

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему «Мобільний чат-бот з підбору рецептів для раціонального харчування»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студентка групи ІТ.м-91 Приходченко Дар'я Вячеславівна

Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою _____

«___» грудня 2020 р.

Науковий керівник _____

(підпис)

к.т.н., доц., Парфененко Ю.В.

Голова комісії _____

(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології
проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2020 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Приходченко Дар'я Вячеславівна

(прізвище, ім'я, по батькові)

1 Тема проекту Мобільний чат-бот з підбору рецептів для раціонального харчування

затверджена наказом по університету від «26» листопада 2020 р. № 1824 - III

2 Термін здачі студентом закінченого проекту « 07 » грудня 2020 р.

3 Вхідні дані до проекту технічне завдання на розробку мобільного чат-бота.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі та методи дослідження, проектування мобільного чат бота “EatArt” розробка мобільного чат-бота «EatArt».

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність проблеми, мета дипломної роботи, задачі проекту, аналіз існуючих технологій, функціональні вимоги до системи, технічні вимоги до пз, проектування чат-бота «EatArt», етапи розробки «EatArt», практична реалізація, висновки, практичне значення, оприлюднення результатів роботи.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	03.08.20 – 07.08.20	
2	Визначення актуальності проекту	08.08.20 – 13.08.20	
3	Деталізація мети та ідеї проекту	14.08.20 – 19.08.20	
4	Аналіз існуючих технологій	20.08.20 – 26.08.20	
5	Вибір оптимальних інструментів розробки	27.08.20 – 02.09.20	
6	WBS планування	03.09.20 – 07.09.20	
7	OBS планування	08.09.20 – 09.09.20	
8	Створення календарного плану	10.09.20 – 16.09.20	
9	Визначення ризиків	17.09.20 – 23.09.20	
10	Ознайомлення з логікою роботи	24.09.20 – 30.09.20	
11	Вивчення інструментів розробки PyCharm та DialogFlow	28.09.20 – 04.10.20	
12	Розробка концепції дизайну	02.10.20 – 06.10.20	
13	Створення програмного коду	07.10.20 – 03.11.20	
14	Робота з логікою штучного інтелекту DialogFlow	21.10.20 – 09.11.20	
15	Тестування розробником	10.11.20 – 15.11.20	
16	Оформлення документації	15.11.20 – 30.11.20	

Магістрант _____

Приходченко Д.В.

Керівник роботи _____

к.т.н., доц. Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Мобільний чат-бот з підбору рецептів для раціонального харчування».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 35 найменувань, додатків. Загальний обсяг роботи – 66 сторінок, у тому числі 38 сторінок основного тексту, 3 сторінки списку використаних джерел, 25 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці мобільного чат-бота з підбору рецептів для раціонального харчування, призначеного для широкого кола користувачів. В роботі проведено аналіз предметної області та визначення потреби у розробці проекту, проведено аналіз існуючих технологій для розробки чат-бота, проведено деталізацію мети проекту з визначенням переліку задач, проведено ідентифікація методів реалізації дипломного проекту. У роботі виконано проектування чат-бота з урахуванням процесу розробки та процесу експлуатації продукту, виконано проектування сценаріїв дій користувача з системою, реалізовано суміжну роботу з елементами штучного інтелекту на прикладі інструменту DialogFlow. Результатом проведеної роботи є мобільний чат-бот з підбору рецептів для раціонального харчування «EatArt». Практичне значення роботи полягає у наданні користувачу повноцінного набору рецептів з урахуванням обраних параметрів та можливість пошуку страв на різних мовах.

Ключові слова: чат-бот, підбір рецептів, раціональне харчування, категорії, страви.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Обґрунтування необхідності створення чат-ботів	8
1.2 Аналіз існуючих чат-ботів для пошуку рецептів	11
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	16
2.1 Мета та задачі	16
2.2 Вибір засобів реалізації мобільного чат-бота	18
3 ПРОЕКТУВАННЯ МОБІЛЬНОГО ЧАТ БОТА “EATART”	20
3.1 Діаграми IDEF0 та їх декомпозиція	20
3.2 Моделювання варіантів використання мобільного чат-бота	24
4 РОЗРОБКА МОБІЛЬНОГО ЧАТ-БОТА «EatArt».....	26
4.1 Реалізація програмного коду Python	26
4.2 Ініціалізація інструменту DialogFlow та його навчання	31
4.3 Створення та ініціалізація бази даних	33
4.4 Тестування чат-бота «EatArt».....	34
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	38
Додаток А	41
Додаток Б	56

ВСТУП

На даний момент часу питання раціонального харчування щодня набуває все більшої популярності. У час, коли люди як ніколи раніше намагаються слідкувати за власним здоров'ям та позбутися шкідливих звичок, питання здорового харчування є дуже актуальним по всьому світу. Сервісів, що допомагають людям окремих прошарків суспільства контролювати власні прийоми їжі, досить багато, але універсального засобу, який буде відповідати потребам широкого загалу, не існує, в результаті чого було прийняте рішення про розробку мобільного інтерактивного чат-бота, який би допомагав дотримуватися раціонального харчування широкому загалу користувачів, враховуючи їх звички, уподобання, фізичні дані та цілі.

Об'єкт дослідження: процес інформаційної підтримки підбору рецептів для раціонального харчування.

Предмет дослідження: процес функціонування чат-бота з підбору рецептів для раціонального харчування.

Мета: розробити мобільний чат-бот з підбору рецептів для раціонального харчування.

Для того, щоб досягти мети даного дипломного проекту, необхідно реалізувати наступні поставлені задачі:

- провести детальний аналіз актуальності та необхідності створення чат-бота для широкого загалу користувачів;
- провести аналіз існуючих телеграм-ботів для пошуку рецептів;
- виконати розробку сценарію роботи чат-бота;
- виконати реалізацію функціоналу чат-бота;
- виконати адаптацію під мобільні засоби різних операційних систем;

– провести тестування продукту розробником та незалежного тестувальника.

Практичне значення: надання користувачу повноцінного набору рецептів з урахуванням обраних параметрів та можливість пошуку страв на різних мовах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування необхідності створення чат-ботів

З потужним розвитком інформаційних технологій та їх можливостей сучасний користувач стає більш вибагливим до функціоналу та якості роботи інформаційних додатків. На даний час в будь-якій сфері суспільства використання інформаційних технологій є невід’ємною складовою: медичні заклади мають власні веб-сайти для комунікації з клієнтами та надання швидкого сервісу підтримки, банківські відділення мають мобільні додатки для відображення всіх грошових потоків клієнтів та інші. Сьогодні користувач з наявністю смартфона має велику кількість месенджерів для швидкого процесу комунікацію з іншими людьми. На даний час на популярнішим месенджером спілкування є Telegram [1,2].

Telegram – це безкоштовний месенджер, що дозволяє спілкуватися з іншими користувачами та обмінюватися файлами різних форматів та розмірів. Ресурс надається безкоштовно, і його творці вклали кошти в їх створення, щоб вільно та безпечно спілкуватися.

З використанням Telegram користувач має можливість:

- спілкуватися в Інтернеті з користувачами по всьому світу, включаючи високоякісні аудіовіклики, що підтримуються;
- надсилати та отримувати файли у різних форматах (фотографії, аудіо, відео тощо) обсягом до 1,5 ГБ;
- використовувати хмарне сховище даних. Через вкладку «Обрані» будь-який файл передається у власний профіль користувача, а потім передається на віддалений сервер.
- використовувати автоматичну пошукову систему чат-бота [3].

Telegram забезпечує комфортне середовище для користувачів та чат-ботів. Спілкування з чат-ботом у телеграмі не складніше, ніж спілкування зі звичайними людьми. Тривалий час самі користувачі Telegram активно використовують ботів для виконання різних завдань. Але перш ніж описувати основні переваги, розглянемо статистику їх використання [4].

Чат-бот – це тип програмного забезпечення для штучного інтелекту (ШІ), яке може імітувати розмову (або чат) із користувачем природною мовою за допомогою програм обміну повідомленнями, веб-сайтів, мобільних додатків або телефонів. Чатових ботів часто описують як один з найдосконаліших та перспективних способів організації діалогової взаємодії людини з комп'ютером. Однак з технічної точки зору чат-боти представляють лише природну еволюцію системи відповідей на питання, що використовують обробку природними мовами (NLP). Формулювання відповідей на питання природною мовою є одним із найбільш типових прикладів обробки природної мови, що використовується в різних кінцевих цілях підприємства [5, 6].

Чат-боти можуть використовуватися для багатьох цілей: від щоденних завдань до розваг та розміщення їх у Messenger та соціальних мережах – Facebook Messenger, Telegram, V Kontakte, Slack тощо, але зазвичай вони використовуються на перших двох платформах [7].

Чат-боти були створені для розуміння цільових клієнтів та надання їм знань. Вони можуть зберігати інформацію про клієнта за необхідності, а потім можуть допомогти провести відповідний аналіз потреб клієнтів. Згідно з доповіддю, чат-боти стануть звичним явищем у 2021 році, оскільки понад 80% підприємств вирішать використовувати чат-боти [8-10].

На сьогоднішній день існує різноманітна кількість чат-ботів, які можуть використовуватися в будь-якій сфері сучасного користувача. Наприклад:

- чати-боти для пошуку книг;
- чат-бот для підбору флористичної композиції;
- чат-бот для підбору плейлисту кінострічок в залежності від вподобань користувача;

- чат-боти для реалізації процесу багатокористувальницької інтерактивної онлайн-гри;
- чат-боти для підбору та купівлі залізничних квитків, квитків для авіасполучення тощо;
- чат-бот для відстежування погодних умов та змін;
- чат-боти системної підтримки користувачів тощо [11, 12].

Не дивлячись на те, що чат-боти набирають популярності використання в різних видах діяльності користувача, на даний момент кількість чат-ботів для гастрономічної сфери замала. Сучасний користувач – це особистість, яка має свої уподобання та особливості в процесі вживання їжі. В світі 2020 популяризуються такі течії харчових звичок:

- здорове харчування;
- вегетеріанство;
- веганство;
- низькокалорійне харчування;
- сирійство тощо [13, 14].

На даний час більшість закладів харчування не задовольняють звички широкого кола користувачів. Тому люди зі специфічними видами харчування вимушені готувати самостійно. Більшість з них застосовують онлайн-ресурси для пошуку рецептів. Навіть користувачі, які не мають особливостей у харчуванні, застосовують ресурси для пошуку страв [15].

Провівши аналіз існуючих чат-ботів та виконавши структурування тематики їх використання, було отримано результат про те, лише 10% даних ресурсів розроблені для гастрономічної сфери. До того, ж малий відсоток чат-ботів для пошуку страв надають функціонал, який потребує замовник.

Тому було прийнято рішення про створення власного чат-бота, який би дозволив задовольнити достатній відсоток користувачів зі специфічними звичками харчування.

1.2 Аналіз існуючих чат-ботів для пошуку рецептів

Після того, як був проведений детальний аналіз проблеми створення чат-бота для пошуку страв, необхідно провести аналіз існуючих чат-ботів з даної тематики та провести їх порівняльну характеристику. Даний процес дозволить виявити переваги та недоліки кожного аналогу для того, щоб виконати розробку власного чат-бота з урахуванням мінусів аналогів та вподобань замовника.

Першим розглянутим аналогом є MyCookBot. Кулінарний бот призначений для пошуку рецептів за назвою страви або інгредієнта. Ботом дуже зручно користуватися – користувачу потрібно лише вибрати мову для спілкування з «помічником» і ввести назву страви або списку інгредієнтів. Наприклад, за запитом "Морква" бот може надати корейську моркву, морквяного зайчика в червоному вині, моркву та нут-кіш, лазанью з кабачків та моркви, моркву в шампанському та апельсиновому соку тощо [16].

В документації по експлуатації чат-бота вказано, що користувач має можливість вибору мови використання бота [17]. Проте відкривши чат-бот та відправивши запит з командою «Встановлення мови», бот не дав відгуку. На рис. 1.1 представлений даний процес.

Наступним розглянутим аналогом є Fooder. Цей бот називає себе Ботом Єнот, а також пропонує рецепти, придатні на всі випадки життя. Єнот надішле користувачу покрокову інструкцію з приготування будь-якої страви, а також інструкції щодо інгредієнтів [18].

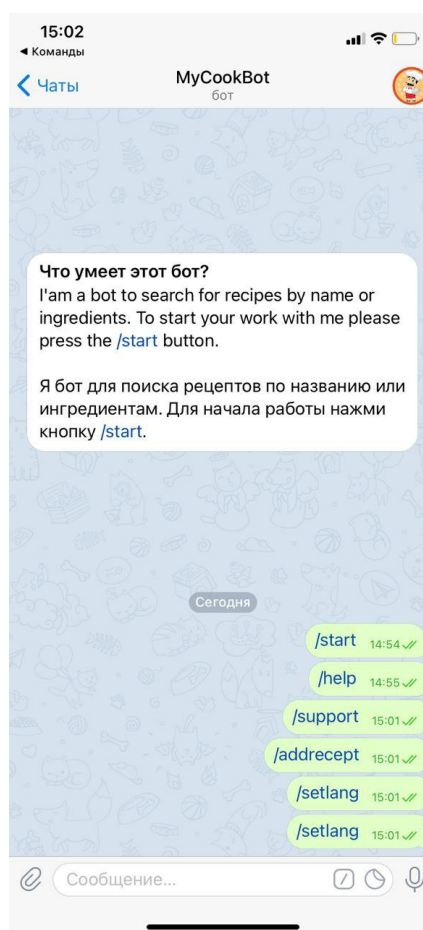


Рисунок 1.1 – Процес проби встановлення мови використання бота «MyCookBot»

Розробники чат-бота надають коротку інструкцію використання даного ресурсу. На рис. 1.2 представлена інструкція до чат-бота.

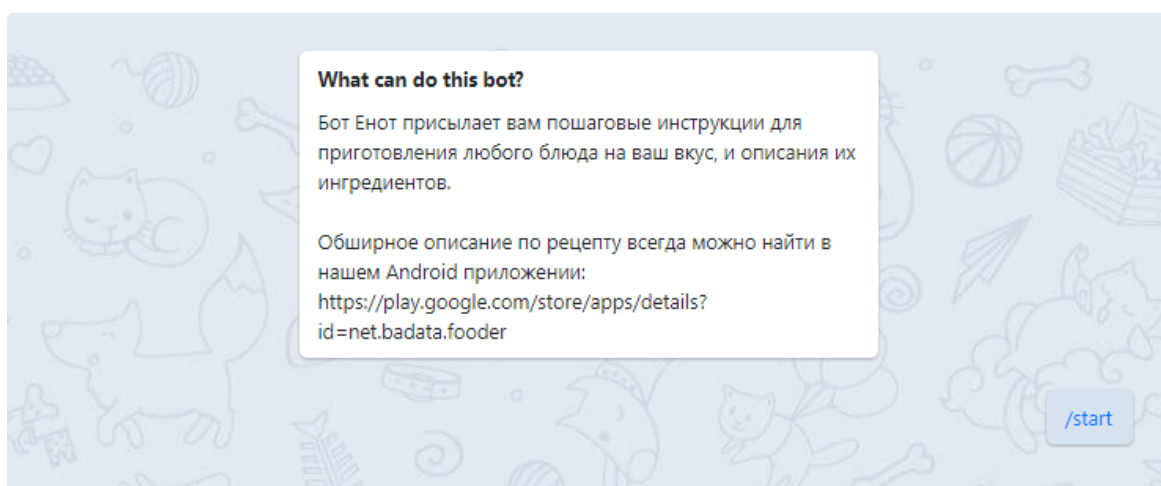


Рисунок 1.2 – Інструкція до чат-бота «Fooder»

Для того, щоб виконати пошук страв, користувачу необхідно ввести ключові слова пошуку. На рис. 1.3 представлений процес пошуку страв за словом «Наполеон» [19].

Останнім розглянутим аналогом є чат-бот «Шеф-повар». Даний чат-бот є найновішим серед розглянутих аналогів, він був розробленим в лютому 2019 року [20].

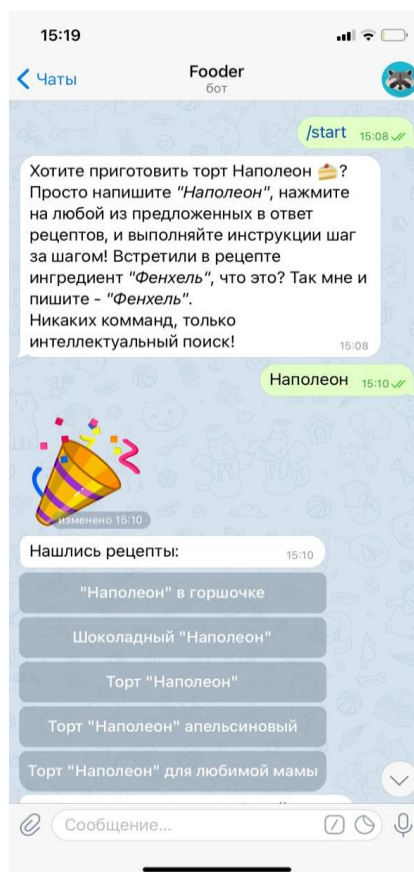


Рисунок 1.3 – Процесс пошук страв за словом «Наполеон» в чат-боті «Fooder»

За офіційними даними розробників анонсований той факт, що чат-бот може надавати інформацію не лише про рецепт, а й поради під час приготування та підбору інгредієнтів [21].

Пошук страв відбувається в залежності від вибору категорії страви (рис. 1.4).

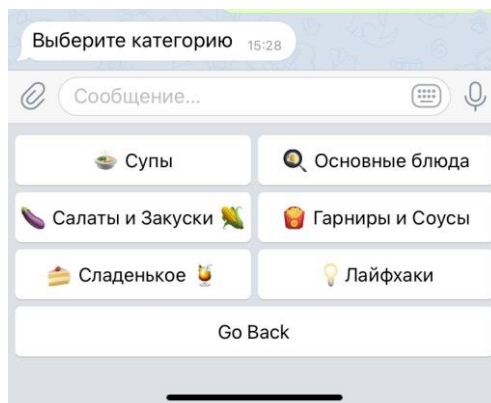


Рисунок 1.4 – Категорії пошуку страв

Після того, як користувач обирає категорію, бот надсилає посилання з відео-рецептами з ресурсу «YouTube» (рис.1.5).

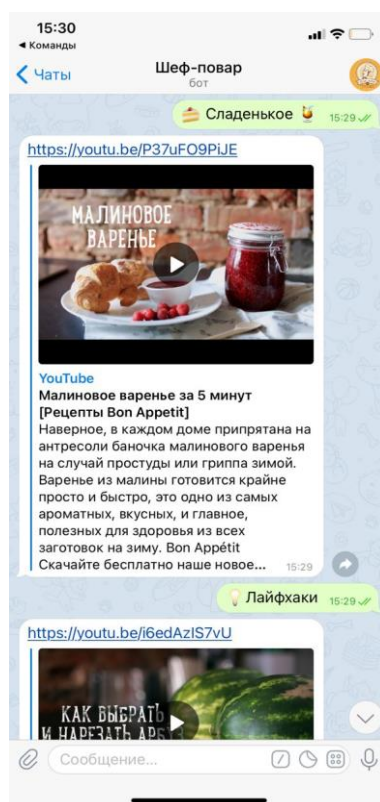


Рисунок 1.5 – Результат вибору категорії страв

Після того, як був виконаний пошук існуючих чат-ботів для пошуку страв, необхідно було провести порівняльну характеристику функціональних можливостей аналогів у порівнянні з власною розробкою (табл.1.1).

Таблиця 1.1 – Порівняльна характеристика існуючих аналогів з власною розробкою

Назва критерію	Назва чат-бота			
	MyCookBot	Fooder	Шеф-повар	EatArt
1. Наявність переліку команд керування	+	-	-	+
2. Пошук страв за категоріями	-	-	+	+
3. Можливість встановлення мови використання	+	-	-	+
4. Подання рецепту в доступній формі розуміння	-	-	+	+

Після проведення порівняльної характеристики існуючих чат-ботів для пошуку страв та власної розробки «EatArt» було прийнято рішення про реалізацію чат-бота «EatArt», оскільки розглянуті аналоги не мають у своєму складі цілісної картини функціоналу, який би задовольняв потреби замовника. У більшості аналогів відсутня функція багатомовного пошуку страв, що значно зменшує коло країн використання ресурсу. До того ж, аналоги MyCookBot та Fooder не надають користувачу детального рецепту страв, так як це відображається у чат-боті «Шеф-повар». Тому було прийнято рішення про розробку чат-бота «EatArt», який би містив у своєму складі переваги розглянутих аналогів та усунути їх недоліки.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Метою даної кваліфікаційної роботи магістра є створення мобільного чат-бота для підбору рецептів для раціонального харчування. Цей проект нестиме соціальну цінність для широкого кола користувачів, які мають специфічні харчові уподобання, а також мають намір слідкувати за своїми харчовими звичками, оскільки існування даного чат-бота значно полегшить щоденні «ритуали» користувачів, пов'язані з приготуванням їжі.

Також створений чат-бот повинен мати наступні функціональні можливості:

- містити команди для інтерактивного спілкування з ботом;
- містити команди для запуску роботи бота;
- містити інструкції для користувача в описі бота;
- містити блок з інформацією про розробників;
- містити розділ для запису власних рецептів;
- містити блок для можливості зміни мови спілкування з ботом.

Мобільний чат-бот має також визначені функціональні вимоги:

- реалізація мобільного чат-бота мовою програмування Python;
- наявність коректного відгуку на запити користувача;
- наявність обробки некоректних запитів користувача.

Нефункціональні вимоги до чат-бота:

- підбор сучасних та актуальних рецептів;
- інтерактивне спілкування з користувачем у неформальному стилі.

Виконавши детальний аналіз предметної області, стало відомо, чат-бот є необхідним для широкого загалу користувачів для зручного пошуку рецептів за різноманітними категоріями, для категоризації рецептів специфічних груп

харчування, для зберігання інформації про харчові звички в одному місці, для пришвидшення процесу підбору рецепту з наявних у користувача інгредієнтів, для підвищення зручності комунікації зі спеціалістом-дієтологом.

Також завдяки проведенню аналізу предметної області стало відомо, що більшість існуючих чат-ботів, які можна вважати аналогами даного проекту є непрацюючими на даний час або ж недостатньо задовольняють потреби широкого загалу користувачів. Тому було прийняте рішення про розробку мобільного чат-бота для підбору рецептів з раціонального харчування. Даний проект має соціальну цінність не лише для користувачів зі специфічними харчовими звичками, а і для великої кількості користувачів, що мають на меті слідкувати за власною системою харчування.

Після проведення детального аналізу предметної області, наступним етапом є безпосередня реалізація чат-бота. Для цього першозавданням є реєстрація майбутнього чат-бота та робота з необхідними бібліотеками Python. Далі є безпосередньо розробка програмного коду проекту.

Після того, як буде виконана розробка чат-бота як програмного продукту, наступним необхідним етапом є тестування, як розробником, так і незалежною особою. Недоліки, виявлені на етапі тестування мають бути проаналізовані та виправлені.

Під час деталізації мети та задач проекту було проведено планування робіт, що представлено детально у Додатку А.

2.2 Вибір засобів реалізації мобільного чат-бота

Для реалізації даного проекту було обрано мову програмування Python як динамічну інтерпретовану об'єктно-орієнтовану для програмування чат-ботів.

Python – це багатоцільова мова програмування, яка дозволяє створювати читабельний код. Відносна простота Python дозволяє створити програму, яка набагато коротша, ніж програма, написана іншою мовою. Python – це кросплатформенна мова програмування. Це означає, що програми Python можна запускати в різних операційних системах без будь-яких змін. Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python та містить готові інструменти для використання з операційними системами, веб-сторінками, базами даних, різними форматами даних, графічними інтерфейсами для побудови програм тощо. Програми, написані мовою програмування Python, можуть бути невеликими сценаріями або складними системами [22].

Задля програмування коду Python було обрано середовище розробки PyCharm. PyCharm – це інтегроване середовище розробки (IDE) для комп'ютерного програмування (зокрема мовою програмування Python) [23]. Він забезпечує аналіз коду, графічний налагоджувач, інтегрований тестер модулів, інтеграцію із системами контролю версій (VCSes), а також підтримує використання Django для веб-розробки та Anaconda для аналізу даних [24]. PyCharm є крос-платформенним і має версії для Windows, macOS та Linux [25].

Для реалізації спілкування чат-бота за допомогою основ штучного інтелекту було обрано середовище розробки DialogFlow. Це хмарна послуга розпізнавання природних мов, що надається Google, яка підтримує кілька мов, включаючи російську. Він має безкоштовні обмеження використання, і тому можна використовувати бібліотеки різними мовами для використання з API,

тому його легко інтегрувати у проект. Також у Dialogflow (скорочено "DF") різні месенджери інтегровані [26].

DialogFlow використовує найсучасніші алгоритми нейронної мережі та технологію глибокого навчання для розпізнавання мови з високою точністю. Оскільки Google вдосконалює технологію розпізнавання внутрішньої мови, що використовується в продуктах Google, точність хмарного перетворення тексту в текст з часом покращується.

3 ПРОЕКТУВАННЯ МОБІЛЬНОГО ЧАТ БОТА “EATART”

Проаналізувавши предметну область, актуальність розробки проекту, технології реалізації проекту, виконавши етап планування проекту, було розпочато наступний етап розробки дипломного проекту – мобільного чат-бота. Протягом періоду проектування була розроблена діаграма представлення IDEF0, яка детально описує послідовність процесу підбору рецепта та сформована діаграма використання, щоб продемонструвати всі можливі функції, які користувач може виконувати під час роботи з продуктом.

3.1 Діаграми IDEF0 та їх декомпозиція

Першим етапом проектування роботи мобільного чат-бота має бути розробка контекстної діаграми A-0, оскільки вона містить короткий та лаконічний опис усіх внутрішніх процесів системи. В процесі моделювання була розроблена контекстна діаграма, що представлена на рис.3.1. Вона є діаграмою верхнього рівня та представляє систему без деталізації.

Діаграма A-0 складається з одного функціонального блоку, вхідних та вихідних даних, елементів управління та інструментів та була побудована в середовищі BPWin [27]. Вхідними даними проекту вважається наповнена база рецептів раціонального харчування, загальні принципи класифікації рецептів на категорії. До елементів управління відноситься безпосередній запит користувача та загальна документація мови програмування Python [28-29]. Інструмент виконання процесу підбору рецепту є мобільний чат-бот та месенджер Telegram. Вихідними даними ж вважається підібраний рецепт, який отримує користувач.

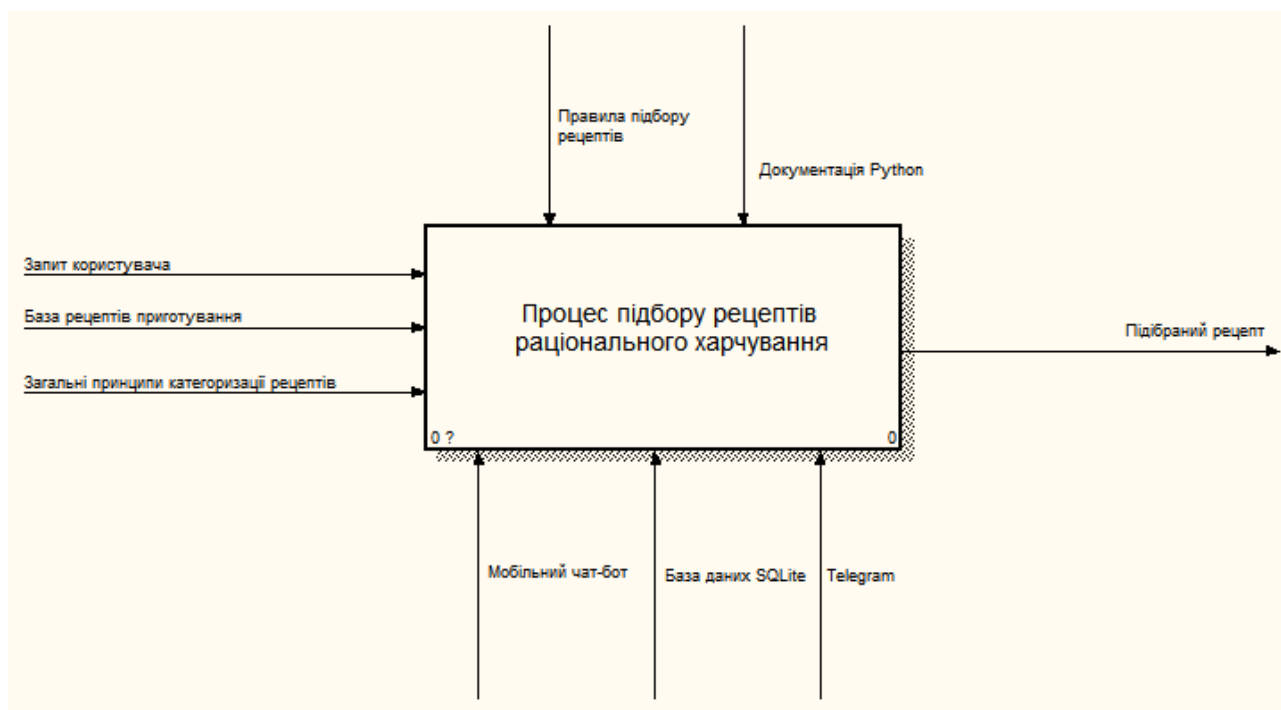


Рисунок 3.1 – Контекстна діаграма в нотації IDEF0

У процесі декомпозиції контекстної діаграми IDEF0 основний процес підбору рецепту було поділено на два етапи:

- обробка запиту користувача згідно існуючої категоризації бази даних;
- робота з існуючою базою даних рецептів.

Перший етап декомпозиції наведений на рис.3.2.

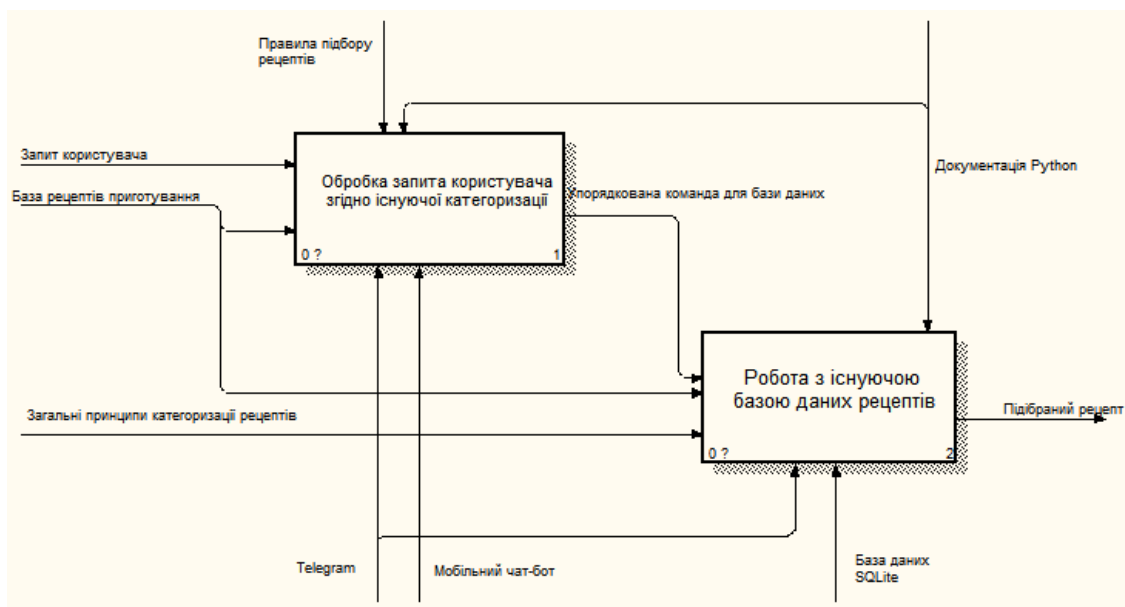


Рисунок 3.2 – Декомпозиція першого рівня

Наступним етапом було проведення декомпозиції другого рівня та деталізація процесів, що відбуваються. Результат декомпозиції другого рівня блоку «Обробка запиту користувача згідно існуючої категоризації бази даних» наведений на рис. 3.3, декомпозиція другого рівня блоку «Робота з існуючою базою даних рецептів» представлений на рис. 3.4.

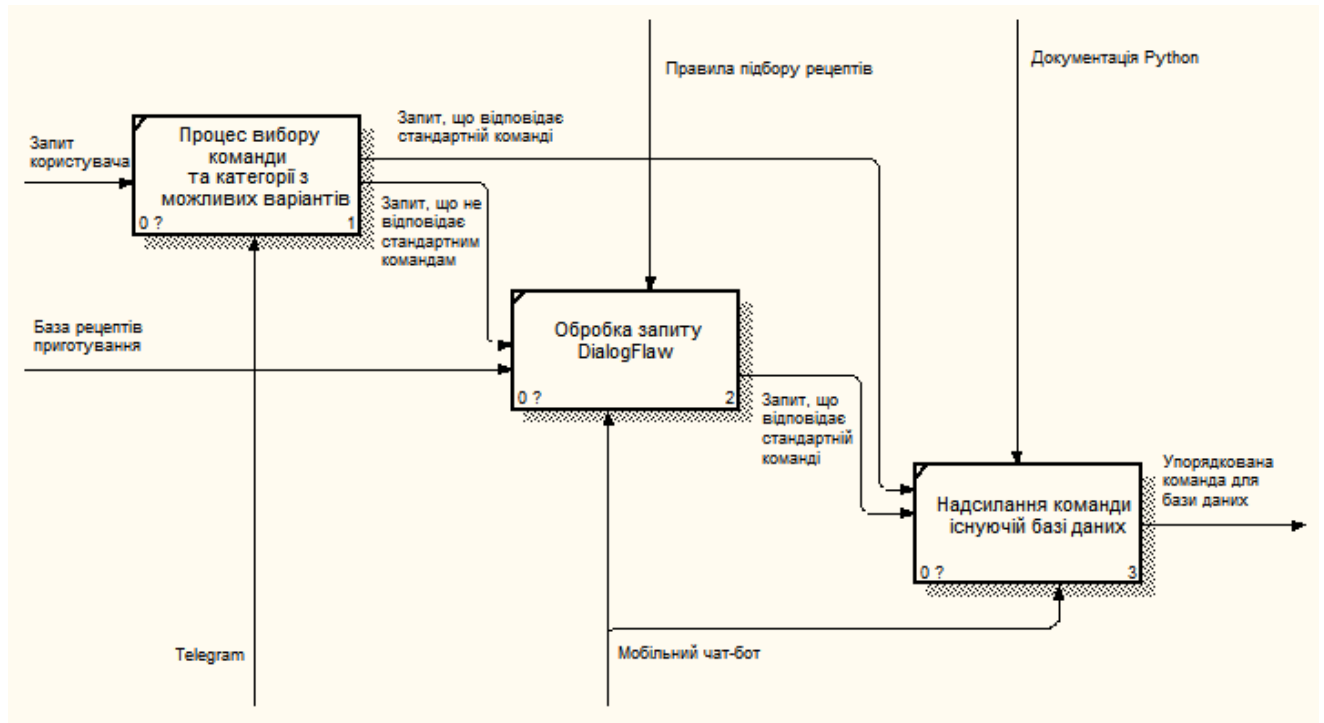


Рисунок 3.3 – Декомпозиція процесу «Обробка запиту користувача згідно існуючої категоризації бази даних»

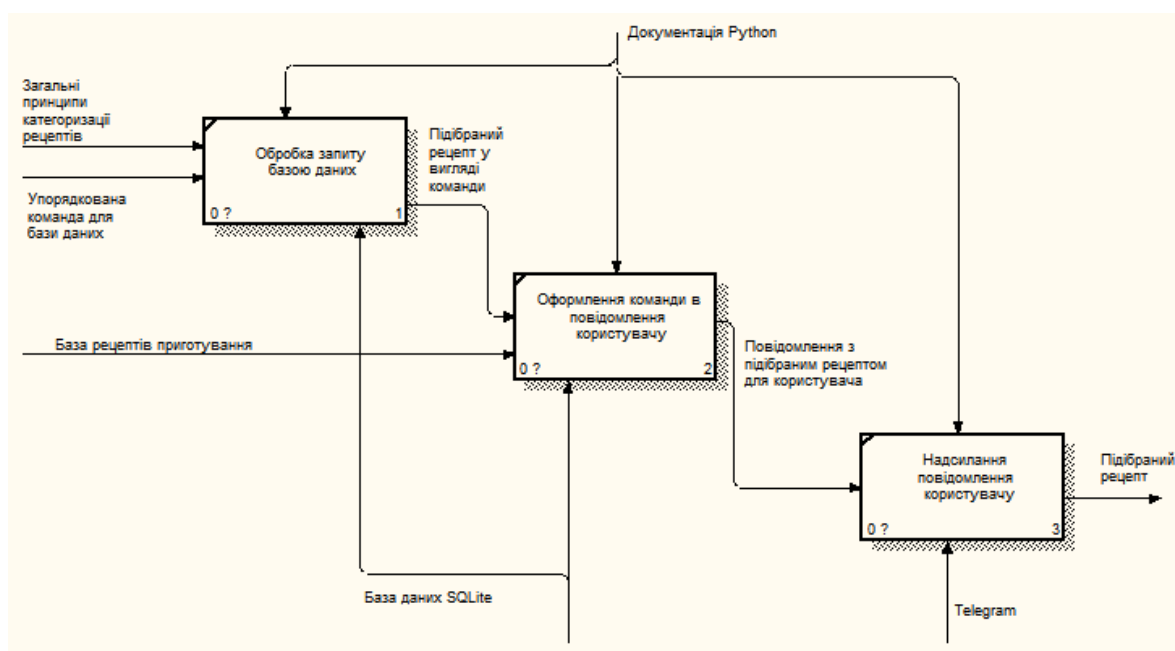


Рисунок 3.4 – Декомпозиція процесу «Робота з існуючою базою даних рецептів»

3.2 Моделювання варіантів використання мобільного чат-бота

За результатами проведення опису процесу підбору рецепту з раціонального харчування, було розроблено Use Case діаграму варіантів використання мобільного чат-бота [30].

Для створення діаграми варіантів використання мобільного чат-бота було визначено актора «Користувач», а також перелік сценаріїв використання продукту:

- почати роботу з ботом;
- ознайомитися з інструкцією з роботи з чат-ботом;
- обрати улюблену категорію рецептів з наведених;
- обрати вид страви;
- обрати основний інгредієнт для страви;
- отримати відео-рецепт з приготування обраної страви;
- відправити відгук про роботу бота.

Розроблена діаграма варіантів використання наведена на рис. 3.5.

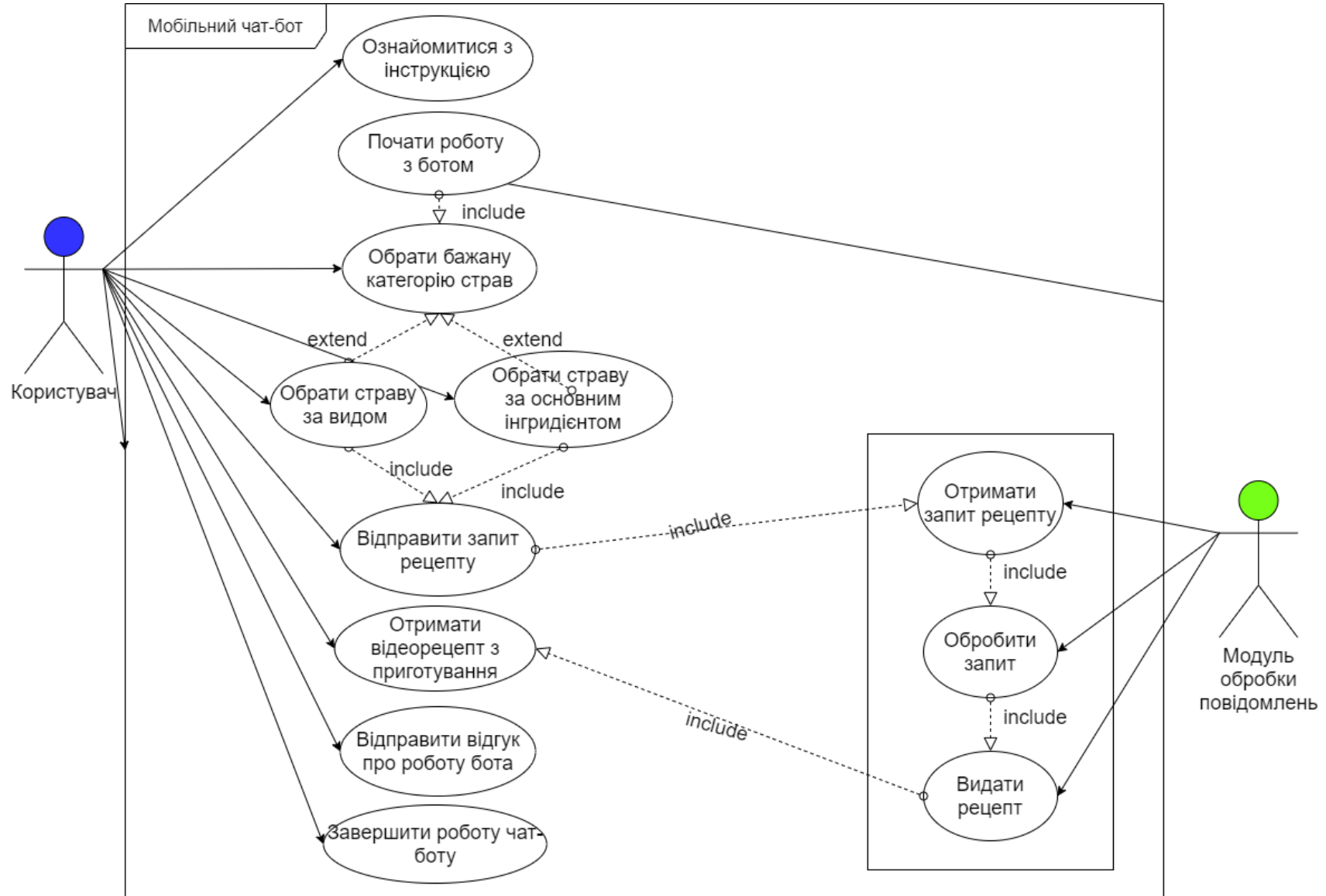


Рисунок 3.5 – Діаграма варіантів використання чат-бота

4 РОЗРОБКА МОБІЛЬНОГО ЧАТ-БОТА «EatArt»

4.1 Реалізація програмного коду Python

Початком першого етапу реалізації є створення бота в системі месенджера Телеграм та отримання API токenu. Даний процес зображений на рис. 4.1.

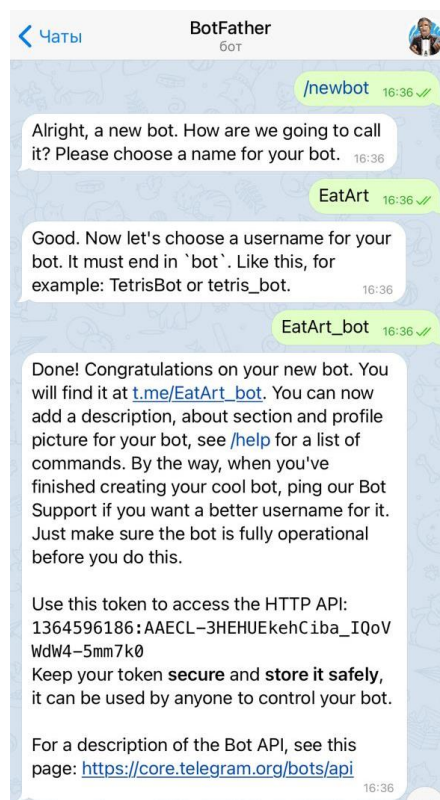


Рисунок 4.1 – Отримання API токenu для майбутнього бота

Наступним кроком є підключення бібліотеки telebot до програмного середовища PyCharm. Фрагмент програмного коду ініціалізації бота:

```
bot = telebot.TeleBot('1364596186:AAECL-3HEHUEkehCiba_IQoVWdW4-5mm7k0')
```

Наступним кроком є редагування бота за допомогою сервісів телеграм месенджеру. На рис. 4.2-4.5 зображені процес додавання опису бота, який буде відображатися у користувача одразу після відкриття бота, а також процес присвоєння боту власного розробленого логотипу та створення стандартних команд.

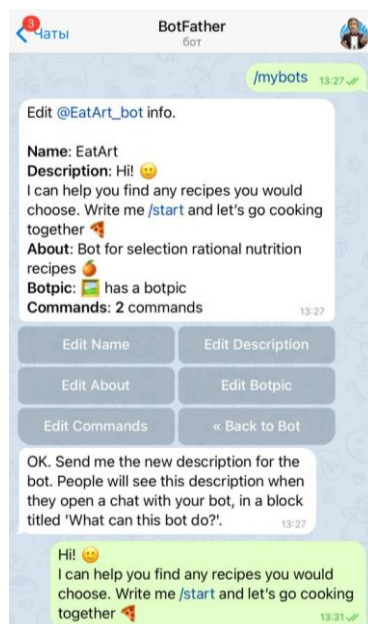


Рисунок 4.2 – Створення опису бота, який одразу бачить користувач



Рисунок 4.3 – Присвоєння боту опису та створеного власного логотипу

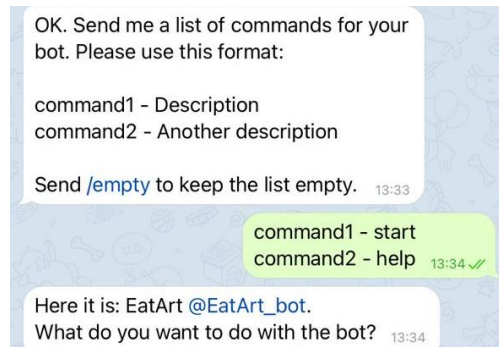


Рисунок 4.4 – Створення стандартних команд чат-бота

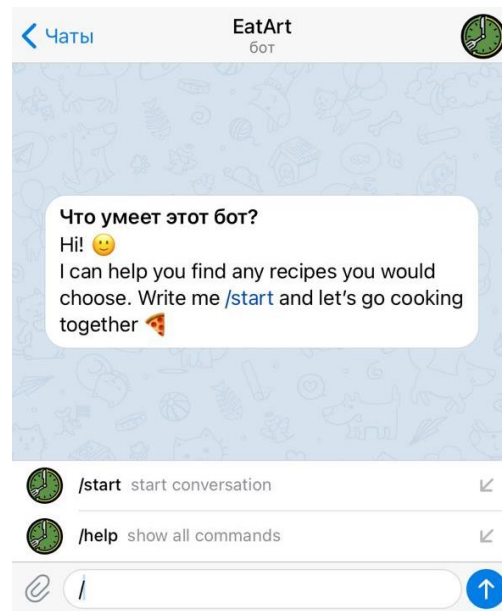


Рисунок 4.5 – Демонстрація першого екрану бота, який бачить користувач

Наступним етапом було створення програмного коду для обробки команди `/start`. Результатом виклику даної команди є привітання бота та демонстрація кнопок для зручного вибору подальшої мови для роботи з ботом. Фрагмент коду для реалізації даного процесу наведено нижче:

```
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привіт! Давай готувати разом',
reply_markup=keyboard1)

keyboard1 = telebot.types.ReplyKeyboardMarkup(True, True)
keyboard1.row('Українська мова', 'English language')
```

На рис. 4.6 наведено демонстрацію роботи обробки команди /start та відкриття програмної клавіатури для вибору мови.

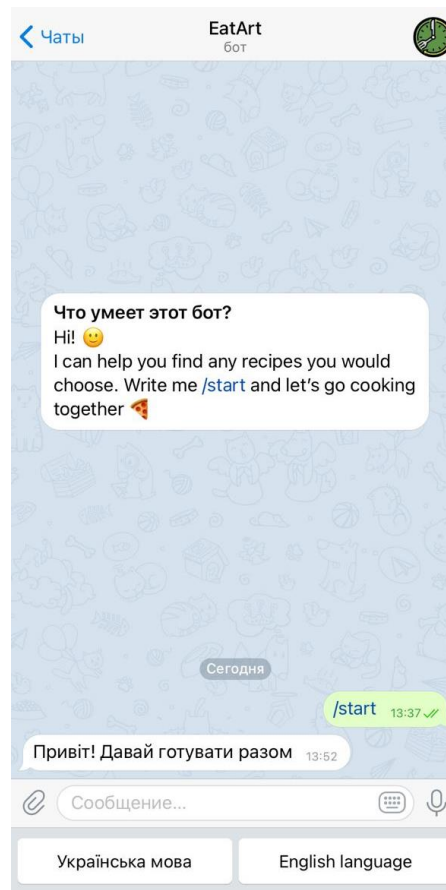


Рисунок 4.5 – Обробка команди /start

Наступним кроком є підготовка до етапу підключення середовища розробки DialogFlow. Для цього було створено оброки вибору користувача мови подальшої роботи та безпосереднє підключення до API DialogFlow.

Фрагмент програмного коду даного процесу наведено нижче:

```
@bot.message_handler(content_types=['text'])
def get_text_messages(message):
    if message.text == 'Українська мова':
        request = apiai.ApiAI('5be1bf8e36124d49v1564dvgca5f').text_request()
        request.lang = 'ua'
        request.session_id = 'EatArt_bot'
        request.query = update.message.text
        responseJson = json.loads(request.getResponse().read().decode('utf-8'))
        response = responseJson['result']['fulfillment']['speech']
        if response:
```

```
    bot.send_message(chat_id=update.message.chat_id, text=response)
elif message.text == 'English language':

    request = apiai.ApiAI('5be1bf8e36124d49v1564dvgca5f').text_request()
    request.lang = 'en'
    request.session_id = 'EatArt_bot'
    request.query = update.message.text
    responseJson = json.loads(request.getresponse().read().decode('utf-8'))
    response = responseJson['result']['fulfillment']['speech']
    if response:
        bot.send_message(chat_id=update.message.chat_id, text=response)
else:
    bot.send_message(message.chat.id, 'Пробач, я тільки вчуся Тебе розуміти.')
```

4.2 Ініціалізація інструменту DialogFlow та його навчання

Для реалізації спілкування зрозумілою мовою до чат-бота був підключений інструмент DialogFlow.

Dialogflow – це онлайн-сервіс Google, який дозволяє створити власний чат-бот або функцію для Google Асистента [31]. Існує велика кількість якісних статей, в яких обговорюються основні інструменти продукту. Dialogflow може порівнювати природні мовні фрази користувача (також українську) із певними "невербальними" значеннями (так звані "intent") та надавати відповіді. Крім того, сервіс зберігає контекст розмови, щоб він міг відреагувати на наступну фразу попереднього висловлювання. Інструментом досить легкий в керуванні, крім того, він також зберігає всю статистику запитів, тому існує можливість в декілька кліків мишкою збільшити підготовку агента з новими intent або допоснити існуючі. Таким чином і відбувається процес навчання: людина показує лише приклади проксі-запитів, пов'язаних з певним наміром, а сервіс автоматично будує модель і краще розуміє користувача [32-33]. Якщо йому потрібно виконати якусь функцію, він також може витягти з фрази деякі сутності, наприклад, дату чи місто. Для того, щоб скористатися даним інструментом, необхідно створити нового агента для власного бота та підключити програмного за допомогою API токена.

На рис. 4.6-4.7 зображено процес створення нового агента та налаштування мов, які він буде розуміти.

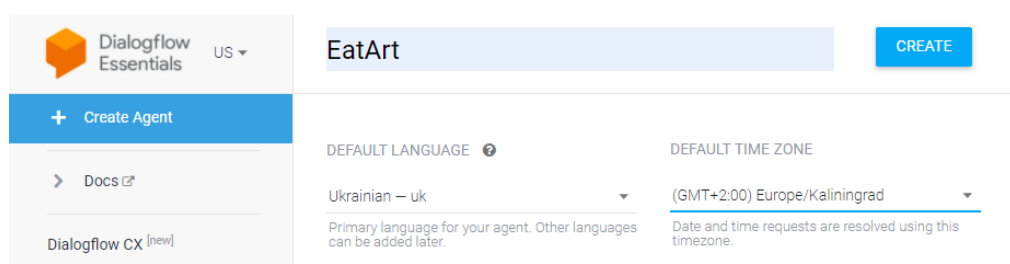


Рисунок 4.6 – Створення нового агента DialogFlow

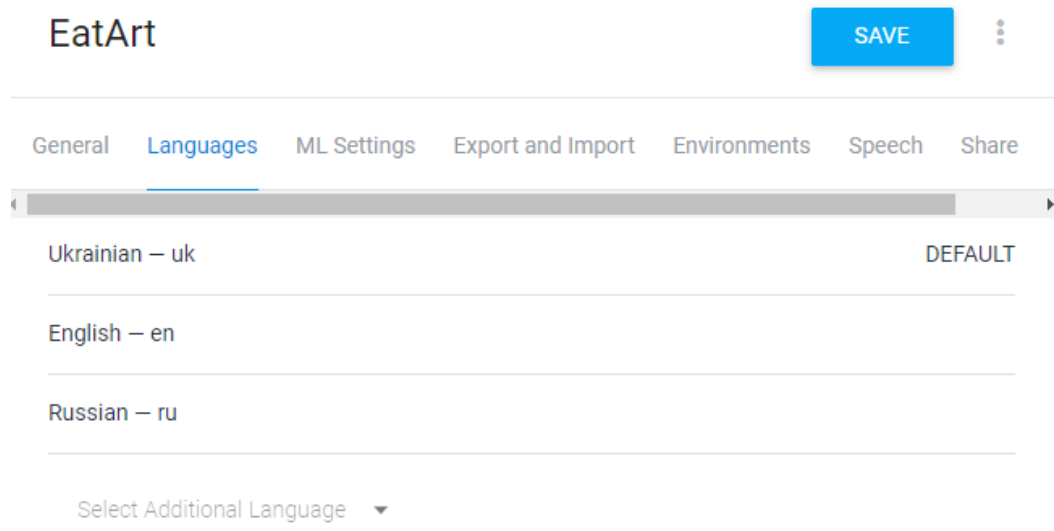


Рисунок 4.7 – Редагування мов розуміння агента

За допомогою створення різних intent відбувається навчання агента та кастомізація його спілкування з користувачем. Задача агента DialogFlow полягає в тому, щоб користувач визначився з категорією страв або основним інгредієнтом для приготування. До моменту визначення з одним з вищеперерахованих пунктів, з користувачем спілкується агент EatArt DialogFlow. Як тільки спілкування досягне категоризації відповідей, агент відправляє запит до існуючої бази даних задля підбору необхідного рецепту.

4.3 Створення та ініціалізація бази даних

Наступним етапом реалізації є створення та ініціалізації бази даних за допомогою вбудованих налаштувань PyCharm Professional та SQLite [24]. Початковим кроком є створення підключення до файлу з базою даних, що представлено на рис. 4.8.

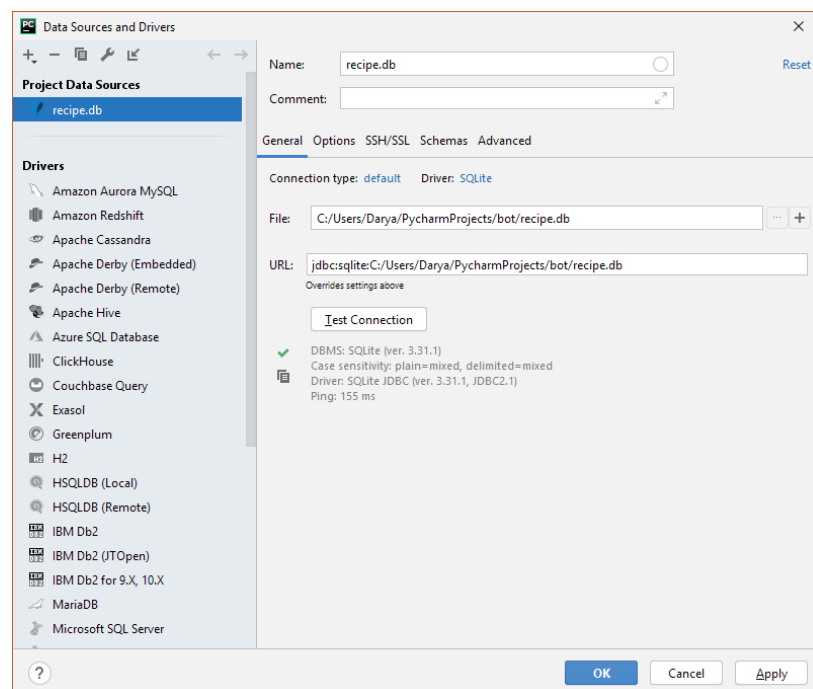


Рисунок 4.8 – Створення файлу бази даних

На етапі роботи з вбудованими інструментами Database SQLite в PyCharm було розроблено таблицю з рецептами, яку можна редагувати/додавати/видаляти дані.

```
db = sqlite3.connect('recipe.db')
sql = db.cursor()
```

```
sql.execute(""" CREATE TABLE IF NOT EXISTS recipes (
    meal TEXT,
    ingredient TEXT,
    recipe TEXT) """)
```

4.4 Тестування чат-бота «EatArt»

Останнім етапом реалізації чат бота є тестування готового продукту та завантаження на віддалений сервер задля безперервної роботи. На етапі тестування було проведено спілкування з ботом з різних пристроїв та акаунтів. Результати тестування наведено на рис. 4.9-4.11.

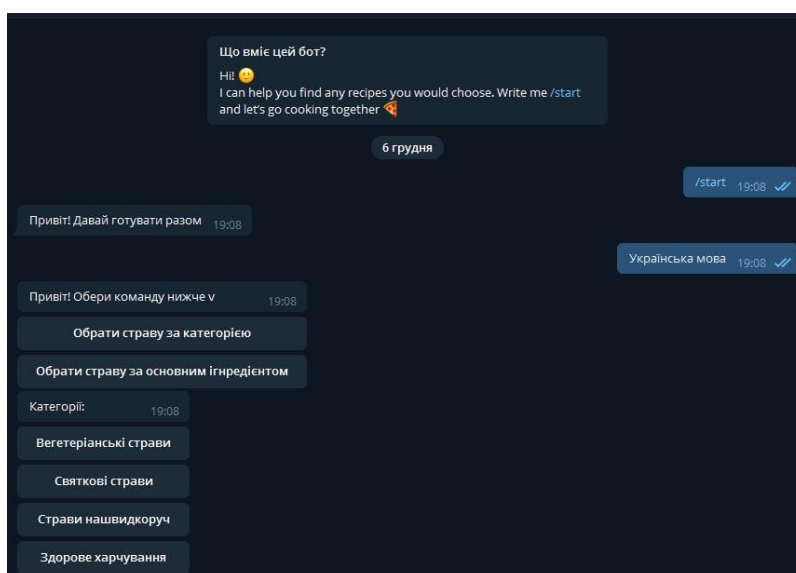


Рисунок 4.9 – Тестування роботи чат-бота на desktop версії месенджера

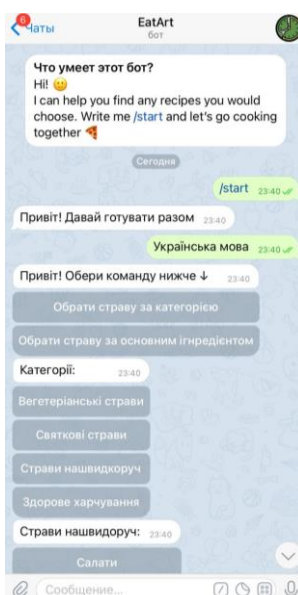


Рисунок 4.10 – Тестування роботи чат-бота на мобільній версії додатку

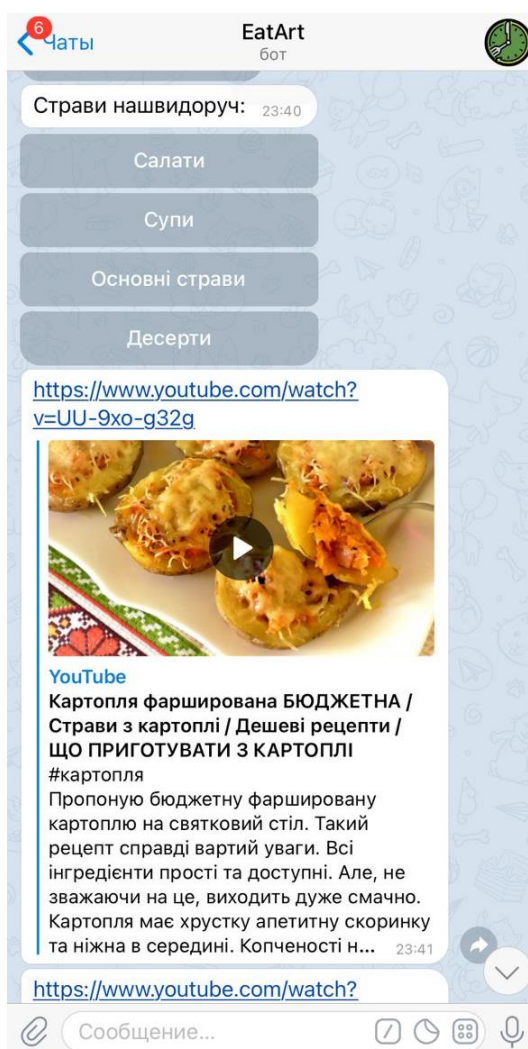


Рисунок 4.11 – Результат тестування роботи чат-бота

Завершальним етапом реалізації проекту можна вважати завантаження готового чат-бота на віддалений сервер. Процес завантаження та відлагодження роботи наведено на рис. 4.12-4.13.

```

21:26 ~$ pip -V
pip 19.3.1 from /usr/local/lib/python2.7/dist-packages/pip (python 2.7)
21:26 ~$ pip3 -V
pip 19.3.1 from /usr/lib/python3.8/site-packages/pip (python 3.8)
21:26 ~$ pip3 install python-telegram-bot --user
Looking in links: /usr/share/pip-wheels
Collecting python-telegram-bot
  Downloading https://files.pythonhosted.org/packages/cc/b3/f363e9c5c2e4690a1fd352c01263eb2938952888c09d73c824b49d288dcc/python_telegram_bot-13.1-py3-none-any.whl (422kB)
    |#####| 430kB 40.9MB/s
Requirement already satisfied: pytz>=2018.6 in /usr/lib/python3.8/site-packages (from python-telegram-bot) (2019.3)
Collecting APScheduler==3.6.3
  Downloading https://files.pythonhosted.org/packages/f3/34/9ef20ed473c4fd2c3df54ef77a27ae3fc7500b16b192add4720cab8b2c09/APScheduler-3.6.3-py3-none-any.whl (58kB)
    |#####| 61kB 2.7MB/s
Requirement already satisfied: certifi in /usr/lib/python3.8/site-packages (from python-telegram-bot) (2019.9.11)
Requirement already satisfied: decorator>=4.4.0 in /usr/lib/python3.8/site-packages (from python-telegram-bot) (4.4.1)
Requirement already satisfied: tornado>=5.1 in /usr/lib/python3.8/site-packages (from python-telegram-bot) (6.0.3)
Requirement already satisfied: cryptography in /usr/lib/python3.8/site-packages (from python-telegram-bot) (2.8)
Requirement already satisfied: tzlocal>=1.2 in /usr/lib/python3.8/site-packages (from APScheduler==3.6.3->python-telegram-bot) (1.5.1)
Requirement already satisfied: six>=1.4.0 in /usr/lib/python3.8/site-packages (from APScheduler==3.6.3->python-telegram-bot) (1.13.0)
Requirement already satisfied: setuptools>=0.7 in /usr/lib/python3.8/site-packages (from APScheduler==3.6.3->python-telegram-bot) (41.6.0)
Requirement already satisfied: cffi>=1.11.3,>=1.8 in /usr/lib/python3.8/site-packages (from cryptography->python-telegram-bot) (1.13.1)
Requirement already satisfied: pycparser in /usr/lib/python3.8/site-packages (from cffi>=1.11.3,>=1.8->cryptography->python-telegram-bot) (2.19)
Installing collected packages: APScheduler, python-telegram-bot
Successfully installed APScheduler-3.6.3 python-telegram-bot-13.1
21:28 ~$ pip3 install telebot --user
Looking in links: /usr/share/pip-wheels
Collecting telebot
  Downloading https://files.pythonhosted.org/packages/05/47/4ca2e24ff82d2f5f20e02a90ad45e97bf596c0a68733c79f29327c92f41/telebot-0.0.4-py3-none-any.whl
Requirement already satisfied: requests in /usr/lib/python3.8/site-packages (from telebot) (2.22.0)
Requirement already satisfied: urllib3<1.25.0, >=1.25.1, <1.26, >=1.21.1 in /usr/lib/python3.8/site-packages (from requests->telebot) (1.24.3)
Requirement already satisfied: chardet<3.1.0, >=3.0.2 in /usr/lib/python3.8/site-packages (from requests->telebot) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3.8/site-packages (from requests->telebot) (2019.9.11)
Requirement already satisfied: idna<2.9, >=2.5 in /usr/lib/python3.8/site-packages (from requests->telebot) (2.8)
Installing collected packages: telebot

```

Рисунок 4.12 – Підключення усіх необхідних бібліотек до серверу

```

/home/daryaprk/main.py
Keyboard shortcuts: Normal Share Save Save as... Run
1 import telebot
2 import random
3
4 bot = telebot.TeleBot('1364596186:AAECL-3HEHUEkehCiba_IQoVWdW4-5mm7k0')
5
6 from telebot import types
7
8 @bot.message_handler(commands=['start'])
9 def start_message(message):
10     bot.send_message(message.chat.id, 'Привіт! Давай готувати разом', reply_markup=keyboard1)
11
12
13 keyboard1 = telebot.types.ReplyKeyboardMarkup(True, True)
14 keyboard1.row('Українська мова', 'English language')
15
16 first = ["https://www.youtube.com/watch?v=kT3Ycb4eXBA", "https://www.youtube.com/watch?v=eI0NUzgQIE", "https://youtu.be/BCIoLPj-0e8"]
17
18
19
20

```

Рисунок 4.13 – Відлагодження роботи завантаженого чат-бота

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проведено аналіз предметної області розробки мобільних додатків за тематикою раціонального харчування, виконано етапи планування та проектування.

У результаті етапу планування було виконано наступні задачі:

- проведено деталізацію визначеного проекту методом SMART;
- створено деталізовані діаграми процесу реалізації проекту;
- визначено організаційну структуру;
- створено календарне планування з розрахунками критичного часу та з урахуваннями виникнення позаштатних ситуацій;
- проведено планування можливих ризиків проекту та роботу з різноманітними реакції на ризики, що можуть виникнути.

У результаті етапу проектування було розроблено контексні діаграми процесу підбору рецепту за допомогою мобільного чат-бота, а також їхня декомпозиція двох рівнів. Також було розроблено діаграму способів використання мобільного чат-бота користувачем та сценарії їхньої взаємодії.

Під час етапу реалізації мобільного чат-бота було створено програмний код мовою програмування Python у середовищі розробки PyCharm Professional, підключено сервіс штучного інтелекту DialogFlow, розроблено та підключено за допомогою середовища розробки PyCharm Professional базу даних рецептів SQLite, а також проведено тестування готового продукту, виправлення недоліків та завантаження на віддалений сервер задля безперервної роботи створеного під час роботи над кваліфікаційною роботою магістра чат-бота.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Telegram-бот на python с помощью библиотеки telebot [Электронный ресурс] – режим доступа: <https://habr.com/ru/post/448310/>
2. Dialogflow Telegram Integration [Электронный ресурс] – режим доступа: <https://cloud.google.com/dialogflow/es/docs/integrations/telegram>
3. Обучаемый Telegram чат-бот с ИИ [Электронный ресурс] – режим доступа: <https://habr.com/ru/post/346606/>
4. Чат-бот понимающий человеческую речь [Электронный ресурс] – режим доступа: <https://habr.com/ru/post/342762/>
5. Створення діаграми Ганта [Электронный ресурс] – режим доступа: <https://ru.smartsheet.com/blog/gantt-chart-excel>
6. Створення діаграм та мереж [Электронный ресурс] – режим доступа: <https://app.diagrams.net/>
7. Разница между INCLUDE и EXTEND диаграмма вариантов использования (use case) [Электронный ресурс] – режим доступа: <http://fkn.ktu10.com/?q=node/2236>
8. 2018: Использование и роль технологий в современном мире. Анализ применения информационных технологий в образовательном процессе [Электронный ресурс] – режим доступа: <https://bit.ly/2HC59hL>
9. Importance of information technology in today world Importance of information technology in today world [Электронный ресурс] – режим доступа: <https://thriveglobal.com/stories/importance-of-information-technology-in-today-world/>
10. Telegram: что это за программа [Электронный ресурс] – режим доступа: <https://semantica.in/blog/telegram-что-это-за-программа.html>

11. Telegram — как платформа для чат бота. Стоит ли его использовать? [Электронный ресурс] – режим доступа: <https://botcreators.ru/blog/telegram-chat-bota/>
12. Chatbot: What is a Chatbot? Why are Chatbots Important? [Электронный ресурс] – режим доступа: <https://www.expert.ai/blog/chatbot/>
13. Чат-боты обрели второе дыхание [Электронный ресурс] – режим доступа: <https://bit.ly/2HDevKd>
14. Почему чатботы захватывают мир [Электронный ресурс] – режим доступа: <https://bit.ly/2TEMfJa>
15. New Features of Chatbot in Upcoming Years [Электронный ресурс] – режим доступа: <https://chatbotnewsdaily.com/new-features-of-chatbot-in-upcoming-years-26bce65597d9>
16. What are Chatbots and Why are They Becoming so Popular? [Электронный ресурс] – режим доступа: <https://www.uctoday.com/contact-centre/what-are-chatbots-and-why-are-they-becoming-so-popular/>
17. The latest market research, trends, and landscape in the growing AI chatbot industry [Электронный ресурс] – режим доступа: <https://www.businessinsider.com/chatbot-market-stats-trends>
18. Чат-боты для бизнеса. Виды и примеры [Электронный ресурс] – режим доступа: <https://webim.ru/blog/13473-chat-bots-for-business-2/>
19. Five Different Types of Chatbot [Электронный ресурс] – режим доступа: <https://medium.com/voiceui/five-different-types-of-chatbot-17bb255b23b4>
20. Какие существуют виды питания! [Электронный ресурс] – режим доступа: <https://bit.ly/3oGo7V9>
21. Инна Дубсон. Виды питания – 2012. – [Электронный ресурс] – режим доступа: <https://health4ever.org/racion-pitanija/vidy-pitanija>
22. AI for Cooking: 4 Amazing Recipe Bots [Электронный ресурс] – режим доступа: <https://discover.bot/bot-talk/4-amazing-recipe-bots/>

23. 5 Telegram-ботов о еде и алкоголе [Электронный ресурс] – режим доступа: <https://bit.ua/2018/01/telegram-bot/>
24. MyCookBot от ProTelegram [Электронный ресурс] – режим доступа: <https://protelegram.ru/2016/08/25/mycookbot/>
25. Fooder [Электронный ресурс] – режим доступа: <https://ru.botostore.com/c/fooderappbot/>
26. Бот Telegram (Телеграм) – Шеф-повар [Электронный ресурс] – режим доступа: <https://bit.ly/3mFxAКВ>
27. Телеграмм бот «Шеф-повар» [Электронный ресурс] – режим доступа: <https://telegram-store.su/bots/021218033910the-chef-bot.html>
28. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. з англ. – СПб.: Символ-Плюс, 2011. – 992 с.
29. Creating a bot in Telegram [Электронный ресурс] – режим доступа: https://cloud.google.com/dialogflow/es/docs/integrations/telegram#creating_a_bot_in_telegram
30. Філіпов Д. JetBrains Debuts PyCharm Educational Edition. – Пер. з англ. – 2014.
31. Haagsman, Ernst. Collaboration with Anaconda, Inc. PyCharm Blog. – 2019.

Додаток А

Планування робіт

Мета була деталізована за методом SMART. Результати даної деталізації представлені в табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробка мобільного чат-бота для підбору рецептів з раціонального харчування
Measurable (вимірювана)	Враховуючи, що даний проект не є комерційним, результатом його виконання є оцінка замовника.
Achievable (досяжна)	Мета проекту досяжна, оскільки розробник проекту володіє необхідними вміннями та навичками з програмування мовою Python, а також розробкою необхідного дизайну проекту.
Relevant (реалістична)	Даний проект є реалістичним, оскільки наявні всі необхідні програмні засоби: PyCharm для створення програмного коду Python, Adobe Photoshop для розробки дизайну проекту[1,2,3].
Time-framed (обмежена в часі)	Робота над проектом відбувається згідно затвердженого календарного плану та має матрицю відповідальності.

Планування структури робіт проекту. Одним з етапів планування робіт є створення WBS діаграми та OBS діаграми. Представлення проекту, виконане у вигляді ієрархічної структури робіт називають структурною декомпозицією (Work Breakdown Structure). Вона досягається за допомогою послідовної декомпозиції. Інструмент спрямований на детальне планування, визначення та розподіл індивідуальної відповідальності виконавців, оцінку вартості продукту та проекту в цілому.

Перший рівень декомпозиції фіксує продукт проекту. На наступному, другому рівні описуються дії та основні заходи для досягнення продукту проекту. Таким чином триває декомпозиція до тих пір, поки не доходить до елементарних робіт. Елементарними роботами називають такі задачі чи завдання, які мають єдиний чіткий результат і він використовується для прийняття даної роботи. Коли декомпозиція доходить до завдань, на яких розуміються вузькі спеціалісти, її можна вважати завершеною. WBS-структура наведена на рис. А.1. Організаційна структура підприємства(виконавців) - (OBS – Organization Breakdown Structure) - визначається нижнім рівнем WBS-структури, в кожній гілці якої представлені відповідальні за конкретні дії та заходи проектної діяльності [4]. За кожен процес обов'язково має бути закріплена одна відповідальна особа. Звичайно, відповідальна особа не має виконувати весь прошарок роботи, відповідальним називають людей, які організують та відповідають за виконання останніх етапів WBS-структури, елементарних робіт. OBS-структура представлена на рис. А.2.

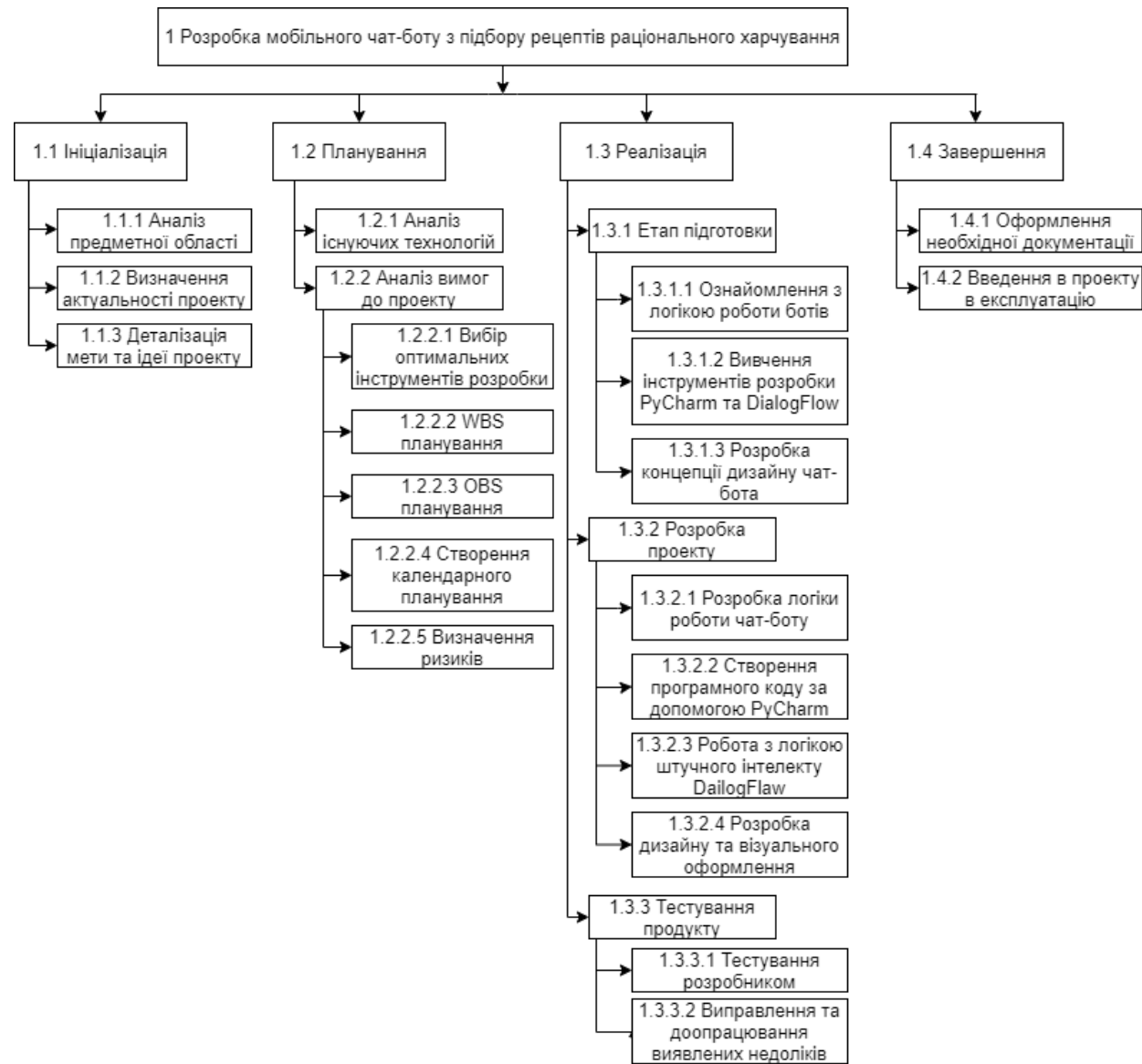


Рисунок А.1 – WBS структура проекту

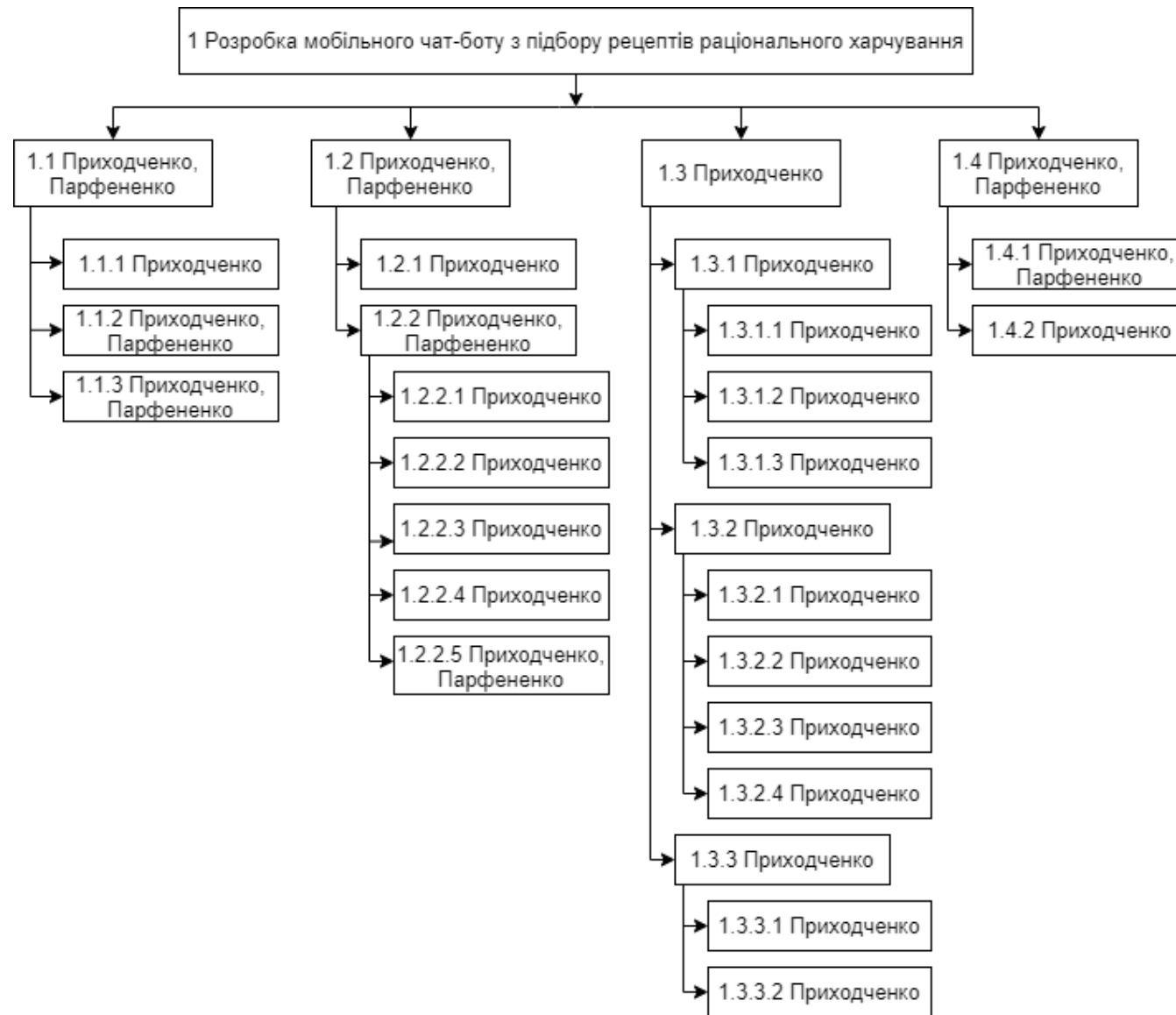


Рисунок А.2 – OBS структура проекту

Після створення організаційної структури та структури робіт проекту, було розроблено матрицю відповідальності. Матрицею відповідальності називають лінійний графік, де позначено кожен елементарну роботу, відповідальних за виконання та безпосередньо виконавців, інколи додають інших учасників проекту. Окрім цього кожній елементарній роботі додається критерій контролю з урахуванням технологічних та технічних факторів продукту. За допомогою матриці відповідальності розробляються робочі завдання, точність яких забезпечує максимальне наближення до результату, описаного при складанні WBS-структури. Робоче завдання обов'язково має бути чітким, повним, але водночас коротким.

На основі розроблених WBS та OBS структур проекту була побудована матриця відповідальності проекту, яка представлена в табл. А.2.

Таблиця А.2 – Матриця відповідальності

WBS/OBS	Приходченко Д.В.	Парфененко Ю.В.
1 Розробка мобільного чат-бота з підбору рецептів раціонального харчування		
1.1 Ініціалізація		
1.1.1 Аналіз предметної області	+	
1.1.2 Визначення актуальності проекту	+	+
1.1.3 Деталізація мети та ідеї проекту	+	+
1.2 Планування		
1.2.1 Аналіз існуючих технологій	+	

Продовження таблиці А.2

1.2.2 Аналіз вимог до проекту	+	+
1.2.2.1 Вибір оптимальних інструментів розробки	+	
1.2.2.2 WBS планування	+	
1.2.2.3 OBS планування	+	
1.2.2.4 Створення календарного планування	+	
1.2.2.5 Визначення ризиків	+	+
1.3 Реалізація		
1.3.1 Етап підготовки	+	
1.3.1.1 Ознайомлення з логікою роботи ботів	+	
1.3.1.2 Вивчення інструментів розробки PyCharm та DialogFlow	+	
1.3.1.3 Розробка концепції дизайну чат-бота	+	
1.3.2 Розробка проекту	+	
1.3.2.1 Розробка логіки роботи чат-бота	+	
1.3.2.2 Створення програмного коду за допомогою PyCharm	+	

Продовження таблиці А.2

1.3.2.3 Робота з логікою штучного інтелекту DailogFlaw	+	
1.3.2.4 Розробка дизайну та візуального оформлення	+	
1.3.3 Тестування продукту	+	
1.3.3.1 Тестування розробником	+	
1.3.3.2 Виправлення та доопрацювання виявлених недоліків	+	
1.4 Завершення		
1.4.1 Оформлення необхідної документації	+	+
1.4.2 Введення в проекту в експлуатацію	+	

Створення календарного плану робіт проекту. Діаграма Ганта вважається якісним інструментом для відображення цілей та задач. Основна відмінність цих інструментів - в способі демонстрації і відображення інформації. Управління проектами з діаграмами Ганта засноване на форматі гістограм. Це допомагає відслідковувати відсоток робіт, виконаних по кожному завданню [5]. Керівникам проектів дуже важливо правильно розподілити завдання і бути впевненими в тому, що проект буде завершений вчасно. Основна увага діаграм Ганта зосереджено на процентному завершенні кожного завдання. Крім того, діаграми Ганта краще для проектів з невеликою кількістю взаємопов'язаних завдань. Діаграма Ганта представлена на рис. А.3-А.4. На рис. А.5 зазначена детальна інформація про кожне завдання ІТ-проекту, строки та терміни його виконання.

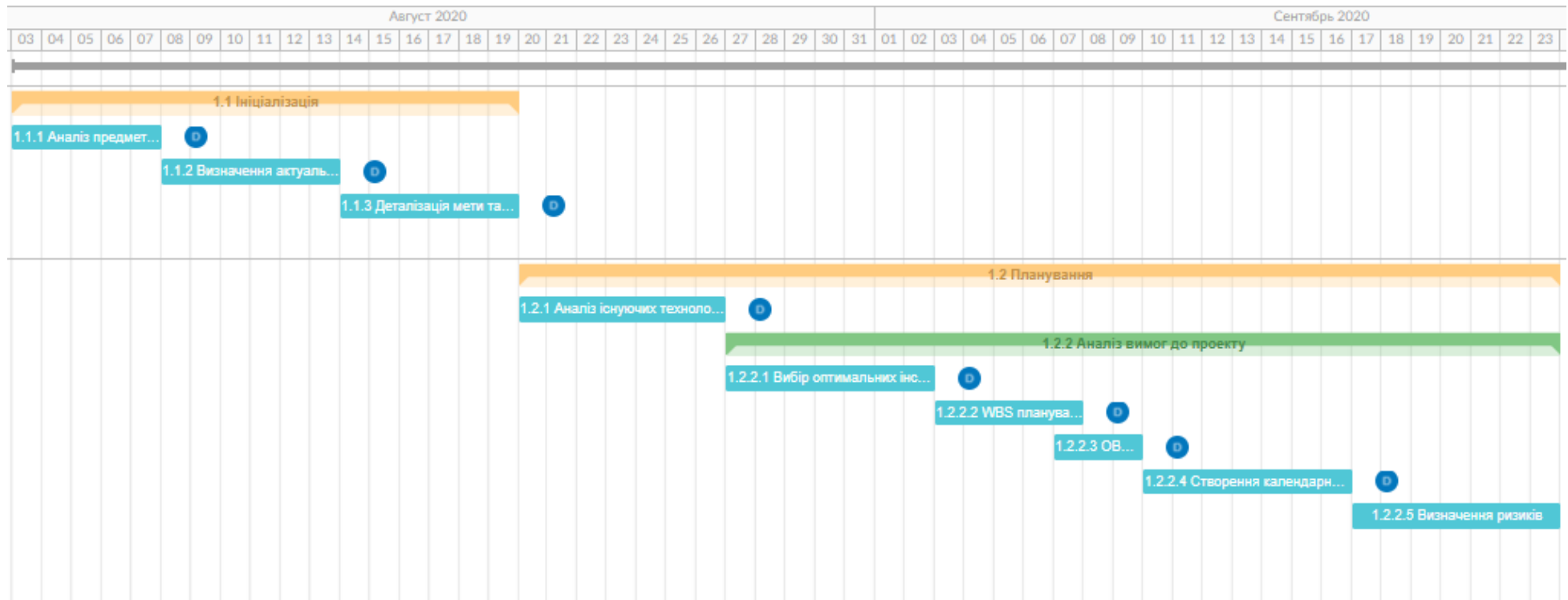


Рисунок А.3 – Календарне планування за допомогою діаграми Ганта

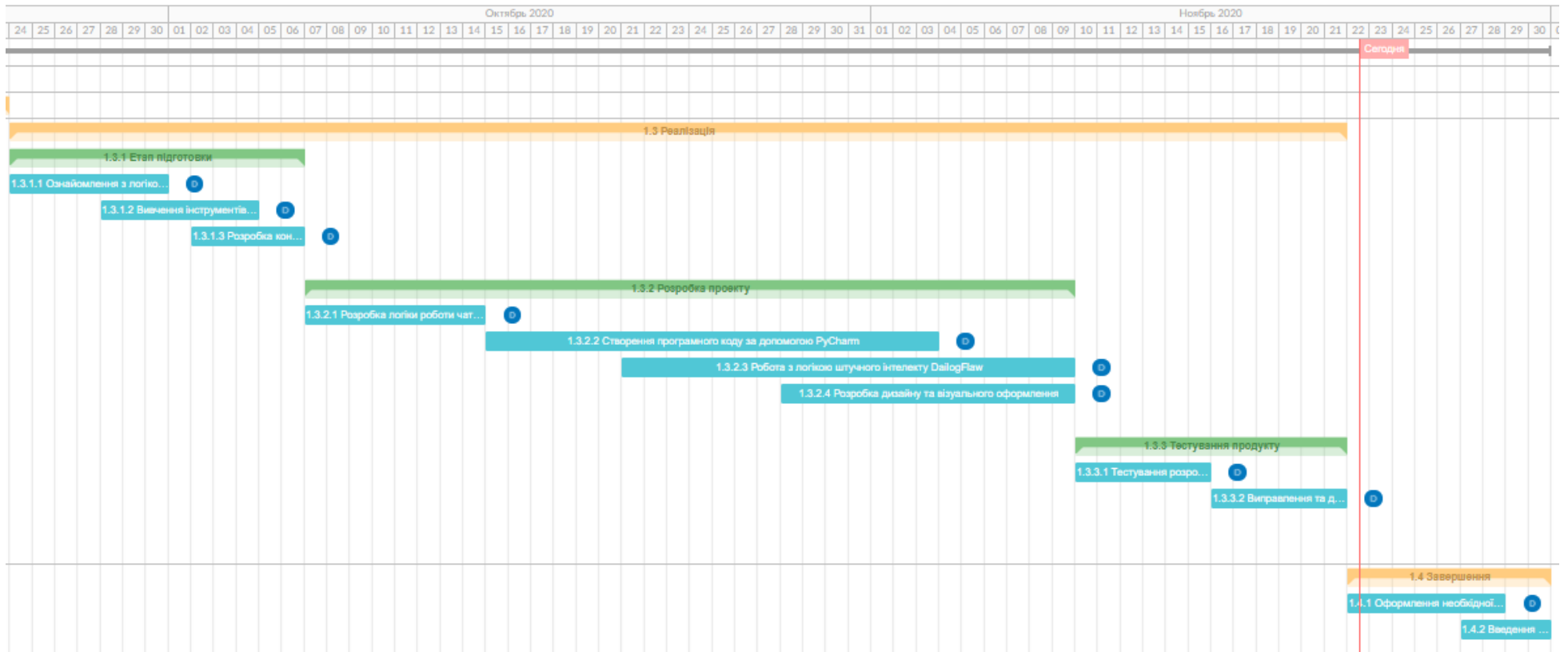


Рисунок А.4 – Календарне планування за допомогою діаграми Ганта

Задача		Начало	Завершение
1.1.1 Аналіз предметної області	①	03.08.2020	07.08.2020
1.1.2 Визначення актуальності проекту	①	08.08.2020	13.08.2020
1.1.3 Деталізація мети та ідеї проекту	①	14.08.2020	19.08.2020
1.2.1 Аналіз існуючих технологій	①	20.08.2020	26.08.2020
1.2.2.1 Вибір оптимальних інструментів розробки	①	27.08.2020	02.09.2020
1.2.2.2 WBS планування	①	03.09.2020	07.09.2020
1.2.2.3 OBS планування	①	07.09.2020	09.09.2020
1.2.2.4 Створення календарного планування	①	10.09.2020	16.09.2020
1.2.2.5 Визначення ризиків	①	17.09.2020	23.09.2020
1.3.1.1 Ознайомлення з логікою роботи ботів	①	24.09.2020	30.09.2020
1.3.1.2 Вивчення інструментів розробки PyCharm та DialogFlow	①	28.09.2020	04.10.2020
1.3.1.3 Розробка концепції дизайну чат-бота	①	02.10.2020	06.10.2020
1.3.2.1 Розробка логіки роботи чат-боту	①	07.10.2020	14.10.2020
1.3.2.2 Створення програмного коду за допомогою PyCharm	①	15.10.2020	03.11.2020
1.3.2.3 Робота з логікою штучного інтелекту DialogFlow	①	21.10.2020	09.11.2020
1.3.2.4 Розробка дизайну та візуального оформлення	①	28.10.2020	09.11.2020
1.3.3.1 Тестування розробником	①	10.11.2020	15.11.2020
1.3.3.2 Виправлення та доопрацювання виявлених недоліків	①	16.11.2020	21.11.2020
1.4.1 Оформлення необхідної документації	①	22.11.2020	28.11.2020
1.4.2 Введення в проект в експлуатацію	①	27.11.2020	30.11.2020




 Коментарі |
  Все файли |
  Время на задачі

Рисунок А.5 – Деталізація календарного планування

Планування ризиків проектування. Ризик – ймовірнісна подія, яка може позитивно чи негативно вплинути на проект. В свою чергу, управлінням ризиками вважається передбачення того, на скільки великою є ймовірність того, що чи інша подія відбудеться, яким чином вона може вплинути на роботу над проектом та ймовірні шляхи вирішення проблем. Процес управління ризиками є динамічним та відображає проектну роботу в умовах виникнення різноманітних ризиків.

Процес управління ризиками включає в себе такі пункти:

- 1) Ідентифікація ризиків (виявлення ризиків)
- 2) Оцінювання ризиків (оцінка ймовірності та впливу)
- 3) Заходи реагування на ризики
- 4) Моніторинг ризиків

Ідентифікація ризиків. Найпоширенішими ризиками проекту є:

- Ризик неправильному розподіленню робіт. Може виникати через помилки проектування; помилки планування; недоліки координації робіт тощо.
- Ризик перевитрати коштів. Може виникати через: зриви планів робіт; невірну стратегію; неузгодження частин проекту; невірний кошторис.
- Ризик нежиттєздатності проекту – передбачені доходи від проекту можуть бути недостатніми для відповідності поставленим меті та задачам проекту.

У результаті проведеного аналізу можливих ризиків проекту «Мобільний чат бот з підбору рецептів з раціонального харчування» було виявлено перелік ризиків, наведених у табл. А.3. Наведена нижче таблиця можливих ризиків проекту відображає їхню ймовірність та вплив на реалізацію та результат проекту, оцінювання проведено за 5-ти бальною шкалою.

Таблиця А.3

№	Можливий ризик	Ймовірність виникнення	Вплив на проект
R1	Помилки в постановці мети та задач проекту	1	4
R2	Некоректне планування – недотримання календарного плану	3	4
R3	Помилки в проектуванні – некоректний вибір програмного забезпечення	2	5
R4	Помилки в процесі безпосередньої реалізації програмного коду	2	4
R5	Некоректне тестування проекту	2	3
R6	Некоректне складення документації	2	1

За результатами аналізу можливих ризиків проекту створення мобільного чат бота, було створено матрицю ризиків для визначення зони ризику та подальшої роботи над можливими шляхами запобігання та роботи з ними. Матриця ризиків наведена у табл. А.4.

Таблиця А.4

Імовірність ризику	5					
	4					
	3				R2	
	2	R6		R5	R4	R3
	1				R1	
		1	2	3	4	5
Вплив ризику						

Завдяки матриці можливих ризиків було проведено класифікацію ризиків на прийнятні (зелений колір), виправдані (жовтий) та неприпустимі (червоні).

Наступним етапом управління ризиками є створення сценарію запобігання та усунення ризику. Результати розробки сценарію наведені у табл. А.5.

Таблиця А.5

№	Можливий ризик	Ідентифікація	Запобігання	Усунення ризику
R6	Некоректне складення документації	Прийнятний	Під час календарного планування відвести достатньо часу на складення документації та відвести час на можливі виправлення.	Консультація з керівником та замовником проекту задля виправлення помилок.
R5	Некоректне тестування проекту	Виправданій	Провести вивчення області тестування схожих проектів, проконсультуватися з незалежним тестувальником.	Залучення незалежного кваліфікованого спеціаліста

Продовження таблиці А.5

R1	Помилки в постановці мети та задач проекту	Виправданий	Мета та задачі проекту мають бути детально проаналізовані та узгоджені керівником, розробником та замовником проекту.	Визначити помилки в існуючих меті та задачах, зробити необхідні виправлення в документації та програмному коді проекту.
R4	Помилки в процесі безпосередньої реалізації програмного коду	Виправданий	Під час календарного планування відвести достатньо часу на вивчення середовища програмування, мови програмного коду та роботи над помилками, що виникають в ході розробки.	Залучення кваліфікованого спеціаліста для консультації розробника у раз виникнення критичних помилок.
R3	Помилки в проектуванні – некоректний вибір програмного забезпечення	Виправданий	Провести детальне вивчення існуючих аналогів програмного забезпечення, провести аналіз переваг та недоліків обраних ПЗ.	Регулярно виконувати перевірку ПЗ, зберігати кожен версію зі значним змінами в коді проекту.
R2	Некоректне планування – недотримання календарного плану	Виправданий	Затвердження календарного планування усіма учасниками проекту, детальне вивчення кожного етапу розробки проекту задля більш точного визначення потрібного відведення часу, якісний розподіл обов'язків між учасниками проекту та делегування обов'язків.	Переорганізація календарного планування таким чином, щоб загальний строк реалізації проекту залишався незмінним.

Додаток Б

Програмний код продукту

Файл main.py

```
import telebot
import random

bot = telebot.TeleBot('1364596186:AAECL3HEHUEkehCiba_IQoVWdW4-
5mm7k0')

from telebot import types

@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привіт! Давай готувати разом',
reply_markup=keyboard1)

keyboard1 = telebot.types.ReplyKeyboardMarkup(True, True)
keyboard1.row('Українська мова', 'English language')

@bot.message_handler(content_types=['text'])
def get_text_messages(message):
    if message.text == 'Українська мова':
        keyboard2 = telebot.types.InlineKeyboardMarkup()
```



```

        key_ukrcategory = telebot.types.InlineKeyboardButton (text = 'Обрати
страву за категорією', callback_data = 'key_ukrcategory')
        keyboard2.add(key_ukrcategory)
        key_ukringredient = telebot.types.InlineKeyboardButton(text='Обрати
страву за основним ігнредієнтом',
                                callback_data='key_ukringredient')
        keyboard2.add(key_ukringredient)
        bot.send_message(message.chat.id, text = 'Привіт! Обери команду
нижче v', reply_markup=keyboard2)

```

```

elif message.text == 'English language':
    keyboard3 = telebot.types.InlineKeyboardMarkup()
    key_engcategory = telebot.types.InlineKeyboardButton(text='Select
recipe category',
                                callback_data='key_engcategory')
    keyboard3.add(key_engcategory)
    key_engingredient = telebot.types.InlineKeyboardButton(text='Select
the main ingredient',
                                callback_data='key_engingredient')
    keyboard3.add(key_engingredient)
    bot.send_message(message.chat.id, text='Hi! Select the command lower
v', reply_markup=keyboard3)

```

```

@bot.callback_query_handler(func=lambda call:True)
def callback_worker(call):

```

```

    if call.data == 'key_ukringredient':

```

```

        bot.send_message(call.message.chat.id, text='Напиши основний
інгредієнт для страви:')

    elif call.data == 'key_engingredient':
        bot.send_message(call.message.chat.id, text='Write main ingredint
for your meal:')

    if call.data == 'key_ukrcategory':
        keyboard4 = telebot.types.InlineKeyboardMarkup()
        key_ukrvegeterian =
telebot.types.InlineKeyboardButton(text='Вегетеріанські страви',
                                    callback_data='key_ukrvegeterian')
        keyboard4.add(key_ukrvegeterian)
        key_ukrholiday =
telebot.types.InlineKeyboardButton(text='Святкові страви',
                                    callback_data='key_ukrholiday')
        keyboard4.add(key_ukrholiday)
        key_ukrfastfood = telebot.types.InlineKeyboardButton(text='Страви
нашвидкоруч',
                                                                callback_data='key_ukrfastfood')
        keyboard4.add(key_ukrfastfood)
        key_ukrrational = telebot.types.InlineKeyboardButton(text='Здорове
харчування',
                                                                callback_data='key_ukrrational')
        keyboard4.add(key_ukrrational)
        bot.send_message(call.message.chat.id, text='Категорії:',
reply_markup=keyboard4)

    elif call.data == 'key_engcategory':

```

```

keyboard5 = telebot.types.InlineKeyboardMarkup()
key_engvegeterian =
telebot.types.InlineKeyboardButton(text='Vegeterian meals',
                                   callback_data='key_engvegeterian')
keyboard5.add(key_engvegeterian)
key_englisholiday = telebot.types.InlineKeyboardButton(text='Holiday
dishes',
                                                         callback_data='key_englisholiday')
keyboard5.add(key_englisholiday)
key_engfastfood = telebot.types.InlineKeyboardButton(text='Fast
food',
                                                       callback_data='key_engfastfood')
keyboard5.add(key_engfastfood)
key_engrational = telebot.types.InlineKeyboardButton(text='Rational
nutrition',
                                                       callback_data='key_engrational')
keyboard5.add(key_engrational)
bot.send_message(call.message.chat.id, text='Categories:',
reply_markup=keyboard5)

```

```

if call.data == 'key_ukrvegeterian':
    keyboard6 = telebot.types.InlineKeyboardMarkup()
    key_ukrvegeterian_salads =
telebot.types.InlineKeyboardButton(text='Салати',
callback_data='key_ukrvegeterian_salads')
    keyboard6.add(key_ukrvegeterian_salads)

```

```

        key_ukrvegeterian_soups =
telebot.types.InlineKeyboardButton(text='Супи',

callback_data='key_ukrvegeterian_soups')
        keyboard6.add(key_ukrvegeterian_soups)
        key_ukrvegeterian_mainmeals =
telebot.types.InlineKeyboardButton(text='Основні страви',

callback_data='key_ukrvegeterian_mainmeals')
        keyboard6.add(key_ukrvegeterian_mainmeals)
        key_ukrvegeterian_desserts =
telebot.types.InlineKeyboardButton(text='Десерти',

callback_data='key_ukrvegeterian_desserts')
        keyboard6.add(key_ukrvegeterian_desserts)
        bot.send_message(call.message.chat.id, text='Вегетеріанські
страви:', reply_markup=keyboard6)

    elif call.data == 'key_engvegeterian':
        keyboard7 = telebot.types.InlineKeyboardMarkup()
        key_engvegeterian_salads =
telebot.types.InlineKeyboardButton(text='Salads',

callback_data='key_engvegeterian_salads')
        keyboard7.add(key_engvegeterian_salads)
        key_engvegeterian_soups =
telebot.types.InlineKeyboardButton(text='Soups',

callback_data='key_engvegeterian_soups')

```

```

        keyboard7.add(key_engvegeterian_soups)
        key_engvegeterian_mainmeals =
telebot.types.InlineKeyboardButton(text='Main meals',

callback_data='key_engvegeterian_mainmeals')
        keyboard7.add(key_engvegeterian_mainmeals)
        key_engvegeterian_desserts =
telebot.types.InlineKeyboardButton(text='Desserts',

callback_data='key_engvegeterian_desserts')
        keyboard7.add(key_engvegeterian_desserts)
        bot.send_message(call.message.chat.id, text='Vegeterian meals:',
reply_markup=keyboard7)

if call.data == 'key_ukrholiday':
    keyboard8 = telebot.types.InlineKeyboardMarkup()
    key_ukrholiday_salads =
telebot.types.InlineKeyboardButton(text='Салати',

callback_data='key_ukrholiday_salads')
        keyboard8.add(key_ukrholiday_salads)
        key_ukrholiday_soups =
telebot.types.InlineKeyboardButton(text='Супи',

callback_data='key_ukrholiday_soups')
        keyboard8.add(key_ukrholiday_soups)
        key_ukrholiday_mainmeals =
telebot.types.InlineKeyboardButton(text='Основні страви',

```

```

callback_data='key_ukrholiday_mainmeals')
        keyboard8.add(key_ukrholiday_mainmeals)
        key_ukrholiday_desserts =
telebot.types.InlineKeyboardButton(text='Десерти',

callback_data='key_ukrholiday_desserts')
        keyboard8.add(key_ukrholiday_desserts)
        bot.send_message(call.message.chat.id, text='Святкові страви:',
reply_markup=keyboard8)

elif call.data == 'key_englishholiday':
        keyboard9 = telebot.types.InlineKeyboardMarkup()
        key_englishholiday_salads =
telebot.types.InlineKeyboardButton(text='Salads',

callback_data='key_englishholiday_salads')
        keyboard9.add(key_englishholiday_salads)
        key_englishholiday_soups =
telebot.types.InlineKeyboardButton(text='Soups',

callback_data='key_englishholiday_soups')
        keyboard9.add(key_englishholiday_soups)
        key_englishholiday_mainmeals =
telebot.types.InlineKeyboardButton(text='Main meals',

callback_data='key_englishholiday_mainmeals')
        keyboard9.add(key_englishholiday_mainmeals)

```

```

        key_english_holiday_desserts =
telebot.types.InlineKeyboardButton(text='Desserts',

callback_data='key_english_holiday_desserts')
        keyboard9.add(key_english_holiday_desserts)
        bot.send_message(call.message.chat.id, text='Holiday meals:',
reply_markup=keyboard9)

```

```

if call.data == 'key_ukrfastfood':
        keyboard10 = telebot.types.InlineKeyboardMarkup()
        key_ukrfastfood_salads =
telebot.types.InlineKeyboardButton(text='Салати',

callback_data='key_ukrfastfood_salads')
        keyboard10.add(key_ukrfastfood_salads)
        key_ukrfastfood_soups =
telebot.types.InlineKeyboardButton(text='Супи',

callback_data='key_ukrfastfood_soups')
        keyboard10.add(key_ukrfastfood_soups)
        key_ukrfastfood_mainmeals =
telebot.types.InlineKeyboardButton(text='Основні страви',

callback_data='key_ukrfastfood_mainmeals')
        keyboard10.add(key_ukrfastfood_mainmeals)
        key_ukrfastfood_desserts =
telebot.types.InlineKeyboardButton(text='Десерти',

callback_data='key_ukrfastfood_desserts')

```

```

        keyboard10.add(key_ukrfastfood_desserts)
        bot.send_message(call.message.chat.id, text='Страви нашвидоруч:',
reply_markup=keyboard10)

```

```

elif call.data == 'key_engfastfood':
    keyboard11 = telebot.types.InlineKeyboardMarkup()
    key_engfastfood_salads =
telebot.types.InlineKeyboardButton(text='Salads',

```

```

callback_data='key_engfastfood_salads')
        keyboard11.add(key_engfastfood_salads)
        key_engfastfood_soups =
telebot.types.InlineKeyboardButton(text='Soups',

```

```

callback_data='key_engfastfood_soups')
        keyboard11.add(key_engfastfood_soups)
        key_engfastfood_mainmeals =
telebot.types.InlineKeyboardButton(text='Main meals',

```

```

callback_data='key_engfastfood_mainmeals')
        keyboard11.add(key_engfastfood_mainmeals)
        key_engfastfood_desserts =
telebot.types.InlineKeyboardButton(text='Desserts',

```

```

callback_data='key_engfastfood_desserts')
        keyboard11.add(key_engfastfood_desserts)
        bot.send_message(call.message.chat.id, text='Fast cooking:',
reply_markup=keyboard11)

```



```

        if call.data == 'key_ukrrational':
            keyboard12 = telebot.types.InlineKeyboardMarkup()
            key_ukrrational_salads =
telebot.types.InlineKeyboardButton(text='Салати',

callback_data='key_ukrrational_salads')
            keyboard12.add(key_ukrrational_salads)
            key_ukrrational_soups =
telebot.types.InlineKeyboardButton(text='Супи',

callback_data='key_ukrrational_soups')
            keyboard12.add(key_ukrrational_soups)
            key_ukrrational_mainmeals =
telebot.types.InlineKeyboardButton(text='Основні страви',

callback_data='key_ukrrational_mainmeals')
            keyboard12.add(key_ukrrational_mainmeals)
            key_ukrrational_desserts =
telebot.types.InlineKeyboardButton(text='Десерти',

callback_data='key_ukrrational_desserts')
            keyboard12.add(key_ukrrational_desserts)
            bot.send_message(call.message.chat.id, text='Здорове
харчування:', reply_markup=keyboard12)

        elif call.data == 'key_engrational':
            keyboard13 = telebot.types.InlineKeyboardMarkup()
            key_engrational_salads =
telebot.types.InlineKeyboardButton(text='Salads',

```

```

callback_data='key_engrational_salads')
        keyboard13.add(key_engrational_salads)
        key_engrational_soups =
telebot.types.InlineKeyboardButton(text='Soups',

```

```

callback_data='key_engrational_soups')
        keyboard13.add(key_engrational_soups)
        key_engrational_mainmeals =
telebot.types.InlineKeyboardButton(text='Main meals',

```

```

callback_data='key_engrational_mainmeals')
        keyboard13.add(key_engrational_mainmeals)
        key_engrational_desserts =
telebot.types.InlineKeyboardButton(text='Desserts',

```

```

callback_data='key_engrational_desserts')
        keyboard13.add(key_engrational_desserts)
        bot.send_message(call.message.chat.id, text='Rational nutrition:',
reply_markup=keyboard13)

```

```

@bot.message_handler(content_types=['sticker'])

```

```

def sticker_id(message):

```

```

    print(message)

```

```

bot.polling()

```