

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

Дипломний проект

**здобувача освіти денної форми навчання
РП.06.02.000.ДП**

***БОРИСКОВА
ДАНИЛА ЄВГЕНОВИЧА***

**м. Одеса
2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка телеграм-боту для відстежування оновлень сучасної літератури

Проектний матеріал складається з пояснювальної записки на 72 сторінках та графічного (презентаційного) матеріалу на 11 аркушах (слайдах).

Дипломник  (Борисков Д.Є.)


Керівник  (Томаченко О.Є.)

Консультанти:

з економічної частини  (Копайгородська Т.Г.)

з охорони праці  (Чорновол Н.І.)

з дотримання вимог ЄСКД  (Петрашова Т.І.)

старший консультант  (Кунуп Т.В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Скорнякова О.В.)

Захист «23» 06 2023 р. Протокол ДКК № 2

Оцінка ДКК 5 (відмінно)

Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 "Інженерія програмного забезпечення"
Освітня програма "Розробка програмного забезпечення"

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР

Беркань І. В.

" " 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Борискову Даніілу Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка telegram-боту для відстежування оновлень сучасної літератури

затверджена наказом по коледжу від "17" жовтня 2022 р. № 235-А2-000

2. Термін здачі закінченого проекту (роботи) 12.06.2023р

3. Вихідні дані до проекту (роботи) 1. У якості месенджера для боту використовувати Telegram; 2. Телеграм бот повинен: Сповіщати користувача про вихід нових глав твору, мати можливість відстежувати більше 1 твору, мати можливість відписуватися від творів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Актуальність обраної тематики дипломного проекту. Огляд та порівняння з існуючими рішеннями

Вибір технологій розробки боту

Розробка алгоритмів відстежування оновлень сучасної літератури

Розробка програмного забезпечення для відстежування оновлень сучасної літератури

Тестування функцій розробленого чат-боту

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Структура проекту; Схема запитів до API MangaDex; Схема сортування отриманих даних з API для визначення






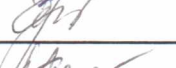
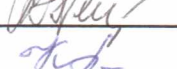
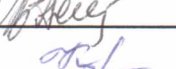
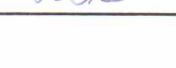
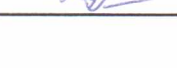
необхідних; Схема відстежування коміксів у проекті; Схема відстежування новел у проекті; Загальна схема

відстежування літератури у проекті; Робота функції novela_all_data() для отримання інформації про новели;

Структура бази даних і таблиць 'mangas' і 'novels' дипломного проекту; Структура таблиці 'users' у базі даних;

Демонстрація кнопочового головного меню бота.

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Томаченко О. Є.		
Економічна частина	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кунуп Т.В.		

7. Дата видачі завдання 01.05.2023

Керівник


(підпис)

Завдання прийняв до виконання


(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка задач	06.04.2023	Виконано
2	Опис загальної проекту	08.04.2023	Виконано
3	Вибір інструментів для реалізації проекту	10.04.2023	Виконано
4	Міркування над інтерфейсом боту та вимогами до проекту	11.04.2023	Виконано
5	Вивчення документації для API	16.04.2023	Виконано
6	Розробка блок-схеми до проекту	17.04.2023	Виконано
7	Вивчення матеріалу по необхідним бібліотекам	22.04.2023	Виконано
8	Розробка функцій запиту до API	25.04.2023	Виконано
9	Розробка функцій обробки отриманої інформації	29.04.2023	Виконано
10	Розробка алгоритмів сповіщення користувача	02.05.2023	Виконано
11	Підготовка презентації на основі виконаної роботи	06.05.2023	Виконано
12	Пошук інформації щодо охорони праці	10.05.2023	Виконано
13	Підготовка графічної частини дипломного проекту	20.05.2023	Виконано
14	Підготовка економічної частини. Розрахунки	27.05.2023	Виконано
15	Підготовка до захисту проекту.	05.06.2023	Виконано

Дипломник


(підпис)

Керівник


(підпис)

Зміст

ВСТУП.....	7
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	8
1.1 Аналіз предметної області. Мета і обґрунтування дипломного проекту	8
1.2 Актуальність обраної тематики дипломного проекту. Огляд та порівняння з існуючими рішеннями	9
1.2.1 Розбір аналогів проекту. Push-сповіщення у браузері	10
1.2.2 Прямі аналоги. Існуючі телеграм-боти для відстежування оновлень сучасної літератури.....	10
1.2.3 Огляд додатків для читання і відстежування	11
1.3 Вибір технологій розробки боту	13
1.3.1 Вибір мови програмування для реалізації проекту	14
1.3.2 Вибір середовища розробки для реалізації дипломного проекту.....	16
1.3.3 Вибір API та бібліотек для реалізації дипломного проекту.....	17
1.4. Розробка алгоритмів відстежування оновлень сучасної літератури.....	22
1.4.1 Опис функцій, які необхідно спроектувати для роботи з джерелами.....	27
1.4.2 Опис функцій, які необхідно спроектувати для роботи з базою даних.....	29
1.5 Опис етапів розробки проекту. Опис функціональності проекту....	30
1.5.1 Перший етап створення телеграм-боту, робота з BotFather	30
1.5.2 Структура кореневої папки проекту та встановлення бібліотек	32
1.5.3 Структура бази даних.....	34
1.5.4 Створення класів для відстежуваних витворів.....	36
1.5.5 Приклад функцій бота.....	37

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

2 ЕКОНОМІЧНА ЧАСТИНА.....	45
2.1 Резюме.....	45
2.2 Визначення трудомісткості розробки програмного забезпечення...	45
2.3 Розрахунок ціни програмного продукту.....	48
3 ОХОРОНА ПРАЦІ.....	51
3.1 Аналіз небезпечних та шкідливих чинників на робочому місці.....	52
3.2 Гігієнічні вимоги до виробничого середовища.....	53
3.3 Вимоги до організації робочого місця працівника.....	54
3.4 Пожежна безпека.....	54
ВИСНОВКИ.....	56
Перелік використаних джерел.....	58
ДОДАТОК А. Робота нескінченного циклу для відстежування оновлень сучасної літератури (мова Python).....	59
ДОДАТОК Б. Функції для додавання в базу даних нових творів та прив'язки творів до користувачів.....	62
ДОДАТОК В. Презентація.....	67

ВСТУП

Сучасні інформаційні технології та месенджери стали невід'ємною частиною нашого повсякденного життя. Вони забезпечують нам миттєвий зв'язок з іншими людьми та надають доступ до різних сервісів і контенту. Одним із таких популярних месенджерів є Telegram, який вирізняється своїми широкими можливостями для розробників.

У рамках цього дипломного проекту пропонується розробка телеграм-бота, спеціально призначеного для відстеження виходу нових глав сучасної літератури. Сучасна література упродовж десятиліть користується величезною популярністю і має мільйони шанувальників по всьому світу. Однак стежити за регулярними оновленнями і виходом нових глав може бути непросто, особливо з огляду на кількість випущених серій, величезну кількість ресурсів для читання, нерегулярність перекладу і велику кількість цих самих перекладачів.

Метою даного проекту є створення інтелектуального телеграм-бота, який автоматизує процес відстеження та надання інформації про вихід нових глав сучасної літератури. Бот буде інтегрований з платформою Telegram, що дасть змогу користувачам отримувати актуальні сповіщення та інформацію про вихід нових розділів на будь яких пристроях зі встановленим месенджером.

Основний акцент у цьому проекті буде зроблено на розробці ефективних алгоритмів збору та обробки даних, аналізі контенту, а також створенні зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. В якості основної мови програмування для реалізації проекту обрано Python, який є потужним і гнучким інструментом для розробки телеграм-ботів.

Цей дипломний проект є унікальною можливістю об'єднати в собі навички програмування, аналізу даних і розробки користувацького інтерфейсу. Він має великий потенціал для спрощення і поліпшення процесу отримання інформації про вихід нових глав сучасної літератури, що значно полегшить життя її фанатам і стане корисним інструментом для всіх, хто цікавиться цим популярним видом розваги.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Аналіз предметної області. Мета і обґрунтування дипломного проекту

Основна мета полягає у створенні чат-бота для платформи *Telegram*. Спочатку необхідно розробити привітальне повідомлення, а потім за допомогою програмного коду створити кнопки для забезпечення навігації користувача у системі. Перш за все, створюється п'ять основних кнопок "Підписатися на оновлення літератури", "Відписатися від оновлень", "На що користувач підписан", Вибір мови та Зв'язок з розробником.

Наступним кроком є створення інших меню з простішими кнопками, що вказують на цифру для подальшого вибору з варіантів після пошуку. Наприклад, при натисканні кнопки "Підписатися на оновлення літератури" користувачеві буде запропоновано вибрати жанр, а потім ввести назву твору. Після цього користувачеві пропонується вибрати слот для зберігання. При натисканні кнопки "Відписатися" користувачу відобразатиметься список всіх творів, на які він підписаний, і він матиме можливість відписатися від них. Функціонал кнопки "На що я підписаний" аналогічний, за винятком того, що відписатися неможливо.

Кнопка "Мова" дозволить користувачу змінювати мову інтерфейсу бота, а кнопка "Зв'язок з розробником" надаватиме посилання на *Telegram*-акаунт розробника. Всі літературні твори та дані користувачів зберігатимуться у базі даних, яка буде реалізована за допомогою системи управління базами даних *MySQL*.

Технічне завдання на розробку передбачає виконання ботом наступних функцій

Сповіщати користувача про вихід нових глав твору протягом 15 хвилин після публікації.

Можливість відстежувати кілька творів одночасно.

видкий пошук потрібного твору.

Можливість відписуватися від оновлень твору.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		8

Можливість зв'язатися з розробником.

Можливість змінити мову інтерфейсу бота.

Основна ідея полягає у створенні зручного та корисного чат-бота, який спростить взаємодію користувача з літературними творами та надасть йому широкий функціонал.

1.2 Актуальність обраної тематики дипломного проекту. Огляд та порівняння з існуючими рішеннями

Існує низка платформ і веб-сайтів, які надають комікси і новели мовами нашого регіону, і деякі з них публікують статистику про свою аудиторію і перегляди контенту. Наприклад, платформа `read an a`, одна з найпопулярніших на пострадянському просторі, повідомляє про мільйони унікальних відвідувачів і мільярди переглядів щомісяця. Це вказує на значний інтерес і активність користувачів у читанні манги онлайн.

Також варто зазначити, що форуми, групи в соціальних мережах і спільноти, присвячені коміксам та новелам, також слугують показниками їхньої популярності та активності. Наприклад, спільнота `i e i`, що спеціалізується на обговоренні літератури, включно з мангою та новелами, має мільйони зареєстрованих користувачів і тисячі активних обговорень.

`Man adex` на відміну від інших ресурсів, які використовуються не обмежується якоюсь однією мовою, а підтримує всі існуючі у світі мови, тому статистика буде для усього світу. Сайт `Man a ex` на сторінці `Про нас` повідомляє наступну інформацію

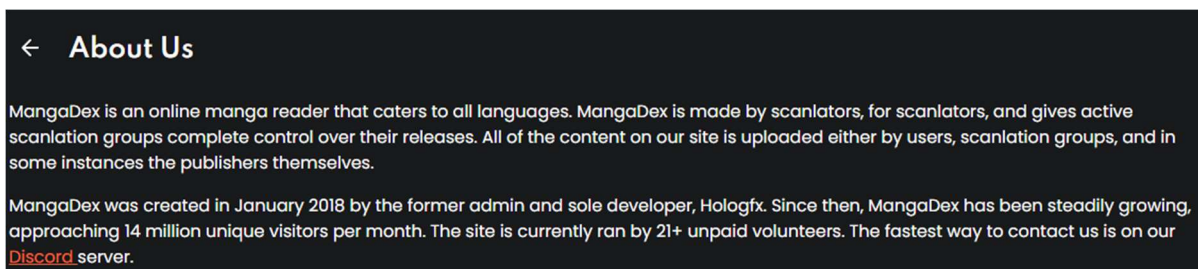


Рисунок 1.1. Сторінка `Про нас` на сайті `Man a ex`

					РП 06. 02 00 . 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		9

Тобто на цьому сайті щомісяця є приблизно 14 мільйонів унікальних відвідувачів.

Таким чином, незважаючи на відсутність точних статистичних даних, доступні дослідження, дані платформ і активність в онлайн-спільнотах свідчать про значну кількість людей, які читають комікси і новели онлайн на пострадянському просторі і у всьому світі. Популярність цих форматів розваги продовжує зростати, а доступність онлайн-ресурсів і зручність читання за допомогою смартфонів і планшетів сприяють їхньому поширенню серед широкої аудиторії.

1.2.1 Розбір аналогів проекту. Push-сповіщення у браузері

Багато різних сайтів пропонують підписатися на витвір. Якщо це зробити, то користувачу будуть надходити сповіщення від браузера, коли новий розділ вийде. Послідовність дій, які потрібні для підписки дуже схожі на багатьох сайтах

1. Для підписки на твір необхідна реєстрація на сайті.
2. Необхідно додати твір до списку. Зазвичай на більшості сайтів список називається *Читаю*.
3. Увімкнути в налаштуваннях акаунту функцію push-сповіщень.
4. Дозволити браузеру відправляти сповіщення.

Спосіб з push-сповіщення має ряд недоліків

Необхідність реєстрації на сайтах для включення подібного функціоналу.

Сповіщення які надсилає браузер можуть бути зручні не всім читачам.

Прив'язка до пристрою.

Сповіщення приходять тільки з одного сайту, а не з того, на якому новий розділ вийде швидше за інших.

1.2.2 Прямі аналоги. Існуючі телеграм-боти для відстежування оновлень сучасної літератури

На форумі одного з ресурсів, знайшлося таке обговорення

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		10



Рисунок 1.2. Обговорення на форумі an a i про створення телеграм-боту для відстежування нових розділів.

декілька відмінностей

Бот перевіряє наявність нових розділів тільки на одному сайті, коли як бот з дипломного проекту буде перевіряти на багатьох.

Інтерфейс через команди, а не через кнопки, які більш зручні для користувача

Бот перевіряє наявність нових розділів тільки в одному сучасному літературному жанрі (азіатські комікси).

Бот підтримує лише російську мову.

Ну і найголовніший мінус - проект був розпочатий 2 роки тому і зараз не працює.

1.2.3 Огляд додатків для читання і відстежування

Проблема розроблюваного боту, що він працює лише для сповіщень, без можливості прямо у додатку прочитати твір, але по перше, телеграм-бот і не є додатком, який потрібно завантажувати, а по друге, це не зовсім так, тому-що у месенджері телеграм є вбудований браузер у якому і можна буде переглянути новий матеріал. Хоча це, можливо, не так зручно як у спеціальних додатках. Додатки, котрі одночасно є і так званими "читалками" для сучасної літератури. Саме такі додатки є найбільшими конкурентами для розроблюваного бота, якщо у них вбудована функція сповіщення, але мінусів в зрівнянні теж немало

Прив'язка до одного додатку.

Прив'язка до одного пристрою.

Необхідність завантаження додатку.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		11

Скоріш за все також треба буде пройти реєстрацію.

Однак є і плюси, наприклад необмежена кількість відстежуваних творів і можливість зручно прочитати розділ у додатку, а не переходити за посиланням на сайт, до того-ж, іноді один розділ в боті можна буде прочитати на одному сайті, а наступний вже на іншому.

Для дослідження на смартфон був завантажений подібний додаток. Для пошуку такого, було введено в пошуковому рядку слово "ранобе"

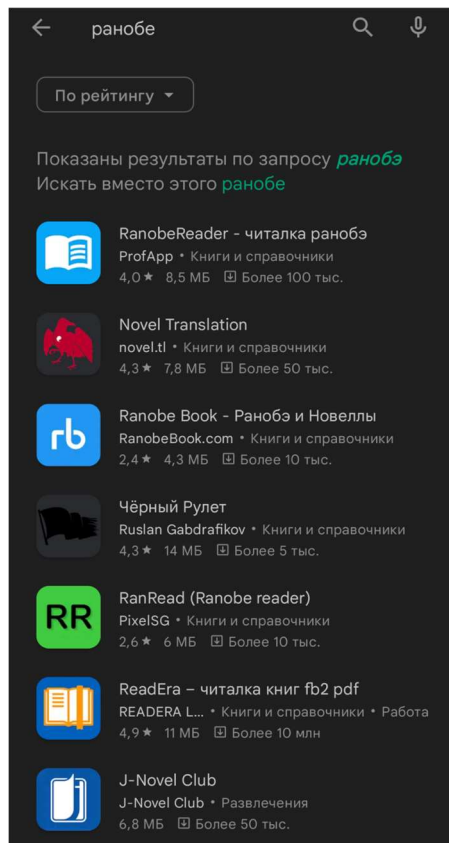


Рисунок 1.3. Результати пошуку у Play Market за словом ранобе

Перший ж результат, наскільки видно з опису підтримує відстежування

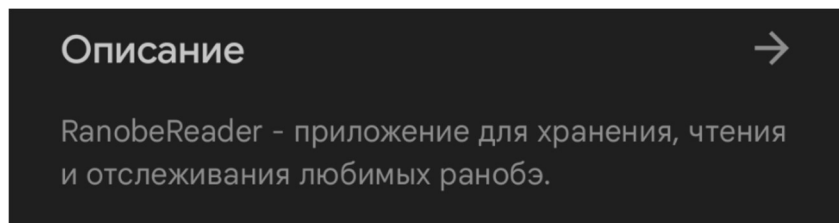


Рисунок 1.4. Опис додатку RanobeReader.

Також цей додаток, як і розроблюваний бот відстежує вихід нових розділів з багатьох різних ресурсів

					РП 06. 02 00 . 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		12

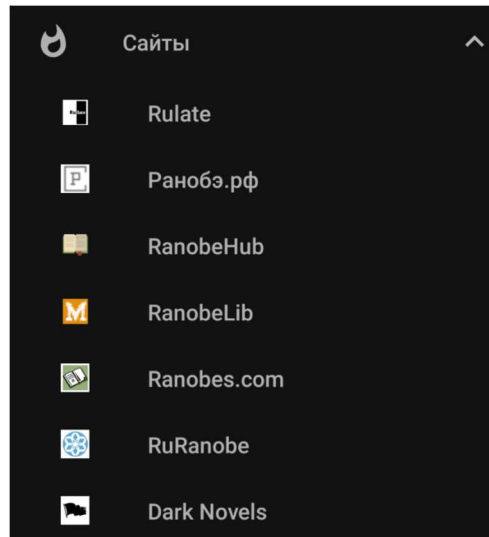


Рисунок 1.5. Демонстрація списку використаних джерел у додатку

Серед однозначних мінусів - тільки російська мова, та відстежування тільки сучасної текстової літератури. Для відстежування наприклад коміксів треба завантажувати ще один додаток. Реєструватись не довелося, що є плюсом. Додаток має вбудований читач нових розділів і важить мало (трохи більше 8 мегабайт). Однак, це все ще сторонній додаток, коли як месенджер Telegram є у більшості людей. Також цей додаток є тільки для смартфонів, коли як месенджер Telegram можна встановити і на ПК, або взагалі користуватися веб версією.

Розроблюваний бот буде надсилати сповіщення зручніше за всіх можливих конкурентів і охоплювати майже усі жанри сучасної літератури на відміну від існуючих рішень, саме тому цей проект і є дуже актуальним.

1.3 Вибір технологій розробки боту

Кожна мова програмування має свої переваги, недоліки та сфери застосування. Вибір мови програмування залежить від вимог проекту, типу завдання, доступності ресурсів і досвіду розробника. Розробники можуть обирати між різними мовами залежно від своїх потреб та уподобань.

Telegram боти можуть бути написані різними мовами програмування, залежно від уподобань розробника. Telegram надає API для створення ботів, який підтримує кілька мов програмування. Деякі з популярних мов, які широко використовуються для створення telegram ботів, включають

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		13

Можна зробити висновок, що Python зараз є найбільш популярною мовою програмування, а це значить, що з величезною кількістю проблем та помилок при розробці, з якою розробник може потенціально зустрітися, вже зустрічалися інші люди і ефективність рішення цих проблем буде вище, ніж в інших мовах. Серед інших причин вибору цієї мови можна назвати

Простота і зручність використання Python має простий і лаконічний синтаксис, що робить його дуже читабельним і зрозумілим. Це дає змогу розробникам швидко писати і підтримувати код, що особливо корисно при створенні та підтримці телеграм-ботів.

Велика кількість бібліотек і фреймворків Python має велику екосистему бібліотек і фреймворків, які значно спрощують розробку телеграм-ботів. Наприклад, для роботи з Telegram API існують популярні бібліотеки, як-от `python-telegram-bot` - `aiogram`, які надають зручний інтерфейс для взаємодії з Telegram.

Підтримка Telegram Bot API Python надає потужний і гнучкий API для розробки ботів. Python має хорошу підтримку для роботи з Telegram Bot API, що дає змогу розробникам легко створювати й налаштовувати телеграм-ботів із використанням різних функцій, як-от надсилання повідомлень, опрацювання команд, робота з клавіатурами тощо.

Розширюваність Python має можливість інтеграції з іншими мовами програмування, такими як C++, що дає змогу використовувати продуктивні бібліотеки та модулі для оптимізації роботи телеграм-ботів. Це особливо корисно, коли потрібне виконання складних обчислень або обробка великих обсягів даних.

Кросплатформеність Python є кросплатформенною мовою, що означає, що код, написаний на Python, працюватиме на різних операційних системах, таких як Windows, macOS і Linux. Це дозволяє розробникам створювати телеграм-боти, які можуть працювати на будь-якій платформі.

Python надає розробникам потужні інструменти та простоту використання, що робить його прекрасним вибором для розробки телеграм-ботів.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		15

1.3.2 Вибір середовища розробки для реалізації дипломного проекту

Для реалізації дипломного проекту було обрано середовище `Visual Studio Code`. Основні плюси `Code` включають

Легкість і швидке завантаження `Code` має невеликий розмір і швидко швидкість роботи, що забезпечує комфортну розробку без затримок.

Підтримка безлічі мов програмування `Code` підтримує велику кількість мов програмування, включно з популярними, як-от `JavaScript`, `Python`, `C#`, `C++`, `Java`, і багато інших. Це робить його підходящим для розробки дипломного проекту.

Інтеграція з `Git` `Code` має вбудовану підтримку системи контролю версій `Git`, що полегшує роботу з репозиторіями, комітами, порівнянням змін і управлінням гілками.

Інструменти налагодження `Code` надає зручні інструменти налагодження, що дають змогу розробникам перевіряти та виправляти помилки у своєму коді. Він підтримує налагодження для різних мов і дає змогу встановлювати точки зупинки, переглядати змінні та виконувати кроки налагодження.

Розширюваність через плагіни `Code` має широкі можливості для розширення функціональності за допомогою плагінів. Розробники можуть встановлювати різні розширення, що додають нові можливості, теми оформлення, інтеграцію з іншими інструментами та багато іншого.

Активна спільнота `Code` має велику й активну спільноту розробників, які діляться своїми напрацюваннями, створюють розширення і пропонують допомогу. Це дає змогу швидко знаходити розв'язання проблем, отримувати підтримку та обмінюватися досвідом з іншими розробниками.

Загалом, `Visual Studio Code` є зручним, легким і гнучким середовищем розробки, яке може бути застосоване для різних мов програмування і типів проектів.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		16

1.3.3 Вибір API та бібліотек для реалізації дипломного проекту

У проекті використовується безліч бібліотек, та 2 API.

Для роботи з Telegram Bot API використовується бібліотека Telegram Bot. Telegram Bot надає простий та інтуїтивно зрозумілий інтерфейс для взаємодії з Telegram Bot API. Завдяки цьому, почати роботу з бібліотекою і створення простих телеграм-ботів може бути досить легким і швидким процесом. Також бібліотека має великий функціонал. Telegram Bot пропонує широкий набір функцій, які дають змогу легко керувати і налаштовувати телеграм-бота. Бот зможе надсилати повідомлення, обробляти команди, працювати з клавіатурами, керувати файлами і медіа-ресурсами, а також багато іншого. Це дає змогу створювати більш складні та інтерактивні телеграм-боти з мінімальними зусиллями. Бібліотека підтримує оновлення Telegram Bot API. Telegram Bot зазвичай активно оновлюється для підтримки останніх функцій і змін у Telegram Bot API. У бібліотеки є детальна документація та велика спільнота, яка може надати підтримку та допомогу в разі виникнення запитань чи проблем.

Як одним з API який використовується в проекті є Manga API, це інтерфейс програмування додатків, що надається Manga API, популярною онлайн-спільнотою для читання манги. API надає розробникам доступ до різних функцій і даних, пов'язаних із мангою, які можна використовувати для створення застосунків, інтеграції з іншими сервісами або автоматизації певних завдань. За допомогою Manga API розробники можуть отримувати інформацію про мангу, таку як назви, описи, обкладинки, жанри, автори, глави, сторінки та багато іншого. Також доступні функції пошуку, що дають змогу шукати мангу за різними критеріями, наприклад, за назвою, жанром або автором. API також надає можливість отримувати інформацію про користувачів, коментарі та переклади. Використання Manga API дає змогу розробникам інтегрувати функціональність Manga API у свої додатки, створювати призначені для користувача інтерфейси для читання манги, створювати статистику або сповіщення про нові глави та випуски, а також проводити аналіз і обробку даних манги.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		17

Для роботи з іншими ресурсами використовується парсинг, тобто процес вилучення даних з вихідного коду або структурованого тексту з метою отримання потрібної інформації. У контексті веб-розробки та даних мережі Інтернет, парсинг зазвичай відноситься до вилучення інформації з веб-сторінок або інших джерел даних. Парсинг виконується з використанням спеціальних програмних інструментів, які називаються парсерами, які аналізують структуру вихідного коду або тексту, розбиваючи його на більш дрібні елементи і витягуючи необхідні дані.

Для парсингу даних в проекті використовується бібліотека `BeautifulSoup4`. Це бібліотека для вилучення даних з HTML і XML документів. `BeautifulSoup4` є хорошим вибором, тому що

Простота використання `BeautifulSoup4` надає простий і зрозумілий API, який дає змогу легко витягувати дані з HTML -коду. Він має інтуїтивний синтаксис, який робить процес парсингу більш доступним навіть для новачків.

Гнучкість `BeautifulSoup4` може працювати з неправильним або складним HTML -кодом, автоматично виправляючи його і надаючи консистентну структуру даних для вилучення інформації. Він дає змогу обробляти різні типи тегів, атрибути та вміст, що робить його гнучким інструментом для різних сценаріїв парсингу.

Підтримка `Unicode` `BeautifulSoup4` надійно обробляє різні символи `Unicode`, що особливо важливо під час роботи з багатомовними веб-сторінками або текстовими даними.

Інтеграція з іншими бібліотеками `BeautifulSoup4` легко інтегрується з іншими бібліотеками Python, як-от `requests` (для завантаження веб-сторінок) і `Pandas` (для аналізу даних). Це дає змогу створювати потужні скрипти для автоматизації збору та обробки даних.

`requests` є ще однією бібліотекою, яка використовується в проекті. Це популярна і проста у використанні бібліотека для надсилання HTTP-запитів у мові програмування Python. Вона надає зручні методи для роботи з різними типами запитів, підтримує HTTP і SSL-сертифікати, дає змогу працювати із сесіями та

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		18

кукісами, а також обробляє отримані HTTP-відповіді. Бібліотека `requests` широко застосовується під час веб-скрапінгу, взаємодії з API та інших сценаріїв розробки.

Для того, щоб нескінченний цикл у проекті не заважав боту опрацьовувати команди, треба цей цикл помістити у інший потік. Для багатопоточності використовується бібліотека `threading`. `threading` - це модуль у мові програмування Python, який дає змогу створювати та керувати потоками виконання в багатопотокових додатках. Потоки (threads) являють собою легковагі підзадачі, які можуть виконуватися паралельно всередині одного процесу, забезпечуючи багатозадачність.

Python забезпечує підтримку багатопоточності за допомогою модуля `threading`, який надає високорівневий інтерфейс для створення та керування потоками. Використання потоків може бути корисним у ситуаціях, коли кілька завдань мають виконуватися паралельно для поліпшення продуктивності або реагування на асинхронні події.

Основні поняття та функції, що надаються модулем `threading`, включають

Створення потоків Модуль `threading` дає змогу створювати потоки шляхом створення підкласів від класу `Thread` і перевизначення методу `run()`, у якому визначається код, що виконується в потоці.

Запуск потоків Створені потоки можна запустити за допомогою методу `start()`, який запускає виконання методу `run()` в окремому потоці.

Синхронізація потоків Модуль `threading` надає різні примітиви синхронізації, як-от блокування (`Lock`), умови (`Condition`), семафори (`Semaphore`) тощо, що дають змогу керувати доступом до спільних ресурсів із різних потоків.

Керування життєвим циклом потоків Методи, як-от `join()` і `is_alive()`, дають змогу керувати життєвим циклом потоків, чекати на їхнє завершення і перевіряти їхній статус.

Робота з глобальними змінними Під час роботи з кількома потоками необхідно враховувати синхронізацію доступу до глобальних змінних, щоб уникнути станів перегонів. Модуль `threading` надає відповідні механізми для

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		19

безпечної роботи зі спільними даними.

Потоки можуть бути корисними для виконання завдань у фоновому режимі, обробки паралельних обчислень, очікування введення виводу та інших сценаріїв, де потрібне паралельне виконання коду. Неправильне використання може призвести до станів перегонів, блокувань та інших проблем синхронізації.

Для того, щоб нескінченний цикл не відправляв запити до бази даних та API постійно, а лише кожні 15 хвилин, я використовую бібліотеку `time`. Бібліотека `time` в Python надає функції та класи для роботи з часом і затримками. Вона дає змогу отримувати поточний час, вимірювати проміжки часу, задавати затримки у виконанні програми та багато іншого. Вона є частиною стандартної бібліотеки Python і широко використовується для керування часом і виконання затримок у програмі.

Для роботи з базами даних MySQL використовується додаток `phpMyAdmin` і бібліотека `mysql`. MySQL - це система управління базами даних (СУБД), яка надає можливість зберігати, керувати та витягувати дані в структурованому форматі. Вона є однією з найпопулярніших СУБД і широко використовується веб-додатками та іншими системами для зберігання та обробки даних.

Я використовую саме цю СУБД для свого телеграм боту через

широке поширення MySQL є однією з найпопулярніших СУБД і має широку підтримку та використання в індустрії. Це означає, що існує велике співтовариство розробників і безліч ресурсів, як-от документація, підручники та форуми, які можуть допомогти в розробці та підтримці бази даних.

Продуктивність і масштабованість MySQL відомий своєю високою продуктивністю і здатністю ефективно обробляти великі обсяги даних. Він також пропонує можливості масштабування, що дають змогу керувати зростанням бази даних у міру необхідності.

Гнучкість і функціональність MySQL підтримує широкий набір функціональних можливостей, включно з підтримкою різних типів даних, індексуванням, транзакціями, збереженими процедурами і тригерами. Це дає

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		20

зможу розробникам ефективно моделювати та керувати даними, відповідаючи вимогам телеграм-бота.

Сумісність з іншими технологіями MySQL може легко інтегруватися з іншими технологіями та мовами програмування, що робить його гнучким вибором для розробки телеграм-бота. Його можна використовувати в поєднанні з різними веб-фреймворками, мовами програмування та іншими інструментами для створення повнофункціонального і надійного бота.

Надійність і безпека MySQL має механізми гарантування безпеки даних, включно з можливістю шифрування, авторизацією та управлінням доступом. Він також відомий своєю стабільністю і надійністю, що є важливим аспектом під час розроблення та обслуговування телеграм-бота.

phpMyAdmin - це веб-додаток, призначений для керування базами даних MySQL через веб-інтерфейс. Він написаний мовою програмування PHP і надає зручний та інтуїтивно зрозумілий інтерфейс для виконання різноманітних операцій із базами даних, як-от створення, видалення та зміна таблиць, вставлення, оновлення та видалення даних, виконання -запитів, керування користувачами та привілеями і багато іншого. phpMyAdmin дає змогу розробникам і адміністраторам легко адмініструвати бази даних MySQL без необхідності використання командного рядка або інших інструментів керування.

Бібліотека pymysql - це бібліотека для мови програмування Python, яка надає простий спосіб взаємодії з базами даних MySQL. Вона дає змогу виконувати різні операції з базою даних, як-от виконання запитів, створення таблиць, вставлення, оновлення та видалення даних. Pymysql надає зручний та інтуїтивно зрозумілий інтерфейс для роботи з MySQL. З її допомогою ви можете встановити з'єднання з базою даних, виконати -запити та отримати результати. Бібліотека забезпечує безпеку і захист від атак -ін'єкцій, надаючи механізми параметризованих запитів. Основні можливості бібліотеки pymysql включають

Встановлення з'єднання з базою даних MySQL.

Виконання -запитів і отримання результатів.

Підтримка параметризованих запитів для безпечного опрацювання

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		21

користувацьких даних.

Робота з транзакціями для забезпечення цілісності даних.

Створення, зміна та видалення таблиць і баз даних.

Управління користувачами та привілеями бази даних.

Обробка помилок і винятків під час роботи з базою даних.

1.4 Розробка алгоритмів відстежування оновлень сучасної літератури

Для відстежування оновлень використовується алгоритм такого типу

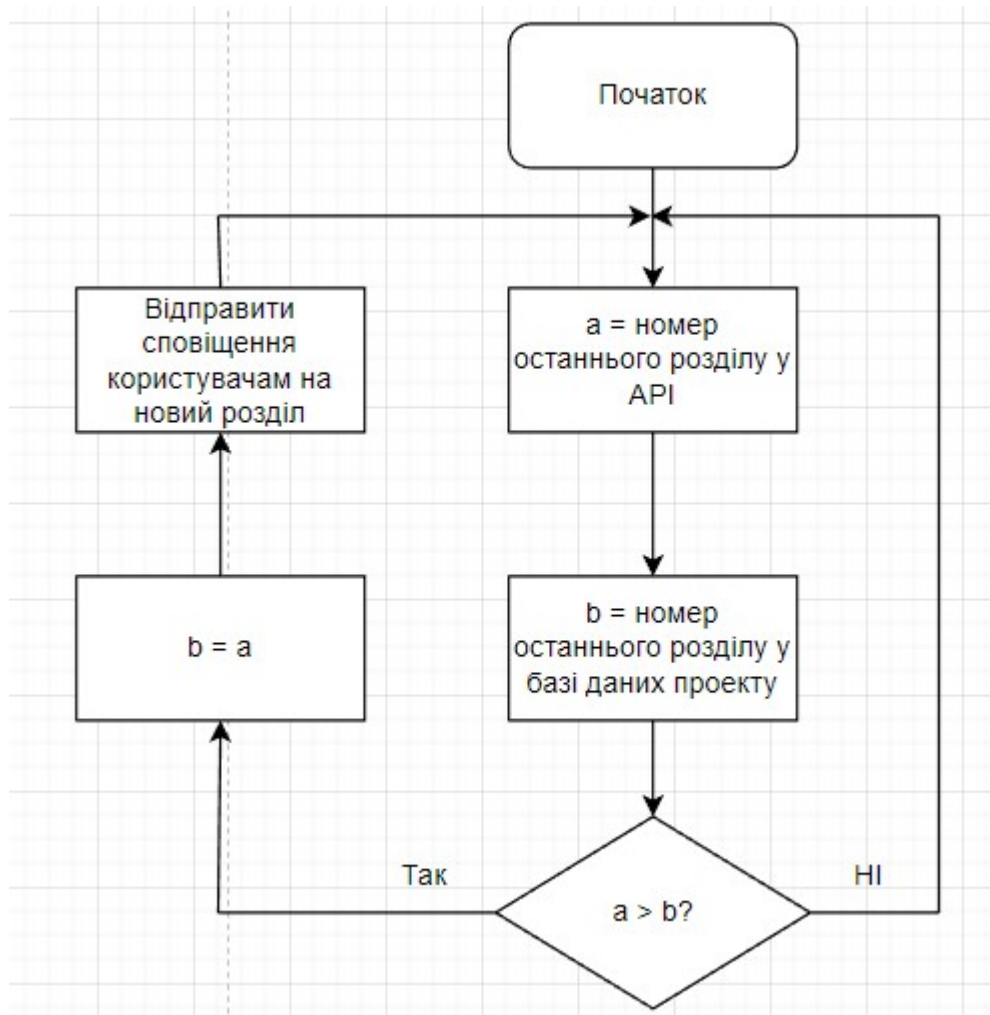


Рисунок 1.8. Алгоритм для відстежування оновлень

Для відстежування оновлень для новел цей алгоритм і був примінен, бо для актуалізації даних у цьому літературному жанрі використовується 1 джерело. Для відстежування оновлень коміксів, до алгоритму, представленого вище, треба

Зм.	Арк.	докум.	Підпис	Дата

внести деякі правки, бо цей жанр літератури відстежується не з одного джерела, тому оновлений алгоритм для коміксів виглядає так

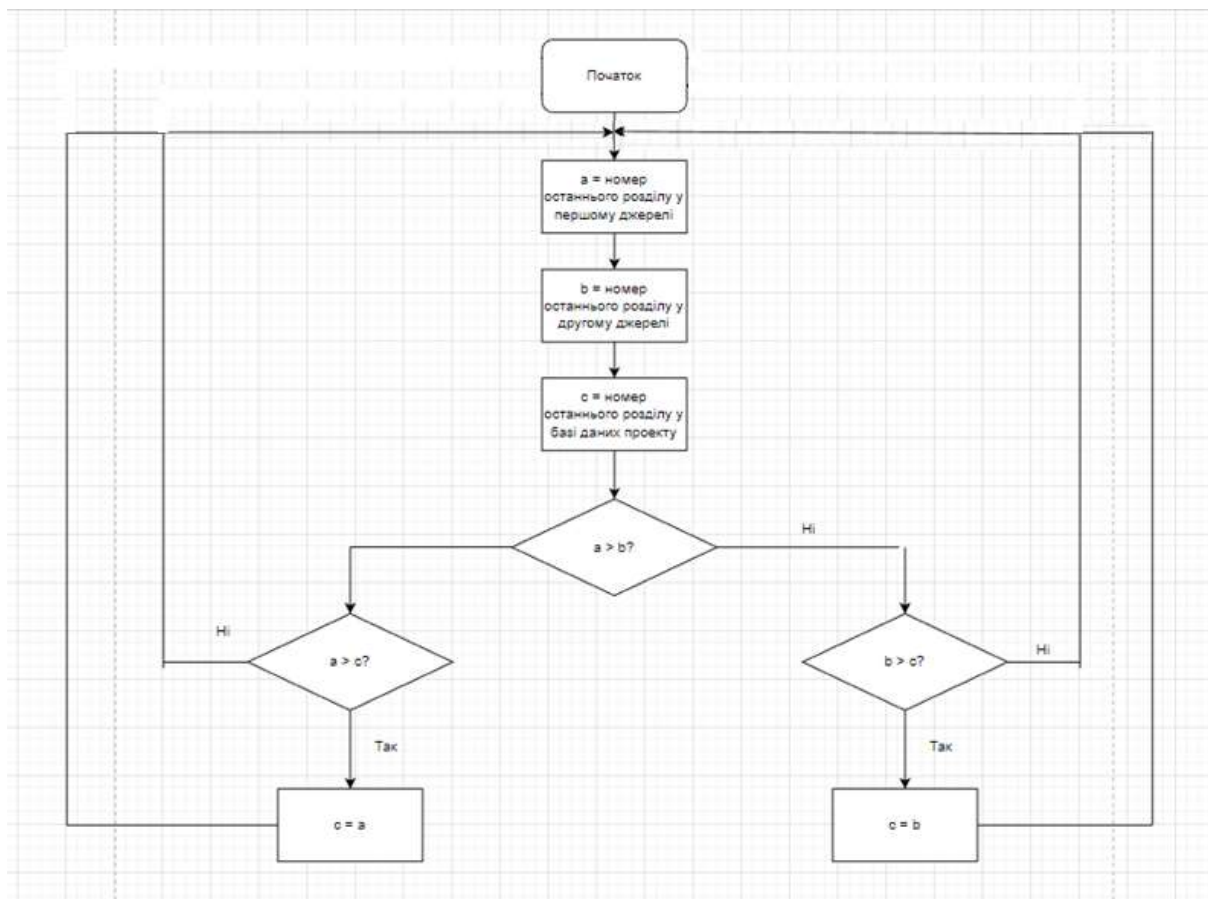


Рисунок 1.9. Алгоритм для відстежування оновлень коміксів у проекті

Число змінних a і b представлених на блок схемах бот буде отримувати за допомогою запитів до API, чи за допомогою парсингу, як-от наприклад

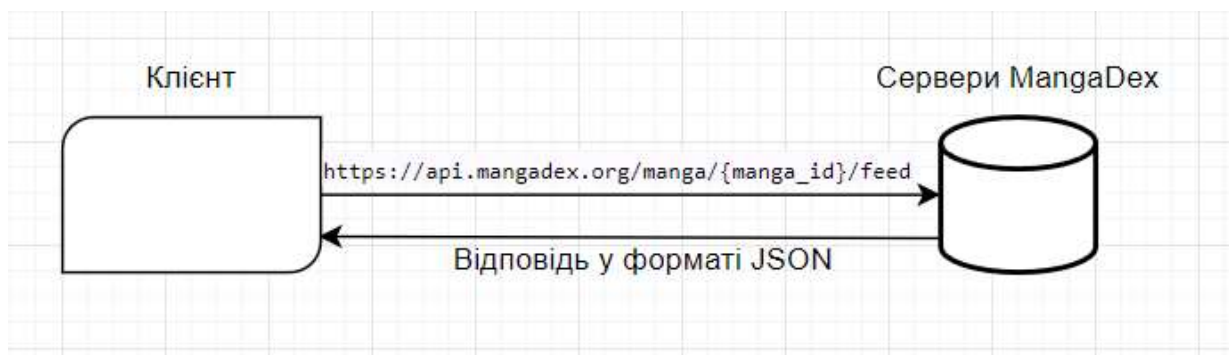


Рисунок 1.10. Схема запиту до MangaDex API

Це найбільш примітивний опис алгоритму отримання даних щодо останнього розділу, який є на серверах MangaDex. Однак на ділі він є більш складним, так як необхідно виловити конкретні дані. Почати треба з того, що GET

запит еед повертає інформацію про усі існуючі розділи на усіх мовах перекладу від усіх перекладачів. Просто так отримати інформацію про останній розділ не вийде ще й тому, що дані видаються лише по 500 розділів на відповідь, тобто якщо витвір має наприклад 200 глав і перекладений на 7 мов, то загальна кількість розділів буде складати 1400. Виникає потреба у розробці алгоритму для сортування усіх отриманих результатів. Для цього необхідно виконати такі дії

1. Проініціалізувати масиви `ast chapter ua`, `ast chapter ru` і `ast chapter en` з початковими значеннями `none`, `0.0`, `none` для зберігання інформації про останні розділи різними мовами (українською, російською та англійською відповідно).
2. Сформувати `url` для запиту до API MangaBox, використовуючи переданий ідентифікатор манги. Цей запит представлений у рисунку 1.23
3. Відправити GET-запит за отриманим `url` і отримати відповідь.
4. Якщо код стану відповіді дорівнює 200 (успішний запит), то функція обробляє дані з відповіді.
5. Витягти загальну кількість глав манги з відповіді та проініціалізувати змінні `total chapters` (зміщення) і `man a chapters` (список глав манги).
6. У циклі виконати послідовні запити з використанням різних значень зсуву (`offset`) для отримання всіх розділів манги і додати їх у список `man a chapters`.
7. Перебирати кожен розділ манги у списку `man a chapters` і порівняти її мову з цільовими мовами (`ua`, `ru`, `en`).
8. Якщо розділ відповідає мові та його номер (у вигляді числа) більший, ніж поточний номер останнього розділу відповідною мовою, то оновити інформацію про останній розділ для цієї мови.
9. Повернути список з інформацією про останні розділи манги різними мовами в такому форматі `ast chapter en`, `ast chapter ru`, `ast chapter ua`, тобто повернути двувимірний масив.
10. Якщо код стану відповіді не дорівнює 200, то функція повинна вивести повідомлення про помилку і повернути `none`.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		24

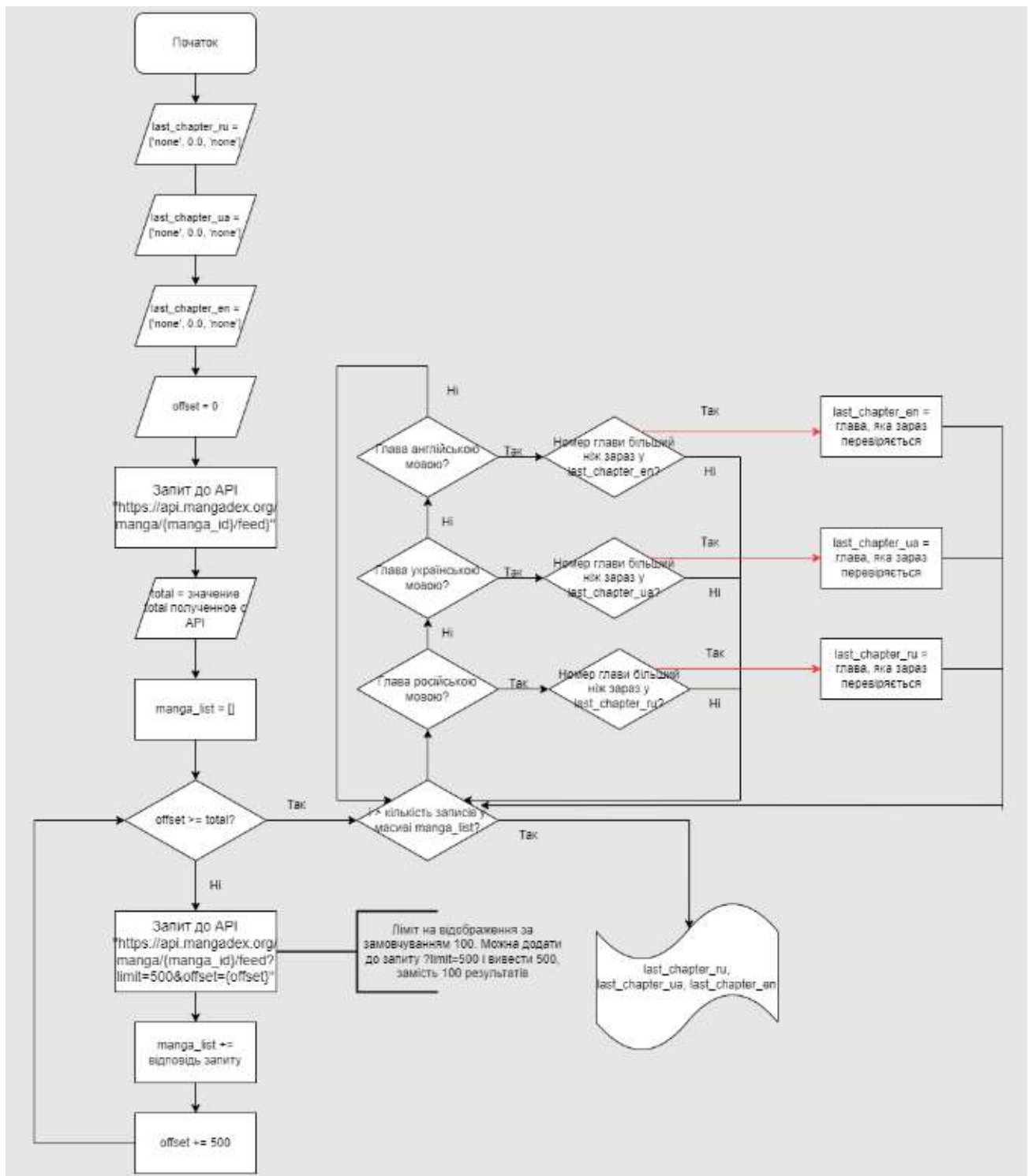


Рисунок 1.11. Алгоритм функції et an a chapters()

Тоді повний алгоритм, який буде використовуватися у боті для відстежування і сповіщення користувачів про нові глави сучасної літератури, буде виглядати наступним чином

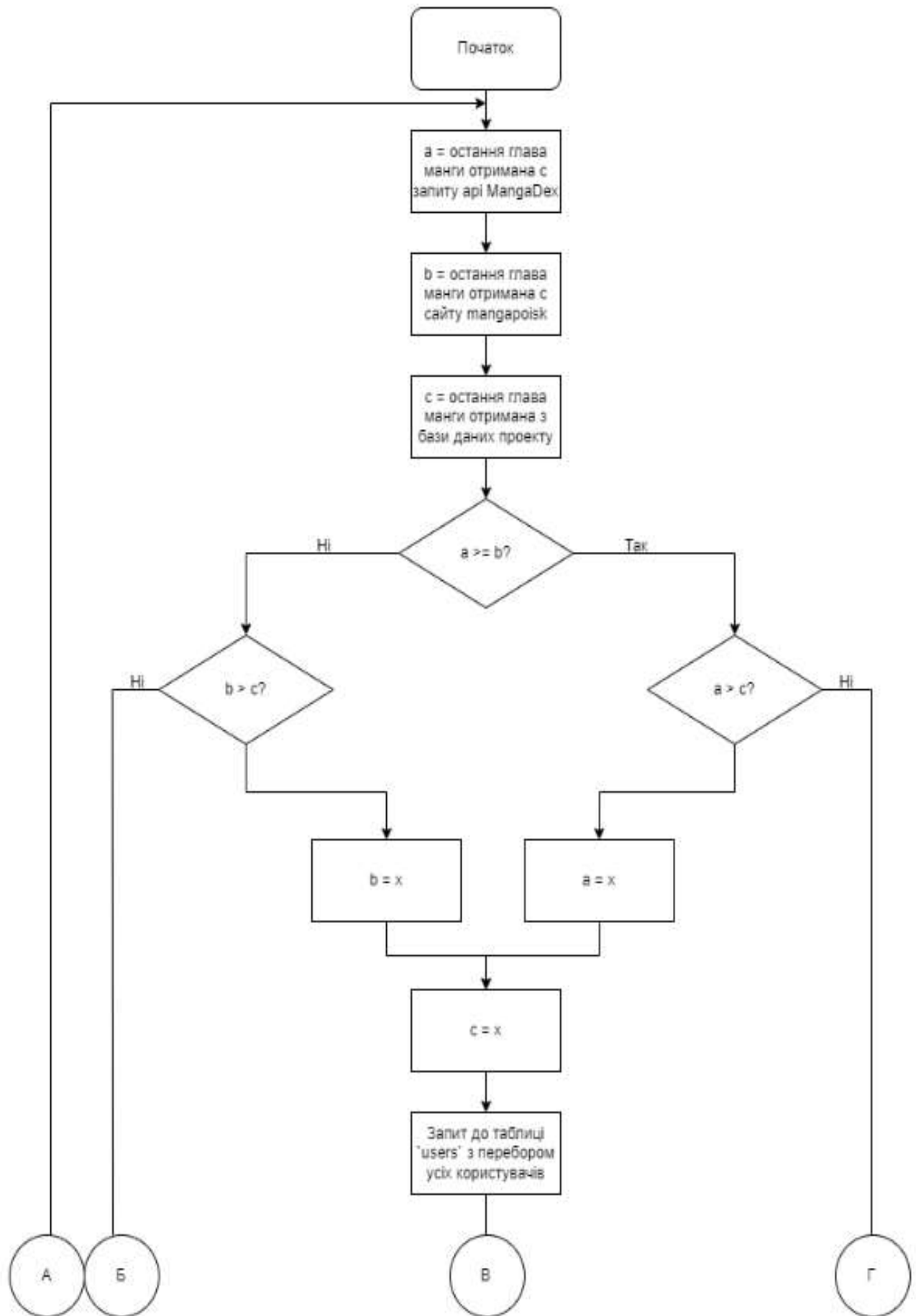


Рисунок 1.12. Алгоритм відстежування усіх жанрів літератури. Частина 1

et an a chapters(an a id) Ця функція прийматиме ідентифікатор манги (an a id) і повертатиме інформацію про останні глави манги різними мовами (англійською, російською та українською) на основі запитів до API Man a ex.

is not so p eted(an a id) Функція прийматиме ідентифікатор манги (an a id) і перевірятиме, чи є манга завершеною (so p eted) на основі запиту до API Man a ex.

an apois ast chapter(na e) Ця функція приймає назву манги (na e) і повертає інформацію про останній розділ манги на основі парсингу сайту " an apois .or ".

no e a data id(id) Функція приймає ідентифікатор новели (id) і повертає різні дані про неї, включно з номером останнього розділу, посиланням на останній розділ, назвою на основі запиту до сайту "rapo e. e".

no e a a data(na e) Функція прийматиме назву новели (na e) і повертатиме різні дані про них, включно з номером останнього розділу, посиланням на останній розділ, назвою, посиланням на твір, посиланням на обкладинку, описом та інформацією про завершеність твору на основі парсингу сайту "rapo e. e".

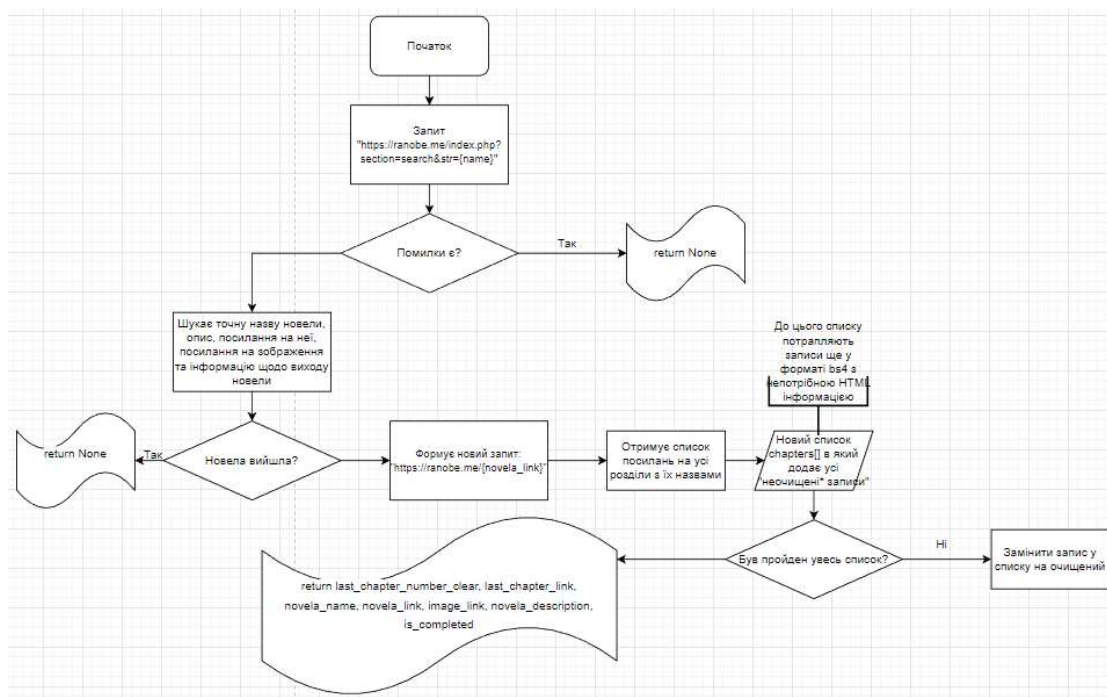


Рисунок 1.14. Алгоритм функції no e a a data()

1.4.2 Опис функцій, які необхідно спроектувати для роботи з базою даних

Функція `add to data ase an a` використовується для додавання інформації про мангу в базу даних. Вона приймає такі параметри

`an a id` ідентифікатор манги

`an a na e` назва манги

`ast chapter ru nu eг` номер останнього розділу російською мовою

`ast chapter ru in` посилання на останній розділ російською мовою

`ast chapter ua nu eг` номер останнього розділу українською мовою

`ast chapter ua in` посилання на останній розділ українською мовою

`ast chapter en nu eг` номер останнього розділу англійською мовою

`ast chapter en in` посилання на останній розділ англійською мовою

`ast chapter an a роis nu eг` номер останнього розділу на Man a роis

`ast chapter an a роis in` посилання на останній розділ на Man a роis .

Функція перевіряє, чи є вже манга з таким ідентифікатором у базі даних.

кщо манга вже є, нічого не робить. якщо манги немає, то вона додає запис про мангу в базу даних.

Функція `add to data ase user` використовується для додавання інформації про користувача в базу даних. Вона приймає параметр `chat id`, який є ідентифікатором користувача. Функція перевіряє, чи є вже користувач із таким ідентифікатором у базі даних. якщо користувач уже є, нічого не робить. якщо користувача немає, то вона додає запис про користувача в базу даних.

Функція `de ete art ro user` використовується для видалення інформації про твір (мангу або новелу) у користувача. Вона приймає параметри `user id` - ідентифікатор користувача та `s ot nu eг` - номер слота, в якому знаходиться твір.

Функція `list_of_subscribers` використовується для отримання списку підписок користувача. Вона приймає параметр `chat_id` - ідентифікатор користувача і повертає список слотів підписок, у яких містяться ідентифікатори творів.

Функція `add_manga_to_user` використовується для додавання манги в слот підписки користувача. Вона приймає параметри `manga_id` - ідентифікатор манги, `user_id` - ідентифікатор користувача та `slot_number` - номер слота, у який потрібно додати мангу.

Функція `add_to_database` використовується для додавання інформації про новелу в базу даних. Вона приймає такі параметри
`last_chapter_number` номер останнього розділу новели
`last_chapter_link` посилання на останній розділ новели
`novel_name` назва новели
`novel_link` посилання на новелу.

Функція `add_novel_to_user` використовується для додавання новели в слот підписки користувача. Вона приймає параметри `novel_id` - ідентифікатор новели та `user_id` - ідентифікатор користувача.

1.5 Опис етапів розробки проекту. Опис функціональності проекту

1.5.1 Перший етап створення телеграм-боту, робота з BotFather

Для створення ботів в месенджері телеграм, спочатку треба скористатися іншим ботом, так званим BotFather.

Для створення нового бота необхідно виконати команду `newbot` у BotFather. Під час виконання команди BotFather запитатиме у вас ім'я та логін (ім'я користувача) для нового бота. Потім він згенерує унікальний токен авторизації для вашого бота.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		30

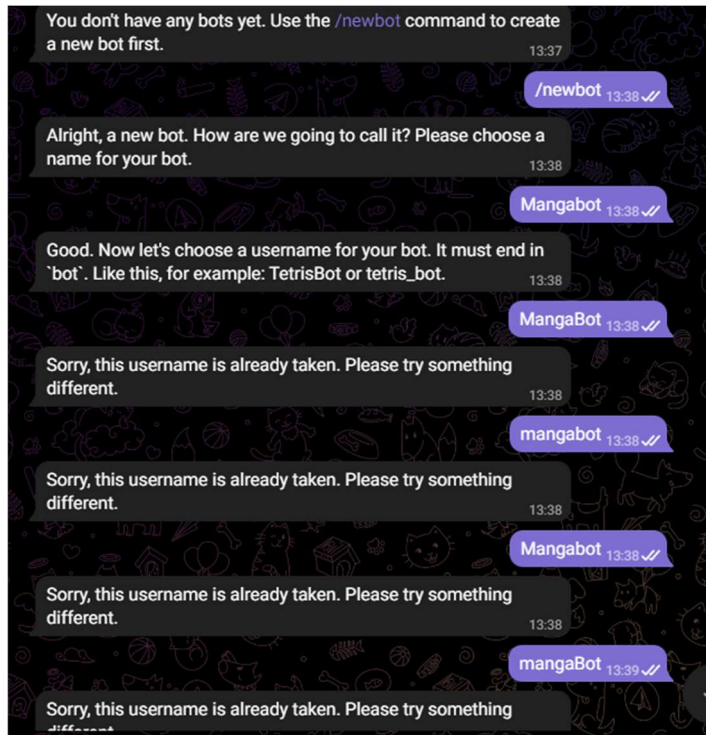


Рисунок 1.15. Демонстрація старту роботи з BotFather і перебор імен для бота дипломного проекту

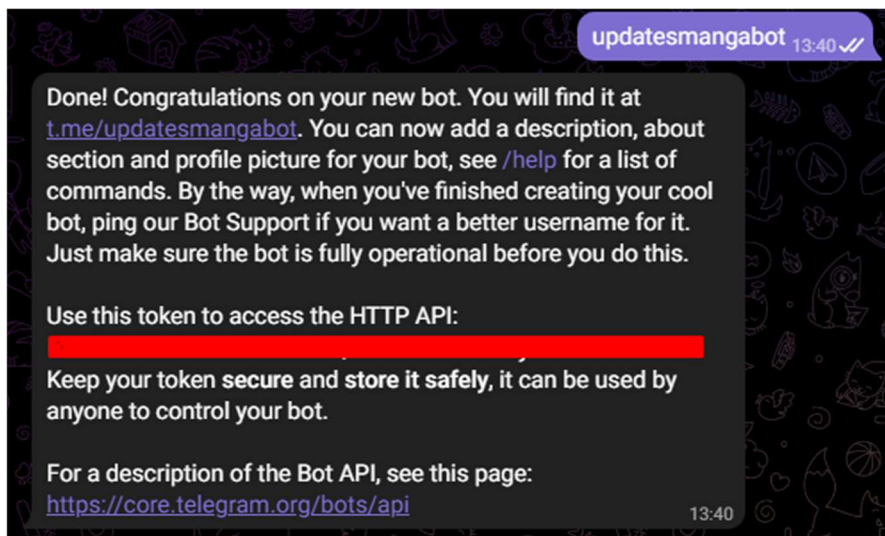


Рисунок. 1.16. Після підбору назви, бот надає токен

Токен, представлений унікальним рядком, є невідомою складовою для авторизації та взаємодії з ботом через його API. Кожен токен має свою унікальну комбінацію символів, яка надає боту право виконувати певні дії та доступ до функцій, які він реалізовує. Токен забезпечує ідентифікацію бота і дозволяє йому передавати запити та отримувати відповіді через API-інтерфейс.

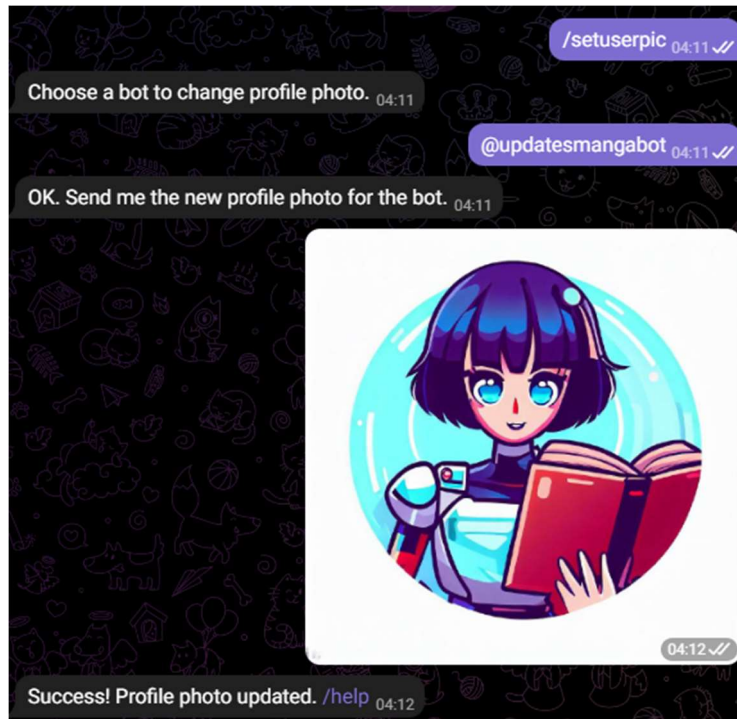


Рисунок 1.17. Демонстрація команди `setuserpic` в BotFather. Дозволяє змінити аватарку бота.

1.5.2 Структура кореневої папки проекту та встановлення бібліотек

Була створена папка, в якій і будуть знаходитися файли проекту. Проект складається з декількох файлів та однієї папки

images	09.03.2023 16:57	Папка с файлами
config.py	17.05.2023 18:42	Python File
db_functions.py	04.06.2023 3:41	Python File
literature_functions.py	04.06.2023 1:38	Python File
main.py	17.05.2023 22:10	Python File
requirements.txt	03.06.2023 22:11	Текстовый докум...
structs.py	03.06.2023 22:11	Python File
test.py	04.06.2023 4:00	Python File

Рисунок 1.18. Структура кореневої папки проекту

Папка `images` відповідає за збереження зображень обкладинок літератури для послідууючої відправки користувачу.

Файл `config.py` відповідає за збереження токена боту та словників з відповідями бота на різних мовах.

Файл `database_functions.py` відвідає за функції, котрі звертаються до бази даних.

Файл `literature_functions.py` відвідає за функції, котрі зв'язані з запитами до джерел з літературою

Файл `main.py` відвідає за основний код, з якого запускається бот

Файл `requirements.txt` відвідає за збереження списку бібліотек використаних в проекті

Файл `structs.py` відвідає за збереження структур для різних жанрів літератури

Файл `test.py` потрібен для того, щоб не додаючи код одразу до основних файлів протестувати працездатність наприклад нової функції і вже після того, як буде достовірно відомо, що функція працює і не має багів, тоді вона додається до основних файлів.

Файли передають змінні, функції та класи між собою через `import`. Ось наприклад `import`'и, які є в файлі `main.py`

`import telebot`

`from telebot import types`

`import requests`

`from literature_functions import get_an_a_id, get_an_a_names, get_a_e_in , get_an_a_chapters, an_a_pois_ast chapter, is not completed`

`from structs import Man_a, o_e`

`from database_functions import add_to_database_an_a, add_to_database_user, list_of_subscribes, add_an_a_to_user`

`from config import TOKEN`

`import time`

`import threading`

`import sys`

Для встановлення бібліотек, описаних у проекті, було прийняте рішення створити файл `requirements.txt`. Для встановлення необхідних бібліотек через файл `requirements.txt` у проекті необхідно виконати такі кроки

1. Створити файл з іменем `requirements.txt` у кореневій папці проекту.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		33

2. Відкрити файл `requirements.txt` за допомогою текстового редактора і додати в нього назви необхідних бібліотек, кожен на окремому рядку.

```
test > requirements.txt
1  telebot
2  requests
3  time
4  threading
5  pymysql
6  requests
7  bs4
```

Рисунок 1.19. Вміст файлу `requirements.txt`

3. Зберегти файл `requirements.txt`.
4. Відкрити командний рядок або термінал і перейти у директорію вашого проекту.
5. Переконайтеся, що встановлено `pip`, інструмент управління пакетами Python. Його наявність можна перевірити, виконавши команду `pip --version`
якщо команда не розпізнається, потрібно встановити `pip`.
6. Встановити необхідні бібліотеки, виконавши таку команду `pip install -r requirements.txt`
Ця команда дозволить `pip` прочитати файл `requirements.txt` і автоматично встановити всі перераховані в ньому бібліотеки та їхні залежності.
7. Дочекайтеся завершення процесу встановлення. `Pip` завантажить і встановить кожен бібліотеку, зазначену у файлі `requirements.txt`.

1.5.3 Структура бази даних

База даних проекту називається складається з трьох таких таблиць

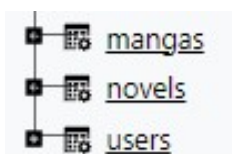


Рисунок 1.20. Таблиці у базі даних `iterature_otd`

Таблиця `an as` має таку структуру

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 <code>manga_id</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Нет	<code>Нет</code>
<input type="checkbox"/>	2 <code>manga_name</code>	varchar(200)	<code>utf8mb4_general_ci</code>		Нет	<code>Нет</code>
<input type="checkbox"/>	3 <code>last_ru_chapter_mangadex_number</code>	float		UNSIGNED	Да	<code>NULL</code>
<input type="checkbox"/>	4 <code>last_ru_chapter_mangadex_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	5 <code>last_ua_chapter_mangadex_number</code>	float		UNSIGNED	Да	<code>NULL</code>
<input type="checkbox"/>	6 <code>last_ua_chapter_mangadex_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	7 <code>last_en_chapter_mangadex_number</code>	float		UNSIGNED	Да	<code>NULL</code>
<input type="checkbox"/>	8 <code>last_en_chapter_mangadex_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	9 <code>last_chapter_mangapoisk_number</code>	float		UNSIGNED	Да	<code>NULL</code>
<input type="checkbox"/>	10 <code>last_chapter_mangapoisk_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	11 <code>last_chapter_number</code>	float		UNSIGNED	Да	<code>NULL</code>
<input type="checkbox"/>	12 <code>last_chapter_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>

Рисунок 1.21. Структура таблиці `an as`

Таблиця `no es` має таку структуру

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 <code>novel_id</code>	varchar(12)	<code>utf8mb4_general_ci</code>		Нет	<code>Нет</code>
<input type="checkbox"/>	2 <code>novel_name</code>	varchar(200)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	3 <code>last_chapter_number</code>	float			Да	<code>NULL</code>
<input type="checkbox"/>	4 <code>last_chapter_link</code>	varchar(220)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>

Рисунок 1.22. Структура таблиці `no es`

Таблиця `users` має таку структуру


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 <code>ID</code> 	int			Нет	<code>Нет</code>
<input type="checkbox"/>	2 <code>telegram_id</code>	bigint			Нет	<code>Нет</code>
<input type="checkbox"/>	3 <code>manga_slot1</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	4 <code>manga_slot2</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	5 <code>manga_slot3</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	6 <code>manga_slot4</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	7 <code>manga_slot5</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>
<input type="checkbox"/>	8 <code>novel_slot</code>	varchar(40)	<code>utf8mb4_general_ci</code>		Да	<code>NULL</code>

Рисунок 1.23. Структура таблиці `users`

Загалом використовувались такі типи даних

Int ціле число

Bi int велике ціле число

F oat число з плаваючою точкою

archar() строка. У скобках вказується обмеження на кількість символів

1.5.4 Створення класів для відстежуваних витворів

Для короткострокового зберігання даних про мангу та новели, перед їх додаванням до бази даних були розроблені їх класи, а також функції цих класів

```
class Manga
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
        this.id = id; this.title = title; this.author = author; this.genre = genre; this.cover = cover; this.description = description; this.rating = rating; this.status = status; this.language = language; this.country = country; this.publisher = publisher; this.releaseDate = releaseDate; this.isbn = isbn; this.series = series; this.volume = volume; this.chapter = chapter; this.episode = episode; this.type = type; this.format = format;
```

```
class Manga
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```
    public Manga(int id, String title, String author, String genre, String cover, String description, String rating, String status, String language, String country, String publisher, String releaseDate, String isbn, String series, String volume, String chapter, String episode, String type, String format)
```

```

no e na e str
no e in str
de set a ues(se , ast chapter nu er, ast chapter in , no e na e,
no e in )
se .ast chapter nu er ast chapter nu er
se .ast chapter in ast chapter in
se .no e na e no e na e
se .no e in no e in

```

1.5.5 Приклад функцій бота

Починається робота з ботом з команди `start`. Після неї бот відправляє таке повідомлення

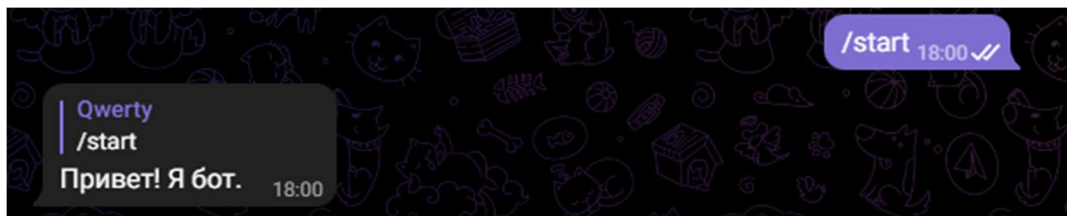


Рисунок 1.24. Робота команди `start` в боті

Відкривається наступне меню

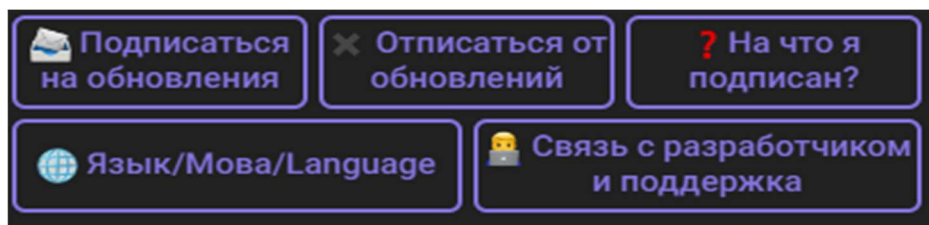


Рисунок 1.25. Кнопкове меню бота

За це відповідає цей шматок коду

```

ot. essa e hand er(co ands start )
de send e co e( essa e)
an ru
user id str( essa e.chat.id)
add to data ase user(user id)
enu ain enu( an )

```

от.реп у то(essa е, essa es an reetin , реп у ар up enu)
кий звертається до функцій
add to data ase user(), ain enu(). Ось що ці функції з себе представляють

```
de add to data ase user(chat id)
    try
        connection py ys .connect(host "ЗАСЕКРЕЧЕНО", port 3306, user
            "root", pass ord "ЗАСЕКРЕЧЕНО", data ase " iterature otd ", cursorc ass
py ys .cursors. ictCursor)
        ith connection.cursor() as cursor
            se ect uery " E ECT F OM users HE E te e ra id
chat id "
            cursor.execute(se ect uery)
            i cursor. etcha () ()
            return one)
        e se
            insert uery "I E T I TO users (te e ra id)
A E ( chat id )"
            cursor.execute(insert uery)
            connection.co it()
    ina y
        connection.c ose()
та
de ain enu( an )
    enu types. реп у ey oardMar up(resi e ey oard True)
    enu tn1 types. ey oardButton( 📧 essa es an su scri e )
    enu tn2 types. ey oardButton( ✕ essa es an unsu scri e )
    enu tn3 types. ey oardButton( ? essa es an su scri e ist )
    enu tn4 types. ey oardButton( 🌐 essa es an an ua e )
```

```

enu tn5 types. eu oardButton( 🗨️ essa es an contact )
enu.add( enu tn1, enu tn2, enu tn3, enu tn4, enu tn5)
return enu

```

Коли користувач вводить команду `start`, то бот бере його `telegram id` та дає його на обробку функції `add to database user()`. Вона, в свою чергу, спочатку перевіряє, чи є користувач у таблиці `users`. Якщо він там є, то функція не робить нічого, якщо користувача там ще нема, то функція додає `id` користувача до бази даних. Далі функція `sendMessage()` відправляє користувачу повідомлення та виводить меню, яке описано в функції `main menu()`. У рамках дослідження спробуємо підписатися на той самий комікс, котрий був розглянутий в аналітичному розділі при пошуку аналогів.

Після натискання кнопки `Підписатися на оновлення` та вибору жанру `література манга` (на рисунках написи російською мовою, однак у боті передбачена можливість змінити мову) вводиться назва твору

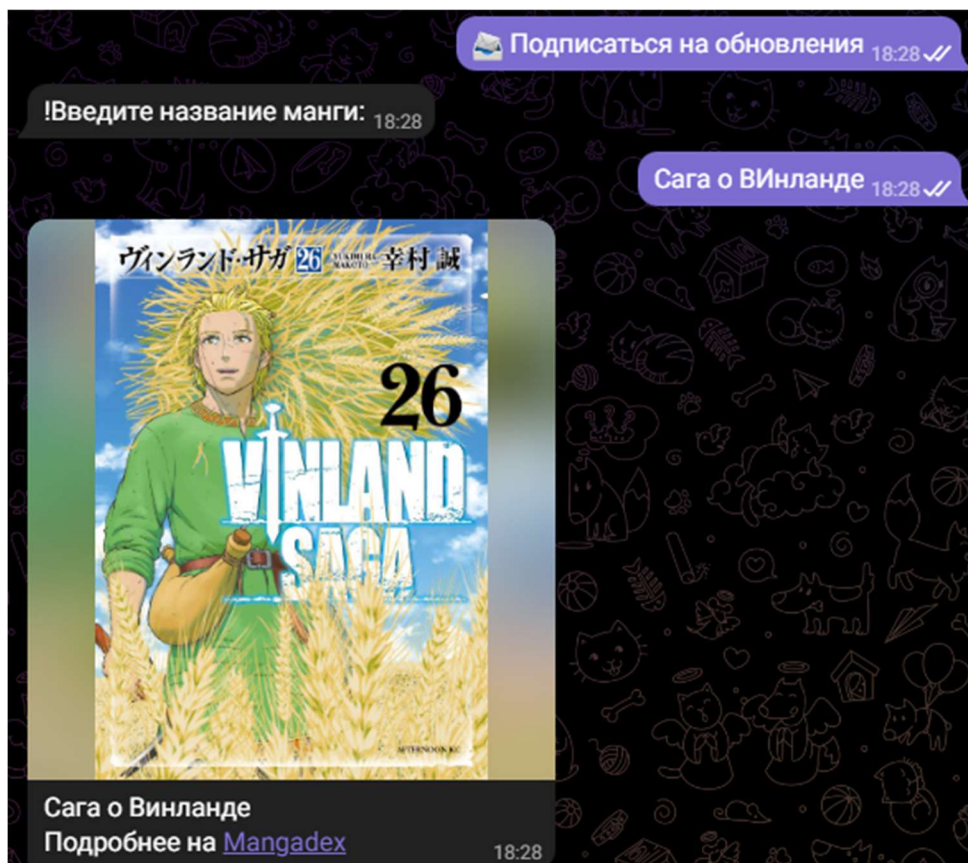


Рисунок 1.26. Пошук творів у боті

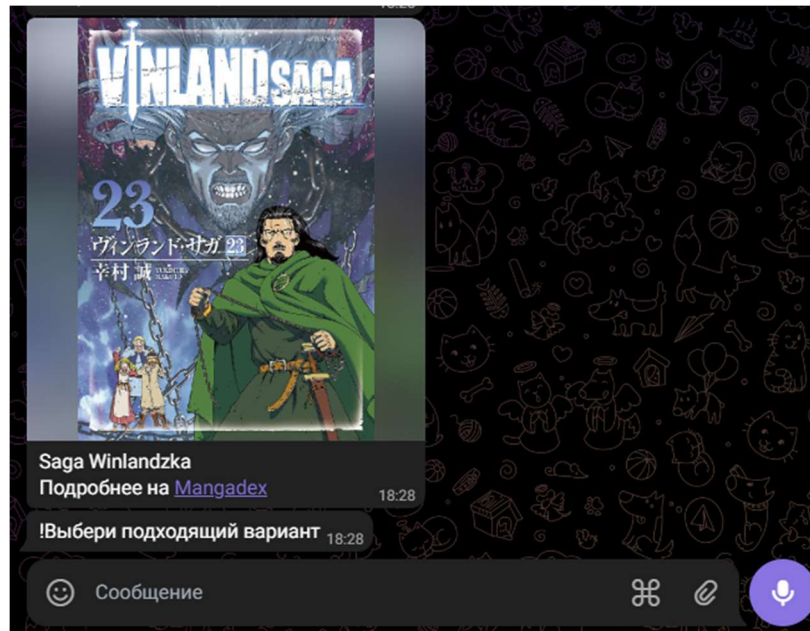


Рисунок 1.27. Вибір варіанту у боті

Бот пропонує одразу декілька варіантів і видає наступне меню

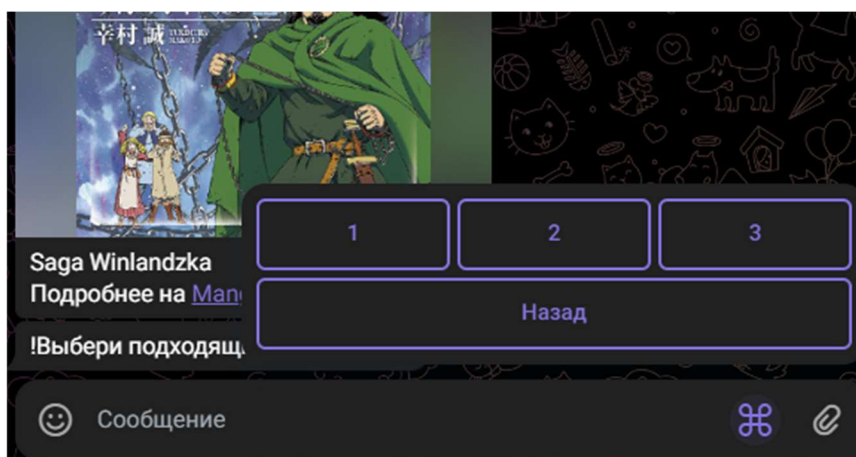


Рисунок 1.28. Меню вибору варіанту у боті

Комікс, який нас цікавить був першим у списку, тому натискаємо кнопку 1

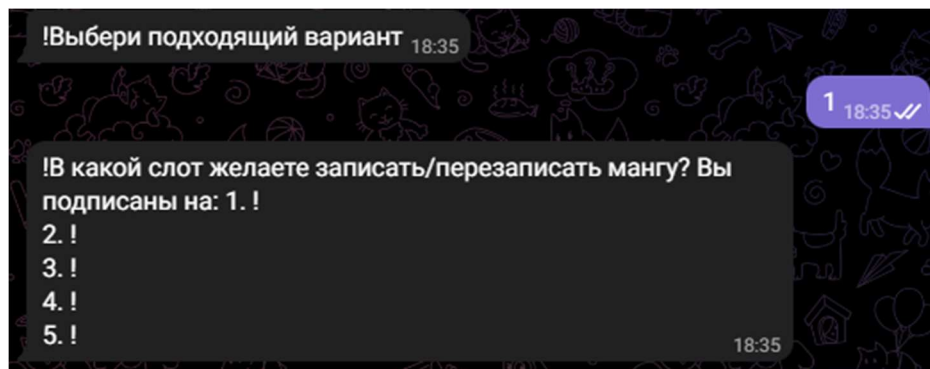


Рисунок 1.29. Вибір слоту у який запишеться витвір

Користувач повинен обрати слот для запису. Всього користувач може відстежувати 5 різних коміксів. Для новел передбачені інші слоти. Користувачу виводиться інше меню

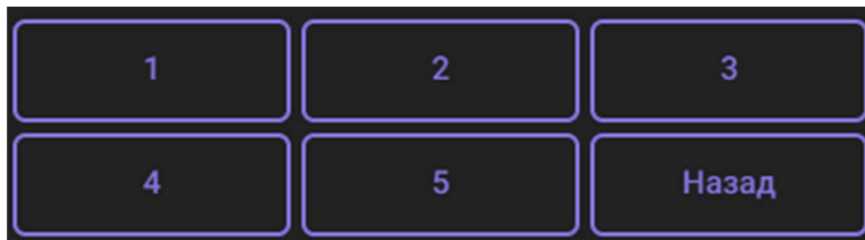


Рисунок 1.30. Меню вибору слоту.

Був вибраний 1 слот

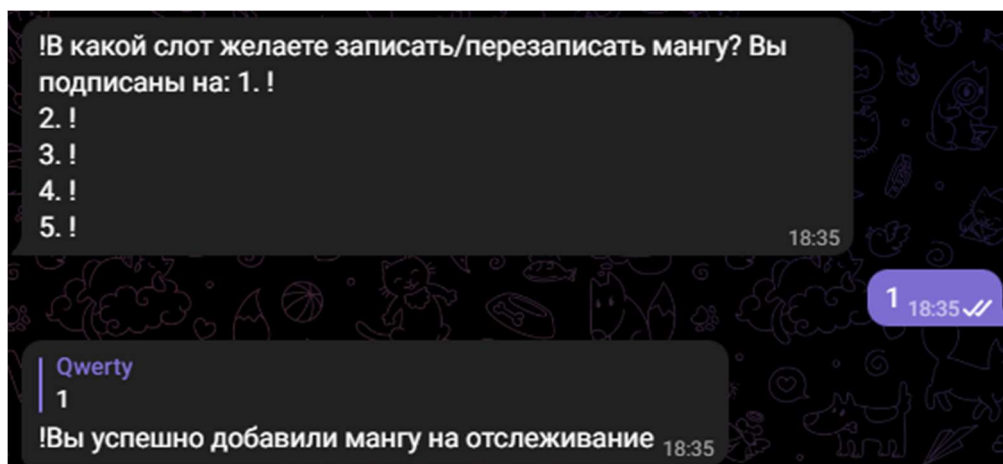


Рисунок 1.31. У рамках дослідження був вибраний 1 слот

Перевірка бази даних з користувачами

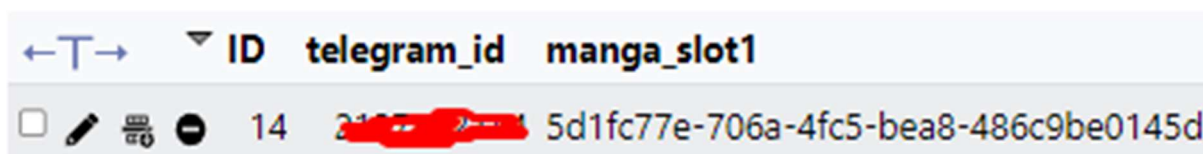



Рисунок 1.32. Запис у таблиці users

Користувач додався до бази даних і додав собі у перший слот серед манг id потрібного коміксу. Для того, щоб переконатися, що в слоті не просто набір символів, а щось осмислене, копіюємо цей id та підставимо в <https://an.adex.org/tite> ПІДСТАВЛЮМО СЮДИ



Рисунок 1.33. Демонстрація того, що ід твору зберігся вірний

Ця функція бота є обробником повідомлень, який реагує на повідомлення, що починаються з символу  .

Коли користувач надсилає таке повідомлення, викликається функція `su scri e(essa e)`. Ця функція витягує ідентифікатор чату (`chat id`) з повідомлення і надсилає повідомлення користувачеві з проханням ввести назву манги. Потім, за допомогою `ot.re ister next step hand er(essa e, sa e tit e)`, вона реєструє наступний обробник, який буде викликаний після введення користувачем назви манги.

Функція `sa e tit e(essa e)` отримує введену користувачем назву манги (`an a tit e`). Потім вона викликає функцію `et an a id(an a tit e)`, яка повертає список найкращих результатів (ідентифікаторів манги) для цієї назви манги.

кщо список `est resu ts` не порожній, то функція проходить по кожному результату і виконує такі дії

Отримує ідентифікатор манги (`an a id`) зі списку `est resu ts`.

Використовуючи функцію `et an a na es(an a id)`, отримує назви манги різними мовами.

Використовуючи `et i a e in (an a id)`, отримує посилання на зображення манги.

За допомогою бібліотеки `re uests` завантажує зображення манги.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		42

Надсилає зображення користувачеві з назвою манги і посиланням на Mangadex.

Закриває файл зображення.

Після надсилання зображень та інформації про мангу, функція надсилає користувачеві меню вибору варіанта і реєструє наступний обробник choice enu(essa e).

Функція choice enu(essa e) отримує обраний варіант від користувача. Якщо обрано опцію "назад", то надсилається головне меню. В іншому разі, функція перевіряє, чи можна відстежувати цю мангу (якщо вона завершена, то не можна).

Якщо мангу можна відстежувати, функція відправляє користувачеві меню вибору слота і реєструє наступний обробник add an a(essa e).

Функція add an a(essa e) отримує обраний слот від користувача. Якщо обрано опцію "назад", то надсилається головне меню. В іншому разі, функція зберігає інформацію про мангу, додає її в базу даних і додає мангу до списку відстежуваних для користувача. Потім надсилається повідомлення про успішне додавання манги і надсилається головне меню.

Бот відправив новий розділ

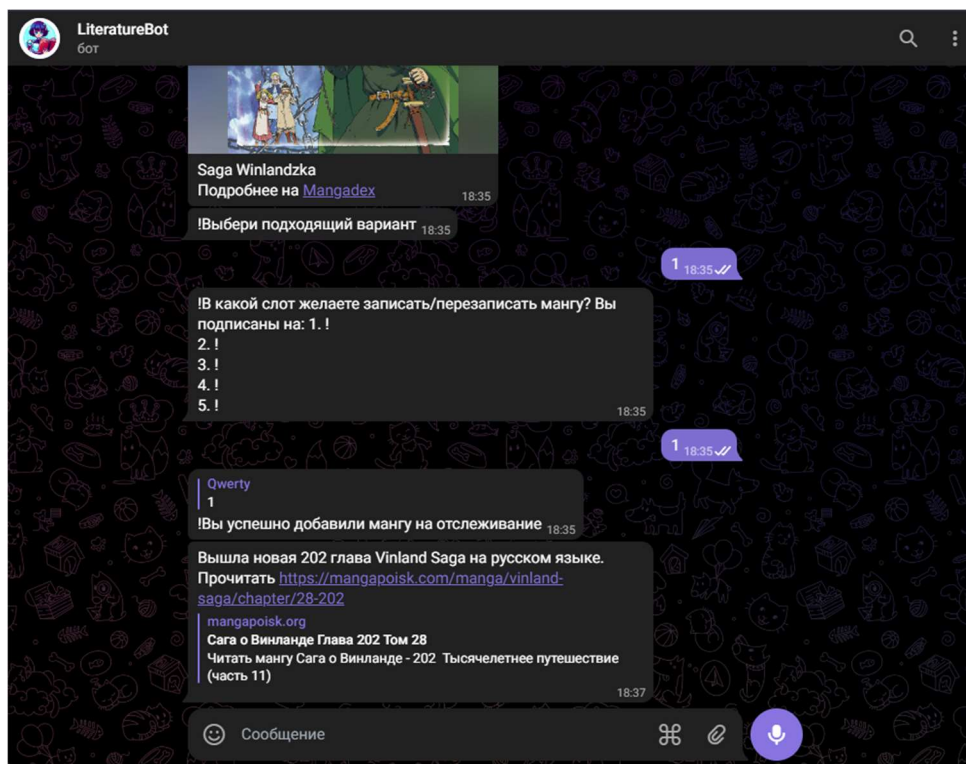


Рисунок 1.34. Бот успішно відправив новий розділ

					РП 06. 02 00 . 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		43

Після натискання на посилання для прочитання було доведено, що бот функціонує згідно з технічним завданням



Рисунок 1.35. Демонстрація працездатності посилання відправленого ботом

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		44

2 Е ОНОМІЧНА ЧАСТИНА

2.1 Резюме

В даному дипломному проекті розроблений Telegram-бот, призначений для відстеження нових глав сучасної літератури. Мета створення бота - надати користувачам зручний інструмент для отримання інформації про нові випуски та останні оновлення своїх улюблених творів.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими з точки зору користувача з позиції використання ресурсів виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуною спроможністю каналів передачі даних.

Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналог

Таблиця 2.1 Каталог аналогів

Найменування ПП	Обсяг функції ПП усл. машинних командах.
1. ПП автоматизованих розрахунків	1300 8600
2. ПП введення інформації	1060 5750
3. ПП оптимізації розрахунків	1300 4200

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить о в умовних машинних командах, трудомісткості визначати на основі табл.2.2

Таблиця.2.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, K_k 0,7 0,8) T^a 229 x 0,8 183,2 (люд годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{TII} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{PTI} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти

i питома вага i -го етапу розробки (див. табл. 2.2.)

K_H поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.)

K_T поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Таблиця 2.2. Значення питомих коефіцієнтів трудомісткості стадії загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (1)	0,15	0,12	0,12
ТП (2)	0,16	0,15	0,11
РП (3)	0,55	0,58	0,61

Таблиця 2.3. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово нові ПП	1,75 1,2
Б	ПП розвиток визначеного параметричного ряду	1,0 0,8
В	ПП маючий аналог	0,7

Таблиця 2.4. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами,	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер розраховуємо трудомісткість по кожному етапу окремо

Трудомісткість технічного завдання

$$T_{ТЗ} = T^a \cdot K_H \cdot 183,2 \cdot 0,12 \cdot 0,9 = 19,78 \text{ (люд годин)} \quad (2.4)$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T^a \cdot K_H \cdot 183,2 \cdot 0,15 \cdot 0,9 = 24,73 \text{ (люд годин)} \quad (2.5)$$

Трудомісткість розробки робочого проекту

$$T_{рп} T^a \quad 3 K_H K_T \quad 183,2 \quad 0,58 \quad 0,9 \quad 0,8 \quad 76,5 \text{ (люд годин)} \quad (2.6)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап технічне завдання t_3 1 (стр), розробка ТП $t_{тп}$ 19 (стр), розробка робочого проекту $t_{рп}$ 17 (стр), пояснювальна записка відповідно $t_{пз}$ 37 (стр)
Розрахунок зведений у таблицю 2.5

Таблиця 2.5. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1	2	3	4
1.Розробка ТЗ	$T_{рТЗ}$ 19,78	$T_{кк}$ 0,7 t_3 0,7 1 0,7	$T_{нк}$ 0,15 t_3 0,15 1 0,15
2.Розробка ТП	$T_{рТп}$ 24,73	$T_{кк}$ 0,7 $t_{тп}$ 0,7 19 13,3	$T_{нк}$ 0,15 $t_{тп}$ 0,15 19 2,85
3.Розробка РП	$T_{ррп}$ 76,5	$T_{кк}$ 0,7 $t_{рп}$ 0,7 17 11,9	$T_{нк}$ 0,15 $t_{рп}$ 0,15 17 2,55
4.Розробка ПЗ	$T_{пз}$ 1,5 $t_{пз}$ 1.5 37 55,5	$T_{кк}$ 0,7 $t_{пз}$ 0,7 37 25,9	$T_{нк}$ 0,15 $t_{пз}$ 0,15 37 5,55
Усього, в т.ч.	T 239,41		
- на розробку	T_p 176,51		
- контроль керівника		$T_{кк}$ 51,8	
- нормоконтроль			$T_{нк}$ 11,1

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.6. Відповідно до статті 8 Закону про Державний бюджет України на 2023 встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року - 6700 гривень мінімальну погодинну тарифну ставку 40.46 грн.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців.

					РП 06. 02 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		48

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн	Розрахунок, грн.
1.Розробка ПП	191,5	40.46	7 141.59
2.Контроль керівника		60.10	3 113.18
3.Нормоконтроль		60.10	667.11
Усього	-	-	Зо 10 921.88

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.7

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	73	3	219
Транспортні витрати (10)				$V_{тр\ з} = 0,1 \times V_{м1}$ 21.90
Усього				$V_{м} V_{мі} V_{тр\ з}$ 240.90

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості ПП за формою, приведеною в таблиці 2.8.

Таблиця 2.8. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	240.90	$V_{м}$
2. Основна заробітна плата	10 921.88	$З_о$
3.Додаткова заробітна плата	1 638.28	$З_д = 0,15 \times З_о$
4.Відрахування до єдиного фонду соціального внеску	2 763.23	$V_{е.с.в.} = 0,22 \times (З_о + З_д)$
5. Накладні витрати	3 276.56	$V_{нак.} = 0,3 \times З_о$
6. Повна собівартість	18 840.85	$V_{м} З_о З_д V_{е.с.в.} V_{нак.}$

					РП 06. 02 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		49

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі

$$П (C_{\text{пов}} P) 100 \quad (18\,840.85 \quad 12,5) 100 \quad 2\,355.10 \quad (2.7)$$

Де P плановий рівень рентабельності (10-15 %).

Оптова ціна (кошторисна вартість) визначається по формулі

$$C_o = C_{\text{пов}} \cdot П = 18\,840.85 \cdot 12,5 = 2\,355.10 \quad (2.8)$$

Податок на додану вартість визначаємо по наступній формулі

$$\text{ПДВ} = 0.2 \cdot C_o = 4\,710.20 \quad (2.9)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме

$$C_p = C_o + \text{ПДВ} = 2\,355.10 + 4\,710.20 = 7\,065.30 \quad (2.10)$$

					<i>РП 06. 02 002. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		50

3 ОХОРОНА ПРАЦІ

Основним завданням охорони праці є запобігання травмам, професійним захворюванням та іншим шкідливим впливам на здоров'я працівників. Законодавство України, зокрема Конституція України, Закон України "Про охорону праці", основи трудового законодавства та інші законодавчі акти містять вимоги щодо забезпечення безпеки працівників та встановлюють обов'язки для роботодавців з питань охорони праці.

Згідно зі Законом України "Про охорону праці", роботодавець зобов'язаний забезпечити безпечні умови праці для своїх працівників та запровадити необхідні заходи щодо охорони праці. Вимоги до охорони праці повинні бути враховані при розробці проекту будь-якого виробництва, а також при експлуатації обладнання та засобів виробництва.

Основні завдання з охорони праці полягають у забезпеченні безпеки і здоров'я працівників на робочому місці. До них належить

1. Виявлення можливих небезпек і ризиків на робочому місці і встановлення заходів щодо їх усунення або зменшення.
2. Забезпечення безпечних технологічних процесів праці.
3. Забезпечення безпеки виробничого середовища організація вентиляції, освітлення, звукоізоляції та інших факторів, що впливають на здоров'я працівників.
4. Забезпечення працівників необхідними засобами індивідуального захисту.
5. Організація навчання та інструктажу працівників з питань охорони праці.

Розробка заходів з охорони праці є необхідною складовою дипломного проекту, оскільки допомагає забезпечити безпеку працівників та виконати вимоги законодавства.

Для програмістів основним положенням щодо їх безпеки є ДСанПіН 3.3.2.-007 98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин щодо безпеки та захисту здоров'я працівників під час роботи з екранними

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		51

пристроями.

Основні вимоги включають такі пункти

1. Розміщення робочого місця повинно забезпечувати відсутність ближнього розташування екранного пристрою від джерел світла, що можуть викликати блиск.
2. Візуальний дисплейний термінал повинен бути зорієнтований відносно вікон, щоб уникнути падіння прямого сонячного світла на екран, що може викликати блиск.
3. скравість екрана має бути налаштована на рівень, що забезпечує комфортні умови роботи, з урахуванням особливостей конкретного виду діяльності.
4. Інтервали роботи з екранним пристроєм повинні бути регулярними і містити перерви для відпочинку очей, що дозволяє зменшити вплив шкідливих факторів.
5. Розміщення робочих місць повинно забезпечувати необхідне просторове розташування обладнання та відповідну організацію робочого місця.
6. Робоче місце повинно бути оснащено стільницею, що дозволяє забезпечити комфортні умови для роботи з екранними пристроями.

3.1 Аналіз небезпечних та шкідливих чинників на робочому місці

Під час експлуатації устаткування можуть мати місце різноманітні фактори, що можуть бути шкідливими та небезпечними для працівників.

Зокрема, можуть бути присутні такі фізичні фактори, як висока температура, надмірний шум, вібрація, іонізуюче та неіонізуюче випромінювання. Також можуть бути присутні хімічні фактори, такі як отруйні речовини, пил, гази та пари, а також біологічні фактори, такі як мікроорганізми, що можуть бути небезпечними для здоров'я працівників.

Психофізіологічні фактори, такі як стрес та психічне перенапруження, також можуть мати негативний вплив на працівників. Для оцінки ступеня впливу

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		52

кожного фактора на працівника використовуються різноманітні методи та дослідження, такі як аналіз робочого місця, медичні огляди працівників та інші.

3.2 Гігієнічні вимоги до виробничого середовища

Мікроклімат виробничих приміщень повинен відповідати нормативам з температури, вологості, швидкості руху повітря та іншим параметрам, щоб забезпечити комфортні умови для працівників. Освітлення повинно бути достатнім та рівномірним для запобігання зоровому перенапруженню та психоемоційному напруженню.

Ум та вібрації повинні бути зменшені до безпечного рівня для запобігання пошкодженню слуху та здоров'я працівників. Ультразвук та інфразвук повинні бути на безпечному рівні, щоб запобігти шкоді для слуху та здоров'я працівників. Виробничі випромінювання повинні відповідати допустимим рівням, щоб запобігти шкоді для здоров'я працівників.

Електричні системи та обладнання повинні відповідати вимогам безпеки, щоб запобігти ураженню електричним струмом. Усі ці параметри повинні бути забезпечені для забезпечення безпеки та здоров'я працівників.

Для безпечної роботи виробничих приміщень із точки зору мікроклімату, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань та небезпеки ураження електричним струмом, необхідно дотримуватись наступних безпечних рівнів

1. Освітлення освітленість на робочій поверхні не менше 300 лк, на входних дверях та проходах не менше 100 лк.
2. Ум рівень шуму на робочому місці не більше 60 дБ.
3. Вібрація вібрація на робочому місці не більше 1,5 м с .
4. Ультразвук рівень ультразвуку не більше 140 дБ.
5. Інфразвук рівень інфразвуку не більше 90 дБ.
6. Виробничі випромінювання доза випромінювання не більше 0,1 рентгена за зміну.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		53

7. Небезпека ураження електричним струмом напруга на дотик не більше 42 В.

3.3 Вимоги до організації робочого місця працівника

При організації робочого місця, яке передбачає роботу з персональним комп'ютером (ПК) та периферійними пристроями (ПП) (клавіатура, маніпулятор миша, дискова система, модем, принтер, сканер тощо), слід передбачити

Достатній простір для працівника площу та простір робочого місця

Площа робочого місця для операторів ЕОМ має бути не менше 6 кв. М. а об'єм 29 куб м

ширина робочого місця для операторів ЕОМ має бути не менше 1,5 м
Висота стіни на робочому місці має бути не менше 3,2 м.

Розміщення робочого місця по відношенню до світлових прорізів

Вікна мають бути розташовані на рівні, вищому за очі оператора ЕОМ

Необхідно забезпечити захист від прямих променів сонця

Відстані між робочими столами з ПК

Відстань між передніми краями робочих столів із ПК має бути не менше 1,2 м

Відстань між бічними поверхнями моніторів (екранів) має бути не менше 1,5 м

Екран монітора має розташовуватися на відстані 600-700 мм від очей оператора, клавіатура має бути розташована на поверхні столу на відстані 100-300 мм від краю столу.

Всі параметри при організації робочого місця відповідають зазначеним вимогам.

3.4 Пожежна безпека

На робочому столі можуть перебувати легкозаймисті матеріали, такі як папір, ручки, канцелярські клеї та інше. У кімнаті також є шафа і диван. У шафі знаходиться легкозаймистий одяг. На дивані можуть бути матеріали, здатні

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		54

швидко спалахнути в разі прямого потрапляння на них джерела тепла.

Для визначення категорії приміщення за вибухопожежною небезпекою та вибору засобів пожежогасіння і запобігання, необхідно проводити комплексне технічне обстеження приміщення та його вмісту згідно з відповідними стандартами і нормативами.

Загалом, для забезпечення безпеки в приміщенні необхідно дотримуватися таких правил

1. Не зберігати легкозаймісті матеріали поруч із джерелами тепла
2. Забезпечити правильне зберігання хімічних речовин, дотримуватися вимог щодо їх маркування та використання
3. Регулярно проводити перевірки та технічні обслуговування всіх електричних пристроїв, а також обладнання для пожежогасіння та попередження.

Для більшої безпеки будівля та приміщення повинні бути оснащені первинними засобами пожежогасіння.

					<i>РП 06. 02 00 . 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		55

У рамках дипломної роботи було розроблено Telegram-бот, призначений для відстеження нових глав сучасної літератури. Метою створення бота було надати користувачам зручний інструмент для отримання інформації про нові випуски та останні оновлення своїх улюблених творів.

Основні функціональні можливості бота включають

Підписку на мангу та ранобе Користувачі можуть підписуватися на твори, які їх цікавлять, вказуючи їхні назви.

Отримання сповіщень про нові розділи У разі появи нового розділу бот автоматично повідомляє передплатників, надаючи інформацію про назву твору та доступні розділи.

Відображення інформації про твір Бот надає детальну інформацію про кожен твір, включно з назвою, автором, жанром, описом і обкладинкою. Хоча деякі з цих даних доступні лише за посиланням.

Надання посилань на ресурси Для кожного твору бот надає посилання на ресурси, де користувачі можуть прочитати нові глави або отримати додаткову інформацію.

Керування підписками Користувачі можуть керувати своїми підписками, додавати нові твори, видаляти або змінювати налаштування наявних підписок

У економічній частині дипломного проекту також було визначено ціну програмного продукту, яка склала 29 410.47 грн.

Технічні деталі реалізації бота включають

Використання Telegram Bot API Бот був розроблений з використанням офіційного Telegram Bot API, що дає змогу взаємодіяти з користувачем і обробляти повідомлення.

Інтеграція із зовнішніми ресурсами Бот взаємодіє із зовнішніми сервісами, як-от бази даних з інформацією про твори, API для отримання

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		56

інформації про нові глави та обкладинки, а також ресурси для читання творів.

Реалізація команд і обробників Бот обробляє команди від користувачів, як-от підписка на твір і керування підписками. Для кожної команди реалізовано відповідний обробник, який виконує необхідні дії.

Взаємодія з базою даних Бот зберігає інформацію про підписки користувачів та інші необхідні дані в базі даних для подальшого використання.

Також були розглянуті питання щодо охорони праці, де виявилось, що під час роботи працівники можуть бути піддані різноманітним фізичним, хімічним та біологічним факторам, які можуть мати шкідливий вплив на їх здоров'я. Оцінка ступеня впливу цих факторів та заходи захисту від них мають велике значення для забезпечення безпеки працівників на робочому місці. Також не можна випускати з виду і пожежну небезпеку. Для захисту від цих факторів використовуються різноманітні технології, устаткування та особисті засоби захисту, а також існують нормативні документи, які встановлюють допустимі та граничні значення дії цих факторів. Виконання заходів забезпечення безпеки на робочому місці є важливим елементом ефективного управління підприємством та покращення умов праці працівників. Розробка заходів з охорони праці є необхідною складовою дипломного проекту і законодавство України містить вимоги щодо забезпечення безпеки працівників.

					<i>РП 06. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	докум.	Підпис	Дата		57

ПЕРЕЛІ ВИ ОРИСТАНИХ Д ЕРЕЛ

1. Простой Python. Современн й стиль программирования. 2-е изд. Любанович Билл. "Издательский дом "Питер"", 2020 г. 592 с.
2. Праворська Н.І. Інформатика та комп'ютерна техніка Навчальнометодичний посібник для студентів вищих навчальних закладів. Хмельницький, 2002. 312с
3. Скрапинг веб-сайтов с помощью Python. Райан Митчелл. ЛитРес, 2022 г. 336 с.
4. Копайгородська Т.Г.Методичні вказівки до виконання економічного розділу. ВСП ОТФК ОНТУ, 2023
5. Документація по створенню Те е га -ботів Електронний ресурс .
<https://core.telegram.org/bots>
6. Грофф Д. полное руководство Д. Грофф, П. Вайнберг. К. ВН , 2005. 608 с.
7. Python is a pro ra in an ua e, Електронний ресурс .
<https://python.org>
8. Основ Python. Научитесь думать как программист. Аллен Б. Дауни. "Манн, Иванов и Фербер", 2021 г. 304 с.
9. Му . Полн. и исчерп вающее руководство по применению и администрированию баз данн х Му 4, а также программированию приложений. Поль Дюбуа, Н.В Воронин Вильямс , 2004 г. 1056 с.
10. Introduction to A orith s Tho as H. Cor en, Char es E. eiseron, ona d . i est, C i ord tein Mc ra -Hi , 2009 г. 1292 с.

					РП 06. 02 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	докум.	Підпис	Дата		58

ДОДАТОК А. Робота нескінченного циклу для відстежування оновлень сучасної літератури

```
while True:
```

```
    connection = pymysql.connect(host="***.*.*.*", port=****, user="root",  
password="*****", database="literaturebotdb",  
cursorclass=pymysql.cursors.DictCursor)
```

```
    with connection.cursor() as cursor:
```

```
        select_query = "SELECT * FROM `mangas`"
```

```
        cursor.execute(select_query)
```

```
        row = cursor.fetchall()
```

```
        connection.commit()
```

```
        for i in row:
```

```
            if float(get_manga_chapters(i["manga_id"])[1][1]) >=  
float(mangapoisk_last_chapter(i["manga_name"])[1]):
```

```
                current_manga = get_manga_chapters(i["manga_id"])
```

```
                if float(current_manga[1][1]) > i["last_chapter_number"]:
```

```
                    update_query = f"UPDATE `mangas` SET `last_chapter_number` =  
{current_manga[1][1]}, `last_chapter_link` =  
'https://mangadex.org/chapter/{current_manga[1][0]}',  
`last_ru_chapter_mangadex_number` = {current_manga[1][1]},  
`last_ru_chapter_mangadex_link` =  
'https://mangadex.org/chapter/{current_manga[1][0]}' WHERE `manga_id` =  
'{i['manga_id']}'"
```

```
                    cursor.execute(update_query)
```

```
                    connection.commit()
```

```
                    select_query = "SELECT * FROM `users`"
```

```
                    cursor.execute(select_query)
```

```
                    row_2 = cursor.fetchall()
```

```
                    connection.commit()
```

```

        for o in row_2:

            if i["manga_id"] == o["manga_slot1"] or i["manga_id"] ==
o["manga_slot2"] or i["manga_id"] == o["manga_slot3"] or i["manga_id"] ==
o["manga_slot4"] or i["manga_id"] == o["manga_slot5"]:

                bot.send_message(o["telegram_id"], f"Вышла новая
{current_manga[1][1]} глава {i['manga_name']} на русском языке. Называется
{current_manga[1][2]}. Прочитать
https://mangadex.org/chapter/{current_manga[1][0]}")

                if float(get_manga_chapters(i["manga_id"])[1][1]) <
float(mangapoisk_last_chapter(i["manga_name"])[1]):

                    current_manga = mangapoisk_last_chapter(i["manga_name"])

                    if float(current_manga[1]) > i["last_chapter_number"]:

                        update_query = f"UPDATE `mangas` SET `last_chapter_number` =
{current_manga[1]}, `last_chapter_link` = '{current_manga[0]}',
`last_chapter_mangapoisk_number` = {current_manga[1]},
`last_chapter_mangapoisk_link` = '{current_manga[0]}' WHERE `manga_id` =
'{i['manga_id']}'"

                        cursor.execute(update_query)

                        connection.commit()

                        select_query = "SELECT * FROM `users`"

                        cursor.execute(select_query)

                        row_2 = cursor.fetchall()

                        connection.commit()

                        for o in row_2:

                            if i["manga_id"] == o["manga_slot1"] or i["manga_id"] ==
o["manga_slot2"] or i["manga_id"] == o["manga_slot3"] or i["manga_id"] ==
o["manga_slot4"] or i["manga_id"] == o["manga_slot5"]:

                                bot.send_message(o["telegram_id"], f"Вышла новая
{current_manga[1]} глава {i['manga_name']} на русском языке. Прочитать
{current_manga[0]}")

                            with connection.cursor() as cursor:

                                select_query = "SELECT * FROM `novels`"

```

```
cursor.execute(select_query)
row = cursor.fetchall()
connection.commit()
for i in row:
    current_chapter = novela_id_data(i["novel_id"])
    if float(current_chapter[2]) > i["last_chapter_number"]:
        update_query = f"UPDATE `novels` SET `last_chapter_number` =
        {current_chapter[2]}, `last_chapter_link` = 'ranobe.me{current_chapter[1]}'
        WHERE `novel_id` = '{i['novel_id']}'"
        cursor.execute(update_query)
        connection.commit()
        select_query = "SELECT * FROM `users`"
        cursor.execute(select_query)
        row_2 = cursor.fetchall()
        connection.commit()
        for o in row_2:
            if i["novel_id"] == o["novel_slot"]:
                bot.send_message(o["telegram_id"], f"Вышла новая
                {current_chapter[0]} глава {i['novel_name']} на русском языке. Прочитать
                {current_chapter[1]}")

time.sleep(900)
```

ДОДАТОК Б. Робота функції для додавання в базу даних нових творів та прив'язки творів до користувачів

```
@bot.message_handler(func=lambda message: message.text.startswith('✉'))
def start_subscribe(message):
    menu = genre_menu(lang)
    bot.send_message(message.chat.id, "!Что вы хотите найти?",
reply_markup=menu)
    bot.register_next_step_handler(message, chocie)

def chocie(message):
    if message.text == "Манга":
        chat_id = message.chat.id
        bot.send_message(chat_id, '!Введите название манги:')
        bot.register_next_step_handler(message, save_title_manga)
    elif message.text == "Новелла":
        chat_id = message.chat.id
        bot.send_message(chat_id, '!Введите название новеллы:')
        bot.register_next_step_handler(message, subscribe_novela)

def subscribe_novela(message):
    global novel_data
    novel_data = novela_all_data(message.text)
    print(message.text)
    print(novel_data)
    print(novela_all_data(message.text))
    if novel_data == None:
        menu = main_menu(lang)
```

```
    bot.reply_to(message, "!Ничего не найдено", reply_markup=menu)
else:
    if novel_data[6] == True:
        menu = main_menu(lang)
        bot.reply_to(message, "!Новелла, которую вы хотите отслеживать
завершена", reply_markup=menu)
    else:
        menu = subscribe_novel_menu(lang)
        chat_id = message.chat.id
        bot.send_message(chat_id, f'{novel_data[4]}\nНазвание:
{novel_data[2]}\nОписание: {novel_data[5]}')
        bot.send_message(chat_id, '!Желаете подписаться на данную новеллу?',
reply_markup=menu)
        bot.register_next_step_handler(message, choice_novela)
```

```
def choice_novela(message):
```

```
    global novel_data
    answer = message.text
    if answer == messages[lang]['back']:
        menu = main_menu(lang)
        bot.reply_to(message, "!Без проблем", reply_markup=menu)
    elif answer == "Подписаться":
        menu = main_menu(lang)
        novel.set_values(novel_data[0], novel_data[1], novel_data[2], novel_data[3])
        add_to_database_novel(novel.last_chapter_number, novel.last_chapter_link,
novel.novel_name, novel.novel_link)
        add_novel_to_user(novel.novel_link, message.chat.id)
        bot.reply_to(message, "!Успешно добавлено", reply_markup=menu)
```

```

def save_title_manga(message):
    manga_title = message.text
    global best_results
    best_results = get_manga_id(manga_title)[:3]
    if best_results != []:
        for i in range(len(best_results)):
            manga_id = get_manga_id(manga_title)[i]
            if get_manga_names(manga_id)[1] != None:
                name = get_manga_names(manga_id)[1]
            elif get_manga_names(manga_id)[2] != None:
                name = get_manga_names(manga_id)[2]
            else:
                name = get_manga_names(manga_id)[0]
            p = requests.get(get_image_link(manga_id))
            out = open(f'D:\Projects\test\images\img{i}.jpg', "wb")
            out.write(p.content)
            out = open(f'D:\Projects\test\images\img{i}.jpg', "rb")
            bot.send_photo(message.chat.id, out, f'{name}\nПодробнее на <a
href="https://mangadex.org/title/{manga_id}">Mangadex</a>',
parse_mode='HTML')
            out.close()
            menu = choose_variant_menu(lang)
            bot.send_message(message.chat.id, "!Выбери подходящий вариант",
reply_markup=menu)
            bot.register_next_step_handler(message, choose_menu_manga)
        else:
            menu = main_menu(lang)
            bot.send_message(message.chat.id, "!Ничего не найдено",
reply_markup=menu)

```

```

def chose_menu_manga(message):
    choice_variant = message.text
    if choice_variant == messages[lang]['back']:
        menu = main_menu(lang)
        bot.reply_to(message, "!Без проблем", reply_markup=menu)
    elif is_not_completed(best_results[int(choice_variant) - 1]):
        global choice_manga
        choice_manga = best_results[int(choice_variant) - 1]
        menu = choose_slot_menu(lang)
        slots = list_of_subscribes(message.chat.id)[0]
        for i in slots:
            if slots[i] == None:
                slots[i] = '!Пусто'
            else:
                slots[i] = get_manga_names(slots[i])

        bot.send_message(message.chat.id, f"!В какой слот желаете записать/перезаписать мангу? Вы подписаны на: 1. {slots['manga_slot1'][0]}\n2. {slots['manga_slot2'][0]}\n3. {slots['manga_slot3'][0]}\n4. {slots['manga_slot4'][0]}\n5. {slots['manga_slot5'][0]}", reply_markup=menu)

        bot.register_next_step_handler(message, add_manga)
    else:
        menu = main_menu(lang)
        bot.reply_to(message, "!Данная манга завершена, вы не можете ее отслеживать", reply_markup=menu)

def add_manga(message):
    choice_variant = message.text

```

```
if choice_variant == messages[lang]['back']:
    menu = main_menu(lang)
    bot.reply_to(message, "!Без проблем", reply_markup=menu)
elif int(choice_variant) > 0 and int(choice_variant) < 6:
    languages = get_manga_chapters(choice_manga)
    if get_manga_names(choice_manga)[1] == None:
        link_for_mangapoisk =
mangapoisk_last_chapter(get_manga_names(choice_manga)[0])
    else:
        link_for_mangapoisk =
mangapoisk_last_chapter(get_manga_names(choice_manga)[1])
        manga.set_values(last_chapter_ru = languages[1], last_chapter_en =
languages[0], last_chapter_ua = languages[2], last_chapter_mangapoisk =
link_for_mangapoisk, manga_name = get_manga_names(choice_manga)[0],
image_link = get_image_link(choice_manga), manga_id = choice_manga)
        add_to_database_manga(manga.manga_id, manga.manga_name,
manga.last_chapter_ru[1], manga.last_chapter_ru[0], manga.last_chapter_ua[1],
manga.last_chapter_ua[0], manga.last_chapter_en[1], manga.last_chapter_en[0],
manga.last_chapter_mangapoisk[1], manga.last_chapter_mangapoisk[0])
        add_manga_to_user(choice_manga, message.chat.id, choice_variant)
        menu = main_menu(lang)
        bot.reply_to(message, "!Вы успешно добавили мангу на отслеживание",
reply_markup=menu)
```

ДОДАТОК В. Презентація

Слайд 1

РОЗРОБКА ТЕЛЕГРАМ-БОТУ ДЛЯ ВІДСТЕЖУВАННЯ ОНОВЛЕНЬ СУЧАСНОЇ ЛІТЕРАТУРИ

- Виконав Борисков Д.Є.
гр. 4РП-06

ВСП «Одеський Технічний Фаховий Коледж ОНТУ»



Слайд 2

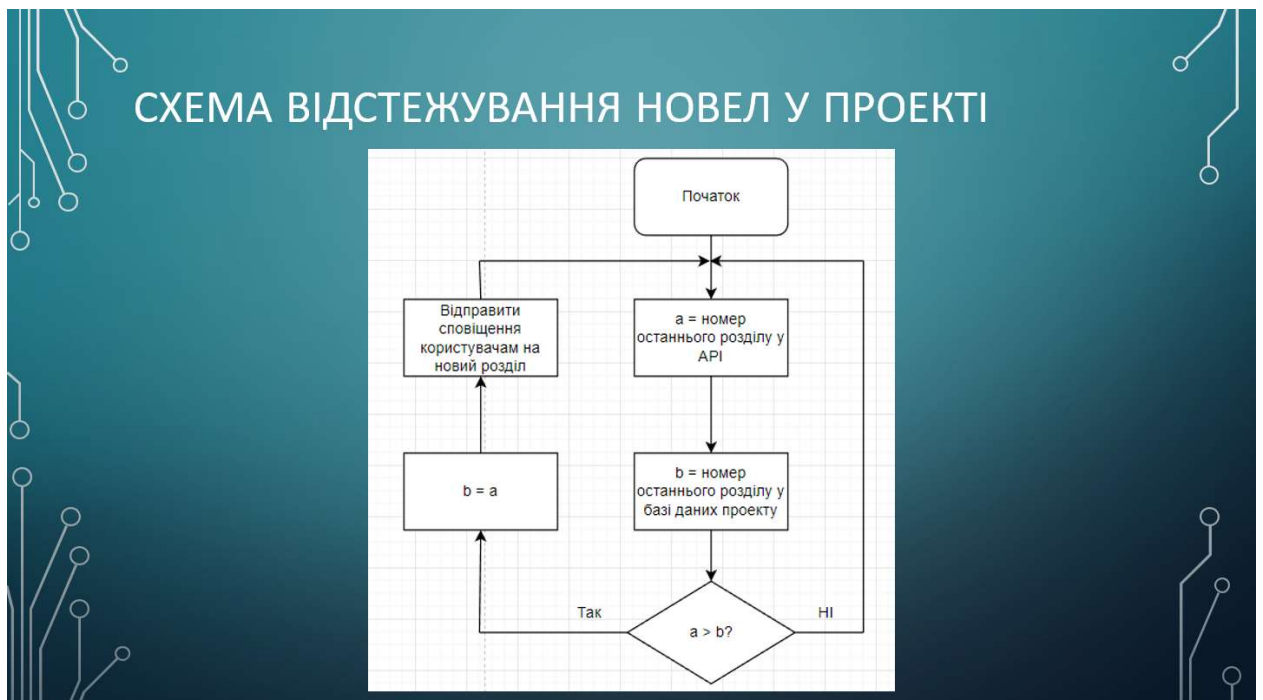
СТРУКТУРА ПРОЕКТУ

images	09.03.2023 16:57	Папка с файлами
config.py	17.05.2023 18:42	Python File
db_functions.py	04.06.2023 3:41	Python File
literature_functions.py	04.06.2023 1:38	Python File
main.py	17.05.2023 22:10	Python File
requirements.txt	03.06.2023 22:11	Текстовый докум...
structs.py	03.06.2023 22:11	Python File
test.py	04.06.2023 4:00	Python File

Слайд 5



Слайд 6



Слайд 9

СТРУКТУРА БАЗИ ДАНИХ І ТАБЛИЦЬ ЛІТЕРАТУРНИХ ЖАНРІВ

Таблиця	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано	
<input type="checkbox"/> manga	☆ 🗄 📄 📊 📌	8	InnoDB	utf8mb4_general_ci	16.0 КиБ	-	
<input type="checkbox"/> novels	☆ 🗄 📄 📊 📌	3	InnoDB	utf8mb4_general_ci	16.0 КиБ	-	
<input type="checkbox"/> users	☆ 🗄 📄 📊 📌	1	InnoDB	utf8mb4_general_ci	16.0 КиБ	-	
3 таблицы		Всего	12	InnoDB	utf8mb4_general_ci	48.0 КиБ	0 Байт

#	Имя	Тип	Сравнение	Атрибуты	Null	по умолчанию
<input type="checkbox"/> 1	manga_id	varchar(40)	utf8mb4_general_ci		Нет	Нет
<input type="checkbox"/> 2	manga_name	varchar(200)	utf8mb4_general_ci		Нет	Нет
<input type="checkbox"/> 3	last_ru_chapter_mangadex_number	float		UNSIGNED	Да	NULL
<input type="checkbox"/> 4	last_ru_chapter_mangadex_link	varchar(220)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 5	last_us_chapter_mangadex_number	float		UNSIGNED	Да	NULL
<input type="checkbox"/> 6	last_us_chapter_mangadex_link	varchar(220)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 7	last_en_chapter_mangadex_number	float		UNSIGNED	Да	NULL
<input type="checkbox"/> 8	last_en_chapter_mangadex_link	varchar(220)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 9	last_chapter_mangapolsk_number	float		UNSIGNED	Да	NULL
<input type="checkbox"/> 10	last_chapter_mangapolsk_link	varchar(220)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 11	last_chapter_number	float		UNSIGNED	Да	NULL
<input type="checkbox"/> 12	last_chapter_link	varchar(220)	utf8mb4_general_ci		Да	NULL

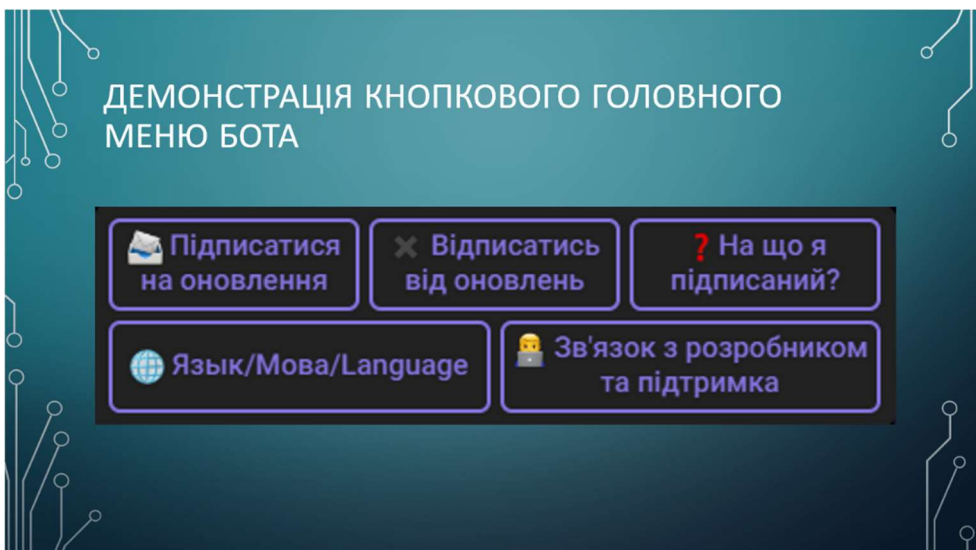
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	novel_id	varchar(12)	utf8mb4_general_ci		Нет	Нет
<input type="checkbox"/> 2	novel_name	varchar(200)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 3	last_chapter_number	float			Да	NULL
<input type="checkbox"/> 4	last_chapter_link	varchar(220)	utf8mb4_general_ci		Да	NULL

Слайд 10

СТРУКТУРА ТАБЛИЦІ `USERS` У БАЗІ ДАНИХ

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	ID 🔑	int			Нет	Нет
<input type="checkbox"/> 2	telegram_id	bigint			Нет	Нет
<input type="checkbox"/> 3	manga_slot1	varchar(40)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 4	manga_slot2	varchar(40)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 5	manga_slot3	varchar(40)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 6	manga_slot4	varchar(40)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 7	manga_slot5	varchar(40)	utf8mb4_general_ci		Да	NULL
<input type="checkbox"/> 8	novel_slot	varchar(40)	utf8mb4_general_ci		Да	NULL

Слайд 11



РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Борискова Данііла Євгеновича

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Розробка програмного забезпечення

Керівник дипломного проекту (роботи) Томаченко Олександр Євгенович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка telegram-боту для відстежування оновлень сучасної літератури

Обсяг розрахунково-пояснювальної записки 72 сторінок

Обсяг графічної (презентаційної) частини 11 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Дипломний проект відповідає поставленим завданням і має добрий рівень відповідності вимогам, встановленим для даного проекту. Розрахунково-пояснювальна записка та графічна частина проекту майже відповідає запланованим обсягам і якості.

б) характеристика виконання кожного розділу дипломного проекту (роботи) В технологічному

розділі автор детально описав технологічний процес розробки та впровадження чат-бота. Подані алгоритми, методи та інструменти, що використовувалися під час реалізації проекту. Розділ містить достатньо інформації для розуміння технічних аспектів проекту.

У економічній частині автор провів економічний аналіз проекту, оцінив його ефективність та рентабельність. Розраховані основні показники економічної ефективності, такі як затрати на розробку та впровадження. У розділі охорони праці, автор описав заходи з охорони праці.

Подана оцінка можливих ризиків та пропозиції щодо їх запобігання.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

(роботи) Пояснювальна записка містить всі необхідні розділи, досить повно описує хід роботи, виконані етапи проекту та досягнуті результати. Вона структурована, логічна та зрозуміла.

Графічна частина проекту включає блок-схеми та ілюстративні матеріали, які належним чином підтримують пояснювальну записку та поліпшують сприйняття матеріалу.

г) перелік позитивних якостей дипломного проекту (роботи) До переваг можна віднести повністю виконану розробку боту, відсутність повноцінних аналогів, детально описаний етап розробки, багато наведеного графічного матеріалу, простота в користуванні розробленого боту, ефективна реалізація, зручний інтерфейс

д) основні недоліки дипломного проекту (роботи) 1. До алгоритму є зауваження
2. Бот може втрачати продуктивність або ставати нестабільним при великому навантаженні.

Оцінка розрахункової частини Добре

Оцінка графічної частини Добре

Загальна оцінка Добре

Прізвище, ім'я, по батькові рецензента Стайкуца Сергій Володимирович

Місце роботи і посада рецензента Державний університет інтелектуальних технологій і зв'язку, к.ф.н., доцент кафедри КБ та ТЗІ, пом.декана факультету інформаційних технологій та кібербезпеки

Підпис: 

« 16 » 06 2023 р.

ПІДПИС ПОСВІАЧУЄ
НАЧАЛЬНИК ВІДДІЛУ
КАДРІВ ДУІТЗ



ВІДДІЛ
КАДРІВ

Стайкуца

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Борисков Данііл Євгенович

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: Розробка програмного забезпечення

Тема дипломного проекту: Розробка telegram-бота для відстежування оновлень сучасної літератури

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконаний повністю і в повному обсязі згідно з вимогами і рекомендаціями. Графічний матеріал, включаючи схеми, блок-схеми та інші графічні елементи, представлені чітко і зрозуміло.

Розрахунково-пояснювальна записка містить повну і детальну інформацію про проект, включаючи опис проблеми, постановку задачі, методи дослідження та отримані результати. Вона написана задовільно.

*Пояснювальна записка виконана на 72 сторінках
Графічний матеріал на 11 слайдах*

б) самостійність роботи над проектом: Студент виявив високий рівень самостійності та ініціативи в роботі над проектом. Він проявив здатність до самостійного пошуку інформації, використання актуальних джерел та методів дослідження.

в) теоретична підготовка випускника (випускниці): Студент виявив задовільні знання у сфері, пов'язаній з дипломним проектом. Він продемонстрував розуміння теоретичних концепцій, методів та моделей, що стосуються проекту. Теоретична база, що була використана в дипломному проекті, є актуальною і відповідає сучасним вимогам.

г) вміння розв'язувати виробничі та конструкторські питання Студент продемонстрував
вміння застосовувати теоретичні знання для розв'язання практичних завдань і проблем,
пов'язаних з проектом. Він виявив здатність до аналізу та вирішення виробничих та
конструкторських питань, включаючи вибір оптимальних рішень та використання
інноваційних підходів, але було виявлено недоліки щодо
якості оформлення алгоритмів

Оцінка розрахункової частини (4)Добре

Оцінка графічної частини (3)Задовільно

Загальна оцінка (4)Добре

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Томаченко Олександр Євгенович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний коледж
ОНТУ», Викладач спеціальних дисциплін КТ.Т.а.М

Підпис _____

«09» 06 2023 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015605590

Дата перевірки:
14.06.2023 21:56:04 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
14.06.2023 21:57:07 EEST

ID користувача:
100011688

Назва документа: 4РП-06 Борисков Д

Кількість сторінок: 53 Кількість слів: 9266 Кількість символів: 65604 Розмір файлу: 5.60 MB ID файлу: 1015253729

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.68% Схожість

Найбільша схожість: 0.43% з Інтернет-джерелом (<https://studopedia.info/7-64346.html>)

3.68% Джерела з Інтернету

469

Сторінка 55

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

28

Підозріле форматування

9
сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Борисков Данііл Євгенович,
здобувач освіти гр. 4РП-06, та

Томаченко Олександр Євгенович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи фахового молодшого бакалавра на тему:

«Розробка telegram-боту для відстежування оновлень сучасної літератури» (автор роботи – Борисков Д.Є., керівник роботи – Томаченко О.Є.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець  / Борисков Д.Є. /

Керівник  / Томаченко О.Є./

« 04 » 06 20 23 р.