

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення




ДИПЛОМНИЙ ПРОЕКТ

Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота

Назва теми


Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр ДППЗ.180119.01.13.ВД

Виконала студентка IV курсу група ІПЗ-18-1  Ю. В. Собкова
Підпис Ініціали, прізвище
Керівник старший викладач  Г. І. Бедратюк
Науковий ступінь, звання Ініціали, прізвище
Нормоконтролер канд. техн. наук, доцент  Н. І. Праворська
Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення

 Л. П. Бедратюк
Підпис Ініціали, прізвище

6 червня 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри і пз
Л. П. Бедратюк
01 03 2022 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ)**

Собковій Юлії Василівні
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота

Керівник проекту (роботи) Бедратюк Ганна Іванівна
старший викладач
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18

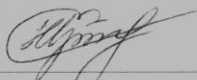
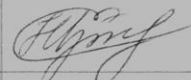

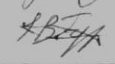
2. Строк подання студентом проекту (роботи) на кафедру 01.06.2022 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Презентаційні матеріали (слайди 12 шт.)

6. Консультанти розділів дипломного проєкту (роботи)

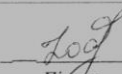
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|--|---|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Праворська Н. І. доцент кафедри ІПЗ |  |  |
| Антиплагиат | Гурман І. В. канд. техн. наук, доцент. |  |  |

7. Дата видачі завдання « 01 » березня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

| Назва етапів (розділів) дипломного проєкту (роботи) | Строк виконання етапів проєкту (роботи) | Примітка |
|--|---|----------|
| 1 Ознайомлення з тематикою дипломного проєктування (ДП), визначення та узгодження індивідуальних тем ДП | 01.12 – 30.12.2021 | |
| 2 Дослідження предметної області, в якій планується використання програмного засобу (ПЗ), визначення задач та вимог розробки технічного завдання | 02.01 – 31.01.2022 | |
| 3 Проєктування програмного забезпечення | 01.02 – 28.02.2022 | |
| 4 Програмна реалізація | 01.03 – 10.04.2022 | |
| 5 Тестування програмного забезпечення | 11.04 – 30.04.2022 | |
| 6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки ДП згідно вимог стандартів | 01.05 – 25.05.2022 | |
| 7 Попередній захист ДП | Травень 2022 (згідно графіка) | |
| 8 Перевірка ДП на плагіат, нормоконтроль, отримання відгуків та рецензій. Брошування (зшиття) пояснювальної записки. | 26.05 – 30.05.2022 | |
| 9 Підготовка до захисту ДП | з 01.06.2022 | |

Студент


Підпис

Собкова Ю. В.

Ініціали, прізвище

Керівник проєкту (роботи)


Підпис

Бедратюк Г. І.

Ініціали, прізвище

ВІДОМІСТЬ ДОКУМЕНТІВ

| № рядка | Формат | Позначення документа | Найменування документа | К-сть аркушів | № екз. | Примітка |
|---------|--------|----------------------|------------------------------|---------------|--------|----------|
| | | | <u>Текстові документи</u> | | | |
| 1 | A4 | ДППЗ.180119.01.13.ПЗ | Пояснювальна записка | 80 | | |
| 2 | A4 | | Завдання на дипломний проєкт | 1 | | |
| 3 | A4 | | Анотація | 1 | | |
| | | | <u>Графічні документи</u> | | | |
| 4 | A4 | | Презентаційні матеріали | 12 | | |

| | | | | | | | |
|-----------|------|------------------|-----------------|------------------------------|---------------|------|---------|
| | | | | ЗПППЗ.180119.01.13.00 | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | Літ. | Арк. | Аркушів |
| Виконав | | Собкова Ю. В. | <i>[Підпис]</i> | 6.06 | | | |
| Керівник | | Бедратюк Г. І. | <i>[Підпис]</i> | 6.06 | | 1 | 1 |
| Н. Контр | | Праворська Н. І. | <i>[Підпис]</i> | 6.06 | ХНУ, ІПЗ-18-1 | | |
| Затвердив | | Бедратюк Л.П. | <i>[Підпис]</i> | 6.06 | | | |

Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота
Відомість документів

АНОТАЦІЯ

Тема дипломного проекту: «Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота».

Автор проекту: Собкова Юлія Василівна.

Керівник проекту: Бедратюк Ганна Іванівна.

Пояснювальна записка: 80 с., 21 рис., 1 табл., 4 дод., 21 джерел.

Графічна частина: 12 презентаційних слайдів.

Ключові слова: HTTP-сервер, об'єктно-орієнтоване програмування, Backend, Frontend, Python-фреймворк, Botfather, Реляційна система управління базами даних, діаграма класів, тестування програмного забезпечення, тестування UI/UX

Робот telegram може спростити розмову між користувачем та машиною та дозволити користувачам та машинам безперешкодно виконувати завдання. Він допомагає підприємствам залучити користувача до розмови на досить тривалий час, збирає необхідні дані та багато іншого.

Сьогодні у нас є розумні чат-боти зі штучним інтелектом, які використовують обробку природної мови (NLP) для розуміння людських команд (текстових та голосових) та навчаються на власному досвіді. Чат-боти стали основним інструментом взаємодії з клієнтами для компаній та брендів, які активно присутні в Інтернеті (веб-сайт та платформи соціальних мереж).

Telegram надає API для створення ботів для соціальних взаємодій, продуктивності, ігор та електронної комерції.

06.06.2022
Дата


Підпис

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ..... | 6 |
| ВСТУП..... | 7 |
| 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ | 8 |
| 1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей | 8 |
| 1.2 Аналіз наявного програмного-технічного забезпечення предметної області | 11 |
| 1.3 Визначення вимог до розробки інтернет-платформи | 14 |
| 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 19 |
| 2.1 Опис реалізації | 19 |
| 2.2 Моделювання процесу роботи чат-боту..... | 19 |
| 2.3 Архітектура проектування та функціональна структура | 21 |
| 2.4 Модель аналізу програмного забезпечення | 26 |
| 2.5 Вибір реляційної системи керування БД | 28 |
| 2.6 Проектування моделі бази даних | 31 |
| 3 ПРОГРАМНА РЕАЛІЗАЦІЯ | 32 |
| 3.1 Реалізація створення чат-боту | 32 |
| 3.2 Реалізація логіки взаємодії з додатком..... | 37 |
| 3.3. Структура проекту | 39 |
| 4 ТЕСТУВАННЯ ТЕЛЕГРАМ БОТУ | 41 |
| 4.1 Аналіз методів тестування..... | 41 |
| 4.2 Тестування чат-боту | 47 |
| 4.3 Аналіз результатів тестування..... | 52 |
| ВИСНОВОК | 53 |

| | | | | | | | | |
|-----------|------|------------------|-----------------|------|---|---------------|------|---------|
| | | | | | ЗППІПЗ.180119.01.13.ВД | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота | Літ. | Арк. | Аркушів |
| Виконав | | Собкова Ю. В. | <i>[Підпис]</i> | 6.06 | | | 4 | 80 |
| Керівник | | Бедратюк Г. І. | <i>[Підпис]</i> | 6.06 | Відомість документів | ХНУ, ІПЗ-18-1 | | |
| Н. Контр | | Праворська Н. І. | <i>[Підпис]</i> | 6.06 | | | | |
| Затвердив | | Бедратюк Л.П. | <i>[Підпис]</i> | 6.06 | | | | |

| | |
|--------------------------------|----|
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 54 |
| ДОДАТОК А | 57 |
| ДОДАТОК Б..... | 59 |
| ДОДАТОК В | 65 |
| ДОДАТОК Г..... | 74 |

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 5 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

ПЕРЕЛІК СКОРОЧЕНЬ

CSS – customer support services

SQL – Structured Query Language

iOS – Операційна система

RDBMS – Реляційна система керування базами даних

API – Прикладний програмний інтерфейс

ЗЗ – Знаки зодіаку

OPC – Комунікаційна платформа маршрутизатора

PHP – Personal Home Page

MVC – Model-View-Controller

UML – Unified Modelling Language

DB – Database

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

ВСТУП

Бот Telegram - це стороння програма, яка працює всередині Telegram. Робот в Telegram може взаємодіяти з користувачами, надсилаючи повідомлення. Telegram-бот - це особливий тип користувача, який є не людиною, а комп'ютерною програмою, здатною обслуговувати компанії або бренди за допомогою багатьох функцій, таких як розсилання інформації, нагадування, відтворення мелодій, оформлення замовлень та багато іншого.

Користувачі можуть взаємодіяти з ботом Telegram, надсилаючи повідомлення. З усіх чотирьох типів аналогів бот Telegram має безліч корисних для бізнесу функцій. Telegram надає API для створення ботів для соціальних взаємодій, продуктивності, ігор та електронної комерції. Крім того, боти Telegram можуть надавати підтримку клієнтам або збирати інформацію, підключаючись до CRM, системи продажу квитків або платформи обміну повідомленнями.

Вони можуть робити будь-що - вчити, грати, шукати, транслювати, нагадувати, підключатися, інтегруватися з іншими сервісами або навіть передавати команди в Інтернет.

Чат-боти стали одними з кращих інноваційних технологій цього десятиліття. Чат-боти стали важливою частиною сучасного бізнесу. Вони використовуються для маркетингу, продажу та підтримки клієнтів.

Безперечно, якщо у вашому онлайн-бізнесі немає боту, ви пропустите безліч переваг, які він дає. Ось деякі з основних переваг ботів:

- Безперешкодне живе спілкування
- Працює 24/7
- Економія часу та грошей
- Пропозиція кращого досвіду користувача
- Можуть забезпечити персоналізований досвід для клієнтів

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

І ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Що таке Telegram і чим він відрізняється від інших програм для обміну повідомленнями?

Програми для обміну повідомленнями стали нашим улюбленим способом спілкування, даючи нам можливість спілкуватися з друзями у будь-який час та в будь-якому місці. І хоча Facebook Messenger і WhatsApp є найвідомішими з них, нещодавно з'явилася програма, яка порушила ринок, заявивши, що є найбезпечнішим з них. Ця програма називається Telegram Messenger.

Telegram - це онлайн-додаток для обміну повідомленнями, який працює так само, як популярні програми для обміну повідомленнями WhatsApp та Facebook Messenger. Це означає, що ви можете використовувати його для надсилання повідомлень своїм друзям під час підключення до Wi-Fi або мобільної мережі. Telegram працює на основі хмарних технологій і стверджує, що для нього пріоритетні безпека та швидкість, що робить його гарною альтернативою іншим популярним програмам для обміну повідомленнями. Сервіс був запущений у 2013 році, і з того часу кількість його активних щомісячних користувачів досягла 200 мільйонів.

Як і WhatsApp, Telegram також має можливість показувати статус друга в мережі, прикріплювати та обмінюватися фотографіями, відео, розташуванням, контактами та документами.

Особливістю Telegram є безпека. Він стверджує, що всі його дії, включаючи чати, групи та медіа, якими обмінюються учасники, зашифровані. Це означає, що їх не буде видно без попереднього розшифрування. Програма також дозволяє встановлювати таймери самознищення повідомлень та медіафайлів, якими ви ділитесь, від двох секунд до одного тижня за допомогою вбудованої функції

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| | | | | | | | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |

"Секретний чат". Він також пропонує наскрізне шифрування, не залишаючи слідів на серверах Telegram.

Також є можливість перевірити безпеку ваших "секретних чатів" за допомогою зображення, яке є ключем шифрування. Порівнюючи свій ключ шифрування з ключем друга, ви можете переконатися, що ваша розмова безпечна і менш уразлива для атак типу "людина посередині".

Telegram можна використовувати та встановлювати так само, як і інші програми обміну повідомленнями. Ви можете завантажити його з App Store від Apple або Play Store від Google - шукайте логотип у вигляді паперового літачка. Після прогортання екрана привітання вам буде запропоновано ввести номер телефону, а потім додати своє ім'я та фотографію. Наступний крок – знайти друзів та почати спілкування.

Програму можна використовувати на смартфонах, планшетах, ноутбуках та настільних комп'ютерах. Telegram доступний для Android, iOS, Windows Phone, Windows NT, MacOS і Linux.

Що потрібно знати про конфіденційність та безпеку Telegram? Фірмовою особливістю Telegram є наскрізне шифрування, але важливо знати, що не всі повідомлення Telegram мають такий рівень конфіденційності. У більшості повідомлень використовується шифрування від клієнта до сервера, яке є менш безпечним, ніж наскрізне, але дозволяє отримати доступ до чатів Telegram з інших пристроїв, у тому числі з Інтернету.

Конфіденційність і безпека Telegram вважаються частково тому, що API сервісу має відкритий вихідний код і доступний для оцінки та інтеграції будь-яким розробником.

Станом на січень 2021 року Telegram оголосив, що має 500 мільйонів MAU. У 2020 році, маючи понад 500 мільйонів завантажень тільки в Google Play Store, Telegram швидко наздоганяє своїх конкурентів, WhatsApp (5 мільярдів) та Facebook (2 мільярди).

Враховуючи швидке зростання компанії та заяви про конфіденційність та безпеку, Telegram став важливою платформою, на яку слід звернути увагу бізнесу.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 9 |

Все більше підприємств використовують платформи соціальних мереж для надання послуг із підтримки клієнтів (CSS). Зі зростанням бази користувачів Telegram став важливою платформою, на яку слід звернути особливу увагу.

Telegram зробив так, що якщо новий контакт підписується на бізнес-канал, компанія не отримує доступу до контактних чи особистих даних користувачів.

У той же час користувачі Telegram можуть шукати назви компаній та підписуватись на них через область винаходів. Така схема працює таким чином, що клієнти можуть зв'язатися з компаніями, якщо захотять, але компанії не можуть зв'язатися з клієнтами без їхнього дозволу.

В останній версії Telegram існує чотири типи послуг:

1. Профіль користувача - це персональний обліковий запис, який зазвичай реєструється на номер телефону користувача. Такий тип користувача використовується для обміну особистими повідомленнями, але іноді може виступати як адміністратор каналів і груп.

2. Група Telegram – це чат-група користувачів та ботів, які називаються її учасниками. У групі Telegram може бути не більше 200 000 учасників, і всі учасники мають право публікувати повідомлення у групі. Адміністратор групи має особливі привілеї та призначає права власності для своїх членів, щоб дозволити повідомлення, пов'язані з відео та зображеннями.

3. Канал Telegram - це підмножина груп Telegram, це спеціальна група, яка може мати необмежену кількість передплатників, але адміністратори і боти мають право публікувати повідомлення в каналі, всі інші користувачі каналу не можуть публікувати повідомлення.

4. Telegram-бот - це особливий тип користувача, який є не людиною, а комп'ютерною програмою, здатною обслуговувати компанії або бренди за допомогою багатьох функцій, таких як розсилання інформації, нагадування, відтворення мелодій, оформлення замовлень та багато іншого. Бот може публікувати повідомлення у групі або каналі.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 10 |

1.2 Аналіз наявного програмного-технічного забезпечення предметної області

Боти Telegram - це кодовані програми на бекенді для автоматизації, з якими будь-який користувач може взаємодіяти і виконувати роботу. Наприклад, є бот Telegram, який може принести вам останню важливу інформацію з усього світу, коли вам це потрібно.

Нижче наведено список кращих ботів Telegram в 2022 році

- **OpenMember.** Вів дозволяє швидко та легко приймати членські платежі для ваших Telegram-каналів або груп. Бот OpenMember та веб-інструмент приладової панелі дозволяють стягувати будь-яку плату за доступ з будь-якою періодичністю (щодня, щотижня, щомісяця, щорічно або одноразово). Почати роботу дуже просто: зайдіть на сайт OpenMember, створіть обліковий запис і пройдіть швидко процедуру реєстрації.(Рисунок 1.1)

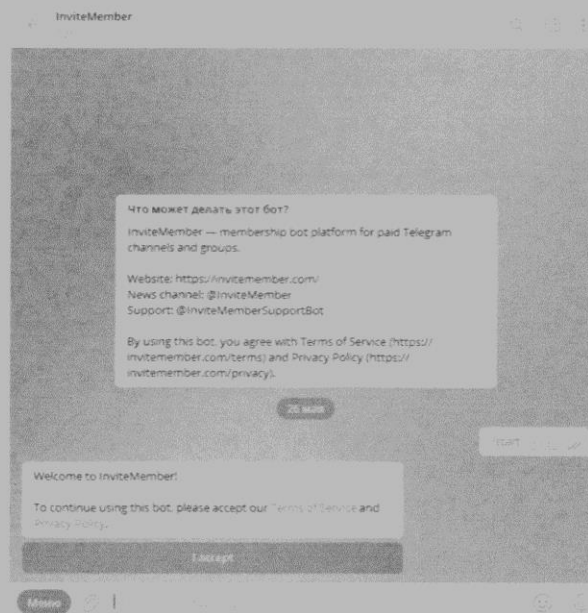


Рисунок 1.1 – чат з ботом InviteMemberBot

| | | | | | | | |
|------|------|----------|--------|------|--|----------------------|------|
| | | | | | | ДППЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 11 |

- GameBot. Ви можете знайти цього бота, ввівши @gamebot у пошуку Telegram. Це дуже простий Telegram-бот, який дозволяє грати з друзями у чат-ігри. Просто запустіть бота, як ми вже пояснювали, та натисніть на "Грати з друзями". Він попросить вас вибрати контакт, з яким хочете грати. (Рисунок 1.2)

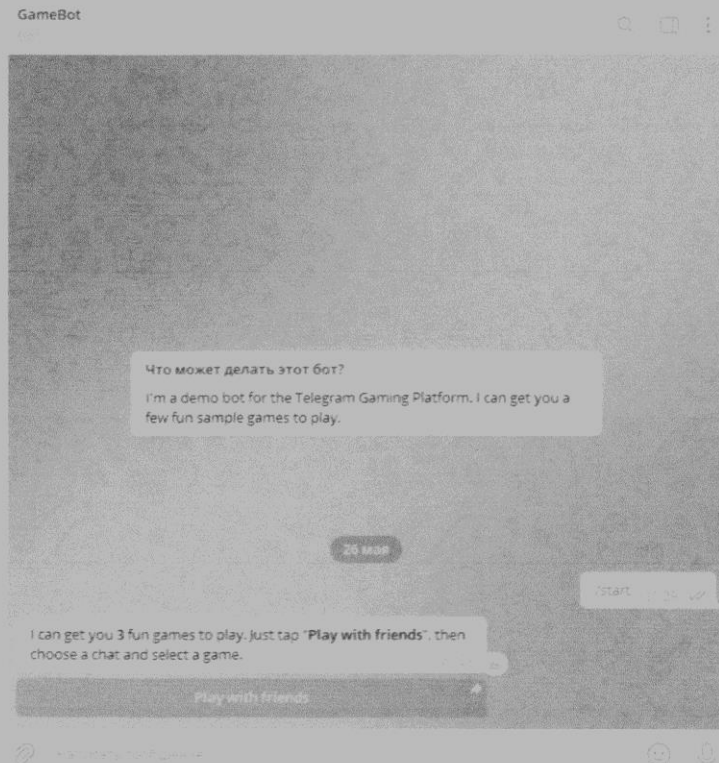


Рисунок 1.2 – чат з ботом GameBot

- Get Media Bot. Вважається, що це дивовижний бот для Telegram. Get Media Bot допоможе вам отримати будь-які медіа файли з усього Інтернету. Вам просто потрібно ввести запит і все готово. Ось список послуг, які він надає: Скачати музику, скачати відео, пошук слів з пісні, завантажити Instagram Story, пряме завантаження медіа.

Функція «Пряме завантаження» може бути використана для завантаження

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 12 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

медіа із Instagram, Twitter, YouTube, Facebook та Soundcloud. Спробувати варто.

- Fortune Horoscope. Бажаєте поринути у майбутнє? Виберіть свій знак зодіаку і нехай гороскоп Фортуни зробить все інше! На рисунку 1.3 зображений потрібний нам бот, тепер можна починати з ним спілкування.

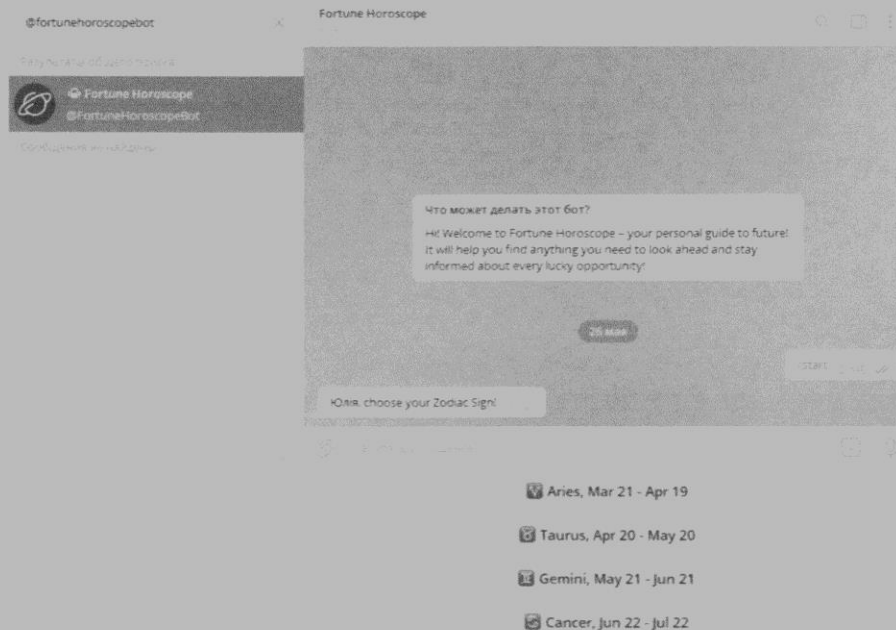


Рисунок 1.3 – Чат з ботом Fortune Horoscope

До бота можна звертатись командами які представлені у вигляді кнопок. Обираємо свій знак зодіаку, якщо вам невідомо хто ви по ЗЗ, то бот допомагає маленькими підказками з періодами днів, саме по яких можна визначити ваш ЗЗ. З кожною вашою відповіддю бот пропонує нові запитання фіксуючи попередні. І по завершенню так званих налаштувань бот регулярно буде надсилати вам віщання майбутнього о тій годині, котру ви обрали для вашої зручності. На рисунку 1.4 зображений чат з початковими налаштуваннями сповіщень бота, які в майбутньому будуть регулярно надсилатись.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 13 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

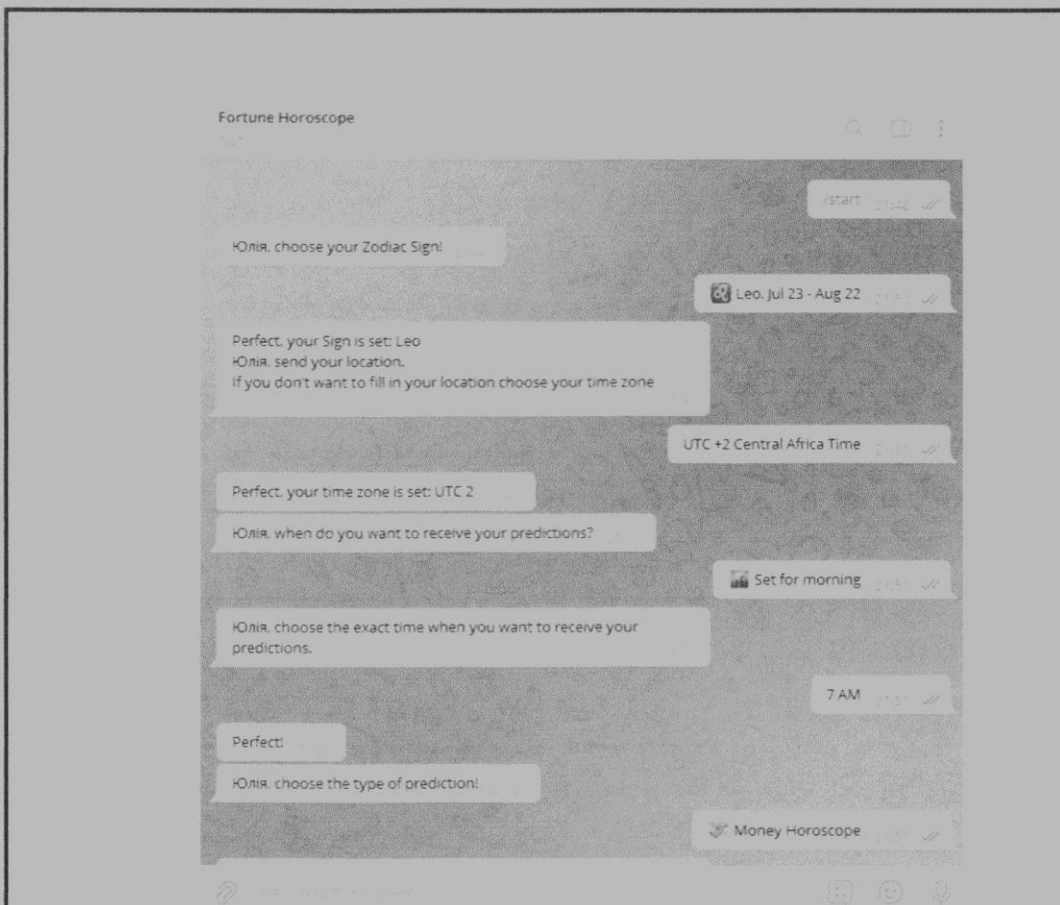


Рисунок 1.4 – початкові налаштування бота Fortune Horoscope

1.3 Визначення вимог до розробки інтернет-платформи

Чат-боти можуть бути створені за допомогою будь-якої мови програмування на ваш вибір. В принципі, можна використовувати всі програми, які працюють на сервері і можуть відповідати на запити через HTTPS. Нижче наведено добірку популярних мов програмування, які часто використовуються для розробки веб-сайтів та програм:

- JavaScript (node.js)

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 14 |

- Java / Kotlin
- PHP
- C#
- Python
- Swift
- Ruby

При виборі мови програмування розробки бота у гру вступають різні моменти: Наскільки «добре» вони чи його команда знають відповідну мову програмування? Які інструменти програмування доступні або до яких бібліотек можна отримати доступ?

Після створення бекенду вам необхідно підключити його до будь-якої з платформ обміну повідомленнями, в нашому випадку до Telegram, щоб ви могли обробляти повідомлення, надіслані користувачами, та надавати відповіді на них.

Ви також можете створити робота без досвіду кодування або програмування. Існує безліч інструментів для створення чат-ботів.

Щоб створити програму чату на Java, необхідно знати програмування сокетів. Наведемо приклад, що складається з інтерфейсу з Swing Applet(Додаток)

Основний принцип у тому, що ми передаємо ехом повідомлення від клієнта, відправлене на сервер, іншому клієнту.

У наведеному прикладі є лише один клієнт та один сервер. Зв'язок відбувається між портами. Ми повинні встановити сокетне з'єднання. Сокет - це канал зв'язку "кінець - кінець". Один клієнт та один сервер мають сокетне з'єднання. Інший клієнт має інше сокетне з'єднання із сервером.

У прикладі початковою частиною є побудова інтерфейсу з допомогою swing. Для перевірки отримання/надсилання повідомлення в реальному часі реалізується потік.

Java - це об'єктно-орієнтована мова програмування загального призначення, заснована на класах, розроблена для того, щоб мати менше залежностей від реалізації. Це обчислювальна платформа для розробки програм. Java є швидкою,

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 15 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

безпечною та надійною мовою. Вона широко використовується для розробки Java-програм в ноутбуках, центрах обробки даних, ігрових консолях, наукових суперкомп'ютерах, стільникових телефонах і т.д.

Java Platform - це набір програм, які допомагають програмістам ефективно розробляти та запускати Java-програми. Вона включає механізм виконання, компілятор і набір бібліотек. Це набір комп'ютерного програмного забезпечення та специфікацій. Джеймс Гослінг розробив платформу Java у компанії Sun Microsystems, а пізніше її придбала корпорація Oracle.

Node.js (Node) - це платформа розробки з відкритим вихідним кодом для виконання JavaScript на стороні сервера. Node корисний для розробки програм, що вимагають постійного з'єднання браузера з сервером, і часто використовується для програм реального часу, таких як чат, стрічки новин та web push-сповіщення.

Node.js призначений для запуску на виділеному HTTP-сервері та для використання одного потоку з одним процесом за раз. Програми Node.js засновані на подіях та виконуються асинхронно. Код, створений на платформі Node, не слід традиційної моделі: отримати, обробити, надіслати, почекати, отримати. Натомість Node обробляє вхідні запити в постійному стеку подій та відправляє невеликі запити один за одним, не чекаючи відповідей.

Це відхід від основних моделей, в яких запускаються більші, складніші процеси і кілька потоків одночасно, причому кожен потік очікує відповідної відповіді, перш ніж рухатися далі.

Однією з головних переваг Node.js, за словами його творця Райана Дала, є те, що він не блокує введення/виведення (I/O). Деякі розробники дуже критично ставляться до Node.js і відзначають, що якщо один процес вимагає значної кількості циклів процесора, програма буде блокуватися, а блокування може призвести до краху програми. Прихильники моделі Node.js стверджують, що процесорний час обробки менше турбує через велику кількість невеликих процесів, на яких базується код Node.

Python - це мова кодування загального призначення, це означає що, на відміну від HTML, CSS та JavaScript, її можна використовувати не тільки для веб-розробки,

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 16 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

але й для інших видів програмування та розробки програмного забезпечення.

Python можна використовувати для таких завдань, як:

- Back end (або серверна сторона) розробка веб-додатків та мобільних додатків;
- Розробка програм та програмного забезпечення для настільних комп'ютерів;
- Обробка великих даних та виконання математичних обчислень;
- Написання системних скриптів (створення інструкцій, що дають команду комп'ютерній системі "зробити" щось).

Не дивлячись на широкий спектр можливостей Python, це легка у вивченні, загребувана мова програмування, яка може експоненційно збільшити ваші шанси на отримання роботи та доходу за лічені місяці.

Botogram - це Python-фреймворк, який дозволяє вам зосередитись тільки на створенні ботів Telegram, не турбуючись про базове API ботів.

У той час як більшість бібліотек для Telegram просто обгортають Bots API, botogram приділяє велику увагу досвіду розробки, прагнучи надати вам найкращий API із можливих. Більшість деталей реалізації Telegram управляється botogram, тому ви можете просто зосередитись на своєму боті.

Botogram доступний в індексі пакетів Python, тому можна легко встановити його за допомогою утиліти командного рядка pip. Перед встановленням переконайтеся, що у вашій системі встановлені Python 3.4, або новіша версія, pip, virtualenv і setuptools. Потім виконайте наступну команду: «\$ python3 -m pip install botogram2», і ваш botogram буде встановлений.

Основна робота Телеграм-бота це партнерство по чату з користувачами, який виконує необхідні функції. Він виконує наперед визначені завдання самостійно та без участі користувача.

Бот Telegram в принципі може робити все те саме, що і партнер по чату - людина. Автоматично або на запит він може надсилати вам наступну інформацію:

- Текстові повідомлення

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 17 |

- Зображення
- Відео
- Файли будь-якого іншого виду

Особливою функцією Telegram-бота є можливість виконувати команди в чаті Telegram, які безпосередньо викликають дії або запитують інформацію. Наприклад, можна надіслати боту команду "/help", яка потім виводить у чаті можливі для цього бота команди у вигляді текстового зворотного зв'язку. Це може бути наступний список команд:

- /status
- /temperature
- /last alarm
- /stop

Якщо розглядати Telegram Bot у контексті джерела даних, фактичні дані зазвичай залишаються на локальних серверах джерела даних. Бот отримує доступ до необхідних наборів даних, а потім готує їх для виведення повідомлень за запитом на кінцевому пристрої. Лише фактична структура програми постійно перебуває у хмарі на серверах Telegram.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 18 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Опис реалізації

Щоб створити Telegram Bot самостійно, достатньо встановити програму Telegram на пристрій та увійти до нього для створення особистого облікового запису. Додаток Telegram для мобільних пристроїв можна знайти в магазинах програм (наприклад, Google Play Store або Apple App Store), а для інших пристроїв на офіційному сайті Telegram.

Botfather – назва є алюзією на мафіозну епопею «Хрещений батько», засновану на романі Маріо Пьюзо і успішно екранізовану в трьох частинах з Марлоном Брандо в ролі «Хрещеного батька» в 1972 по суті є ботом Telegram і відповідає на команди. Він допомагає у створенні власного робота, організує унікальні імена роботів, наприклад, і призначає дозволи доступу для програмування.

За допомогою комунікаційної платформи OPC(Рисунок 2.1), Телеграм бот можна створити лише за кілька клацань миші без складного програмування. За допомогою плагіна OPC Router Telegram Bot Plug-in бот реалізується за допомогою простої графічної конфігурації. Повідомлення від інших плагінів (SAP, OPC, MQTT і т.д.) можуть бути легко надіслані роботу і потім надіслані зареєстрованим користувачам чату. Створення команд можливе через просте створення тригерів у проєкті OPC Router. Таким чином, ви можете створити Telegram Bot для своєї власної системи лише за кілька хвилин.

2.2 Моделювання процесу роботи чат-боту

Саме створення свого робота зазвичай дуже просте. У цьому вам допоможе

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 19 |

віртуальний користувач Botfather, який є центральним інструментом розробки ботів Telegram. Тому спочатку потрібно знайти цього користувача в Telegram і почати з ним чат. Переконайтеся, що ви вибрали верифікованого користувача з синьою галочкою(Рисунок 2.2).

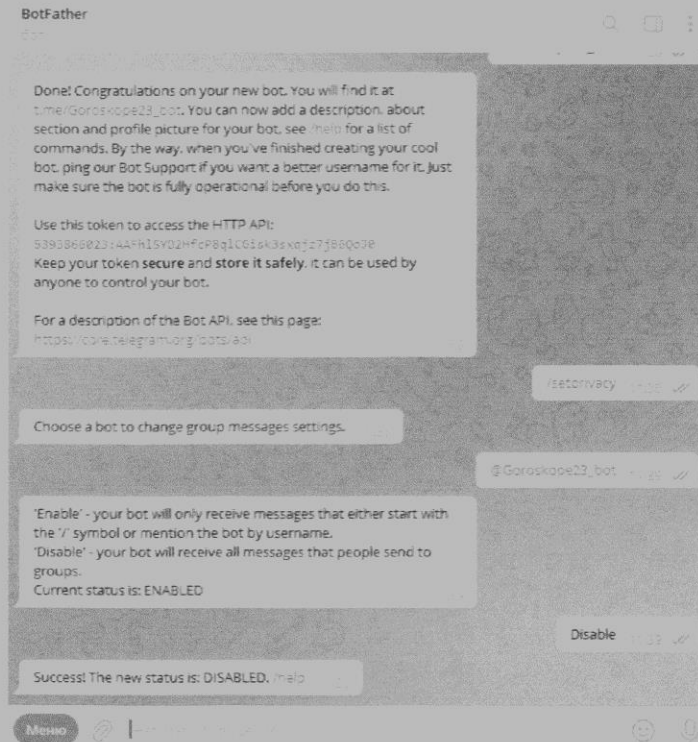


Рисунок 2.2 – чат з Botfather

Можна почати із введення команди «/newbot» у вигляді текстового повідомлення в чаті з бот-батьком. Тепер потрібно дати своєму боту ім'я, наприклад "Перший бот для тестування". Потім запитується ім'я користувача робота. Воно має закінчуватись на «bot». У нашому випадку, наприклад «first_test_bot». Тепер бот-батько повинен привітати Вас з налаштуванням нового бота та повідомити адресу та персональний «токен». Це, так би мовити, пароль для повного контролю над роботом. На цьому базове налаштування робота Telegram завершено. Тепер бота

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |
| | | | | | 20 | |

потрібно лише наповнити функціоналом або запрограмувати.

2.3 Архітектура проектування та функціональна структура

Архітектура програмного забезпечення системи відображає організацію або структуру системи та дає пояснення того, як вона поводиться. Система є сукупністю компонентів, що виконують певну функцію або набір функцій. Іншими словами, архітектура програмного забезпечення забезпечує міцний фундамент, на якому можна побудувати програмне забезпечення. Ряд архітектурних рішень та компромісів впливає на якість, продуктивність, ремонтпридатність та загальний успіх системи. Якщо не врахувати загальні проблеми та довгострокові наслідки, система може опинитися у зоні ризику. Існує кілька високорівневих архітектурних моделей та принципів, які зазвичай використовуються в сучасних системах. Їх часто називають архітектурними стилями. Архітектура програмної системи рідко обмежується одним архітектурним стилем. Навпаки, комбінація стилів часто складає повну систему.

Представимо опорну модель для обговорення частково автоматизованих ботів і з її допомогою порівняємо існуючі приклади. Йдеться лише про чат-ботах, що вимагають втручання людини, а не про боти, заздалегідь навчені на даних.

На рисунку показані типові компоненти архітектури чат-бота. Після отримання питання від користувача (наприклад, «Яка погода в Хмельницькому?») компонент розуміння мови (language understanding, LU) аналізує його для з'ясування наміру та пов'язаних відомостей (намір: «дізнатися передбачення на сьогодні»; сутності: [час: «сьогодні»]). Коли запит зрозумілий, відбувається виконання дії та вилучення інформації (activity execution, information retrieval, AEIR) із джерел даних (data source, DS) чат-бота (наприклад, надсилання запиту до API метеорологічного сайту та отримання відповіді). Після цього модуль створення відповіді (response generation, RG) формує відгук («Сьогодні вас очікує удача в

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 21 |

описує очікувану поведінку системи у вигляді функціональних та нефункціональних вимог. Ці вимоги повинні бути чіткими, здійсненними, вимірними та простежуваними до бізнес-вимог. Вимоги також повинні визначати, як програмне забезпечення має взаємодіяти з людьми, обладнанням та іншими системами.

- **Високорівневий дизайн.** Високорівневе проектування розбиває архітектурний проект системи на менш абстрактне уявлення підсистем та модулів та зображує їхню взаємодію один з одним. Ця перспектива високорівневого проектування фокусується у тому, як система разом із її компонентами реалізується як модулів. Вона визнає модульну структуру кожної підсистеми та їхню взаємодію між собою.

- **Детальне проектування.** Детальне проектування включає реалізацію того, що видно як система та її підсистеми у високорівневому проектуванні. Ця діяльність є більш детальною по відношенню до модулів та їх реалізації. Він визначає логічну структуру кожного модуля та його інтерфейси взаємодії коїться з іншими модулями.

Який зв'язок між архітектурою програмного забезпечення та проектуванням програмного забезпечення?

Архітектура програмного забезпечення розкриває структуру системи, приховуючи у своїй деталі реалізації. Архітектура також фокусується на тому, як елементи та компоненти системи взаємодіють один з одним. Проектування програмного забезпечення заглиблюється у деталі реалізації системи. Питання проектування включають вибір структур даних та алгоритмів або деталі реалізації окремих компонентів.

Питання архітектури та проектування часто перетинаються. Замість того, щоб використовувати жорсткі та швидкі правила для розмежування архітектури та дизайну, має сенс об'єднати їх. У деяких випадках рішення мають швидше архітектурний характер. В інших випадках рішення значною мірою зосереджені на дизайні та на тому, як він допомагає реалізувати цю архітектуру.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 23 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

Важливо відзначити, що архітектура – це дизайн, але не всякий дизайн є архітектурним. Насправді архітектор - це той, хто проводить кордон між архітектурою програмного забезпечення (архітектурний дизайн) та детальним дизайном (неархітектурний дизайн). Немає правил чи рекомендацій, які б підходили всім випадків, хоча спроби формалізувати цю відмінність робилися.

Сучасні тенденції в архітектурі програмного забезпечення припускають, що дизайн розвивається з часом і що архітектор програмного забезпечення не може знати все наперед, щоб повністю спроектувати систему. Дизайн зазвичай розвивається на етапах реалізації системи. Архітектор програмного забезпечення постійно навчається та тестує проект на відповідність вимогам реального світу.

Закладка правильної архітектури для програмного рішення подібна до закладки фундаменту для будинку. Якщо на фундаменті, призначеному для 1-поверхового будинку, почати будувати 9 поверхів, то природно, що вся конструкція розвалиться одночасно.

Тому у випадку з програмуванням не можна зробити робота з коротким скриптом на Python, а потім очікувати, що його функціональність можна буде доопрацювати до повноцінної системи.

У чому різниця між простим ботом та системою і що таке система взагалі?

Будь-яка система або програмний комплекс складається з двох основних частин – Front End та Back End.

Основним завданням Front End є забезпечення інтерфейсу та взаємодії з користувачем, а Back End, зі свого боку, це обробка запитів користувачів, адміністрування як системи, так і самих користувачів.

З іншого боку, наявність веб-інтерфейсу не означає, що він є Front End.

Тому під час проектування системи обов'язковими елементами є:

- Модульна система
- Наявність Front End, Back End, системи адміністрування
- Можливість розширення функціональності програмного забезпечення

за допомогою модулів або плагінів

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 24 |

2.4 Модель аналізу програмного забезпечення

Діаграма варіантів використання є типом поведінкової діаграми UML і часто використовується для аналізу різних систем. Вони дозволяють візуалізувати різні типи ролей у системі і те, як ці ролі взаємодіють із системою. Цей підручник з діаграм варіантів використання охопить наступні теми і допоможе вам краще створювати варіанти використання.

Важливість діаграм варіантів використання полягає в тому, що вони застосовуються для збору вимог до використання системи. Залежно від ваших вимог, ви можете використовувати ці дані різними способами. Нижче наведено кілька способів їхнього використання.

- Для визначення функцій та того, як ролі взаємодіють з ними – основна мета діаграм варіантів використання.
- Для високорівневого представлення системи – особливо корисно при поданні керівникам чи зацікавленим особам. Ви можете виділити ролі, які взаємодіють із системою, та функціональність, що надається системою, не заглиблюючись у внутрішню роботу системи.
- Для ідентифікації внутрішніх та зовнішніх факторів це може здатися простим, але у великих складних проектах система може бути ідентифікована як зовнішня роль в іншому прикладі використання.

Для моделювання системи найважливішим аспектом є захоплення динамічної поведінки. Динамічна поведінка означає поведінку системи під час її роботи.

Тільки статичної поведінки недостатньо для моделювання системи, швидше динамічна поведінка важливіша за статичну. У UML існує п'ять діаграм для моделювання динамічної природи, і діаграма використання сценаріїв є однією з них. Тепер, оскільки ми маємо обговорити, що діаграма варіантів використання є динамічною за своєю природою, маємо існувати деякі внутрішні чи зовнішні фактори для здійснення взаємодії.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | ДПІПЗ.180119.01.13.ВД | 26 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Ці внутрішні та зовнішні агенти відомі як дійові особи. Діаграма варіантів використання складається з дійових осіб, варіантів використання та їх взаємовідносин. Діаграма використовується для моделювання системи/підсистеми програми. Одна діаграма варіантів використання відбиває певну функціональність системи.

Отже, для моделювання всієї системи використовують кілька діаграм варіантів використання.

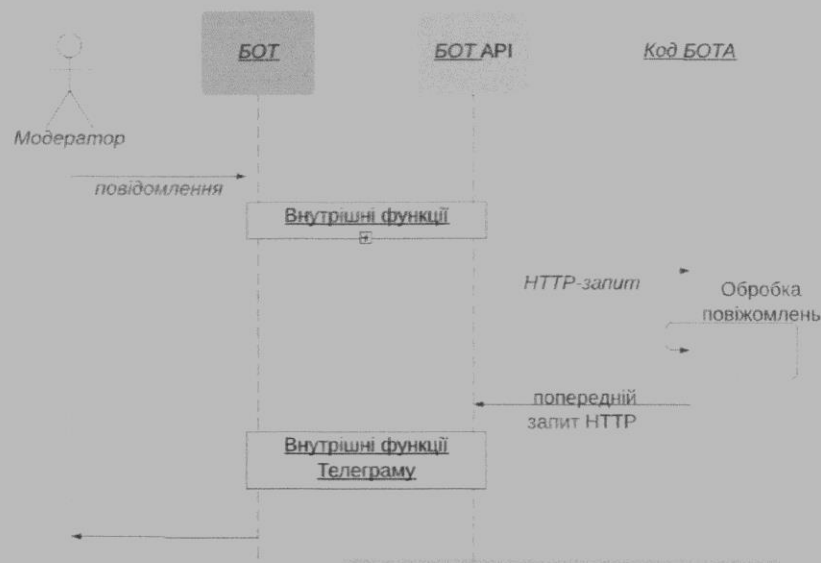


Рисунок 2.5

Обов'язково слід знати, як працює фоновий процес. Якщо говорити простіше, припустимо, що користувач надсилає повідомлення, скажімо, "Розкажи мені щось цікаве" вашому боту. Потім ваш бот надсилає цю інформацію до Telegram API (або Bot API). Цей Telegram API відповідає за надсилання цієї інформації туди, де знаходиться ваш PHP-код. Ваш PHP-код буде десь розміщений, тому Telegram API надсилає інформацію до вашого PHP-файлу, використовуючи ваше доменне ім'я. Ваш PHP-файл містить код для створення випадкового факту. Після того як ваш код згенерує випадковий факт, цей факт відправляється назад в Telegram API. І, нарешті,

Telegram API відправляє факт назад користувачеві як повідомлення.

Telegram дозволяє надсилати інформацію через Telegram API (Bot API) лише для того, щоб переконатися, що бот авторизований. Він перевіряє це за допомогою унікального токена, який ви отримуєте під час створення нового бота.

2.5 Вибір реляційної системи керування БД

База даних – це набір даних, які у комп'ютері. Ці дані зазвичай структуровані таким чином, щоб зробити їх доступними.

Реляційна база даних – це тип бази даних. У ній використовується структура, яка дозволяє нам визначати дані та отримувати доступ до них залежно від того, як вони пов'язані з іншими даними в базі даних. Часто дані у реляційній базі даних.

Реляційна система управління базами даних (РСУБД) - це програма, що дозволяє створювати, оновлювати та адмініструвати реляційну базу даних. Більшість систем керування реляційними базами даних використовують мову SQL для доступу до бази даних.

У моделі реляційної бази даних структури даних – включаючи таблиці даних, індекси та уявлення – залишаються окремими від фізичних структур зберігання, що дозволяє адміністраторам баз даних редагувати фізичне сховище даних, не торкаючись логічну структуру даних.

На підприємствах реляційні бази даних використовуються для організації даних та виявлення взаємозв'язків між ключовими точками даних. Вони дозволяють легко сортувати та знаходити інформацію, що допомагає організаціям більш ефективно приймати бізнес-рішення та мінімізувати витрати. Вони добре працюють із структурованими даними.

Таблиці даних, що використовуються в реляційній базі даних, зберігають інформацію про пов'язані об'єкти. Кожен рядок містить запис із унікальним ідентифікатором - відомим як ключ, - а кожен стовпець містить атрибути даних.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 28 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

Кожен запис надає значення кожній ознаки, що дозволяє легко визначити взаємозв'язок між точками даних.

Стандартним користувальницьким та прикладним програмним інтерфейсом (API) реляційної бази даних є мова структурованих запитів. Оператори SQL коду використовуються як для інтерактивних запитів до інформації з реляційної бази даних, так і для збору даних для звітів. Для забезпечення точності та доступності реляційної бази даних необхідно дотримуватися певних правил цілісності даних.

SQL – це мова програмування, використовуваний взаємодії з даними, які у реляційної системі управління базами даних.

Багато RDBMSs використовують SQL (і його різновиди) доступу до даних у таблицях. Наприклад, SQLite – це реляційна система управління базами даних. SQLite містить мінімальний набір команд SQL. В інших RDBMSs можуть використовуватись інші варіанти.

MySQL – найпопулярніша база даних SQL з відкритим вихідним кодом. Вона зазвичай використовується для розробки веб-застосунків, і доступ до неї часто здійснюється за допомогою PHP.

Основні переваги MySQL полягають у тому, що вона проста у використанні, недорога, надійна та має велику спільноту розробників, які можуть допомогти відповісти на запитання.

До недоліків можна віднести низьку продуктивність при масштабуванні, відставання у розвитку з відкритим кодом після того, як Oracle взяла MySQL під свій контроль, а також відсутність деяких додаткових функцій, до яких звикли розробники.

PostgreSQL – це база даних SQL з відкритим вихідним кодом, яка не контролюється якоюсь корпорацією. Вона зазвичай використовується для розробки веб-програм.

PostgreSQL має ті ж переваги, що і MySQL. Він простий у використанні, недорогий, надійний і має велику спільноту розробників. Він також надає деякі додаткові можливості, такі як підтримка зовнішніх ключів, не потребуючи складної конфігурації.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 29 |

Основним недоліком PostgreSQL є те, що він може бути повільнішим за продуктивністю, ніж інші бази даних, такі як MySQL. Він також дещо менш популярний, ніж MySQL.

Корпорація Oracle володіє Oracle Database і її код не є відкритим.

Oracle DB призначена для великих програм, особливо в банківській сфері. Більшість провідних банків світу використовують програми Oracle, тому що Oracle пропонує потужне поєднання технологій та комплексних, попередньо інтегрованих бізнес-додатків, включаючи важливі функції, створені спеціально для банків.

Основним недоліком використання Oracle є те, що він не є безкоштовним для використання як його конкуренти з відкритим вихідним кодом, і може бути досить дорогим.

Microsoft має SQL Server. Як і Oracle DB, код закритий.

Великі корпоративні програми переважно використовують SQL Server.

Microsoft пропонує безкоштовну версію початкового рівня під назвою Express, але вона може стати дуже дорогою в міру масштабування програми.

SQLite – це популярна база даних SQL з відкритим вихідним кодом. Вона може зберігати всю базу даних одному файлі. Одне з найважливіших переваг цієї бази даних у тому, що це дані можуть зберігатися локально, без необхідності підключення бази даних до серверу.

SQLite – популярний вибір для баз даних у мобільних телефонах, КПК, MP3-плеєрах, телеприставках та інших електронних гаджетах. Курси SQL на Codecademy використовують SQLite.

Реляційні бази даних зберігають дані у таблицях. Таблиці можуть бути більшими і містити безліч стовпців та записів. Системи управління реляційними базами даних (RDBMS) використовують SQL (і різновиди SQL) для управління даними у великих таблицях. Вибір використовуваної вами РСУБД залежить від складності вашої програми.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 30 |

2.6 Проектування моделі бази даних

Діаграма класів UML – це графічна нотація, що використовується для побудови та візуалізації об'єктно-орієнтованих систем. Діаграма класів в Уніфікованій мові моделювання (UML) – це тип статичної структурної діаграми, яка описує структуру системи, показуючи її:

- класи
- їх атрибути
- операції (або методи)
- та відносини між об'єктами.

Діаграма класів – це статична діаграма, яка використовується для моделювання статичного уявлення системи. Статичний вигляд визначає словниковий запас системи.

Діаграма класів також вважається основою для діаграм компонентів та розгортання. Діаграми класів використовуються не тільки для візуалізації статичного уявлення системи, але і для побудови коду, що виконується при прямому і зворотному проектуванні будь-якої системи.

Як правило, діаграми UML не порівнюються безпосередньо з об'єктно-орієнтованими мовами програмування, але діаграма класів є винятком.

Діаграма класів чітко показує зіставлення з об'єктно-орієнтованими мовами, як-от Java, C++ та інших. З практичного досвіду діаграма класів зазвичай використовується з метою конструювання.

Двома словами можна сказати, що діаграми класів використовуються для:

- Опис статичного уявлення системи.
- Показати взаємодію між елементами статичного уявлення.
- Опис функціональних можливостей, що виконуються системою.
- Побудова програмних програм з використанням об'єктно-орієнтованих мов.

мов.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 31 |

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Реалізація створення чат-боту

Telegram работает в облаке, а это значит, что вы можете заходить в чаты на разных платформах одновременно. Благодаря функции мгновенной синхронизации.

Спочатку завантажте програму Telegram для настільного комп'ютера. Після цього вам потрібно буде взаємодіяти з основним ботом Telegram - botfather.

Досить помітна назва, тому що всі боти, коли-небудь створені в Telegram, походять від нього. Знайдіть його на вкладці пошуку.

Щоб викликати реакцію у відповідь, введіть «/start» на botfather, який видасть вам список команд. Процес видно на Рисунку 3.1

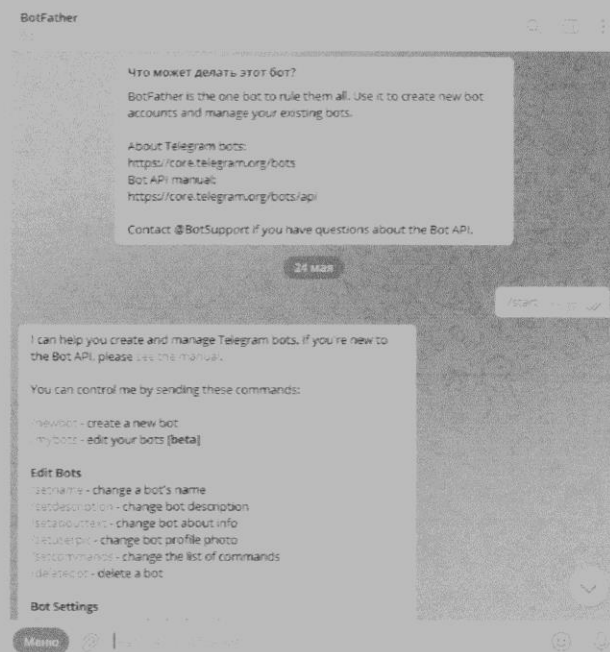


Рисунок 3.1 – чат з botfather

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 32 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

Команда «/newbot» попросить вас назвати вашого бота та дати йому ім'я користувача.

Ім'я – це те, що ваші підписники бачитимуть у розмові. Вони знайдуть робота на ім'я користувача. Рекомендується дати боту дружнє ім'я, що містить прогалини – Goroskope.

Ім'я користувача відрізняється, воно не повинно містити пробілів і повинно мати слово «bot» як суфікс. Воно має складатися з 5-32 символів і може містити латиницю, цифри або символи підкреслення.

Ми дали нашому роботі ім'я користувача «Goroskope23_bot».

Після створення імені користувача ви отримаєте токен (Рисунок 3.2). Токен необхідний для керування ботом та відправлення його в Vots API.

Зберігайте його у безпеці та нікому не показуйте. Токен знадобиться вам пізніше.

```
Use this token to access the HTTP API:  
5393866023:AAFh15xD2HfcP8q1CGIsk3sXqjC7jB8Qo70  
Keep your token secure and store it safely, it can be used by  
anyone to control your bot.
```

Рисунок 3.2 – Токен

Якщо хтось украде ваш токен або ви його втратите, скористайтесь командою «/token», щоб згенерувати новий.



Рисунок 3.3 – Токен в коді

Ця бібліотека забезпечує чистий Python, асинхронний інтерфейс для Telegram

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 33 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

Bot API. Вона сумісна з версією Python 3.7 і вище.

Крім чистої реалізації API, ця бібліотека містить ряд високорівневих класів, щоб зробити розробку роботів простий і зрозумілий. Ці класи містяться у підмодулі telegram.ext.

Чиста реалізація API без telegram.ext доступна у вигляді окремого пакету python-telegram-bot-raw.

Спільне встановлення python-telegram-bot та python-telegram-bot-raw призведе до небажаних побічних ефектів, тому потрібно встановлювати лише один із них.

Файл «sqlite_db.py» наповнений базою даних, якщо ви підключили свій токен до коду, то в консольному вікні виведеться повідомлення «ПІДКЛЮЧЕННЯ ДО БАЗИ ДАНИХ ВДАЛЕ!». Далі потрібно впевнитись що ви є модератором, а інакше ви не зможете вносити в базу даних нову інформацію. Наш бот налаштований на два повідомлення, які записує в базу.

Користувач повинен вписати слово, в нашому випадку це знак зодіаку, і на виході бот має відповісти прогноз для вписаного знаку зодіаку, в іншому випадку бот відповість користувачу «Можливо, з часом, ми знайдемо для Вас відповідь але не наразі».

```
def sql_start():
    global base, cur
    base=sq.connect('baze_menu.db')
    cur=base.cursor()
    if base:
        print('ПІДКЛЮЧЕННЯ ДО БАЗИ ДАНИХ ВДАЛЕ!')
        base.execute('CREATE TABLE IF NOT EXISTS horoscope (img
TEXT, k_s TEXT PRIMARY KEY, description TEXT, size TEXT)')
        base.execute('CREATE TABLE IF NOT EXISTS table_user
(user_id INTEGER PRIMARY KEY,user_name TEXT, clef_ INTEGER)')
        base.commit()
```

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | ДПІПЗ.180119.01.13.ВД | 34 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

async def sql_add_command(state):
    async with state.proxy() as data:
        cur.execute('INSERT INTO horoscope VALUES (?, ?, ?,
?), tuple(data.values()))
        base.commit()
    async def sql_read(message):
        j=message.text
        print(j)
        k=f"SELECT img, description, k_s FROM horoscope WHERE
k_s='{j}'"
        try:
            if cur.execute(k).fetchmany(1):
                for ret in cur.execute(k).fetchmany(1):
                    await bot.send_photo(message.from_user.id,
ret[0],f'\n{ret[1]}')
            else:
                await bot.send_message(message.from_user.id,
"Можливо, з часом, ми знайдемо для Вас відповідь але не наразі.")
        except:
            await bot.send_message(message.from_user.id, "На
жаль, тут ми безсилі Вам щось повідомити.")
        finally:
            print('[INFO 0] sql_read')
    async def sql_add_user_(message):
        user_id=message.from_user.id
        username=message.from_user.username
        k_user = f"SELECT user_id, clef_ FROM table_user WHERE
user_id='{user_id}'"
        print(cur.execute(k_user).fetchall())

```

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПППЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 35 |

```

result_k_user=cur.execute(k_user).fetchone()
try:
    print('[INFO 0] sql_add_user_')
    if not result_k_user:
        print('[INFO 00] sql_add_user_')
        cur.execute('INSERT INTO table_user VALUES (?,
?, ?)', (user_id, username, 1))
        base.commit()
except:
    print('[INFO 1] sql_add_user_')
finally:
    print(f'[INFO 2] sql_add_user_')
async def sql_audit_user_(message):
    user_id=message.from_user.id
    username=message.from_user.username
    k_user = f"SELECT user_id, clef_ FROM table_user WHERE
user_id='{user_id}'"
    result_k_user = cur.execute(k_user).fetchone()
    if result_k_user[-1]==2:
        global ID, moderator_statys
        ID = result_k_user[0]
        moderator_statys=f'ЗАВАНТАЖИТИ -/download \nВИ
ПЕРЕДУМАЛИ РОБИТИ\nЗАПИС, НАТИСНІТЬ \n-/abolition'
        print(f'[INFO 8] Ви модератор')
    elif not result_k_user[-1]==2:
        moderator_statys=f'Ви не модератор.'
        print(f'[INFO 9] Ви не модератор')

```

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |
| | | | | | 36 | |

3.2 Реалізація логіки взаємодії з додатком

Роботи Telegram - це спеціальні облікові записи, які не потребують додаткового номера телефону. Ці облікові записи є інтерфейсом для коду, запущеного десь на вашому сервері.

Bot API – це HTTP-інтерфейс, створений для розробників, які бажають створювати ботів для Telegram.

Архітектура робота Telegram. Наш смартфон, що взаємодіє з сервером Telegram як мост між клієнтом і сервером бота. Використовуються два основних терміни, а саме запити та оновлення. Сервер Admin запитує через сервер Telegram як міст між мобільним пристроєм і контрольованим сервером, потім сервер Telegram запускає процес опитування за протоколом HTTPS сервер, а сервер бота надає оновлення за результатами запитів сервера Admin. Для цього при проектуванні використовується Telegram Bot API як проміжне середовище для Admin Server. Telegram Bot API - це API (інтерфейс прикладного програмування), який використовується для віддаленого моніторингу за допомогою роботів як програм, запущених на сервері. Telegram Bot API використовує протокол шифрування MTProto, який був підготовлений для розробників як засіб безпеки.

Ідея полягає в тому, щоб згенерувати для кожного бота шаблон додатка, який буде наповнений вашою власною логікою.

Згенерований додаток реалізує тривіальну логіку ехо-сервера, це просто приклад для використання як скелет для вашого справжнього додатка бота. Щоб згенерувати скелетну програму для токена YOUR_BOT_TOKEN, відповідного Telegram-боту Yourbot:

```
storage = MemoryStorage()
bot = Bot(token=config.TOKEN)
dp = Dispatcher(bot, storage=storage)
```

MemoryStorage() – Bot Framework SDK дозволяє зберігати дані, що вводяться

| | | | | | | |
|------|------|----------|--------|------|--|------|
| | | | | | | Арк. |
| | | | | | | 37 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ДПІПЗ.180119.01.13.ВД

користувачем, у пам'яті. Оскільки при перезапуску бота сховище в пам'яті очищається щоразу, воно найкраще підходить для тестування і не призначене для використання у виробництві. Постійні типи зберігання, такі як зберігання бази даних, найкраще підходять для виробничих ботів.

Dispatcher() – це об'єкт контексту, що передається в зворотний дзвінок, викликаний «telegram.ext.Handler» або «telegram.ext.Dispatcher» в обробнику помилок, доданому «telegram.ext.Dispatcher.add_error_handler» або в зворотний дзвінок «telegram.ext.Job».

«telegram.ext.Dispatcher» буде створювати один контекст для оновлення. Це означає, що якщо у вас є два обробники в різних групах і вони обидва викликаються, їм буде переданий той самий об'єкт CallbackContext (звісно, з відповідними атрибутами, такими як .matches differing). Це дозволяє додавати атрибути користувача до зворотного виклику нижчої групи обробників, а потім звертатися до цих атрибутів у зворотний виклик більш високої групи обробників. Зверніть увагу, що атрибути CallbackContext можуть змінитися в майбутньому, тому переконайтеся, що ви використовуєте досить унікальне ім'я для атрибутів.

#Ловимо першу відповідь і записуємо в словник if message.from_user.id ID:

```
async def load_photo(message: types.Message, state: FSMContext):
```

```
    if message.from_user.id == sqlite_db.ID:
```

```
        async with state.proxy() as data:
```

```
            data['photo'] = message.photo[0].file_id
```

```
        await FSMAdmin.next()
```

```
        await message.reply("Введи ключове слово (або фразу)
```

```
по якому отримувати інформацію на гороскоп\n\nВИ ПЕРЕДУМАЛИ РОБИТИ\nЗАПИС, НАТИСНІТЬ\n-/abolition")
```

Firebase Admin Python SDK дозволяє розробникам Python на стороні сервера (backend) інтегрувати Firebase у свої сервіси та програми.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 38 |

3.3. Структура проекту

По закінченню проекту буде видно структуру нашого запрограмованого Телеграм-боту. З цього списку компонентів програми найважливіші є папка «venv» та «base_menu.db»(Рисунок 3.4)

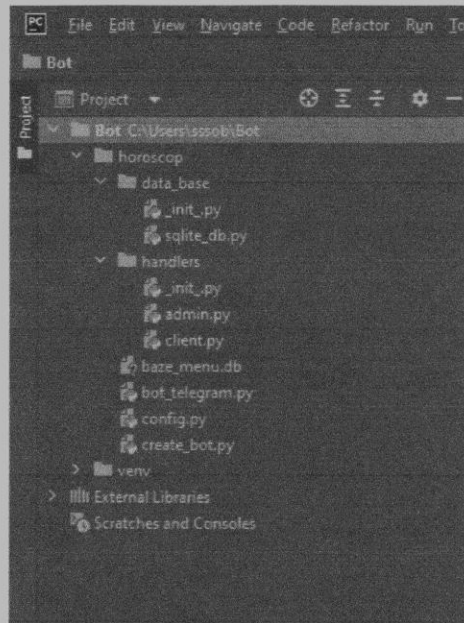


Рисунок 3.4 – Компоненти програми

Модуль venv забезпечує підтримку для створення легких «віртуальних середовищ» із власними каталогами сайтів, за бажанням ізольованими від системних каталогів сайтів. Кожне віртуальне середовище має власний двійковий файл Python (що відповідає версії двійкового файлу, який був використаний для створення цього середовища) і може мати свій незалежний набір встановлених пакетів Python у каталогах сайтів.

base_menu.db – це типічна база даних (БД), яка являється організованою структурою, призначена для зберігання, зміни та обробки взаємопов'язаної

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | ДПІПЗ.180119.01.13.ВД | 39 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

інформації, переважно більших обсягів. Бази даних активно використовуються для динамічних сайтів із значними обсягами даних – часто це інтернет-магазини, портали, корпоративні сайти. Такі сайти зазвичай розроблені за допомогою серверної мови програмування (наприклад, PHP) або на основі CMS (наприклад, WordPress), і не мають готових сторінок з даними за аналогією з HTML-сайтами. Сторінки динамічних сайтів формуються на льоту в результаті взаємодії скриптів і баз даних після відповідного запиту клієнта до веб-сервера.

Програма містить головні каталоги: `data_base`, `handlers`.

- `Handlers` – директорія яка буде містити різні хендлери. Для зручності там створений файл `«_init_.py»` з імпортами `«admin.py»` та `«client.py»`;
- `data_base` – директорія які містить імпорт модуля `«sqlite_db.py»`. Так як бот асинхронний, то він викоистовує один потік виконання. База даних влаштована як встроєний модуль `sqlite3` працює прямо у додатку.

SQLite — це бібліотека C, яка забезпечує легку дискову базу даних, яка не вимагає окремого серверного процесу та дозволяє отримати доступ до бази даних за допомогою нестандартного варіанту мови запитів SQL. Деякі програми можуть використовувати SQLite для внутрішнього зберігання даних. Також можна створити прототип програми за допомогою SQLite, а потім перенести код до більшої бази даних, наприклад PostgreSQL або Oracle.

Модуль `sqlite3` забезпечує інтерфейс SQL, сумісний зі специфікацією DB-API 2.0, описаною в PEP 249 і вимагає SQLite 3.7.15 або новішої версії.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| | | | | | | | 40 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |

4 ТЕСТУВАННЯ ТЕЛЕГРАМ БОТУ

4.1 Аналіз методів тестування

Тестування програмного забезпечення - це процес оцінки та перевірки того, що програмний продукт або програма робить те, що має робити. Переваги тестування включають запобігання помилок, зниження витрат за розробку і підвищення продуктивності.

Існує безліч різних типів тестування програмного забезпечення, кожен з яких має певні цілі та стратегії:

- Перевірка того, чи працює вся система так, як задумано.
- Інтеграційне тестування: Забезпечує спільну роботу компонентів або функцій програмного забезпечення.
- Модульне тестування: Перевірте, що кожна одиниця програмного забезпечення працює так, як очікується. Блок - це найменший компонент програми, що тестується.
- Функціональне тестування: Перевірка функцій шляхом емуляції бізнес-сценаріїв, що базуються на функціональних вимогах. Тестування "чорної скриньки" є найпоширенішим способом перевірки функцій.
- Тестування продуктивності: Перевірка роботи програмного забезпечення за різних робочих навантажень. Навантаження тестування, наприклад, використовується для оцінки продуктивності в умовах реального навантаження.
- Регресійне тестування: Перевірка того, чи не порушують нові функції, чи не погіршують вони функціональність. Регресійне тестування може використовуватись для перевірки меню, функцій та команд на поверхневому рівні, коли немає часу на проведення повного регресійного тестування.
- Стрес-тестування: Перевірка того, яке навантаження може витримати

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 41 |

система, перш ніж вона вийде з ладу. Вважається одним із видів дисфункційного тестування.

- usability-тестування: Перевірка того, наскільки добре клієнт може використовувати систему або веб-програму для виконання завдання.

У іншому разі, перевірка базових вимог є критичною оцінкою. Не менш важливо і те, що дослідницьке тестування допомагає тестувальнику або команді тестувальників виявити складні сценарії та ситуації, які можуть призвести до помилок у програмному забезпеченні.

Навіть простий додаток може бути підданий великій кількості та різноманітності тестів. План управління тестуванням допомагає визначити пріоритети, які види тестування забезпечують найбільшу цінність з урахуванням наявного часу та ресурсів. Ефективність тестування оптимізується шляхом проведення найменшої кількості тестів виявлення найбільшої кількості дефектів.

Мало хто може оскаржити необхідність контролю якості під час розробки програмного забезпечення. Невчасне постачання або дефекти програмного забезпечення можуть завдати шкоди репутації бренду, що призведе до розчарування та втрати клієнтів. В екстремальних випадках помилка або дефект можуть погіршити роботу взаємопов'язаних систем або спричинити серйозні збої.

Саме тестування коштує грошей, компанії можуть заощадити мільйони на рік на розробці та підтримці, якщо вони мають хорошу методику тестування та процеси QA. Раннє тестування програмного забезпечення дозволяє виявити проблеми, перш ніж продукт вийде з ринку. Чим раніше команди розробників отримують відгуки про тестування, тим швидше вони зможуть вирішити такі проблеми, як:

- архітектурні недоліки
- погані проектні рішення
- неправильна або неправильна функціональність
- уразливості безпеки
- проблеми масштабованості

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 42 |

Коли розробка залишає достатньо місця для тестування, це підвищує надійність програмного забезпечення і високоякісні програми поставляються з невеликою кількістю помилок. Система, яка відповідає очікуванням клієнтів або навіть перевершує їх, веде до потенційно більшої кількості продажів та збільшення частки ринку.

Тестування програмного забезпечення слід загальному процесу. Завдання або етапи включають визначення тестового середовища, розробку тестових прикладів, написання сценаріїв, аналіз результатів тестування та подання звітів про дефекти.

Тестування може забирати багато часу. Для невеликих проектів може бути достатньо ручного тестування або спеціального тестування. Однак для великих систем часто використовуються інструменти для автоматизації завдань. Автоматизоване тестування допомагає командам реалізовувати різні сценарії, тестувати відмінні риси (наприклад, переміщення компонентів у хмарне середовище) та швидко отримувати відгуки про те, що працює, а що ні.

Хороший підхід до тестування охоплює інтерфейс прикладного програмування (API), інтерфейс користувача і системний рівень. Крім того, чим більше тестів автоматизовано та запущено на ранній стадії, тим краще. Деякі команди створюють власні засоби автоматизації тестування. Однак, рішення постачальників пропонують функції, які можуть спростити ключові завдання управління тестуванням, такі як:

- **Безперервне тестування:** Проектні команди тестують кожен збірку в міру її надходження. Цей тип тестування програмного забезпечення спирається на автоматизацію тестування, інтегровану із процесом розгортання. Це дозволяє перевіряти програмне забезпечення у реалістичних тестових середовищах на ранніх етапах процесу, що покращує розробку та знижує ризики.
- **Управління конфігурацією:** Організації централізовано підтримують тестові активи та відстежують, які збирання програмного забезпечення підлягають тестуванню. Команди отримують доступ до таких активів, як код, вимоги, проектна

| | | | | | | | |
|------|------|----------|--------|------|-----------------------|--|------|
| | | | | | | | Арк. |
| | | | | | | | 43 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | | |

документація, моделі, сценарії тестування та результати тестування. Хороші системи включають автентифікацію користувачів та журнали аудиту, щоб допомогти командам виконати вимоги до відповідності з мінімальними адміністративними зусиллями.

- Віртуалізація послуг: Тестові середовища можуть бути недоступними, особливо на ранніх етапах розробки коду. Віртуалізація сервісів імітує відсутні або ще не завершені сервіси та системи, дозволяючи командам скоротити кількість залежностей та швидше проводити тестування. Вони можуть повторно використовувати, розгортати та змінювати конфігурацію для тестування різних сценаріїв без необхідності вносити зміни у вихідне середовище.

- Відстеження дефектів або помилок: Відстеження дефектів важливо як для команд тестування, так і для команд розробників для вимірювання та покращення якості. Автоматизовані інструменти дозволяють командам відслідковувати дефекти, вимірювати їх масштаб та вплив, а також виявляти пов'язані з ними проблеми.

- Метрики та звітність: Звітність та аналітика дозволяють членам команди ділитися статусом, цілями та результатами тестування. Передові інструменти інтегрують показники проекту та представляють результати у вигляді панелі приладів. Команди швидко бачать загальний стан проекту та можуть відслідковувати взаємозв'язки між тестуванням, розробкою та іншими елементами проекту.

Коли розпочинати тестування: Тестування слід починати якомога раніше, щоб скоротити витрати та час на доопрацювання та створити програмне забезпечення без помилок, щоб його можна було передати клієнту. Однак у життєвому циклі розробки програмного забезпечення (SDLC) тестування може бути розпочато з етапу збору вимог і продовжуватись доти, доки програмне забезпечення не буде запущене у виробництво. Це також залежить від моделі розробки. Наприклад, у водоспадній моделі тестування починається з фази тестування, яка знаходиться досить низько в дереві, а у V-моделі тестування

| | | | | | | |
|------|------|----------|--------|------|--|------|
| | | | | | | Арк. |
| | | | | | | 44 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

ДПІПЗ.180119.01.13.ВД

проводиться паралельно з фазою розробки.

Коли припиняти тестову діяльність: Програма ніколи не може бути на 100% вільною від помилок. Тому визначити коли можна припинити тестування, досить складно. Тим не менше, при припиненні тестової діяльності слід пам'ятати про такі речі:

- терміни тестування
- завершення циклу виконання тестових прикладів
- завершення функціонального тестування та покриття коду до узгодженої межі

узгодженої межі

- не знайдено жодної високопріоритетної помилки і швидкість виявлення помилок знижується після певного рівня.

- управлінне рішення Процес тестування програмного забезпечення складається з наведених нижче дій:

- планування та контроль
- аналіз та проектування
- реалізація та виконання
- виконання критеріїв завершення
- заходи щодо завершення тестування.

Ось переваги використання тестування програмного забезпечення:

- Економічність: Це одна з найважливіших переваг тестування програмного забезпечення. Своєчасне тестування будь-якого ІТ-проекту допомагає вам заощадити гроші у довгостроковій перспективі. Якщо помилки виявлені на ранній стадії тестування програмного забезпечення, їх виправлення обходиться дешевше.

- Безпека: Це найбільш уразлива та чутлива перевага тестування програмного забезпечення. Люди шукають надійних продуктів. Це допомагає усунути ризики та проблеми на ранній стадії.

- Якість продукту: Це найважливіша вимога до будь-якого програмного продукту. Тестування гарантує, що клієнтам буде надано якісний продукт.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 45 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

| | |
|----------------------------|--|
| | І так далі |
| Нефункціональне тестування | Продуктивність Витривалість Навантаження Об`єм Масштабованість Зручність використання І так далі |
| Технічне обслуговування | Регресія Технічне обслуговування |

Це не повний список, оскільки існує понад 150 видів тестування, і вони продовжують поповнюватися. Також зверніть увагу, що не всі види тестування застосовуються до всіх проектів, а залежать від характеру та масштабу проекту.

4.2 Тестування чат-боту

Отже настав час переглянути та скористатись Телеграм ботом. Першим ділом відкриваємо проект в середовищі програмування на Python – PyCharm. Запускаємо проект(Рисунок 4.2).

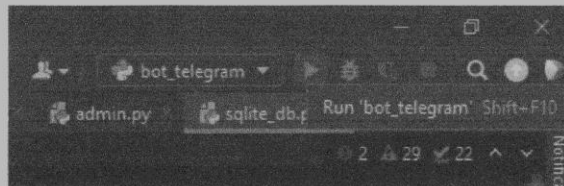
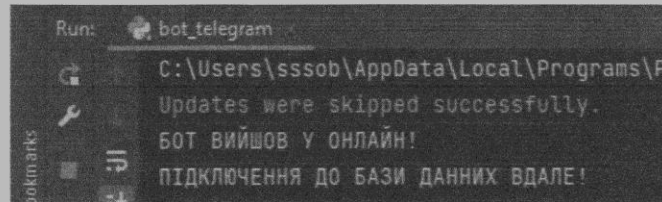


Рисунок 4.2 – Кнопка запуску проекту

| | | | | | | |
|------|------|----------|--------|------|--|------|
| | | | | | | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 47 |

ДПІПЗ.180119.01.13.ВД

Якщо вдасться підключитись, то в консольному вікні висвітиться таке повідомлення (див. на Рисунок 4.3).



```
Run: bot_telegram
C:\Users\sssob\AppData\Local\Programs\P
Updates were skipped successfully.
БОТ ВИЙШОВ У ОНЛАЙН!
ПІДКЛЮЧЕННЯ ДО БАЗИ ДАНИХ ВДАЛЕ!
```

Рисунок 4.3 – Повідомлення про успішність запуску програми

Тепер можна переходити в чат з ботом і починати з ним «знайомитися». Розпочинаємо з команди `"/start"`, нас вітає бот і просить ввести 33 або перейти у режим модератора. Якщо ви не являєтесь модератором, то бот вам відповість таким повідомленням «Ви не модератор»(Рисунок 4.4)

Вирішуємо проблему з модератором та переходимо до етапу додавання прогнозу для 33 за допомогою команди `«/download»`. Бот дозволяє нам загрузити картинку(Рисунок 4.5)

Далі бот вимагає ввести ключове слово по якому користувач буде отримувати інформацію про гороскоп(Рисунок 4.6).

Після ключового слова потрібно ввести саму інформацію яку отримає користувач.(Рисунок 4.6)

Telegram бот дозволяє за допомогою команди `«/download»` вносити в базу даних нову інформацію. Ми завантажимо ще 11 таких передбачень. Інформацію можна змінити в будь-який момент часу.

Коли Користувач вводить 33, то йому бот виводить повідомлення з його передбаченням.(Рисунок 4.7)

В консольному вікні впливає повідомлення прощання (Рисунок 4.8)

На рисунку 4.9 зображено кінцевий результат, як користувач буде використовувати бот.

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 48 |

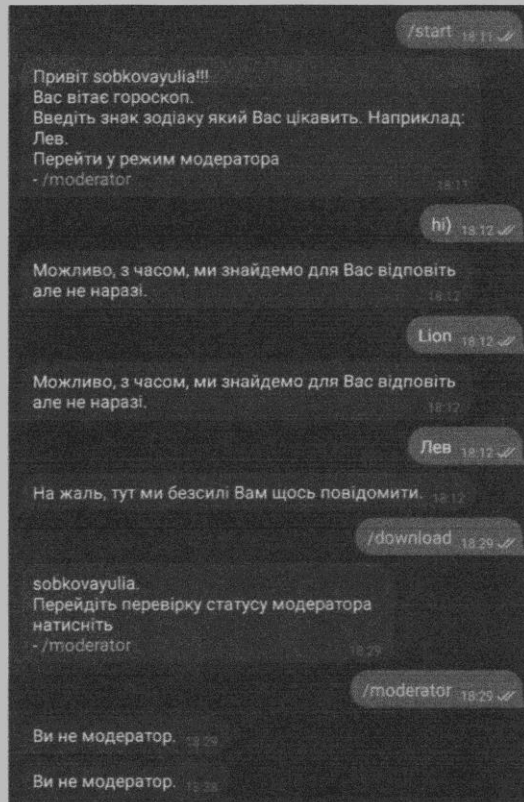


Рисунок 4.4 – чат з ботом(початок)

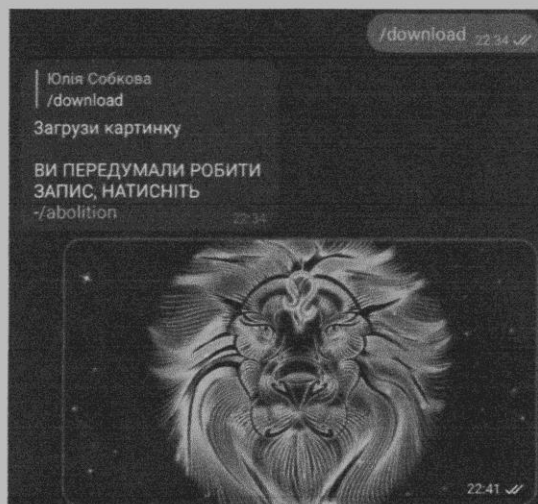


Рисунок 4.5 – чат з ботом(завантаження фото)

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 49 |

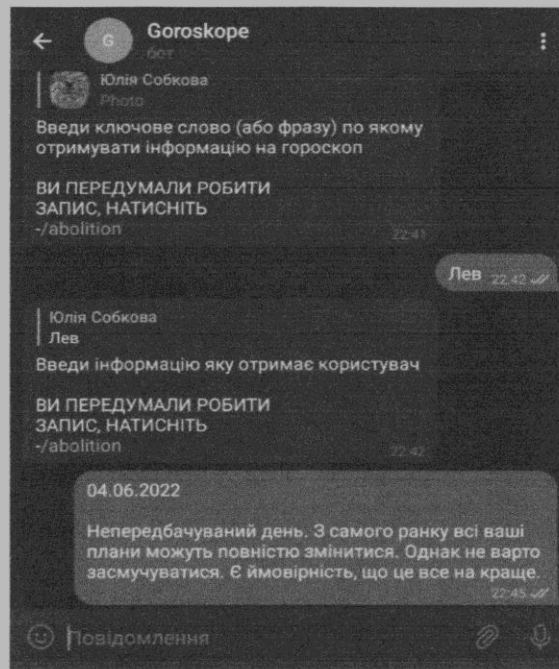


Рисунок 4.6 – чат з ботом(додавання ключового слова та інформацію)



Рисунок 4.7 – чат з ботом(перевірка роботи з ботом)

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 50 |

```

Run: bot telegram
  Пауза
  [INFO 0] sql_read
  Риби
  [INFO 0] sql_read
  Козеріг
  [INFO 0] sql_read
  Goodbye!

Process finished with exit code 0

```

Рисунок 4.8 – Вікно запуску



Рисунок 4.9 – чат користувача з ботом

| | | | | | | |
|------|------|----------|--------|------|--|------|
| | | | | | | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 51 |

ДППЗ.180119.01.13.ВД

4.3 Аналіз результатів тестування

Для початку, створення Телеграм боту за допомогою мови Python пройшло успішно. Було проведено базовий план по створенню бота. Під час користування ботом можна покращувати стабільність та можливості чату.

Телеграм бот працює коректно, також він виконує ті функції як йому було присвоєно. Чат повністю відповідає зазначеним вимогам і готовий використовуватися.

Було ефективно протестовано роботу чат-боту:

1. Перевірили процес розмови з чат-ботом. Залучили чат-бота в розмову і пройшлися по різних сценаріях спілкування, від потенційно найбільш частих або важливих фраз і випадків до виняткових ситуацій.

2. Використали специфічні терміни для перевірки надання достовірної інформації про продукт або послуги.

3. Перевірили ситуації, коли користувач вводить безглузде речення або вираз, який не дуже часто використовується. Звичайно, неможливо передбачити всю інформацію, яку можуть вводити користувачі та яка не відноситься до справи. Всі виключення, з якими чат-боту доведеться звертатися, перевірити не вийде. І все ж було розроблено «екстрені відповіді» для тих винятків, які можна передбачити. Це та фаза процесу тестування, коли потрібно перевірити, чи дає чат-бот зрозумілі відповіді в таких «виняткових» сценаріях.

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | 52 |

ВИСНОВОК

Всі поставлені перед нами задачі було виконано. Неодноразово виникали проблеми з програмуванням коду та підключення до бібліотек, але кінцевий результат був того вартий.

Тепер готового бота можна використовувати як віртуального помічника для вирішення найпростіших завдань. Додавання чат-ботів-помічників знижує накладні витрати, більш раціонально використовує час співробітників служби підтримки та дозволяє організаціям надавати послуги клієнтам у години, коли живі агенти недоступні.

Також організаціям, які прагнуть підвищити продуктивність продажів або обслуговування, можуть використовувати чат-ботів для економії часу та підвищення ефективності, оскільки чат-боти зі штучним інтелектом (ШІ) можуть спілкуватися з користувачами та відповідати на повторювані питання.

Чим більше кінцевий користувач взаємодіє з ботом, тим краще його система розпізнавання голосу передбачає відповідні відповіді.

Згідно з галузевими дослідженнями, пандемія COVID-19 значно прискорила впровадження та прийняття користувачами чат-ботів по всьому світу.

Однак можливості чат-ботів все ще обмежені. Навіть завдяки обробці природної мови вони можуть повністю розуміти введення даних клієнта і давати безладні відповіді. Багато чат-ботів також обмежені обсягом запитів, на які вони можуть відповісти. Це може призвести до розчарування через відсутність емоцій, співчуття та персоналізації у досить загальних відповідях. На додаток до незадоволеності клієнтів тим, що вони не можуть зв'язатися з людиною, чат-боти можуть бути дорогими у реалізації та обслуговуванні, особливо якщо їх необхідно часто налаштовувати та оновлювати.

| | | | | | | | |
|------|------|----------|--------|------|--|----------------------|------|
| | | | | | | ДППЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 53 |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Gerabot [Електронний ресурс] / Веб сайт Gerabot – режим доступу до ресурсу:https://gerabot.com/article/etapi_soglasovaniya_razrobotki_i_sozdaniya_chatbota. (Дата звернення – 19.05.2022)
- 2) Faq.in.ua Бібліотека користувача [Електронний ресурс] / Веб сайт Faq.in.ua – режим доступу до ресурсу:<https://faq.in.ua/articles/37-chatbot-dlya-telegram-stvorenniya-ta-nalashtuvannya-botiv-dlya-chativ-telehram.html> (Дата звернення – 19.05.2022)
- 3) Python [Електронний ресурс] / Веб сайт Python – режим доступу до ресурсу:<https://www.python.org/downloads/> (Дата звернення – 19.05.2022)
- 4) PyCharm [Електронний ресурс] / Веб сайт PyCharm – режим доступу до ресурсу:<https://www.jetbrains.com/ru-ru/pycharm/> (Дата звернення – 19.05.2022)
- 5) Justaskthales [Електронний ресурс] / Веб сайт Justaskthales – режим доступу до ресурсу:<https://justaskthales.com/en/telegram-different-messaging-apps/> (Дата звернення – 19.05.2022)
- 6) Insider [Електронний ресурс] / Веб сайт Insider – режим доступу до ресурсу: <https://www.businessinsider.com/what-is-telegram> (Дата звернення – 19.05.2022)
- 7) Medium [Електронний ресурс] / Веб сайт Medium – режим доступу до ресурсу: <https://chatbotslife.com/what-is-a-telegram-bot-reasons-to-use-bot-for-telegram-46b0d0579337> (Дата звернення – 19.05.2022)
- 8) Lucidchart [Електронний ресурс] / Веб сайт Lucidchart – режим доступу до ресурсу: <https://lucid.app/persona/lucidchart> (Дата звернення – 19.05.2022)
- 9) Vaadin [Електронний ресурс] / Веб сайт Vaadin – режим доступу до ресурсу: <https://vaadin.com/blog/building-a-chatbot-in-java> (Дата звернення – 19.05.2022)

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| | | | | | | | 54 |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |

– 19.05.2022)

10) Create and learn [Електронний ресурс] / Веб сайт Create and learn – режим доступу до ресурсу: <https://www.create-learn.us/blog/java-tutorial-for-beginners/> (Дата звернення – 19.05.2022)

11) Guru99 [Електронний ресурс] / Веб сайт Guru99 – режим доступу до ресурсу: <https://www.guru99.com/java-platform.html> (Дата звернення – 19.05.2022)

12) WhatIs [Електронний ресурс] / Веб сайт WhatIs – режим доступу до ресурсу: <https://www.techtarget.com/whatis/definition/Nodejs#:~:text=James%20Denman-,Node.,feeds%20and%20web%20push%20notifications.> Дата звернення – 19.05.2022)

13) Tutorialspoint [Електронний ресурс] / Веб сайт Tutorialspoint – режим доступу до ресурсу: https://www.tutorialspoint.com/python/python_overview.htm Дата звернення – 19.05.2022)

14) Synopsys [Електронний ресурс] / Веб сайт Synopsys – режим доступу до ресурсу: <https://www.synopsys.com/glossary/what-is-software-architecture.html> (Дата звернення – 03.05.2022)

15) Creately [Електронний ресурс] / Веб сайт Creately – режим доступу до ресурсу: <https://creately.com/blog/diagrams/use-case-diagram-tutorial/> (Дата звернення – 03.05.2022)

16) Communication [Електронний ресурс] / Веб сайт Communication – режим доступу до ресурсу: <http://communication.iresearchnet.com/interpersonal-communication/relational-control/> (Дата звернення – 03.05.2022)

17) Codecademy [Електронний ресурс] / Веб сайт Codecademy – режим доступу до ресурсу: <https://www.codecademy.com/article/what-is-rdbms-sql> (Дата звернення – 03.05.2022)

18) Tutorialspoint [Електронний ресурс] / Веб сайт Tutorialspoint –

| | | | | | | | |
|------|------|----------|--------|------|--|-----------------------|------|
| | | | | | | ДПІПЗ.180119.01.13.ВД | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | | 55 |

режим доступу до ресурсу:
https://www.tutorialspoint.com/uml/uml_class_diagram.htm (Дата звернення – 03.05.2022)

19) Realpython [Електронний ресурс] / Веб сайт Realpython – режим доступу до ресурсу: <https://realpython.com/python-mysql/> (Дата звернення – 03.05.2022)

20) IBM [Електронний ресурс] / Веб сайт IBM – режим доступу до ресурсу: <https://www.ibm.com/topics/software-testing> (Дата звернення – 03.05.2022)

21) The Economic Times Tutorialspoint [Електронний ресурс] / Веб сайт The Economic Times – режим доступу до ресурсу: <https://economictimes.indiatimes.com/definition/software-testing> (Дата звернення – 03.05.2022)

| | | | | | | |
|------|------|----------|--------|------|-----------------------|------|
| | | | | | | Арк. |
| | | | | | | 56 |
| Змн. | Арк. | № докум. | Підпис | Дата | ДПІПЗ.180119.01.13.ВД | |

ДОДАТОК А
(обов'язковий)

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



ДОДАТОК Б
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Опираючись на призначення бота, у нього буде свій набір команд, за допомогою яких ви зможете включати різні його функції.

Згодом його функцію можна змінити під будь-яку фірму чи будь-яке інше завдання.

Існують найрізноманітніші корисні боти для чатів Telegram:

- чат-боти для Telegram, які публікують новини, курси валют, прогноз погоди;
- різні боти-довідники, словники, довідники, анонси заходів;
- боти, які періодично розміщують добірки іноземних слів для вивчення, цитати, фотографії, цікаві посилання і інший розважальний контент;
- корисні боти, які допомагають із завантаженням чого-небудь з інтернету, з пошуком книг, фільмів, квитків, різні нагадування і т.д .;
- бот автопродажів в Telegram, WhatsApp, Instagram, боти зі списками і каталогами товарів або послуг, оповіщенням про акції.

1 Ціль розробки

Існують найрізноманітніші корисні боти для чатів Telegram:

- чат-боти для Telegram, які публікують новини, курси валют, прогноз погоди;
- різні боти-довідники, словники, довідники, анонси заходів;
- боти, які періодично розміщують добірки іноземних слів для вивчення, цитати, фотографії, цікаві посилання і інший розважальний контент;
- корисні боти, які допомагають із завантаженням чого-небудь з інтернету, з пошуком книг, фільмів, квитків, різні нагадування і т.д .;
- бот автопродажів в Telegram, WhatsApp, Instagram, боти зі

списками і каталогами товарів або послуг, оповіщенням про акції.

2 Призначення застосунку

Чат-бот в Telegram може виявитися дуже корисний, якщо ваша робота вимагає постійних недовгих діалогів з клієнтами. Він забезпечить додатковий і зручний канал зв'язку для клієнтів, допоможе в організації воронки продажів і сервісного обслуговування. Даний Чат-бот стане в пригоді для щоденного використання. Користувачем цього бота може стати кожна звичайна людина.

Попередньо налаштовувати чат-бот непотрібно, покрокова інструкція буде міститися на початку користування ботом.

3 Вимоги до сервісу

Список користувацьких перспектив, які має надавати чат-бот:

- надавати користувачу конкретну інформацію;
- вільний вибір щодо запитань до бота;
- можливість перепроверити відповідь бота;
- зберігати історію повідомлень;
- звертатись до чату в будь-який зручний час для користувача;
- доступність чат-боту для кожного користувача;
- кожного дня користувачу надається оновлена інформація;
- легкий доступ з будь-якого смартфона, де зареєстрований ваш номер телефону у Telegram.

Умова до інтерфейсу:

- кожен користувач має змогу змінити гаму кольорів під себе
- подібний інтерфейс має більшість месенджерів, це допомагає швидко адаптуватись.
- бот керується через команди, які вписують в поле вводу тексту.

4 Вимоги до надійності

Чат-бот у Telegram має відповідати наступним вимогам до надійності:

- кожен крок який здійснює користувач фіксується і без дозволу самого користувача видалити неможливо;
- бот відповідає на конкретні команди, які були в ньому запрограмовані.

Ось декілька інструментів, які використовують QA-спеціалісти QATestLab, для покращення роботи чат-ботів: ChatMapper, Botanalytics, Dimon.

ChatMapper – це інструмент для написання та тестування нелінійних комунікацій у більш складних сферах, таких як: розробка, електронне навчання, реабілітація в надзвичайних ситуаціях, діагностика здоров'я, продажі, обслуговування клієнтів та стратегічне планування.

Насправді це візуальний редактор, який дозволяє створювати діалоги та керувати скриптами.

Botanalytics – сервіс, який дозволяє протестувати кожен ключовий аспект чат-бота: від юзабіліті до ведення розмови.

Dimon – відмінною рисою цього інструменту тестування чат-ботів є те, що він легко інтегрується з основними платформами, такими як Telegram, Slack, WeChat, Facebook Messenger. Цей інструмент під час тестування охоплює весь функціонал вашого чат-боту.

5 Умови експлуатації

Якщо ви поглянете на технічні характеристики смартфонів, то побачите такі цифри:

- температура при експлуатації: від 0 до 35 °C (від 32° до 95° F);

- температура при збереженні: від -20 до 45 °С (от -4° до 113° F);
- відносна вологість: від 5 до 95% без конденсації;
- висота при експлуатації: досліджено до 3000 м (до 10 000 футів).

Введення об'єкта(комп'ютера чи ноутбука) в експлуатацію оформляють типовими формами № ОЗ-1, № ОЗ-6.

6 Вимоги до інформаційної та програмної сумісності

Telegram раціонально працює на пристроях, які мають версію: Android 4.1 і вище, iOS 9.0 і вище, Android 4.1 і вище, Windows Phone 8.1 і вище, Windows 7 і вище, macOS 10.10 і вище, Linux, macOS 10.11 і вище, Google Chrome і Chrome OS, Веббраузер, Windows 10, Windows 10 Mobile, Linux, FreeBSD і macOS.

Розмір передавання файлів обмежується до 2 Гб.

7 Стадії та етапи роботи

- Аналіз роботи: пошук аналогічних сервісів та дослідження позитивних та негативних сторін;
- створення ескізного проекту;
- визначення методів та проектування системи залежно від закладених
 - потреб сервісу;
 - написання програмного продукту;
 - тестування програмного продукту;
 - доопрацювання сервісу;
 - перевірка на відповідність сервісу до заданих завдань;
 - супровід програмного продукту.

8 Порядок контролю та приймання

Ціль випробувань – перевірка програмного застосунку на відповідність до поставлених вимог та загальної праці системи.

- перевірка наявності елементів які були наголошені у вимогах.
- перевірка авторизації та аутентифікації;
- перевірка коректності відображення інтерфейсу;
- перевірка адаптації інтерфейсу під різні розміри екранів;
- перевірка коректності праці логіки програми;
- перевірка системи безпеки програмного продукту.

ДОДАТОК В
(обов'язковий)

ПРОГРАМНИЙ КОД

bot_telegram.py

```
from aiogram.utils import executor
from data_base import sqlite_db
from create_bot import dp
from handlers import client, admin
async def on_startup(_):
    print('БОТ ВИЙШОВ У ОНЛАЙН!')
    sqlite_db.sql_start()
client.register_handlers_client(dp)
admin.register_handlers_admin(dp)
executor.start_polling(dp, skip_updates=True, on_startup=on_startup)
```

config.py

```
TOKEN = "5393866023:AAFh15YD2HfcP8qlCGisk3sxqjz7jB6QoJ0"
```

create_bot.py

```
from aiogram import Bot
from aiogram.dispatcher import Dispatcher
from aiogram.contrib.fsm_storage.memory import MemoryStorage
import config
storage = MemoryStorage()
bot = Bot(token=config.TOKEN)
dp = Dispatcher(bot, storage=storage)
```

init.py

```
from handlers import admin
from handlers import client
```

admin.py

```
from aiogram import types, Dispatcher
from create_bot import dp, bot
from data_base import sqlite_db
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import State, StatesGroup
from aiogram.dispatcher.filters import Text
class FSMAdmin (StatesGroup):
    photo = State()
    k_s = State()
    description = State()
    size = State()
#Отримуємо ID поточного модератора
# @dp.message_handler(commands=['moderator'], is_chat_admin=True)
async def make_changes_command (message: types.Message):
    await sqlite_db.sql_audit_user_(message)
    #global ID
    #ID = message.from_user.id
    await bot.send_message(message.from_user.id, sqlite_db.moderator_statys)
#Початок діалогу загрузки нового пункту меню
# @dp.message_handler (commands='Завантажити', state=None) async def cm
start(message: types.Message):
async def cm_start (message: types.Message):
    try:
```

```

if message.from_user.id == sqlite_db.ID:
    await FSMAdmin.photo.set()
    await message.reply('Загрузи картинку \n\nВИ ПЕРЕДУМАЛИ
РОБИТИ\nЗАПИС, НАТИСНІТЬ \n-/abolition')
except:
    print('[INFO 1] cm_start_')
    await message.answer(f'{message.from_user.username}.\nПерейдіть
перевірку статусу модератора \nнатисніть\n- /moderator")
    finally:
        print(f'[INFO 2] cm_start')
        #await message.reply('Спочатку авторизуйся - /moderator')
# Вихід із станів
# @dp.message_handler(state="*", commands="Відміна")
# @dp.message_handler (Text(equals='Відміна', ignore_case=True), state="*")
async def cancel_handler(message: types.Message, state: FSMContext):
    if message.from_user.id == sqlite_db.ID:
        current_state = await state.get_state()
        if current_state is None:
            return
        await state.finish()
        await message.reply('OK')
#Ловимо першу відповідь і записуємо в довідник if message.from_user.id
ID:
# @dp.message_handler (content_types=['photo'], state=FSMAdmin.photo)
async def load_photo(message: types.Message, state: FSMContext):
    if message.from_user.id == sqlite_db.ID:
        async with state.proxy() as data:
            data['photo'] = message.photo[0].file_id

```

```

        await FSMAdmin.next()
        await message.reply("Введи ключове слово (або фразу) по якому
отримувати інформацію на гороскоп\n\nВИ ПЕРЕДУМАЛИ
РОБИТИ\nЗАПИС, НАТИСНІТЬ\n-/abolition")
#Ловимо другу відповідь
# @dp.message_handler(state=FSMAdmin.name)
async def load_name(message: types.Message, state: FSMContext):
    if message.from_user.id == sqlite_db.ID:
        async with state.proxy() as data:
            data['k_s'] = message.text
            await FSMAdmin.next()
            await message.reply("Введи інформацію яку отримає користувач\n\nВИ
ПЕРЕДУМАЛИ РОБИТИ\nЗАПИС, НАТИСНІТЬ\n-/abolition")
#Ловимо третю відповідь
# @dp.message_handler (state=FSMAdmin.description)
async def load_description(message: types.Message, state: FSMContext):
    if message.from_user.id == sqlite_db.ID:
        async with state.proxy() as data:
            data['description'] = message.text
            await FSMAdmin.next()
            await message.reply("Вкажи цифру за яким прорядковим номером знак
зодіаку\n\nВИ ПЕРЕДУМАЛИ РОБИТИ\nЗАПИС, НАТИСНІТЬ\n-
/abolition")
#Ловимо останню відповідь і використовуємо отримані дані
# @dp.message_handler(state=FSMAdmin.price)
async def load_price(message: types.Message, state: FSMContext):
    if message.from_user.id == sqlite_db.ID:
        async with state.proxy() as data:

```

```

        data['size'] = float(message.text)
        await sqlite_db.sql_add_command(state)
        await state.finish()
        await message.reply("Ви все записали.")
    async def menu_command(message: types.Message):
        await sqlite_db.sql_read(message)
#Регіструємо хендлери
    def register_handlers_admin(dp: Dispatcher):
        dp.register_message_handler(cm_start, commands=['download'], state=None)
        dp.register_message_handler(cancel_handler, state="*",
        commands='abolition')
        dp.register_message_handler(cancel_handler, Text(equals='abolition',
        ignore_case=True), state="*")
        dp.register_message_handler(load_photo, content_types=['photo'],
        state=FSMAdmin.photo)
        dp.register_message_handler(load_name, state=FSMAdmin.k_s)
        dp.register_message_handler(load_description, state=FSMAdmin.description)
        dp.register_message_handler(load_price, state=FSMAdmin.size)
        dp.register_message_handler(make_changes_command,
        commands=['moderator'])
        dp.register_message_handler(menu_command)

```

client.py

```

from aiogram import types, Dispatcher
from create_bot import dp, bot
from data_base import sqlite_db
    async def command_open(message: types.Message):

```

```

await sqlite_db.sql_add_user_(message)
await message.answer(f"Привіт {message.from_user.username}!!!\nВас
вітає гороскоп.\nВведіть знак зодіаку який Вас цікавить. Наприклад: Лев. "
                    f"\nПерейти у режим модератора \n- /moderator")
def register_handlers_client(dp: Dispatcher):
    dp.register_message_handler(command_open, commands=['start'])

```

init.py

```

from data_base import sqlite_db

```

sqlite_db.py

```

import sqlite3 as sq
from create_bot import bot
def sql_start():
    global base, cur
    base=sq.connect('baze_menu.db')
    cur=base.cursor()
    if base:
        print('ПІДКЛЮЧЕННЯ ДО БАЗИ ДАННИХ ВДАЛЕ!')
        base.execute('CREATE TABLE IF NOT EXISTS horoscope (img TEXT, k_s
TEXT PRIMARY KEY, description TEXT, size TEXT)')
        base.execute('CREATE TABLE IF NOT EXISTS table_user (user_id
INTEGER PRIMARY KEY,user_name TEXT, clef_ INTEGER)')
        base.commit()
    async def sql_add_command(state):
        async with state.proxy() as data:

```

```

        cur.execute('INSERT INTO horoscope VALUES (?, ?, ?, ?)',
tuple(data.values()))
        base.commit()

async def sql_read(message):
    j=message.text
    print(j)
    k=f"SELECT img, description, k_s FROM horoscope WHERE k_s='{j}'"
    try:
        if cur.execute(k).fetchmany(1):
            for ret in cur.execute(k).fetchmany(1):
                await bot.send_photo(message.from_user.id, ret[0],f'\n {ret[1]}')
        else:
            await bot.send_message(message.from_user.id, "Можливо, з часом, ми
знайдемо для Вас відповідь але не наразі.")
    except:
        await bot.send_message(message.from_user.id, "На жаль, тут ми безсилі
Вам щось повідомити.")
    finally:
        print('[INFO 0] sql_read')
async def sql_add_user_(message):
    user_id=message.from_user.id
    username=message.from_user.username
    k_user = f"SELECT user_id, clef_ FROM table_user WHERE
user_id='{user_id}'"
    print(cur.execute(k_user).fetchall())
    result_k_user=cur.execute(k_user).fetchone()
    try:

```

```

print('[INFO 0] sql_add_user_')
if not result_k_user:
    print('[INFO 00] sql_add_user_')
    cur.execute('INSERT INTO table_user VALUES (?, ?, ?)', (user_id,
username, 1))
    base.commit()
except:
    print('[INFO 1] sql_add_user_')
finally:
    print(f'[INFO 2] sql_add_user_')
async def sql_audit_user_(message):
    user_id=message.from_user.id
    username=message.from_user.username
    k_user = f"SELECT user_id, clef_ FROM table_user WHERE
user_id='{user_id}'"
    result_k_user = cur.execute(k_user).fetchone()
    if result_k_user[-1]==2:
        global ID, moderator_statys
        ID = result_k_user[0]
        moderator_statys=f'ЗАВАНТАЖИТИ -/download \nВИ ПЕРЕДУМАЛИ
РОБИТИ\nЗАПИС, НАТИСНІТЬ \n-/abolition'
        print(f'[INFO 8] Ви модератор')
    elif not result_k_user[-1]==2:
        moderator_statys=f'Ви не модератор.'
        print(f'[INFO 9] Ви не модератор')

```

ДОДАТОК Г
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ДИПЛОМНА РОБОТА НА ТЕМУ: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПЕРЕДБАЧЕНЬ ЗА ГОРОСКОПОМ З РЕАЛІЗАЦІЄЮ ІНТЕРФЕЙСУ У ВИГЛЯДІ TELEGRAM-БОТА

Виконала: Собкова Ю. В.
Керівник: Бедратюк Г. І.

МЕТА ПРОЕКТУ

Метою даного проекту було створити зручне програмне забезпечення для користування на кожен день.

Поставлені завдання на дипломне проектування:

- Розглянути предметну область даного продукту, виявити потреби клієнтів та ціль користування даним механізмом.
- Побудувати структуру та базу даних програми.
- Обрати метод проектування програмного забезпечення
- Провести дослідження на коректність програми.

АКТУАЛЬНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Telegram являється одним з найбільш популярніших месенджерів, тому ми не можемо залишити без уваги такий для бізнесу напрямок, як створення і налаштування чат-бота Telegram.

Зараз у месенджері представлені сотні тисяч каналів на різні теми: новини, історія, психологія, мистецтво, наука, освіта, маркетинг та багато іншого.

На каналах можна читати поради з SMM, стежити за знижками на авіаквитки або просто підписатися на канал зі смішними картинками

ВСТУП

Головна ідея та ціль представленого програмного продукту є створення Telegram бота.

- Обстеження та розгляд зіставлень основних моделей і етапів створення ботів;
- Розбір головних функціональних особливостей;
- Побудова інтерфейсу чата і створення витриманої структури;
- Випробовування Telegram-бота;

ЕТАПИ СТВОРЕННЯ

- створення блок-схеми проекту та технічного завдання,
- Створення функціонального чат-бот для telegram

Приклади типів чат-ботів для різних сфер бізнесу і відповідних месенджерів, які найбільш підходять для даного типу:

- Корпоративний чат-бот,
- Боти для сфери послуг,
- Боти для інтернет-магазинів,
- Боти для маркетингу

ПРОЦЕС РОЗРОБКИ БОТА, СКЛАДАЄТЬСЯ З ТРЬОХ ЕТАПІВ

- Експертиза та аналіз бізнес-процесів
- Створення та затвердження ТЗ /блок схеми
- Розробка бота і системи управління

Тільки дотримуючись даної схеми етапності розробки бота, Ви отримаєте той продукт, який виведе Ваш бізнес на новий рівень.

ЗМІСТ ПРОЕКТУ

- Вивчення процесів, які належать автоматизації, визначення кінцевої мети чату;
- Побудова структури бота, блок-схеми;
- Аналіз вимог і створення інтерфейсу, функціональних елементів, інформаційного вмісту;
- Уточнення вимог для створення чату в відповідності до результатів проведених аналізів наявних платформ

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Так як бот в Телеграмі створити досить просто, то їх є досить велика кількість щоб ви могли знайти чат-бот який підходить вам найбільше

Пропоную продивитися найбільш відомі боти з гороскопом.

- @tophoroscopebot
- @vsegudbot
- @zodiakbot
- @gadaniebot

ГРАФІЧНИЙ І ІНФОРМАЦІЙНИЙ ДИЗАЙН

Структура дизайну буде залежати від того, яким способом ви створите бота. Від простішого до складнішого:

- Ехо-бот
- Wikipedia-бот
- Бот з двома віртуальними кнопками
- Бот, який запускає повідомлення автоматично
- Бот, який може вести розмову з користувачем

СПОСОБИ СТВОРЕННЯ

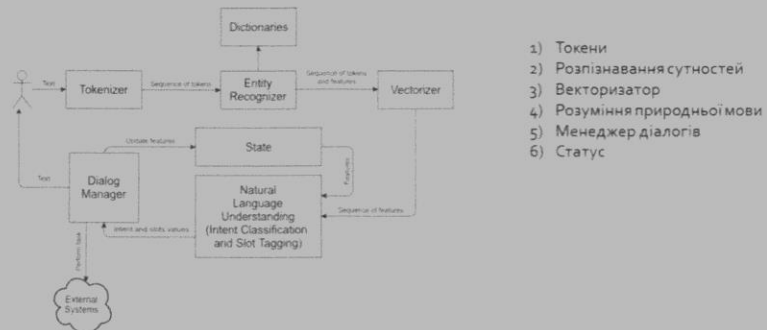
Для написання коду самостійно потрібно знати мови програмування (Python, PHP або Java), а також є ще безліч нюансів типу реєстрації хостингу та іншого. Інший варіант створення – просто скористатися спеціальним сервісом:

Manybot.io

Саме на ньому можна знайти інтерфейс з потрібною мовою, зрозумілу інструкцію та потрібний на перших етапах функціонал. Роботи на цій платформі вміють:

- надсилати повідомлення передплатникам,
- робити гарне меню,
- проводити автопостинг із RSS ваших ресурсів.

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ І КОНТЕКСТНА МОДЕЛЬ НУЛЬОВОГО РІВНЯ



- 1) Токени
- 2) Розпізнавання сутностей
- 3) Векторизатор
- 4) Розуміння природньої мови
- 5) Менеджер діалогів
- 6) Статус

ВИСНОВОК

Впродовж опрацювання дипломної роботи ми проаналізували існуючі Телеграм-боти за допомогою сучасної мови програмування Python. Покращили метод розробки чатів та освіжили дизайн для кращого сприймання боту користувачем.

Телеграм-ботом можна користуватись в будь який момент часу і змінювати його під себе.

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.
здобувача вищої освіти

Собкова Ю. В.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-18-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозиторії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

06.06.2022

Дата



Підпис



Ім'я користувача:
Кафедра ІПЗ

Дата перевірки:
05.06.2022 13:45:04 EEST

Дата звіту:
05.06.2022 13:52:24 EEST

ID перевірки:
1011464601

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005589

Назва документа: Дипломна робота Собкова Юлія1

Кількість сторінок: 58 Кількість слів: 10116 Кількість символів: 79530 Розмір файлу: 2.77 MB ID файлу: 1011342821

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.2%

Схожість

Найбільша схожість: 5.23% з джерелом з Бібліотеки (ID файлу: 1011231415)

3.66% Джерела з Інтернету

122

Сторінка 60

7.61% Джерела з Бібліотеки

98

Сторінка 61

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

20

Підозріле форматування

11
сторінок

Sun Jun 05 12:50:48 EEST 2022, Хіврич Володимир Русланович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 6.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 10%

| | | | | |
|---|----------|---------|-----------------------------|---------|
| ID: 104484 Назва: Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота Додано в БД: 2022-06-05 Автора: Ю. В. Собкова Керівники: Г. І. Бедратюк Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 57547 | 871 | 4177 (7%) | 51 (6%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ
освітнього ступеня «Бакалавр»Дипломник Собкова Юлія ВасилівнаТема Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-ботаСпеціальність 121 – Інженерія програмного забезпечення**Обсяг дипломного проекту:**

Кількість листів креслень _____ ; кількість сторінок записки _____

1. Короткий зміст пояснювальної записки та прийнятих рішень У дипломному проекті було досліджено і проаналізовано предметну область, усі функціональні та не функціональні вимоги. Було проведено аналіз наявного програмного забезпечення. За результатами аналізу було розроблено програмне зєбпечення, яке доводить свою актуальність на сьогоднішній час. Розробка відбувалась з використанням найсучасніших інструментів для розробки інтернет-платформ. Також було проведено ручне тестування, за результати, якого видно, що програмне забезпечення працює відповідно до вимог дипломного проекту.

2. Висновок про відповідність проекту поставленому завданню Дипломна робота освітнього ступеня «бакалавр» відповідає у повній мірі поставленому завданню, як в практичній частині так і в теоретичній частині роботи.

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено, що дана тема актуальна. У першому розділі дипломної роботи аналізується предметна область та аналіз наявного програмного забезпечення, які допомогли визначити завдання та цілі проекту, також у даному розділі описуються вимоги до програмного забезпечення. У другому розділі аналізуються та впроваджуються вимоги, проектується архітектура програмного забезпечення. У третьому розділі проведені всі можливі етапи програмної реалізації, підготовленні всі можливі залежності, описано їх особливості, в результаті чого був написаний програмний продукт. У четвертому розділі виконано ручне тестування програмного забезпечення, що підтверджує працездатність роботи програмного продукту.

4. Позитивні сторони проекту Тематика дипломного проекту досить актуальна на сьогоднішній день, оскільки на сьогоднішній день оскільки всі стаціонарні магазини, які не продають свій товар в інтернеті втрачають дуже велику кількість потенційних клієнтів. Для розробки програмного забезпечення було застосовано сучасні технології розробки . Програмний продукт містить достатній функціонал, щоб задовільнити тему дипломної роботи

5. Негативні сторони проекту У проекті було б доцільно створити форму для замовлення, щоб користувач міг заповнити данні для замовлення, щоб опростили замовлення для клієнта і не потрібно було підтверджувати статус замовлення з клієнтом, також було б доцільно створити меню бистрого пошуку товару.

6. Оцінка графічного оформлення та пояснювальної записки проекту Графічне оформлення виконано відповідно до теми дипломного проекту та подано у вигляді рисунки та діаграм. Пояснювальна записка оформлена згідно усіх вимог та стандартів.

7. Відгук про дипломний проект в цілому Дипломний проект повноцінно описує вирішення всіх задач та завдань. Матеріал чіткий та структурований, який повністю розкриває тему дипломного проекту. Графічний матеріал роботи дає змогу наочно побачити деталі системи.

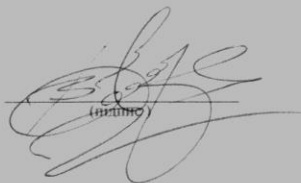
8. Інші зауваження _____

9. Оцінка дипломного проекту Позитивні та негативні сторони дипломного проекту, дає можливість зробити висновок, що дипломний проект виконано в повному обсязі та відповідає поставленій задачі та заслуговує на оцінку «відмінно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи)

Чешун Віктор Миколайович, кандидат технічних наук, доцент кафедри КБ, Хмельницький національний університет.

"01" червня 2022 р.


(підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Програмне забезпечення для передбачень за гороскопом з реалізацією інтерфейсу у вигляді Telegram-бота»

Автор: Собкова Юлія Василівна

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Ганна Іванівна, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідальність |
|---|--|-------------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедрі за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи. | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту(наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 5 | Інше: | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті дипломного проекту системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, бланк завдання на проектування, відомість документів, у структурі змісту, написах в рамках, назвах розділів/підрозділів тощо) та в назвах переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформлені посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлених збігів ідентичності схожості, складає 12.8% і адресується до 324 джерел, що з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь дипломного проекту

Керівник



Г. І. Бедратюк

Гарант ОП



Л. П. Бедратюк

Завідувач кафедри



Л. П. Бедратюк