

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи на
ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА TELEGRAM-БОТУ ДЛЯ ПОШУКУ КУЛІНАРНИХ
РЕЦЕПТІВ МОВОЮ PYTHON»**

Виконав: студент 4 курсу, групи ПД– 44
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

ЯЛОВИК Д.В.

(прізвище та ініціали)

Керівник Негоденко О.В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2023 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА**

Яловик Денис Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка Telegram-боту для пошуку кулінарних рецептів мовою Python»

Керівник роботи: Негоденко Олена Василівна, кандидат технічних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «24» лютого 2023 року №26.

2. Строк подання студентом роботи «01» червня 2023 року

3. Вхідні дані до роботи:

- 3.1 Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки Telegram-додатків.
 - 3.2 Офіційна документація мови Python.
 - 3.3 Офіційна документація VisualStudio Code.
4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).
 - 4.1 Аналіз актуальності та проблематики розроблюваного веб-сервісу.
 - 4.2 Аналіз та вибір інструментів для реалізації продукту.
 - 4.3 Проектування програмного продукту.
 - 4.4 Висновки.
5. Перелік демонстраційного матеріалу (назва основних слайдів)
 - 5.1 Титульний слайд.
 - 5.2 Мета, об'єкт, предмет, наукова новизна дослідження.
 - 5.3 Аналіз аналогів.
 - 5.4 Технічні завдання.
 - 5.5 Програмні засоби та інструменти реалізації.
 - 5.6 Архітектура системи
 - 5.7 Use case та діаграма класів програми
 - 5.8 Приклад застосування
 - 5.9 Висновки.
 - 5.10 Кінцевий слайд.
6. Дата видачі завдання «11» березня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	11.04.2023	Виконано
2	Аналіз потреб та вимог потенційної аудиторії майбутнього продукту	13.04.2023	Виконано
3	Підбір науково-технічної літератури	19.04.2023	Виконано
4	Аналіз ринку існуючих інструментальних засобів, їх функціональних недоліків, недоліків з огляду потреб потенційних користувачів продукту.	20.04.2023	Виконано
5	Аналіз та вибір інструментів для розробки додатку	21.04.2023	Виконано
6	Проектування та реалізація	14.05.2023	Виконано
7	Вступ, висновки, реферат	19.05.2023	Виконано
8	Розробка обов'язкових демонстраційних матеріалів	22.05.2023	Виконано
9	Попередній захист роботи	26.05.2023	
10	Здача роботи	01.06.2023	

Студент _____ Яловик Д.В.
 (підпис) (прізвище та ініціали)

Керівник роботи _____ Негоденко О.В.
 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 59 с., 14 рис., 20 джерел

Ключові слова: Telegram, Python, кулінарія, телеграм-додаток, чат-бот, python-telegram-bot.

Мета роботи — спрощення процедури пошуку рецептів за рахунок Telegram-боту .

Об'єкт дослідження — процес автоматизації процесу пошуку кулінарних рецептів.

Предмет дослідження — Telegram-бот для автоматизації процесу пошуку кулінарних рецептів.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Провести аналіз вимог, потреб, технічних можливостей потенційної аудиторії даного програмного продукту, переваги та недоліки існуючих застосунків;
2. Проаналізувати інструменти та програмні засоби реалізації, що покажуть найкращу продуктивність системи, під час застосування продукту в реальних умовах, та що спричинять найбільшу зручність під час написання продукту;
3. Спроекувати та розробити телеграм-бота для пошуку кулінарних рецептів у месенджері Телеграм;
4. Провести тестування телеграм-бота на виявлення помилок в роботі.

Розроблено систему для пошуку рецептів мовою Python для взаємодії з системою месенджеру Телеграм та можливістю її інтегрування у чат-бота.

ЗМІСТ

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	11
1.1 Загальний аналіз кулінарії.....	11
1.2 Аналіз розробки Телеграм-бота для пошуку кулінарних рецептів.....	29
1.3 Аналіз потреб потенційної цільової аудиторії продукту	33
1.4 Огляд засобів реалізації продукту для створення Telegram-бота для пошуку кулінарних рецептів.	36
РОЗДІЛ 2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	44
2.1 Огляд мови програмування	44
2.2 Огляд середовища розробки	47
РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....	49
3.1 Проєктування діаграми послідовності	49
3.2 Проєктування внутрішньої будови класів	50
3.3 Тестування програмного продукту	56
ВИСНОВКИ.....	64
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
Додаток А.....	68
ПРЕЗЕНТАЦІЯ ДО ЗВІТУ	68

ВСТУП

Актуальність теми. Існує кілька причин, чому розробка Telegram-боту для пошуку кулінарних рецептів мовою Python залишається актуальною:

Завдяки популярності Telegram, який має широку базу користувачів по всьому світу, розробка боту для цієї платформи дозволить залучити велику аудиторію, що активно користується месенджером для комунікації та отримання інформації.

Бот для пошуку кулінарних рецептів в Telegram надає зручний і простий спосіб знаходження рецептів без потреби виходити з додатку або переходити на веб-сайти. Користувачі можуть швидко знайти рецепти за різними критеріями, такими як інгредієнти, кухня, дієтичні обмеження або час приготування.

Мова програмування Python є популярною для розробки ботів, а Telegram надає зручне API та документацію для створення ботів. Використання Python разом з Telegram API дозволяє розробникам швидко створювати і налаштовувати бота для пошуку рецептів.

Python має багату екосистему бібліотек і модулів для обробки тексту, машинного навчання та обробки зображень, які можна використовувати для поліпшення функціональності бота. Наприклад, бібліотеки, такі як NLTK або SpaCy, дозволяють аналізувати текст рецептів або питань користувачів, а TensorFlow або PyTorch можуть використовуватись для створення моделей машинного навчання для персоналізації рекомендацій.

Розробка бота для пошуку кулінарних рецептів в Telegram відкриває шлях до постійного розвитку та інновацій. Розробники можуть експериментувати з новими технологіями, вдосконалювати алгоритми рекомендацій та розширювати функціональність бота для надання ще зручнішого та персоналізованого досвіду користувачам.

Отже, розробка Telegram-боту для пошуку кулінарних рецептів мовою Python є цікавою темою, яка поєднує популярну месенджерську платформу, можливості програмування та потреби користувачів у швидкому та зручному доступі до кулінарних рецептів.

Мета роботи — спрощення процедури пошуку рецептів за рахунок Telegram-боту .

Об'єкт дослідження — процес автоматизації процесу пошуку кулінарних рецептів.

Предмет дослідження — Telegram-бот для автоматизації процесу пошуку кулінарних рецептів.

Методи — методи проектування та розробки програмного забезпечення, методи тестування програмного забезпечення.

Для досягнення цієї мети потрібно виконати такі завдання:

1. Провести аналіз вимог, потреб, технічних можливостей потенційної аудиторії даного програмного продукту, переваги та недоліки існуючих застосунків;
2. Проаналізувати інструменти та програмні засоби реалізації, що покажуть найкращу продуктивність системи, під час застосування продукту в реальних умовах, та що спричинять найбільшу зручність під час написання продукту;
3. Спроекувати та розробити телеграм-бота для пошуку кулінарних рецептів у месенджері Телеграм;
4. Провести тестування телеграм-бота на виявлення помилок в роботі.

Отримані результати матимуть практичне застосування в розробці функціональності для використання API Telegram. Це буде здійснюватись з використанням бібліотеки `python-telegram-bot` та мови програмування Python.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальний аналіз кулінарії

Кулінарія - це сукупність навичок та методів, пов'язаних з готуванням та приготуванням їжі. Це галузь, що має багатовікову історію та змінювалася протягом багатьох століть у різних країнах та культурах. Вона відображає культурні, географічні, кліматичні та історичні особливості різних регіонів, і кожна кухня має свої унікальні риси.

Походження кулінарії не є повністю зрозумілим. Існують припущення, що перші люди випадково спробували смажене м'ясо, коли м'ясо загиблої тварини виявилось більш смачним і легкозасвоюваним, ніж сире. Проте, ймовірно, вони не мали наміру готувати їжу, доки не засвоїли використання вогню для отримання світла і тепла. Хоча є припущення, що пекінець смажив м'ясо, але немає чітких доказів, що підтверджують цю теорію. Проте, смаження на вогнищі залишалося основною кулінарною технікою до періоду палеоліту, коли народ Оріньяка з почав готувати їжу на пару над гарячим вугіллям, використовуючи вологе листя. Крім таких примітивних способів, як підсмажування диких зерен на плоских скелях і використання раковин, черепів або видовбаних каменів для нагрівання рідини, жодних значних кулінарних вдосконалень не було досягнуто до появи кераміки в період неоліту.

Найдавнішою відомою стравою була сира паста, яка була виготовлена шляхом змішування води з подрібненими ядрами диких трав. Ця паста була підсмажена до утворення скоринки, коли її поклали на розпечений камін. Таким чином було створено перший хліб, який став прототипом пульментуму римських легіонів і поленти пізніших італійців.



Рисунок 1.1.1 – Приготування їжі, ілюстративно.

Загальний аналіз кулінарії можна розглядати з кількох аспектів:

- **Інгредієнти:** Різні країни та регіони використовують різні набори інгредієнтів, залежно від доступності та культурних традицій. Використання свіжих продуктів, спецій і трав, м'яса, риби, овочів, фруктів та зернових - це невід'ємна частина кулінарії. Також значну роль відіграє використання традиційних інгредієнтів, які характерні саме для певної кухні.
- **Техніки готування:** Існує безліч технік готування, включаючи варіння, жування, смаження, запікання, тушкування, випікання, копчення, маринування та багато інших. Кожна кухня має свої унікальні техніки та прийоми готування, які використовуються для досягнення певного смаку, текстури та аромату страв.
- **Страви:** Різноманітність страв є ключовою частиною кулінарії. Вони можуть бути простими або складними, використовувати різні інгредієнти та бути прикладами кулінарного мистецтва. У кожній кухні можна знайти свої типові страви, такі як паста у італійській кухні, суші в японській кухні, тако у мексиканській кухні, борщ у українській кухні і так далі.

- Культурні особливості: Кулінарія відображає культурні особливості та традиції кожної країни і регіону. Вона може бути пов'язана з релігійними віруваннями, сезонними святами, сімейними традиціями та способом життя. Їжа може мати символічне значення і відігравати важливу роль у соціальних зустрічах та святкуваннях.

Кулінарія - це захоплююча галузь, що має безліч цікавих фактів і особливостей. Ось кілька захопливих відомостей про кулінарію:

Світові кухні: Кожна країна має свою унікальну кухню з характерними стравами та традиціями. Французька кухня відома своїми складними техніками та делікатними смаками, італійська кухня славиться своїми пастами та піццями, а японська кухня відома своєю пристрастю до свіжих інгредієнтів та суші. Світова кухня включає в себе широкий спектр кулінарних традицій і стилів приготування їжі з різних країн і культур. Кожна національна кухня має свої унікальні особливості, і вона може бути представлена в різних регіонах світу завдяки міграції і впливу глобалізації.

Італійська кухня: Вона відома своїми пастами, піццями, сиром, оливковою олією та використанням свіжих інгредієнтів. Традиційні італійські страви включають лазанью, ризотто, тірамісу та гелато.



Рисунок 1.1.1 – Приклад італійської кухні.

Французька кухня: Французька кухня відома своїм вишуканим стилем і використанням складних технік приготування. Вона включає в себе страви, такі як булон, ескарго, фуа-гра, рататуй та крем-брюле. Французька кухня вважається однією з найвищих форм кулінарного мистецтва і має величезний вплив на гастрономічну культуру світу. Вона відома своєю вишуканістю, розкішшю та різноманітністю страв. Французька кухня базується на використанні свіжих і якісних інгредієнтів, майстерному використанні технік приготування та зіграності смаків.

Ось кілька типових страв французької кухні:

Рататуй: Рататуй - це страва з тушкованих овочів, таких як баклажани, кабачки, перці, цибуля, часник та томати. Вони пасують разом, щоб створити багатий смак та аромат. Це одна з традиційних вегетаріанських страв французької кухні.



Рисунок 1.1.2 – Рататуй, відома французька страва.

Ескаго: Ескаго - це смажений шпажкою кусочок м'яса, зазвичай говядина або ягнятина, який може бути приправлений різними спеціями і травами. Він має ніжний смак і подається з різноманітними соусами.

Фуа-гра: Фуа-гра - це делікатесна страва, приготована з печени гуся або качки, яка була годувана за допомогою техніки форсування. Вона має багатий і насичений смак і часто подається як перед-або основна страва у вишуканих ресторанах.

Булабес: Булабес - це традиційний французький суп з цибулі, який має багато версій. Він готується шляхом плавлення на маслі нарізаної цибулі до золотистого кольору, а потім запікання з вершковим сиром або груюєром на верху.

Класичні випічка та десерти: Французька кухня славиться своїми випічкою та десертами, такими як круасани, еклери, макарони, кремові торти та креми брюле. Ці страви мають високу майстерність і деталізацію у приготуванні.

Крім того, Франція також відома своїми сирами, винами, морепродуктами та великим вибором сезонних страв, які використовують найсвіжіші продукти, що доступні. Французька кухня надає особливу увагу деталям, презентації та зіграності смаків, що робить її унікальною та визнаною у всьому світі.

Іспанська кухня: Іспанська кухня відома своїми тапасами (невеликі страви), паельєю (рис з морепродуктами), хамоном (іспанська шинка) та сангрією (традиційний напій з винограду та фруктів). Іспанська кухня вважається однією з найвпливовіших і популярних кухонь у світі. Вона багата на смаки, аромати та регіональні варіації. Іспанська кухня відображає багато різних культурних, географічних і історичних впливів, що робить її дуже різноманітною.

Це лише деякі приклади страв іспанської кухні:

Паелля - це традиційна іспанська страва, що включає в себе рис, м'ясо (зазвичай курку або креветки), овочі, спеції та бульйон. Це популярна страва з регіону Валенсія, існують різні варіації паеллі, такі як паелля з морепродуктами, паелля з куркою та ковбасою, або вегетаріанська паелля з різними овочами.



Рисунок 1.1.3 – Паелля (рис з морепродуктами), іспанська кухня.

Газпачо - це традиційний іспанський холодний суп, приготований з томатів, огірків, перців, цибулі, часнику, оливкової олії, оцту та спецій. Він зазвичай подається охолодженим і має освіжаючий смак.

Тапас - це асорті страв, що подаються у барах і тавернах Іспанії як закуски. Тапас можуть включати різноманітність інгредієнтів, таких як шинка, оливки, сир, креветки, крокети, печена овочі, крокети та багато інших. Вони ідеально підходять для ділової зустрічі або дружнього зібрання, а також є чудовим доповненням до вина чи пива.

Хамон - це сушений іспанський шинкований свинячий окіст, який вважається дуже цінним і вишуканим продуктом. Іспанці вдівають його на тонкі скибки і подають як закуску або частину тапас.

Пататас Бравас - це традиційна іспанська страва з печених картоплі, приправлених гострим соусом або соусом айолі. Це популярна закуска, яка може бути супроводжена різними соусами та приправами.

Іспанія також відома своїми сирами (наприклад, манчего), морепродуктами (наприклад, паелья з морепродуктами), десертами (такими як крем каталан або чуррос) та винами, які часто супроводжують страви.

Японська кухня: Японська кухня відома своєю чистотою, естетикою та акцентом на свіжість та натуральний смак інгредієнтів. Вона має багато різновидів страв, але основні компоненти включають рис, рибу, морепродукти, сою, водорості та свіжі овочі.

Ось кілька типових страв японської кухні:

Суші: Це, можливо, найвідоміша страва японської кухні. Суші складається зі свіжих сирих або маринованих риб, морепродуктів, омлету або овочів, розміщених на рисовій основі із додаванням васабі (паста з хрону), соєвого соусу та імбиру.

Сашимі: Це тонко нарізані шматочки свіжої риби або морепродуктів, які подаються без готування. Сашимі часто приправляють соєвим соусом та васабі

Рамен: Це популярний японський суп, в якому основною складовою є нудлі в бульйоні. Рамен може містити різні типи м'яса (як правило, свинину або курку), морепродукти, овочі та інші додатки.

Теппан'які: Це страви, приготовані на великій плоскій пательні (теппані) перед очима гостей. Зазвичай це м'ясо (наприклад, яловичина або курка), морепродукти та овочі, які смажаться разом з соєвим соусом на високій температурі.

Удон: Це товсті пшеничні нудлі, які часто подаються у супі з бульйоном на основі сої або рибного фонду. Удон можуть бути прикрашені різними інгредієнтами, такими як товчений яйця, м'ясо або овочі.

Крім цих страв, в японській кухні використовуються також водорості (наприклад, норі для ролів), місо супи, японські омлети (тамаго), зелений чай та багато інших страв і напоїв.

Важливим аспектом японської кухні є також пристань до естетики та краси подачі страв, де акцент робиться на привабливому вигляді, деталізації та балансі кольорів та текстур.



Рисунок 1.1.4 – Японська кухня, ілюстрація.

Індійська кухня: Індійська кухня відома своїми складними спеціями та ароматами. Вона включає в себе каррі, тандурі, бірйані, даля та наан(тонкий хліб). Індійська кухня вважається однією з найрізноманітніших і найсмачніших кухонь у світі. Вона відображає багатогранність культур, регіонів та традицій Індії. Індійська кухня характеризується використанням ароматних спецій, свіжих трав і різноманітних інгредієнтів.

Наан - це популярний індійський хліб, який печеться у тандурі або на сковороді. Він має м'яку і пухку текстуру та часто подається з каррі або далею.

Індійська кухня також відома своїми солодощами, наприклад, гулаб-джамун (солодкі м'ячики у сиропі), джалєбі (крихкі пиріжки у сиропі) та кульфі (індійське морозиво). Кожен регіон Індії має свої унікальні страви та смакові предпочтєння, що робить індійську кухню дуже різноманітною і захоплюючою.

Китайська кухня: Китайська кухня є однією з найстаріших та найрозповсюдженіших кухонь у світі. Вона має багато регіональних варіацій, кожна з яких має свої унікальні особливості та страви. Основні компоненти китайської кухні - це рис, пшениця, соя, овочі, м'ясо та морепродукти.

Ось деякі типові страви китайської кухні:

Пекінська качка: Це одна з найвідоміших страв китайської кухні. Качку запікають у спеціальній печі, доки шкірка стає хрусткою, а м'ясо ніжним. Подається з пляцками, печеними огірками, цибулею та соусом.

Гуляш: Це м'ясна страва, в якій шматочки м'яса тушкуються разом з овочами у соєвому соусі. Гуляш може бути приготовлений зі свинини, яловичини, курки або морепродуктів.

Вок: Це швидкозастосовувана страва, приготована на великій сковороді з високими бортиками. Інгредієнти, такі як м'ясо, овочі та соуси, швидко смажаться на вогні, щоб зберегти текстуру та свіжість.

Димлені страви: Китайська кухня також відома своїми димленими стравами, такими як димлений чай, димлена риба або димлена качка. Цей процес надає стравам характерного смаку та аромату.

Суші: Хоча суші є популярною стравою японської кухні, вони також широко розповсюджені в Китаї. Суші складаються зі свіжих риби, морепродуктів, омлету або овочів, розміщених на рисовій основі.

Крім цього, в китайській кухні використовуються багато спецій, таких як соя, імбир, часник, кинза, сім'янка кунжуту та багато інших, що надають стравам характерного смаку.



Рисунок 1.1.6 – Пекінська качка, відома на весь світ страва китайської кухні.

Це лише кілька прикладів світових кухонь, але насправді їх багато і кожна має свої унікальні страви, техніки приготування та інгредієнти. Кулінарні традиції можуть різнитися в залежності від регіону, історії, клімату та культурних впливів.

Спеції: Спеції - це різноманітні ароматичні речовини, які використовуються для приготування їжі, надаючи їй аромат, смак і колір. Спеції можуть бути походженням рослинного, тваринного або мінерального походження. Спеції використовуються для надання смаку та аромату стравам. Кардамон, кориця, куркума, кумин, імбир та інші спеції використовуються в різних кухнях світу, щоб додати характерний смак та запах.

Деякі з найпоширеніших спецій включають:

- Чорний перець: одна з найпопулярніших спецій, використовується як приправа до багатьох страв.
- Куркума: жовто-помаранчевий порошок, який надає стравам яскравий колір і характерний смак. Він часто використовується в каррі і індійській кухні.
- Коріандр: насіння коріандру використовують як спецію, а листя - як зелень. Він має легкий цитрусовий аромат і використовується у багатьох кухнях світу.
- Кориця: солодкий і ароматний приправа, яка використовується як додаток до десертів, кави і інших страв.
- Імбир: гострий і ароматний корінь, використовується як приправа і ліки. Він широко використовується в азіатській кухні.
- Кардамон: має сильний аромат і використовується як приправа в каррі, десертах і чаї.
- Кмин: використовується в багатьох кухнях для приготування м'ясних страв і хлібобулочних виробів.
- Часник: має гострий смак і аромат і використовується в багатьох стравах, включаючи соуси, маринади і супи.
- Паприка: має гострий або солодкий смак і використовується як приправа і для надання стравам червоного кольору.
- Куркума: жовта спеція з гострим і пряним смаком, часто використовується в індійській кухні для надання смаку і колору стравам.

Кожна спеція має свій власний унікальний смак, аромат і застосування.

Фаст-фуд: Фаст-фуд є популярним видом харчування в багатьох країнах. Це швидкі та зручні страви, які готуються та подаються швидко. Гамбургери, піца, картопля фрі та курячі крильця - це лише кілька прикладів популярних фаст-фуд страв.



Рисунок 1.1.7 – МакДональдз, одна з найбільших у світі мережа закладів швидкого харчування.

Фаст-фуд - це швидка їжа, яка готується і подається в швидкому темпі. Вона зазвичай доступна у вигляді готових страв, які можна швидко придбати в ресторанах фаст-фуду або через виготовлення на місці.

Основні риси фаст-фуду включають:

- Швидкість: одна з головних характеристик фаст-фуду полягає у швидкому обслуговуванні. Страви готуються швидко і майже завжди готові для негайного споживання.
- Стандартизація: фаст-фудові ланцюги використовують стандартизовані рецепти та процеси готування для забезпечення послідовності і якості продуктів.
- Простота: багато страв фаст-фуду прості в приготуванні та подачі. Це можуть бути бургери, хот-доги, піца, картошка фрі, сендвічі та інші.

- Низька вартість: фаст-фудові страви часто відносно дешеві порівняно з іншими видами ресторанного харчування. Це одна з причин популярності фаст-фуду, особливо серед швидкого та бюджетного харчування.
- Широкий вибір: фаст-фудові ресторани зазвичай пропонують широкий асортимент страв і напоїв, що відповідає смакам та вподобанням різних людей.

Проте, важливо враховувати, що фаст-фуд може бути висококалорійним і містити велику кількість насичених жирів, холестеролу та солі. Часте споживання фаст-фуду може бути пов'язане з ризиком розвитку ожиріння, серцево-судинних захворювань та інших проблем зі здоров'ям. Тому важливо вживати його з помірком та збалансованої харчової системи.

Солодощі: Солодощі - це вироби, які мають солодкий смак і зазвичай споживаються як десерти або ласощі. Вони можуть бути виготовлені з різних інгредієнтів, таких як цукор, шоколад, фрукти, мед, молоко, мука та інші. Солодощі мають особливе місце в світі кулінарії. Торти, печиво, шоколад, морозиво та багато інших солодощів задовольняють наші солодкі зуби. Кожна кухня має свої унікальні солодкі страви, які варіюються від країни до країни.

Деякі з найпопулярніших видів солодощів включають:

- Шоколад: це один з найулюбленіших видів солодощів у багатьох країнах. Шоколад може бути темним, молочним або білим і використовується для приготування шоколадних батончиків, цукерок, тортів та інших солодких виробів.
- Тістечка та пироги: це випічка, яка може бути солодкою та надана різноманітними начинками, які можуть бути фруктовими, кремовими, шоколадними, мигдалевими тощо.
- Морозиво: холодне десертне лакомство, яке виготовляється з молока або вершків, цукру та інших інгредієнтів. Морозиво може бути в різних смаках та текстурах.

- Мармелад та желе: ці м'які і желеобразні солодощі виготовляються з фруктового пюре, цукру та желатину. Вони можуть мати різні форми і смаки.
- Льодяники та карамель: цукерки, які можуть бути твердими, м'якими, клейкими або хрусткими. Вони виготовляються з цукру, глюкози та інших інгредієнтів та можуть бути різних смаків і кольорів.
- Печиво та пироги: різні види печива, включаючи шоколадне печиво, печиво з горіхами, м'які пироги з фруктами або кремом.
- Цукерки: різноманітні цукерки, включаючи шоколадні цукерки, льодяники, карамелі, карамельні та маршмеллоу.

Важливо поміркувати про споживання солодощів та включати їх у раціон з розумінням і поміркованістю для збалансованого харчування.

Здорове харчування: Здорове харчування - це підходящий спосіб харчування, який забезпечує нашому організму необхідну кількість поживних речовин, вітамінів і мінералів, зменшує ризик розвитку захворювань і сприяє загальному здоров'ю. Зростаюча увага до здорового способу життя призводить до зростання популярності здорового харчування. Люди все більше звертають увагу на свої харчові звички, включаючи вегетаріанство, веганство та безглютенову дієту. Здорові інгредієнти, свіжі продукти та збалансована харчова піраміда стають основою цього підходу до харчування.

Здорове харчування не тільки поліпшує фізичне здоров'я, але й має позитивний вплив на настрій, енергетику та загальний стан організму. Консультація з дієтологом або харчовими спеціалістами може допомогти визначити індивідуальні потреби і розробити план харчування, що відповідає вашим потребам і цілям.

Кулінарні шоу та конкурси: Кулінарні шоу та конкурси стали популярними по всьому світу. Шоу, такі як "MasterChef", "Top Chef" та "The Great British Bake Off", пропонують пекарям та кухарям показати свої навички і творчість на кулінарній арені.



Рисунок 1.1.8 – МастерШеф (Україна), найпопулярніше кулінарне шоу в Україні.

Культурні обміни: Кулінарні обміни - це процес обміну кулінарними знаннями, рецептами, техніками та ідеями між людьми з різних культур і кухонь. Це може відбуватися через особисті зустрічі, подорожі, кулінарні курси, книги, інтернет-форуми або соціальні мережі. Кулінарні обміни між країнами та культурами є важливою частиною глобалізації. Ресторани з екзотичною кухнею, фузійні страви та інгредієнти з різних кухонь стають доступними в різних куточках світу.

Кулінарні обміни мають декілька переваг:

- Розширення гастрономічного досвіду: обмін кулінарними знаннями дозволяє вивчати нові страви, інгредієнти і способи приготування з різних культур. Це допомагає розширити гастрономічний репертуар і спробувати нові смаки.
- Взаємний культурний обмін: кулінарні обміни сприяють взаємному розумінню і відкриттю інших культур через кухню. Вони дають можливість

познайомитися зі звичаями, традиціями, історією та способом життя інших народів через харчування.

- Творчість і інновації: кулінарні обміни можуть спонукати до творчого підходу до готування. Вони надають можливість поєднувати різні кулінарні традиції, експериментувати зі смаками, створювати нові версії традиційних страв і відкривати нові кулінарні комбінації.
- Соціальна взаємодія: кулінарні обміни створюють можливість для соціальної взаємодії, спілкування і спільного готування. Вони сприяють побудові зв'язків, обміну історіями та створенню незабутніх спогадів.

Під час кулінарних обмінів важливо проявляти повагу до культурних особливостей і традицій інших народів. Розуміння правильного використання інгредієнтів, технік приготування та повага до кулінарних звичаїв і обрядів є ключовими аспектами успішних кулінарних обмінів.

Використання традиційних методів готування: Незважаючи на сучасний прогрес, багато кухонь у світі продовжують використовувати традиційні методи готування. Пароварки, тандири, димлення, варіння в паєльї та багато інших технік стали несамоовито популярними через свою унікальність та смакові переваги. Використання традиційних методів готування є чудовим способом збереження культурної спадщини та насолоди кулінарними традиціями. Традиційні методи готування базуються на довголітньому досвіді і використовуються для створення аутентичних страв з виразним смаком та текстурою. Деякі з найпоширеніших традиційних методів готування включають:

- Приготування на вогні: цей метод включає приготування страв на відкритому вогні, будь то на грилі, барбекю або багатті. Він додає особливого смаку та аромату стравам і часто застосовується для м'яса, риби, овочів і печива.
- Припекання або запікання: цей метод використовується для створення страв, які готуються в печі або духовці. Він дозволяє створювати хрустку скоринку

на м'ясі, пісні або запіканках, а також розподіляти рівномірне тепло для добре пропеченого продукту.

- Тушкування: цей метод передбачає довготривале тушкування продуктів у власному соку або рідині. Він забезпечує м'якість і соковитість м'яса, а також інтенсивний смак і аромат у страві. Тушкування використовується для готування рагу, гуляшу, традиційних страв у соте та багатьох інших.
- Ферментація: цей метод використовується для збереження та надання стравам особливого смаку і текстури. Він включає використання мікроорганізмів, таких як дріжджі або молочнокислі бактерії, для перетворення інгредієнтів. Ферментація використовується для приготування хліба, сиру, капусти та інших продуктів.
- Смаження: цей метод включає готування їжі у гарячому оливі або іншому жирі на плиті або у фритюрі. Він додає хрустку скоринку та аромат до страви і часто використовується для приготування картоплі фрі, котлет, риби та інших страв.

Ці традиційні методи готування не лише надають стравам аутентичного смаку та текстури, але й допомагають зберегти і передати культурні традиції та спадщину. Вони відображають унікальність кухонь різних культур і дозволяють насолоджуватися багатогранністю кулінарних традицій по всьому світу.

Соціальна спільнота кулінарії: Інтернет та соціальні медіа великим чином змінили спосіб, яким ми ділимося своїми кулінарними вподобаннями. Блоги про їжу, YouTube-канали, Instagram-акаунти та кулінарні форуми створюють глобальну спільноту кулінарних ентузіастів, які обмінюються рецептами, фотографіями та порадами.

Загальний аналіз кулінарії показує, наскільки різноманітним та багатогранним є це мистецтво. Кулінарія не тільки забезпечує нас їжею, але також відображає наші історію, культуру та ідентичність. Вивчення та

експериментування з кулінарією може бути захоплюючим способом розширити свої знання про світ і насолодитися смаками різних культур.

1.2 Аналіз розробки Телеграм-бота для пошуку кулінарних рецептів

Розробка Телеграм-бота для пошуку кулінарних рецептів мовою Python може бути цікавим і корисним проєктом з багатьма можливостями.

Використання бібліотеки `python-telegram-bot` для зручного з'єднання з Telegram API.

Бібліотека `python-telegram-bot` є популярним і потужним інструментом для роботи з Telegram API в мові Python. Вона надає зручний і простий спосіб з'єднання з Telegram API та взаємодії з ботом. Ось кілька кроків для використання цієї бібліотеки:

- Встановлення бібліотеки:

Встановлюємо бібліотеку `python-telegram-bot`, використовуючи пакетний менеджер `pip`, за допомогою наступної команди:

```
pip install python-telegram-bot
```

- Імпорт бібліотеки та створення об'єкта бота:

Імпортуємо необхідні класи та функції з бібліотеки:

```
import telegram
```

```
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters
```

- Створення об'єкта `telegram.Bot` з використанням API-ключа:

```
bot = telegram.Bot(token='YOUR_API_KEY')
```

- Налаштування обробників повідомлень:

Визначаємо функції-обробники для різних типів повідомлень, таких як команди, текстові повідомлення, фотографії, тощо. Наприклад, створимо функцію-обробник для команди `/start`:

```
def start(update, context):
```

```
context.bot.send_message(chat_id=update.effective_chat.id, text='Hello!
Welcome to the recipe bot.')
```

```
# Створення об'єкта обробника команди '/start'
```

```
start_handler = CommandHandler('start', start)
```

- Налаштування обробника повідомлень та запуск бота:

Створюємо об'єкт telegram.ext.Updater і передаємо йому бота. Додаємо об'єкти обробників до об'єкта Updater, а потім запускаємо бота:

```
updater = Updater(bot=bot, use_context=True)
```

```
dispatcher = updater.dispatcher
```

```
# Додавання об'єкта обробника команди '/start' до диспетчера
```

```
dispatcher.add_handler(start_handler)
```

```
# Запуск бота
```

```
updater.start_polling()
```

- Обробка повідомлень:

Додаємо функції-обробники для різних типів повідомлень та виконуємо потрібні дії в цих функціях. Наприклад, для обробки текстових повідомлень:

```
def handle_text(update, context):
```

```
    text = update.message.text
```

```
    # Виконати потрібні дії з текстом повідомлення
```

```
    text_handler = MessageHandler(Filters.text, handle_text)
```

```
    dispatcher.add_handler(text_handler)
```

- Відправлення повідомлень:

Використовуємо методи об'єкта telegram.Bot для відправлення повідомлень користувачам. Наприклад, для відправлення текстового повідомлення:

```
bot.send_message(chat_id=chat_id, text='Hello, this is a message from the bot.')
```

Можна використовувати різні методи, такі як send_photo(), send_document(), send_location(), тощо, для відправлення різних типів повідомлень.

Це загальний огляд використання бібліотеки `python-telegram-bot` для зручної взаємодії з Telegram API. Бібліотека надає багато інших можливостей, таких як робота з клавіатурами, кнопками, розсилкою повідомлень, отриманням фотографій, тощо. Документація та приклади використання доступні на офіційному сайті бібліотеки `python-telegram-bot`.

Налаштування та реєстрація нового бота в Telegram через BotFather для отримання API-ключа.

Щоб налаштувати та зареєструвати нового бота в Telegram через BotFather і отримати API-ключ, слід виконати наступні кроки:

Встановити Telegram на своєму пристрої та відкрити його.

Знайти BotFather:

- У полі пошуку в Telegram вводимо "BotFather".
- Вибираємо офіційний обліковий запис BotFather з підтвердженою галочкою.

Створення нового бота:

- У полі чату з BotFather натискаємо кнопку "Start".
- Вводимо команду `/newbot` для створення нового бота.
- Слідуюємо інструкціям BotFather.

Надаємо ім'я та ідентифікатор бота:

- Вводимо ім'я свого бота. Це ім'я буде видно користувачам.
- Вводимо унікальний ідентифікатор бота. Це повинно закінчуватися на "bot" (наприклад, "RecipeBot").

Отримуємо API-ключ:

- Після успішного створення бота BotFather надішле нам повідомлення, що містить API-ключ розробленого бота.
- Збережіть цей ключ. Він потрібний для з'єднання вашого бота з Telegram API.

Налаштовуємо інші параметри бота:

- Опціонально, ми можемо налаштувати інші параметри свого бота через BotFather, такі як опис, фотографія профілю, команда старту, дозвіл на приєднання до групи тощо.

Збір даних про рецепти:

- Вибір підходящого джерела даних про кулінарні рецепти, такого як Spoonacular, Edamam або власна база даних.
- Використання API цих джерел або написання власних парсерів для отримання даних про рецепти.
- Збереження отриманих даних у внутрішній базі даних або в пам'яті для подальшого використання.

Реалізація функціоналу пошуку рецептів:

- Розробка логіки для пошуку рецептів за ключовими словами, інгредієнтами, типом страви або іншими параметрами.
- Фільтрація результатів пошуку залежно від потреб користувача.
- Виведення списку рецептів з необхідною інформацією, такою як назва, складники та інструкції.

Навігація та взаємодія з користувачем:

- Додавання команд для взаємодії з ботом, наприклад, пошук (/search), випадковий рецепт (/random), додавання до улюблених, тощо.
- Реалізація логіки для навігації між сторінками результатів та подробицями рецептів.
- Обробка команд та повідомлень від користувача, забезпечення зручного інтерфейсу для взаємодії.

Додатковий функціонал:

- Реалізація можливості зберігання улюблених рецептів користувача.
- Додавання функції списку покупок для зручного збереження складників, необхідних для приготування обраних рецептів.

- Можливість поділитися рецептом з іншими користувачами або додати його до власного списку.

Розгортання та вдосконалення:

- Розгортання бота на вибраному хостингу або хмарному сервісі для забезпечення постійного доступу.
- Покращення функціоналу бота, додавання нових функцій або інтеграція з іншими сервісами.
- Використання логів та аналізу поведінки користувачів для вдосконалення функціоналу та забезпечення кращої взаємодії.

Важливо також враховувати деталі реалізації, безпеку, обробку помилок та оптимізацію швидкості взаємодії з джерелами даних. Залежно від вашої креативності та вимог, можна додати багато цікавого функціоналу до такого бота.

1.3 Аналіз потреб потенційної цільової аудиторії продукту

Основною категорією цільової аудиторії користувачів для створення Telegram-боту з пошуку кулінарних рецептів є друзі, кулінари-аматори, родичі, які хочуть набути кулінарних навичок, або просто побалувати себе апетитними стравами. Під час аналізу аудиторії було виявлено тенденції думок та потреб потенційних користувачів. Ці потреби було розділено на декілька категорій:

- доступність;
- технічні можливості;
- кулінарна складова;
- ціль пошуку страв.

Кулінарна складова в телеграм-боті є цікавою та корисною функцією, яка може привернути увагу шанувальників кулінарного мистецтва та полегшити їх кулінарний досвід. Додавання кулінарної складової до телеграм-бота має декілька переваг:

- **Рецепти:** Надання рецептів у телеграм-боті дозволяє користувачам отримувати широкий вибір страв та інсайти щодо їх приготування. Рецепти можуть бути категоризовані, що полегшує пошук та вибір страв для готування.
- **Планування харчування:** Функція планування харчування допомагає користувачам створювати раціони харчування, враховуючи їхні дієтичні вимоги та вподобання. Це забезпечує організованість та здорове харчування.
- **Списки покупок:** Надання списків необхідних інгредієнтів для рецептів та можливість створення списків покупок у телеграм-боті полегшує процес закупівлі продуктів та планування походу до супермаркету.
- **Поради та підказки:** Надання корисних порад та підказок щодо кулінарних технік, збереження продуктів та покращення страв додає цінність до кулінарної складової. Відповіді на запитання користувачів та підтримка також є важливою частиною цієї функції.
- **Відгуки та рейтинги:** Можливість залишати відгуки та оцінювати рецепти, ресторани або продукти дозволяє користувачам обмінюватися своїми думками та рекомендаціями, створюючи кулінарну спільноту в телеграм-боті.

Кулінарна складова додає велику цінність до телеграм-бота, допомагаючи користувачам знайти рецепти, спланувати своє харчування, отримати корисні поради та ділитися досвідом з іншими. Вона створює зручну та інтерактивну платформу для всіх, хто зацікавлений у кулінарному мистецтві.

- Технічні можливості телеграм-бота для пошуку кулінарних рецептів можуть включати наступні аспекти: Взаємодія з базою даних: Бот може мати доступ до бази даних, в якій зберігаються рецепти. Це дозволяє швидкий та ефективний пошук рецептів за назвою, категорією, інгредієнтами та іншими параметрами.

- Пошук за ключовими словами: Бот може мати функцію пошуку, яка дозволяє користувачам шукати рецепти за ключовими словами. Наприклад, користувач може ввести "паста з лососем" і отримати список рецептів, пов'язаних з цією стравою.
- Фільтри та категорії: Бот може надавати можливість вибору рецептів за певними фільтрами та категоріями. Наприклад, користувач може вибрати "десерти" або "вегетаріанські страви" для отримання відповідних рецептів.
- Рекомендації та випадковий вибір: Бот може мати функцію рекомендацій, де він пропонує користувачам певні рецепти на основі їхніх попередніх виборів або враховуючи популярність страв. Крім цього, бот може пропонувати випадковий вибір рецептів для тих, хто хоче спробувати щось нове.
- Деталі рецептів: Бот може надавати деталізовану інформацію про кожен рецепт, включаючи список інгредієнтів, кроки готування, час приготування, складність та інші деталі. Це допомагає користувачам зрозуміти, чи вони зацікавлені у конкретному рецепті.
- Зображення та відео: Бот може підтримувати зображення та відео, що допомагають візуально представити рецепт. Користувачі можуть бачити фотографії готових страв або відео з процесом готування.
- Коментарі та оцінки: Бот може дозволяти користувачам залишати коментарі та оцінювати рецепти. Це створює можливість для спілкування та обміну думками між користувачами, а також допомагає визначити популярність та якість рецепту.

Важливо розробити зручний та інтуїтивно зрозумілий інтерфейс, щоб користувачі могли легко знаходити та використовувати бот для своїх потреб.

Іншим параметром для вибору платформи була доступність. Одною з потреб стала можливість роботи додатку на всіх можливих платформах, можливість пошуку рецептів навіть на природі. Це означає, що сайт повинен бути доступним

у місцях, де немає доступу до Інтернету. Доступність також означає, що сайтом можна було користуватися з застарілих пристроїв. Тобто, додаток, що має бути сумісним з різними операційними системами.

У ході аналізу потреб користувачів, одним із пунктів опитування було запитання про можливість приготування страв. Також із наявних пунктів було про можливість приготування страв з наявних удома продуктів, редагування списку наявних продуктів, додавання інгредієнтів способом відправки фотографії боту.

1.4 Огляд засобів реалізації продукту для створення Telegram-бота для пошуку кулінарних рецептів.

Розробка телеграм-бота для пошуку кулінарних рецептів орієнтована на мобільні платформи, такі як смартфони, і також на настільні і портативні комп'ютери. Вибір мови програмування для розробки продукту залежить від типу додатку або додатків, які планується створити. Розглянемо деякі сучасні види додатків для різних операційних систем.:

Нативні додатки – це програми, розроблені для конкретної платформи (наприклад, iOS або Android) та використовують мови програмування та інструментарій, які підтримуються цією платформою.

Web-додатки – це програма, яка розробляється з використанням веб-технологій, таких як HTML, CSS та JavaScript, і запускається у веб-браузері. Ці додатки потребують наявності Інтернет-з'єднання для своєї роботи. Оскільки всі сучасні операційні системи підтримують Інтернет-браузер, існує велика ймовірність, що такі додатки будуть працювати на всіх популярних платформах..

Гібридні додатки – це додатки, які поєднують риси веб-додатків і нативних додатків. Розробляється з використанням веб-технологій, таких як HTML, CSS та JavaScript, і запускаються внутрішньо у веб-переглядачі, який вбудований у нативний контейнер. Крім того, є можливість завантажити ці додатки з локального

магазину застосунків і використовувати їх без необхідності у підключенні до Інтернету.

Інші додатки – додатки, які розроблені для використання на готовій платформі. Вони можуть бути плагінами для популярних програм або повноцінними програмами, які працюють на серверах і взаємодіють з платформами, на яких їх використовують.

Таблиця 1.4.1 – Оцінювання потреб користувачів та вибір додатків на основі проведеного аналізу типів додатків.

Варіант	Необхідність у завантаженні	Процесинг на сервері	Мультиплатформеність	Необхідність оновлення додатку після програмних змін
Web-додаток	-	+	+	-
Нативний додаток	+	-	-	-
Гібридний додаток	+	-	+	+
Telegram бот	-	+	+	-

У даному продукті буде використана найсучасніша технологія розробки додатків. Він буде базуватися на платформі Telegram, яка доступна на різних платформах. Оскільки Telegram відомий своєю низькою вимогою до інтернет-з'єднання і вже встановлений на більшості молодіжних мобільних пристроїв в Україні, він ідеально підходить для потенційної аудиторії. Ця платформа також

надає можливість спілкуватися з автоматизованими допоміжними чат-ботами, розробленими спільнотою. Для написання серверної частини для Telegram-ботів була обрана мова програмування Python, яка є надзвичайно функціональною та легкою для використання.

Telegram є популярним месенджером та платформою для обміну повідомленнями, яка надає широкий спектр функцій та можливостей. Він працює на різних пристроях і операційних системах, включаючи мобільні телефони, планшети і комп'ютери. Telegram пропонує безпечне та шифроване спілкування, можливість створювати групові чати, обмінюватися фотографіями та відео, надсилати голосові повідомлення, створювати канали для публічних сповіщень, використовувати ботів та багато іншого. Крім того, Telegram підтримує функцію "конфіденційних чатів", яка дозволяє користувачам встановлювати таймери на повідомлення для їх самознищення. Також відомо, що Telegram має потужне API, що дозволяє розробникам створювати свої власні додатки та ботів для платформи. Загалом, Telegram є популярним серед користувачів з-за своєї широкої функціональності, безпеки та простоти використання.



Рисунок 1.4.1 – Логотип Telegram

Існують клієнтські додатки Telegram, доступні для різних платформ, включаючи настільні і мобільні. Офіційні додатки Telegram доступні для Android, iOS, Windows, macOS і Linux (проте для реєстрації потрібен пристрій з операційною системою iOS або Android та дійсний номер телефону). Крім того,

існують два офіційних веб-додатки Telegram, WebK і WebZ, а також багато неофіційних клієнтів, які використовують протокол Telegram. Всі офіційні компоненти і додатки Telegram мають відкритий вихідний код, за винятком пропрієтарного сервера, який має закритий код.

Telegram також надає додаткову функцію зашифрованих бесід end-to-end, що означає, що повідомлення не зберігаються на серверах Telegram і можуть бути прочитані тільки на двох пристроях: відправником та отримувачем. Хмарні чати і групи шифруються між додатком та сервером, тому постачальники послуг і треті сторони в мережі не можуть отримати доступ до цих даних, хоча сервер Telegram має доступ до них. Користувачі можуть відправляти як текстові, так і голосові повідомлення, проводити голосові та відеодзвінки, а також обмінюватися необмеженою кількістю зображень, документів (розміром до 2 ГБ на файл), локацій користувачів, анімованих і неанімованих стікерів, контактів і аудіофайлів.

Telegram надає загальнодоступне API, розробники можуть отримати доступ до функцій, що присутні у офіційних програмах Telegram, для створення власних програм обміну повідомленнями. Також Telegram пропонує API для створення ботів, які є обліковими записами та можуть бути контрольованими програмами. Також боти використовуються для різних цілей, включаючи емуляцію та відтворення старих ігор у програмі, а також для надання інформації користувачам.

Telegram-боти - це програми, які працюють у месенджері Telegram і використовують його функціонал для взаємодії з користувачами. Вони можуть виконувати різні завдання, такі як надсилання повідомлень, обробка команд, отримання інформації з Інтернету, розрахунки та багато іншого. Розробники можуть створювати свої власні Telegram-боти, використовуючи Telegram Bot API, який надає необхідні інструменти та можливості для створення функціональних інтерактивних ботів. Боти стали дуже популярними серед користувачів Telegram і використовуються для різних цілей, включаючи розваги, ігри, отримання новин, пошук інформації, роботу зі списками, переклади та багато іншого. Вони також

можуть обробляти онлайн-платежі, що здійснюються за допомогою кредитних карт або сервісу Apple Pay. Також існують inline-боти, які можна використовувати безпосередньо з будь-якого екрану чату. Щоб активувати inline-бота, користувачу потрібно ввести ім'я користувача бота та поставити запит у поле повідомлення. Після цього бот надасть доступ до свого вмісту, з якого користувач може вибрати та надіслати його у чаті.

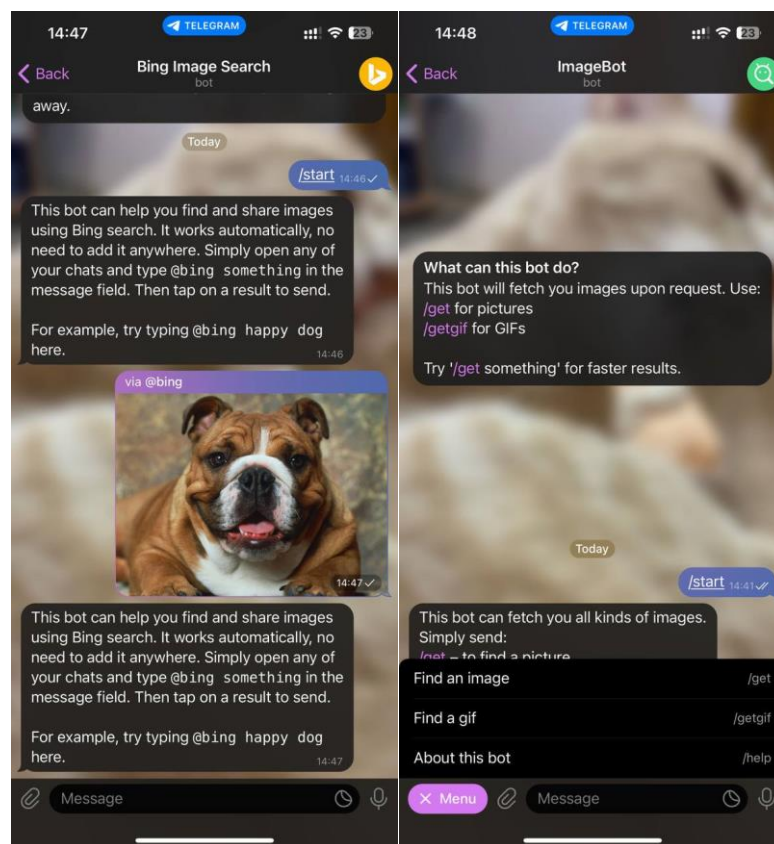


Рисунок 1.4.2 – Приклад інтерфейсу ботів у додатку Telegram

Завдяки інтеграції з IFTTT (If This Then That), у Telegram існує можливість використання сервісів Інтернету речей (IoT), які забезпечують двосторонню взаємодію з користувачами.

У лютому 2018 року Telegram розробив і представив функцію соціального входу, відому як Telegram Login, яка стала доступною для користувачів. Ця

функція включає в себе віджет, що може бути вбудований на веб-сайті, який може бути вставлений на сторонні веб-сайти. Віджет Telegram Login забезпечує передачу імені користувача та зображення профілю власнику веб-сайту, при цьому номер телефону користувача залишається конфіденційним.

Python є однією з найбільш популярних мов програмування в наш час. Згідно з даними популярного ресурсу серед програмістів, Stack Overflow, Python посідає одне з лідируючих місць у списку найбільш популярних та затребуваних мов програмування впродовж останніх п'яти років. Це пояснюється його універсальністю та простотою синтаксису. Python відрізняється мінімальним набором символів, що робить його легким у вивченні та використанні порівняно з іншими мовами програмування. Його універсальність дозволяє використовувати його для різних видів роботи: від написання плагінів та скриптів до розробки ігор, мобільних додатків, онлайн-сервісів та супроводження програмного забезпечення. Python займає визначне місце в світі програмування завдяки своїм багатим можливостям та широкому спектру застосувань

Дослідження використання мов програмування для розробки ботів у месенджері Telegram показали, що Python є найбільш популярною мовою для цієї цілі. Python займає провідну позицію серед інших мов програмування завдяки своїм перевагам та високій популярності серед розробників, що обирають його для створення ботів у Telegram.

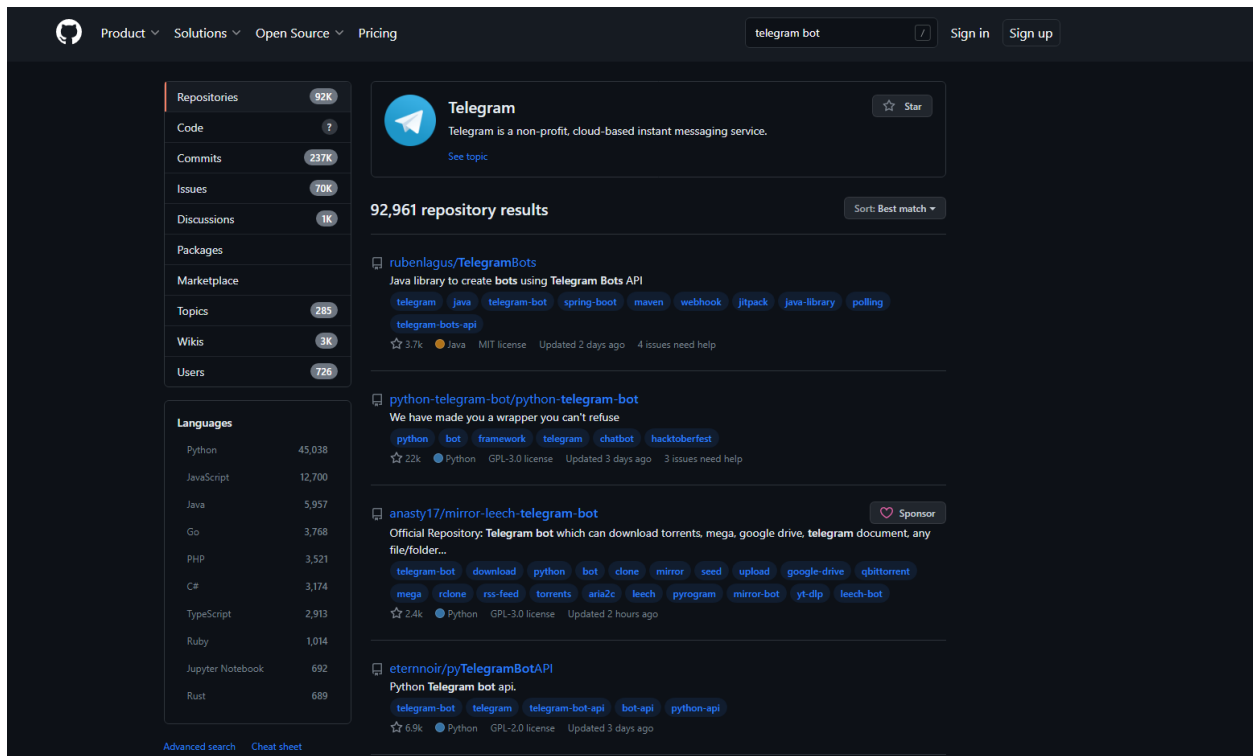


Рисунок 1.4.3 – Запит «Telegram bot» на платформі GitHub.

Щоб забезпечити швидку відповідь сервера на запити від різних користувачів одночасно, потрібно зробити функціонал сервера асинхронним. Одним із способів цього досягнути є використання наступних технологій:

- Технологія "Threading" в Python: Ця технологія полягає в налаштуванні виконання функцій на різних потоках обробки процесора. Це дозволяє розробнику обробляти багато запитів одночасно, забезпечуючи швидку відповідь сервера. Використання потоків дозволяє виконувати операції паралельно, що забезпечує ефективну обробку запитів.
- Бібліотека python-telegram-bot: Ця бібліотека є популярним інструментом для розробки телеграм-ботів на Python. Вона надає зручний інтерфейс для створення функціональності бота та взаємодії з Telegram API. Використання цієї бібліотеки дозволить забезпечити асинхронну обробку запитів від користувачів і швидку відповідь бота.

Таким чином, для забезпечення швидкої відповіді сервера на запити різних користувачів одночасно, можна використовувати технологію "Threading" в Python або бібліотеку `python-telegram-bot`. Обидва варіанти дозволять зробити функціонал сервера асинхронним і забезпечити ефективну обробку запитів.

РОЗДІЛ 2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Огляд мови програмування

Мова програмування Python є відкритою із вихідним кодом та була написана на мові програмування C. Її найпоширенішим вимовлянням є "Пайтон". Назва мови походить від телевізійного шоу "Літаючий цирк Монті Пайтона", і творець мови обрав її на честь цього шоу. У офіційній документації, прикладах та підручниках Python можна знайти велику кількість жартів, пов'язаних з шоу "Літаючий цирк Монті Пайтона".

Python є високорівневою інтерпретованою мовою програмування, яка використовує об'єктно-орієнтований підхід і має строгую динамічну типізацію. Вона була створена Гвідо ван Россумом, голандцем, у 1990 році. Інтерпретована мова програмування означає, що код Python не потребує компіляції перед запуском. Замість цього, інтерпретатор Python може виконувати код на практично будь-якому комп'ютері. Це дозволяє програмістам швидко вносити зміни до коду та бачити результати безпосередньо. Однак, через цю інтерпретацію, Python може бути менш продуктивним порівняно з компільованими мовами, такими як C, оскільки він не виконує машинний код безпосередньо.

Python має декілька ключових переваг, серед яких велика спільнота програмістів і висока швидкість розробки. Наявність великої та активної спільноти програмістів дозволяє легко знайти підтримку, відповіді на питання та навіть готові рішення для різних завдань. Це сприяє ефективній розробці програмного забезпечення та зменшує час, необхідний для вирішення завдань.

Python - це мова програмування високого рівня, що дозволяє програмістам зосередитися на вирішенні завдань, не поглинаючись у деталі апаратного забезпечення. Вона є чудовим вибором для початківців у програмуванні.

Написання програм на Python займає менше часу порівняно з іншими мовами. Python був інспірований такими мовами програмування як C, C++, Java та Perl.

Розробники Python активно працюють над оптимізацією мови, уникаючи незначних змін у несуттєвих частинах довідкової програми CPython. Це дозволяє підвищити швидкість виконання програм, особливо в ситуаціях, коли швидкість є важливою. Крім того, програміст Python може ефективно поєднувати його з іншими мовами, такими як C або PyPy, що перекладають скрипти Python у мову C або використовують JIT-компіляцію.

Python має велику спільноту програмістів, яка активно використовує мову в різних областях. Стандартна бібліотека Python, яка встановлюється разом з мовою, надає багато корисних функцій. Крім того, в Інтернеті доступно безліч сторонніх бібліотек, як офіційних, так і розроблених спільнотою, які розширюють можливості мови Python. Ці бібліотеки роблять Python потужним інструментом, що дозволяє вирішувати різноманітні завдання.

Python знаходить своє застосування у різних галузях програмування. Він використовується для:

- Веб-розробки: Python має популярні фреймворки, такі як Django і Flask, які допомагають розробляти потужні веб-додатки та API.
- Наукового програмування: Python став улюбленим серед науковців і дослідників завдяки багатому набору бібліотек, таких як NumPy, SciPy, Pandas і Matplotlib. Ці бібліотеки допомагають здійснювати обробку числових даних, проводити аналіз даних, робити статистичні обчислення та візуалізацію.
- Розробки програм з графічним інтерфейсом: Python має багато інструментів для створення програм з графічним інтерфейсом, таких як PyQt і Tkinter, що дозволяють створювати інтуїтивно зрозумілі та взаємодіючі програми.

- Мережного програмування: Python має вбудовану підтримку мережного програмування, що дозволяє створювати сервери, клієнти і робити маніпуляції з мережевими протоколами.
- Розробки ігор: Python використовується як мова програмування для розробки ігор, зокрема за допомогою бібліотеки Pygame.
- Розробки складних алгоритмів: Python є популярним вибором для реалізації складних алгоритмів завдяки його простоті та високому рівню абстракції.
- Автоматизації: Python дозволяє автоматизувати повторювані завдання та обробку даних завдяки розширеним можливостям роботи з файловою системою, регулярними виразами та іншими інструментами.
- Розробки аудіо та відео програм: Python має бібліотеки, такі як librosa та OpenCV, які допомагають роботі з аудіо- та відеоданими.

З версії 3.5 Python також підтримує асинхронний код, що дозволяє ефективно обробляти багатопотокові операції, забезпечуючи високу продуктивність та низький час відгуку для веб-додатків.

Однією з вагомих переваг Python є його лаконічна синтаксична конструкція, яка робить код зрозумілим та зручним для написання. Багатий екосистема бібліотек Python також є великим плюсом, оскільки вона надає готові рішення для різних завдань та допомагає прискорити розробку. Завдяки цьому, проекти на Python можуть бути виконані дешевше, швидше та простіше. Ще однією перевагою Python є його широкий набір функцій, який дозволяє вирішувати не тільки типові завдання для веб-додатків, але й реалізовувати унікальні ідеї. Python є відмінним вибором для розробки різноманітних проектів, від веб-додатків до аналізу даних, наукових досліджень, штучного інтелекту та багатьох інших сфер.

Python володіє надзвичайно чистим синтаксисом і лаконічністю. Цей лаконічний та читабельний код стає більш зрозумілим, легше інспектується, виправляється та підтримується. У порівнянні з іншими сучасними мовами програмування, Python виглядає більш сучасним, оскільки код, написаний на

Python, зазвичай значно коротший, ніж код, що виконує ті самі дії, написаний іншими мовами програмування.

2.2 Огляд середовища розробки

Середовище розробки відіграє важливу роль у процесі програмування та написання коду в сучасних умовах розробки програмного забезпечення. Одним із найпопулярніших та зручних середовищ розробки, згідно з опитуванням Stack Overflow 2021 Developer Survey, є засіб, розроблений компанією Майкрософт та відкритою спільнотою.

Цей продукт, який має відкритий вихідний код, є першим у виборі багатьма розробниками та вважається найпопулярнішим середовищем розробки для мови програмування Python. Використання цього середовища сприяє зручному та ефективному процесу розробки, дозволяючи зосередитися на написанні коду. Завдяки активній спільноті розробників і відкритому вихідному коду, цей продукт надає широкий функціонал, розширення та налаштування для зручної роботи з мовою Python.

Варто відзначити, що на ринку також існує ряд інших популярних середовищ розробки для Python, таких як PyCharm, Jupyter Notebook, Spyder та інші. Вибір конкретного середовища розробки залежить від індивідуальних вподобань, проекту та специфічних потреб розробки.

Visual Studio Code (VS Code) — це безкоштовний та відкритий редактор вихідних кодів, розроблений компанією Microsoft. Він призначений для програмістів і надає зручні інструменти для редагування та відлагодження коду. Основними особливостями Visual Studio Code є:

Кросплатформенність: VS Code підтримує роботу на операційних системах Windows, macOS та Linux, що дозволяє розробникам використовувати його на будь-якій платформі.

Розширюваність: Редактор має розширену систему плагінів, яка дозволяє розробникам розширювати його функціональність та додавати нові можливості. Існує велика кількість плагінів, доступних у магазині розширень, які покривають різноманітні потреби розробників.

Підтримка різних мов програмування: VS Code надає вбудовану підтримку для багатьох мов програмування, включаючи JavaScript, Python, C++, Java, HTML, CSS та інші. Редактор надає можливості автодоповнення, підсвічування синтаксису, перевірки помилок та форматування коду для кожної мови.

Інтегрований відлагоджувач: В VS Code є можливість відлагодження коду безпосередньо в редакторі. Розробники можуть встановлювати точки зупинки, крокувати по коду, переглядати значення змінних та аналізувати виконання програми.

Інтеграція з системами контролю версій: Редактор має вбудовану підтримку систем контролю версій, зокрема Git. Розробники можуть легко керувати репозиторіями, виконувати команди Git та вирішувати конфлікти злиття прямо в VS Code.

Зручна робота з завданнями: В редакторі можна налаштовувати та виконувати різні завдання, такі як збирання, тестування або розгортання програм. Це дає змогу автоматизувати рутинні процеси розробки.

Робота в команді: За допомогою розширення Live Share, розробники можуть спільно працювати над проектами, ділитися кодом та відлагоджувати його в реальному часі.

Конфігурація та налаштування: VS Code дозволяє налаштовувати різні параметри редактора, включаючи кольорову схему, шрифти, горячі клавіші та інші налаштування.

Visual Studio Code є популярним вибором серед розробників завдяки своїй широкій функціональності, зручному інтерфейсу та активній спільноті користувачів та розробників.

РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Проєктування діаграми послідовності

Діаграми послідовності використовуються для уточнення прецедентних діаграм і надають більш детальне уявлення про логіку використання. Вони є ефективним засобом документування випадків використання проєкту. Діаграма послідовності відображає взаємодію між елементами моделі програмної системи та порядок передачі повідомлень між ними упорядкований за часом. Вона показує зв'язок та обмін інформацією або керування між елементами.

Ці діаграми широко використовуються в реальних проєктах для переходу від вимог до програмного забезпечення до етапу проєктування. Діаграми послідовності зосереджені на візуалізації послідовності повідомлень у часі, а не на відображенні структурних відносин між процесами. Вони дозволяють детально розробляти взаємодії та послідовності подій в програмній системі.

На діаграмах послідовності можна показати об'єкти, що взаємодіють у сценарії, повідомлення, які вони обмінюють, та результати повернення, пов'язані з цими повідомленнями. Графічне зображення діаграми послідовності використовує горизонтальну вісь для елементів взаємодії, кожен з яких має свою лінію життя. Горизонтальні стрілки з ліній життя вказують на взаємодії. Вертикальна вісь представляє хід часу, де нижче розташовані взаємодії відбуваються пізніше. Траєкторії використовуються для деталізації взаємодій і послідовностей подій.

Діаграми послідовності використовуються для проєктування взаємодій та опису послідовностей подій в мові UML 2.0. Вони можуть бути представлені з різною деталізацією та орієнтовані на розробників або кінцевих користувачів програмної системи. Об'єкти, повідомлення та життєві лінії графічно відображають взаємодії. Для позначення моменту знищення об'єкта використовується символ хреста. Деякі діаграми послідовності можуть містити елементи "Актор", які

представляють взаємодію з користувачем. Прямокутник вздовж лінії життя показує активний період дії.

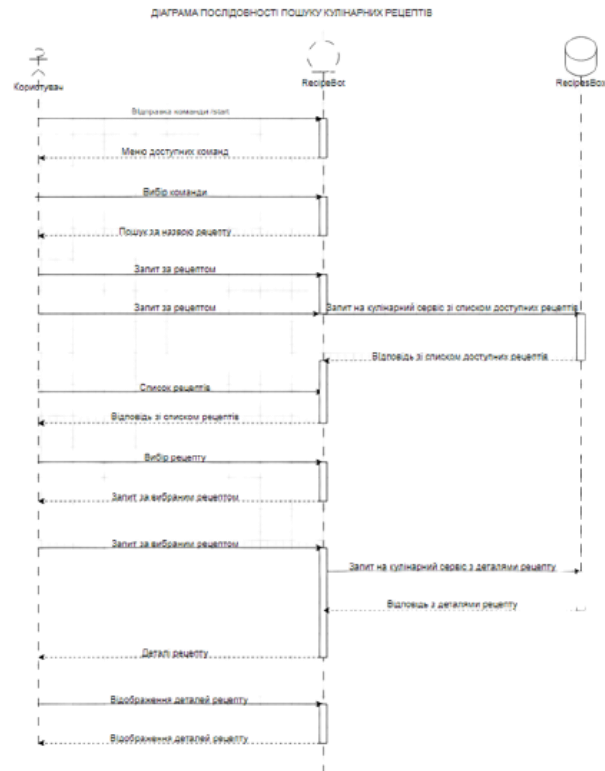


Рисунок 3.1.1 – Діаграма послідовності пошуку кулінарних рецептів.

3.2 Проектування внутрішньої будови класів

Проектування діаграми класів (class diagram) відноситься до процесу візуалізації структури класів і взаємозв'язків між ними у програмному забезпеченні. Діаграма класів є графічним представленням класів, їх полів, методів та залежностей.

Основна мета проектування діаграми класів - це відображення структури програми і взаємозв'язків між класами. На діаграмі класів можна показати такі елементи:

- Класи: Кожен клас зображується у вигляді прямокутника, в якому вказується назва класу.

- Поля: Поля класу (змінні) відображаються у вигляді списку або таблиці всередині прямокутника класу. Кожне поле має свою назву, тип і модифікатор доступу (наприклад, `public`, `private`).
- Методи: Методи класу (функції) також відображаються всередині прямокутника класу. Кожний метод має свою назву, параметри, тип повернення і модифікатор доступу.
- Зв'язки: Зв'язки між класами показуються за допомогою стрілок, які вказують на напрямок залежності. Наприклад, можна показати асоціацію, агрегацію, композицію, успадкування та інші типи зв'язків.
- Коментарі: Діаграма класів може містити коментарі та пояснення, які допомагають краще розуміти структуру програми і зв'язки між класами.

Проектування діаграми класів дозволяє візуалізувати архітектуру програми, з'ясувати залежності між класами, спрощує комунікацію між розробниками та допомагає уникнути помилок при проектуванні програмного забезпечення.

У моделях програмного забезпечення залежність означає, що один елемент або компонент системи використовує або потребує інший елемент або компонент для своєї роботи. Залежність визначається взаємодією між різними частинами системи і може включати в себе виклики методів, передачу параметрів, використання ресурсів або обмін даними. Залежності між елементами в моделях можуть бути представлені різними способами, залежно від типу моделі.

Асоціація - це зв'язок між двома класами або об'єктами в об'єктно-орієнтованому програмуванні. Вона вказує на взаємодію або відношення між цими класами. Асоціація може бути використана для відображення, як один клас використовує функціональність або має доступ до іншого класу.

Асоціація може бути односторонньою або двосторонньою. В односторонній асоціації, один клас використовує функціональність іншого класу без зворотного зв'язку. Наприклад, може бути клас "Клієнт" і клас "Замовлення", де клієнт створює замовлення. Клас "Клієнт" асоціюється з класом "Замовлення".

У двосторонній асоціації, обидва класи можуть використовувати функціональність один одного. Наприклад, може бути клас "Батько" і клас "Дитина", де батько має дитину, а дитина також має батька. Таким чином, обидва класи асоціюються один з одним.

Асоціація може мати атрибути або методи, які використовуються для передачі даних або здійснення дій між класами. Наприклад, в класі "Клієнт" може бути атрибут "ім'я" і метод "зробитиЗамовлення()", які використовуються класом "Замовлення" для доступу до імені клієнта та створення замовлення.

Асоціація є одним з фундаментальних концептів об'єктно-орієнтованого програмування і допомагає побудувати взаємозв'язки та взаємодію між класами для створення більш складних систем.

Агрегація є видом асоціації в об'єктно-орієнтованому програмуванні, який відображає "має-частину" відношення між об'єктами. В агрегації один об'єкт є власником або контейнером для іншого об'єкта, тобто один об'єкт містить посилання на інший об'єкт як свою складову частину.

Агрегація відрізняється від звичайної асоціації тим, що об'єкти можуть існувати незалежно один від одного. Окрім того, об'єкт, який є частиною агрегації, може належати до багатьох агрегацій одночасно. Зв'язок агрегації може бути слабким (некерованим) або сильним (керованим).

Прикладом агрегації може бути клас "Команда", який містить об'єкти класу "Гравець". У цьому випадку "Команда" є власником гравців і може керувати їх додаванням, видаленням, отриманням списку гравців тощо. Гравці можуть бути частиною однієї команди, але також можуть належати до інших команд або існувати самостійно поза командою.

У кодї агрегацію можна виразити за допомогою посилань на об'єкти або за допомогою вкладення об'єктів одного класу в інший. Наприклад, клас "Команда" може мати список об'єктів класу "Гравець", які є її частинами.

Агрегація дозволяє створювати більш складні структури із зв'язками між об'єктами, де один об'єкт виконує роль контейнера або власника для іншого об'єкта. Вона допомагає організувати код і взаємодію між об'єктами в більш систематичний спосіб.

Сукупність є об'єктом, який має семантичне значення і розглядається як один цілісний елемент, незважаючи на те, що фактично складається з кількох менших об'єктів. Це додаткове семантичне значення надається сукупності.

Це означає, що один клас, відомий як підклас, є більш конкретним екземпляром іншого класу, відомого як супертип, тоді як суперклас є більш загальним підкласом. У контексті цього відношення кожен екземпляр підкласу також є екземпляром суперкласу, до якого він належить.

Узагальнення в UML можна показати за допомогою графічного зображення порожнього трикутника на кінці лінії, що з'єднує суперклас з одним або кількома підкласами. Ця форма може бути знайдена в кінці дерева класів.

Узагальнення також може мати назву успадкування або відношення "є". У контексті узагальнень термін "батько" відноситься до суперкласу, також відомого як базовий клас або базовий тип.

Підтип, який є підтипом іншого підтипу відносно спеціалізації, також відомий як "дочірній" підклас, похідний клас або похідний тип, клас успадкування або тип успадкування.

Узагальнення можна відобразити за допомогою діаграм класів і діаграм використання.

У моделюванні UML, зв'язок реалізації використовується для відображення відношення між класами, де один клас (відомий як клас реалізації) реалізує функціональність або інтерфейс, визначений в іншому класі (відомий як клас інтерфейсу або клас-батько).

Зв'язок реалізації вказує, що клас реалізації виконує всі обов'язки та методи, визначені у класі інтерфейсу. Це означає, що клас реалізації надає конкретну

реалізацію методів, визначених у класі інтерфейсу. Клас інтерфейсу визначає набір методів, які повинні бути реалізовані класом реалізації.

Зв'язок реалізації зображується за допомогою стрілки з пустим трикутником на боці класу реалізації, що вказує на клас інтерфейсу. На стрілці може бути позначка «реалізує» або «реалізовано за допомогою», щоб уточнити відношення між класами.

Зв'язок реалізації використовується для показу, що клас реалізує певний контракт або інтерфейс, і забезпечує конкретну реалізацію його методів. Він використовується для забезпечення абстрактного типу даних або інтерфейсу, який може бути реалізований декількома класами. Це сприяє модульності, розширюваності та повторному використанню коду в програмі.

У графічному зображенні реалізації UML використовується трикутник з отвором посередині, який знаходиться в кінці дерева класів або інтерфейсу. Цей трикутник з'єднується пунктирною лінією з одним або кількома реалізаторами. Інтерфейсна пунктирна лінія, що з'єднує його з користувачами, позначається прямою стрілкою для показу зв'язку. На компонентних діаграмах, що використовують "м'яч і розетку" (куля або льодяник для реалізаторів та розетка для користувачів), це умова для відображення реалізацій. Реалізації можна представляти лише на діаграмах класів або компонентів.

Зв'язок між класами, інтерфейсами, компонентами та пакетами, що встановлює зв'язок між елементом замовника та елементом постачальника, називається реалізацією. Відношення реалізації між інтерфейсами та класами або компонентами показує, що клас або компонент дійсно втілюють на практиці операції, які надаються інтерфейсом.

У моделюванні UML, залежність використовується для відображення відношення між двома елементами моделі, де один елемент залежить від іншого елемента у вигляді використання або звернення до його функціональності.

Залежність показує, що зміни в одному елементі можуть впливати на інший елемент, але немає жодного фізичного або структурного зв'язку між ними.

Залежність може бути напрямлена або ненапрямлена. У напрямленій залежності вказується, який елемент залежить від іншого, за допомогою стрілки, що вказує напрямок залежності. Ненапрямлена залежність показує, що обидва елементи взаємозалежні, але не вказує напрямок.

Залежність використовується для показу, що один елемент використовує функціональність або послуги іншого елемента, але без прямого зв'язку або залежності від його структури або реалізації. Наприклад, клас може мати залежність від іншого класу, коли він використовує його методи або властивості для своїх потреб. Інший приклад - залежність між пакетами або модулями програми, де один пакет використовує функціональність або класи з іншого пакету.

Залежність в моделі UML зображується за допомогою пунктирної стрілки, що виходить з одного елемента та має позначку «залежить від» або «використовує». Це допомагає визначити залежності між елементами і розуміти, які елементи можуть бути вплинуті змінами в інших елементах.

У представленні асоціацій в UML, два класи, які з'єднані між собою, показуються за допомогою лінії. На кінцях цієї лінії додаються додаткові символи, щоб передати додаткову інформацію. Наприклад, для показу видимості можна використовувати наконечник стрілки, який вказує, який кінець видимий. Право власності можна позначити за допомогою м'яча, а роль компонентів в асоціації можна назвати, а також вказати кількість зустрічей цих компонентів (множинність).

Класи сутностей, які моделюють довговічну інформацію, що обробляється системою, і, можливо, пов'язану з нею поведінку, відповідають за цю інформацію. Вони не повинні бути сприйняті як таблиці бази даних або будь-який конкретний тип сховища даних. У діаграмах UML вони зображаються у вигляді кола з

короткою лінією, що виходить знизу кола. Альтернативно, їх можна зобразити як звичайні класи зі стереотипом "сутність", що накладається на їх назви.

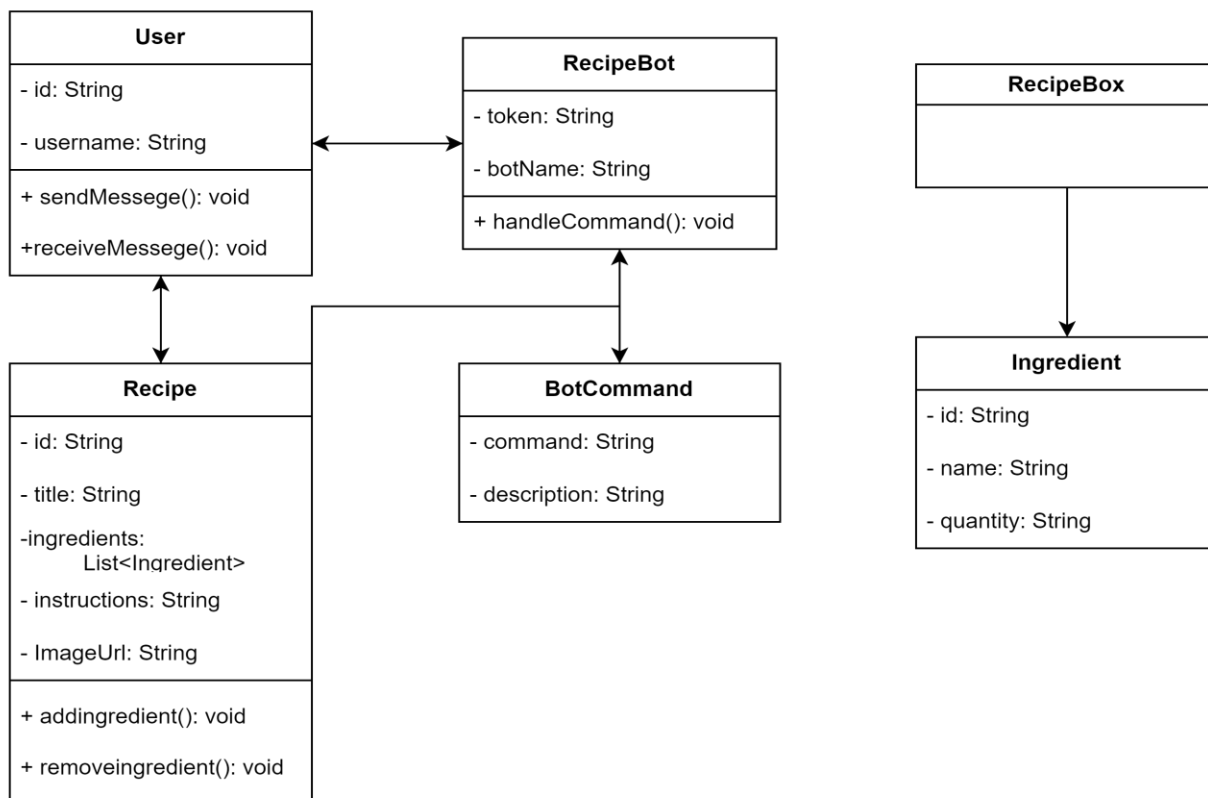


Рисунок 3.2.1 — Діаграма класів Telegram-боту для пошуку кулінарних рецептів

3.3 Тестування програмного продукту

Тестування програмного продукту є важливим етапом в розробці, який має на меті виявлення помилок, перевірку функціональності та забезпечення якості програмного забезпечення. Основні етапи тестування програмного продукту включають:

- **Планування тестування:** На початковому етапі визначаються цілі та стратегія тестування. Планується, які аспекти програмного продукту будуть

тестуватись, які тестові сценарії будуть використовуватись та які ресурси потрібні для проведення тестування.

- Розробка тестових випадків: На основі вимог та функціональності програмного продукту розробляються тестові випадки. Кожен тестовий випадок описує початкові умови, дії користувача та очікувані результати.
- Виконання тестів: Тестові випадки виконуються згідно з планом тестування. Результати тестування фіксуються, включаючи помилки, відхилення від очікуваних результатів та інші спостереження.
- Виявлення та виправлення помилок: Помилки, які були виявлені під час тестування, фіксуються і повідомляються розробникам для виправлення. Після виправлення помилок повторно виконуються відповідні тестові випадки, щоб перевірити, чи були проблеми виправлені.
- Регресійне тестування: Після виправлення помилок проводиться регресійне тестування, щоб переконатися, що внесені зміни не вплинули на раніше протестовані функції.
- Валідація та верифікація: Завершальний етап тестування включає валідацію програмного продукту, перевірку відповідності вимогам та забезпечення, що він працює згідно з очікуваннями користувачів.

Тестування може проводитись як вручну, так і за допомогою автоматизованих інструментів тестування. Важливо забезпечити достатню покриття тестами, включити різні сценарії використання та переконатися, що програмний продукт працює стабільно та забезпечує очікувану функціональність.

Несправності та збої в програмному продукті можуть виникати з різних причин. Деякі з найпоширеніших проблем, які можуть впливати на функціональність та стабільність програми, включають:

- Помилки програмування: Недоліки або помилки в коді програми можуть призводити до непередбачуваних поведінок та збоїв. Це можуть бути синтаксичні помилки, помилки логіки або проблеми з обробкою даних.

- **Неправильні вхідні дані:** Якщо програма не впорається з некоректними вхідними даними, це може призвести до непередбачуваної поведінки або збоїв. Некоректні дані можуть бути введені користувачем або отримані зовнішніми джерелами.
- **Проблеми з мережевим з'єднанням:** Якщо програмний продукт залежить від мережевого з'єднання, проблеми зі з'єднанням, недоступність серверів або повільна швидкість мережі можуть призвести до збоїв або перерв у роботі.
- **Навантаження та ресурси:** Великі обсяги даних, велика кількість користувачів або недостатні ресурси (пам'ять, процесор, мережа) можуть призвести до падінь продуктивності та збоїв.
- **Несумісність з платформою або середовищем:** Програмний продукт може мати проблеми зі сумісністю з операційною системою, бібліотеками, середовищем виконання або залежностями.
- **Неправильна конфігурація:** Несправність може виникати внаслідок неправильної конфігурації програмного продукту, параметрів, налаштувань або залежностей.
- **Безпекові проблеми:** Вразливості в програмному продукті можуть призводити до збоїв або можуть бути використані для зловживання. Це можуть бути проблеми з безпекою даних, аутентифікацією, авторизацією тощо.

Для виявлення та усунення неполадок та збоїв в програмному продукті важливо проводити систематичне тестування, моніторинг роботи програми, аналізувати звіти про помилки та реагувати на них швидко та ефективно. Також, ретельне тестування перед випуском нових версій або оновлень може допомогти уникнути багатьох проблем.

Варто зауважити, що не кожна помилка завжди призводить до збою. Наприклад, мертвий код, тобто недосяжний код, може працювати належним чином кожен раз. Також може виникнути проблема, коли помилка не виявляється і

середовище не змінюється там, де це потрібно. Наприклад, програмне забезпечення може працювати на іншій апаратній платформі комп'ютера, використовувати змінені вихідні дані або взаємодіяти з іншим програмним забезпеченням. Одна помилка може призвести до різноманітних ознак та симптомів відмови системи.

Варто відзначити, що помилки в сирцевому коді не завжди є єдиними винуватцями помилок у програмному забезпеченні. Прогалини у вимогах, які також називаються невизнаними потребами, є поширеною причиною дорогих дефектів. Ці прогалини призводять до того, що програміст не помічає помилок, що в свою чергу призводить до виникнення дорогих дефектів. Відсутність нефункціональних вимог, таких як тестованість, масштабованість, легкість обслуговування, продуктивність та інші, є поширеною причиною прогалин у вимогах безпеки.

Тестування програмного забезпечення може бути проведено різними способами. Один з підходів - статичне тестування, яке включає огляд коду, покрокові інструкції або перевірки без виконання програми. Другий підхід - динамічне тестування, яке вимагає виконання програми з використанням певного набору тестових випадків.

Статичне тестування - це метод тестування програмного забезпечення, який виконується без активного виконання програми. Замість цього, статичне тестування фокусується на аналізі вихідних кодів, документації, специфікацій та інших артефактів, пов'язаних з програмою, з метою виявлення помилок, недоліків та відхилень від вимог і стандартів.

Під час статичного тестування можуть використовуватись різні методи, такі як розгляд коду (code review), інспекція коду, аналіз дефектів та перевірка відповідності стандартам. Такі методи можуть включати ручний аналіз коду, використання автоматичних інструментів аналізу коду, перевірку документації на повноту та консистентність і багато іншого.

Статичне тестування допомагає виявити потенційні проблеми в програмному забезпеченні на ранніх етапах розробки, що дозволяє забезпечити виправлення помилок та вдосконалення якості програми ще до її активного виконання. Цей метод тестування може бути особливо корисним для виявлення проблем, пов'язаних з архітектурою, логікою програми та дотриманням внутрішніх стандартів кодування. Динамічне тестування - це метод тестування програмного забезпечення, при якому виконується активне виконання програми з метою виявлення помилок, недоліків та перевірки її правильності та працездатності. У процесі динамічного тестування вхідні дані подаються до програми, а отримані результати порівнюються з очікуваними результатами для виявлення відхилень.

Динамічне тестування може включати запуск програми з різними наборами тестових даних, виконання різних дій та перевірку реакції програми на ці дії. Це може включати тестування функцій, продуктивності, навантаження, стабільності, взаємодії з іншими компонентами та багато іншого.

У процесі динамічного тестування використовуються різні техніки, такі як тестування чорного ящика (не знаючи внутрішньої реалізації програми), тестування часткової доступності (тестування окремих компонентів програми) та тестування повного покриття коду (переконавання, що кожна частина коду була виконана принаймні один раз).

Динамічне тестування допомагає виявити помилки та дефекти, поліпшити якість програмного забезпечення та забезпечити коректну роботу програми в різних умовах.

Перевірка є складовою частиною як статичного, так і динамічного тестування. Обидва види тестування включають процес перевірки.

Пасивне тестування - це метод тестування програмного забезпечення, коли тестувальник не здійснює активних дій, а спостерігає за роботою системи або програми для виявлення помилок, неправильностей або недоліків. Під час

пасивного тестування зазвичай аналізується вихідна інформація, реакції системи на вхідні дані та спостерігається за загальною продуктивністю і стабільністю програми. Він може бути корисним для виявлення помилок, які можуть виникнути при навантаженні системи або під час тривалої роботи програми. Пасивне тестування може бути ефективним методом доповнення активного тестування, де тестувальник активно взаємодіє з системою, виконуючи різні дії та перевірки.

Отже, це описує різні підходи до тестування програмного забезпечення, включаючи статичне та динамічне тестування, а також активне та пасивне тестування.

Дослідницьке тестування (research testing) - це метод тестування програмного забезпечення, який передбачає проведення дослідницьких дій з метою виявлення нових проблем, несподіваних ситуацій або недоліків у програмі. Відмінністю дослідницького тестування є його активна роль у пошуку неочікуваних проблем, зокрема, використовуючи нетрадиційні сценарії або тестові дані.

У процесі дослідницького тестування тестувальник може застосовувати творчі підходи, нестандартні методи або спеціалізовані техніки для активного пошуку вразливостей або недоліків у програмі. Це може включати випадкові тестові сценарії, стрес-тестування, тестування в межах некоректного введення або незвичайних ситуацій, що можуть виникнути в реальному використанні програми. У дослідницькому тестуванні тестувальник самостійно приймає рішення про те, які тести розробляти, як їх проводити і як інтерпретувати результати. Це дає тестувальнику велику свободу і можливість впливати на якість тестування. Тестувальник вчиться під час процесу тестування, набуває нових знань і навичок, і використовує це для поліпшення якості своєї роботи.

Дослідницьке тестування спрямоване на виявлення потенційних проблем, які не можуть бути виявлені за допомогою стандартних методів тестування або сценаріїв. Воно може бути особливо корисним у випадках, коли програмне

забезпечення має важливу роль або високу критичність, і коли стандартні тестові сценарії не можуть повністю покрити всі можливі варіанти використання.

Дослідницьке тестування включає в себе розробку тестів, виконання їх і аналіз результатів паралельно протягом усього часу проекту. Тестувальник не обмежений заздалегідь заданим планом тестування, але залучається до процесу прийняття рішень щодо тестування на основі своїх знань і досвіду.

Основна ідея дослідницького тестування полягає в тому, що тестувальник виступає активним учасником процесу тестування і самостійно визначає кращі підходи до досягнення мети тестування - покращення якості програмного забезпечення.

Тестування білої скриньки (white-box testing) і тестування чорної скриньки (black-box testing) є двома основними категоріями підходів до тестування програмного забезпечення. Вони використовуються для опису точки зору тестера під час створення тестових випадків.

Тестування білої скриньки враховує знання про внутрішню структуру та реалізацію програмного забезпечення. Тестувальник аналізує вихідний код, алгоритми та логіку програми для розробки тестових сценаріїв. Цей підхід дозволяє глибоко розуміти внутрішні механізми програми та здійснювати точне тестування окремих компонентів або функцій. Тестування білої скриньки використовує такі методи, як структурне тестування, тестування границь, тестування рішень прийняття та інші.

Тестування чорної скриньки, навпаки, не враховує внутрішню структуру програмного забезпечення. Тестувальник працює на рівні інтерфейсу програми, використовуючи специфікації, функціональні вимоги та інші вхідні дані для розробки тестових випадків. Тестування чорної скриньки перевіряє зовнішню поведінку програми без знання її внутрішньої реалізації. Цей підхід дозволяє оцінювати програмне забезпечення з точки зору функціональності, стабільності та відповідності вимогам користувача.

Тестування сірого ящика (gray-box testing) є гібридним підходом, який поєднує елементи тестування білої та чорної скриньки. Тестувальник має обмежену кількість знань про внутрішню структуру програмного забезпечення, що дозволяє здійснювати тестування на рівні компонентів або модулів, але без докладного розуміння всієї системи. Тестування сірого ящика може використовувати аналіз специфікацій, виконання тестових сценаріїв та інші методи для перевірки програмного забезпечення.

Останнім часом розрив між тестуванням білої та чорної скриньки значно зменшився. З'явилась ідея тестування сірої скриньки, де тести розробляються з урахуванням конкретних компонентів або елементів дизайну програми. Цей підхід поєднує переваги обох методів і дозволяє ефективно тестувати програмне забезпечення.

ВИСНОВКИ

Під час проведення атестаційної роботи було розглянуто потреби, вимоги та технічні можливості потенційної аудиторії продукту. Метою було створення мінімального за розміром застосунку або повного розміщення його на сервері, який би підтримував широкий спектр сучасних операційних систем і працював на застарілих та бюджетних пристроях з обмеженим доступом до Інтернету.

Для відповіді на ці потреби було обрано Telegram як платформу для розробки. Telegram є популярним месенджером, який користується великим попитом серед молоді в Україні і підтримує багато пристроїв.

В результаті дипломної роботи було створено телеграм-бота для пошуку кулінарних рецептів з використанням мови програмування Python. Для взаємодії з системою використовується Telegram і його можливість інтеграції у чат-бота.

Під час виконання кваліфікаційної роботи були досягнуті такі поставлені завдання:

1. Проаналізовані вимоги, потреби, технічні можливості потенційної аудиторії даного програмного продукту. У результаті обрано платформу Телеграм для розробки додатку для пошуку кулінарних рецептів.
2. Проведений аналіз інструментів та програмних засобів реалізації, що покажуть найкращу продуктивність системи, під час застосування її в реальних умовах, та що спричинять найбільшу зручність під час написання програмного продукту. У ході аналізу обрано використання VisualStudio Code як середовища розробки, мову програмування Python, асинхронну бібліотеку `python-telegram-bot` для взаємодії з API Telegram.
3. Спроектовано та розроблено телеграм-додаток для пошуку кулінарних рецептів мовою Python у месенджері Телеграм.

4. Проведено тестування телеграм-додатку на виявлення помилок в роботі. Досліджено більшу частину сегментів взаємодії користувача з додатком, помилок не було виявлено, система працює стабільно та надійно.

Робота пройшла апробацію на Науково-технічній конференції «Застосування програмного забезпечення в інфокомунікаційних технологіях», м. Київ, ДУТ, 20 квітня 2023 року.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 5 takeaways from the Python Developer Survey 2019 [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://blog.jetbrains.com/pycharm/2020/05/5-takeaways-from-the-pythondeveloper-survey-2019/>
2. Python-telegram-bot [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://python-telegram-bot.org/>
3. Bot Code Examples [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://core.python.org/bots/samples/>
4. Spoonacular API [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://spoonacular.com/food-api/docs>
5. Telegram (software) [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://en.wikipedia.org/wiki/Telegram_\(software\)](https://en.wikipedia.org/wiki/Telegram_(software))
6. Офіційна документація Telegram Bot API [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://core.telegram.org/bots/api>
7. Telegram Bot Platform [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://telegram.org/blog/bot-revolution?setln=en>
8. All that cooking – History of Cooking [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://allthatcooking.com/history-of-cooking/>
9. G. van Rossum – Python (programming language) [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://www.wikizero.com/en/Python_\(language\)](https://www.wikizero.com/en/Python_(language))

10. Visual Studio Code - Open Source ("Code - OSS") [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://github.com/microsoft/vscode/blob/main/README.md>
11. Stack Overflow Developer Survey 2021 [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://insights.stackoverflow.com/survey/2021#most-popular-technologiesnew-collab-tools-prof>
12. Introduction to Python [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: https://www.w3schools.com/python/python_intro.asp
13. “OMG Unified Modeling Language™ (OMG UML), Superstructure” UML Superstructure Specification, v2.3 / Object Management Group, 2010.
14. Діаграми UML для для моделювання процесів і архітектури проекту [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
15. Reddit [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.reddit.com/r/Python/>
16. Стаття на Medium [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://towardsdatascience.com/building-a-telegram-bot-using-python-part-1-539f2ef79833>
17. BBC GoodFood Recipes and cooking tips [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.bbcgoodfood.com/>
18. Allrecipes | Recipes, How-Tos, Videos and More [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.allrecipes.com/>
19. Recipes, Dinners and Easy Meal Ideas | Food Network [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.foodnetwork.com/recipes>
20. Draw.io Blog [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.drawio.com/blog>

Додаток А

ПРЕЗЕНТАЦІЯ ДО ЗВІТУ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Telegram-боту для пошуку кулінарних рецептів мовою Python

Виконав студент 4 курсу

Групи ПД-44

Ялович Денис Володимирович

Керівник роботи

К.т.н, доц, завідувач кафедри ІПЗ Негоденко Олена Василівна
Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ РОБОТИ

- **Мета роботи** — спрощення процедури пошуку рецептів за рахунок Telegram-боту.
- **Об'єкт дослідження** — процес автоматизації процесу пошуку кулінарних рецептів.
- **Предмет дослідження** — Telegram-бот для автоматизації процесу пошуку кулінарних рецептів.

ЗАВДАННЯ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз вимог, потреб, технічних можливостей потенційної аудиторії даного програмного продукту, переваги та недоліки існуючих застосунків.
2. Проаналізувати інструменти та програмні засоби реалізації, що покажуть найкращу продуктивність системи, під час застосування продукту в реальних умовах, та що спричинять найбільшу зручність під час написання продукту.
3. Спростувати та розробити Telegram-бот для пошуку кулінарних рецептів у месенджері Телеграм.
4. Провести тестування Telegram-бот на виявлення помилок в роботі.

3

АНАЛІЗ АНАЛОГІВ

Показник	BestRecipeBot	CulinaryRecipesUaBot	RecipeBot
Платформи	Android, iOS, Web	Android, iOS, Web	Android, iOS, Web
Необхідність доступу для користування ботом	-	+	-
Наявність пошуку за інгредієнтами	+	-	+
Можливість редагувати список наявних інгредієнтів	-	-	+
Можливість пошуку по фото	-	-	+
Загальна інформація про страву та її приготування	+	+	+

4

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги:

1. Користувач повинен мати можливість пошуку за інгредієнтами.
2. Користувач повинен мати можливість редагування списку наявних інгредієнтів.
3. Можливість пошуку інгредієнтів за допомогою фото.
4. Можливість пошуку рецептів шляхом вибору країни походження страв.
5. Користувач повинен отримати повну інформацію приготування страви.

Нефункціональні вимоги:

1. Додаток повинен підтримувати різні платформи.
2. Бот повинен працювати швидко та ефективно.
3. Бот повинен бути здатним працювати зі зростаючим обсягом користувачів та розширюватися без проблем.

5

ПРОГРАМНІ ЗАСОБИ ТА ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ



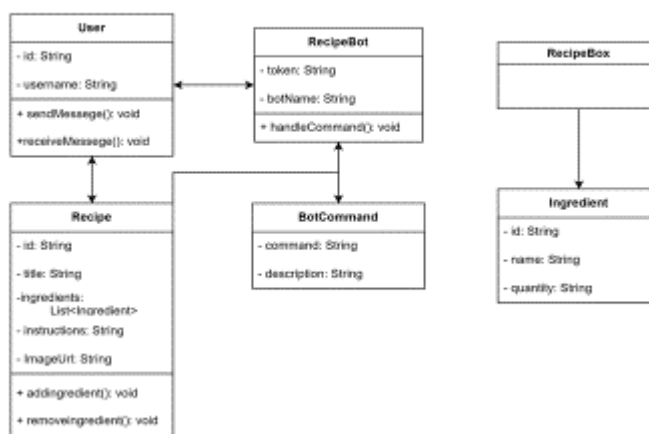
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



7

ДІАГРАМА КЛАСІВ ДОДАТКУ



8

ЕКРАННІ ФОРМИ



Відповідь на команду /start



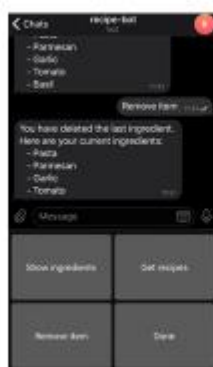
Додавання інгредієнту за допомогою фото



Результат пошуку рецепту

9

ЕКРАННІ ФОРМИ



Повідомлення про видалення інгредієнту зі списку



Результат пошуку за вибором країни походження страви

10

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Ялович Д.В. Розробка Telegram-бота для пошуку кулінарних рецептів мовою Python/ Ялович Д.В., Негоденко О.В. // Застосування програмного забезпечення в інфокомунікаційних технологіях: Матеріали всеукраїнської науково-технічної конференції. Збірник тез. 20.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 72.

11

ВИСНОВКИ

1. Проаналізовані вимоги, потреби, технічні можливості потенційної аудиторії даного програмного продукту. У результаті обрано платформу Телеграм для розробки додатку для пошуку кулінарних рецептів.
2. Проведений аналіз інструментів та програмних засобів реалізації, що покажуть найкращу продуктивність системи, під час застосування її в реальних умовах, та що спричинять найбільшу зручність під час написання програмного продукту. У ході аналізу обрано використання VisualStudio Code як середовища розробки, мову програмування Python, асинхронну бібліотеку `python-telegram-bot` для взаємодії з API Telegram.
3. Спроектовано та розроблено телеграм-додаток для пошуку кулінарних рецептів мовою Python у месенджері Телеграм.
4. Проведено тестування телеграм-додатку на виявлення помилок в роботі. Досліджено більшу частину сегментів взаємодії користувача з додатком, помилок не було виявлено, система працює стабільно та надійно.

12

ДЯКУЮ ЗА УВАГУ!