

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Подуфалов Вячеслав Валентинович,
студент групи РФ-171

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Розробка додатку для упорядкування електронних документів з
використанням чат-бота

Спеціальність:
122 Комп'ютерні науки

Освітня програма Комп'ютерні науки та інформаційна безпека

Керівник:
Кушніренко Наталія Ігорівна,
к.т.н., доцент

Одеса – 2022

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки та інформаційна безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ

д.т.н., проф. А.А.Кобозєва
_____ 202_ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Подуфалову Вячеславу Валентиновичу

- 1.Тема роботи: *Розробка додатку для упорядкування електронних документів з використанням чат-бота, керівник роботи Кушніренко Наталія Ігорівна, к.т.н., доцент, затверджені наказом ректора НУ «Одеська політехніка» від „_____” _____ 20__ р. №_____ .*
- 2.Зміст роботи: *аналіз предметної області та наявних продуктів, основні технології та способи розробки, реалізація програмного продукту.*
- 3.Перелік ілюстративного матеріалу: *10 найпопулярніших мов програмування на грудень 2022 року, схема роботи чат-бота на базі Telegram, інтерфейс програми з модулем Tkinter, інтерфейс програми з модулем PyQt5, інтерфейс програми з модулем CustomTkinter, процес створення нового бота, встановлення пакету мови Python у Visual Studio Code, частина програмного код файлу «config.py», список імпортованих*

модулів, програмний код класу *User*, код боту для реагування на певні повідомлення користувачів, діалог користувача із чат-ботом для надсилання файлу, вибір між студентом та викладачем в додатку, поля для введення особистих даних користувачем, таблиця із файлами для студента, шлях збереження файлів від викладача Сергія Попова до групи РФ-171.

4. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Дата видачі завдання “ _____ ” _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	<i>Аналіз джерел з теми випускної кваліфікаційної роботи</i>	<i>25-09-2022</i>	<i>виконано</i>
2	<i>Обґрунтування вибору рішення. Збір даних</i>	<i>04-10-2022</i>	<i>виконано</i>
3	<i>Аналіз технологій необхідних для реалізації продукту</i>	<i>15-10-2022</i>	<i>виконано</i>
4	<i>Аналіз умов функціонування чат-бота</i>	<i>22-10-2022</i>	<i>виконано</i>
5	<i>Розробка програмного продукту для функціонування чат-бота</i>	<i>03-11-2022</i>	<i>виконано</i>
6	<i>Розробка програмного додатка для управління документами</i>	<i>10-11-2022</i>	<i>виконано</i>
7	<i>Підготовка тексту роботи</i>	<i>22-11-2022</i>	<i>виконано</i>
8	<i>Підготовка презентації та доповіді</i>	<i>28-11-2022</i>	<i>виконано</i>
9	<i>Попередній захист</i>	<i>08-12-2022</i>	<i>виконано</i>
10	<i>Нормоконтроль, рецензування</i>	<i>19-12-2022</i>	<i>виконано</i>
11	<i>Занесення роботи в електронний архів</i>	<i>20-12-2022</i>	<i>виконано</i>
12	<i>Допуск до захисту у завідувача кафедри</i>	<i>21-12-2022</i>	<i>виконано</i>

Здобувач вищої освіти _____

Подуфалов В.В..

Керівник роботи _____

Кушніренко Н.І.

АНОТАЦІЯ

Кваліфікаційна робота на тему “Розробка додатку для упорядкування електронних документів з використанням чат-бота” на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 122 Комп’ютерні науки, спеціалізація, освітня програма: Комп’ютерні науки та інформаційна безпека, містить 16 рисунків, 9 таблиць, 4 додатки, 23 літературних джерела за переліком посилань. Робота виконана на 80 сторінках загального тексту і 50 сторінках основного тексту.

Метою роботи є автоматизація процесу управління документами за допомогою чат-бота на основі месенджера та програмного додатка.

У роботі проведено аналіз месенджерів, які можуть бути взяті за основу при розробці чат-бота, а також аналіз мов програмування, які нададуть перелік необхідних можливостей для швидкої та ефективної взаємодії програмного додатка із обраним месенджером.

У результаті виконання роботи був розроблений чат-бот на основі Telegram, який призначений для відправлення студентами або викладачами файлів та програмний додаток для упорядкування та завантаження документів викладачем або студентом відповідно.

Отримані результати кваліфікаційної роботи можуть бути використані для автоматизації процесу документообігу та спрощення управління документами на персональних комп’ютерах викладачів або на кафедрі університету.

ЧАТ-БОТ, АВТОМАТИЗАЦІЯ ПРОЦЕСУ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ, УПОРЯДКУВАННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ.

ANNOTATION

Qualification work on the topic "Development of an application for organizing electronic documents using a chatbot" for the second (master's) level of higher education in specialty 122 Computer Science, specialization, educational program: Computer science and information security, contains 16 figures, 9 tables, 4 appendices, 23 references. The work is performed on 80 pages of the general text and 50 pages of the main text.

The aim of the work is to automate the document management process using a chatbot based on a messenger and a software application.

The work analyzes the messengers that can be used as a basis for the development of a chatbot, as well as the analysis of programming languages that will provide a list of necessary features for fast and efficient interaction of the software application with the selected messenger.

As a result of the work, a Telegram-based chatbot was developed, which is designed to send files by students or teachers and a software application for organizing and downloading documents by a teacher or student, respectively.

The results of the qualification work can be used to automate the workflow process and simplify document management on personal computers of teachers or at the university department.

CHAT-BOT, AUTOMATION OF THE ELECTRONIC DOCUMENT MANAGEMENT PROCESS, ARRANGEMENT OF ELECTRONIC DOCUMENTS.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Обґрунтування обраної теми для кваліфікаційної роботи	10
1.2 Аналіз наявних продуктів	17
2 ОСНОВНІ ТЕХНОЛОГІЇ ТА СПОСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ	22
2.1 Вибір месенджера для реалізації чат-бота	22
2.2 Вибір мови програмування для реалізації програмного продукту....	31
2.3 Вибір інтерпретатора для написання програмного коду продукту ...	32
2.4 Розробка схеми роботи програмного продукту	34
2.5.Вибір системи управління баз даних	34
2.6 Вибір пакету додатків для написання користувацького інтерфейсу.	35
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ.....	39
3.1 Створення боту у Telegram	39
3.2 Встановлення додатка Python у середі Visual Studio Code.....	40
3.3 Написання вихідного коду для чат-бота	41
3.4 Розробка додатка для управління файлами.....	46
ВИСНОВКИ	50
ПЕРЕЛІК ПОСИЛАНЬ.....	52
Додаток А. Програмний код чат-бота.. Ошибка! Закладка не определена.	
Додаток Б. Програмний код додатка для студента ... Ошибка! Закладка не определена.	
Додаток В. Програмний код додатка для викладача. Ошибка! Закладка не определена.	
Додаток Г. Програмний код додатку “config.py” Ошибка! Закладка не определена.	

ВСТУП

За останні декілька років сталися події, які зробили складним, а інколи взагалі неможливим очне навчання в навчальних закладах в Україні: спочатку це був вірус COVID-19, а тепер російсько-українська війна [1-2]. Загроза життю та здоров'ю людей змусила заклади навчання і їх працівників перейти на дистанційне навчання та працю, адже продовжувати функціонувати в очному режимі навчальні заклади не могли: станом на 1 вересня і з початку повномасштабного вторгнення російські війська пошкодили 2,4 тисячі навчальних закладів в Україні, з них 270 зруйновані повністю. І навіть на даний момент майже 50% навчальних закладів в Україні продовжують дистанційне навчання [3]. Через це сьогодні дуже гостро стоїть питання з автоматизації організації електронного документообігу, бо відправляти всі документи самотійно поодиноці досить трудомістко, особливо якщо кількість файлів, що потрібно відправляти може сягати декількох сотень в день.

На основі аналізу існуючих на ринку рішень, таких як Вчасно, Мій Арт-Офіс, Alfresco, Microsoft SharePoint, Directum, NauDoc та інших було визначено, що жодна з цих систем не підходить для продукту, що розроблюється по різних причинах, таким як обмеження використання за часом або за кількістю користувачів, або через велику кількість зайвого функціоналу. Таким чином, не було знайдено програмних рішень для сфери освіти, які б задовольняли усім потребам і одночасно не були б перенавантажені зайвими у обраній сфері можливостями. Через такі недоліки було вирішено створити програмний продукт, який полегшив би дистанційну роботу викладачам університетів та студентам, був би легким у використанні і водночас мав би можливість виконувати ті функції, які необхідні користувачам, ким будуть студенти та викладачі.

Після детального аналізу обраної теми, можна зазначити, що об'єкт дослідження – автоматизація упорядкування електронних документів, а предмет дослідження – розробка програмного рішення для автоматизації процесу документообігу.

Новизна кваліфікаційної роботи полягає в тому, що вперше розроблено програмне забезпечення для упорядкування документів, яке є специфічним для сфери освіти і базується на взаємодії чат-бота і програмного додатку, розробленого з використанням новітніх технологій.

Практичні результати полягають в тому, що запропонований програмний продукт може буде використано в навчальному процесі для взаємодії між студентами та викладачами з метою економії часу, який витрачається на передачу та упорядкування електронних документів.

Метою розробки програмного продукту є автоматизація процесу управління документами за допомогою програмного додатку та чат-боту на базі Telegram.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- а) виконати аналіз предметної області та наявних програмних рішень;
- б) виконати аналіз найбільш гнучких технологій для роботи із Telegram та документами;
- в) розробити чат-бота за допомогою месенджеру Telegram;
- г) розробити додаток для упорядкування та завантаження документів.

Програмний продукт, що буде розроблюватися у кваліфікаційній роботі, розглядатиметься на прикладі взаємодії між студентом та викладачем і повинен мати можливість виконувати такі функції, як:

- отримувати, зберігати та надсилати файли;
- упорядковувати файли за іменем, прізвищем, предметом та групою;
- використовувати базу даних для зберігання інформації про користувачів та їх документи;
- надавати можливість вибору каталогу для зберігання файлів;
- розрізняти між собою користувачів чат-бота;
- отримувати та зберігати файли будь-якого формату та розміру.

Результати досліджень були подані у науковій статті «Організація електронного документообігу з використанням чат-боту», що опублікована у журналі «Інформатика та математичні методи в моделюванні» [4].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування обраної теми для кваліфікаційної роботи

На момент написання кваліфікаційної роботи на території України вже понад 9 місяців продовжується повномасштабна війна через агресію російської федерації . Постійні повітряні тривоги, руйнування цивільної інфраструктури внаслідок вибуху російських ракет унеможлиблює очне відвідування навчальних закладів, які не обладнані бомбосховищами. До початку повномасштабного вторгнення російських військ в Україну більшість навчальних закладів декілька років поспіль також навчалися дистанційно, але через інші причини - поширення респіраторної інфекції, викликаной вірусом COVID-19.

Через перехід від очного навчання і праці на дистанційне, тепер будь-яка сфера, де проводиться обмін документами, повинна повністю переходити в електронний обмін необхідними файлами. Наразі, зазвичай, для відправлення та отримання файлів використовують електронну пошту.

Для кваліфікаційної роботи було вирішено реалізувати програмний продукт, що зможе автоматизувати сортування та полегшити електронний документообіг для користувачів [5]. Розробка такого програмного рішення може бути досить корисною, адже завантажувати та зберігати документи використовуючи електронну пошту або інші ресурси вручну - це досить довгий та трудомісткий процес, а якщо кількість документів для обміну може сягати сотень за день, то процес автоматизації для такого документообігу стане не просто корисним, а необхідним для максимального збереження часу користувачів.

Тож зараз популярний процес отримання документів виглядає приблизно так : студенти відправляють виконані роботи викладачеві на електронну пошту, після цього викладачеві потрібно перевірити пошту,

знайти ті роботи, які він ще не завантажив і після цього завантажити необхідні файли у відповідні директорії, після чого переходити до перевірки обраних робіт. Процес отримання студентами документів від викладачів мало чим відрізняється від описаного вище процесу для викладача.

У програмному продукту кваліфікаційної роботи реалізується розробка чат-бота для автоматизації сортування та зберігання файлів на прикладі викладача та студента з ціллю максимального зменшення часу, який кожен з користувачів витрачає на завантаження та зберігання цих файлів в необхідних директоріях вручну, що надасть користувачам більше часу на виконання інших справ : викладачеві на перевірку робіт студентів, а студентам - на виконання робіт з інших предметів.

Для реалізації чат-бота для відправлення документів було випробувано більшість популярних в Україні та світі месенджерів, серед яких Viber, Telegram та WhatsApp. Також серед студентів і викладачів Національного Університету “Одеська політехніка” було проведено опитування, яким з цих месенджерів вони частіше за все користуються і який з них є найбільш зручним для користувачів. Опитування показало, що велика кількість викладачів та студентів віддають перевагу Telegram, перш за все через приємний дизайн додатка, швидкість роботи та можливості, які месенджер надає користувачам.

Результати опитування не виявилися неочікуваними тому, що навіть з власного досвіду користування месенджером можу зазначити, що саме Telegram серед всіх інших месенджерів є найбільш зручним для користувачів між аналогів. Окрім текстових повідомлень Telegram дозволяє надсилати також відео та аудіозаписи, документи, зображення, голосові повідомлення та наліпки, які, на відміну від конкурентів обраного

месенджеру, також повністю безкоштовні та можуть бути створені кожним користувачем за наявності необхідних здібностей.

Одним із найбільших плюсів реалізації чат-бота саме на основі месенджера Telegram є захищеність даних, яку він надає користувачеві [6]. У месенджері використовується одразу кілька систем захисту інформації із застосуванням таких алгоритмів [7], як SHA-256, RSA-2048, AES-256, та DH-2048 при використанні додатка Telegram на персональному комп'ютері або телефоні, а також HTTPS-протокол [8] у випадку, коли Telegram використовується у вигляді WEB-версії, а ключі шифрування, що не дивно, не передаються третім особам незалежно від пристрою, де Telegram використовується.

Telegram, на відміну від своїх конкурентів, надає найбільші можливості конфіденційності та безпеки кожному користувачу абсолютно безкоштовно. У месенджері також можна створити секретний чат, який надасть ще більший захист особистої інформації від зловмисників, адже користувачі секретного чату з'єднуються напряму, із використанням наскрізного шифрування, це унеможлиблює доступ до діалогу навіть працівникам месенджера. Секретний чат сповістить співрозмовника, якщо ви зробите знімок екрана телефону, надає можливість налаштовувати видалення відправлених повідомлень через обраний проміжок часу, що дозволить не залишати слідів навіть на серверах компанії, і не дозволяє пересилати повідомлення в інші діалоги із секретного чату. Іншою перевагою месенджера Telegram з боку програміста є відкритий API [9], що надає можливість користуватись та управляти всім необхідним функціоналом Telegram використовуючи програмний код. API це аббревіатура від словосполучення "Application Programming Interface", що може перекладатися як "Інтерфейс програмування додатків" або "програмний інтерфейс програми", тобто це сервіс операційної системи або бібліотек, що надає можливості взаємодії між декількома програмами.

Приклад застосування API: програмний додаток повинен безпосередньо звертатися до сервера Telegram із запитом на створення події з зазначеними деталями, отримувати відповідь Telegram, обробляти його, і передавати відповідну інформацію в додаток, наприклад, повідомлення із запитом на підтвердження користувача. В якості альтернативи програмний додаток може зробити запит до API сервера Telegram, минаючи сервер компанії.

У програмному продукту, що розроблюється у кваліфікаційній роботі, буде використовуватися програмний додаток на комп'ютері користувача, який при запуску програми або при виконанні деякої функції буде створювати запит до API обраного месенджера та отримувати відповідь у вигляді даних, які пізніше можуть бути записані у змінну в кодї, яка буде використана надалі.

Сьогодні, коли обчислювальна потужність персональних комп'ютерів стала досить високою, існує попит на інтерпретовані мови програмування [10]. Інтерпретована мова програмування - це мова, в якій початковий програмний код не перетворюється попередньо повністю у машинний код для виконання, як це робиться у компільованих мовах, а виконується рядок за рядком з допомогою спеціальної програми-інтерпретатора.

Мова програмування, яка була обрана для реалізації програмного додатку - Python. З 2007 року Python кілька разів обирали мовою року, як мова, що має найбільший зріст рейтингу за рік, що можна побачити на рис. 1.1. Також на момент грудня 2022 роки обрана мова займає перше місце в списку за кількістю запитів користувачів у пошукових системах за індексом ТЮВЕ [11], і це місце Python займає вже більше року із рейтингом 16.66%, обганяючи мову С на 0.10%, а С++ на 4.72%. Індекс спільноти програмістів ТЮВЕ - це показник популярності мов програмування. Індекс оновлюється раз на місяць. Рейтинги базуються на

кількості кваліфікованих інженерів по всьому світу, курсів та сторонніх постачальників. Для розрахунку рейтингів використовуються популярні пошукові системи, такі як Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube та Baidu. Індекс може бути використаний для перевірки того, чи залишаються ваші навички програмування актуальними, або для прийняття стратегічного рішення про те, яку мову програмування слід взяти на озброєння, починаючи будувати нову програмну систему.


Dec 2022	Dec 2021	Change	Programming Language	Ratings	Change
1	1		 Python	16.66%	+3.76%
2	2		 C	16.56%	+4.77%
3	4	▲	 C++	11.94%	+4.21%
4	3	▼	 Java	11.82%	+1.70%
5	5		 C#	4.92%	-1.48%
6	6		 Visual Basic	3.94%	-1.46%
7	7		 JavaScript	3.19%	+0.90%
8	9	▲	 SQL	2.22%	+0.43%
9	8	▼	 Assembly language	1.87%	-0.38%
10	12	▲	 PHP	1.62%	+0.12%

Рисунок 1.1 – 10 найпопулярніших мов програмування на грудень 2022 року

Переваги Python включають в себе :

- високий попит на ринку праці. Однією з головних переваг вивчення Python є чудова заробітна плата, яку він пропонує, що може зробити ваше майбутнє безпечним. Вивчивши Python, ви зможете використовувати навички програмування, щоб зробити чудову кар'єру в індустрії науки про дані, що швидко розвивається. Крім того, попит на програмістів Python зростатиме з появою нових додатків для машинного навчання;

- широке застосування. Мова Python може бути використана для широкого застосування. Вона широко використовується вченими, інженерами та математиками для всіх видів додатків. Python також корисний для створення прототипів різного роду експериментів. Вона використовується в багатьох новаторських галузях, таких як створення анімації для фільмів, розробка настільних додатків на основі графічного інтерфейсу, наукових та обчислювальних додатків, в машинному навчанні, обробці зображень та графічному дизайні, розробка веб та десктопних додатків, розробка складних розрахункових систем, систем управління життєзабезпеченням, Інтернету речей (IoT), ігор та багато іншого;
- легкість у вивченні та підтримці. Python як мова пропонує чудову читабельність і простий у вивченні синтаксис для початківців, які хотіли б використати її можливості. Якщо ви вирішили вивчати цю мову, вона підтримує вас за допомогою рекомендацій щодо стилю коду, PEP 8, та надає набір правил для полегшення форматування коду;
- навчання організованості. Python доволі суворо ставиться до відступів, що навчить програміста зберігати код своєї програми красивим та акуратним;
- сумісність із багатьма платформами. Мова може використовуватися не тільки на ОС (велика частина мов програмування оптимізована її розробниками тільки для Windows), таких як Ubuntu та Linux;
- підтримка великих IT-компаній. Багато відомих компаній з розробки програмного забезпечення використовують або раніше використовували Python для своїх проектів;
- низький поріг входу. Python має легкий для розуміння синтаксис для програмістів-початківців;
- багато додаткових пакетів (PIP). Програміст часто може зіткнутися із проблемою, яку неможливо вирішити використовуючи стандартні засоби Python, або вирішення цієї проблеми вимагає занадто багато

ресурсів комп'ютера. У більшості випадків в такій ситуації можна знайти створений іншими користувачами пакет додатків, який допоможе із розв'язанням цієї задачі. PIP – це стандартний менеджер для керування пакетами у Python. PIP також дозволяє змінювати версії пакетів за необхідністю, видаляти їх тощо;

- має великі бібліотеки та фреймворки. Python має широкий вибір бібліотек та фреймворків, що є однією з найбільших переваг Python. Від NumPy до TensorFlow бібліотеки Python використовуються для всього, від візуалізації даних, машинного навчання, науки про дані, обробки природної мови до комплексного аналізу даних. Наявність великої бібліотеки з управлінням пам'яттю та порожніми конструкціями допомагає підвищити продуктивність роботи Python-розробника. Як результат, розробники можуть керувати базами даних, документацією, веб-браузерами; виконувати модульне тестування та багато інших функцій. Переваги великих бібліотек Python майже такі ж, як і широкий вибір фреймворків. Це може допомогти вашим проектам зрушити з мертвої точки, заощаджуючи ваш час і зусилля. Існує безліч фреймворків, таких як Pyramid, Twisted, Django, Falcon, Flask. Ці бібліотеки та фреймворки дуже допомагають розробнику при кодуванні, оскільки йому не потрібно кодувати вручну;

- Python безкоштовний. Python - це мова програмування, яка є безкоштовною та відкритою. Затверджена OSI ліцензія з відкритим вихідним кодом, за якою розробляється Python, робить цю мову вільною для використання та розповсюдження, в тому числі в комерційних цілях. Це зменшить ваші витрати на обслуговування. В той час як розробники можуть ділитися, копіювати та змінювати її. Що стосується Python-спільноти, то вона надає можливість ділитися знаннями з молодшими спеціалістами;

- Об'єктно-орієнтований. Python підтримує об'єктно-орієнтоване програмування і є процедурно-орієнтованим. Об'єктно-орієнтоване програмування в тому сенсі, що воно використовує об'єкти, які засновані на даних і функціональності. Процедурно-орієнтована функція пропонує застосовувати багаторазово використовувані фрагменти коду [12].

1.2 Аналіз наявних продуктів

Обирати необхідну систему документообігу потрібно виходячи з потреб майбутніх користувачів цієї системи, тобто знайти таку, яка буде відповідати усім необхідним користувачам характеристикам, а також не буде перенасичена непотрібними у цій галузі можливостями.

Системи електронного документообігу в основному поділяються на два види:

- СЕД – система електронного документообігу. СЕД відрізняється від ЕСМ тим, що СЕД забезпечує роботу тільки із документами, але більш орієнтовану: виконується обробка вхідних і вихідних файлів або організаційних документів. Одним з популярних підтипів таких систем є локальні, тобто ті, що необхідно встановити на персональний комп'ютер користувача, але в деяких із систем можуть використовуватися хмарні технології;

- ЕСМ – це система керування контентом. Окрім документів термін «контент» включає також і зображення, аудіофайли, відеофайли тощо. ЕСМ відповідає за надсилання та отримання контенту, його експортування, імпортування та архівування.

Залежно від цілей і завдань, розрізняють зовнішній і внутрішній документообіг. Електронний документообіг всередині організації дозволяє обмінюватися документами тільки в межах структурних підрозділів обраної компанії. Зовнішній документообіг – це обмін

вхідною та вихідною документацією з клієнтами або перевірочними органами, тобто з контрагентами.

Переваги та функції систем електронного документообігу:

- Скорочення часу на операції з документами
- Підвищення відповідальності працівників підприємства
- Поліпшення якості обслуговування клієнтів
- Прозорість всіх етапів діяльності компанії
- Зручність роботи з електронними документами

Головні питання, які вирішує система електронного документообігу:

- Правильність ведення електронного документообігу відповідно до застосованих комп'ютерних технологій;
- автоматизація файлових операцій;
- надання користувачу можливості перегляду та редагування документів;
- захист електронного документа і його реквізитів;
- наявність шаблонів, що допоможуть створити документ;
- зниження матеріальних витрат на придбання паперу;
- забезпечення мобільності системи;
- контроль за правами доступу.

Розглянемо деякі наявні приклади систем документообігу, що було знайдено в мережі:

- Perfectum ERP-box [13]. Сервіс позиціонує себе як систему для будь-якої діяльності, середнього та великого бізнесу, пропонує максимальну функціональність, можливість кастомізувати дизайн, змінювати стилі та кольори та повну безпеку даних з контролем доступу і модулем захисту по IP-адресі на будь-якому пристрої. Має більш ніж 150 готових інтеграцій та модулів доступних у магазині для розширення і так широкої функціональності системи. Вартість безстрокової ліцензії починається з

цінника 45 000 грн за систему, розраховану на 5 користувачів, а закінчується ціною в 260 000 грн, де кількість користувачів необмежена.

- Сервіс «Мій Арт-Офіс» [14]. Дозволяє створювати та підписувати документи у мінімальні строки завдяки переводу їх у електронну форму. Дозволяє отримувати доступ до документів у будь-який час та із будь-якої точки світу при наявності підключення до інтернет-мережі. Крім того, усі документи, створені або підписані через сервіс, є юридично значимими;

- М.Е.Дос [15]. Функціональний і надійний сервіс підготовки, здачі бухгалтерської та податкової звітності з великим функціоналом та зручним інтерфейсом. Без ліцензії в демоверсії можна відправити 50 документів безкоштовно. Працює лише на операційній системі Windows 7 або вище.

- Alfresco. Система має відкритий вихідний код, доступний для користувацького редагування для сумісного керування документами через Інтернет. Крім бібліотеки документів має пошукову строку, календарі, форуми, віртуальні простори для роботи та інші корисні функції. Для роботи сервісу також не потрібно встановлення додаткового програмного забезпечення на користувацький персональний комп'ютер та має один із найдосконаліших, зручних і простих для розуміння веб-інтерфейсів серед аналогів;

- FREDO:ДокМен [16]. Дозволяє створювати документи безпосередньо в сервісі за готовими шаблонами. Надає можливість відправляти електронні документи без попереднього створення їх друкованих копій. Немає безкоштовної демоверсії продукту.

- Microsoft SharePoint. Система для сумісного керування документами та роботи з ними, має тісний зв'язок з MS Exchange та MS Office та багато додатків та інтегрованих систем;

- Сервіс «Вчасно» [17]. Сервіс дозволяє управляти документами будь-яких форматів та завантажувати до 500 файлів одночасно. Документи

можуть бути підписані завдяки електронним ключам. Сервіс також дозволяє відправляти вибрані файли необмеженій кількості одержувачів (компаній-партнерів). А головною перевагою є те, що всі підписані електронним підписом документи є офіційно затвердженими, іншими словами мають юридичну силу;

- NauDoc. Програмне забезпечення із відкритим кодом, яке автоматизує бізнес-процеси і документообіг. Має можливість надати різним користувачам доступ до різних файлів, використовує пошук по тегах, змісту або ключовим словам для швидкого пошуку потрібної інформації. Шаблони дозволяють зберігати графічні образи паперових документів в системі за для більш зручного і швидкого створення файлів;

- Directum. Система, яка надає можливість керувати обігом веб-контенту та документів. Використовує веб-сервер для взаємодії з користувачами. Є версія програми для мобільних браузерів. До двохсот користувачів на систему, яка використовує хмарні технології.

Крім очевидних зручних функцій, усі перераховані системи документообігу мають деякі спільні недоліки, які роблять їх непридатними до моїх цілей. Найважливішими з недоліків є, по-перше, дуже дорога ціна за покупку підписки на сервіс, по-друге, обмежені за часом безкоштовного використання, за кількістю можливих відправлених документів або за кількістю користувачів в демоверсії програм і по-третє, системи мають забагато зайвого функціоналу, непотрібного при документообігу в навчальній сфері, такі як юридична цінність документа або цифровий підпис. Тобто після пошуку систем документообігу у відкритому доступі не було знайдено рішення для сфери освіти, яке б задовольнило усім потребам викладачів та студентів і водночас не було б перевантажено непотрібними можливостями. Програмний продукт, який буде розроблено в ході виконання кваліфікаційної роботи, має бути максимально легким для використання та водночас виконувати всі необхідні користувачам

основні функції, щоб максимально автоматизувати процес обміну документами.

2 ОСНОВНІ ТЕХНОЛОГІЇ ТА СПОСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Вибір месенджера для реалізації чат-бота

Перед початком розробки програмного додатка кваліфікаційної роботи було розглянуто основні популярні в Україні та світі месенджери, такі як Telegram, WhatsApp та Viber. Після порівняння особливостей кожного із обраних месенджерів між собою було зроблено висновок, що саме Telegram на відміну від інших варіантів пропонує програмістам більше простору для дій, а також має найбільш захищені сервери серед аналогів. Також Telegram був обраний через те, що він має найбільш комфортний, зручний та максимально інтуїтивно зрозумілий інтерфейс серед своїх конкурентів.

Іншими важливими позитивними сторонами обраного для роботи месенджера є:

- висока швидкість роботи;
- можливість використання одного облікового запису на кількох пристроях одночасно;
- доступність версій додатка для всіх основних операційних систем, як на мобільних пристроях, так і на комп'ютерах;
- відкритий API із багатьма корисними функціями, які дозволяють програмному коду та Telegram легко обмінюватися інформацією один з одним;
- можливість обміну документами будь-якого вмісту та розміром до 1 Гігабайта;
- відсутність реклами у додатку на всіх платформах;
- максимальний захист від злому серед аналогів.

Усі перелічені та ще безліч інших можливостей Telegram дозволяють прийти до висновку, що саме цей месенджер буде найбільш

оптимальним та зручним рішенням як для користувача (викладача або студента), так і для програміста, який буде взаємодіяти із Telegram за допомогою програмного коду.

Після визначення месенджера для реалізації чат-бота, була досліджена документація Telegram API для ознайомлення з базовою інформацією, яка буде корисною при написанні програмного коду. Після перегляду та ознайомлення з прикладами використання функцій та методів, наданих у документації, було вирішено, що з використанням цієї бібліотеки, програмне рішення зможе користуватися безліччю необхідних та важливих для реалізації завдання функцій, що буде корисно під час розробки програмного продукту [18-19].

Далі буде наведено деякі корисні функції або об'єкти, які можуть бути створені і використані під час розробки програмного рішення з використанням Telegram API.

1. Отримання оновлень. Це одна з багатьох корисних і необхідних функцій Telegram API. Кожного разу, коли користувач відправляє повідомлення в чат-бот, якій пізніше переводить повідомлення на сервер Telegram, програмний код продукту буде отримувати та обробляти нещодавно введену користувачем інформацію. Telegram API надає можливість використовувати різні методи отримання оновлень, наприклад `getUpdates` або “вебхук”. Кожний з них надає можливість отримувати оновлення різними способами, але після завершення операції у будь-якому випадку буде отримано об'єкт типу `Update`. Разом із новим об'єктом та отриманим значенням цього об'єкта також може отримуватися один із необов'язкових параметрів в залежності від типу запиту, який було обрано. Список параметрів та їх можливостей приведені в таблиці 2.1.

Таблиця 2.1 – Параметри об'єкту оновлення Update

Поле	Тип	Опис
message	Message	Нові вхідні повідомлення, можуть мати тип тексту, фотографій, аудіо або відеофайлу, тощо. Не є обов'язковим параметром.
update_id	Integer	Унікальний ідентифікатор для оновлення. Ідентифікатори оновлення починаються з певного додатного числа (зазвичай з одиниці) і послідовно збільшуються.
callback_query	CallbackQuery	Новий запит зворотного виклику. Не є обов'язковим параметром.
chosen_inline_result	ChosenInlineResult	Результати вбудованих запитів, які обирає користувач і надіслає співрозмовнику. Не є обов'язковим параметром.
inline_query	InlineQuery	Новий вбудований запит. Не є обов'язковим параметром.

2. Метод `getUpdates`. Метод `getUpdates` отримує оновлення, отримані після деякої дії користувача, які повертаються у форматі масиву об'єктів `Update`. Параметри методу та їх опис приведені у таблиці 2.2

Таблиця 2.2 – Параметри методу `getUpdates` та їх опис

Параметри	Тип	Опис
<code>limit</code>	<code>Integer</code>	Параметр обмежує кількість тих оновлень, що непотрібно отримувати. Значення параметру встановлюються від одиниці до сотні (сто за замовчанням).
<code>timeout</code>	<code>Integer</code>	Тайм-аут в секундах у проміжку між запитами. За замовчуванням 0, тобто звичайний короткий запит.
<code>offset</code>	<code>Integer</code>	Ідентифікатор першого оновлення, що повертається. Повинен бути більшим на одиницю за найбільший серед ідентифікаторів раніше отриманих оновлень. За замовчуванням повертається найстаріше із непідтверджених оновлень. Оновлення вважається підтвердженим, як тільки викликається <code>getUpdates</code> зі зміщенням більшим, ніж його <code>update_id</code> . Від'ємне зміщення може бути вказано для отримання оновлень, починаючи з <code>-offset</code> з кінця черги оновлень. Всі попередні оновлення будуть забуті.

3. Метод `setWebhook`. Цей метод потрібен для створення посилання на вебхука (наприклад, `https://name.com/<token>`), на який бот надсилатиме оновлення. Token містить у собі унікальний набір символів, який відомий тільки тому, хто його створив. Зроблено це для того, щоб усунути можливість стороннього втручання. Цей метод має два обов'язкові параметри, описані в таблиці 2.3, які можна передавати разом із ним.

Таблиця 2.3 – Параметри методу `setWebhook`.

Параметри	Тип	Опис
<code>url</code>	<code>String</code>	Посилання для відправлення запитів.
<code>certificate</code>	<code>InputFile</code>	Параметр завантажує відкритий ключ щоб перевірити сертифікат.

4. Метод `getWebhookInfo`, який можна використовувати для отримання інформації про деякі параметри вебхука. Параметри та їх опис наведено в таблиці 2.4.

Таблиця 2.4 – Параметри методу `getWebhookInfo`

Поле	Тип	Опис
<code>url</code>	<code>String</code>	Посилання на вебхук, яке можна залишити порожнім.
<code>has_custom_certificate</code>	<code>Boolean</code>	Повертає значення <code>True</code> , у випадку, коли вебхук використовує нестандартний сертифікат, інакше повертає <code>False</code> .
<code>last_error_message</code>	<code>String</code>	Текстове повідомлення про помилку надсилання останнього на вебхук.

Продовження таблиці 2.4

Поле	Тип	Опис
last_error_date	Integer	Час та дата останньої помилки надсилання оновлення вебхуку.
pending_update_count	Integer	Кількість оновлень, які очікують на доставку.

Також можна зазначити доступні типи даних, що використовуються в Telegram API та сконцентруватися на тих, що будуть використані у програмному додатку.

1. Тип User. Цей об'єкт представляє бота або користувача Telegram, приклад параметрів об'єкта в таблиці 2.5.

Таблиця 2.5 – Параметри типу User

Поле	Тип	Опис
id	Integer	Унікальний ідентифікатор користувача або бота
first_name	String	Ім'я бота або користувача
last_name	String	Опціонально. Прізвище бота або користувача
username	String	Опціонально. Username користувача або бота

2. Тип Chat. Цей об'єкт являє собою чат бота із користувачем та надає можливість керувати об'єктами чату. Параметри типу Chat приведені у таблиці 2.6.

Таблиця 2.6 – Параметри типу Chat

Поле	Тип	Опис
id	Integer	Унікальний ідентифікатор чату. Абсолютне значення не перевищує 1e13
type	Enum	Тип чату: "приватний", "груповий", "супергрупа" або "канал"
title	String	Опціонально. Назва, для каналів або груп
username	String	Опціонально. Username, для чатів і деяких каналів

3. Тип Message. Цей об'єкт являє собою повідомлення, відправлені користувачем до бота. Опис параметрів в таблиці 2.7.

Таблиця 2.7 – Параметри типу Message

Поле	Тип	Опис
message_id	Integer	Унікальний ідентифікатор повідомлення.
from	User	Опціонально. Відправник. Може бути порожнім у каналах.
date	Integer	Дата відправлення повідомлення (Unix time).
chat	Chat	Діалог, у якому було надіслано повідомлення.

Продовження таблиці 2.7

Поле	Тип	Опис
text	String	Опціонально. Для текстових повідомлень: текст повідомлення, 0-4096 символів.
document	Document	Опціонально. Інформація про файл.
forward_from	User	Опціонально. Для пересланих повідомлення.

4. Тип Document. Цей об'єкт являє собою файл, який не є фотографією, голосовим повідомленням або аудіозаписом. Параметри наведені в таблиці 2.8.

Таблиця 2.8 – Параметри типу Document.

Поле	Тип	Опис
file_id	String	Унікальний ідентифікатор файлу.
thumb	PhotoSize	Опціонально. Превью документа, визначений відправником.
filename	String	Опціонально. Оригінальне ім'я файлу, визначене відправником.
mime_type	String	Опціонально. MIME файлу, заданий відправником.
file_size	Integer	Опціонально. Розмір файлу.

5. Тип ReplyKeyboardMarkup. Цей об'єкт представляє клавіатуру з опціями відповіді. Параметри у таблиці 2.9.

Таблиця 2.9 – Параметри типу ReplyKeyboardMarkup.

Поле	Тип	Опис
keyboard	Масив масивів с KeyboardButton	Масив рядів кнопок, кожен з яких є масивом об'єктів KeyboardButton.
resize_keyboard	Boolean	Опціонально. Вказує клієнту підігнати висоту клавіатури під кількість кнопок. За замовчуванням False, тобто клавіатура розиуром як і стандартна клавіатура пристрою.
one_time_keyboard	Boolean	Опціонально. Вказує клієнту приховати клавіатуру після використання (після натискання на кнопку). Її, як і раніше, можна буде відкрити через іконку в полі введення повідомлення. За замовчуванням False.
selective	Boolean	Опціонально. Цей параметр потрібен, щоб показувати клавіатуру тільки певним користувачам. Цілі: 1) користувачі, які були @згадані в полі text об'єкта Message; 2) якщо повідомлення бота є відповіддю (містить поле reply_to_message_id), автори цього повідомлення.

Проаналізувавши деякі основні методи, типи та їх параметри, вже стає зрозуміло, що Telegram API пропонує широкий спектр можливостей для використання у програмному продукті. Насправді існує набагато більше корисних методів, ніж декілька вищезазначених, але були виділені саме ті, які будуть використані під час розробки програмного додатка.

Після ретельного дослідження можливостей Telegram API було остаточно виявлено, що Telegram є найзручнішою база, на основі якої може бути здійснена розробка чат-бота та додатка для автоматизації документообігу.

2.2 Вибір мови програмування для реалізації програмного продукту

В якості мови програмування для реалізації кваліфікаційного завдання як найдоступніший варіант було обрано Python. Це дозволяє чітко, компактно і організовано писати програмний код. Однією із великих переваг Python серед конкурентів є простота розуміння, тобто буде легше освоїти його нові можливості, ніж вивчити такі в аналогах. Є також інші причини, що роблять Python фаворитом при виборі мови програмування для реалізації чат-бота :

- Багатоплатформеність. Python дозволяє розробляти програмні рішення на більшості популярних комп'ютерних операційних систем, таких як Windows, Ubuntu та Linux;
- простота. Синтаксис Python дозволяє реалізувати необхідне завдання при створенні програмного продукту з використанням набагато меншої кількості строк коду ніж аналоги. Часто те, що в інших мовах програмування реалізується за 20 строк коду, у Python можна зробити за 5, що значно прискорює роботу програмного продукту;

- відсутність непотрібного синтаксису. На відміну від інших мов, Python не вимагає визначення типів даних змінної, він сам визначить чи це символ, чи ціле число або щось інше з контексту програмного коду. Крім того, що це скоротить роботу програмісту, це також зробить вихідний код більш доступним і зрозумілим для інших, хто його переглядатиме;
- універсальність. У Python є безліч додаткових бібліотек, призначених для вирішення практично будь-якої проблеми: NumPy, щоб працювати із масивами, лінійною алгеброю та машинним навчанням; SciPy для статистики, лінійної алгебри або оптимізації; Matplotlib для візуалізації графіків; Tkinter, PyQt5 або CustomTkinter для реалізації інтерфейсу користувача; Pillow для зображень; Scikit-Learn для задач машинного навчання та інтелектуальних досліджень даних, os для створення та змінення файлів, каталогів тощо [20].

Виходячи із перелічених вище та ще багатьох інших факторів, було визначено, що виконувати кваліфікаційне завдання з реалізації програмного продукту буде оптимально на Python не тільки для програміста, але й також для користувача, надавши йому усі необхідні можливості для швидкого та простого документообігу у програмному продукті [21].

2.3 Вибір інтерпретатора для написання програмного коду продукту

Перед початком реалізації програмного продукту мовою Python потрібно обрати та встановити середу розробки, де буде проходити написання програмного коду програми. Проаналізувавши найпопулярніші рішення для реалізації вихідного програмного коду було визначено їх позитивні й негативні риси.

- Notepad++. Один із найпростіших безкоштовних редакторів коду, що дозволяє редагувати правила підсвічування та візуальні стилі для різних мов програмування;
- Python Fiddle. Онлайн версія інтерпретатора, що підтримує додаткові бібліотеки Python а також надає можливості імпортування програмного коду з інших вебсайтів, надаючи на них посилання;
- Visual Studio Code – багатоплатформовий та багатофункціональний редактор програмного коду, для всіх відомих операційних систем, включаючи Windows, Mac OS X та Linux . Середня має відкритий код, що пропонує можливість адаптувати її під вимоги потрібних задач. Також VS Code має власний магазин розширень, що дозволяє знайти та завантажити багато створених користувачами додатків, налаштувати які просто навіть для програмістів-новачків [22];
- PyScripter. Безкоштовний популярний інтерпретатор програмного коду для мови Python, має інтерфейс більшістю наявних мов, працює з кодуванням Unicode і пропонує можливі виправлення помилок у програмному коді;
- PythonAnywhere. Зручний хмарний сервіс для програмування для мовою Python, із вбудованим веб-редактором кода, сумісним із такими популярними вебдодатками як Django та Flask;
- Codechef. Популярний онлайн-сервіс, позиціює себе як один із перших універсальних інструментів, що може використовуватися для написання та компіляції програмного коду. Може працювати із десятками мов програмування, однією з яких є й обраний для кваліфікаційної роботи Python;
- Repl.it. Онлайн-компілятор, який підтримує понад 40 мов програмування та дозволяє у пару кліків завантажувати додаткові пакети Python та встановлювати їх.

Проаналізувавши усі приведені вище середі розробки, та вивчивши їх плюси й мінуси, було обрано Visual Studio Code як інтерпретатор тому, що саме ця середа надає максимальний простір роботи з методами та функціями, які потребує програміст для реалізації програмного продукту. Також має простий та інтуїтивно зрозумілий програмний інтерфейс, що значно зменшує час, який необхідно витратити при створенні програмного продукту.

2.4 Розробка схеми роботи програмного продукту

Перед початком написання коду програмного продукту було вирішено створити функціональну схему роботи кінцевого програмного рішення для того, щоб далі було простіше реалізовувати необхідний функціонал програми, адже візуальний контент сприймається простіше, ніж текстовий. Схема програмного продукту, яка показує, як буде працювати програмне рішення, наведена на рис. 2.1.

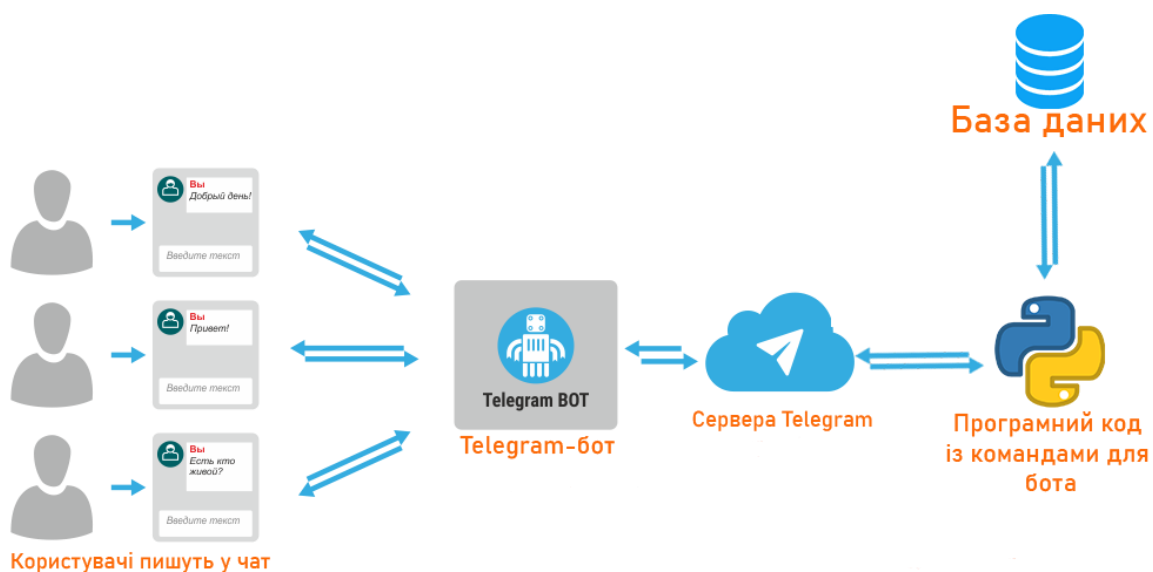


Рисунок 2.1 – Схема роботи чат-бота на базі Telegram

2.5. Вибір системи управління баз даних

Наступним кроком стане визначення, яку систему управління базами даних буде використано для створення програмного продукту,

адже між програмним додатком та чат-ботом мають передаватися деякі значення, які буде зручно зберігати в таблицях баз даних. Для цього було здійснено пошук систем для роботи з базами даних (СУБД) та порівняння цих програм між собою. Було розглянуто та проаналізовано наступні системи:

- MySQL. Найбільш відома безкоштовна система управління базами даних, з відкритим програмним кодом. Є сумісною з більшістю операційних систем а також надає можливість зберігати бази локально;
- Microsoft SQL Server. Перевагами цієї СУБД є висока продуктивність і безпека за рахунок шифрування значень в базі даних;
- SQLite. Ще одна з відомих систем управління базами даних. Дані, додані у базу, можна переглядати використовуючи будь-яку операційну систему. SQLite використовується під час розробки продуктів таких відомих компаній, як Microsoft, Google, Facebook та Apple.
- PostgreSQL. Система має відкритий програмний код, може працювати на всіх популярних операційних системах та дозволяє користувачу створювати власні типи даних або методів.

Проаналізувавши наявні варіанти доступних систем управління базами даних було обрано СУБД SQLite через її зручність і простоту, адже у програмному продукті база даних буде невеликого розміру і буде містити 2-3 таблиці. Було вирішено, що функціонал SQLite повністю задовольняє усім потребам для реалізації програмного продукту.

2.6 Вибір пакету додатків для написання користувацького інтерфейсу

Останнім кроком було вирішено проаналізувати популярні модулі для реалізації візуального інтерфейсу програмного додатка та обрати з них

оптимальний, тобто той, що надасть більше можливостей для розв'язання поставленої задачі з розробки візуалу.

При пошуку найкращого модуля для реалізації інтерфейсу програмного додатка було розглянуто три основних та найбільш популярних :

- Tkinter. Мабуть, найбільш відомий модуль інтерфейсу для мови Python. Є частиною стандартної бібліотеки додатків для Python, тобто нема потреби встановлювати додаткові пакети для роботи з бібліотекою. Tkinter має простий набір інструментів, синтаксис модуля також зовсім не складний і не потребує довгого дослідження документацій для початку написання програмного коду інтерфейсу. Однак сам візуал модуля Tkinter виглядає доволі старим, тому не є візуально приємним для користувача. Також модуль не підтримує розробку інтерфейсу в режимі Drag-n-Drop. Приклад інтерфейсу, написаного за допомогою Tkinter наведено на рис. 2.2.



Рисунок 2.2 – інтерфейс програми з модулем Tkinter

- PyQt5. PyQt5 - це повний набір зв'язок Python для Qt v5, який реалізований у вигляді більш ніж 35 модулів розширення і дозволяє використовувати Python як альтернативну мову розробки додатків до C++. Підтримує усі популярні операційні системи, до яких входить Windows, Linux, OS X для персональних комп'ютерів та iOS та Android на мобільних пристроях. Містить в собі понад тисячу класів із різними можливостями які дозволяють створювати приємний та легкий у розумінні програмний інтерфейс. Додаток має власний дизайнер, що дає можливість створювати візуальний інтерфейс простим перетягуванням необхідних елементів на форму. У порівнянні із Tkinter візуал виглядає більш приємним, більшість елементів схожі за стилем на програмний інтерфейс операційної системи Windows 7 (рис. 2.3).

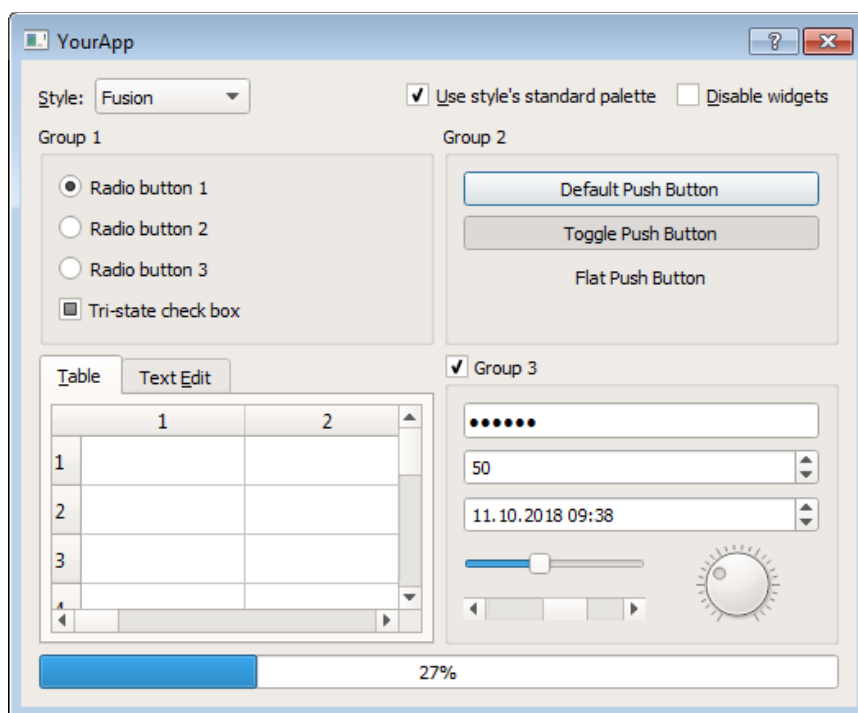


Рисунок 2.3 – інтерфейс програми з модулем PyQt5

- CustomTkinter. CustomTkinter - це Python UI-бібліотека на основі Tkinter, яка надає нові, сучасні та повністю налаштовані віджети. Вони

створюються і використовуються як звичайні віджети Tkinter, а також можуть використовуватися в поєднанні зі звичайними елементами Tkinter. Віджети та кольори вікон адаптуються до зовнішнього вигляду системи або до встановленого вручну режиму ("світлий", "темний"), і всі віджети та вікна CustomTkinter підтримують масштабування HighDPI (Windows, macOS). З CustomTkinter ви отримаєте послідовний та сучасний вигляд на всіх настільних платформах - Windows, macOS, Linux (рис. 2.4).

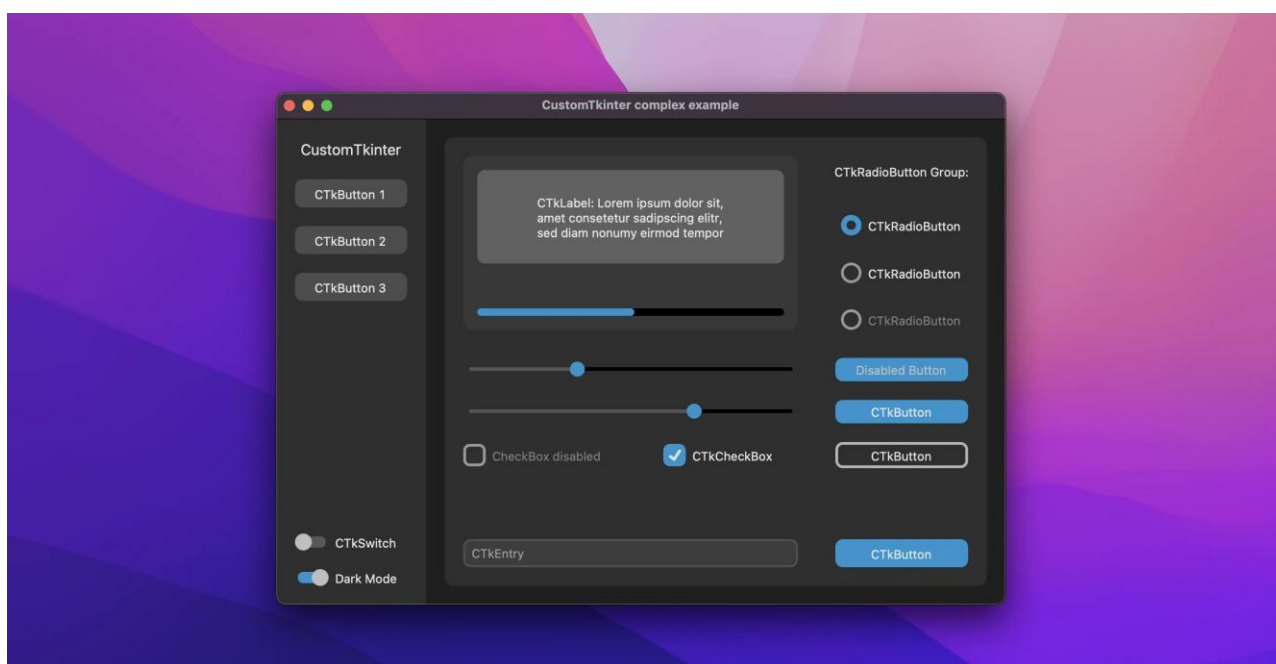


Рисунок 2.4 – інтерфейс програми з модулем CustomTkinter

Після порівняння трьох модулів для реалізації інтерфейсу, їх позитивних та негативних рис, стає очевидним, що найбільш візуально приємним для користувача модулем є саме CustomTkinter, тому при написанні програмного додатка було використано саме цей модуль [23].

Усі необхідні програмні компоненти були визначені та обрані для реалізації програмного продукту кваліфікаційного завдання, тому можна починати реалізацію самого програмного рішення.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Створення боту у Telegram

Для початку реалізації програмного проекту потрібно створити чат-бота в Telegram- месенджер. Зробити це можна прямо з самого додатку Telegram на будь-якому пристрої. Для цього потрібно в пошуковому рядку в Telegram набрати «@BotFather» або «BotFather». Після реалізації пошуку першим результатом пошукового запиту буде необхідний бот, який створила команда розробників Telegram. Він був зроблений для максимальної автоматизації створення та регулювання власних чат-ботів користувачами. Деякі можливості цього боту: вибір фото профілю чат-бота, вибір імені бота, змінення опису та статусу бота, можливість додавання команд, можливість обрати, чи матиме можливість чат-бот бути доданий у групі діалоги тощо. Діалог із ботом для створення власного чат-бота наведений на рис. 3.1.

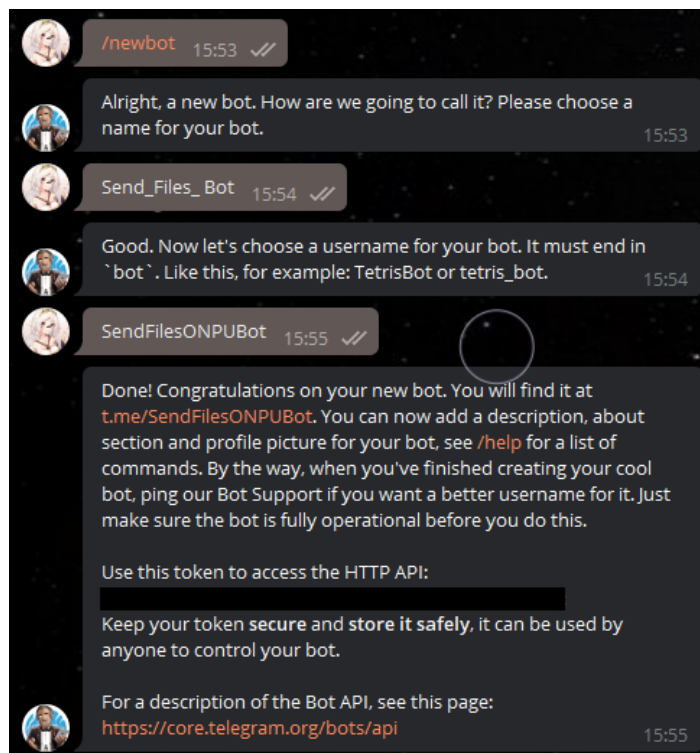


Рисунок 3.1 - Процес створення нового боту користувачем

У закритому чорним прямокутником рядку бот відправив так званий токен нового створеного користувачем бота, що є унікальним набором символів, який надає можливість керування власним ботом тому, хто є його власником (або має доступ до токена). Саме тому на рисунку він прихований, щоб не надавати доступ до створеного бота третім лицам.

Попри те, що BotFather має багато корисних команд для керування власним ботом, він не надає поглиблених можливостей керування та редагування можливостей чат-бота, тому для реалізації необхідних функцій та команд, які має виконувати чат-бот, було написано вихідний код мовою Python.

3.2 Встановлення додатка Python у середі Visual Studio Code.

Незважаючи на те, що чат-бот вже був створений, наразі він не реагуватиме на повідомлення від користувачів, адже до дій користувача не були прив'язані функції, які необхідно виконувати боту. Ці команди та функції будуть передані боту через програмний код мовою Python.

По-перше, потрібно завантажити Python останньої доступної версії. Зробити це можна через офіційний сайт python.org. Важливим пунктом під час встановлення програми обрати «Add Python to PATH» за допомогою встановлення галочки поряд із цим рядком. Якщо це не зробити, то можуть виникнути проблеми у подальшому використанні мови програмування, наприклад середа програмування може не розпізнавати встановлену мову, або не буде змоги встановлювати додаткові пакети Python.

У обраній раніше середі програмування Visual Studio Code необхідно додати можливість сприймати програмний код мови програмування Python. Для початку написання коду потрібно у боковому меню Visual Studio Code обрати пункт Visual Studio Marketplace, іконка

якого виглядає як чотири квадрати, далі у пошуковому рядку ввести назву необхідної мови програмування, яку необхідно встановити і натиснути Install для початку завантаження додатку, що зображено на рис. 3.2. Visual Studio Code є тільки інтерпретатором мови програмування і не має можливості розпізнавати програмний код та працювати із ним без додатку Python та необхідних бібліотек, саме тому заздалегідь було завантажено Python з офіційного сайту компанії.

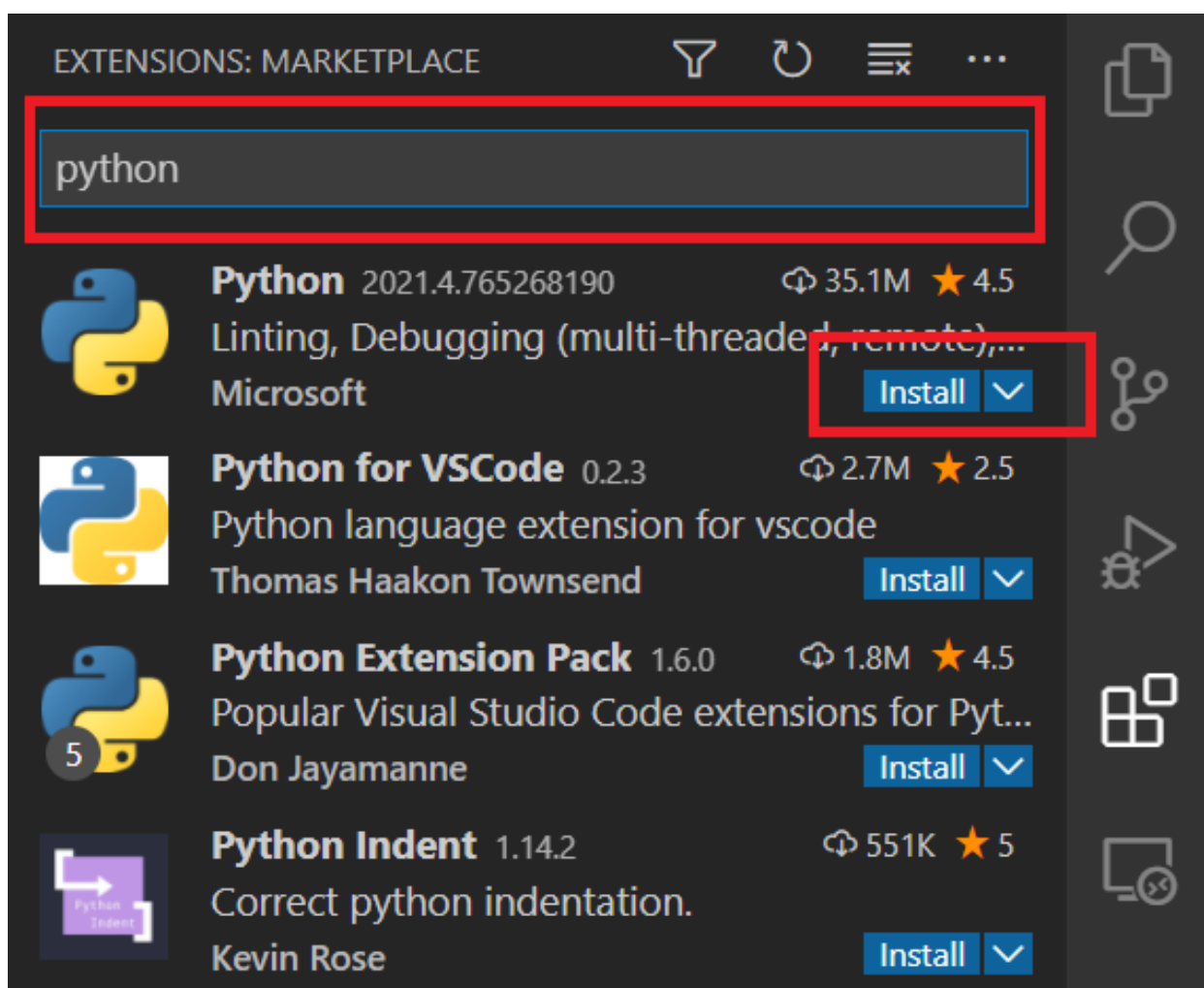


Рисунок 3.2 – Встановлення пакету мови Python у Visual Studio Code

3.3 Написання вихідного коду для чат-бота

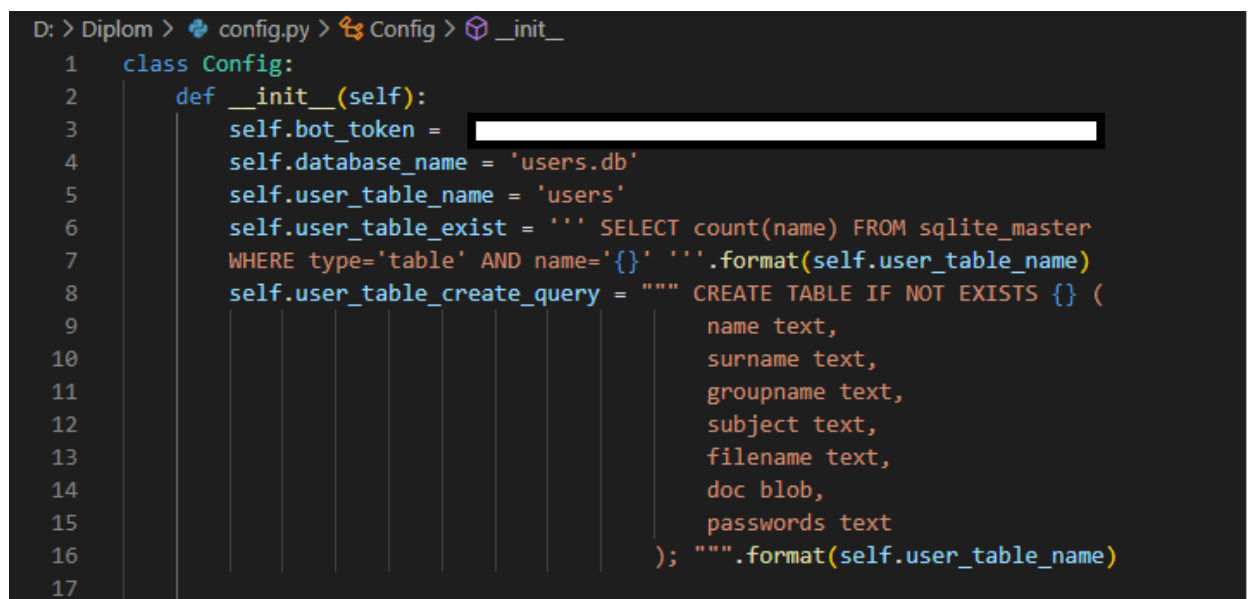
Для початку роботи та зв'язку програмного коду із чат-ботом, у програмному коді потрібно вказати токен бота. Для полегшення читання програмного коду та легшої з ним взаємодією це робиться в окремому від

основного програмного коду файлі. У цьому ж файлі було дописано код для створення бази даних “users.db”, якщо її ще не існує на пристрої, та таблиць бази даних. Таблиці, які будуть створені у базі даних, міститимуть у собі дані користувачів чат-бота. Одна таблиця буде використана для зберігання особистих даних студентів, а інша - для зберігання даних викладачів.

Таблиця users, яка містить в собі дані студентів, складається з наступних стовпчиків :

- ім'я;
- прізвище;
- група;
- предмет;
- назва файлу;
- відправлений файл у бінарному вигляді;
- пароль студента.

Вихідний код файлу зображений на рис. 3.3.



```
D: > Diplom > config.py > Config > _init_
1 class Config:
2     def __init__(self):
3         self.bot_token = 
4         self.database_name = 'users.db'
5         self.user_table_name = 'users'
6         self.user_table_exist = ''' SELECT count(name) FROM sqlite_master
7         WHERE type='table' AND name='{}' '''.format(self.user_table_name)
8         self.user_table_create_query = """ CREATE TABLE IF NOT EXISTS {} (
9         name text,
10        surname text,
11        groupname text,
12        subject text,
13        filename text,
14        doc blob,
15        passwords text
16        ); """.format(self.user_table_name)
17
```

Рисунок 3.3 – Частина програмного код файлу «config.py»

Таблиця `tusers`, в якій містяться дані викладачів, майже не відрізняється від таблиці студентів, окрім стовпчика з паролем.

Почнемо код самого файлу боту із імпортування всіх необхідних модулів, серед яких можна виділити модуль `telebot`, який буде використаний для керування чат-ботом за допомогою програмного коду, та модуль `sqlite3`, який надає можливість користуватися системою управління баз даних. Всі імпортовані модулі можна побачити на рис. 3.4

```
1 import telebot
2 import sqlite3
3 from telebot import types
4 from config import Config
```

Рисунок 3.4 – Список імпортованих модулів

Тепер потрібно створити клас `User`, який буде використовуватися для зберігання особистих даних користувачів бота. Перед відправкою документа користувач буде заповняти форму із іменем, прізвищем, групою та предметом, після чого зможе обрати необхідний файл для відправки. У таблицю бази даних будуть записуватися усі введені користувачем дані, а також назва файлу, його вміст у форматі `blob`, який пізніше буде використаний для завантажування документа іншими користувачами за допомогою додатку, та пароль користувача. Програмний код класу `User` висвітлено на рис. 3.5

```

39 class User:
40     def __init__(self, name, surname, group, subject, id):
41         self.name = name
42         self.surname = surname
43         self.group = group
44         self.subject = subject#
45         self.id = id

```

Рисунок 3.5 – Програмний код класу User

Наступним кроком було вирішено навчити бота відповідати на відповідні повідомлення від користувача. Для початку треба навчити бота реагувати на команду /start, що буде перенаправляти користувача на секцію /help, щоб дізнатися, як користуватися ботом. Далі можна обрати команду /filesend, після активації якої бот спитає користувача, чи є він студентом або викладачем і обиратиме таблицю бази даних для збереження інформації користувача відповідно до відповіді. Реалізацію у вигляді програмного коду можна побачити на рис. 3.6.

```

@bot.message_handler(content_types=['text'])
def hey(message):
    if message.text == '/start':
        bot.send_message(message.chat.id, 'Привіт, я бот. Приємно познайомитися. Щоб дізнатися, як мною користуватися, напишіть /help.')
    elif message.text == 'Привіт':
        bot.send_message(message.chat.id, 'Привіт, я бот. Приємно познайомитися. Щоб дізнатися, як мною користуватися, напишіть /help.')
    elif message.text == '/help':
        bot.send_message(message.chat.id, 'Щоб завантажити файл використайте команду /filesend ')
    elif message.text == '/filesend':
        bot.message_handler(content_types=['text'])
        def func(message):
            if(message.text == "Викладач"):
                bot.send_message(message.chat.id, text="Введіть пароль", reply_markup = types.ReplyKeyboardRemove() )
                bot.register_next_step_handler(message, sendtextteacher)
            elif(message.text == "Студент"):
                bot.send_message(message.chat.id, text='Щоб завантажити файл заповніть наступну форму: "Ім'я,Прізвище,Група,Предмет" \n')
                bot.register_next_step_handler(message, filesend)

        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        btn1 = types.KeyboardButton("Викладач")
        btn2 = types.KeyboardButton("Студент")
        markup.add(btn1, btn2)
        bot.register_next_step_handler(message, func)

    bot.send_message(message.chat.id, text="Ви викладач чи студент?".format(message.from_user), reply_markup=markup)#

```

Рисунок 3.6 – Код боту для реагування на певні повідомлення користувачів

Після визначення ролі користувача надається значення змінної в коді, яка рівна 0 якщо користувач студент і 1, якщо користувач є викладачем. Надалі це буде використано в якості умови виконання деяких

функцій, де в залежності від значення цього параметру будуть запускатися різні частини програмного коду.

Далі користувач заповнює форму з особистими даними, якщо все було заповнено вірно, тоді бот пропонує надіслати необхідний файл, після відправлення файлу бот записує дані користувача в таблицю і зберігає їх разом із документом. На цьому взаємодія користувача із ботом закінчується. Приклад процесу діалогу з чат-ботом наведено на рис. 3.7.

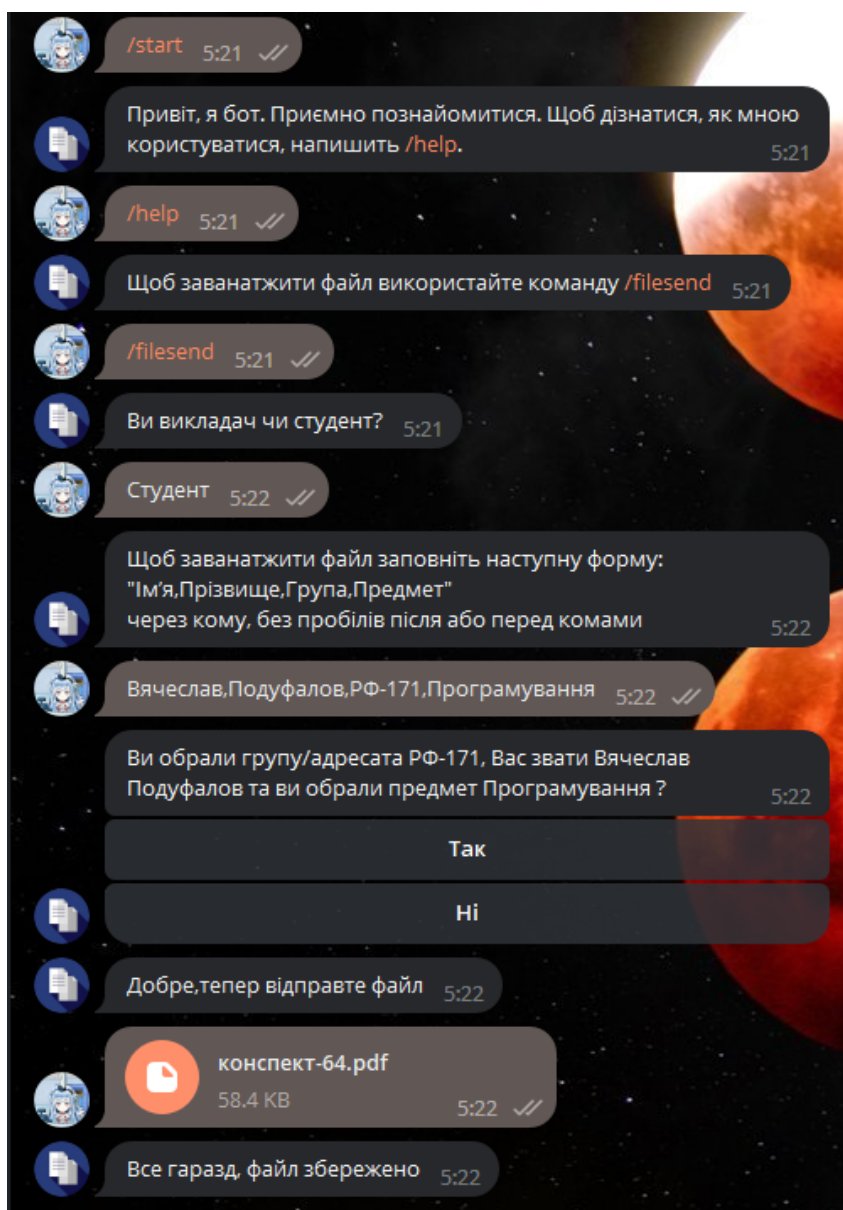


Рисунок 3.7 – Діалог користувача із чат-ботом для надсилання файлу

3.4 Розробка додатка для управління файлами

Для того, щоб почати працювати над додатком для керування документами, потрібно реалізувати інтерфейс програми. Як було зазначено раніше, інтерфейс користувача буде створюватися за допомогою додаткового пакету CustomTkinter, який візуально виглядає найбільш привабливо для користувача. Цей додатковий пакет було розроблено користувачами мови програмування для покращення можливостей інших програмістів при розробці інтерфейсу.

Тепер подивимось, як виглядає розроблений програмний додаток. Перше, що потрібно зробити користувачу при вході у програму, це обрати між викладачем та студентом. В залежності від вибору буде наданий доступ до відповідної таблиці бази даних (рис. 3.8).

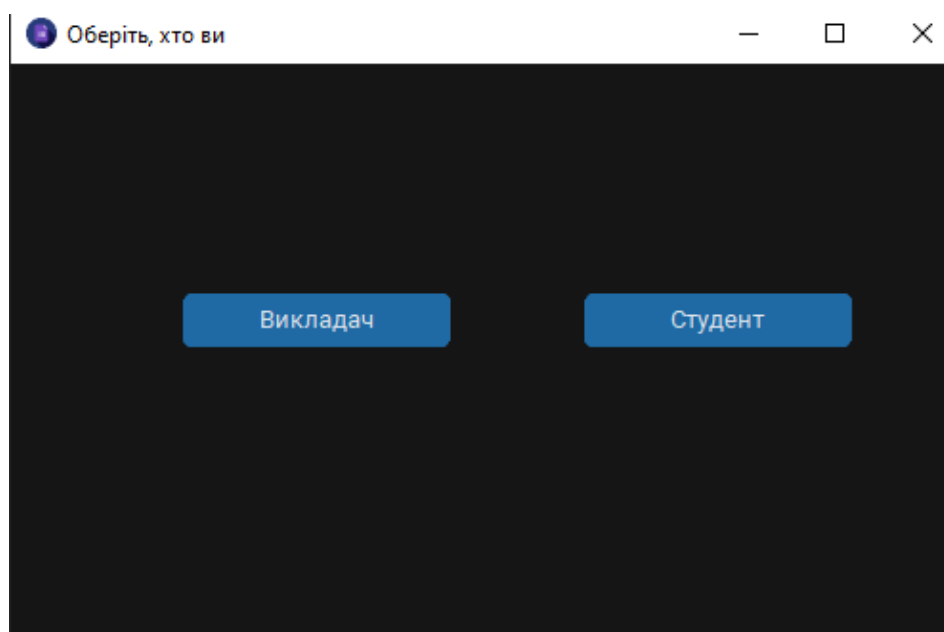
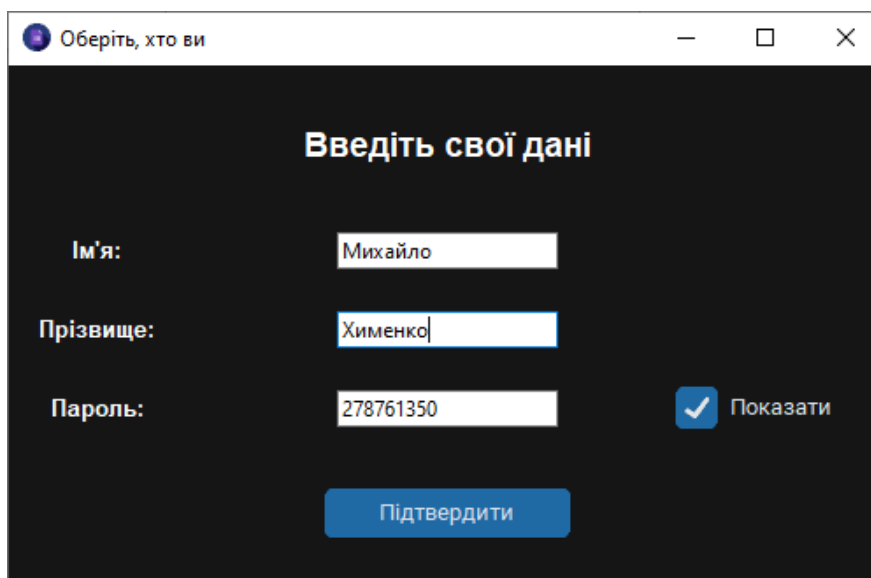


Рисунок 3.8 – Вибір між студентом та викладачем в додатку

Після вибору одного із варіантів, користувач записує свої дані та пароль, після чого програма перевіряє наявність користувача із такими даними та таким паролем. Якщо користувача було знайдено - то програма

покаже таблицю із файлами, що призначені для цього конкретного користувача або для групи, до якої він належить (якщо користувач є студентом). Програмний додаток дозволяє сортувати файли за будь яким параметром, завантажувати окремі або всі файли а також обирати шлях для збереження документів (рис. 3.9, рис.3.10).



Оберіть, хто ви

Введіть свої дані

Ім'я: Михайло

Прізвище: Хименко

Пароль: 278761350 Показати

Підтвердити

Рисунок 3.9 – Поля для введення особистих даних користувачем

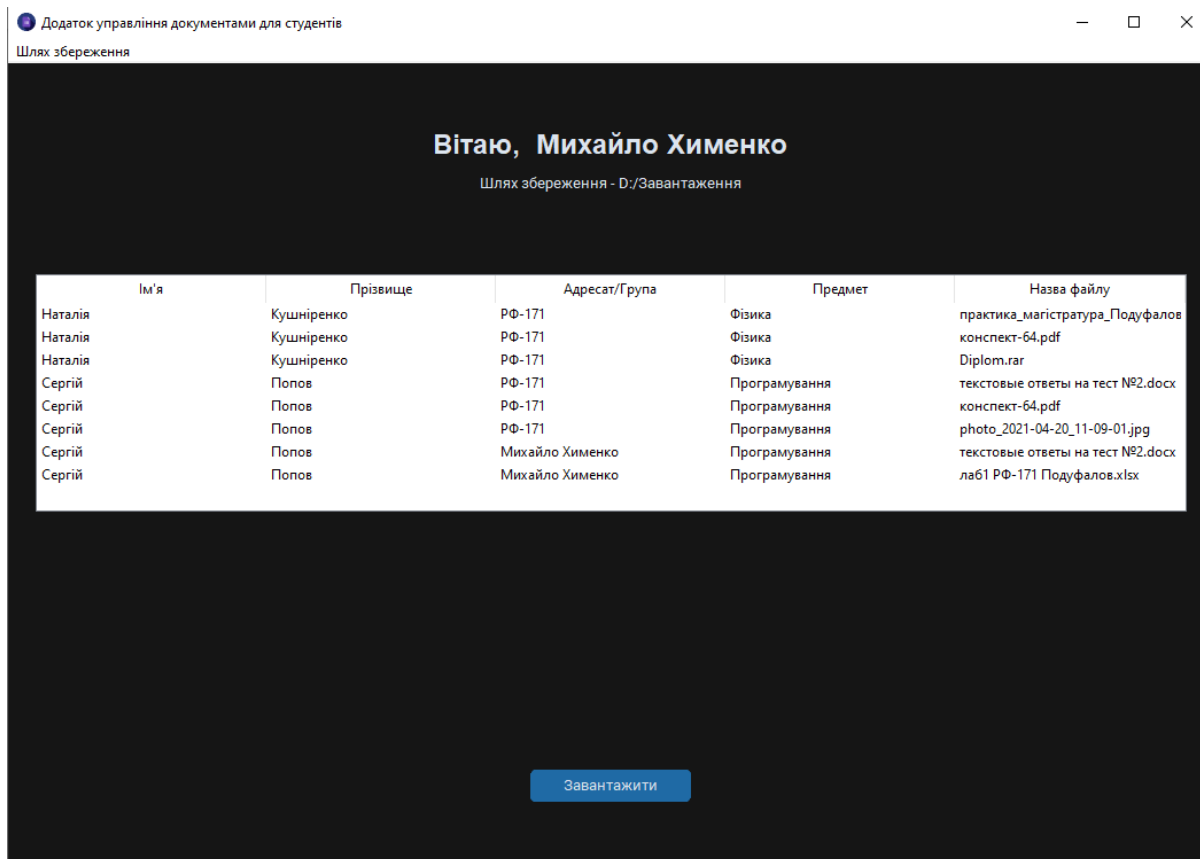


Рисунок 3.10 – Таблиця із файлами для студента

Після вибору файлів для завантаження вони автоматично розподіляються за папками. На обраному користувачем шляху створюється папка “Завантаження” у якій дані, введені користувачем стають назвами для наступних каталогів у такому порядку: Предмет, Група (або окремих студент, кому призначено файл), Ім’я та Прізвище відправника файлу, як показано на рис. 3.11.

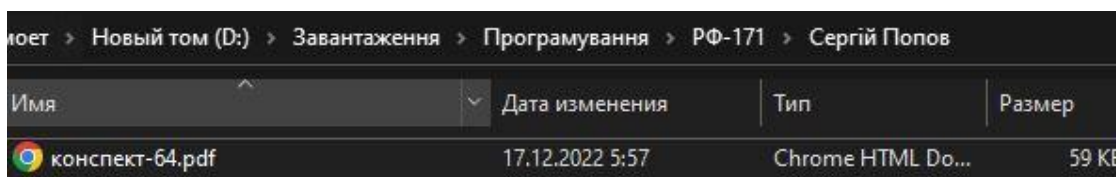


Рисунок 3.11 – Шлях збереження файлів від викладача Сергія Попова до групи РФ-171

На цьому робота користувача із додатком закінчується. У цьому розділі було продемонстровано розроблений для кваліфікаційної роботи

програмний продукт для автоматизації організації електронного документообігу. Фінальним результатом кваліфікаційної роботи стало створення чат-бота на базі месенджера Telegram, який може приймати і зберігати у базу даних документи користувачів і програмний додаток мовою програмування Python із використанням VS Code в якості інтерпретатору, модулю CustomTkinter для реалізації інтерфейсу та системи управління базами даних sqlite3 в якості СУБД.

ВИСНОВКИ

В результаті виконання завдання кваліфікаційної роботи було розроблено програмний продукт для упорядкування електронних документів студентів та викладачів університету з використанням чат-бота на основі месенджера Telegram та програмний додаток для управління та завантажування відправлених до чат-бота документів на персональні комп'ютери користувачів.

Для досягнення поставленої мети було розв'язано наступні задачі:

- виконано аналіз предметної області та наявних програмних рішень. Не було знайдено програмних рішень для сфери освіти, які б задовольняли усім потребам і водночас не були б перенавантажені зайвими у обраній сфері можливостями;
- виконано аналіз найбільш гнучких технологій для роботи із Telegram та документами. Обрано мову Python, яка має найбільш компактний програмний код та надає безліч можливостей завдяки стороннім бібліотекам. У якості системи управління базами даних було обрано SQLite через легкість використання та наявність усього необхідного функціоналу. Месенджер Telegram для реалізації чат-боту було обрано через його популярність серед студентів і викладачів а також через відкритий API, що дозволяє керувати ботом за допомогою програмного коду;
- розроблено чат-бота на основі Telegram, який має наступні можливості:
 1. приймати файли будь-якого розміру та розширення;
 2. приймати, зберігати та упорядковувати файли;
 3. розрізняти користувачів чат-боту між собою;
 4. перевіряти, яку роль має відправник файлів: студент або викладач;

- розроблено додаток для упорядкування та завантаження документів для викладачів та студентів, який має наступний функціонал:
 1. упорядкування документів за ім'ям, прізвищем, групою та предметом;
 2. використання бази даних для зберігання файлів;
 3. надання можливості вибору директорії зберігання файлів;
 4. можливість пошуку результатів по базі даних.

У підсумку можна зазначити, що розроблений програмний продукт з використанням чат-бота в Telegram може значно полегшити процес отримання файлів викладачем від студентів і навпаки тому, що користувачам більше не потрібно буде зберігати кожен файл окремо та вручну і упорядковувати їх по необхідним папкам самостійно, усе це за них тепер може робити розроблений додаток, що дасть викладачам більше часу на перевірку прийнятих робіт або на наукову та методичну діяльність, а студентам - більше часу для виконання інших завдань.

ПЕРЕЛІК ПОСИЛАНЬ

1. Pfizer. Коронавірус: загальна інформація. URL: <https://www.pfizermed.com.ua/public/covid19-first>
2. BBC News Україна. Російсько-українська війна URL: <https://www.bbc.com/ukrainian/topics/czpbw6bedqpt>
3. Zmina info. З початку вересня навчання в Україні розпочали майже 4 мільйони учнів, ще близько пів мільйона досі лишаються за кордоном. <https://zmina.info/news/z-pochatku-veresnya-navchannya-v-ukrayiny-rozpochaly-majzhe-4-miljony-uchniv-shhe-blyzko-piv-miljona-dosi-lyshayutsya-za-kordonom/>
4. Подуфалов В.В., Кушніренко Н.І., Козаченко Н.Г. Організація електронного документообігу з використанням чат-боту. Інформатика та математичні методи в моделюванні. 2022. № 3-4. URL: <http://immm.op.edu.ua>
5. Круковский М. Рішення електронного документообігу. Азимут-Україна, 2006.
6. Security Guidelines for Client Developers. URL: https://core.telegram.org/mtproto/security_guidelines
7. Шнайер Б. Прикладная криптография: протоколы, алгоритмы и исходный код на С. Київ: Диалектика, 2018.
8. Gourley D., Totty B., Sayer M., Aggarwal A., Reddy S. HTTP: The Definitive Guide, O'Reilly Media, Inc., 2002
9. Що таке API? Просте пояснення від Петра Газарова URL: <https://dev.ua/news/chto-takoe-api-prostym-yazykom>
10. Рейтинг мов програмування 2022. URL: <https://dou.ua/lenta/articles/language-rating-2022/>
11. TIOBE Index for December 2022. URL: <https://www.tiobe.com/tiobe-index/>

12. Top Advantages of Python Over Other Programming Languages. URL: <https://www.edoxitraining.com/studyhub-detail/advantages-of-python-over-other-programming-languages>
13. Perfectum ERP-box. URL: <https://perfectum.ua/ua/erp>
14. Мій Арт-Офіс. URL: <https://my.art-office.com.ua/>
15. M.E.Doc. URL: <https://aktiv.ua/ua/m-e-doc>
16. FREDO:ДокМен. URL: <https://aktiv.ua/ua/1s-fredo-docmen>
17. СЕДО «Вчасно». URL: <https://landing.vchasno.com.ua/>
18. Telegram Bot API Documentation. URL: <https://tigrm.ru/docs/bots/api>
19. Nicolas M. Building Telegram Bots: Develop Bots in 12 Programming Languages using the Telegram Bot API. Apress, 2019.
20. Мусієнко Ю. Merehead. Лучшие библиотеки Python в 2022. URL: <https://merehead.com/ru/blog/best-python-libraries-know-in-2022/>
21. Matthes E. Python Crash Course. A Hands-On, Project-Based Introduction To Programming. No Starch Press, 2019.
22. Straus D. Getting Started with Visual Studio 2019: Learning and Implementing New Features. Apress, 2019.
23. CustomTkinter. URL: <https://github.com/TomSchimansky/CustomTkinter>