

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
"ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Спеціальність: 121 "Інженерія програмного забезпечення"
Освітня програма: "Розробка програмного забезпечення"

ДИПЛОМНИЙ ПРОЄКТ

для отримання освітньо-кваліфікаційного рівня "Молодший спеціаліст"
на тему:
"Keyword Binder Bot"

Шифр ВСП "ХКТФК НТУ "ХПІ"-РПЗ417.121.20.20

Керівник викладач Росіца МАНЄВА

Виконавець:

Студент групи РПЗ-417 Володимир ЮШКО

Харків 2021 р.

[illegible]

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
"ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Циклова комісія: "Комп'ютерних та інформаційних дисциплін"

Освітньо-кваліфікаційний рівень: молодший спеціаліст

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітня програма: "Розробка програмного забезпечення"

"Затверджено"

Заступник директора

з навчальної роботи

Олександр ІГНАТЕНКО _____

"21" травня 2021 р.

"Погоджено"

Голова циклової комісії

_____ Павло КОЛОМІЄЦЬ

"21" травня 2021 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ

Юшку Володимир Романовичу

(прізвище, ім'я, по батькові)

Тема проекту: "Keyword Binder Bot"

керівник проекту Росіца МАНЄВА

(ім'я та прізвище)

затверджені наказом НТУ "ХПІ" від 05 травня 2020 р. № 817 СТ

Строк подання студентом проекту на затвердження 24 червня 2021 р.

Вихідні дані до проекту. Зміст розрахунково-пояснювальної записки

Вступ.....	2
1 Огляд предметної області.....	3
1.1 Опис проблеми. Актуальність	3
1.2 Огляд аналогів	4
1.3 Технічне завдання	7
1.3.1 Словник.....	7
1.3.2 Перелік команд.....	7
1.3.3 Початок роботи	8
1.3.4 Команда /start.....	8
1.3.5 Команда /help.....	8
1.3.6 Команда /list.....	8
1.3.7 Команда /bind	9
1.3.8 Команда /unbind	9
1.3.9 Ключове слово.....	10
1.3.10 Шаблон.....	10
1.3.11 Збереження даних	10

1.3.12 Програмна частина	10
1.3.13 Локалізація.....	10
1.3.14 Відмовостійкість	11
1.3.15 Розгортання.....	11
1.4 Вибір засобів розробки.....	11
1.5 Вибір моделі життєвого циклу	12
2 Алгоритмічне забезпечення	16
2.1 Діаграма варіантів використання	16
2.2 Діаграма розгортання.....	17
2.3 Діаграма компонентів	17
2.4 Опис бази даних	19
3 Планування та реалізація тестування.....	21
3.1 План тестування програмного продукту	21
3.2 Тест дизайн	22
3.3 Звіт про тестування	23
4 Опис програмного продукту	24
4.1 Системні вимоги.....	24
4.2 Посібник користувача.....	24
4.2.1 Початок роботи	24
4.2.2 Команда /help.....	28
4.2.3 Команда /list.....	29
4.2.4 Команда /bind	29
4.2.5 Команда /unbind	30
4.2.6 Змінні.....	31
5 Техніко-економічне обґрунтування проекту.....	33
5.1 Загальні положення.....	33
5.2 Визначення трудомісткості розробки	33
5.3 Розрахунок витрат на заробітну плату та єдиного соціального внеску	34
5.4 Розрахунок витрат на матеріали.....	35
5.5 Витрати на електроенергію для технологічних потреб	36
5.6 Розрахунок амортизаційних відрахувань	36
5.7 Накладні витрати.....	38
5.8 Розрахунок кошторисної вартості та ціни програмного продукту	38
5.9 Аналіз постійних і змінних витрат	39
5.10 Визначення точки беззбитковості	40
5.11 Обґрунтування використання програмного продукту	42
5.12 Висновок	43
6 Охорона праці та навколишнього середовища	45
6.1 Загальні положення охорони праці	45
6.2 Аналіз умов праці.....	46
6.2.1 Вимоги до приміщення при експлуатації ПК	46
6.2.2 Вимоги до мікроклімату.....	47
6.2.3 Вимоги до освітлення робочих місць користувачів ПК	48
6.3 Електробезпека.....	49

6.4 Пожежна безпека.....	52
6.5 Охорона навколишнього середовища	53
Висновки	55
Перелік джерел інформації.....	56
Додаток А Лістинг програмного коду.....	57
Додаток Б Економічні розрахунки	61
Ілюстративні матеріали	62

Консультанти розділів проекту:

Розділ	Ім'я та прізвище і посада консультанта	Завдання видав		Завдання прийняв	
		Дата	Підпис	Дата	Підпис
теоретико-розробницький	Росіца МАНЄВА, викладач	20.05.21		20.05.21	
організаційно-економічний	Вадим ТОЛСТОП'ЯТОВ, викладач	22.05.21		22.05.21	
охорона праці	Тетяна ТКАЧЕНКО, викладач	22.05.21		22.05.21	

Дата видачі завдання: "20" травня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів роботи	Примітка
1	Вивчення рекомендованих джерел інформації	22.05.2021 р.	
2	Аналіз сучасного стану розв'язання задач	24.05.2021 р.	
3	Постановка та формалізація задачі	25.05.2021 р.	
4	Розробка інформаційного забезпечення	27.05.2021 р.	
5	Алгоритмізація задачі	03.06.2021 р.	
6	Обґрунтування використовуваних технічних засобів	04.06.2021 р.	
7	Розробка програмного забезпечення	05.06.2021 р.	
8	Налагодження програми	06.06.2021 р.	
9	Вирішення питань економіки	07.06.2021 р.	
10	Вирішення питань охорони праці	10.06.2021 р.	
11	Виконання графічних документів	11.06.2021 р.	
12	Оформлення пояснювальної записки	12.06.2021 р.	
13	Подання ДП на попередній захист	13.06.2021 р.	
14	Подання ДП рецензенту	14.06.2021 р.	
15	Подання ДП на основний захист	17.06.2021 р.	

Студент групи РПЗ-417

Володимир ЮШКО

Керівник дипломного проекту

Росіца МАНЄВА

РЕФЕРАТ

Звіт про виконання ДП: 77 с., 26 рис., 17 табл., 2 додатки, 10 джерел.

Ключові слова: БОТ, ШАБЛОН, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ПОБУДОВА БАЗИ ДАНИХ, ТОЧКА БЕЗЗБИТКОВОСТІ, КОШТОРИСНА ВАРТІСТЬ

Представлений проект присвячено розробці програмного та алгоритмічного забезпечення для розв'язання задачі з розробки Telegram бота для створення шаблонів повідомлень.

Розроблене програмне забезпечення дозволяє створювати шаблони повідомлень та відтворювати їх.

На базі розглянутих моделей та алгоритмів розроблені база даних та програмне забезпечення, створені основні програмні документи, такі як технічне завдання та звіт до дипломного проекту.

Запропоновано техніко-економічний розрахунок, що дозволяє оцінити трудомісткість розробки, постійні та змінні витрати, пов'язані з розробкою програмного забезпечення.

За рахунок застосування і сполучення професій забезпечується зниження чисельності працюючих, ріст продуктивності праці. Обґрунтований вибір способу розробки програмного засобу та кількості розповсюджених копій програмного забезпечення.

ABSTRACT

Notes: 77 pp., 26 figs., 17 tables, 2 annex, 10 sources

Keywords: BOT, TEMPLATE, SOFTWARE DEVELOPMENT, DATABASE DEVELOPMENT, BREAK-FREE POINT, ESTIMATED COST

The presented project is devoted to the development of software and algorithms for solving the problem of developing a Telegram bot for creating message templates.

Developed software allows you to create message templates and use them.

On the basis of the considered models and algorithms the database and the software are developed, the basic program documents, such as the technical task and the report to the diploma project are created.

A feasibility study is proposed, which allows you to estimate the complexity, fixed and variable costs associated with software development.

Due to the application and combination of professions provides a reduction in the number of employees, increasing productivity. The choice of the method of software development and the number of distributed copies of the software is substantiated.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
"ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Циклова комісія: "Комп'ютерних та інформаційних дисциплін"

До захисту допущено
Заступник директора
з навчальної роботи

_____ Олександр ІГНАТЕНКО
" " червня 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту
освітньо-кваліфікаційного рівня "Молодший спеціаліст"
на тему: **"Keyword Binder Bot"**

Виконав: студент 4 курсу, групи РПЗ-417
спеціальності 121 "Інженерія програмного забезпечення"
освітньої програми: "Розробка програмного забезпечення"

Володимир ЮШКО

(підпис)

(дата)

Керівник

(підпис)

(дата)

Росіца МАНЄВА

(ім'я та прізвище)

Консультанти:

Старший консультант

(підпис)

(дата)

Павло КОЛОМІЄЦЬ

(ім'я та прізвище)

Технологічної частини

(підпис)

(дата)

Росіца МАНЄВА

(ім'я та прізвище)

Економічної частини

(підпис)

(дата)

Вадим ТОЛСТОП'ЯТОВ

(ім'я та прізвище)

Охорона праці

(підпис)

(дата)

Тетяна ТКАЧЕНКО

(ім'я та прізвище)

Рецензент

(підпис)

(дата)

Росіца МАНЄВА

(ім'я та прізвище)

Харків 2021 р.

ВСТУП

При розробці дипломного проекту планується розподіл пояснювальної записки на наступні розділи.

Перший розділ розкриватиме загальні питання, а саме огляд предметної області, постановку задачі, огляд засобів розробки програмного продукту та модель життєвого циклу проекту.

Другий розділ міститиме в собі детальний опис алгоритмічного забезпечення програмного продукту: діаграму варіантів використання, діаграму розгортання, діаграму компонентів та опис бази даних.

В третьому розділі буде розказано про планування і реалізацію тестування, а саме: створюватиметься план тестування ПП, буде проводитися звіт про тестування, які баг-репорти були виявленні при тестування програмного продукту.

У четвертому розділі проводиться та розробляється керівництво користувача.

У п'ятому розділі розповідається про техніко-економічне обґрунтування проекту.

У шостому розділі описані норми охорони праці щодо розробки ПП.

Очікуваним результатом від розробки дипломного проекту є чат бот, який зможе зберігати шаблони повідомлень користувача та відтворювати їх у будь-якому чаті.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис проблеми. Актуальність

Сучасні месенджери досить придатні для того, щоб інтегрувати багато корисних функцій у них, зазвичай це робиться за допомогою чат ботів. Чат бот – це програма, яка імітує реальний діалог з користувачем. Чат боти присутні на багатьох рекламних сайтах в якості асистентів. Також сучасні месенджери дають платформу для створення чат ботів у рамках їхнього додатку для спілкування. Ці боти дають змогу виконувати деякі складні дії лише даючи текстові, аудіо або графічні команди.

Кожен месенджер має свій інтерфейс для розробки і керування чат ботами. Для розробки програмного продукту був обраний месенджер «Telegram».

Основні переваги Telegram:

- Популярність

На теперішній час Telegram займає першу позицію у рейтингу Google Play серед інших месенджерів, таким чином розробка боту під цю платформу охопить більше аудиторії

- Зручність

Telegram має досить багато вбудованих функцій, які дозволяють досить зручно інтегрувати майже будь-який інтерфейс.

Досить часто люди ведуть бізнес через месенджери. Іноді для цього залучають спеціальних чат ботів, які допомагають робити купівлі, відповідають на часті запитання. Але для декого створення повністю свого боту є не зовсім доречним або занадто складним за багатьма причинами. Ці люди часто вимушені надсилати однакові або майже однакові повідомлення багатьом контактам. Знаходження схожого повідомлення у попередніх чатах, його копіювання, вставка цього тексту у потрібний чат, можливо дрібне редагування може займати досить багато часу. Мій бот пропонує рішення цієї проблеми.

«Keyword Binder Bot» дозволяє зберігати деякі шаблонні повідомлення та отримувати до них доступ майже одразу у будь-якому чаті. Цей бот працює індивідуально з кожним користувачем, тому ці шаблони доступні лише тому, хто їх створив у цілях безпеки.

Цим ботом може користуватися будь-хто, від старости, який зможе одразу надіслати усі контакти викладачів будь-кому хто його попросить, до адміністрації закладів, які повідомляють про необхідність продовження абонементу до зали. Цей бот простий у використанні та не потребує часу для налаштування. Невелика інфографіка для KeyBinderBot представлена на рисунку 1.1.

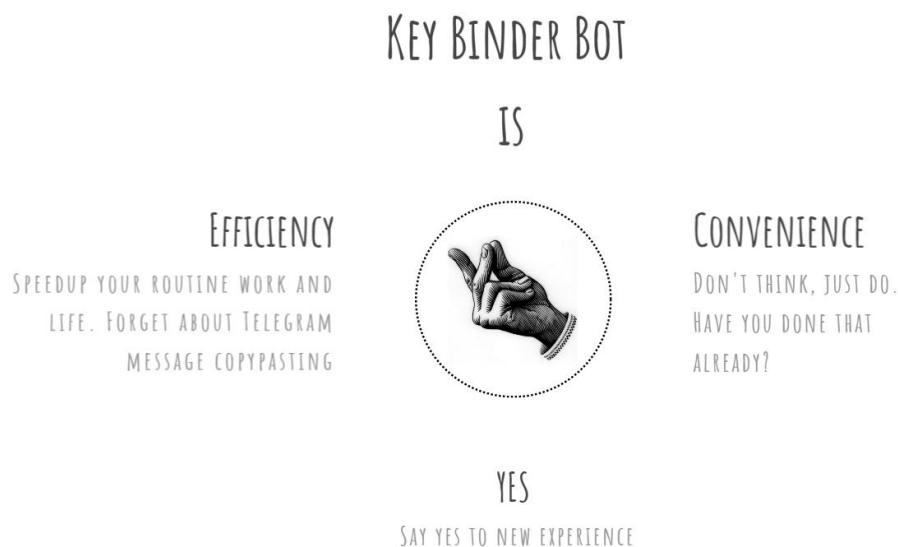


Рисунок 1.1 – Інфографіка

1.2 Огляд аналогів

SendPulse – це платформа, що дозволяє створювати ботів для багатьох месенджерів. Вона дозволяє налаштовувати автовідповіді, запускати розсилки і оптимізувати бізнес процеси за допомогою бота [1]. Ця платформа потребує деякого розуміння принципу роботи ботів та змушує створювати свого бота для

окремої задачі через спеціальний графічний конструктор (Рисунок 1.2). Також цей сервіс є умовно безкоштовним до деякої кількості користувачів.



Рисунок 1.2 – Графічний конструктор SendPulse

На відміну від SendPulse, KeyBinderBot не створює нового бота під кожного користувача, а працює одразу для всіх. Будь-хто може почати користуватися ним коли завгодно та де завгодно.

PrivateBoxBot – це бот, який надає деяке сховище для зберігання даних різних видів, деякі з них представлені на рисунку 1.3. На відміну від попереднього аналогу цей бот не змушує створювати окремого бота для кожного користувача, однак це лише сховище, яке не дозволяє проводити операції з даними під час їх використання. Також він є не досить зрозумілим для використання.

На відміну від PrivateBoxBot, KeyBinderBot має набагато простіший інтерфейс, що не потребує жодних додаткових налаштувань та дозволяє проводити деякі зміни у створених шаблонах шляхом використання змінних.

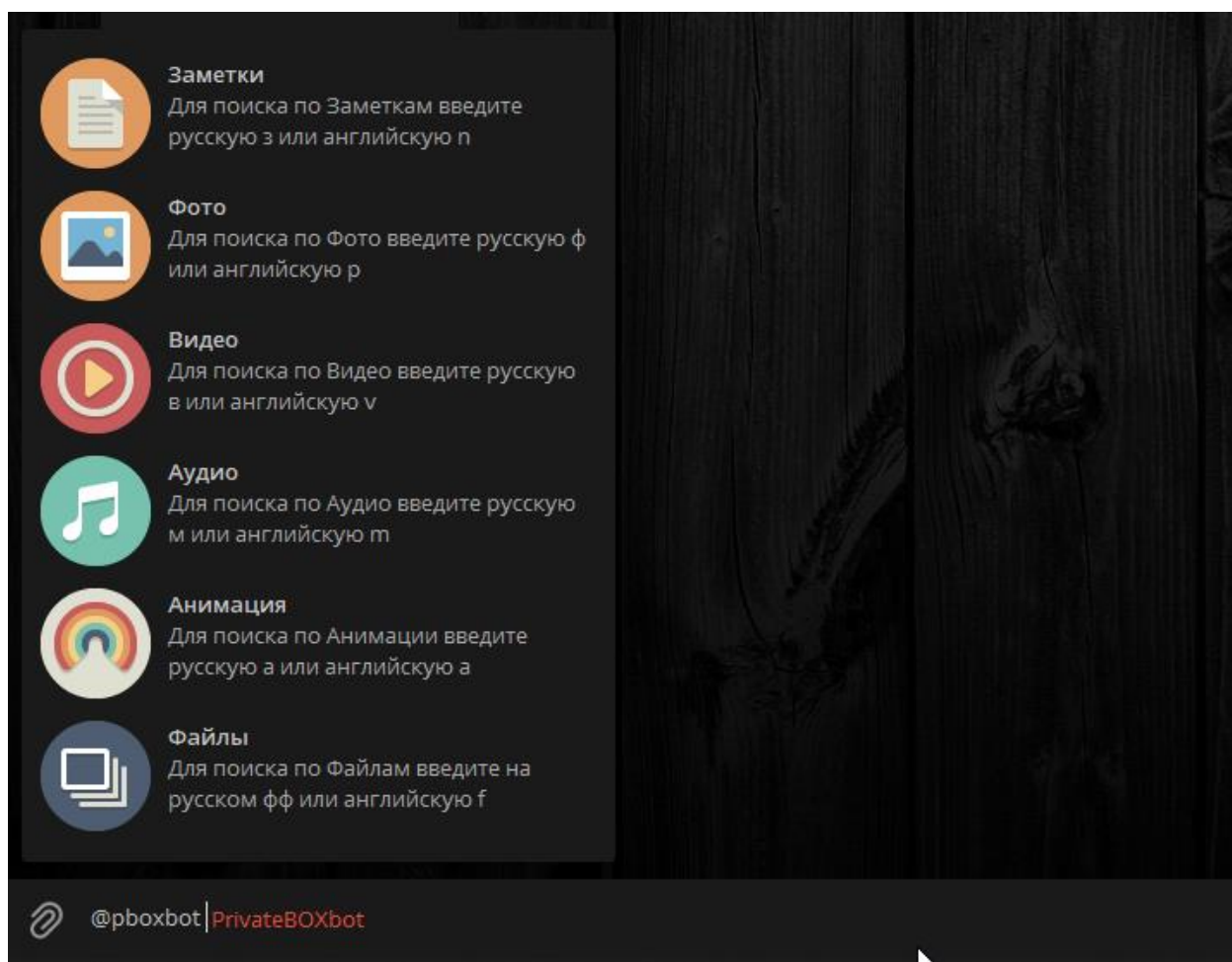


Рисунок 1.3 – Формати даних PrivateBoxBot

Для більш наочного порівняння цих аналогів з даним продуктом була створена порівняльна таблиця 1.4.

Таблиця 1.4 – Порівняння аналогів

Ознака	KeyBinderBot	SendPulse	PrivateBoxBot
Гнучкість	Дозволяє зберігати будь-яку інформацію, що підтримується Telegram, та примітивний функціонал для редагування цих даних під час використання	Майже не має обмежень у можливостях та сферах використання	Дозволяє лише зберігати та відтворювати віділені формати даних

Поріг входу	Функціонал доступний для будь-кого, для початку використання досить лише написати /start та створити шаблон	Перед початком роботи треба повністю скласти бота за допомогою спеціального графічного інтерфейсу	Перед початком використання треба пройти досить довгий процес реєстрації.
Вартість	Умовно безкоштовний, залежить від кількості збережених шаблонів	Умовно безкоштовний, залежить від потоку користувачів	Умовно Безкоштовний, залежить від максимально обсягу даних, що можуть бути збережені

1.3 Технічне завдання

Технічне завдання (ТЗ) – це затверджений документ, на основі якого виконується розробка проекту. У ньому максимально точно і детально описані вимоги до компонентів та характеристик майбутнього продукту. [2]

1.3.1 Словник

- Нікнейм – Ім'я користувача. Обов'язково є у будь-яких Telegram ботів та необов'язково є у звичайних користувачів.
- Інлайн – режим Telegram боту. Якщо увімкнений, користувач може викликати бота за допомогою набирання його нікнейму та запиту в текстовому полі повідомлення у будь-якому чаті. Запит надходить до боту під час отримання оновлень. Таким чином користувач може запрошувати деякий контент у бота в будь-де з його чатів, груп або каналів без надсилання жодних повідомлень [3]
- Інлайн кнопки – кнопки, що прикріплені до деякого повідомлення
- Шаблон – деяке збережене повідомлення користувача, яке може бути відтворено під час використання боту інлайн.
- Ключове слово – деяке слово-ключ за допомогою якого користувач зможе отримувати доступ до деякого шаблону.

1.3.2 Перелік команд

Бот має обробляти наступні команди:

- /start
- /help
- /bind
- /unbind
- /list

На кожному з цих команд бот має дати видиму для користувача відповідь, яка має бути локалізована.

1.3.3 Початок роботи

Під час первинного додавання бота усі користувачі повинні відправити боту команду /start (це зумовлено системою безпеки Telegram, що не дозволяє ботам писати користувачам першими для запобігання спаму від ботів). У цей момент бот надсилає стандартне вітання та не робить жодних записів у базі.

1.3.4 Команда /start

При надсиланні користувачем команди /start бот повинен відправити стандартне вітання, яке має містити посилання на команду /help, при натисканні якого буде автоматично відправлена команда /help.

1.3.5 Команда /help

При надсиланні користувачем команди /help бот повинен відправити більш детальну інформацію про існуючі команди та їх призначення.

Самі команди в повідомленні мають бути виділені.

Команди повинні дозволяти користувачу натискати на них задля їх виконання, окрім команд /bind та /unbind бо ці команди мають виконуватися з обов'язковими параметрами.

1.3.6 Команда /list

При надсиланні користувачем команди /list бот повинен надіслати повідомлення з переліком усіх шаблонів користувача. Також команда має «таємний» параметр для користувачів, що наявні у колекції адміністрації, user_id – індекс користувача з яким буде проводитися операція.

Перелік має бути зроблений за допомогою інлайн кнопок кнопки мають бути розташовані у 2 стовбця (якщо кількість непарна – остання кнопка займає 2 стовбця). На кожній кнопці має бути написане одне ключове слово. При натисканні кожна кнопка дає команду ботові надіслати шаблон відповідний до ключового слова на кнопці для перед перегляду.

У випадку, якщо користувач немає жодних збережених шаблонів, він має отримати відповідне повідомлення.

1.3.7 Команда /bind

Команда /bind має обов'язковий параметр kwd – ключове слово, яке користувач буде використовувати для доступу до поточного шаблону. Допускається перелік ключових слів розділених комою.

Після ключового слова користувач може вводити будь-який текст з будь-яким форматуванням, яке має бути збережене при відтворенні.

При виконанні команди шаблон має бути доданий до колекції шаблонів та має бути зв'язаний з користувачем з колекції користувачів у базі даних. У випадку, якщо в колекції ще не існує запису користувача, вона повинна бути створена.

Після виконання команди має бути надісланий новий шаблон для перед перегляду або повідомлення про помилку.

1.3.8 Команда /unbind

Команда /unbind має обов'язковий параметр kwd – ключове слово, шаблон відповідний якому має бути видалений. Допускається перелік ключових слів розділених комою. Також є «таємний» параметр для користувачів, що наявні у колекції адміністрації, user_id – індекс користувача з яким буде проводитися операція.

При виконанні видаляє зв'язок користувача та шаблону, видаляє шаблон з колекції шаблонів, якщо жодне з інших ключових слів користувача не має зв'язку з цим шаблоном.

Після виконання команди має бути надіслано повідомлення про успішність операції або про помилку.

1.3.9 Ключове слово

Ключове слово має містити лише букви та цифри, є нечутливим до регістру. З одним ключовим словом може бути зв'язаний лише один шаблон.

1.3.10 Шаблон

Шаблон може містити форматований текст, файл будь-якого формату або деякий список файлів, таким чином підтримуючи усі типи повідомлень до яких можна додати текст.

Типи повідомлень, що мають підтримуватися:

- Текстові
- Зображення
- Відео
- Файл
- Аудіо
- Медіа група (сукупність файлів одного типу, зображення та відео можуть бути разом)

Текст після ключового слова є необов'язковим для всіх типів повідомлень окрім текстових.

1.3.11 Збереження даних

База даних має містити наступні колекції:

- users – список користувачів
- keywords – список шаблонів
- admins – список індексів адміністраторів

Усі файли зберігаються на платформі Telegram, у базу записуються лише індивідуальні ідентифікатори файлів.

1.3.12 Програмна частина

Увесь код має бути написаний на мові програмування Python з використанням бібліотеки для розробки Telegram ботів aiogram.

В якості бази даних має бути використана MongoDB.

1.3.13 Локалізація

Увесь текст повідомлень має бути локалізованим.

Інформація про локалізацію має братися з повідомлення користувача.

Мови, що мають бути реалізовані:

- Англійська
- Російська

У майбутньому можливе додавання інших мов.

У випадку, якщо мови користувача немає у існуючому переліку, за замовчуванням має бути обрана англійська мова.

1.3.14 Відмовостійкість

У випадку помилки на стороні користувача, бот має вказати користувачу на помилку

У випадку внутрішньої помилки, повна інформація про помилку має бути надіслана адміністрації

1.3.15 Розгортання

База даних повинна бути доступна онлайн та зберігатися на серверах, що надаються самою MongoDB.

Скрипт програми повинен бути розгорнутий на сервері з підтримкою мови Python не нижче 3.8.

Кожен розробник повинен мати власно створеного бота для роботи. Забороняється використання основного боту.

Параметри підключення для бази даних та токен боту повинні зберігатися у змінних середовища, що відрізняються на сервері та локально.

1.4 Вибір засобів розробки

Для розробки Telegram боту використовувалося середовище розробки PyCharm 2020 від компанії JetBrains.

PyCharm – це спеціальне інтегроване середовище розробки Python (IDE), що забезпечує широкий спектр важливих інструментів для розробників Python, тісно інтегрованих для створення зручного середовища для продуктивного розвитку Python, Інтернету та науки про дані. [4]

Для моделювання бази даних використовувався додаток MongoDB Compass

MongoDB Compass – графічний інтерфейс для роботи з MongoDB, що дозволяє робити всі базові операції CRUD, переглядати та оптимізувати запити та візуально досліджувати наявні дані. [5]

В якості мови програмування була використана мова Python.

Python – це інтерпретована, об’єктно-орієнтована мова програмування високого рівня з динамічною семантикою.

Для зручної роботи з Telegram API була використана спеціальна бібліотека для Python Aiogram.

Aiogram – це простий і повністю асинхронний фреймворк для API Telegram Bot, написаний на Python 3.7 з asyncio та aiohttp, що допомагає робити ботів простішими та швидшими.

1.5 Вибір моделі життєвого циклу

При розробці програмного продукту використовувалася інкрементна модель життєвого циклу, під час якої програмне забезпечення розробляється з лінійною послідовністю стадій однак в декілька інкрементів (версій). Таким чином покращення продукту проходить заплановано весь час, доки життєвий цикл розробки не завершиться.

Потреби до системи визначаються в самому початку роботи, після чого процес розробки проводиться у вигляді послідовності версій, кожна з яких являє собою закінчений працездатний продукт.

Ця модель є досить зручною тому, що спираючись на технічне завдання, є можливість створити деякий працездатний продукт на початкових етапах розробки.

Для розбиття завдань за пріоритетами використовувалася техніка Story Mapping.

Story Mapping – це метод організації користувацьких історій для створення більш цілісного уявлення про те, як вони вписуються у загальний користувацький досвід. [6]

Результат розбиття завдань за пріоритетами вказаний на рисунку 1.5.

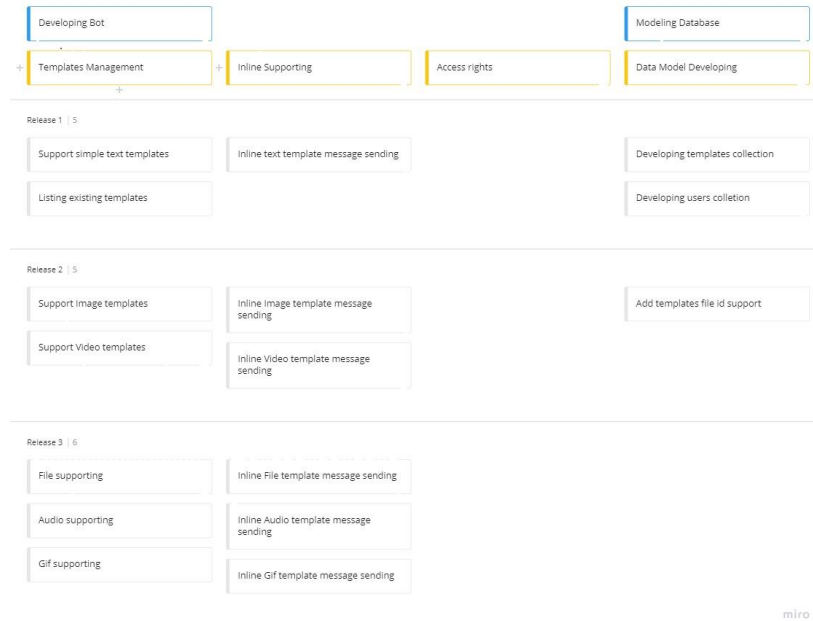


Рисунок 1.5 – Пріоритетність задач за методом Story Mapping

Після виділення пріоритетності завдань, вони були розбиті на спринти за принципом 1 реліз – 1 спринт. Кожен спринт триває 2 тижні. Для розбиття цих завдань та подальшої роботи з ними був використаний веб-додаток Clubhouse. Який дозволяє зв'язати дошку з деяким git репозиторієм та контролювати прогрес деякої історії шляхом зміни статусів обраної гілки. Приклад дошки з Clubhouse продемонстровано на рисунку 1.6

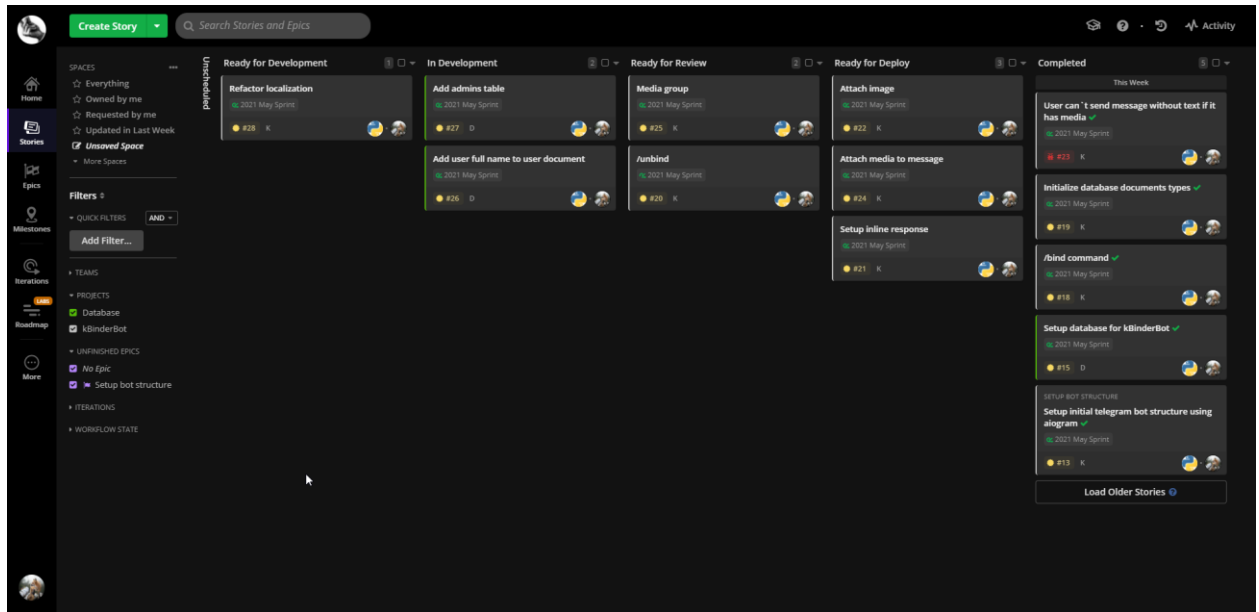


Рисунок 1.6 – дошка Clubhouse

Для початку роботи з дошкою треба обрати завдання та знайти на ньому поле, у якому зазначена спеціальна назва гілки завдяки якій сервіс буде відстежувати статус історії (Рисунок 1.7).

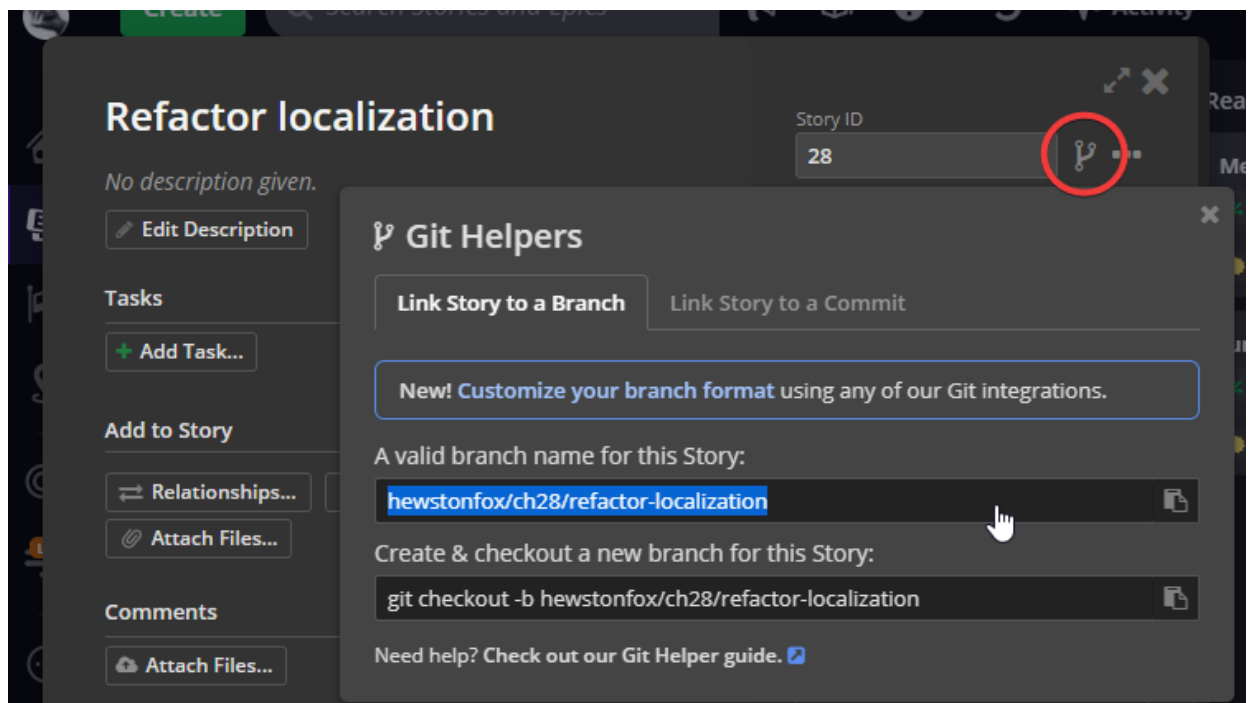


Рисунок 1.7 – Отримання корекної назви гілки

Статус історії визначається за правилами нижче:

- Одразу після створення гілки історія переходить із статусу «Готова до розробки» у статус «У розробці».
- Коли від гілки поступає запит на злиття з гілкою `dev` історія переходить у статус «Готова для перевірки»
- Після злиття цієї гілки з `dev` історія переходить у статус «Готова для розгортання»
- Після злиття цієї гілки з гілкою `master`, історія переходить у статус «Виконано»

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Діаграма варіантів використання

Діаграми варіантів використання використовуються для відображення сценаріїв використання системи та користувачів системи, які використовують її функції.

Актори на діаграмі варіантів використання позначаються символом людини, а варіанти використання – еліпсом.

Актори та варіанти використання поєднуються напрямленою асоціацією – стрілкою, що спрямована від актора до варіанта використання. Також актори можуть поєднуватися з використанням зв'язків узагальнення [7].

Діаграма варіантів використання зображена на рисунку 2.1.

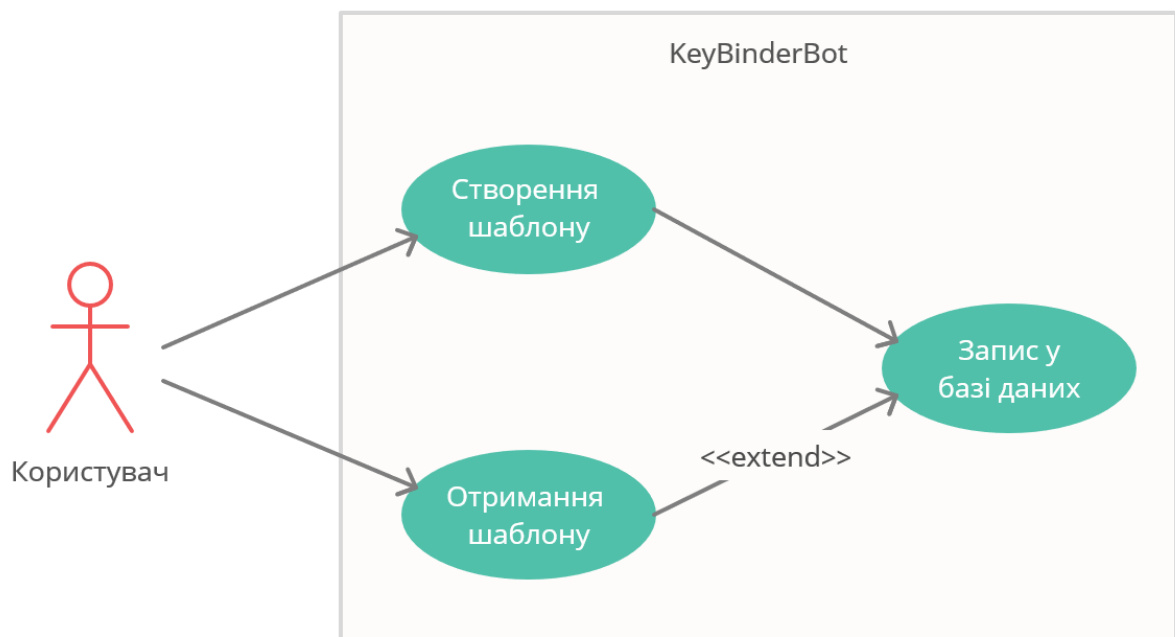


Рисунок 2.1 – Діаграма варіантів використання

Спочатку кожен користувач повинен створити шаблон, який буде збережений до бази даних. Далі користувач може зайти у будь-який чат та використовуючи бота у режимі inline отримати доступ до збереженого шаблону та надіслати його.

2.2 Діаграма розгортання

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, які існують лише на етапі її виконання (runtime). При цьому подаються тільки компоненти-екземпляри програми, які є здійсними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не відображаються. Так, компоненти з вихідними текстами програм можуть бути присутніми тільки на діаграмі компонентів. На діаграмі розгортання вони не вказуються.

Діаграму розгортання до дипломного проекту подано на рисунку 2.2.

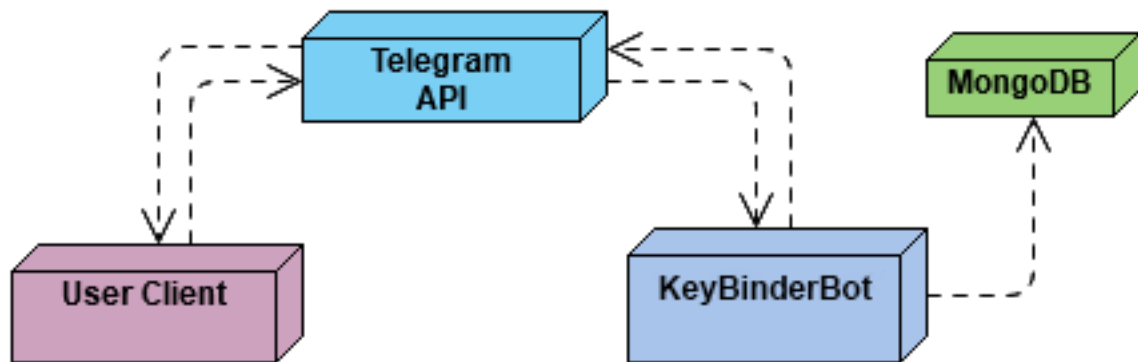


Рисунок 2.2 – Діаграма розгортання

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного представлення, діаграма розгортання є єдиною для системи в цілому, оскільки повинна повністю відображати особливості її реалізації. Ця діаграма, по суті, завершує процес ООАП для конкретної програмної системи і її розробка, як правило, є останнім етапом специфікації моделі [8]

2.3 Діаграма компонентів

Діаграма компонентів описує особливості фізичного представлення системи. Діаграма компонентів дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код. У багатьох середовищах

розробки модуль або компонент відповідає файлу. Пунктирні стрілки, що з'єднують модулі, показують відношення взаємозалежності, аналогічні тим, які мають місце при компіляції початкового програмного коду. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними [9]

Діаграма компонентів розробляється для наступних цілей:

- Візуалізації загальної структури вихідного коду програмної системи.
- Специфікації виконуваного варіанту програмної системи.
- Забезпечення багаторазового використання окремих фрагментів програмного коду.
- Уявлення концептуальної і фізичної схем баз даних.

Структура програми складається з 6 файлів. Першим запускається `main.py`, що ініціалізує бота та запускає весь потрібний функціонал. `config.py` містить у собі необхідні дані для авторизації бота у телеграм та `mongodb`. `db.py` містить у собі функції для роботи з базою даних. `bot_utils.py` дає деякі корисні функції для обробки даних. `bot_types.py` описує типи усіх сутностей, що наявні у програмі. `Requirements` це файл, у якому описані усі бібліотеки, що мають бути встановлені для інтерпретатора, що буде виконувати скрипт.

Діаграма компонентів для дипломного проекту зображена на рисунку 2.3.

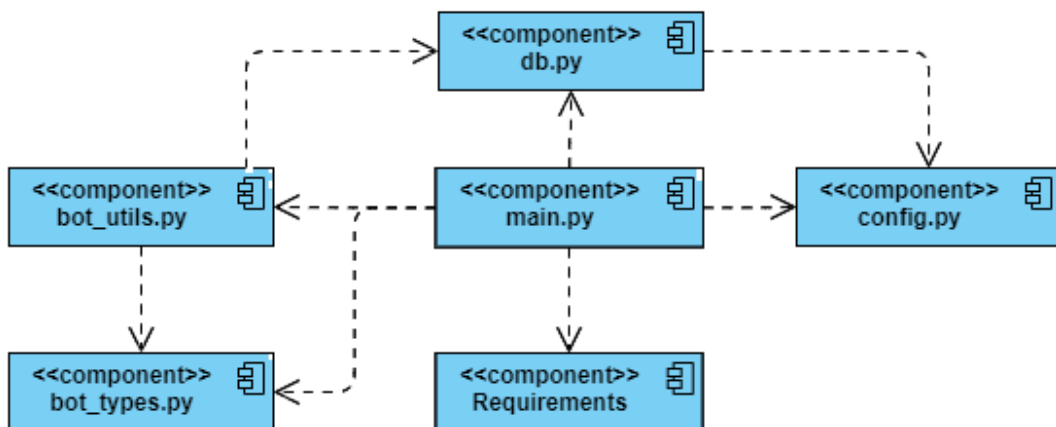


Рисунок 2.3 – Діаграма компонентів

2.4 Опис бази даних

Для збереження інформації у проекті використовується MongoDB, це нереляційна база даних. Якщо в реляційних бд вміст складають таблиці, то в mongodb база даних складається з колекцій.

Кожна колекція має своє унікальне ім'я - довільний ідентифікатор, що складається з не більше ніж 128 різних алфавітно-цифрових символів і знаку підкреслення.

На відміну від реляційних баз даних MongoDB не використовує табличний пристрій з чітко заданим кількістю стовпців і типів даних. MongoDB є документо-орієнтованою системою, в якій центральним поняттям є документ.

Документ можна уявити як об'єкт, який зберігає деяку інформацію. У певному сенсі він подібний до рядкам в реляційних СУБД, де рядки зберігають інформацію про окремий елемент [10].

Структура бд дипломного проекту зображена на рисунку 2.4.

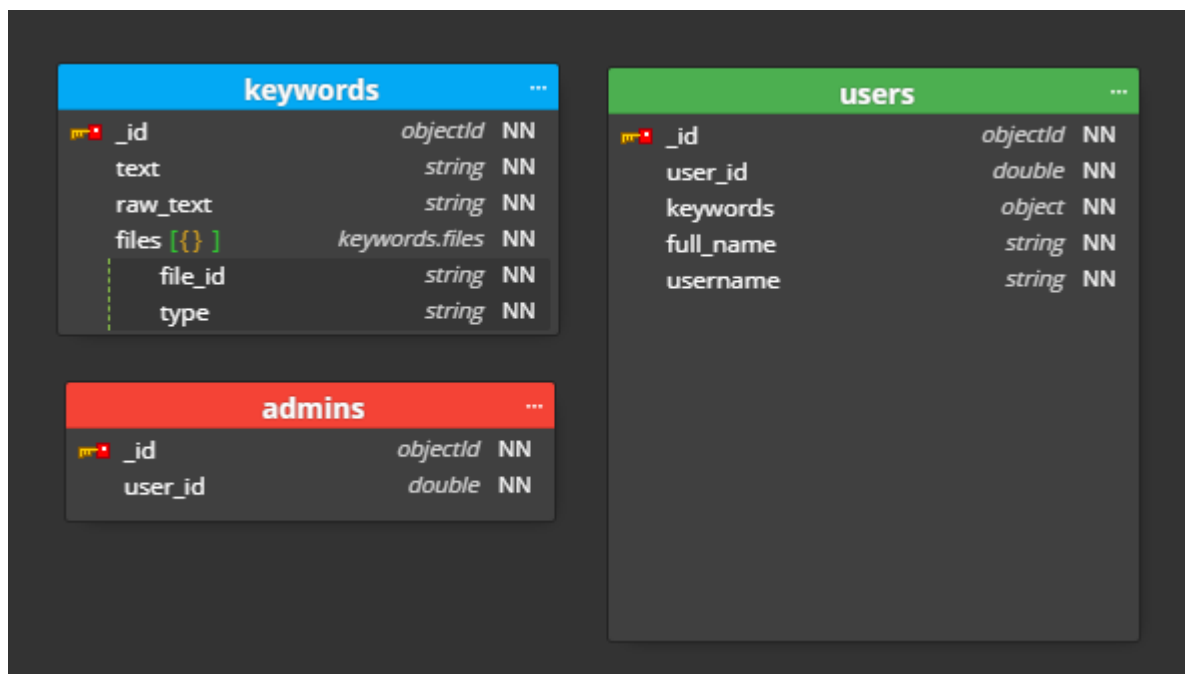


Рисунок 2.4 – Структура бази даних

Кожний документ за замовчуванням має поле «_id», у якому зберігається його ідентифікатор.

Кожен користувач з таблиці `users` має поля `«user_id»`, що є ідентифікатором користувача у Telegram та `«keywords»` що є об'єктом у якому ключ це деяке ключове слово, а значення – ідентифікатор документу з таблиці `keywords`. Решта полів використовується для наочної перевірки коректності роботи бота з базою даних та не використовується скриптом.

Шаблони усіх користувачів збережені у колекції `keywords`. Кожен шаблон має звичайний текст, `«raw_text»`, для перед перегляду, та текст з форматуванням у форматі HTML, `«text»`, що буде відправлений. Поле `files` це масив об'єктів файлу, що складаються з полів `«file_id»` - ідентифікатору файлу в Telegram та `«type»` - типу файлу, що має бути відправлений.

У колекції `admins` збережені ідентифікатори Telegram користувачів, що будуть мати певні привілеї для адміністрування контенту.

3 ПЛАНУВАННЯ ТА РЕАЛІЗАЦІЯ ТЕСТУВАННЯ

3.1 План тестування програмного продукту

Тестування розроблюваного проекту полягає у послідовному тестуванні кожного з етапів його роботи.

Етапи розробки бота можна умовно поділити на такі частини:

1 Створення шаблону

2 Відтворення шаблону з текстом

На кожному з етапів програмний продукт має не тільки виконувати свої основні функції, але й обробляти більшість непередбачуваних ситуацій, наприклад використання невірного формату для створення шаблону.

Кожна з частин є модулем для тестування. Більш детально про кількість модулів, що тестуються та кількість тест-кейсів у таблиці 3.1.

Таблиця 3.1 – Кількість тест кейсів

Модуль	Кількість тест-кейсів
Створення шаблону	6
Відтворення шаблону	6
Загально	12

Кожний модуль буде підданий тестуванням усіх вище перелічених видів. Спочатку будуть перевірені усі базові функції поточного етапу. Після їх налагодження будуть відтворені майже всі можливі ситуації, що можуть статися під час проходження поточного етапу. Останнім кроком тестування буде відтворення «живої» ситуації та її найгіршого можливого розвитку подій.

Для проведення тестування знадобиться клієнт офіційний клієнт Telegram та коло 10 файлів кожного типу.

Почати тестування можна після написання основного функціоналу бота.

Тестування можна вважати завершеним, коли частина є повністю працездатною та дає змогу перейти на наступну частину.

Часові рамки тестування можна побачити на таблиці 3.2.

Таблиця 3.2 – Часові рамки тестування

Етапи тестування	Дата початку	Дата закінчення
Складання тест-плану	08.06.2021, 12:00	08.06.2021, 13:00
Складання чек-листів для тестування	08.06.2021, 14:00	08.06.2021, 15:00
Тестування модулів	08.06.2021, 16:00	08.06.2021, 17:00
Звіт про тестування	08.06.2021, 18:00	08.06.2021, 19:30

3.2 Тест дизайн

Тест-кейси для тестування «Keyword Binder Bot» представлені у таблиці 3.3 нижче.

Таблиця 3.3 – Тест-кейси

Назва тест-кейса	Кроки для відтворення	Очікуваний результат	Реальний результат
Створення текстового шаблону	Відправити боту повідомлення з текстом «/bind text any simple text»	Бот відправить таке саме повідомлення, але без команди /bind на початку та ключового слова, що стоїть за ним	TRUE
Створення шаблону з зображенням	Відправити боту зображення з описом «/bind img image caption»	Бот відповість зображенням з описом «image caption»	TRUE
Створення шаблону з відео	Відправити боту відео з описом «/bind vid video caption»	Бот відповість відео з описом «video caption»	TRUE
Створення шаблону з файлом	Відправити боту файл з описом «/bind file file caption»	Бот відповість файлом з описом «file caption»	TRUE
Створення шаблону з аудіо	Відправити боту аудіо з описом «/bind aud audio caption»	Бот відповість аудіо з описом «audio caption»	TRUE
Створення шаблону медіа групи	Відправити боту відео та зображення в одному повідомленні з описом «/bind media media group caption»	Бот відповість медіа групою з описом «media group caption»	TRUE
Відтворення шаблону з текстом	У полі для введення повідомлення ввести «@kBinderBot text»	Над полем для вводу повідомлення повинен відобразитися шаблон,	TRUE

		при натисканні на який бот відправить в чат повідомлення з підготованим текстом	
Відтворення шаблону з зображенням	У полі для введення повідомлення ввести «@kBinderBot img»	Над полем для вводу повідомлення повинен відобразитися шаблон, при натисканні на який бот відправить в чат зображення з підготованим текстом	TRUE
Відтворення шаблону з відео	У полі для введення повідомлення ввести «@kBinderBot vid»	Над полем для вводу повідомлення повинен відобразитися шаблон, при натисканні на який бот відправить в чат відео з підготованим текстом	TRUE
Відтворення шаблону з файлом	У полі для введення повідомлення ввести «@kBinderBot file»	Над полем для вводу повідомлення повинен відобразитися шаблон, при натисканні на який бот відправить в чат файл з підготованим текстом	TRUE
Відтворення шаблону з аудіо	У полі для введення повідомлення ввести «@kBinderBot aud»	Над полем для вводу повідомлення повинен відобразитися шаблон, при натисканні на який бот відправить в чат аудіо з підготованим текстом	TRUE
Відтворення шаблону з медіа групою	У полі для введення повідомлення ввести «@kBinderBot media»	Над полем для вводу повідомлення повинний відобразитися перелік файлів з підготованого шаблону. Після вибору одного з них, він відправиться у чат з підготовленим текстом	TRUE

3.3 Звіт про тестування

Критерії закінчення тестування досягнені, було пройдено 12 тест-кейсів, на даному етапі тестування можна вважати завершеним.

При проведенні тестування критичних багів знайдено не було

4 ОПИС ПРОГРАМНОГО ПРОДУКТУ

4.1 Системні вимоги

Для роботи телеграм бота потрібно лише з'єднання з інтернетом та будь-який клієнт телеграм, однак найкраща та еталонна підтримка усіх нових функцій Telegram є у клієнтах для IOS та Android. Детальніші системні вимоги зображені на таблиці 4.1.

Таблиця 4.1 – Системні вимоги «Keyword Binder Bot»

Системні вимоги	Інтернет	Клієнт
Мінімальні	2G	Будь-який
Рекомендовані	3G+	Telegram IOS/Android

4.2 Посібник користувача

4.2.1 Початок роботи

Керувати ботом можна за допомогою деякого переліку команд.

Для початку роботи треба написати `/start`, після чого бот відповість локалізованим повідомленням (рисунок 4.2).

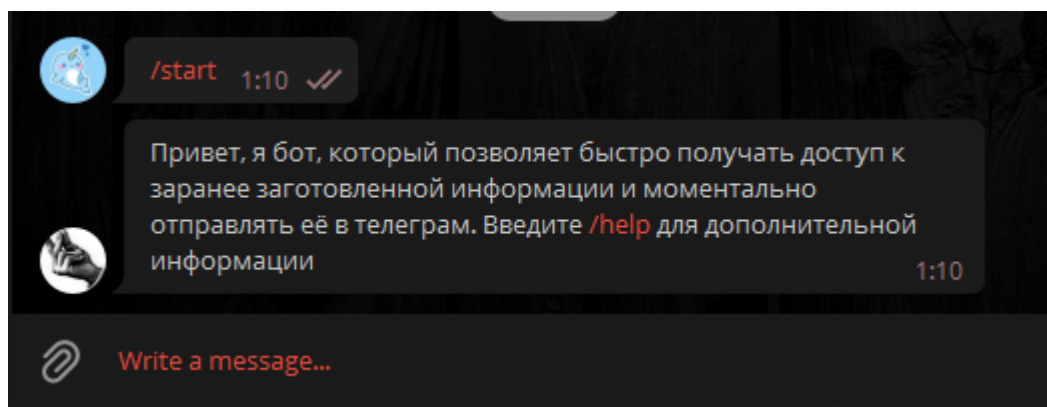


Рисунок 4.2 – результат команды `/start`

Далі треба слідкувати за вказівками та ввести (або натиснути у повідомленні) команду `/help`. Після чого бот відправить локалізоване повідомлення у якому буде описана уся потрібна інформація (Рисунок 4.3).

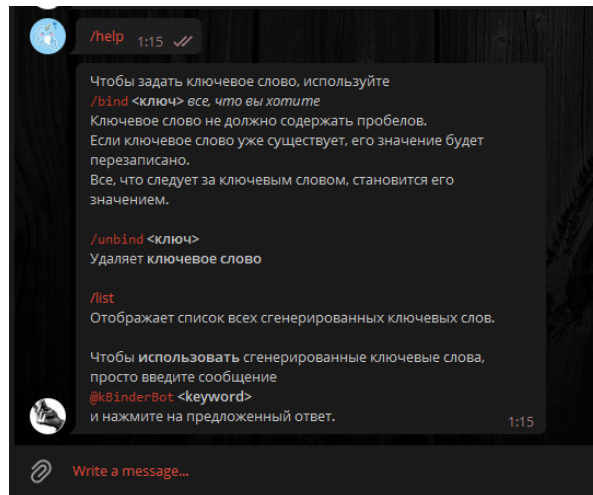


Рисунок 4.3 – Результат команды `/help`

Ділі так само слідуємо інструкцією, що була надіслана ботом. Для того щоб створити ключове слово треба використати команду `/bind`.

Бот підтримує будь-яке форматування повідомлень, що підтримується самим Telegram, це форматування може бути як у тексті повідомлення, так і в описі файлу. Після успішного виконання команди, бот надішле новий шаблон у тому вигляді, в якому він був записаний, для перед перегляду. Процес відповіді ніяк не зв'язаний з відправленим повідомленням, бот бере та інтерпретує дані, що записані у базі (Рисунок 4.4).

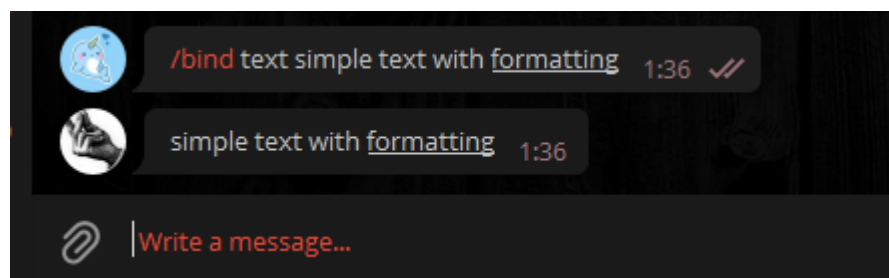


Рисунок 4.4 – текст з відтворенням форматування Telegram

У випадку неправильного, неочікуваного, неактуального, тощо результату, шаблон може бути перезаписаний використовуючи ту саму команду `/bind` з таким самим ключовим словом (у випадку на зображенні ключове слово «text»). У такому випадку попередній шаблон буде видалено.

Для відправки шаблону із зображенням (або з будь-яким іншим файлом), команду `/bind` треба додати в описі самого зображення разом з ключовим словом. Якщо до повідомлення прикріплений файл, текст після ключового слова можна не писати, у такому випадку зображення буде надіслано без опису. Однак, будь-який текст після ключового слова стане описом зображення (Рисунок 4.5).

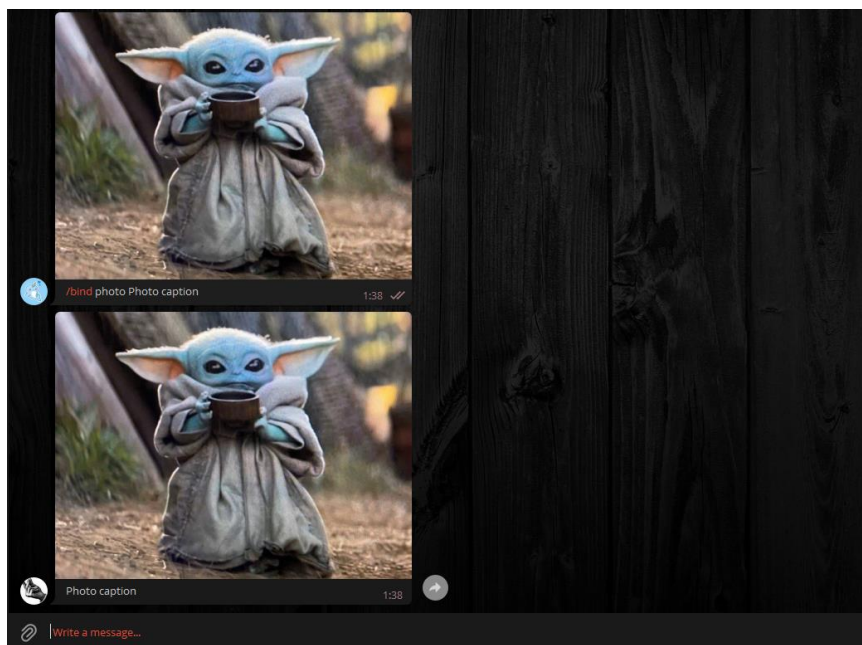


Рисунок 4.5 – Фотографія з описом

Для відправлення групи файлів використовується такий самий алгоритм, що і для відправлення одного файлу, за виключенням того, що прикріпити треба декілька файлів. Для цього треба звернути увагу на опцію «Згрупувати файли», вона має бути увімкнена (Рисунок 4.6).

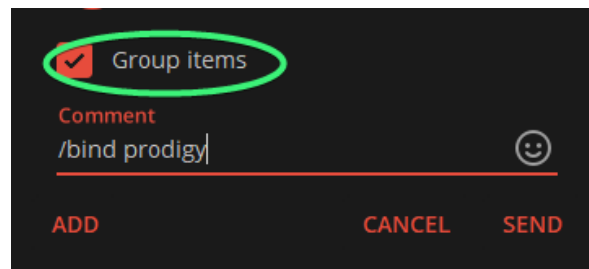


Рисунок 4.6 – Опція «Згрупувати файли»

Після того, як повідомлення буде відправлено, ми отримаємо очікувану відповідь, що зображена на рисунку 4.7. При цьому важливо прослідкувати, щоб опис був надісланий разом з файлами, а не окремо. У такому разі буде записаний лише текст після ключового слова, або станеться помилка, якщо цього тексту немає. Таким чином відправляти декілька файлів рекомендується лише через мобільний додаток, тому що Telegram Desktop відправляє опис перед повідомленням з файлами, а у клієнті Unigram аудіо файли не розпізнається як аудіо та відправляються як інші файли без можливості прослуховування усередині клієнту.

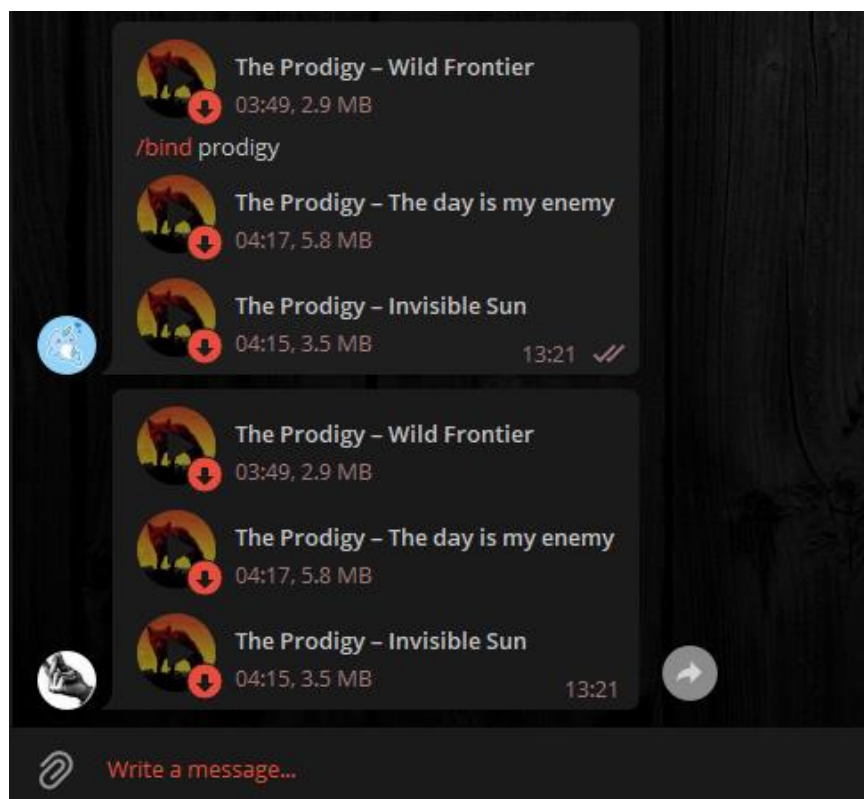


Рисунок 4.7 – Декілька аудіофайлів

Для того, щоб скористатися ботом, треба перейти у будь-який чат та ввести у поле для повідомлення «@kBinderBot» та через пробіл написати своє ключове слово, після чого бот запропонує в якості варіанту відповідей збережений шаблон (Рисунок 4.8).

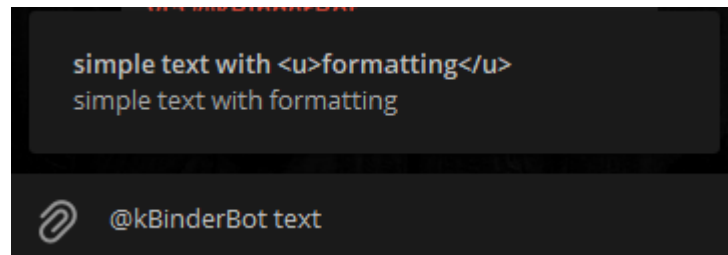


Рисунок 4.8 – Відповідь бота у режимі inline

У випадку, якщо до шаблону було додано декілька файлів, вони будуть відображатися списком у якому буде можливість обрати який саме файл має бути надісланий (Рисунок 4.9).

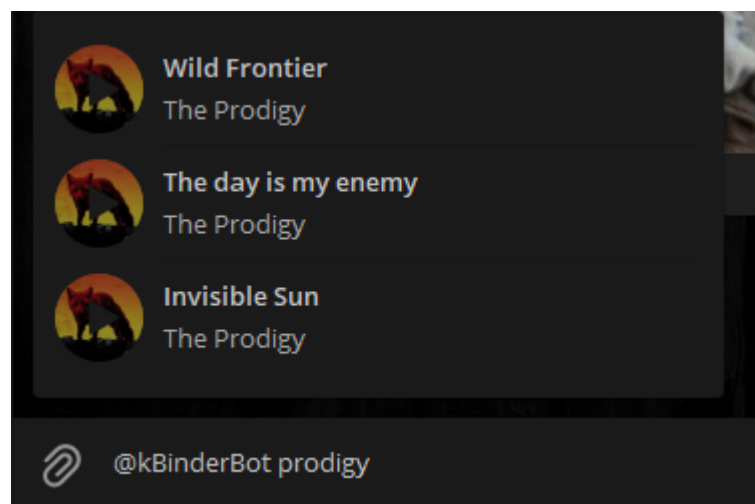


Рисунок 4.9 – Відповідь при декількох файлах у шаблоні

4.2.2 Команда /help

Команда /help відображає усю необхідну інформацію про бота у чаті. Приклад команди /help для російської локалізації зображений на рисунку 4.10.

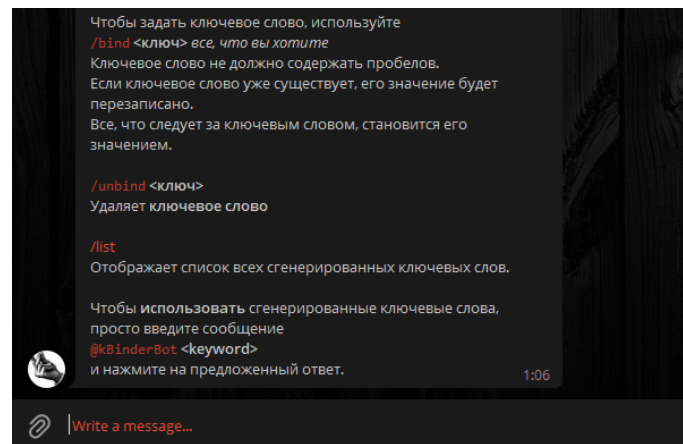


Рисунок 4.10 – результат команды /help

4.2.3 Команда /list

Команда /list відображає список наявних ключових слів у вигляді кнопок, при натисканні на які, відправиться прикріплений до цього слова шаблон для перед перегляду (Рисунок 4.11).

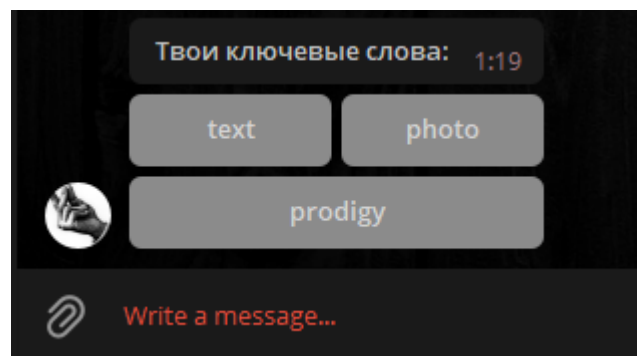


Рисунок 4.11 – результат команды /list

4.2.4 Команда /bind

Команда /bind дозволяє створювати шаблон та прив'язувати його до ключового слова або декількох ключових слів. Увесь текст, що знаходиться після ключового слова буде збережений разом із його форматуванням. Усі файли, що були прикріплені до повідомлення, відобразяться у списку під час використання для вибору який саме з них повинен бути відправлений.

Ключове слово повинно складатися з будь-яких символів окрім пробілу та коми, регістр ігнорується. Для того, щоб задати декілька ключових слів для

одного шаблону, треба лише записати усі ключові слова розділяючи їх комами (Рисунок 4.12). Кожне з цих ключових слів буде відображатися у списку окремо. Якщо ключове слово вже використовується для іншого шаблону, його значення перепишеться новим.

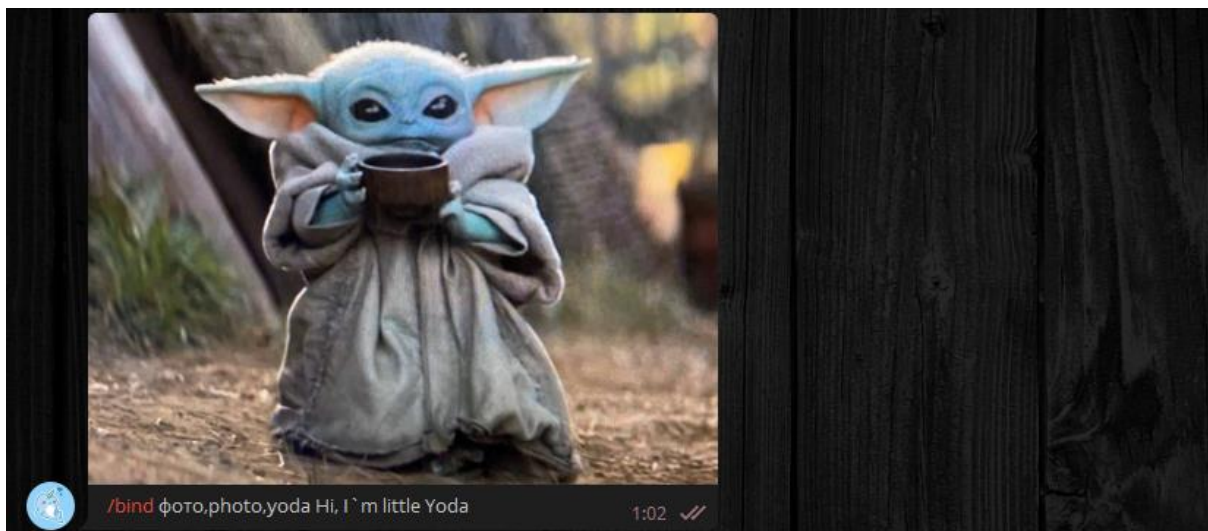


Рисунок 4.12 – Шаблон з декількома ключовими словами

4.2.5 Команда /unbind

Команда /unbind видаляє ключове слово або перелік ключових слів. Якщо жодне інше слово не відповідає тому самому шаблону – шаблон видаляється. Для видалення декількох ключових слів, ці слова мають бути перелічені комою (Рисунок 4.12).

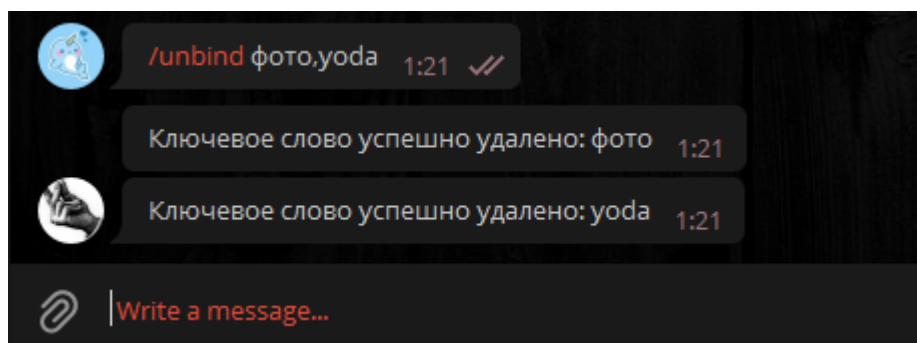


Рисунок 4.12 – Видалення ключових слів

4.2.6 Змінні

У шаблонах є можливість додавати динамічні частини, значення яких буде виділятися під час використання. Для того щоб це зробити, потрібно у тексті шаблону обернути значення за замовчуванням у фігурні дужки (якщо такого значення немає – залишити дужки порожніми). Приклад зображений на рисунку 4.13. Значення змінних не muszą містити пробілів.

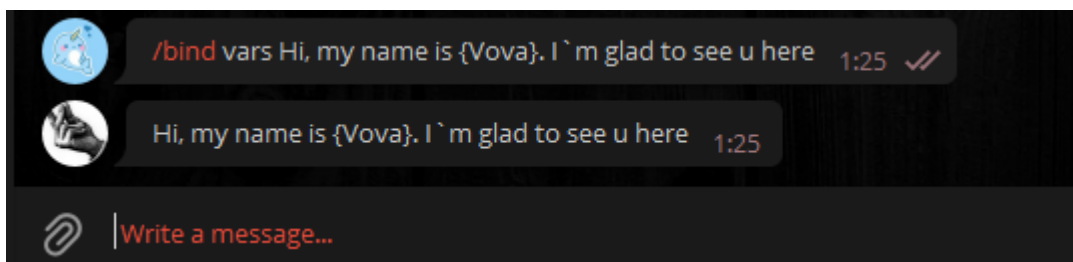


Рисунок 4.13 – Створення шаблону, що підтримує змінні

Для того, щоб змінювати ці значення під час використання треба лише передавати наступні значення одразу після ключового слова під час виклику бота у режимі inline, як показано на рисунку 4.14.

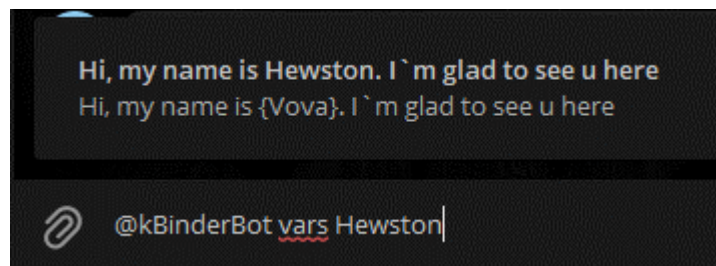


Рисунок 4.14 – Використання шаблону зі змінними

Якщо у шаблоні присутні декілька місць у яких використовуються зміні, під час використання їх значення розділяються пробілами. Для пропуску змінної (задання їй значення за замовчуванням) та доступу до наступних змінних, на місті поточної ставиться «_» (Рисунок 4.15).

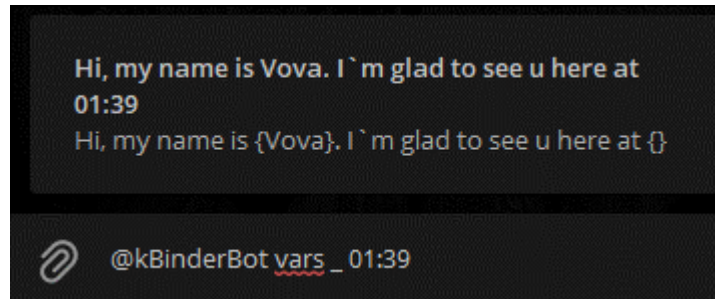


Рисунок 4.15 – Пропуск змінних

Для того, щоб використовувати формат змінної як статичний текст, перед фігурними дужками треба поставити обернений слеш (\), у такому випадку цей шаблон буде проігнорований під час підстановки значень.

Для того, щоб передати «_» у якості значення, а не як сигнал для пропуску, варто перед цим символом ставити обернений слеш (_), тоді під час підстановки до відповідної змінної підставиться значення «_».

В обох випадках у тексті буде відображатися на 1 обернений слеш менше, ніж вказано у шаблоні або під час підстановки.

5 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ

5.1 Загальні положення

В дипломному проекті розроблено алгоритм і програмне забезпечення «Keyword Binder Bot».

Техніко - економічне обґрунтування проекту передбачає проведення аналізу показників у розробника програмного продукту і споживача.

Доцільність здійснення діяльності по розробці і реалізації спеціального ПЗ визначається собівартістю робіт, рівнем рентабельності, обсягом замовлень...

Собівартість - це витрати на розробку (виробництво) і реалізацію програмного продукту. В даному проекті передбачається розрахунок кошторисної собівартості продукту.

Рентабельність - це відносний показник ефективності діяльності підприємства, який у загальному вигляді визначається як відношення прибутку до витрат. В даному проекті закладено мінімальний рівень рентабельності 35% при обсязі замовлень - 1 шт.

В проекті проводиться аналіз структури витрат та визначається точка беззбитковості (в пунктах 5.2-5.10).

Обґрунтування діяльності використання програмного продукту у споживача передбачає розрахунок різниці приведених витрат при використанні базової та нової програми та строку окупності інвестицій.

5.2 Визначення трудомісткості розробки

Загальна трудомісткість розробки складається з трудомісткості виконання окремих етапів роботи, які зведені до таблиці 5.1. Трудомісткість робіт (Т) визначається за допомогою експертних оцінок і вимірюється в годинах (або в інших одиницях: днях, місяцях...).

Таблиця 5.1 – Етапи робіт і їх трудомісткість

№ з/п	Етапи робіт	Виконавець, посада	Трудомісткість робіт, Ті чоловіко-годин
-------	-------------	--------------------	---

1	2	3	4
1	Отримання технічного завдання	developer	1
2	Вивчення предметної області та постановка задачі	developer	8
3	Побудова математичної моделі	developer	4
4	Розробка алгоритму	developer	24
5	Розробка програми	developer	112
6	Налагодження та тестування програми	developer	40
7	Оформлення пояснювальної записки	developer	168
	Разом		T = 357

В сучасній ІТ компанії до виконання етапів робіт в залежності від проекту може бути залучений як декілька працівників, так і декілька десятків. Найбільш трудомісткими є створення проектів в сфері гейм індустрії. Зокрема при розробці віртуальної гри над проектом можуть працювати наступні працівники: 2D artist, 3D artist, designer, developer, QA engineer, Team Lead, manager та інші. Кількість виконавців залежить від складності і обсягу робіт можливості і необхідності паралельного виконання етапів проекту.

5.3 Розрахунок витрат на заробітну плату та єдиного соціального внеску

Заробітна плата персоналу, що безпосередньо виконує роботи з розробки програмного продукту визначається на основі штатно-окладної форми оплати за формулою:

$$ЗП_i = \frac{O_i \cdot T_i \cdot K_d}{H} \quad (5.1)$$

де O_i - оклад i - го працівника, грн. (Прийняти 15000÷30000 грн.);

T_i - трудомісткість робіт i -го працівника (прийняти з таблиці 5.1), чоловіко-годин;

K_d коефіцієнт додаткової оплати праці, 1,35;

H - нормативна кількість годин в місяці, прийняти умовно 150 годин.

Розрахунок витрат на заробітну плату виконується в таблиці 5.2. Загальні витрати на заробітну плату визначаються як сума зарплат всіх працівників, що безпосередньо виконують роботи з розробки програмного продукту:

$$З = \sum_1^i ЗП_i \quad (5.2)$$

Таблиця 5.2 – Витрати на заробітну плату

№ з/п	Виконавець	О _і , грн.	Т _і , чоловіко-годин	Н, годин	К _д	ЗП _і , грн.
1	2	3	4	5	6	7
1	Developer	25000	357	150	1,35	80325
	Разом					З=80325

Єдиний соціальний внесок визначається за формулою:

$$C = \frac{З \cdot Пс}{100} \quad (5.3)$$

де Пс - процент єдиного соціального внеску, 22%.

$$C = 80325 \cdot 22 / 100 = 17671,5 \text{ грн}$$

5.4 Розрахунок витрат на матеріали

Витрати на матеріали визначаються згідно з нормами витрат та ціною на них за формулою:

$$В_m = Ц_m \cdot Р_m \quad (5.4)$$

де Ц_т - ціна т - матеріалу, грн.;

Р_т - норма витрат т - матеріалу на проект.

Розрахунок виконується в таблиці 5.3.

Таблиця 5.3 – Розрахунок витрат на матеріали

№ з/п	Найменування матеріалів	Одиниця	Ц _т , грн.	Р	В _т , грн.
1	2	3	4	5	6
1	Папір	Аркушів	0.5	30	15
2	Диски	шт.	2	3	6
3	Папки	Шт.	15	2	30
	Разом				ВМ=51

Загальні витрати на матеріали складуть:

$$BM = \sum_1^m Bm \quad (5.5)$$

5.5 Витрати на електроенергію для технологічних потреб

Витрати на електроенергію при експлуатації технічних засобів визначаються за формулою:

$$E_n = N_n \cdot C_{el} \cdot (T \cdot K_{\kappa} \cdot n) \quad (5.6)$$

де N_n - потужність n - го обладнання, кВт;

C_{el} - ціна 1 кВт/год електроенергії, грн.;

$(T \cdot K_{\kappa})$ - час роботи n - го обладнання, годин;

K_{κ} - коефіцієнт використання n - го обладнання протягом періоду розробки проекту.

Розрахунок виконується в таблиці 5.4.

Таблиця 5.4 – Розрахунок витрат на електроенергію

№ з/п	Обладнання	N, кВт	T, годин	K_{κ}	C_{el} , грн.	E_n , грн.
1	2	3	4	5	6	7
1	Ноутбук	0.1	357	1	2,9393	104,93
	Разом					$E = 104,93$

Загальні витрати на електроенергію складуть:

$$E = \sum_1^n E_n \quad (5.7)$$

5.6 Розрахунок амортизаційних відрахувань

Амортизаційні відрахування визначаються окремо для таких груп основних фондів: приміщення (площі), обладнання за формулою:

$$A = \frac{N_a \cdot C \cdot T}{100 \cdot F_{\phi}} \quad (5.8)$$

де N_a - норма амортизації основних фондів. (Для приміщення $N_a=5\%$, устаткування $N_a=25\%$);

T - трудомісткість робіт (з таблиці 1.1), годин;

C - вартість основних фондів, грн.;

F_d - річний фонд часу роботи основних фондів. (Прийняти 1870 годин).

Вартість основних фондів, приміщення, розраховується за формулою:

$$C_{пл} = C_{пл} \cdot R \cdot f \quad (5.9)$$

де C - вартість 1м² площі, прийняти 800 грн.;

R - кількість працівників, осіб;

f - площа службово-побутових приміщень на одного працівника, прийняти 7м².

Вартість основних фондів - комп'ютера, принтера, програмного забезпечення та іншого устаткування, - визначається за формулою:

$$C_{уст} = \sum_1^B (C_{уст} \cdot S_{уст}) \cdot K \quad (5.10)$$

де $C_{уст}$ - ціна кожного виду устаткування (складається з $C_{пр}$ - ціни принтера, $C_{к}$ - ціни комп'ютера, $C_{пз}$ - загального системного програмного забезпечення та ін.), грн.;

S - кількість устаткування кожного виду, шт;

K - коефіцієнт, що враховує витрати на доставку, монтаж, налаштування, тощо, прийняти 1,1;

v - кількість видів устаткування.

Загальні амортизаційні відрахування визначаються як сума амортизаційних відрахувань від вартості обладнання (комп'ютерів, принтерів, ПЗ) та амортизаційних відрахувань від приміщень (площі):

$$A_{заг} = A_{пл} + A_{уст} \quad (5.11)$$

$$C_{пл} = 800 \cdot 1 \cdot 7 = 5600 \text{ грн.}$$

$$A_{пл} = (5600 \cdot 357 \cdot 5) / (100 \cdot 1870) = 53,45 \text{ грн.}$$

$$C_{уст} = (21000 \cdot 1) \cdot 1,1 = 23100 \text{ грн.}$$

$$A_{уст} = (23100 \cdot 357 \cdot 25) / (100 \cdot 1870) = 1102,5 \text{ грн.}$$

$$A_{заг} = 53,45 + 1102,5 = 1155,95 \text{ грн.}$$

5.7 Накладні витрати

Накладні витрати враховують витрати на опалення, освітлення, охорону, рекламу, управління, організацію, загальногосподарські потреби та ін.

Розмір накладних витрат визначається пропорційно фонду заробітної плати за формулою:

$$HB = \frac{3 \cdot \Pi_{н.в.}}{100} \quad (5.12)$$

де $\Pi_{н.в.}$ - процент накладних витрат. (Прийняти 20÷30%).

$$HB = 80325 \cdot 25 / 100 = 20081,25 \text{ грн.}$$

5.8 Розрахунок кошторисної вартості та ціни програмного продукту

Загальна кошторисна вартість визначається як сума витрат за статтями, що розраховані в розділах 1.2 - 1.6 за формулою:

$$K = 3 + C + BM + E + A \quad (5.13)$$

При реалізації розробленого програмного продукту в одному екземплярі відпускна ціна формується з урахуванням нормативного рівня рентабельності, що забезпечує мінімально допустиме значення прибутку підприємства.

$$\Pi_B = K + \Pi_n \quad (5.14)$$

де Π_n - нормативний прибуток, грн.

$$\Pi_n = K \cdot R_n \quad (5.15)$$

де R_n - нормативна рентабельність, прийняти 0,35.

Ціна з врахуванням податку на додану вартість складе:

$$\Pi_{ПДВ} = \Pi_B \cdot \left(1 + \frac{\Pi_{ПДВ}}{100} \right) \quad (5.16)$$

де $\Pi_{ПДВ}$ - процент податку на додану вартість (20%).

Розрахунок кошторисної вартості і ціни провести в таблиці 5.5.

Таблиця 5.5 – Розрахунок кошторисної вартості і ціни

№ з/п	Найменування статті	Позначення	Сума, грн.	Питома вага в кошторисі, %.
1	2	3	4	5
1	Витрати на заробітну плату	З	80325	67.28
2	Єдиний соціальний внесок	С	17671,5	14.80
3	Витрати на матеріали	ВМ	51	0.04
4	Витрати на електроенергію	Е	104,93	0.09
5	Амортизаційні відрахування	А _{заг}	1155,95	0.97
6	Накладні витрати	НВ	20081,25	16.82
•	Разом кошторисна вартість	К	119389,63	100
•	Нормативний прибуток	П _н	412786,37	-
•	Відпускна ціна	Ц _в	161176	-
•	Ціна з врахуванням податку на додану вартість	Ц _{пдв}	193411,2	-

На підставі виконаних розрахунків будується діаграма.

5.9 Аналіз постійних і змінних витрат

При збільшенні кількості замовників даного програмного продукту (більше, ніж один) з'являються додаткові витрати:

- витрати на тиражування;
- витрати на адаптацію програмного продукту до вимог споживача.

Витрати на тиражування (В_т) можна прийняти 150 грн. на кожному одиницю замовленого програмного продукту.

Витрати на адаптацію (В_а) згідно з експертними оцінками і з врахуванням середнього рівня уніфікації розробленої програми прийняти 20-50% від витрат на заробітну плату:

$$B_A = \frac{3 \cdot P_A}{100} \quad (5.17)$$

де П_А - процент витрат на адаптацію (20-50%).

$$B_a = \frac{80325 \cdot 40}{100} = 32130 \text{ грн.}$$

Вказані витрати є умовно-змінними, тобто такими, що прямо залежать від обсягу замовлених програм, або кількості замовників.

Витрати, які вказані в колонці 2 з 1 по 6 пункти в таблиці 5.21, є умовно-постійними, тобто такими, що не змінюються при збільшенні кількості замовлень даного програмного продукту.

Відпускна ціна програмного продукту при обсязі продажу N штук визначається за формулою:

$$C_B^N = \left(\frac{K+B_A}{N} + B_T \right) * (1 + P_H) \quad (5.18)$$

Ціна з врахуванням податку на додану вартість складе:

$$C_{ПДВ}^N = C_B^N \cdot \left(1 + \frac{P_{ПДВ}}{100} \right) \quad (5.19)$$

Розрахунки виконуються в таблиці 1.6 для де-кількох варіантів обсягів замовлень $N_1 = 1$ шт.; $N_2 = 1000$ шт. Значення показників при $N_1 = 1$ шт. прийняти з таблиці 5.6.

Таблиця 5.6 – Розрахунок загальних витрат і ціни при різних обсягах замовлень

№ з/п	Показники	Позначення	При обсязі замовлення N шт.	
			1 шт.	1000 шт.
1	2	3	4	5
1	Постійні витрати на 1 програму	K	80325	80325
2	Змінні витрати на програму на адаптацію	B_A		32130
3	Загальні витрати на 1 програму	$3B$	80325	112455
•	Випускна ціна одиниці програмного продукту	C_B^N	108438,75	151,81
•	Ціна з врахуванням податку на додану вартість одиниці програмного продукту	$C_{ПДВ}^N$	130126,5	182,18

5.10 Визначення точки беззбитковості

Для визначення обсягу реалізації програмного продукту, що забезпечує рентабельну діяльність проводиться аналіз зв'язку витрат, обсягу та прибутку.

Точка беззбитковості - це такий обсяг продажу, коли доходи (без ПДВ) від реалізації продукту дорівнюють витратам, а його подальше збільшення приведе до появи прибутку.

Точка беззбитковості визначається за формулою:

$$N_B = \frac{K}{C_{B.B.} - (B_A + B_T)} \quad (5.23)$$

де K - постійні витрати, грн.;

$C_{B.B.}$ - відпускна ціна за одиницю продукції без ПДВ, грн.;

$(B_A + B_T)$ - змінні витрати на одиницю продукції.

Значення постійних витрат (K) прийняти з таблиці 1.6 для $N = 1$ шт.

Відпускна ціна, для якої виконується розрахунок ($C_{B.B.}$) враховує ємність ринку, купівельну спроможність замовників, рівень конкуренції та інші фактори. В розрахунку приймаємо відпускну ціну одиниці програмного продукту для $N = 1000$ шт. = 151,81 грн. (Таблиця 5.22).

Змінні витрати ($B_A + B_T$) прийняти з таблиці 1.6 для $N = 1000$ шт.

В нашому випадку витрат на тираж немає, а витрати на адаптацію виникають тільки при налагоджуванні другого та послідовуючих примірників програмного забезпечення. Таким чином витрати на адаптацію на одиницю продукції становить: $32130 / 999 = 32,16$ грн.

$$N_B = \frac{80325}{151,81 - 32,16} \approx 671$$

Точка беззбитковості може бути визначена графічно, на підставі вихідних даних, що наведені в таблиці 5.7.

Таблиця 5.7 – Вихідні дані для визначення точки беззбитковості

Показник	Позначення, розрахунок	Значення і розрахунок показника, в грн.	
		При обсязі замовлення N шт.	
		1 шт.	1000 шт.

Постійні витрати на весь обсяг замовлення	K	80325	80325
Змінні витрати на весь обсяг замовлення	$N \times (B_A + B_T)$	-	$999 \times 32,16 = 32127,84$
Валові витрати	$N \times (B_A + B_T) + K$	80325	$80325 + 32127,84 = 112452,84$
Виручка від реалізації при ціні Ц _{В.Б.1} 79.56 грн.	$N \times Ц_{В.Б.}$	151,81	$151,81 \times 1000 = 151810$
Виручка від реалізації при ціні Ц _{В.Б.2} 70 грн.	$N \times Ц_{В.Б.}$	120	120000

Отримані вихідні дані проілюстровано на графіку рисунку 5.8.

На графіку показати лінії залежності від обсягу замовлень:

- постійних витрат;
- змінних витрат;
- валових витрат;

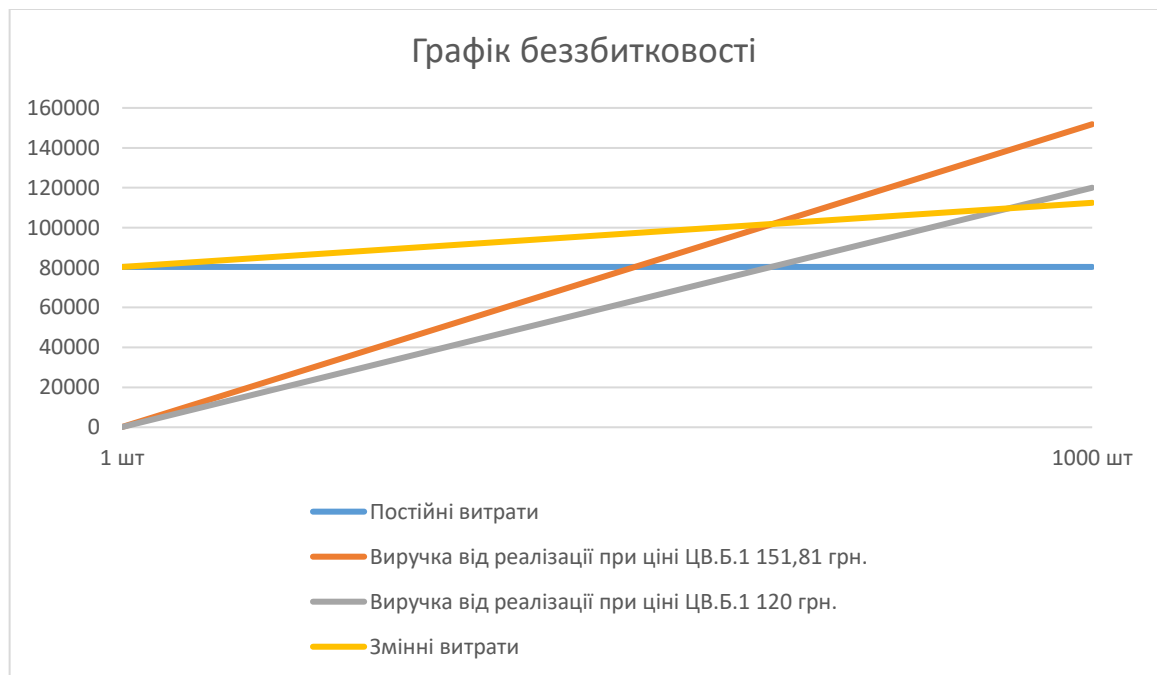


Рисунок 5.8 - Графік беззбитковості

Аналогічний розрахунок виконується для іншої ціни Ц_{В.Б.2} і наводиться графік беззбитковості.

$$N_B = \frac{80325}{120 - 32,16} \approx 914$$

5.11 Обґрунтування використання програмного продукту

Обґрунтування діяльності використання програмного продукту у споживача передбачає розрахунок різниці приведених витрат при використанні базової та нової програми та строку окупності інвестицій.

Річний економічний ефект визначається за формулою:

$$E_p = V_{\text{пр.б.}} - V_{\text{пр.н.}} \quad (5.26)$$

де $V_{\text{пр.б.}}$ - приведені витрати при використанні базового ПЗ(або взагалі без ПЗ), грн.;

$V_{\text{пр.н.}}$ - приведені витрати при використанні нового ПЗ, грн.;

$$V_{\text{пр}} = C + E_n \cdot K_{\text{пр}} \quad (5.27)$$

де C - собівартість використання ПЗ з розв'язання задачі, грн.;

E_n - нормативний коефіцієнт економічної ефективності;

K - сума капітальних вкладень.

При розрахунку приведених витрат враховується вартість придбаного обладнання, програмного продукту, експлуатаційні витрати, час і розмір інвестицій, рівень інфляції. Бажаний термін окупності вкладень до 2-х років.

Для виконання таких розрахунків в даному дипломному проекті відсутні вихідні дані.

В цілому використання у споживача розробленого боту «Keyword Binder Bot» дозволяє:

- скоротити терміни виконання робіт;
- усунути помилки в роботі;
- зменшити навантаження на працівників;
- підвищити загальну культуру виконання робіт;
- скоротити окремі види ресурсів.

5.12 Висновок

Трудомісткість проекту 357 годин.

Кількість виконавців 1 особа.

Кошторисна вартість розробки при замовленні 1 шт. 80325 грн.

Кошторисна вартість розробки при замовленні 1000 шт. 112453 грн.

Ціна програмного продукту (без ПДВ) 108438,75 грн. при замовленні 1 шт.

Ціна програмного продукту (без ПДВ) 151,81 грн. при замовленні 1000 шт.

Точка беззбитковості 671 шт. при ціні 151,81 грн.

Точка беззбитковості 914 шт. при ціні 120 грн.

6 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Загальні положення охорони праці

У даному розділі було розглянуто загальні положення з охорони праці та її основного завдання по забезпеченню безпечних та нешкідливих умов праці.

В цьому розділі по охороні праці було розпочато з аналізу небезпечних та шкідливих чинників, які можуть діяти на техника-програміста відповідно до класифікації шкідливих та небезпечних виробничих факторів, наведеної в ГОСТ 12.0.003-74 ССБТ «Вредные и опасные производственные факторы. Классификация», ці фактори і джерела їх виникнення перелічені у таблиці 6.1.

Відомо, що одним із принципових напрямків захисту від дії шкідливих та небезпечних факторів виробничого середовища є нормування за відповідними нормативно-технічними актами, тому надалі для кожного з приведених у таблиці 6.1 виробничих факторів указано нормативні регламентуючі документи, джерела виникнення.

Таблиця 6.1 – Шкідливі і небезпечні фактори виробничого середовища

№ п/п	Назва фактору	Джерела виникнення	Регламентуючі документи
1	2	3	4
1	Незадовільне освітлення	Нераціональна організація освітлення та робочих місць	ДБН В.2.5-28-06
2	Несприятливі метеорологічні умови: -температура повітря -відносна волога -швидкість руху повітря	Неефективна робота системи вентиляції та обігріву	ГОСТ 12.1.005-88
3	Підвищений рівень шуму	Системний блок, охолоджуючі системи, робота окремої апаратури	ГОСТ 12.1.003-83, ДСанПіН 3.3.2-007-98
4	Вібрація	Системний блок, охолоджуючі системи, робота окремої апаратури	ГОСТ 12.1.012-90, ДСанПіН 3.3.2-007-98
5	Підвищений рівень електромагнітних випромінювань	Монітор	ГОСТ 12.1.006-84, ДСанПіН 3.3.2-007-98

Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічно-обґрунтованого положення тіла з врахуванням його індивідуальних характеристик;
- вибір ергономічно-обґрунтованих меблів;
- раціональну розстановку обладнання на робочому місці;
- урахування характеру та особливостей трудової діяльності.

Наявність шкідливих чинників приводить до виникнення необхідності вирішувати задачі по забезпеченню виробничої безпеки при розробці «Keyword Binder Bot».

6.2 Аналіз умов праці

6.2.1 Вимоги до приміщення при експлуатації ПК

Аналіз умов праці починається з опису виробничого приміщення.

Робота виконувалась в лабораторії №42 Харківського комп'ютерно-технологічного коледжу НТУ «ХПІ».

Загальна характеристика офісного приміщення наведена у таблиці 6.2.

№п/п	Показник	Його призначення	Критерій (обґрунтування вибору)	Регламентуючі документи
1	2	3	4	5
1	Габарити приміщення, кількість робочих місць	42 кв. м, 10 робочих місць	норма площі – не менш 6м ² на людину, не відповідають нормативним вимогам	НПАОП 0.00-1.28-2010
2	Поверх, поверховість будівлі	3 поверх, 4 поверхи	Відповідає будівельним проектам	СНиП 2.09.02-85, НПАОП 0.00-1.28-2010
3	Вид природного освітлення, азимут	Одностороннє бокове, південний схід	Переважають робота з документами, необхідна увага	ДБН В.2.5-28-06
4	Вид штучного освітлення, джерела світла	Комбіноване, ЛБ-40	Переважають робота з документами, необхідна увага	ДБН В.2.5-28-06
5	Клас приміщення з безпеки поразки	підвищеної безпеки	існує можливість одночасного дотику людини до тих металоконструкцій будівель, що мають з'єднання із землею, з	ПУЭ-87

	електричним струмом		одного боку, і до металевих корпусів електрообладнання – з іншого	
6	Категорія приміщення з пожежної безпеки	В	Тверді матеріали, що здатні згорати (папір, деревина та інші)	НАПБ Б.03.002-2007
7	Клас зони приміщення з пожежної безпеки	П – ПА	Присутні тверді горючі речовини	ПУЭ-87
8	Ступінь вогнестійкості будівельних конструкцій	I-II	Не нижче II, для семиповерхової будівлі категорії В, і тому що оснащено ПЕОМ	ДБН В.1.1-7-02

Таблиця 6.2 – Загальна характеристика офісного приміщення

Користуючись ДНАОП 0.00-1.31-99 "Правила охорони праці під час експлуатації електронно-обчислювальних машин" визначити вимоги щодо облаштування робочих місць, рівнів шуму та вібрації. Рівні шуму на робочих місцях користувачів ПК не повинні перевищувати значень, встановлених ГОСТ Р50923-96 и СанПіН 2.2.2./2.4.1340-03.

6.2.2 Вимоги до мікроклімату

Метеорологічні умови на постійних робочих місцях, які визначаються температурою, відносною вологістю і швидкістю руху повітря в приміщенні, повинні вибиратися згідно з вимогами ГОСТ 12.1.005-88 ССБТ, ДСН.3.3.6.042-99 «Загальні санітарно-гігієнічні вимоги до повітря робочої зони» визначили метеорологічні умови з урахуванням категорії робіт за енерговитратами, характеристиками виробничого приміщення (табл. 6.1).

Оскільки об'єкти проектування і людина у більшості випадків знаходяться у тих самих умовах навколишнього середовища, необхідно забезпечити узгодження параметрів означених умов для людини і об'єкту.

Дані занесені в таблицю 6.3.

Таблиця 6.3 – Параметри мікроклімату

Категорія робіт згідно енерговитратам	Період року	Температура $t, ^\circ\text{C}$	Відносна вологість, $\varphi \%$	Швидкість руху повітря, $V, \text{м/сек}$
Легка – I а	Теплий	23 – 25	60 – 40	0,1
Легка – I б	Теплий	22 – 24	60 – 40	0,2

Для підтримки в приміщенні даних параметрів мікроклімату відповідно до вимог ДСТУ Б А.3.2-12:2009, ДБН.В.2.5.-67 є кондиціювання та централізоване опалення (загальне парове), загальна механічна вентиляція.

Іонізуюче випромінювання та розряди статичної електрики призводять до іонізації повітря робочої зони. Нормативні рівні концентрації позитивних та негативних іонів повинні відповідати вимогам ГН 2152-80. Дані занесені у таблицю 6.4.

Таблиця 6.4 – Рівень іонізації повітря

Рівні	Кількість іонів в 1 см ³ повітря	
	n^+	n^-
Мінімальні значення	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

6.2.3 Вимоги до освітлення робочих місць користувачів ПК

Згідно з ДБН В.2.5-28-2006 (для нових будівель) та СНиП II-4-79 (для старих будівель), для об'єктів, які світяться, відповідно до розміру об'єкту розрізнення та характеристики зорової роботи визначені нормативні характеристики зорової роботи та занесені до таблиці 6.5.

Таблиця 6.5 – Нормативні характеристики зорової роботи

Характеристика зорової роботи	Найменший розмір об'єкта розпізнавання, мм	Розряд зорової Роботи	Підрозряд зорової роботи	Контраст об'єкта розпізнавання з фоном	Характеристика фону	Освітленість при штучному освітленні, лк	КПО, e_n , при природному освітленні, %
						Загальному	Боковому

Середн . точн.	Від 0,5 до 1	IV	а	Середній	Середній	300	0,9
-------------------	-----------------	----	---	----------	----------	-----	-----

Для умов м. Харкова, необхідно розрахувати нормативне значення коефіцієнта природної освітленості, згідно з ДБН В.2.5-28-06 за формулою (6.1):

$$e_N = e_H \cdot m_N \quad (6.1)$$

де e_H – КПО за характеристикою зорової роботи;

m_N – коефіцієнт, що враховує умови освітлення 0,9

$$e_N = e_H \cdot m_N = 0,9 \cdot 0,9 = 0,81$$

В лабораторії є два види освітлення: природне (в світлу частину доби) та штучне (у темну).

Природне – бокове одностороннє.

Для загального рівномірного освітлення використані світильники серії ПВЛМ-ДР з люмінесцентними лампами типу ЛД-40.

Згідно з ДБН В.2.5-28-2006 (для нових будівель) та СНиП II-4-79 (для старих будівель) визначено вимоги до системи освітлення робочого місця користувача ПК.

6.3 Електробезпека

В даному підрозділі було розглянуто питання, які пов'язані з безпекою при роботі на електрообладнанні згідно вимогам ПУЕ, ПБЕ, ГОСТ 12.2.007.0-75, та ін. нормативно-технічним документам.

Визначено вимоги до осіб, які допускаються до роботи на ПК, засоби ті заходи, які забезпечують електробезпеку.

Електробезпека забезпечується комплексом конструктивних, схемно-конструктивних та експлуатаційних засобів і заходів захисту. Системи електропостачання, монтаж силового електрообладнання і електричного освітлення відповідають вимогам ПУЕ, ПБЕ, ГОСТ 12.2.007.0-75, та ін. нормативно-технічним документам.

Електробезпека забезпечується комплексом конструктивних, схемно-конструктивних й експлуатаційних засобів і заходів захисту.

Конструктивні заходи захисту призначені для запобігання дотику людини до струмоведучих частин електроустаткування, для цього апаратура знаходиться в ізолюючому корпусі, а дроти – ізольовані. Корпуси розкривають після відключення від живлення відповідно до вимог ПУЕ-87. Ступінь захисту електроапаратури прийнятий IP-44, де перший знак «4» захист від твердих тіл розміром 1 мм, другий знак «4» – захист від бризок (ГОСТ 14254-96).

Сучасний пристрій, що реагує на диференціальний струм, разом з пристроями захисту від надструму, відноситься до додаткових видів захисту.

Пристрій захисного відключення призначений для захисту людини від ураження електричним струмом при несправностях електроустаткування або при контакті з тими, що знаходяться під напругою частин електроустановки, а також для запобігання спалахам і пожежам, викликаним струмами витоку і замикання на землю, що виникає унаслідок тривалого протікання струмів витоку і струмів короткого замикання, що розвиваються з них.

При малих струмах замикання, зниженні рівня ізоляції, а також при обриві нульового захисного провідника занулення недостатньо ефективно, тому в цих випадках УЗО є єдиним засобом захисту людини від електричного ураження. Ці функції не властиві звичайним автоматичним вимикачам, що реагують лише на перевантаження або коротке замикання.

У основі дії захисного відключення, як електрозахисного засобу, лежить принцип обмеження (за рахунок швидкого відключення) тривалості протікання струму через тіло людини при ненавмисному дотику його до елементів електроустановки, що знаходяться під напругою.

Зі всіх відомих електрозахисних засобів УЗО є єдиним, що забезпечує захист людини від поразки електричним струмом при прямому дотику до однієї з токоведучих частин.

Особливостями УЗО є висока надійність, простота монтажу і невеликі розміри. Використання таких пристроїв дозволяє досягти високого ступеня захисту.

Схемно-конструктивні засоби захисту знижують небезпеку дотику людини до не струмоведучих струмопровідних частин електричних пристроїв при випадковому пробіі ізоляції й виникненні електричного потенціалу на них

Живлення установки здійснюється від однофазної трипровідної мережі з заземленою нейтраллю напругою 220 В та частотою 50 Гц. Приміщення лабораторії є приміщенням з підвищеною небезпекою ураження електричним струмом, то згідно з ГОСТ 12.1.030-81 з метою захисту від ураження електричним струмом використовуємо занулення. Відповідно до ГОСТ 12.2.007.0-75 це відповідає класу 1 за способом захисту від ураження електричним струмом.

Занулення – навмисне електричне з'єднання металевих не струмопровідних частин електроустаткування, яке може опинитися під напругою, з глухо заземленою нейтральною точкою обмотки джерела струму в трифазних мережах, з глухо заземленим виводом обмотки джерела струму в однофазних мережах та з глухо заземленою середньою точкою обмотки джерела енергії в мережах постійного струму.

Принцип дії занулення – перетворення замикання на корпус в однофазне коротке замикання (тобто замикання між фазним та нульовим захисними провідниками) з метою викликати великий струм, здатний забезпечити спрацювання захисту і тим самим автоматично відключити пошкоджене електроустаткування від мережі живлення.

Занулення здійснює дві захисні дії – швидке автоматичне відключення пошкодженого електроустаткування від мережі живлення та зниження напруги занулених металевих не струмопровідних частин, які опинилися під напругою, відносно землі.

Заходи захисту від статичної електрики:

- кілька разів на протязі робочого дня мити руки і обличчя водою, а після закінчення роботи вимити руки й лице з милом;
- щоденно протирати екран монітора, клавіатуру, пристрій "миша", а якщо є приєднаний фільтр то і його антистатичною серветкою;
- щоденно в приміщенні з ПК проводити вологе прибирання;
- установити нейтралізатори статичної електрики;
- підтримувати у приміщенні вологість повітря зазначену в нормативних документах;
- виконати у відповідності з ДНАОП 0.00-1.21-98 "Правила безпечної експлуатації електроустановок споживачів" заземлення ВДТ;
- користувачу ПК бажано носити одяг з природних (льняних) волокон, в т.ч. і шкарпетки

6.4 Пожежна безпека

Вірогідні причини виникнення пожежі при розробці програмного забезпечення «Keyword Binder Bot»:

- несправність електроприладів і електропроводки;
- перегрів апаратури і електромережі;
- розряд статичної електрики;
- порушення ізоляції і коротке замикання електричних ланцюгів;
- блискавка.

У відповідність з ГОСТ 12.1.004-91 пожежну безпеку здійснюють системами: запобігання пожежі, пожежного захисту, організаційно-режимних заходів .

Система запобігання пожежі спрямована на створення умов, що виключають можливість виникнення пожежі і містить:

- контроль і профілактику ізоляції;
- максимальний струмовий захист від коротких замикань;
- охолодження елементів, що нагріваються, і блоків апаратури за допомогою вбудованих вентиляторів;

- для даного класу будівель, приміщення яких відносяться по ПУЕ до класу П-ІІА і місцевості з середньою грозовою діяльністю 10 і більше годин в рік (умови міста Харкова) встановлено ІІІ рівень захисту від блискавки; на будівлі встановлений блискавковідвід, призначений для захисту від прямого удару блискавки;

- виконання, застосування і режим експлуатації електроапаратури відповідають класу приміщення з пожежної небезпеки П-ІІА, згідно (чому) електроапаратура має ступінь захисту ІР-44, електроосвітлювачі – ІР-2Х.

Система пожежного захисту призначена для локалізації пожежі і включає:

- аварійне відключення апаратури і комунікацій;
- сповіщення про пожежу здійснюється по телефону, крім того в приміщенні встановлена димова пожежна сигналізація;
- приміщення лабораторії оснащене вуглекислотними вогнегасниками ОУ-2 із розрахунку 2 вогнегасники на кожні 20 м площі;
- ступінь вогнестійкості будівельних конструкції ІІ вибрано з урахуванням категорії приміщення з пожежної небезпеки В і поверховості будівлі – семиповерхове;
- для успішної евакуації людей, на випадок пожежі, відстань до найближчого евакуаційного виходу не більше 40 метрів, розміри дверей і ширина коридору відповідають протипожежним нормам проектування виробничих будівель.

Відповідно до ГОСТ 12.1.004 – 91 пожежну безпеку здійснюють системами: запобігання пожежі, пожежного захисту, організаційно-режимних заходів .

Згідно НАПБ Б.03.002 – 2007 встановити категорію вибухо-, пожежо-небезпечності приміщення, визначити категорію вогнестійкості будівлі розробити заходи щодо забезпечення пожежної безпеки .

6.5 Охорона навколишнього середовища

Природні небезпечні процеси в більшості випадків виникають через діяльність людини. Відбувається деградація навколишнього природного середовища за рахунок вирубки лісу, викидів в атмосферу, забруднення водного середовища та інше. Все це відбувається через бажання підвищити рівень розвитку промисловості.

Згідно Закону України «Про охорону навколишнього природного середовища», завданням законодавства про охорону навколишнього природного середовища (ОНПС) є регулювання відносин в області охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, попередження і ліквідації негативної дії господарської і іншої діяльності на навколишнє природне середовище (НПС). Основними принципами ОНПС є: попереджувачий характер заходів щодо ОНПС, гарантування екологічно безпечного середовища для життя і здоров'я людей та збереження просторової і видової різноманітності і цілісності природних об'єктів і комплексів.

ВИСНОВКИ

У сучасному світі люди найбільше часу проводять у месенджерах та чат боти дозволяють автоматизувати або прискорити деякі дії без необхідності встановлювати додаткове програмне забезпечення.

При розробці дипломного проекту було вирішено розділити пояснювальну записку на наступні розділи.

Перший розділ розкрив загальні питання, а саме огляд предметної області, постановку задачі, огляд засобів розробки програмного продукту та модель життєвого циклу проекту.

В другому розділі описано алгоритмічне забезпечення у вигляді діаграм.

Третій розділ розкрив питання тестування програмного продукту.

У четвертому розділі відбувся опис керівництва користувача та опис системних вимог.

У п'ятому розділі розповідається про техніко-економічне обґрунтування проекту.

У шостому розділі описані норми охорони праці щодо розробки ПП.

Результатом розробки дипломного проекту став чат бот, який може зберігати шаблони повідомлень користувача та відтворювати їх у будь-якому чаті.

ПЕРЕЛІК ДЖЕРЕЛ ІНФОРМАЦІЇ

- 1 SendPulse // <https://sendpulse.ua/features/chatbot/telegram> 08.06.2021
- 2 Технічне завдання як запорука успішної роботи над веб-проектом // <https://internetdevels.ua/blog/technical-specification-project-management> 08.06.2021
- 3 Inline Bots // <https://core.telegram.org/bots/inline> 08.06.2021
- 4 PyCharm // <https://www.jetbrains.com/help/pycharm/quick-start-guide.html> 08.06.2021
- 5 MongoDB Compass // <https://www.mongodb.com/products/compass> 08.06.2021
- 6 Story Mapping // <https://www.productplan.com/glossary/story-mapping> 08.06.2021
- 7 Діаграми варіантів використання // <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-5-diahramy-variantiv-vykorystannja.pdf> 08.06.2021
- 8 Диаграмма развертывания // <http://khpi-iip.mipk.kharkiv.edu/library/case/leon/g111/g111.html> 08.06.2021
- 9 Диаграмма компонентов // <http://khpi-iip.mipk.kharkiv.edu/library/case/leon/g110/g110.html> 08.06.2021
- 10 Работа с базой данных // <https://metanit.com/nosql/mongodb/2.1.php> 08.06.2021

ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ

main.py

```

import logging
from aiogram import Bot, Dispatcher, executor, types
from aiogram.types import BotCommand
from aiogram.types.message import ContentType
from aiogram.dispatcher import filters
from aiogram.contrib.middlewares.i18n import I18nMiddleware
from bot_types import KeywordNotFoundError, DocumentNotFoundError, ACTION
from config import TOKEN, I18N_DOMAIN, LOCALES_DIR
from db import remove_keyword, insert_keyword, get_user_keywords, get_bind,
remove_all_keywords, check_admin
from locales import TEXT
from bot_utils import \
    input_media, \
    create_query_results, \
    create_keywords_keyboard, \
    unpack_keyword_query_data, \
    message2bind_record_with_keys

logging.basicConfig(level=logging.INFO)

bot: Bot = Bot(token=TOKEN, parse_mode=types.ParseMode.HTML)
dp: Dispatcher = Dispatcher(bot)

i18n = I18nMiddleware(I18N_DOMAIN, LOCALES_DIR)
dp.middleware.setup(i18n)

t = i18n.gettext

async def send_input_media_message(user_id: (int, str), text: str, file: (dict, None))
-> types.Message:
    f_type = file['type']
    file_id = file['file_id']
    params = {'chat_id': user_id, 'caption': text}
    if f_type == ContentType.PHOTO:
        return await bot.send_photo(photo=file_id, **params)
    if f_type == ContentType.VIDEO:
        return await bot.send_video(video=file_id, **params)
    if f_type == ContentType.AUDIO:
        return await bot.send_audio(audio=file_id, **params)
    if f_type == ContentType.DOCUMENT:
        return await bot.send_document(document=file_id, **params)
    if f_type == ContentType.ANIMATION:
        return await bot.send_animation(animation=file_id, **params)

async def send_keyword_answer(user_id: (int, str), key: str, query_id: str = None):
    try:
        bind = get_bind(user_id, key)
    except DocumentNotFoundError:
        if query_id:
            await bot.answer_callback_query(query_id, t(TEXT.BIND.NOT_FOUND))
        return
    f_len = len(bind['files'])
    text = bind['text']
    if f_len > 1:
        media = types.MediaGroup()

```

```

        first = True
        for file in bind['files'][:10]:
            media.attach(input_media(file, text if first else ''))
            first = False
        await bot.send_media_group(chat_id=user_id, media=media)
    elif f_len > 0:
        await send_input_media_message(user_id, text, bind['files'][0])
    else:
        await bot.send_message(chat_id=user_id, text=text)

@dp.callback_query_handler()
async def callback_query_answer(query: types.CallbackQuery):
    action, user_id, meta = unpack_keyword_query_data(query)
    if action == ACTION.SHOW:
        await send_keyword_answer(user_id, meta[0], query.id)
    if action == ACTION.DELETE:
        remove_keyword(user_id, meta[0])
    await bot.answer_callback_query(query.id)

@dp.inline_handler()
async def inline_answers(inline_query: types.InlineQuery):
    try:
        (key, *args) = inline_query.query.split()
        key = key.lower()
    except ValueError:
        return

    query_id = inline_query.id

    try:
        bind = get_bind(inline_query['from'].id, key)
    except DocumentNotFoundError:
        await bot.answer_inline_query(
            query_id,
            results=[],
            switch_pm_text=t(TEXT.BUTTON.BIND_KEYWORD),
            switch_pm_parameter=key
        )
        return

    await bot.answer_inline_query(
        query_id,
        results=create_query_results(query_id, key, bind, args),
        is_personal=True,
        cache_time=1
    )

@dp.message_handler(commands=['help'])
async def on_help(message: types.Message):
    await bot.send_message(chat_id=message.chat.id, text=t(TEXT.COMMAND.HELP))

@dp.message_handler(commands=['start'])
async def on_start(message: types.Message):
    await bot.send_message(chat_id=message.chat.id, text=t(TEXT.COMMAND.START))

@dp.message_handler(commands=['list'])

```

```

async def on_list(message: types.Message):
    await bot.send_chat_action(message.chat.id, 'typing')
    try:
        user_id = message.from_user.id
        keys = get_user_keywords(user_id)
        if not len(keys):
            raise DocumentNotFoundError
        keyboard = create_keywords_keyboard(user_id, keys)
        await bot.send_message(
            chat_id=message.chat.id,
            text=t(TEXT.COMMAND.LIST),
            reply_markup=keyboard
        )
    except DocumentNotFoundError:
        await bot.send_message(chat_id=message.chat.id,
            text=t(TEXT.COMMAND.LIST_EMPTY))

@dp.message_handler(filters.Command('bind', ignore_caption=False),
    content_types=types.ContentTypes.ANY)
async def on_bind(message: types.Message):
    try:
        (bind, keys) = message2bind_record_with_keys(message, await bot.get_updates())
        insert_keyword(message.from_user, bind, keys)
    except (ValueError, IndexError):
        await bot.send_message(chat_id=message.chat.id, text=t(TEXT.BIND.ERROR))
    else:
        await send_keyword_answer(message.from_user.id, keys[0])

@dp.message_handler(commands=['unbind'])
async def on_unbind(message: types.Message):
    try:
        divided = message.text.split(maxsplit=2)[1:3]
        keys = divided[0].split(',')
        if len(divided) > 1 and check_admin(message.from_user.id):
            user_id = int(divided[1])
        else:
            user_id = message.from_user.id
        if '*' in keys:
            remove_all_keywords(user_id)
            await bot.send_message(chat_id=message.chat.id,
                text=t(TEXT.KEYWORD.DELETE.SUCCESS))
        else:
            for key in keys:
                try:
                    remove_keyword(user_id, key, True)
                except (KeywordNotFoundError, DocumentNotFoundError):
                    await bot.send_message(chat_id=message.chat.id,
                        text=t(TEXT.KEYWORD.DELETE.ERROR) + f': {key}')
                else:
                    await bot.send_message(chat_id=message.chat.id,
                        text=t(TEXT.KEYWORD.DELETE.SUCCESS) + f': {key}')
            except (IndexError, ValueError):
                await bot.send_message(chat_id=message.chat.id,
                    text=t(TEXT.KEYWORD.DELETE.ERROR))

async def on_startup(dispatch: Dispatcher):
    await bot.set_my_commands([
        BotCommand('bind', 'keyword your value - bind new keyword'),

```

```

        BotCommand('list', '- show all keywords'),
        BotCommand('help', '- additional information'),
        BotCommand('unbind', 'keyword - delete bound keyword'),
    ])

```

```

if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True, on_startup=on_startup)

```

bot_types.py

```

from typing import TypedDict, List, Dict, Union
from aiogram.types.message import ContentType
from pymongo.collection import ObjectId

```

```

class BindFile(TypedDict):
    file_id: str
    type: ContentType

```

```

class BindRecord(TypedDict):
    text: str
    raw_text: str
    files: List[BindFile]

```

```

class UserRecord(TypedDict):
    _id: ObjectId
    user_id: Union[int, str]
    full_name: str
    username: str
    keywords: Dict[str, str]

```

```

class ACTION:
    SHOW = 'show'
    DELETE = 'delete'

```

```

class DocumentNotFoundError(Exception):
    """Document not exist in collection"""
    pass

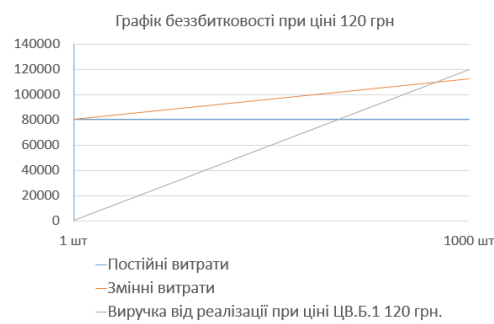
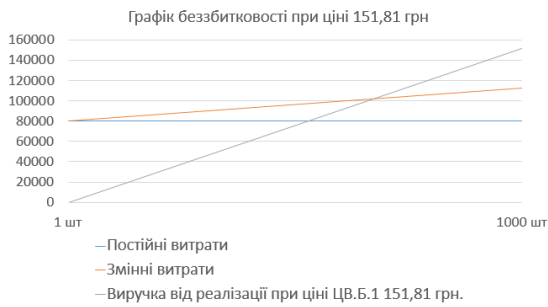
```

```

class KeywordNotFoundError(Exception):
    """Keyword not found"""
    pass

```

ДОДАТОК Б ЕКОНОМІЧНІ РОЗРАХУНКИ



- Висновок**
- Трудомісткість проекту 357 годин.
 - Кількість виконавців 1 особа.
 - Кошторисна вартість розробки при замовленні 1 шт. 80325 грн.
 - Кошторисна вартість розробки при замовленні 1000 шт. 112453 грн.
 - Ціна програмного продукту(без ПДВ) 108438,75 грн. при замовленні 1 шт.
 - Ціна програмного продукту(без ПДВ) 151,81 грн. при замовленні 1000 шт.
 - Точка беззбитковості 671 шт. при ціні 151,81 грн.
 - Точка беззбитковості 914 шт. при ціні 120 грн

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
"ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітня програма: "Розробка програмного забезпечення"

ІЛЮСТРАТИВНІ МАТЕРІАЛИ
ДО ЗАХИСТУ ДИПЛОМНОГО ПРОЄКТУ

Тема: **"Keyword Binder Bot"**

Виконавець

Володимир ЮШКО

Керівник ДП

Росіца МАНЄВА

Харків 2021 р.

Keyword Binder Bot

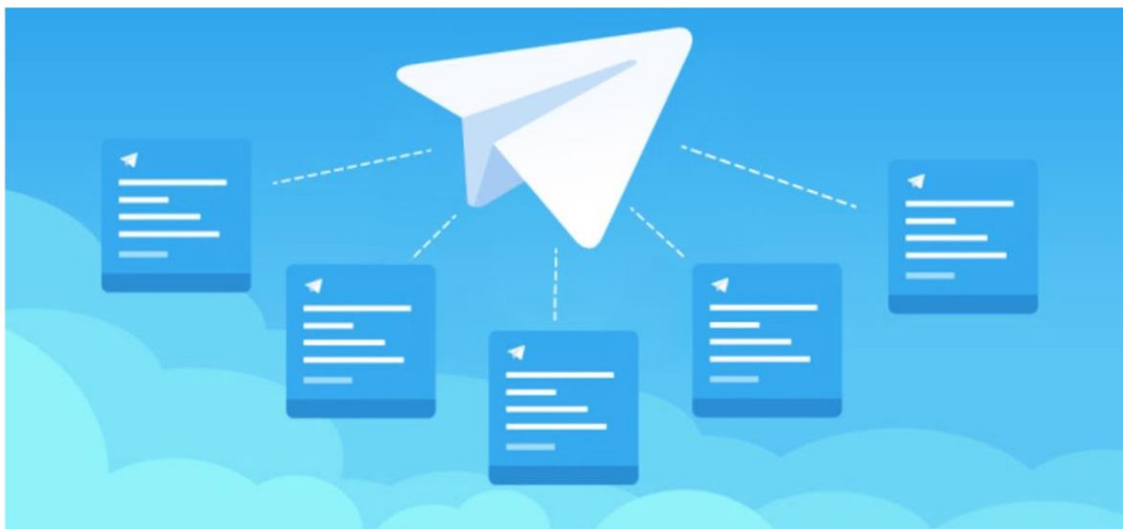
Студент: Юшко В.Р.

Керівник: Манєва Р.І.



Слайд 1/13

Актуальність



Слайд 2/13

Авдиторія

	App ⓘ	Publisher ⓘ
1	 Telegram	Telegram FZ-LLC
2	 Getcontact	Getverify LDA
3	 WhatsApp Messenger	WhatsApp LLC
4	 My Kyivstar	Kyivstar
5	 Viber Messenger - Free Video Calls & Group Chats	Viber Media S.à r.l.



Рейтинг додатків Google Play у розділі «комунікація» в Україні

Ріст кількості користувачів Telegram

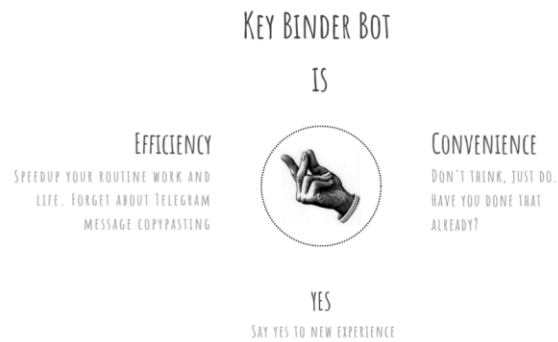
Слайд 3/13

Огляд аналогів



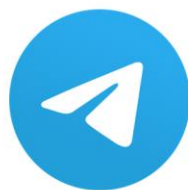
Слайд 4/13

Огляд аналогів



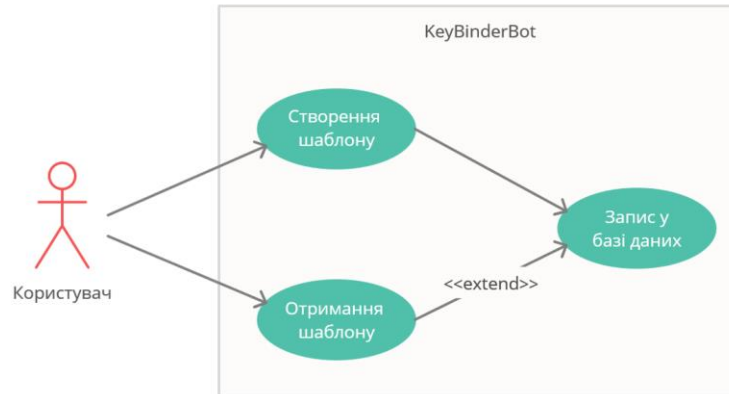
Слайд 5/13

Засоби розробки



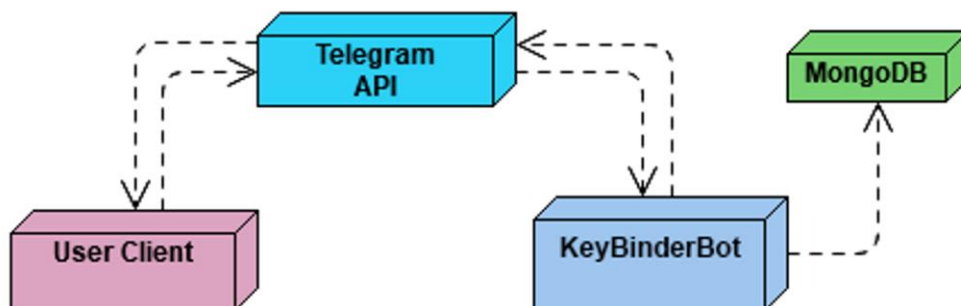
Слайд 6/13

Діаграма варіантів використання



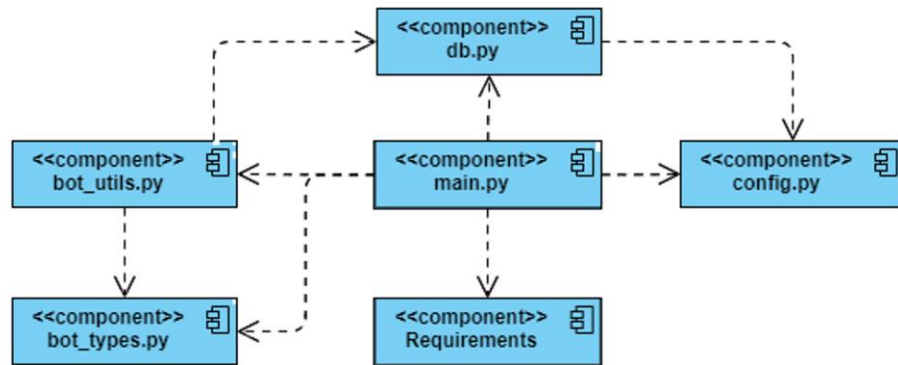
Слайд 7/13

Діаграма розгортання



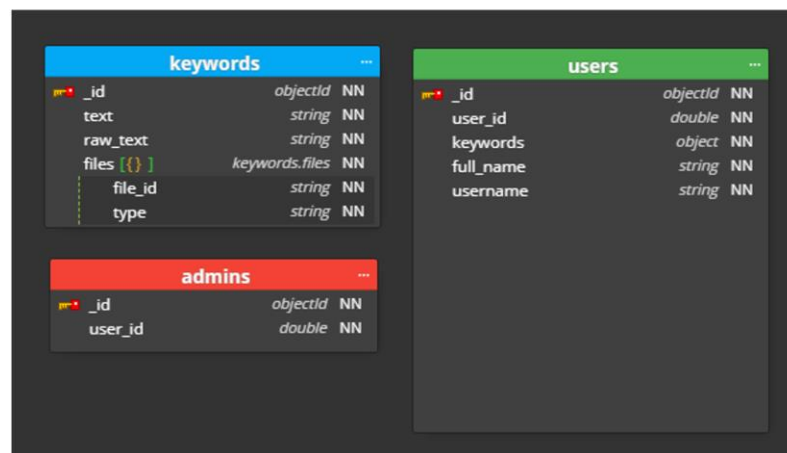
Слайд 8/13

Діаграма компонентів



Слайд 9/13

Опис бази даних



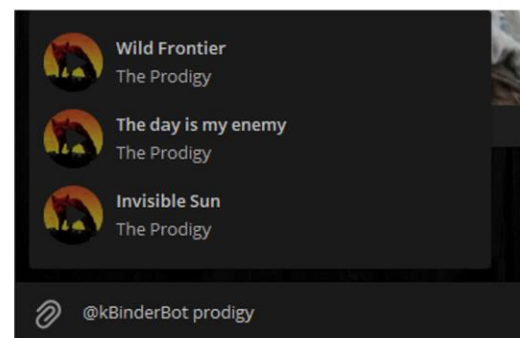
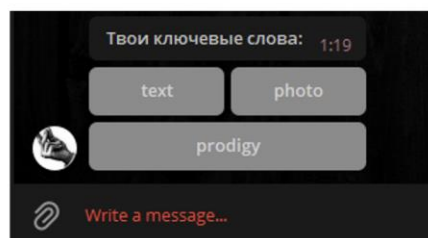
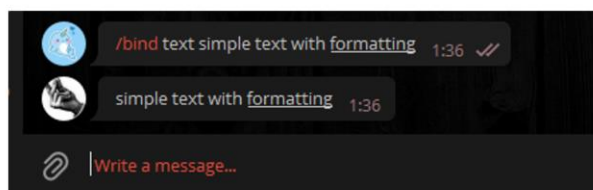
Слайд 10/13

Вимоги до ПЗ


Назва тест-кейса	Кроки для відтворення	Очікуваний результат	Реальний результат
Створення текстового шаблону	Відправити боту повідомлення з текстом «/bind text any simple text»	Бот відправить таке саме повідомлення, але без команди /bind на початку та ключового слова, що стоїть за ним	TRUE
Відтворення шаблону з текстом	У полі для введення повідомлення ввести «@kBinderBot text»	Над полем для вводу повідомлення повинен відобразитися шаблон, при натисканні на який бот відправить в чат повідомлення з підготованим текстом	TRUE

Слайд 11/13

Робота боту



Слайд 12/13



ПЕРЕЙДЕМО ДО ДЕМОНСТРАЦІЇ
ПРОГРАМНОГО ПРОДУКТУ

Слайд 13/13