

3.1 Варіантний аналіз і обґрунтування вибору програмних засобів

Перед початком розробки програмних засобів необхідно провести огляд доступних програмних засобів та відібрати такі, що найкраще підходять для розробки даної програми.

Для написання Telegram ботів можна обрати одну із сучасних мов програмування, таких як Python, JavaScript, Java та C#. Проведемо їх огляд.

3.1.1 Python

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована. Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Те, що це інтерпретована мова з динамічною обробкою типів, робить її ідеальною для написання скриптів та швидкої розробки прикладних програм.

Для написання чат-ботів мовою Python розроблено бібліотеки, що надають готовий функціонал та інструментарій:

- python-telegram-bot — потужна бібліотека, яка активно розвивається
- PyTelegramBotAPI — альтернативна популярна бібліотека
- telebot — невелика бібліотека, підходить для розгортання на Google App Engine.

3.1.2 Java

Java — мова програмування високого рівня, випущена компанією «Sun Microsystems» у 1995 році. Найактуальніша версія мови Java Standard Edition 10 вийшла 20 березня 2018 року. Має С-подібний синтаксис. Підтримує парадигму об'єктно-орієнтованого програмування. Має дещо гіршу швидкодію у порівнянні з мовами С++ та С#. Характеризується наявністю великої кількості сторонніх бібліотек.

Мова Java використовує автоматичний збирач сміття для керування пам'яттю під час життєвого циклу об'єкта. Програміст вирішує, коли створювати об'єкти, а віртуальна машина відповідальна за звільнення пам'яті після того, як об'єкт стає непотрібним. Коли до певного об'єкта вже не залишається посилань, збирач сміття може автоматично прибирати його із пам'яті.

Для створення ботів мовою Java використовується бібліотека TelegramBots, яка є проста у використанні та підтримує вбудований режим.

3.1.3 С#

С# — мова програмування високого рівня, розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде в корпорації Microsoft. Перша версія мови була випущена у 2002 році. Остання актуальна версія мови С# 7.0 вийшла 7 березня 2017 року.

Мова програмування С# була розроблена під впливом С++ та Java, має С-подібний синтаксис. Підтримує парадигму об'єктно-орієнтованого програмування. Має дещо нижчу швидкість роботи, ніж мова С++, проте вищу, ніж Java. Виділяється наявністю великої кількості сторонніх бібліотек, автоматичного регулювання пам'яті та вбудованою реалізацією деяких шаблонів проектування.

Telegram.Bot — бібліотека для створення ботів мовою С#, підтримує вбудований режим.

3.1.4 JavaScript

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП.

JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.[17]

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
- функції як об'єкти першого класу
- обробка винятків
- автоматичне приведення типів
- автоматичне прибирання сміття

JavaScript є стандартом розробки інтерфейсу користувача на даний момент. Крім того JavaScript широко використовується для розробки чат-ботів з використанням модуля для Node.js — `node-telegram-bot-api`.

3.1.5 Обґрунтування вибору мови програмування

Із наведених вище варіантів була обрана бібліотека `python-telegram-bot`: вона в повній мірі дає реалізувати функціонал боту, будучи при цьому максимально наближеною до Telegram Bot API і такою, що розвивається. Відповідно до цього мова розробки проекту — Python. Мова Python найкраще підходить для створення серверних програм. Крім того, враховуючи і те, що на сьогоднішній день вона є найбільш популярною та вживаною.

Вибір обґрунтовано потужністю мови програмування та зручністю роботи з бібліотекою `python-telegram-bot`.

3.2 Вибір середовища розробки

Для більш зручної розробки програмних засобів використовуються IDE (інтегровані середовища розробки). Інтегроване середовище розробки — це комплекс програмних засобів для розробки, що складається з текстового редактора коду, інструментів для компіляції та відлагодження проєктів.

Для порівняння середовищ розробки для подальшого вибору буде розглянуто середовища PyCharm та Visual Studio Code.

3.2.1 Інтегроване середовище PyCharm

PyCharm — це інтегроване середовище розробки для Python, яке має повний комплект засобів, необхідних для ефективного програмування на Python.

Вбудований відладчик допомагає при налагодженні та тестуванні. Можна використовувати командний рядок для роботи в середовищі. PyCharm також допоможе виявити та оптимізувати неоптимальний програмний код, який виконується дуже довго. Є можливість використовувати віддалений інтерпретатор. Є інтегрований інструмент, що використовується для створення віртуального середовища і його налаштування для розробки Vagrant та розвитку в середовищі, що підтримує контейнерезацію Docker.

PyCharm підтримує модульне тестування, ви можете перевірити, чи правильно працюють код, модулі та різні частини програми.

Під час створення доступу до бази даних середовище розробки надає інструменти для створення запитів SQL до Oracle, SQL Server, PostgreSQL, MySQL та інших видів баз даних. Ви можете писати код SQL та редагувати його вид та переглядати дані на які прийшов запит за його допомогою.

Переваги:

— наявність редактору коду зі всіма корисними функціями: підсвічування синтаксису, автоматичне форматування тощо;

- PyCharm дозволяє перевірити версію інтерпретатора мови на сумісність, а також використовувати шаблони коду;
- можливість переглядати документацію прямо у вікні редактора (для елементів) чи в браузері (для зовнішньої документації);
- можливість швидко проводити рефакторинг коду;
- утиліта підтримує всі свіжі версії Django, а також IronPython, Jython, Cython.

3.2.2 Середовище Visual Studio Code

Visual Studio Code — текстовий редактор призначений для розробки застосунків для хмарних систем та веб-додатків. Visual Studio Code розповсюджується безкоштовно і є кросплатформним, тобто підтримує всі сучасні операційні системи.

VS Code підтримує різноманітні плагіни, які з легістю можна інтегрувати в нього, він найкраще підходить для написання коду інтерпретованими мовами програмування.

В результаті аналізу було обрано середовище Visual Studio Code через свою доступність та підтримку необхідного функціоналу для реалізації поставленого завдання.

3.3 Розробка функцій чат-бота

3.3.1 Реєстрація чат-бота для Telegram Bot Api

Telegram Bot API — це HTTP-інтерфейс для роботи з ботами у Telegram.

По суті бот — це спеціальний обліковий запис, створений для автоматичної відправки та обробки повідомлень.

Для роботи з TelegramBotAPI вивчено документацію, в якій описані всі методи і параметри, що передаються.

Існує два протилежних за логікою способи отримання оновлень від бота:

- `getUpdates` (використовуючи `long polling`) використовується для отримання оновлень через `long polling`;
- `setWebhook` необхідний для задання URL вебхука, на який бот відправлятиме оновлення.

Логіка `getUpdates` полягає у періодичному опитуванні серверів Telegram на предмет наявності нової інформації. З'єднання відкривається на нетривалий час і всі оновлення одразу відправляються боту. Відповідь повертається у вигляді масиву об'єктів `Update`. Параметри методу наведені у таблиці 3.1.

Таблиця 3.1 — Параметри методу `getUpdates`

Параметри	Тип	Обов'язковий	Опис
<code>offset</code>	Integer	Ні	Дозволяє переглядати нові повідомлення, ігноруючи старі.
<code>limit</code>	Integer	Ні	Обмежує кількість елементів масиву <code>Update</code> . Приймає значення від 1 до 100.
<code>timeout</code>	Integer	Ні	Таймаут в секундах для <code>long polling</code> . За замовчуванням 0, тобто короткий запит.

Мається на увазі, що тепер якщо в чат приходять повідомлення, то Telegram сам говорить про це. Кожний раз при отриманні оновлення на цю адресу відправлятиметься HTTPS POST із JSON-об'єктом `Update`. При невдалому запиті до сервера спроба повториться декілька разів. Для більшої безпеки рекомендується включити токен в URL вебхука, наприклад, наступним чином: `https://yourwebhookserver.com/ <token>`. Для того щоб отримати токен (власний ключ авторизації) необхідно написати спеціальному боту `@BotFather`. Оскільки

ніхто сторонній не знає особистий токен користувача, запити до вебхуку посилає саме Telegram. Параметри методу наведені в таблиці 3.2.

Таблиця 3.2 — Параметри методу setWebhook

Параметри	Тип	Обов'язковий	Опис
url	String	Ні	HTTP url для відправки запитів. Щоб видалили вебхук, треба відправити порожній рядок.
certificate	InputFile	Ні	Завантаження публічного ключа для перевірки кореневого сертифікату

Принцип взаємодії чат-бота та користувача зображено на рисунку 3.1.

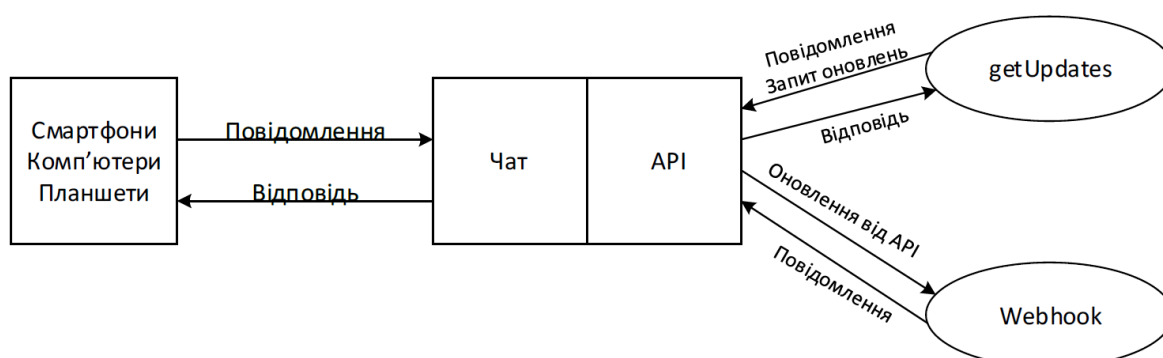


Рисунок 3.1 — Принцип роботи чат-бота на платформі Telegram

Приклади інших доступних для TelegramBotAPI методів описані нижче:

- sendMessage — метод для відправки текстових повідомлень;
- sendLocation — метод для відправки точки на карті;
- editMessageText — метод для редагування текстових повідомлень, відправлених ботом чи за допомогою бота;
- sendAudio — метод для відправки аудіофайлів, якщо бажаєте, щоб клієнти Telegram відображали їх в музичному плеєрі;
- sendContact — метод для відправки телефонних контактів.

Допускаються POST і GET запити. Існує 4 способи для передачі параметрів в Bot API:

- запит в URL;
- application / x-www-form-urlencoded;
- application / json (не прийнятний для завантаження файлів);
- multipart / form-data (для завантаження файлів).

Для реєстрації чат-бота, що розробляється, у Telegram існує головний бот «BotFather». Він призначений для створення нових ботів та керування ними. Його дуже просто знайти, використовуючи пошук Telegram. Після початку роботи з ним за допомогою команди «/start» зі списку команд потрібно обрати «/newbot», що передбачає створення власного бота. Після цього варто ввести назву, яка міститиме в собі слова «Bot» чи «_bot». Якщо назву введено правильно, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) та URL-адресу для доступу до чат-бота.

Процес реєстрації чат-бота наведено на рисунку 3.2.

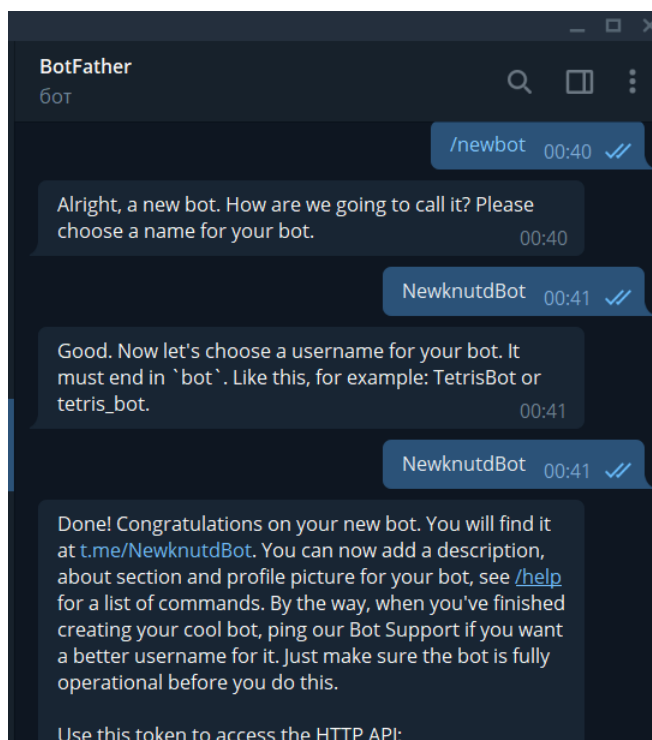


Рисунок 3.2 — Реєстрація бота у Telegram

Для редагування параметрів чат-бота використовується ряд команд (таблиця 3.3), наприклад, для зміни зображення або опису.

Таблиця 3.3 — Параметри чат-бота

Команда	Опис
/setname	Змінює існуюче ім'я
/setdescription	Встановлює текст, який відображається при першому відкритті чат-бота
/setabouttext	Встановлює текст у поле «Про чат-бот»
/setuserpic	Встановлює обране зображення
/setcommands	Дозволяє створити список доступних команд
/deletebot	Видаляє обраного чат-бота

Після реєстрації чат-бота отримується токен бота для керування від Telegram.

Токен — це спеціальний ключ від бота, за допомогою якого здійснюється підключення його до сторонніх сервісів.

3.3 Реалізація програмної частини

Проект складається з файлів, які наведені на рисунку 3.3, що утворюють програму реалізацію чат-бота на мові програмування Python.

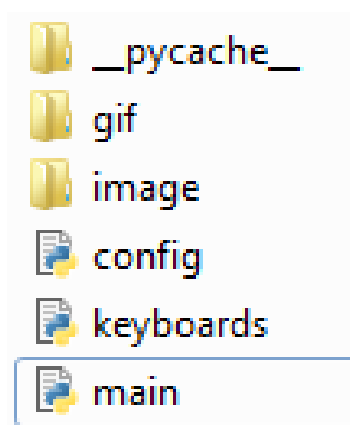


Рисунок 3.3 — Файлова структура чат-бота

Розглянемо кожен файл проекту детальніше:

- config.py — містить інформацію з токеном бота та ідентифікатор адміністратора;
- database.db — файл бази даних;
- keyboards.py — файл в якому створені та описані кнопки чат-бота;
- main.py — головний файл програми, в якому прописана логіка роботи.

Спочатку необхідно підключити до проекту всі необхідні бібліотеки та модулі для розробки (рисунок 3.4).

```
import telebot
import config
import datetime
import pytz
import json
import traceback
```

Рисунок 3.4 — Підключення бібліотек та модулів до проекту

3.3.1. Створення першого повідомлення та інтерфейсу чат-бота

Для початку роботи чат-бота потрібно відправити йому команду «/start», що буде першою командою боту для користувача після вибору діалогу з ним. Команді «/start» відповідає кнопка «Розпочати», що з'являється при переході користувачем в чат-бот (рисунок 3.5).

Щоб бот сприймав команду «/start» та починав працювати після неї, створено декоратор.

Декоратор — це функція, яка дозволяє змінити функціональність іншої функції у деякому напрямку, при цьому не змінюючи її код. Приклад декоратора можна побачити на рисунку 3.6. Він виступає в ролі «слухача», який обробляє вхідні повідомлення.

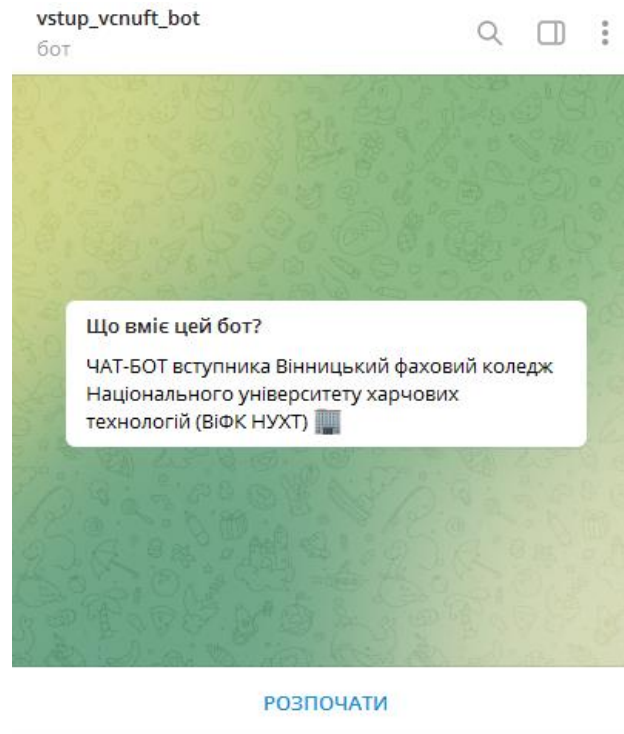


Рисунок 3.5 — Інтерфейс чат-бота при першому вході

```
@bot.message_handler(commands=['start'])
```

Рисунок 3.6 — Декоратор для початку роботи бота

За допомогою методу `send_message` помічник відправить у чат перше повідомлення з описом чат-бота та переліком пунктів головного меню (рисунок 3.7).

```
bot.send_message(
    message.chat.id,
    "Я помічник Вінницького Фахового коледжу "+
    "Національного університету харчових технологій (ВіФК НУХТ) 🏢 \n" +
    "\n"+
    "Я допоможу тобі знайти всю необхідну інформацію щодо вступу "+
    "на освітні програми ВіФК НУХТ 📖📄"+
    " " )
```

Рисунок 3.7 — Метод `send_message`

Параметри методу ReplyKeyboardMarkup для створення клавіатури вказують на ширину рядків (row_width) та на те, що клавіатура змінюватиме свій розмір в залежності від кількості кнопок для кращого її сприйняття користувачем (resize_keyboard=True). На рисунку 3.8 неведено програмний код для створення клавіатури «Головне меню».

```
keyboard = telebot.types.InlineKeyboardMarkup()
keyboard.row(
    telebot.types.InlineKeyboardButton("Сайт коледжу", url="http://www.vcnuft.vn.ua/index.php"),
    telebot.types.InlineKeyboardButton("Приймальна комісія 📄", callback_data='prujmalna_komisija'))
keyboard.row(
    telebot.types.InlineKeyboardButton("Обрати спеціальність 🎓", callback_data='choose_profession'),
    telebot.types.InlineKeyboardButton("Дні відкритих дверей 🚪", callback_data='open_days'))
keyboard.row(
    telebot.types.InlineKeyboardButton("Підготовчі курси для вступників 📖", callback_data='courses'),
    telebot.types.InlineKeyboardButton("Вартість навчання 💰", callback_data='pay'))
```

Рисунок 3.8 — Метод ReplyKeyboardMarkup

3.3.2. Розробка головного меню та його складових

Після запуску чат-бота користувач побачить привітання , в якому з'явиться ім'я користувача, анімований робот та «Головне меню» з наступними кнопками:

- «Сайт коледжу»;
- «Приймальна комісія»;
- «Обрати спеціальність»;
- «Дні відкритих дверей»;
- «Підготовчі курси для вступників»;
- «Вартість навчання».

Вигляд першого стартового повідомлення зображено на рисунку 3.9.

Кожен підрозділ головного меню містить інформацію на відповідну тематику і після вибору того чи іншого пункту з'являється перелік інформації, яка вміщується там.

Розглянемо кожен підрозділ головного меню більш детально.

Пункт меню «Сайт коледжу» забезпечує перехід на сайт коледжу для отримання більш повної інформації та представлення про навчальний заклад. Пункт меню «Сайт коледжу» реалізовано за допомогою запиту URL:

```
telebot.types.InlineKeyboardButton("Сайт коледжу",
url="http://www.vcnuft.vn.ua/index.php")
```



Рисунок 3.9 — Вигляд першого стартового повідомлення

На рисунку 3.10 наведено перелік інформації пункту меню «Приймальна комісія».

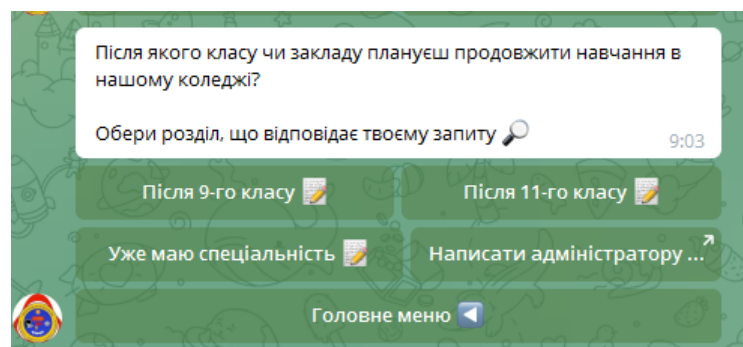


Рисунок 3.10 — Інформація пункту меню «Приймальна комісія»

Пункт меню «Приймальна комісія» реалізує функція `prujmalna_komisija_command`:

```
@bot.message_handler(commands=['prujmalna_komisija']) #Приймальна
комісія
def prujmalna_komisija_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Після 9-го класу 📖",
callback_data='nine'),
        telebot.types.InlineKeyboardButton("Після 11-го класу 📖",
callback_data='eleven'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Уже маю спеціальність 📖",
callback_data='okr'),
        telebot.types.InlineKeyboardButton("Написати адміністратору 📖",
url='telegram.me/galina_09_09'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Обрати спеціальність 📖",
callback_data='specialty'))
    bot.send_message(
        message.chat.id,
        'Після якого класу чи закладу плануєш продовжити навчання в нашому
коледжі?\n' +
        '\n'+
        'Обери розділ, що відповідає твоєму запити 📖 ',
        reply_markup=keyboard
    )
```

На рисунку 3.11 наведено зображення пункту меню «Обрати спеціальність».

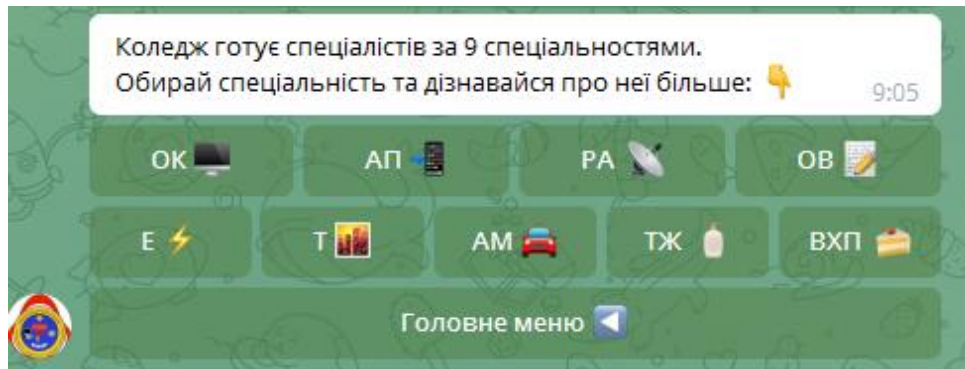


Рисунок 3.11 — Інформація пункту меню «Обрати спеціальність»

Пункт меню «Обрати спеціальність» реалізує функція `choose_profession`:
`@bot.callback_query_handler(func=lambda call: call.data=='choose_profession')`

`#Обрати спеціальність`

```
def choose_profession_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("ОК 🖥️", callback_data='ok'),
        telebot.types.InlineKeyboardButton("АП 📱", callback_data='ap'),
        telebot.types.InlineKeyboardButton("РА 📡", callback_data='ra'),
        telebot.types.InlineKeyboardButton("ОВ 📄", callback_data='ov'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Е ⚡", callback_data='e'),
        telebot.types.InlineKeyboardButton("Т 🏭", callback_data='t'),
        telebot.types.InlineKeyboardButton("АМ 🚗", callback_data='am'),
        telebot.types.InlineKeyboardButton("ТЖ 🥛", callback_data='tj'),
        telebot.types.InlineKeyboardButton("ВХП 📁", callback_data='vxp'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Головне меню ⬅️", callback_data='main_menu'))

    bot.send_message(message.chat_id, "Коледж готує спеціалістів за 9 спеціальностями.  

    Обирай спеціальність та дізнавайся про неї більше:", reply_markup=keyboard)
```

message.chat.id,

"Коледж готує спеціалістів за 9 спеціальностями. \n" +

"Обирай спеціальність та дізнавайся про неї більше: 🖱️\n",

reply_markup=keyboard

)

На рисунку 3.12 зображений результат відображення пункту меню «Дні відкритих дверей».

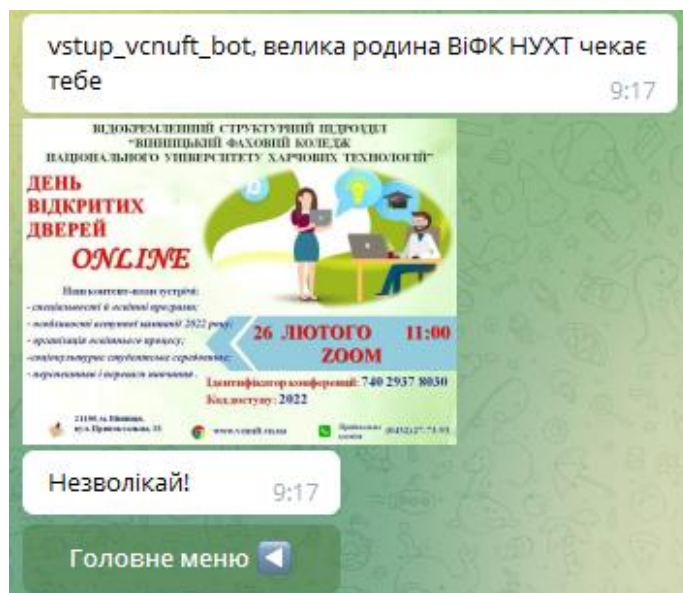


Рисунок 3.12 — Інформація пункту меню «Дні відкритих дверей»

Пункт меню «Дні відкритих дверей» реалізує функція `open_days`:

```
@bot.callback_query_handler(func=lambda call: call.data=='open_days')
```

```
def open_days_command(message):
```

```
    keyboard = telebot.types.InlineKeyboardMarkup()
```

```
    keyboard.row(
```

```
        #telebot.types.InlineKeyboardButton('Назад 🏠', callback_data='back-3'),
```

```
)
```

```

username = message.from_user.first_name
bot.send_message(
    message.chat.id,
    username + ", велика родина ВіФК НУХТ чекає тебе
\n".format(message.from_user, bot.get_me()),
    parse_mode='html')
sti = open('D:/Python/Bot/image/opendays.jpg', 'rb')
bot.send_sticker(message.chat.id, sti)
bot.send_message(
    message.chat.id, "Незволікай!",
    reply_markup=keyboard)

```

Аналогічним чином реалізовані інші функції та методи чат-бота. Для реалізації чат-бота створено 29 функцій. Вигляд структури проєкту наведено на рисунку 3.13, які становлять структуру програмного проєкту.

Розбивка проєкту на окремі блоки, які є самостійними об'єктами, становить структуру проєкту і являє собою сукупність взаємопов'язаних елементів і процесів проєкту, які представлені з різним ступенем деталізації. В термінах управління проєктами структура проєкту являє собою «дерево» орієнтованих на продукт компонентів.

Структуризація проєкту спрощує процес розробки і відповідно тестування програмного продукту.

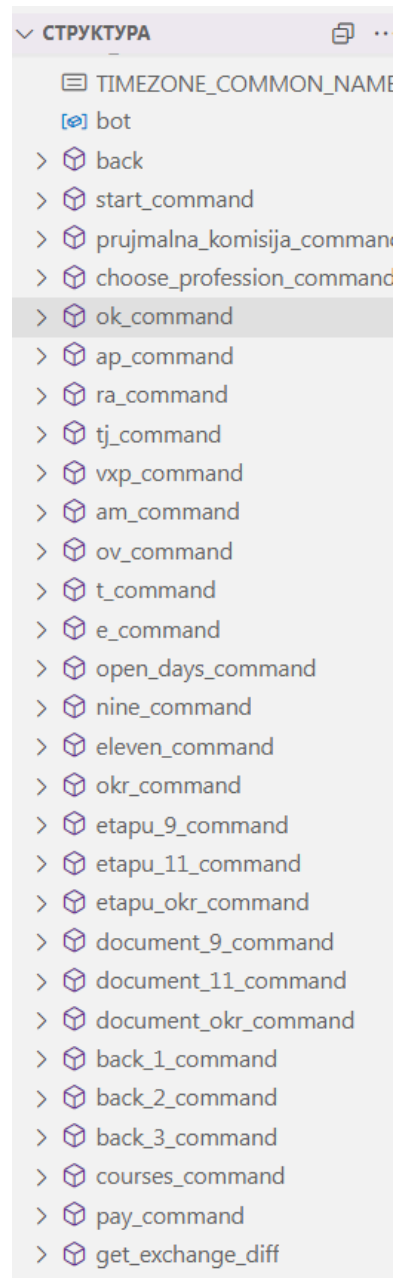


Рисунок 3.13 — Структура проекту

3.4 Тестування програмного модуля

Тестування — це процес аналізу та дослідження, який надає змогу виявити інформацію про якість продукту відносно умов, в яких він буде застосовуватись. Методика тестування також включає в себе процес пошуку дефектів, помилок, несправностей. Також це є випробуванням програмних складових з метою оцінити

готовність програмного продукту до використання. Результат тестування оцінюється за наступними критеріями:

- відповідність вимогам, які надавалися розробниками та проєктувальниками;
- відповідність вихідних даних;
- прийнятний час виконання функцій;
- практичність;
- відповідність вимогам замовника.

Кількість тестів навіть для простих програмних компонентів може бути ледь не нескінченним, тому тактика тестування має полягати в тому, що будуть проведені тільки необхідні тести з урахуванням доступного часу та ресурсів. Як результат, програмні засоби тестуються стандартним виконанням програми з метою виявлення багів, помилок або інших дефектів.

Існує багато видів тестування: одні зазвичай виконують самі розробники, а інші — спеціалізовані групи. В нашому ж випадку буде використовуватись тестування чат-бота.

Тестування чат-бота — це перевірка роботи чат-бота на різних платформах.

Тестування є одним з найголовніших аспектів в розробці програмного забезпечення.

Є дві основних методики тестування: тестування «білої скриньки» та тестування «чорної скриньки».

Методика «білої скриньки». Об'єктом тестування тут є не зовнішня, а внутрішня поведінка програми. Перевіряється коректність побудови всіх елементів програми та правильність їхньої взаємодії один з одним. Зазвичай аналізуються керуючі зв'язки елементів, рідше — інформаційні зв'язки. Тестування за принципом «білої скриньки» характеризується ступенем, в якому тести виконують або покривають логіку (вихідний текст) програми.

Одним із способів вивчення поставленого питання є дослідження методики «чорної скриньки», Основне місце програми тестів «чорної скриньки» це інтерфейс ПЗ. Ці тести демонструють:

- як виконуються функції програми;
- як приймаються вихідні дані;
- як виробляються результати;
- як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе. Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10¹⁰. Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування програмних засобів буде доречно проводити використовуючи методику «чорної скриньки». Вона базується на використанні шаблонів тестування, або ж тест-кейсів. Для тестування чат-бота орієнтування абітурієнтів буде створено декілька ситуацій у яких перевіряється працездатність чат-бота, коректності основних функцій та кросплатформне та міжплатформне тестування. Кросплатформне тестування проводиться для визначення поведінки чат-бота в різних середовищах. Міжплатформне тестування допомагає у виявленні проблем, які можуть відрізнятись в залежності від платформи або конфігурації, таких як узгодженість, користувацький інтерфейс, проблеми з юзабіліті та продуктивність.

Розроблені тест-кейси описано в таблиці 3.4.

Тестування програми проводилось на кожному етапі розробки (unit тести), та проводився інтеграційний тест наприкінці розробки чат-бота.

Під час тестування не виявлено критичних багів, які б впливали на роботу чат-бота. Це значить, що розроблений чат-бот орієнтування абітурієнтів відповідає вимогам технічного завдання та вимогам технології чат-ботів.

Таблиця 3.4 — Тест-кейси

Код тест-кейса	Опис тест-кейса		
	Хід тестування		Очікуваний результат
	Дата тестування	Результат	Примітка
001	Перевірка роботи пунктів меню		
	1. Запустити чат-бот 2. Перевірити роботу кожного пункту меню.		Коректне виведення інформації
	17.05.2022	Пройдено	-
002	Завантаження файлів		
	1. Вибрати команду 2. Завантажити файл 3. Перевірити відповідність завантаженого файлу		Команда відповідає файлам завантаження. Файли відповідають формату.
	17.05.2022	Пройдено	-
003	Перевірка запуску сайту		
	1. Вибрати команду 2. Завантажити сторінку 3. Перевірити відповідність		Сайт завантажується коректно, з головної сторінки в браузері встановленому за замовчуванням
	18.05.2022	Пройдено	-
004	Перевірка зворотного зв'язку з адміністратором		
	1. Вибрати команду «Написати адміністратору» 2. Написати повідомлення 3. Отримати відповідь		Команда забезпечує перехід в чат з адміністратором
	18.05.2022	Пройдено	-
	Перевірка роботи на різних платформах		
	1. Перевірити на платформах Windows, Android, IOS		Робота і відображення результатів не залежить від платформи
	18.05.2022	Пройдено	

3.5 Інструкція користувача

Для початку роботи потрібно запустити додаток через месенджер Telegram. Перед користувачем буде інтуїтивно зрозумілий інтерфейс, який передбачає вибір кнопок для отримання інформації про вступ до коледжу для орієнтування абітурієнтів.

ВИСНОВКИ

У ході виконання бакалаврської дипломної роботи було проведено розгляд розробки чат-ботів, проаналізовані види ботів та технології розробки. Вибір мови програмування та середовища розробки було аргументовано тим, що мова Python забезпечує розробку серверних програм та містить спеціальні модулі для роботи з чат-ботами: TelegramBotAPI. Вибір середовища Visual Studio Code обґрунтовано легкістю та доступністю створення програмних файлів, особливостями перегляду та редагування коду.

В бакалаврській роботі було розроблено вимоги до чат-боту для реалізації поставлених завдань, розроблено структуру проєкту, блок-схему алгоритма роботи та програмно реалізовано програмний додаток. В пояснювальні записці описано програмну реалізацію чат-бота, який надає абітурієнту можливість отримати інформацію щодо вступу за запитом. Було розглянуто основні методики тестування, виконано тестування методикою «чорної скриньки», складено перелік тест-кейсів для тестування, а також розроблено інструкцію користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Топ-10 корисних Telegram-ботів для українських користувачів: URL: <https://www.epravda.com.ua/publications/2017/08/7/627822> (дата звернення 10.04.2022)
2. Telegram Bot API: URL: <https://core.telegram.org/bots/api> (дата звернення 15.04.2022)
3. Марк Лутц, Learning Python :2019, 720с.
4. Telegram FAQ: URL: <https://telegram.org/faq> (дата звернення 20.04.2022)
5. Telegram-bot: URL: <https://habr.com/ru/post/442800/> (дата звернення 22.04.2022)
6. Телеграм бот за допомогою TeleBot: URL: <https://habr.com/ru/post/448310/> (дата звернення 22.03.2022)
7. Git: URL: <https://uk.wikipedia.org/wiki/Git> (дата звернення 05.04.2022)
8. Python Requests : URL : <https://2.python-requests.org/en/master/> (дата звернення 10.04.2022)
9. Огляд протоколу HTTP: URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> (дата звернення 13.04.2022)
10. Черга повідомлень : URL: <https://aws.amazon.com/ru/message-queue/> (дата звернення 13.04.2022)
11. What is REST: URL: <https://restfulapi.net/> (дата звернення 15.04.2022)
12. REST : URL: <https://habr.com/ru/post/351890/> (дата звернення 19.04.2022)
13. JSON: URL: <https://www.json.org/json-ru.html> (дата звернення 26.04.2022)

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н., Азаров О.Д.

«28» квітня 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи

«Телеграм чат-бот для орієнтування абітурієнтів»

08-23.БДР.003.00.000 ТЗ

Науковий керівник: доцент к.т.н.

_____ Черняк О.І.

Студент групи КІ-20мсЗ

_____ Гребінь О.О.

1 Підстави для виконання бакалаврської дипломної роботи

1.1 Актуальність розробки, яка полягає у необхідності інформування абітурієнтів про умови та етапи вступу в автоматичному та віддаленому форматі, які дозволять зменшити навантаження на працівників приймальної комісії та полегшить пошук інформації абітурієнтам;

1.2 Наказ про затвердження теми бакалаврської дипломної роботи.

2 Мета БДР і призначення розробки

2.1 Мета роботи — є розробка чат-бота для популярного месенджера Telegram мовою програмування Python в середовищі Visual Studio Code, який допоможе абітурієнтам зорієнтуватися щодо вступу в навчальний заклад.

2.2 Призначення розробки — впровадження та застосування розробки в навчального закладу для інформування абітурієнтів.

3 Вихідні дані для виконання БДР наступні:

3.1 Проведення аналізу технологій розробки чат-ботів та аналогів чат-ботів;

3.2 Розробка структури та алгоритму роботи чат-бота;

3.3 Проведення аналізу та вибору засобів для реалізації чат-бота та написання програмного коду чат-бота;

3.4 Проведення тестування та апробації чат-бота

4 Вимога до виконання БДР

Створення чат-бота з дотриманням вимог розробки API-додатків для роботи на платформі Telegram.

5 Етапи БДР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи БДР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Постановка задачі роботи			Вступ
2	Огляд і аналіз предметної області, технологій створення та застосування чат-ботів			Розділ 1
3	Структурне проєктування чат-боту орієнтування абітурієнтів			Розділ 2
4	Розробка блок-схеми алгоритму роботи чат-бота та загального алгоритму реалізації проєкту			Розділ 2
5	Реєстрація чат-бота для TelegramBotApi та розробка методів та функцій чат-бота для взаємодії з користувачем			Розділ 3
6	Апробація та впровадження результатів дослідження			Тези доповідей
7	Оформлення пояснювальної записки та графічного матеріалу			Пояснювальна записка
8	Підготовка супроводжуючих документів, їх підписування, проходження нормоконтролю та тесту на плагіат			Оформлені документи

6 Матеріали, що подаються до захисту БДР

До захисту подаються: пояснювальна записка БДР, графічні і ілюстративні матеріали, протокол попереднього захисту БДР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до БДР українською та іноземною мовами, довідка про відповідність оформлення БДР діючим вимогам.

7 Порядок контролю виконання та захисту БДР

Виконання етапів графічної та розрахункової документації ДР контролюється науковим керівником згідно зі встановленими термінами. Захист

ДР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення БДР

8.1 При оформлюванні БДР використовуються:

— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до оформлення дипломних проектів (робіт) для студентів всіх спеціальностей /Уклад. В.В. Кухарчук, О.Г. Ігнатенко, Р.Р.Обертюх, — Вінниця: ВДТУ, 2002. —55 с.;

— Документами на які посилаються у вище вказаних.

Технічне завдання до виконання отримав _____ Гребінь О.О.

ДОДАТОК Б

Взаємодія користувача та telegram-бота

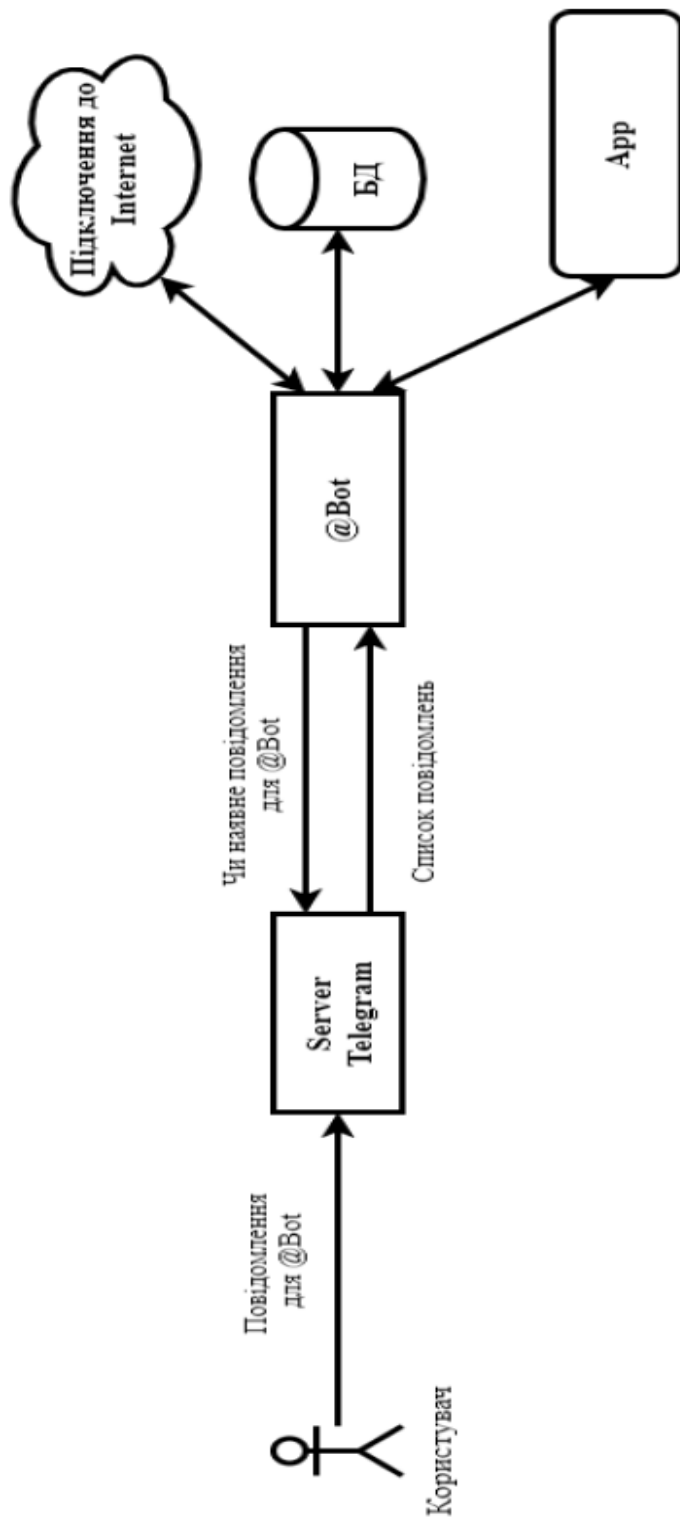


Рисунок Б.1 — Схема взаємодії користувача з телеграм чат-ботом

ДОДАТОК В

Алгоритм роботи чат-бота

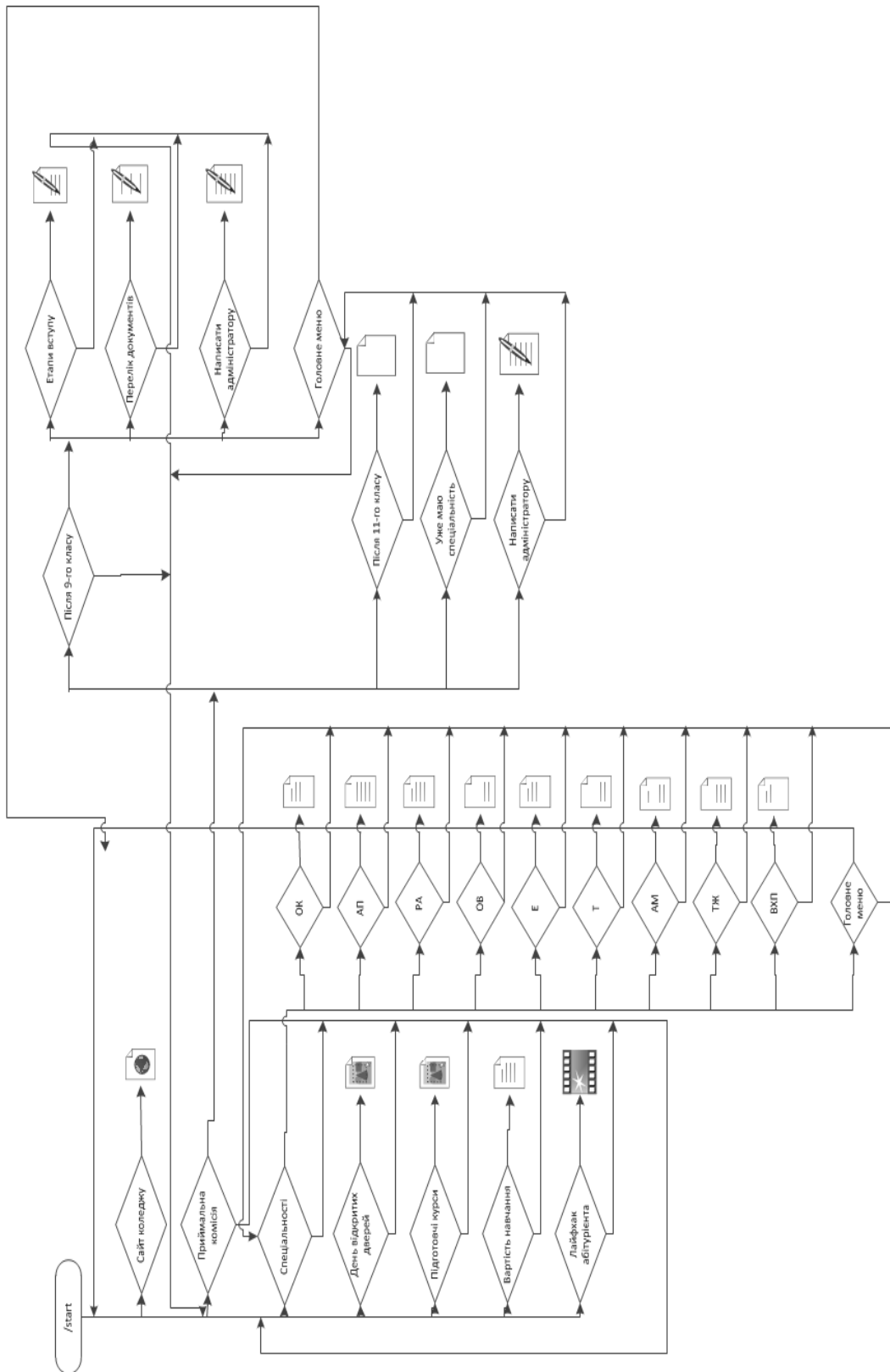


Рисунок В.1 — Блок-схема алгоритму чат-бота

ДОДАТОК Г

Схема принципу роботи чат-бота на платформі Telegram

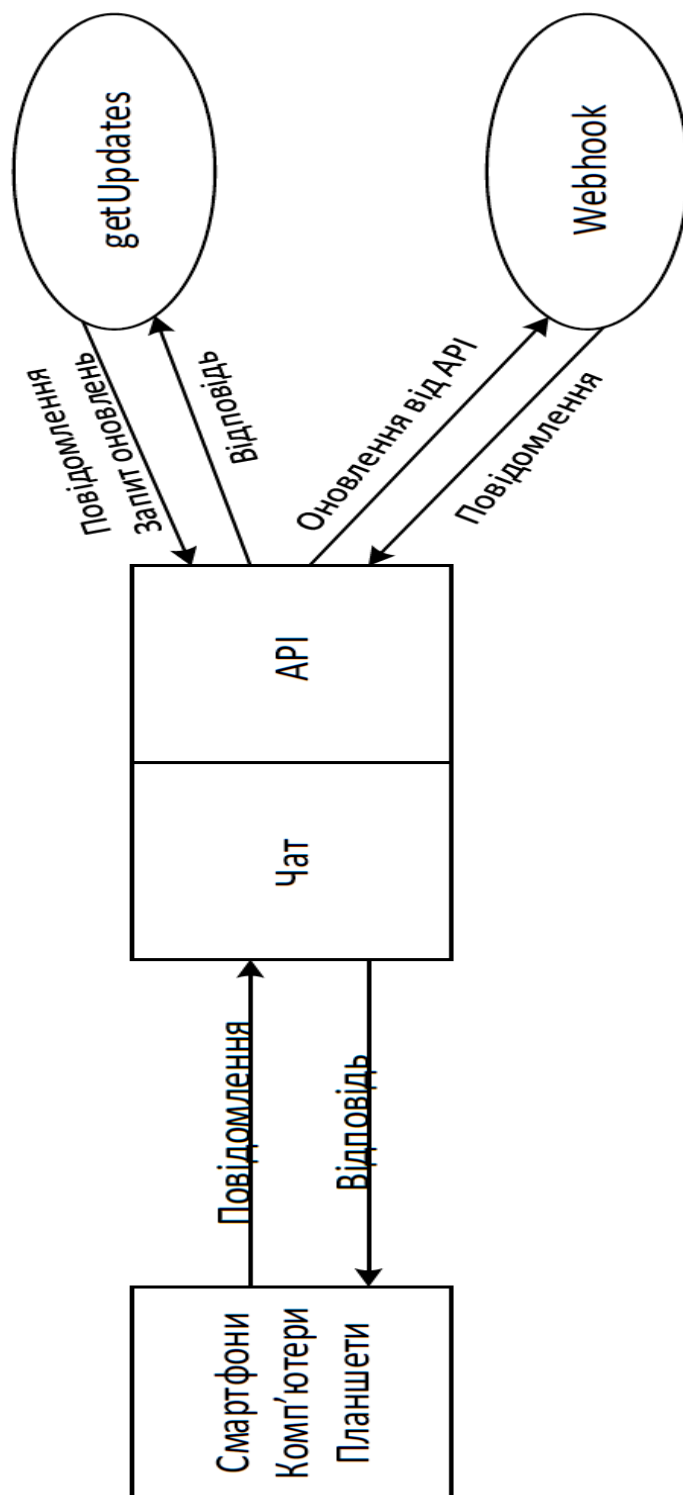


Рисунок Г.1 — Схема принципу роботи чат-бота на платформі Telegram

ДОДАТОК Д

Лістинг чат-бота

```

Файл many.py
import telebot
import config
import datetime
import pytz
import json
import traceback

P_TIMEZONE = pytz.timezone(config.TIMEZONE)
TIMEZONE_COMMON_NAME = config.TIMEZONE_COMMON_NAME
bot = telebot.TeleBot(config.TOKEN)
#bot.send_animation(chat_id=chat_id, document=open('gif/hi.gif', 'rb'))
@bot.callback_query_handler(func=lambda call: True)
def back(call):
    if call.data=='prujmalna_komisija':
        prujmalna_komisija_command(call.message)#виклик функції
prujmalna_komisija (Приймальна комісія)
    elif call.data=='choose_profession':
        choose_profession_command(call.message)
    elif call.data=='open_days':
        open_days_command(call.message)
    elif call.data=='back_1':
        back_1_command(call.message)
    elif call.data=='back_2':
        back_2_command(call.message)
    elif call.data=='nine':
        nine_command(call.message)

```

```
elif call.data=='eleven':
    eleven_command(call.message)
elif call.data=='okr':
    okr_command(call.message)
elif call.data=='etapu_9':
    etapu_9_command(call.message)
elif call.data=='etapu_11':
    etapu_11_command(call.message)
elif call.data=='etapu_okr':
    etapu_okr_command(call.message)
elif call.data=='document_9':
    document_9_command(call.message)
elif call.data=='document_11':
    document_11_command(call.message)
elif call.data=='document_okr':
    document_okr_command(call.message)
elif call.data=='ok':
    ok_command(call.message)
elif call.data=='ap':
    ap_command(call.message)
elif call.data=='ra':
    ra_command(call.message)
elif call.data=='tj':
    tj_command(call.message)
elif call.data=='ov':
    ov_command(call.message)
elif call.data=='vxp':
    vxp_command(call.message)
elif call.data=='am':
```

```

    am_command(call.message)
elif call.data=='e':
    e_command(call.message)
elif call.data=='t':
    t_command(call.message)
elif call.data=='courses':
    courses_command(call.message)
elif call.data=='pay':
    pay_command(call.message)
bot.answer_callback_query(callback_query_id=call.id)
@bot.message_handler(commands=['start']) #Початкове меню
def start_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Сайт коледжу",
url="http://www.vcnuft.vn.ua/index.php"),
        telebot.types.InlineKeyboardButton("Приймальна комісія 📄",
callback_data='prujmalna_komisija'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Обрати спеціальність 🎓",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton("Дні відкритих дверей 🚪",
callback_data='open_days'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Підготовчі курси для вступників 📖",
callback_data='courses'),
        telebot.types.InlineKeyboardButton("Вартість навчання 💰",
callback_data='pay'))

```



```

username = message.from_user.first_name
print(message.from_user.id, " ", message.from_user.first_name, " ",
message.from_user.last_name, " ", message.from_user.username)
bot.send_message(message.chat.id, "Привіт," + username +
"!\\n".format(message.from_user, bot.get_me()),
parse_mode='html')
img = open('D:/Python/Bot/gif/hi.gif', 'rb')
bot.send_video(message.chat.id, img, None, 'Text')
img.close()
#sti = open('D:/Python/Bot/gif/hi.webp', 'rb')
#bot.send_sticker(message.chat.id, sti)

bot.send_message(
message.chat.id,
"Я помічник Вінницького Фахового коледжу "+
"Національного університету харчових технологій (ВіФК НУХТ) 📄 \\n"
+
"\\n"+
"Я допоможу тобі знайти всю необхідну інформацію щодо вступу "+
"на освітні програми ВіФК НУХТ👤♂️"+
" ")
bot.send_message(
message.chat.id,
"Обери одну з категорій, щоб дізнатися більше інформації 📄",
reply_markup=keyboard )
@bot.message_handler(commands=['prujmalna_komisija']) #Приймальна
комісія
def prujmalna_komisija_command(message):

```

```

keyboard = telebot.types.InlineKeyboardMarkup()
keyboard.row(
    telebot.types.InlineKeyboardButton("Після 9-го класу 🏰",
callback_data='nine'),
    telebot.types.InlineKeyboardButton('Після 11-го класу 🏰',
callback_data='eleven'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Уже маю спеціальність 🏰',
callback_data='okr'),
    telebot.types.InlineKeyboardButton('Написати адміністратору 🏰',
url='telegram.me/galina_09_09'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Головне меню ◀️',
callback_data='back_1'))
bot.send_message(
    message.chat.id,
    'Після якого класу чи закладу плануєш продовжити навчання в нашому
коледжі?\n' +
    '\n'+
    'Обери розділ, що відповідає твоєму запити 🔍 ',
    reply_markup=keyboard
)
@bot.callback_query_handler(func=lambda call: call.data=='choose_profession')

```

#Обрати спеціальність

```

def choose_profession_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(

```

```

telebot.types.InlineKeyboardButton("OK □", callback_data='ok'),
telebot.types.InlineKeyboardButton("АП ➡️", callback_data='ap'),
telebot.types.InlineKeyboardButton('РА 📄', callback_data='ra'),
telebot.types.InlineKeyboardButton("ОБ 📄", callback_data='ov'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Е ↘', callback_data='e'),
    telebot.types.InlineKeyboardButton('Т 📄', callback_data='t'),
    telebot.types.InlineKeyboardButton("АМ □", callback_data='am'),
    telebot.types.InlineKeyboardButton('ТЖ □', callback_data='tj'),
    telebot.types.InlineKeyboardButton('ВХП 📄', callback_data='vxp'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Головне          меню          ◀️',
callback_data='back_1'))
bot.send_message(
    message.chat.id,
    "Коледж готує спеціалістів за 9 спеціальностями. \n" +
    "Обирай спеціальність та дізнавайся про неї більше: 📄\n",
    reply_markup=keyboard
)
@bot.message_handler(commands=['ok']) #Приймальна комісія
def ok_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші          спеціальності          📄",
callback_data='choose_profession'),

```

```

telebot.types.InlineKeyboardButton('Головне          меню          ◀',
callback_data='back_1'))
    bot.send_message(
        message.chat.id,
        '_Освітня програма_: *Обслуговування комп'ютерних систем і мереж
*\n'+
        "_Спеціальність_: *123 Комп'ютерна інженерія*\n"+
        "_Галузь знань_: *12 Інформаційні технології*\n"
        "_Термін навчання_: 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+
        "Фаховий молодший бакалавр (*технік-програміст*) готується для
роботи з експлуатації, налагодження, розробки комп'ютерних систем
обслуговування, створення програмного забезпечення та користувальних програм
і їх впровадження; призначається для роботи на підприємствах, в інформаційно-
обчислювальних центрах, фірмах, банках, комерційних організаціях, науково-
дослідницьких інститутах, лабораторіях на посадах: регулювальника
обчислювальної техніки, техніка з системного програмного забезпечення, майстра
з ремонту приладів та апаратури, начальника зміни, електромеханіка з ремонту та
обслуговування лічильно-обчислювальних машин; монтажника приладів та
апаратури автоматичного контролю, регулювання та керування."+
        '\n',
        parse_mode="Markdown"
    )
    bot.send_message(
        message.chat.id,
        'Обери розділ, що відповідає твоєму запиту 🔍 ',
        reply_markup=keyboard,
        parse_mode="Markdown"
    )

```

```

)
@bot.message_handler(commands=['ap']) #Приймальна комісія
def ap_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton('Головне меню ◀️',
callback_data='back_1'))
    bot.send_message(
        message.chat.id,
        '_Освітня програма_: *Монтаж, обслуговування засобів і систем
автоматизації технологічного виробництва *\n'+
        "_Спеціальність_: *151 Автоматизація та комп'ютерно-інтегровані
технології*\n"+
        "_Галузь знань_: *15 Автоматизація та приладобудування*\n"
        "_Термін навчання_: 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+
        "Фаховий молодший бакалавр (*технік-електромеханік*) готується для
роботи на підприємствах харчової промисловості (м'ясокомбінатах,
молокозаводах, хлібозаводах, консервних заводах) на посадах метролога, майстра
по ремонту електронних приладів, начальника цеху, дільниці, майстерні,
оператора ЕОМ. Значний обсяг навчального плану передбачає вивчення сучасної
комп'ютерної техніки."+
        '\n',
        parse_mode="Markdown"
    )
    bot.send_message(

```

```

message.chat.id,
'Обери розділ, що відповідає твоєму запиту 🔍 ',
reply_markup=keyboard,
parse_mode="Markdown"
)
@bot.message_handler(commands=['ra']) #Приймальна комісія
def ra_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton("Головне меню ◀",
callback_data='back_1'))
    bot.send_message(
        message.chat.id,
        "_Освітня програма_ : *Технічне обслуговування і ремонт апаратури зв'язку та оргтехніки *\n"+
        "_Спеціальність_ : *172 Телекомунікації та радіотехніка*\n"+
        "_Галузь знань_ : *17 Електроніка та телекомунікації*\n"
        "_Термін навчання_ : 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+
        "Студенти цієї спеціальності отримують глибокі знання з елементної бази сучасних виробів електронної техніки, зокрема в області апаратури зв'язку та організаційної техніки: принтерів, сканерів, плотерів, копіїв, факсів. Вивчення схемотехніки допоможе зрозуміти принципи роботи будь-якого електронного пристрою, а також дасть можливість самостійно проектувати вироби досить високого рівня складності. Отримані професійні знання і навички у галузі сучасних інформаційних і телекомунікаційних технологій, мереж телефонного і

```

мобільного зв'язку, мереж передачі даних, цифрових мереж інтегрального обслуговування, структурованих кабельних систем, надання ресурсів і послуг мережі Internet перетворюють випускника спеціальності на універсала, здатного виконувати більшу частину регламенту обслуговуючих та ремонтних робіт для комп'ютерної мережі та організаційної техніки. Фахівці даної спеціальності можуть займати такі посади: диспетчер електрозв'язку, технік електрозв'язку, технік з сигналізації, технік із конфігурованої комп'ютерної системи, технік із структурованої кабельної системи, технік обчислювального центру."

```

    '\n',
    parse_mode="Markdown"
)
bot.send_message(
    message.chat.id,
    'Обери розділ, що відповідає твоєму запиту 🔍 ',
    reply_markup=keyboard,
    parse_mode="Markdown"
)
@bot.message_handler(commands=['tj']) #Приймальна комісія
def tj_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton('Головне меню 🏠',
callback_data='back_1'))
    bot.send_message(
        message.chat.id,
        '_Освітня програма_ : *Виробництво жирів та жирозамінників *\n'+

```

```
"_Спеціальність_ : *181 Харчові технології*\n"+
"_Галузь знань_ : *18 Виробництво та технології*\n"
"_Термін навчання_ : 3 роки 10 місяців - на основі БЗСО;\n"+
"Фаховий молодший бакалавр (*технік-технолог*) готується до професійної діяльності у харчовій та переробній промисловості для проведення технологічних процесів олійно-жирового виробництва для роботи на посадах: керівника виробничих підрозділів у промисловості, технолога, техника-лаборанта технічного аналізу, майстра виробничого навчання, виробництва (дільниці); лаборанта-мікробіолога."
```

```
\n',
parse_mode="Markdown"
)
bot.send_message(
message.chat.id,
'Обери розділ, що відповідає твоєму запити 🔍 ',
reply_markup=keyboard,
parse_mode="Markdown"
)
@bot.message_handler(commands=['vxp']) #Приймальна комісія
def vxp_command(message):
keyboard = telebot.types.InlineKeyboardMarkup()
keyboard.row(
telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
telebot.types.InlineKeyboardButton("Головне меню 🏠",
callback_data='back_1'))

bot.send_message(
```



```

message.chat.id,
'_Освітня програма_ : *Виробництво харчової продукції *\n'+
"_Спеціальність_ : *181 Харчові технології*\n"+
"_Галузь знань_ : *18 Виробництво та технології*\n"
"_Термін навчання_ : 3 роки 10 місяців - на основі БЗСО;\n"+
"Фаховий молодший бакалавр (*технік-технолог з технології харчування*) - грамотний фахівець у галузі ресторанного господарства, професіонал-консультант із технології приготування кулінарних, кондитерських виробів, страв та напоїв, сервірування столів, керівник середньої ланки у вмінні приготувати якісні страви, гарно їх оформити, подати, у вмінні засервірувати стіл відповідно до свята, замовлень відвідувачів, фахівець, здатний проявляти максимум фантазії, творчості при підході до справи, вплинути на настрій гостей, продемонструвати культуру обслуговування, діловий працівник сфери торгівлі, який підтримує вітчизняного виробника та вміло задовольняє потреби споживачів. Випускники спеціальності можуть обіймати наступні посади: помічник керівника виробничого підрозділу, помічник керівника малого підприємства, технік-технолог з технології харчування, виробник харчових напівфабрикатів, кухар, бармен, офіціант, сомельє, кондитер, кулінар, метрдотель."
\n',
parse_mode="Markdown"
)
bot.send_message(
message.chat.id,
'Обери розділ, що відповідає твоєму запиту 🔍 ',
reply_markup=keyboard,
parse_mode="Markdown"
)
@bot.message_handler(commands=['am']) #Приймальна комісія

```

```

def am_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 🛠️",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton('Головне меню ◀️',
callback_data='back_1'))

    bot.send_message(
        message.chat.id,
        '_Освітня програма_: *Обслуговування та ремонт автомобілів і
двигунів *\n'+
        "_Спеціальність_: *274 Автомобільний транспорт*\n"+
        "_Галузь знань_: *27 Транспорт*\n"
        "_Термін навчання_: 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+
        "Фаховий молодший бакалавр (*механік*) готується для виробничо-
технічної та організаційно-управлінської діяльності, пов'язаної з технічним
обслуговуванням та ремонтом автомобілів і двигунів для роботи на посадах:
начальника дільниці (майстерні); майстра виробничих зон і дільниць ТО і ремонту
автомобілів; старшого механіка (механіка автотранспортних, автообслуговуючих
і авторемонтних підприємств); завідувача автогаражами підприємств, організацій,
закладів різних форм власності; керівника приватних малих підприємств по
технічному обслуговуванню і ремонту автомобілів."+
        '\n',
        parse_mode="Markdown"
    )
    bot.send_message(

```

```

message.chat.id,
'Обери розділ, що відповідає твоєму запиту 🔍 ',
reply_markup=keyboard,
parse_mode="Markdown"
)
@bot.message_handler(commands=['ov']) #Приймальна комісія
def ov_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton("Головне меню ◀",
callback_data='back_1'))

    bot.send_message(
        message.chat.id,
        '_ Освітня програма_ : *Організація виробництва*\n'+
        "_ Спеціальність_ : *073 Менеджмент*\n"+
        "_ Галузь знань_ : *07 Управління та адміністрування*\n"
        "_ Термін навчання_ : 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+
        "Фаховий молодший бакалавр (*фахівець з виробничого менеджменту*) готується для цілеспрямованої організаційно-управлінської діяльності в підрозділах та підприємствах усіх галузей виробництва усіх форм власності; призначається для роботи на посадах: керівника виробничих підприємств у промисловості; керівника малого підприємства, агента із зайнятості і трудових контрактів, фахівця у галузі управління, секретаря адміністративних органів, інспектора з кадрів, техника з підготовки виробництва."+)

```

```

    '\n',
    parse_mode="Markdown"
)
bot.send_message(
    message.chat.id,
    'Обери розділ, що відповідає твоєму запиту 🔍 ',
    reply_markup=keyboard,
    parse_mode="Markdown"
)
@bot.message_handler(commands=['e']) #Приймальна комісія
def e_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Інші спеціальності 📄",
callback_data='choose_profession'),
        telebot.types.InlineKeyboardButton("Головне меню ◀️",
callback_data='back_1'))

    bot.send_message(
        message.chat.id,
        '_Освітня програма_ : *Монтаж і експлуатація електроустаткування підприємств і цивільних споруд*\n'+
        "_Спеціальність_ : *141 Електроенергетика, електротехніка та електромеханіка*\n"+
        "_Галузь знань_ : *14 Електрична інженерія*\n"+
        "_Термін навчання_ : 3 роки 10 місяців - на основі БЗСО;\n"+
        ".....2 роки 10 місяців - на основі ПЗСО.\n"+

```

"Фаховий молодший бакалавр (*технік-електрик*) готується як безпосередній організатор виробничого процесу на дільниці, в бригаді, зміні, цеху з експлуатації, технічного обслуговування і ремонту електрообладнання і установок промислових підприємств різних галузей господарства; призначається для роботи в цехах основного і допоміжного виробництва, в лабораторіях головного енергетика, технічного контролю, на посадах: технік-проектувальника, - конструктора, - налагоджувальника, - організатора виробництва, технік з монтажу, з експлуатації, з ремонту."

```

    '\n',
    parse_mode="Markdown"
)
bot.send_message(
    message.chat.id,
    'Обери розділ, що відповідає твоєму запиту 🔍 ',
    reply_markup=keyboard,
    parse_mode="Markdown"
)
@bot.callback_query_handler(func=lambda call: call.data=='open_days')
def open_days_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        #telebot.types.InlineKeyboardButton('Назад ⬅️', callback_data='back-3'),
        telebot.types.InlineKeyboardButton('Головне меню ◀️',
callback_data='back_1')
    )

    username = message.from_user.first_name
    bot.send_message(

```

```

message.chat.id,
username + ", велика родина ВіФК НУХТ чекає тебе
\n".format(message.from_user, bot.get_me()),
parse_mode='html')
sti = open('D:/Python/Bot/image/opendays.jpg', 'rb')
bot.send_sticker(message.chat.id, sti)
bot.send_message(
    message.chat.id, "Незволікай!",
    reply_markup=keyboard)
@bot.message_handler(commands=['nine']) #На основі базової загальної
середньої освіти (9 клас)
def nine_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Етапи вступу ✓',
callback_data='etapu_9'),
        telebot.types.InlineKeyboardButton('Перелік документів 📄',
callback_data='document_9'))
    keyboard.row(
        telebot.types.InlineKeyboardButton('Написати адміністратору 📩',
url='telegram.me/galina_09_09'),
        telebot.types.InlineKeyboardButton('Головне меню ◀',
callback_data='back_1'))
    bot.send_message(
        message.chat.id,
        "Завантажуємо інформацію ... ☐" )
    bot.send_message(
        message.chat.id,

```



```

"\nОбери необхідний розділ 📁 ",
reply_markup=keyboard )
#etapu_9, etapu_11, etapu_okr
@bot.callback_query_handler(func=lambda call: call.data=='etapu_9')
def etapu_9_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Назад ⬅️', callback_data='back_3'),
        telebot.types.InlineKeyboardButton('Головне        МЕНЮ        ◀️',
callback_data='back_1')
    )

    username = message.from_user.first_name
    bot.send_message(
        message.chat.id,
        username + ", завантаж файл, щоб не забути про важливі дати
!\n".format(message.from_user, bot.get_me()),
        parse_mode='html')
    file = open('D:/Python/Bot/image/etapu_9.jpg', 'rb')
    bot.send_document(message.chat.id, file)
    #працює завантаження з серера bot.send_document(message.chat.id,
'https://atikegalle.com/uploads/1514125303.pdf')
    #sti = open('D:/Python/Bot/image/documentu_11', 'rb')
    #bot.send_sticker(message.chat.id, sti)
    #або
    #img = open('D:/Python/Bot/gif/etapu.gif', 'rb')
    #bot.send_video(message.chat.id, img, None, 'Text')
    #img.close()

```

```

bot.send_message(
    message.chat.id,
    "Не зволікай! 😊 ",
    reply_markup=keyboard)
@bot.callback_query_handler(func=lambda call: call.data=='etapu_11')
def etapu_11_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Назад  меню ',
callback_data='back_1')
    )
    username = message.from_user.first_name
    bot.send_message(
        message.chat.id,
        username + ", завантаж файл, щоб не забути про важливі дати !
\n".format(message.from_user, bot.get_me()),
        parse_mode='html')
    file = open('D:/Python/Bot/image/etapu_11.jpg', 'rb')
    bot.send_document(message.chat.id, file)
    bot.send_message(
        message.chat.id,
        "Не зволікай! 😊 ",
        reply_markup=keyboard)
#Функції для document_9, document_11, document_okr
@bot.callback_query_handler(func=lambda call: call.data=='document_9')
def document_9_command(message):

```



```

keyboard = telebot.types.InlineKeyboardMarkup()
keyboard.row(
    telebot.types.InlineKeyboardButton('Назад ↩', callback_data='back_3'),
    telebot.types.InlineKeyboardButton('Головне меню ◀',
callback_data='back_1')
)

username = message.from_user.first_name
bot.send_message(
    message.chat.id,
    "Перелік документів для вступу 📄 \n".format(message.from_user,
bot.get_me()),
    parse_mode='html')
file = open('D:/Python/Bot/image/documentu_9.pdf', 'rb')
bot.send_document(message.chat.id, file)
bot.send_message(
    message.chat.id,
    "Краще оригінали 😊 ",
    reply_markup=keyboard)
@bot.callback_query_handler(func=lambda call: call.data=='document_11')
def document_11_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Назад ↩', callback_data='back_3'),
        telebot.types.InlineKeyboardButton('Головне меню ◀',
callback_data='back_1')
)

```

```

username = message.from_user.first_name
bot.send_message(
    message.chat.id,
    "Перелік документів для вступу 📄 \n".format(message.from_user,
bot.get_me()),
    parse_mode='html')

file = open('D:/Python/Bot/image/documentu_11.pdf', 'rb')
bot.send_document(message.chat.id, file)
bot.send_message(
    message.chat.id,
    "Краще оригінали 😊 ",
    reply_markup=keyboard)
@bot.callback_query_handler(func=lambda call: call.data=='document_okr')
def document_okr_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Назад ⬅️', callback_data='back_3'),
        telebot.types.InlineKeyboardButton('Головне меню ⬅️',
callback_data='back_1')
    )

username = message.from_user.first_name
print(username)
bot.send_message(
    message.chat.id,

```

```

        "Перелік документів для вступу 📄 \n".format(message.from_user,
bot.get_me()),
        parse_mode='html')
file = open('D:/Python/Bot/image/documentu_okr.pdf', 'rb')
bot.send_document(message.chat.id, file)
bot.send_message(
    message.chat.id,
    "Краще оригінали 😊 ",
    reply_markup=keyboard)
@bot.message_handler(commands=['back_3']) #До Обрати спеціальності
def back_3_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton("Мене́джмент 📁",
callback_data='nine'),
        telebot.types.InlineKeyboardButton("Комп'ютерна інженерія 📁",
callback_data='class-11'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Електроенергетика, електротехніка
та електромеханіка ⚡", callback_data='fah_robittn'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Теплоенергетика 📁",
callback_data='fah_robittn'),
        telebot.types.InlineKeyboardButton("Автомобільний транспорт 📁",
callback_data='fah_robittn'))
    keyboard.row(
        telebot.types.InlineKeyboardButton("Автоматизація та комп'ютерно-
інтегровані технології ➡📁", callback_data='fah_robittn'))

```

```

keyboard.row(
    telebot.types.InlineKeyboardButton('Харчові технології (Виробництво
жирів та жирозамінників) 🍷', callback_data='fah_rob1tn'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Харчові технології (Виробництво
харчової продукції) 🍷', callback_data='fah_rob1tn'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Телекомунікації та радіотехніка 📡',
callback_data='fah_rob1tn'))
keyboard.row(
    telebot.types.InlineKeyboardButton('Назад ⬅️', callback_data='back-1'),
    telebot.types.InlineKeyboardButton('Головне меню 🏠',
callback_data='back_1'))
bot.send_message(
    message.chat.id,
    'Сподобалася спеціальність? 😊 \n' +
    'Обирай іншу, вона теж перспективна та актуальна в наш час: 🍷',
    reply_markup=keyboard
)

@bot.callback_query_handler(func=lambda call: call.data=='pay')
def pay_command(message):
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(
        telebot.types.InlineKeyboardButton('Назад ⬅️', callback_data='back_3'),
        telebot.types.InlineKeyboardButton('Головне меню 🏠',
callback_data='back_1')

```

```

)

username = message.from_user.first_name
bot.send_message(
    message.chat.id,
    "Інвестиції в самого себе — краще за вклад не буває
!\n".format(message.from_user, bot.get_me()),
    parse_mode='html')

file = open('D:/Python/Bot/image/pay.jpg', 'rb')
bot.send_document(message.chat.id, file)
def get_exchange_diff(last, now):
    return {
        'sale_diff': float("%.6f" % (float(now['sale']) - float(last['sale']))),
        'buy_diff': float("%.6f" % (float(now['buy']) - float(last['buy'])))
    }
def get_edited_signature():
    return '<i>Updated ' +
str(datetime.datetime.now(P_TIMEZONE).strftime('%H:%M:%S')) + ' (' +
TIMEZONE_COMMON_NAME + '</i>'
    bot.polling(none_stop=True)

```

ДОДАТОК Е

Протокол перевірки навчальної (кваліфікаційної) роботи