

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ  
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

**ЗВІТ**

**з технологічної практики**

«Розробка і супроводження програмного продукту»

на тему Telegram бот для прийняття заявок від клієнтів

Виконав: здобувач освіти 4 курсу,

групи 45

спеціальності 121

Інженерія програмного забезпечення

Патьоха М.В.

(прізвище та ініціали)

Керівник Серебреннікова І.В.

(підпис)

(прізвище та ініціали)

Полтава – 2023

## ЗМІСТ

ВСТУП .....	3
1. ПОСТАНОВКА ЗАВДАННЯ .....	4
1.1. Основні вимоги до продукту .....	8
1.2. Вимоги до інтерфейсу.....	10
2. ПЛАНУВАННЯ СИСТЕМИ .....	12
2.1. Архітектура системи.....	13
2.2. Тестування .....	14
2.3. Інструкція з використання системи.....	15
ВИСНОВКИ.....	18
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	19
ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ .....	20
ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ .....	21
ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ .....	22
ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ (ТА/АБО ІНШІ).....	23
ДОДАТОК Д. ВИХІДНІ КОДИ.....	25
ДОДАТОК Е. ЗНІМКИ ЕКРАНУ .....	27

## ВСТУП

Місцем моєї технологічної практики була компанія СТ Інфотех. Компанія працює на ринку послуг з 3 квітня 2013 року.

Основними напрямками її діяльності є діяльність із керування комп'ютерним устаткуванням.

Послуги, що надає СТ Інфотех можна класифікувати наступним чином:

- Електромонтажні роботи;
- Оптова торгівля комп'ютерами, периферійним устаткуванням і програмним забезпеченням;
- Обслуговування систем безпеки;
- Ремонт комп'ютерів і периферійного устаткування;
- Ремонт обладнання зв'язку;
- Комп'ютерне програмування
- Консультування з питань інформатизації
- Ремонт і технічне обслуговування електронного й оптичного устаткування

В ході практики я ознайомився з роботою підприємства, де переді мною було поставлено завдання розробити Telegram бота для замовлення послуг в компанії Інфотех.

Для виконання поставленого завдання мені стали в нагоді знання, вміння й навички та знання з мови програмування Python в особливості знання математичних функцій та взаємодії з таблицями.

Під час практики я:

- Працював з Telegram API;
- Вивчав мову програмування Python;
- Навчився взаємодіяти з API;
- Удосконалив навички логічного мислення;
- Навчився використовувати Telegram боти;

## 1. ПОСТАНОВКА ЗАВДАННЯ

Я отримав завдання розробити Telegram бота для замовлення у компанії СТ Інфотех тих послуг які хоче клієнт.

Були поставленні такі вимоги:

- Розробити Telegram бота;
- Використовувати сучасні технології для розробки;
- Використовувати сучасний дизайн;
- Можна використовувати, як шаблони так і свій програмний код.

Терміни розробки 2 неділі. Функціональність програми повин....  
Середовище і мова розробки PyCharm[1].

PyCharm - одна з найпопулярніших IDE для Python(рис 1.1).



Рис. 1.1 – PyCharm

Для цього є багато причин, у тому числі той факт, що він був розроблений JetBrains, [2] розробником популярної IntelliJ IDEA, яка є однією з трьох основних середовищ Java IDE і найбільш "розумною" середовищем Javascript WebStorm. Підтримка веб-розробки за допомогою Django-ще одна вагома причина.

Потрібно зробити бота якомога простим щоб легко покращити або додати більше функції в майбутньому.

У ході дослідження було виділено такі аналоги:

### PERSONALFINTRACKERBOT

Telegram-бот PersonalFinTrackerBot [3] допомагає вести облік власних доходів і витрат в Google-таблиці. Бот доступний українською та російською мовою, підтримує кілька валют, дозволяє вести бюджет спільно з родиною та зберігати дисконтні карточки.

По суті, при запуску бота створюється Google-таблиця, в яку зручно через бот можна вносити витрати чи доходи за категоріями. Можна окремо отримати доступ до самої Google-таблиці і працювати з нею окремо – для цього доведеться вказати справжню адресу на Gmail. Втім, працювати з таблицею на телефоні не зручно, тож бот нівелює це і дозволяє оперативно вносити дані в звичному інтерфейсі.

Як працює та які функції

- вводите цифру і бот пропонує обрати категорію витрат для внесення або ж категорію доходу.
- бот розпізнає символи валют і може здійснювати автоматичний перерахунок в стандартну валюту
- можна в боті відразу подивитися витрати та доходи за категоріями
- також підтримується зберігання дисконтних карток. Для цього потрібно відправити фото з її штрихкодом
- картки також будуть доступні на людей, з якими спільний бюджет. По факту, при запиті вам потім в бот присилається фото картки.

Зауважимо, що таблиця при запуску бота, вочевидь, створюється в акаунті розробника. Тож він може бачити всі ваші доходи та витрати. Навіть якщо ви не будете надсилати власний емейл, то він також зможе бачити ваше ім'я в Telegram. Втім, поки уявити якісь ризики важко і навіщо комусь витрати на “Їжу поза домом” чи доходи без жодної конкретизації. Але у випадку розширення функціоналу, про ці ризики варто пам'ятати. Приклад інтерфейсу чат-бота зображено на рисунках 1.2.

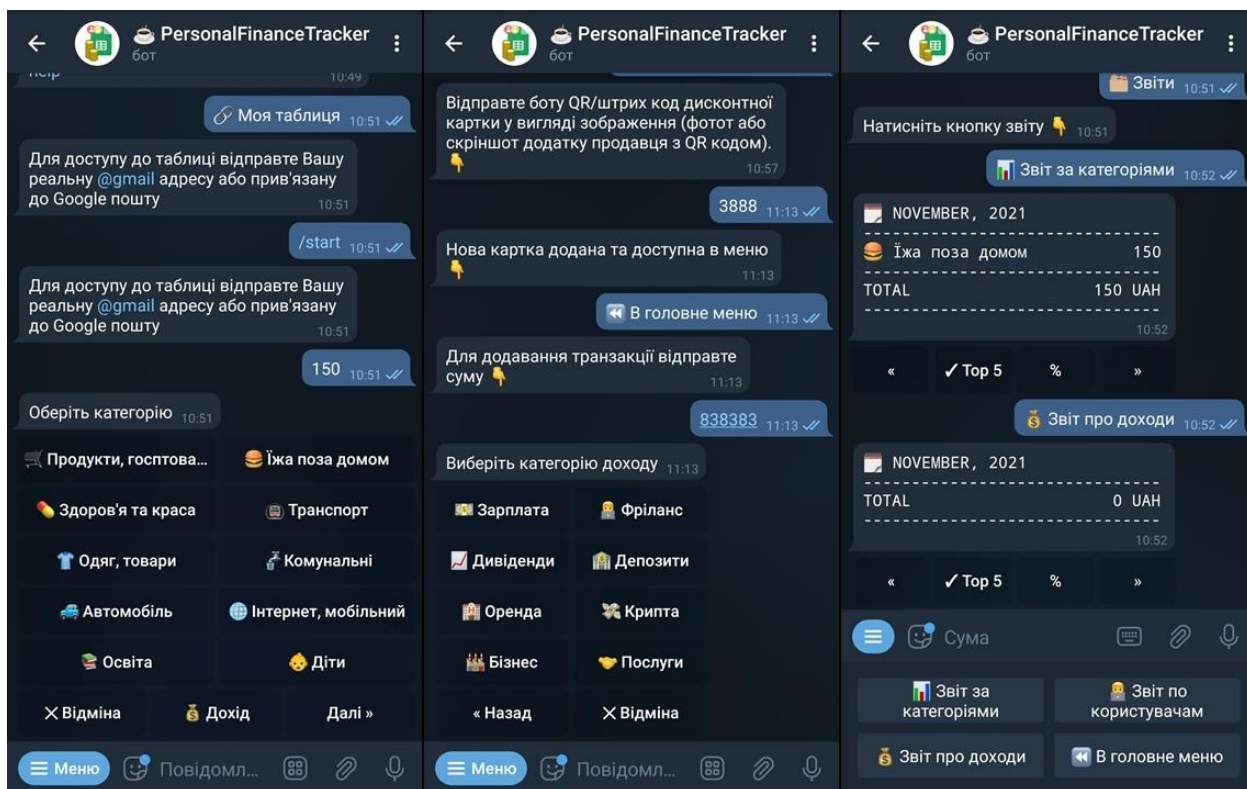


Рисунок 1.2 – Інтерфейс чат-бота

## MOEMISTO

MoeMisto.ua [4] – бот, який надає можливість користувачеві отримувати інформацію про всі заходи у місті. Можливості бота містять пошук та підбір анонсів, подій та заходів по відповідним параметрам, а саме:

- пошук анонсів по даті проведення заходу;
- підбір анонсів по тематичним рубрикам, таким як концерти, вечірки, дитячі заходи, кіно, подорожі;
- афіші розваг та відпочинку у закладах чи по місцях проведення.

Чат-бот надає можливість пошуку подій у таких містах України як: Вінниця, Житомир, Тернопіль, Одеса, Хмельницький, Київ, Львів, Кропивницький.

Приклад інтерфейсу бота MoeMisto.ua зображено на рисунку 1.3.

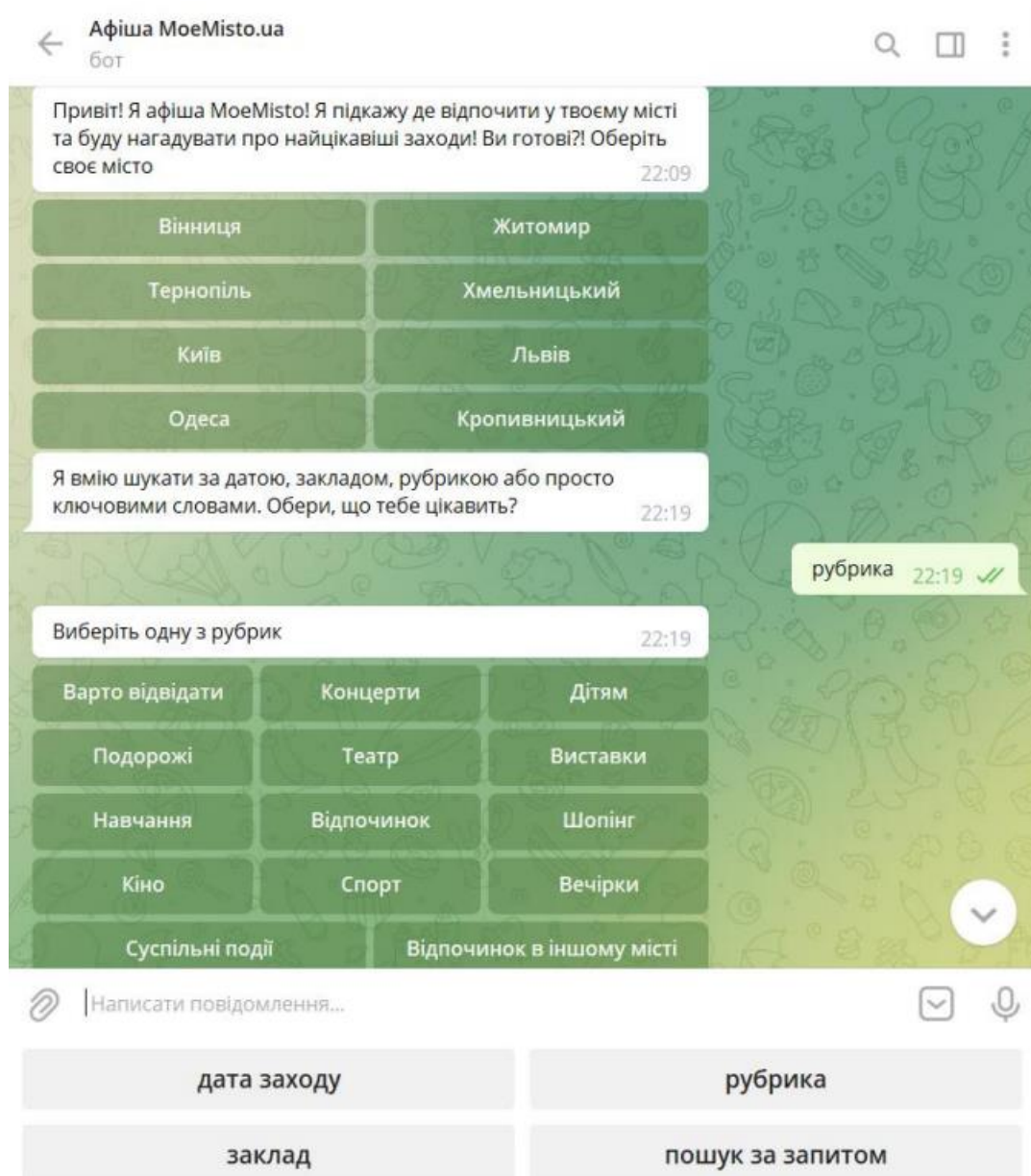


Рисунок 1.3 – Приклад інтерфейсу бота MoeMisto.ua  
MUSICTRIPPLANNER

Music Trip Planner [5] – бот для пошуку концертів музичних гуртів. Пошук здійснюється по назві гурту або місці проведення концерту. На рисунку 1.4 зображено приклад інтерфейсу чат-бота.

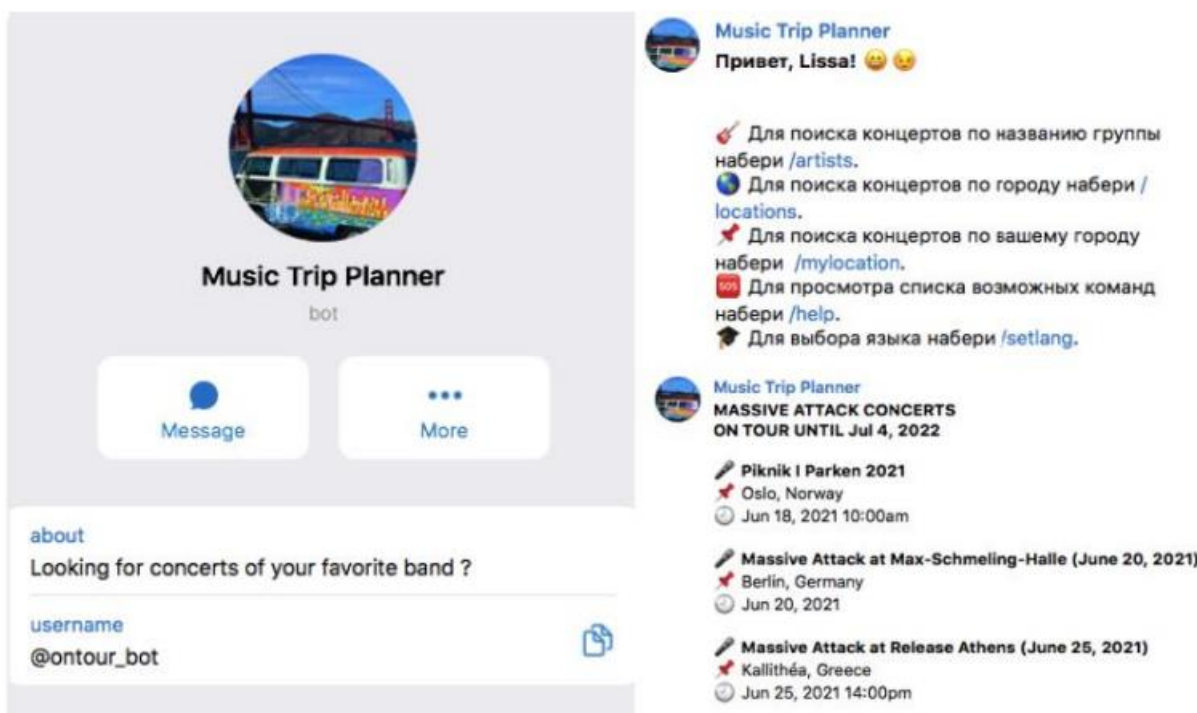


Рисунок 1.4 – Інтерфейс Music Trip Planner бота

Порівняльну характеристику розглянутих чат-ботів зображено в додатку А

### 1.1. Основні вимоги до продукту

Постановка вимог до програмного забезпечення (ПЗ) є одним з перших та найголовніших кроків. Маючи певні завдання, які повинна виконувати система, програміст розуміє, яким чином потрібно організувати роботу над програмою. Вимоги допомагають продумати функціональні та нефункціональні можливості ПЗ. При проектуванні будь-якого проекту потрібно виділити сферу діяльності, в якій буде працювати програма, та основні чинники, а саме роботоспроможність та відмовостійкість ПЗ, щоб уникнути моментів «падіння» та припинення функціонування сервісу.

Визначення головних задач, які повинні виконуватись системою, мають передувати початку проектування чат-бота, тому що неясність у вимогах може призвести до проблематики в реалізації сервісу.

Необхідні кроки, які потрібно зробити для постановки вимог:

- зрозуміти структуру та форму системи, спрямувавши увагу на її функціях та розгляд того, як вони задовільняють та виконують запити користувачів;



- оцінити вичерпність, обґрунтованість системи та відповідність для середовища, в якому програмне забезпечення буде використовуватись;
- використовувати дані про модель, що створюється, та інформацію з пунктів вище для визначення можливості побудови сервісу та управління його масштабом перед тим, як робити значні інвестиції до проекту.

Вимоги користувача до системи зручно представляти з допомогою діаграми прецедентів.

Розроблену діаграму прецедентів взаємодії користувача з телеграм ботом наведено в додатку Б.

#### ПРЕЦЕДЕНТ «СТАРТ ТЕЛЕГРАМ БОТУ»

Актор: користувач.

Передумова: доступ до телеграм бота.

Післяумова – введення даних.

Сценарій:

- Перейти за посиланням до бота.
- Нажати кнопку /start;
- Вибрати вид послуги;
- Ввести ПІБ;
- Ввести номер телефона;

#### ПРЕЦЕДЕНТ «ПЕРЕГЛЯД GOOGLE ТАБЛИЦІ З ДАНИМИ»

Актор: адміністратор.

Передумова: зайти в Google акаунт до якого прив'язана Google таблиця.

Післяумова: доступ до даних користувачів

Сценарій:

- Перейти до сторінки входу в Google акаунт.
- Ввести дані.
- Знайти таблицю та відкрити.
- Моніторити та перевіряти їх.
- Якщо що зв'язатися з користувачем.

## ФУНКЦІОНАЛЬНІ ВИМОГИ:

Функціональні вимоги (Functional Requirements) [6] описують як саме веде себе сервіс в експлуатаційний період. Ці потреби зазвичай орієнтовані на дії, які виконує користувач, та як система на це реагує.

Для чат бота функціональні вимоги потрібно виділити для клієнта та для менеджера. Кожен з них має свій варіант взаємодії з сервісом і тому потрібно описати потреби в залежності від ролі користувача системи.

Клієнт повинен мати змогу:

- заповняти дані, які просить бот;
- оформити замовлення;

Менеджер повинен мати змогу:

- передивлятися Google таблицю з даними клієнтів;

## НЕФУНКЦІОНАЛЬНІ ВИМОГИ:

Нефункціональні вимоги (Non-functional Requirements) [7] – це вимоги, які, як правило, використовують для опису деяких «атрибутів системи» або «атрибути системного середовища».

До них відносять:

- атрибути якості;
- обмеження в проектуванні;
- реалізація, зовнішні або прикладні програмні інтерфейси (ППІ).

Якщо звернутись до класифікації вимог до програмних систем (FURPS), то до нефункціональних вимог належать:

- практичність (usability);
- надійність (reliability);
- продуктивність (performance);
- можливість підтримки (supportability).

### 1.2. Вимоги до інтерфейсу

Вимога практичності спрямована на створення простого у використанні інтерфейсу програмного забезпечення. Необхідно описати, наскільки просто

новий користувач може оволодіти сервісом і чи не принесе це труднощів, тому важливо додати побільше текстових повідомлень з описом елементів або кроків, які необхідно зробити покупцеві, щоб досягти поставленої цілі.

Також варто порівняти практичність використання створеної системи з працюючими схожими сервісами, які вже відомі та здобули у користувачів певного успіху. Це порівняння здійснюється через досвід користувача (UX). Потрібно віднайти вже існуючі системи інтерактивних підказок, програм помічників, засобів попередження, посібників для користувача та інших форм документації можливостей програмного забезпечення для спрощення орієнтації по сервісу для клієнта.

Також важливим є використання угод та стандартів, які спрямовані на опис розробки людино-машинного інтерфейсу. Це необхідно, щоб сервіс працював не гірше аналогів та мав індивідуальні якості, які будуть надавати йому вираженого повноцінного вигляду та виділятимуть серед інших створених систем.

Визначивши вимоги до інтерфейсу було створено прототип інтерфейсу (Додаток В).

## 2. ПЛАНУВАННЯ СИСТЕМИ

Поняття життєвого циклу розробки ПЗ відоме багатьом. Він складається з 6 етапів:

- Дослідження та аналіз.
- Планування.
- Проектування.
- Реалізація.
- Тестування та інтеграція.
- Розгортання та обслуговування.

### ДОСЛІДЖЕННЯ ТА АНАЛІЗ

Розробка телеграм боту завжди починається із дослідження. На цьому етапі збираємо додаткову інформацію, дивимося подібні боти та як вони працюють це стане в нагоді при плануванні.

### ПЛАНУВАННЯ

На другому етапі процесу розробки починаємо планувати всі складові проекту: тип боту та його характеристики, кількість користувачів, що відбуватиметься, цільова аудиторія, наявність статичного або динамічного контенту, використовувані технології.

### ПРОЕКТУВАННЯ

На цьому етапі вже відомо, що саме створювати, хто цільова аудиторія і який план дій щодо розробки самого боту. А якщо відмінно попрацювали на двох попередніх етапах, то у вас вже є чітке уявлення про тексти та майбутню логіку нашого боту.

### РЕАЛІЗАЦІЯ

На попередньому етапі визначилися, як має виглядати бот, і тепер саме час зайнятися реалізацією.

### ТЕСТУВАННЯ ТА ІТЕГРАЦІЯ

Цей етап має важливе значення для успішної розробки боту. Вся виконана до цього моменту робота втрачає сенс, якщо продукт функціонує не

так, як замислювалося на користь користувачів. Етап включає низку аспектів фази розробки, які розглядаються більш цілеспрямовано.

## РОЗГОРТАННЯ ТА ОБСЛУГОВУВАННЯ

Як правило, обслуговування включає аспекти етапів розробки та тестування життєвого циклу. Під час розробки нових функціональностей проводяться регресійні тести задля забезпечення сумісності. Для програмного продукту було обрано каскадну модель життєвого циклу. Продовження процесу виконання реалізується за допомогою впорядковано послідовності кроків.

В моделі передбачено, що кожна наступна фаза починається лише тоді коли повністю завершено виконання попередньої фази. Кожна фаза має певні критерії входу та виходу: вхідні та вихідні дані. В результаті виконання генеруються внутрішні та зовнішні дані проекту, включаючи документацію та ПЗ.

На даний момент часу проект на етапі супроводження.

### 2.1. Архітектура системи

Архітектура чат-бота представляє концепцію, яка визначає структуру і взаємозв'язки між його компонентами [8].

До компонентів архітектури чат-бота відносяться:

- Telegram – месенджер, що є платформою для розробки чат-бота;
- Telegram Bot API – є посередником між Telegram та обробником команд, саме він отримує команди, у вигляді повідомлень, від користувача, а також відправляє відповідь на ці команди користувачу;
- Обробник команд отримує команди від Telegram Bot API і за допомогою прописаних сценаріїв та бази даних відправляє відповідь до Telegram Bot API.

Архітектура чат-бота зображена на рисунку 2.1

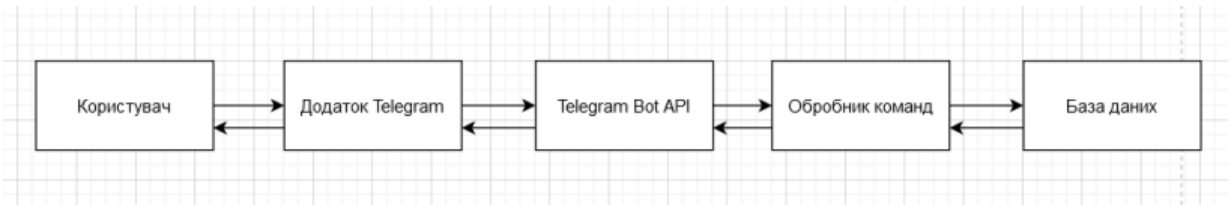


Рисунок 2.1 – Архітектура чат-бота

Також архітектуру Telegram боту описують (Додаток Б), (Додаток В), (Додаток Г), (Додаток Д).

## 2.2. Тестування

Завершальним етапом реалізації проекту є тестування готового продукту. Воно представляє собою процес перевірки та оцінки того, як створений програмний продукт виконує те, що має робити. За допомогою тестування є можливість запобігти виникненню помилок під час функціонування розробленого продукту. У таблиці 4.1 наведено тестування готового продукту з застосуванням набору test cases.

Таблиця 2.1 - Test cases для навчального чат-бота

Опис Test Case	Послідовність кроків	Очікуваний результат	Результат тестування
Перевірка відображення необхідних даних про види діяльності, ПІБ, номер телефону	1. Запустити чат-бот командою /start 2. Обрати необхідний вид діяльності 3. Ввести ПІБ 4. Ввести номер телефону	Всі дані відображаються коректно	Пройдено
Перевірка відображених даних в Google таблицях	1. Війти в свій Google аккаунт на компютері	Всі дані відображаються коректно	Пройдено

	2. Перейти на сайт Google таблиць		
	3. Перевірити дані		

Отже, під час тестування готового продукту помилок не було виявлено

### 2.3. Інструкція з використання системи

1. Заходимо в Telegram та шукаємо там нашого бота (<https://t.me/inphotechposlugybot>)

2. У вікні що з'явилося вводим команду /start. Див рис. 2.2.

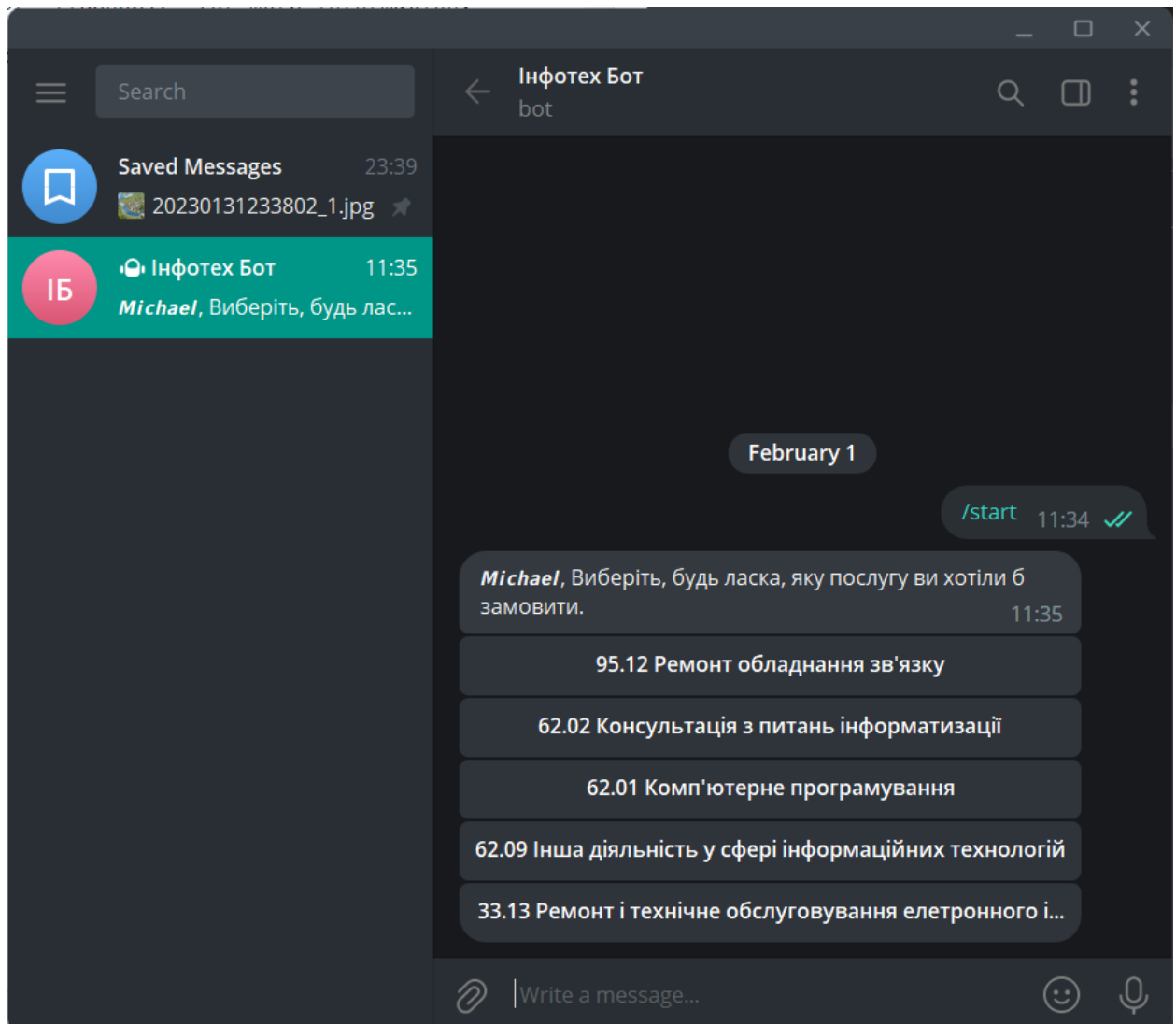


Рисунок 2.2 – Старт боту

3. Наступним чином вибираємо вид діяльності яку хочете замовити.

4. Потім потрібно поділитися номером телефону з ботом. Див рис. 2.3

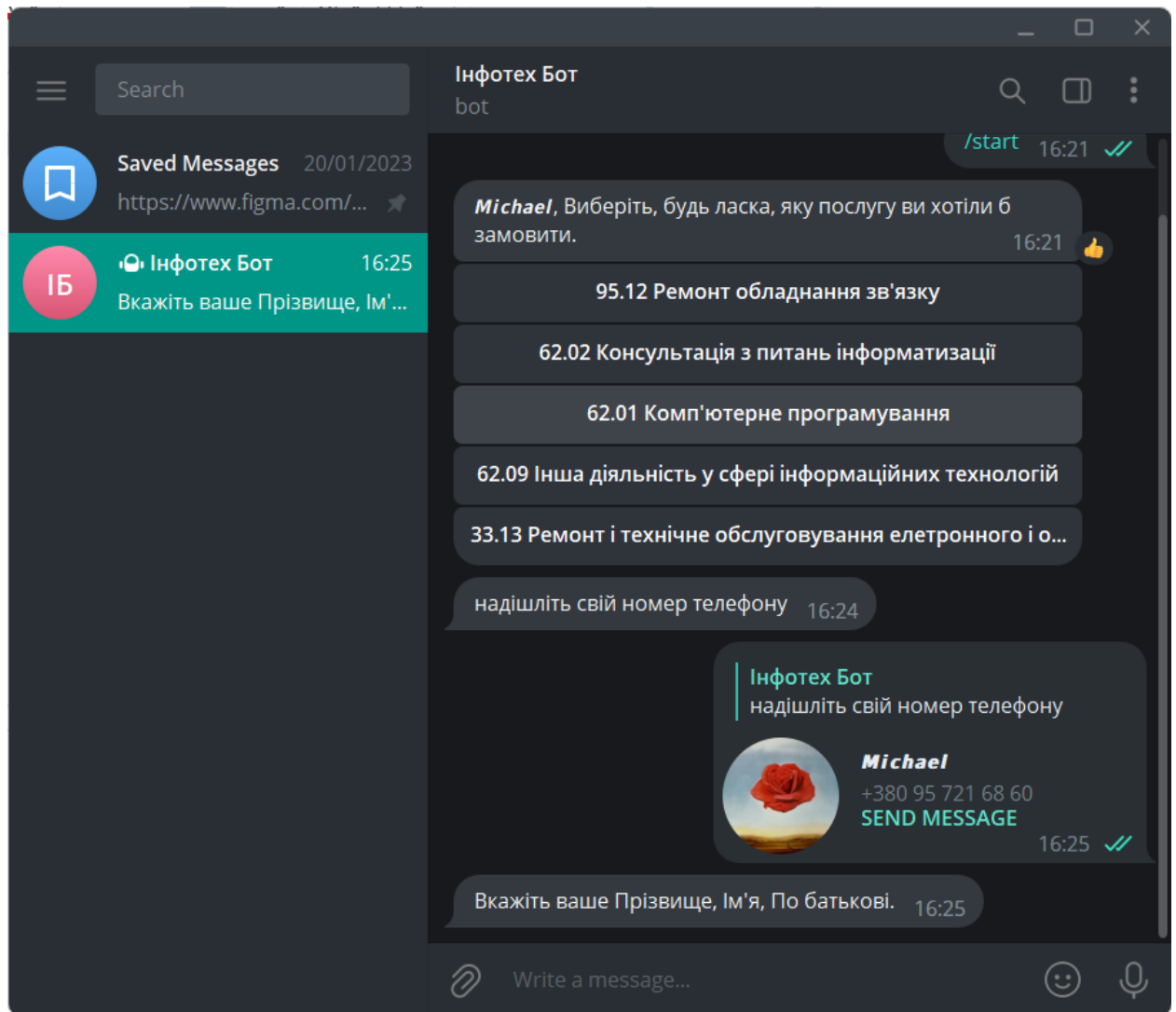


Рисунок 2.3 – Введення ПІБ

4. Наступним чином треба написати своє Прізвище, Ім'я, по батькові.  
Див рис. 2.4



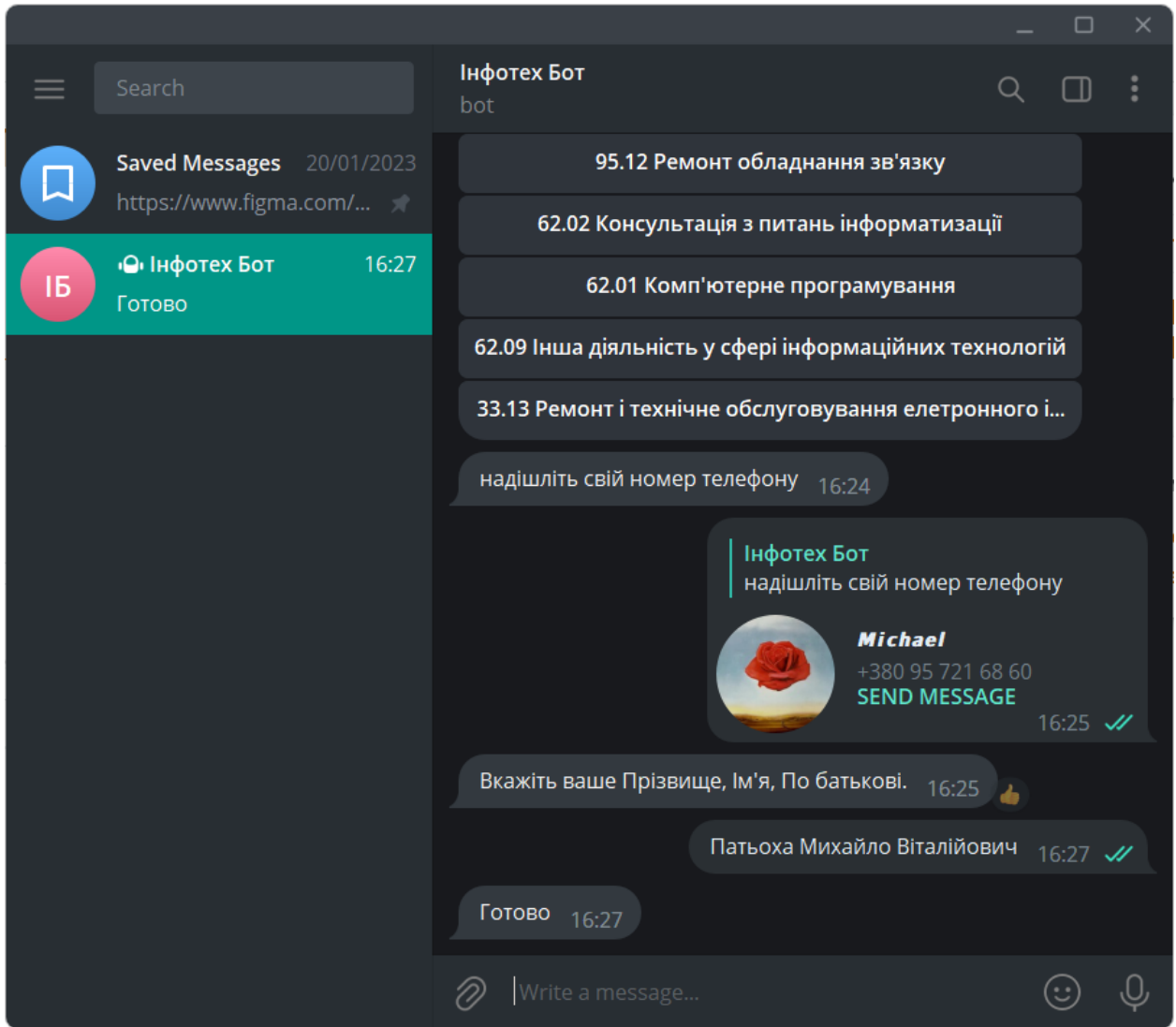


Рисунок 2.4 – Введення ПІБ

5. Дані будуть записані до Google таблиці. Див рис. 2.5.

Інфотех бот					
файл Правка Вид Вставка Формат Данные Инструменты Расширения Справка Посл					
100% р. % .0 .00 123 По умолча... 10 B I S A					
D8	fx				
	A	B	C	D	E
1	ПІБ	Вид діяльності	Номер телефону		
2	Патьоха Михайло Віталійович	1	+380957216860		
3	Думініке Вікторія Сергіївна	3	380663279406		
4	Патьоха Михайло Віталійович	2	+380957216860		
5					
6					
7					
8					
9					
10					
11					
12					

Рисунок 2.5 – Данні в гугл таблиці

## ВИСНОВКИ

Місцем моєї технологічної практики була компанія СТ Інфотех.

Переді мною була поставлена задача розробити Telegram бота для запису виду діяльності яку хотіли б замовити в компанії.

Мій Telegram бот використовує в якості API для заповнення таблиць даними про відповіді клієнтів.

Telegram бот не є частиною дипломного проекту.

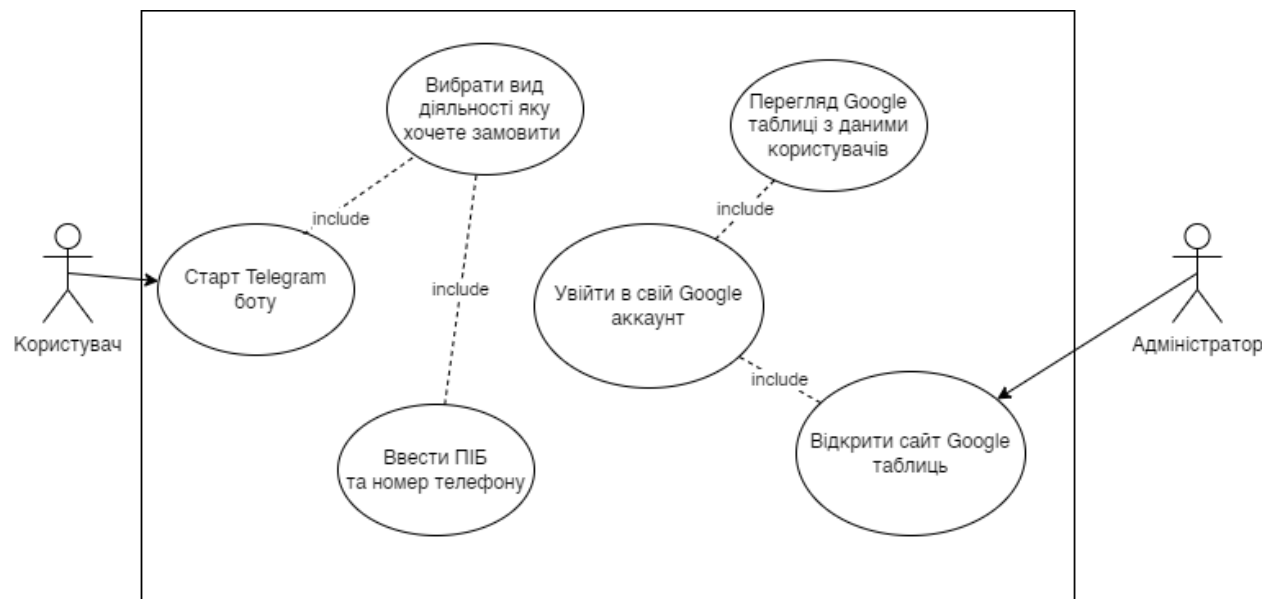
Я отримав навички роботи з мовою програмування Python, яка допомагала зручно розробляти бота. Познайомився з IDE Pycharm покращило швидкість розробки боту. Навчився використовувати Telegram API, який являє собою зручний та простий інструмент для розробки Telegram бота. Також я покращив свої знання з Google API, а саме Google Sheets API, завдяки чому вдалося передавати дані з боту напряму в Google таблицю.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. PyCharm [Електронний ресурс]. – Режим доступу: URL: <https://apricode.com/uk/razrabotka/vidy-sajtov/> (дата звернення 11.01.2023). – Назва з екрану.
2. JetBrains [Електронний ресурс]. – Режим доступу: URL: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/> (дата звернення 16.01.2023). – Назва з екрану.
3. PersonalFinTrackerBot [Електронний ресурс]. – Режим доступу: URL: <https://t.me/PersonalFinTrackerBot> (дата звернення 18.01.2023). – Назва з екрану.
4. МоеMisto [Електронний ресурс]. – Режим доступу: URL: [https://t.me/moemisto\\_vn\\_bot](https://t.me/moemisto_vn_bot) (дата звернення 18.01.2023). – Назва з екрану.
5. Music Trip Planner [Електронний ресурс]. – Режим доступу: URL: [https://t.me/ontour\\_bot](https://t.me/ontour_bot) (дата звернення 18.01.2023). – Назва з екрану.
6. Функціональні вимоги [Електронний ресурс]. – Режим доступу: URL: [https://er.nau.edu.ua/bitstream/NAU/54311/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86\\_2021\\_122\\_%D0%9C%D0%B0%D1%81%D0%BD%D0%B8%D0%B9%D0%9C%D0%92.pdf](https://er.nau.edu.ua/bitstream/NAU/54311/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86_2021_122_%D0%9C%D0%B0%D1%81%D0%BD%D0%B8%D0%B9%D0%9C%D0%92.pdf) (дата звернення 18.01.2023). – Назва з екрану.
7. Нефункціональні вимоги [Електронний ресурс]. – Режим доступу: URL: [https://er.nau.edu.ua/bitstream/NAU/54311/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86\\_2021\\_122\\_%D0%9C%D0%B0%D1%81%D0%BD%D0%B8%D0%B9%D0%9C%D0%92.pdf](https://er.nau.edu.ua/bitstream/NAU/54311/1/%D0%A4%D0%9A%D0%9A%D0%9F%D0%86_2021_122_%D0%9C%D0%B0%D1%81%D0%BD%D0%B8%D0%B9%D0%9C%D0%92.pdf) (дата звернення 18.01.2023). – Назва з екрану.
8. Understanding The Conversational Chatbot [Електронний ресурс].  
– Точка доступу: URL: <https://blog.vsoftconsulting.com/blog/understanding-thearchitecture-of-conversational-chatbot>

**ДОДАТОК А. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА  
ПРОГРАМНИХ ПРОДУКТІВ АНАЛОГІЧНОГО  
ПРИЗНАЧЕННЯ**


Категорії	PersonalFinTracker	МоеMisto	MusicTripPlaner	Свій програмний продукт
Наявність української мови	+	+	+	+
Інтуїтивний дизайн	+	+	—	+
Швидка функціональність	-	—	+	+

**ДОДАТОК Б. UML ДІАГРАМА ПРЕЦЕДЕНТІВ**

ДОДАТОК В. ПРОТОТИП ІНТЕРФЕЙСУ

15:24

←



InphotechBot

бот

⋮

/start

Михайле, добрий день. Виберіть, будь ласка, яку послугу ви б хотіли замовити

95.12 Ремонт обладнання зв'язку

62.02 Консультація з питань інформатизації

62.01 Комп'ютерне програмування

62.09 Інша діяльність у сфері інформаційних технологій

33.13 Ремонт і технічне обслуговування електронного і оптичного устаткування

Написати повідомлення...

ДОДАТОК Г. UML ДІАГРАМА КЛАСІВ

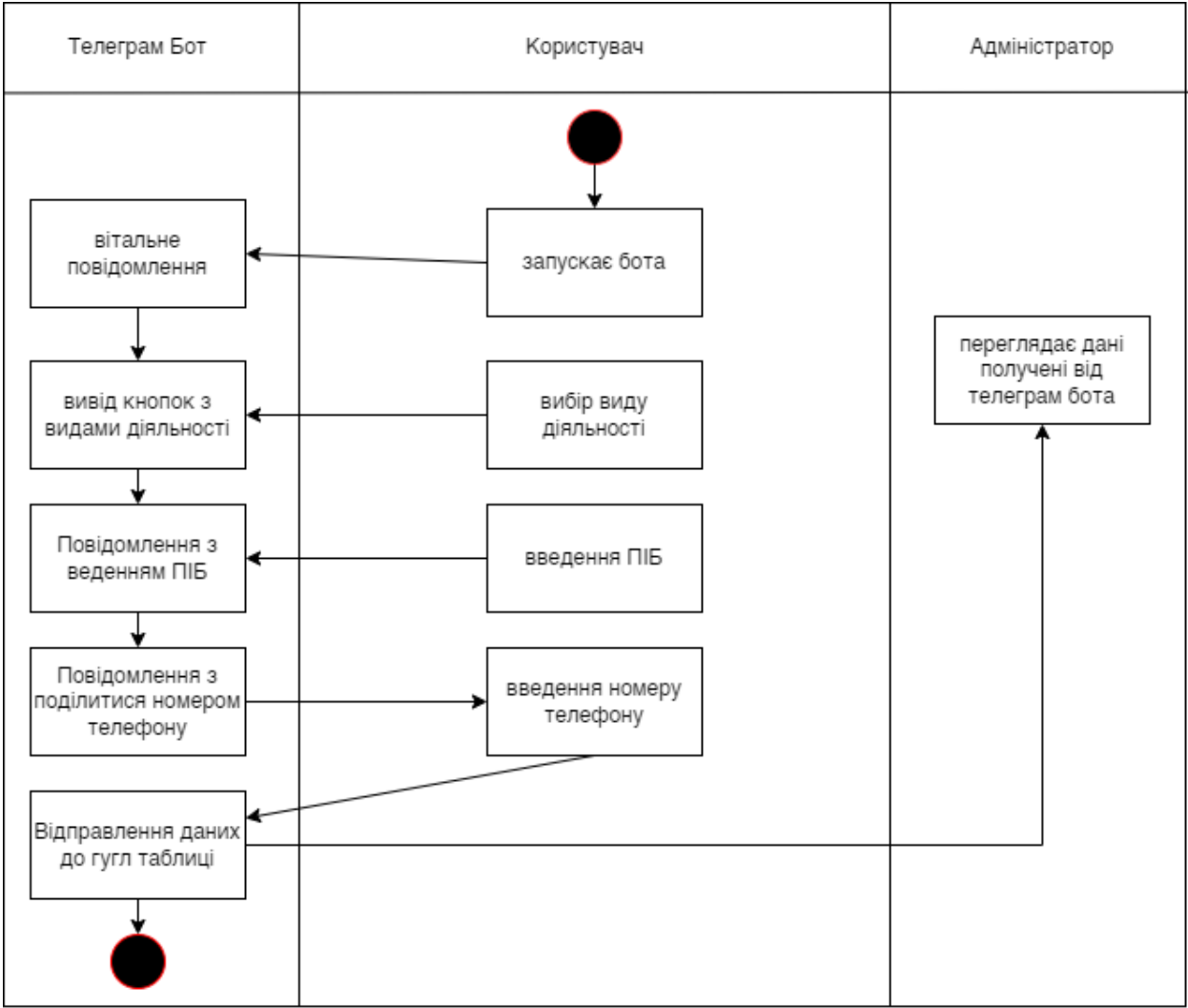


Рисунок Г.1 – Діаграма діяльності

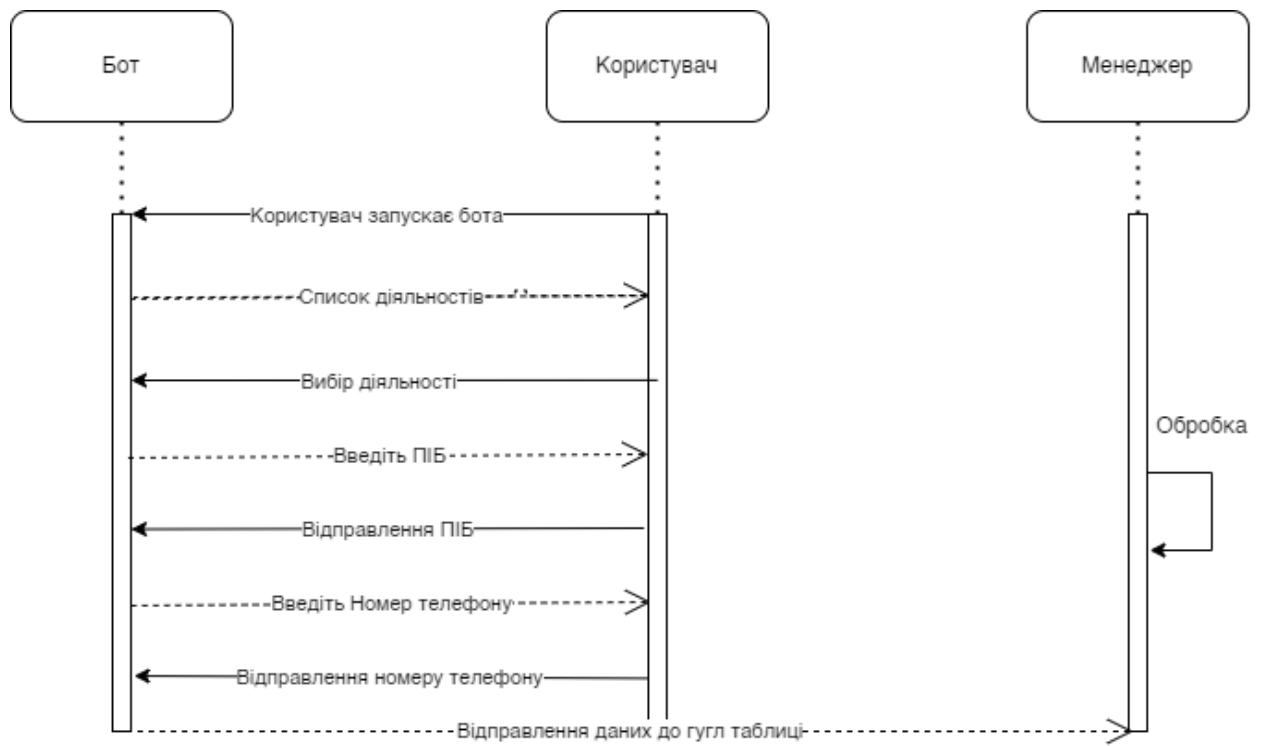


Рисунок Г.2 – Діаграма послідовності



## ДОДАТОК Д. ВИХІДНІ КОДИ

```

import telebot
from telebot import types
import httpplib2
import apiclient.discovery
from oauth2client.service_account import ServiceAccountCredentials
CREDENTIALS_FILE = 'annular-text-376213-d643b2a6f2e7.json'
credentials =
ServiceAccountCredentials.from_json_keyfile_name(CREDENTIALS_FILE,
['https://www.googleapis.com/auth/spreadsheets',
'https://www.googleapis.com/auth/drive'])
httpAuth = credentials.authorize(httpplib2.Http())
service = apiclient.discovery.build('sheets', 'v4', http = httpAuth)
spreadsheetId='1MHJMiQkmaeNNazqzYvQ7i-SjDzZDXyrBS6aKludMFto'

data=[]
def find_type_by_id(id_to_find):
    global data
    return next(filter(lambda x: x['id'] == id_to_find, data),
None) ['type']

def find_number_by_id(id_to_find):
    global data
    return next(filter(lambda x: x['id'] == id_to_find, data),
None) ['number']

def setnumber(id_to_set, new_type):
    global data
    for element in data:
        if element['id'] == id_to_set:
            element['number'] = new_type
    return data

token='5631969958:AAHxm0IpX61JYtvMKGrS6lYqrzgh5QtatbE'
bot=telebot.TeleBot(token)
@bot.message_handler(commands=['start'])
def start(message):
    buttons=[
        ["95.12 Ремонт обладнання зв'язку"],
        ["62.02 Консультація з питань інформатизації"],
        ["62.01 Комп'ютерне програмування"],
        ["62.09 Інша діяльність у сфері інформаційних технологій"],
        ["33.13 Ремонт і технічне обслуговування електронного і оптичного
устаткування"]
    ]
    keyboard = types.InlineKeyboardMarkup()
    b=0
    for button in buttons:
        b+=1
        keyboard.add(types.InlineKeyboardButton(text=str(button[0]),
callback_data=str(b)))
    bot.send_message(message.chat.id,f"{message.from_user.first_name},
Виберіть, будь ласка, яку послугу ви хотіли б замовити.",
reply_markup=keyboard)

@bot.callback_query_handler(func=lambda call: True)
def query_handler(call):
    global data
    data.append({'id':call.message.chat.id,'type':call.data})

```

```

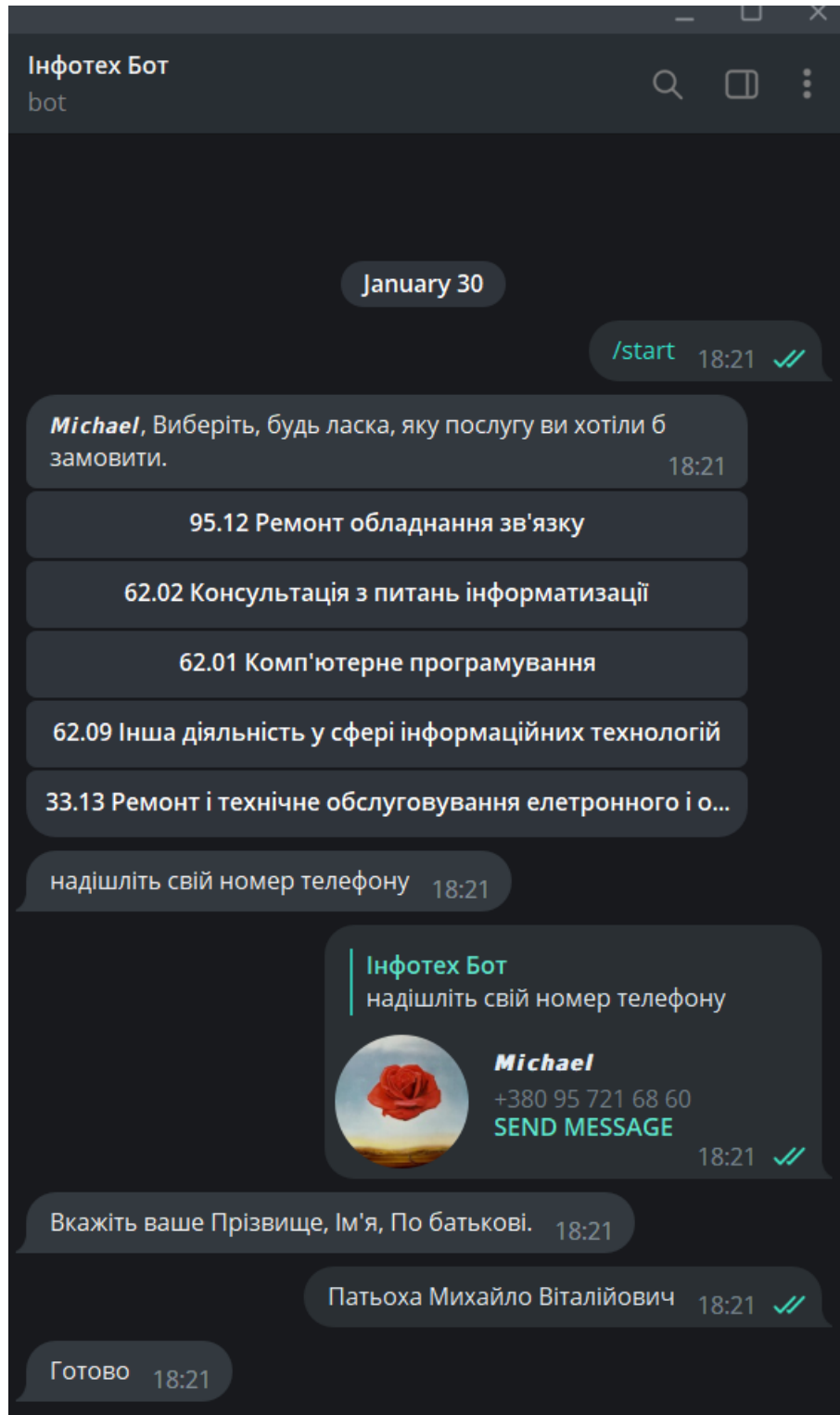
keyboard = types.ReplyKeyboardMarkup(one_time_keyboard=True)
reg_button = types.KeyboardButton(text="☎ Надіслати номер телефону",
request_contact=True)
keyboard.add(reg_button)
bot.send_message(call.message.chat.id, f"надішліть свій номер телефону",
reply_markup=keyboard)
bot.register_next_step_handler(call.message, get_number)


def get_number(message):
    try:
        number=message.contact.phone_number
        print(number)
        setnumber(message.from_user.id, number)
        bot.send_message(message.chat.id, f"Вкажіть ваше Прізвище, Ім'я, По
Батькові.", reply_markup=types.ReplyKeyboardRemove())
        bot.register_next_step_handler(message, save)
    except:
        keyboard = types.ReplyKeyboardMarkup(one_time_keyboard=True)
        reg_button = types.KeyboardButton(text="☎ Надіслати номер телефону",
request_contact=True)
        keyboard.add(reg_button)
        bot.send_message(message.chat.id, f"надішліть свій номер телефону!",
reply_markup=keyboard)
        bot.register_next_step_handler(message, get_number)

def save(message):
    global spreadsheetId, service, data
    print(data)
    bot.send_message(message.chat.id, f"ГОТОВО")
    results = service.spreadsheets().values().append(
        spreadsheetId = spreadsheetId,
        range="Лист!A1",
        valueInputOption='RAW',
        body={'values': [[message.text, find_type_by_id(message.from_user.id),
find_number_by_id(message.from_user.id)]]}).execute()
    bot.polling()




```

## ДОДАТОК Е. ЗНІМКИ ЕКРАНУ









Інфотех бот



ФайлПравкаВидВставкаФорматДанныеИнструменты




100% ▾

р. % .0 .00 123 ▾

По умолча... ▾

10

B6 ▾



	A	B	C
1	ПІБ	Вид діяльності	Номер телефону
2	Патьоха Михайло Віталійович	1	+380957216860
3	Думініке Вікторія Сергіївна	3	380663279406