

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Створення телеграм-боту для просування ростерії «Soloway»

Виконала: студентка IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

(підпис)

Мушинська Г.Є.

(прізвище та ініціали)

Керівник

(підпис)

Струтинська І.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Д.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Луцків А.М.

(прізвище та ініціали)

Тернопіль  
2021

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(прізвище та ініціали)

«\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Мушинській Галині Євгенівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Створення телеграм-боту для просування ростерії «Soloway»

Керівник роботи доктор економічних наук, професор кафедри КН Струтинська Ірина  
Володимирівна.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» березень 2021 року № 4/7-171

2. Термін подання студентом завершеної роботи Дата захисту 2021р.

3. Вихідні дані до роботи Літературні та інтернет джерела щодо створення телеграм боту для просування ростерії «Soloway»

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Теоретичні та прикладні положення створення телеграм-ботів. 1.1 Історія виникнення та актуальність чат-ботів. 1.2 Диверсифіковані платформи для інтегрування чат-бота. 1.3 Дослідження методів створення чат-ботів. 1.3.1 Диверсифіковані платформи для створення чат-ботів. 1.3.2 Мови програмування для створення чат-ботів. 1.4. Аналіз існуючих телеграм-ботів. 1.5. Висновок до першого розділу. 2 Створення телеграм-боту для ростерії «Soloway». 2.1 Вимоги замовника. 2.2 Вибір фірмових кольорів та створення логотипу. 2.3 Розробка гри на HTML, JS та CSS. 2.4 Розробка телеграм-боту для просування ростерії «Soloway». 2.4.1 Створення телеграм-боту. 2.5 Тестування телеграм-боту. 2.6 Висновок до другого розділу. 3 Безпека життєдіяльності, основи охорони праці. 3.1 Маркетингова діяльність на підприємстві. 3.2 Суть та зміст управління охороною праці. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 02.03.2021

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		<i>Виконано</i>
2.	Підбір джерел про створення телеграм-боту для просування ростерії «Soloway»		<i>Виконано</i>
3.	Переклад та опрацювання джерел про створення телеграм-боту для просування ростерії «Soloway»		<i>Виконано</i>
4.	Виконання дослідження щодо створення телеграм-боту для просування ростерії «Soloway»		<i>Виконано</i>
5.	Розроблення телеграм-боту для просування ростерії «Soloway»		<i>Виконано</i>
6.	Оформлення розділу «Теоретичні та прикладні положення створення телеграм-боту»		<i>Виконано</i>
7.	Оформлення розділу «Створення телеграм-боту для просування ростерії «Soloway»»		<i>Виконано</i>
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		<i>Виконано</i>
9.	Виконання завдання до підрозділу «Основи охорони праці»		<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи		<i>Виконано</i>
11.	Нормоконтроль		<i>Виконано</i>
12.	Перевірка на плагіат		<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи		<i>Виконано</i>
14.	Захист кваліфікаційної роботи		

Студент

(підпис)

Мушинська Г.Є.

(прізвище та ініціали)

Керівник роботи

(підпис)

Струтинська І.В.

(прізвище та ініціали)

## АНОТАЦІЯ

Створення телеграм-боту для просування ростерії «Soloway»// Кваліфікаційна робота освітнього рівня «Бакалавр» // Мушинська Галина Євгенівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2021 // С. – 75, рис. – 27, табл. – 1, кресл. – 0 , додат. – 5 , бібліогр. – 37.

**Ключові слова:** чат-бот, телеграм, телеграм-бот, python, інтернет-маркетинг, база даних.

Кваліфікаційна робота присвячена створенню телеграм-боту для просування ростерії «Soloway».

Метою даної кваліфікаційної роботи є дослідження та пошук рішень, щоб створити телеграм-бота для просування ростерії «Soloway».

У першому розділі кваліфікаційної роботи розглянуто теоретичні та прикладні положення створення чат-ботів, досліджено історичний аспект їх розвитку. Розглянуто диверсифіковані платформи для інтеграції та проведено порівняльну характеристику існуючих чат-ботів.

У другому розділі кваліфікаційної роботи розглянуто вимоги замовника, представлено основні технології розробки телеграм-боту, створено телеграм-бота для просування ростерії «Soloway» та протестовано його роботу.

У третьому розділі кваліфікаційної роботи проведено огляд маркетингової діяльності на підприємстві та досліджено суть та зміст управління охороною праці.

## ANNOTATION

Telegram bot development for coffee shop «Soloway» promotion // Qualification work degree «Bachelor» // Mushynska Halyna // Ternopil Ivan Pul'uj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2021 // P. – 75, Fig. – 27, Tables – 1, Annexes. – 5, References – 37.

Qualification work is Telegram bot development for coffee shop «Soloway» promotion.

In the first section of the qualification work the theoretical and applied provisions of creation of chat bots are considered, the historical aspect of their development is investigated. Diversified platforms for integration are considered and a comparative description of existing chat bots is made.

In the second section of the qualification work the requirements of the customer are considered, the basic technologies of telegram bot development are presented, the telegram bot for promotion of the «Soloway» roster is created and its work is tested.

The third section of the qualification work reviews the marketing activities at the enterprise and explores the essence and content of occupational safety management.

**Key words:** chat-bot, telegram, telegram-bot, python, marketing, database.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ЧБ – Чат-бот.

ШІ – Штучний інтелект.

A.L.I.C.E. – Штучна лінгвістична комп'ютерна сутність в Інтернеті.

ПЗ – Програмне забезпечення.

Соцмережа – Соціальна мережа.

JS – JavaScript (мова програмування).

ММ – Метамова (мова для опису мови).

API – Application programming interface.

Ps – Adobe Photoshop.

HTML – HyperText Markup Language.

CSS – Cascading Style Sheets.

API – Application programming interface.

ІК – Inline keyboards.

БД – База даних.

ОС – Операційна система.

## ЗМІСТ

ВСТУП .....	7
1 ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ ПОЛОЖЕННЯ СТВОРЕННЯ ТЕЛЕГРАМ-БОТІВ .....	9
1.1 Історія виникнення та актуальність чат-ботів .....	9
1.2 Диверсифіковані платформи для інтегрування чат-бота.....	13
1.3 Дослідження методів створення чат-ботів.....	18
1.3.1 Диверсифіковані платформи для створення чат-ботів.....	18
1.3.2 Мови програмування для створення чат-ботів .....	19
1.4 Аналіз існуючих телеграм-ботів .....	23
1.5 Висновок до першого розділу .....	24
2 СТВОРЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РОСТЕРІЇ «SOLOWAY» .....	28
2.1 Вимоги замовника .....	28
2.2 Вибір фірмових кольорів та створення логотипу .....	28
2.3 Розробка гри на HTML, JS та CSS .....	31
2.4 Розробка телеграм-бота для просування ростерії «Soloway» .....	33
2.4.1 Створення телеграм-боту .....	34
2.5 Тестування телеграм-боту .....	39
2.6 Висновок до другого розділу .....	43
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	44
3.1 Маркетингова діяльність на підприємстві.....	44
3.2 Суть та зміст управління охороною праці. ....	48
ВИСНОВКИ.....	49
ПЕРЕЛІК ДЖЕРЕЛ .....	50
ДОДАТКИ	

## ВСТУП

**Актуальність теми.** У останні роки компанії з різних галузей все активніше впроваджують чат-боти в месенджери і соціальні мережі. Це зумовлено тим, що в соцмережах сконцентрована дуже велика і активна аудиторія, тобто потенційні споживачі. Інтернет в сучасних умовах – це глобальне середовище для спілкування, розваг та інструментів розвитку бізнесу. Актуальність теми «Створення телеграм-боту для просування ростерії «Soloway»» виникла через те, що з розвитком інформаційних технологій месенджери стають все більш популярні [1], і чат-боти стають важливою частиною цього програмного середовища.

Що стосується корпоративної інфраструктури, чат-боти дозволяють спростити такі завдання, як замовлення, забезпечують електронні черги, надають цілодобову підтримку, можливість оплати картками тощо. Це можливість оптимізувати витрати та зробити робочий персонал менш завантаженим, що призведе до покращення виконання завдань, зменшення психологічної напруженості та збільшить взаємодію між фірмою та клієнтом, а це перш за все буде означати зростання якості послуг. Оскільки технології не стоять на місці, з часом боти стануть розуміти людські запити краще, що зробить їх незамінними помічниками людського ресурсу.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є створення телеграм-боту для просування ростерії «Soloway».

Досягнення окресленої мети викликало необхідність виконання таких **завдань:**

- Проаналізувати літературні та інтернет-джерела з метою виявлення технологій, способів і засобів для створення телеграм-боту.
- Порівняти аналоги телеграм-ботів.



– Сформулювати вимоги, спроектувати і розробити телеграм-бот для просування ростерії «Soloway».

– Провести тестування створеного телеграм-боту для просування ростерії «Soloway».

*Об'єктом дослідження* є платформа Телеграм для розробки сервісу, що надає клієнтам можливість робити замовлення, проводити оплату та бронювання в одному середовищі.

*Предметом дослідження* є теоретико-практичні положення створення телеграм-боту для просування ростерії «Soloway».

### **Практичне значення одержаних результатів.**

Полягає у розробці сервісу для платформи Телеграм, що надає клієнтам можливість робити замовлення, проводити оплату та бронювання в одному середовищі. Оптимізує роботу людського ресурсу та зменшить навантаження, приваблюючи клієнтів за простоту використання.

# 1 ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ ПОЛОЖЕННЯ СТВОРЕННЯ ТЕЛЕГРАМ-БОТІВ

## 1.1 Історія виникнення та актуальність чат-ботів

Важливу роль відіграє обслуговування клієнтів у максимізації прибутковості організації. Компанії витрачають достатньо грошей, щоб гарантувати якість послуг на високому рівні. Персонал виділяє багато часу на запитання по телефону або через додатки для обміну повідомленнями, щоб переконатися, що люди задоволені наданими послугами. Подібне обслуговування клієнтів має багато недоліків. По-перше, компанія зазвичай отримує однакові запитання від різних покупців. По-друге, важко організувати цілодобову підтримку, особливо для невеликих організацій та підприємств [2].

Однак нові технологічні розробки відкрили двері для більш швидкого та простого рішення цих проблем: чат-ботів. Він може спілкуватися з клієнтом, але не обмежується технологіями, оскільки технологія ШІ вбудована в програмне забезпечення.

Чат-боти побудовані за допомогою програм, які автоматично взаємодіють з отриманими повідомленнями. Їх можна запрограмувати на використання машинного навчання для реагування на різні повідомлення, особливо для адаптації своїх реакцій до ситуації [3].

Ці програми прості у використанні і багато клієнтів нададуть перевагу їм, ніж дзвінку до представника, оскільки другий спосіб, як правило, довший. Вони також можуть заощадити гроші для компаній і їх легко створити. Оскільки більшість чат-ботів використовують програми обміну повідомленнями, які вже є на більшості телефонах у всьому світі, що означає, що клієнти вже підключені до мережі і готові до роботи з ботом. Чат-боти –

це майбутнє взаємодії з клієнтами, і вони найближчим часом можуть замінити пошукові системи та багато додатків.

Боти вміють налагоджувати взаємодію з клієнтами. Вони можуть надавати рекомендації, приймати замовлення, пропонувати спеціальні пропозиції та відповідати на будь-які запитання чи занепокоєння клієнта через розмовний інтерфейс. Процес швидкий, простий і потенційно знижує ризик конфронтації. Оскільки ми живемо у столітті, де більшість процесів відбувається за допомогою смартфонів та комп'ютерів, даний спосіб просування є дуже важливою складовою у бізнесі.

Загалом, чат-боти з'явилися більше п'ятдесяти років тому і розвиваються до нашого часу.

Все почалось з того, що у 1950 році Алан Тьюрінг написав наукову роботу під назвою «Комп'ютери та інтелект», у якій говорилося, що комп'ютерна програма може мислити і говорити як людина. Щоб довести це, Тьюрінг запропонував експеримент під назвою "Імітаційна гра", суть якого полягала у тому, що людина, призначена суддею, розмовляла по комп'ютеру з людиною і машиною, яких не можна було побачити. Завдання судді полягало у тому, щоб відрізнити комп'ютер від реальної людини. Тьюрінг припустив, що якщо суддя не зможе визначити, якими були відповіді комп'ютера, це доведе, що комп'ютер здатний імітувати людську мову.

Предком усіх чат-ботів прийнято вважати Елізу (ELIZA) – віртуального співрозмовника, якого створив Джозеф Вейценбаум в 1966 році. Програма пародіювала діалог з психотерапевтом, використовуючи техніку «активного слухання». На рисунку 1.1 показано, що Еліза отримувала репліки від користувача, перефразувала їх і створювала ілюзію діалогу. Тобто програма надала можливість спілкування між комп'ютером та людиною.

```

Welcome to
          EEEEE LL   IIII  ZZZZZ  AAAAA
          EE    LL   II    ZZ    AA  AA
          EEEEE LL   II    ZZZ   AAAAAA
          EE    LL   II    ZZ    AA  AA
          EEEEE LLLLL IIII ZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:

```

Рисунок 1.1 – Інтерфейс програми Еліза [4]

На початку 1970-х психіатр Кеннет Колбі зробив кроки вперед принципами ELIZA. Запровадивши PARRY, Колбі скоріше застосував розмовну стратегію чат-ботів, використовуючи модель людини з параноїчною шизофренією, щоб допомогти підвищити правдоподібність відповідей.

У 1984 році Вільям Чемберлен та Томас Еттер написали Racter, який ще називали "штучно божевільний" доповідач, та ця програма ніколи не випускалась публічно.

Наступною розробкою був Jabberwacky, створена британським програмістом Ролло Карпентером. Це була одна з найперших спроб створити ШІ за допомогою людської взаємодії. Метою розробки була імітація чатів людини в цікавій, розважальній та жартівливій формі.

У 1991 був розроблений комп'ютеризований психолог-чат-бот із цифровим голосом Доктор Сбайцо [5]. Це був синтез мовлення штучного інтелекту, створений Creative Labs, щоб показати вражаючий на той час діапазон оцифрованих голосів звукової карти.

A.L.I.C.E., яку також називають Alicebot, або просто Аліса, – це чат-бот з обробки природних мов, вперше розроблений у 1995 році, який тричі вигравав Л'юбнера. Алісу надихнула програма ELIZA. У 2000 році був створений чат-бот, який використовує сарказм і дотепність, а також іронію та власний штучний інтелект для розваги людей. Elbot був створений Фредом Робертсом та Artificial Solutions.

Чат-бот Smarterchild розроблений ActiveBuddy Inc у 2001 році Робертом Хоффером, Тімоті Кей та Пітером Левітаном [6]. Він був доступний у мережах обміну повідомленнями MSN Instant Messenger. Чат-бот пропонував веселу персоналізовану розмову і вважався попередником Siri від Apple і S Voice від Samsung.

У 2005 та 2006 були створені Mitsuku та IBM Watson відповідно. Mitsuku стверджує, що є підлітком жінкою-чат-ботом з Лідса, Англія. Її інтелект включає здатність міркувати про певні речі, вона може грати в ігри та робити магію. IBM Watson спочатку розроблений для участі в американській телевізійній програмі «Jeopardy!», в 2011 році він переміг двох колишніх чемпіонів.

Siri вперше привернула увагу громадськості в лютому 2010 року, коли була запущена як новий додаток для iPhone. Згодом Apple придбала компанію та інтегрувала голосового асистента в iPhone 4S під час її випуску в жовтні 2011 року [7], завдяки чому голосові додатки назавжди з'явилися на основному споживчому ринку. У 2012 компанія Google розробила Google Now спеціально для мобільного додатка Google Search. Він використовує природний інтерфейс користувача, щоб відповідати на запитання, робити рекомендації та виконувати дії, передаючи запити набору веб-служб.

До 2015 року Siri залишалася чи не найвідомішим з мобільних голосових помічників, поки компанія Amazon не запустила Alexa. Вона передавала команди телефону, а, отже, привернула увагу споживачів та відкрила величезний нині ринок розумних домашніх колонок.

У тому ж році Microsoft створила Cortana – це інтелектуального особистого помічника. Cortana розпізнає природні голосові команди, може встановлювати нагадування та відповідати на запитання за допомогою пошукової системи Bing. А у 2016 компанія створила Tay – чат-бот, створений для імітації мови та звичок американської дівчинки-підлітка. Чат-бот викликав суперечки і був закритий через 16 годин після запуску [8], коли почав публікувати образливі твіти.

У 2017 році Woebot Labs створили Woebot – ЧБ, що підтримує ПІ і використовує інтелектуальне відстеження, щоб допомогти їм дізнатися про емоції [9].

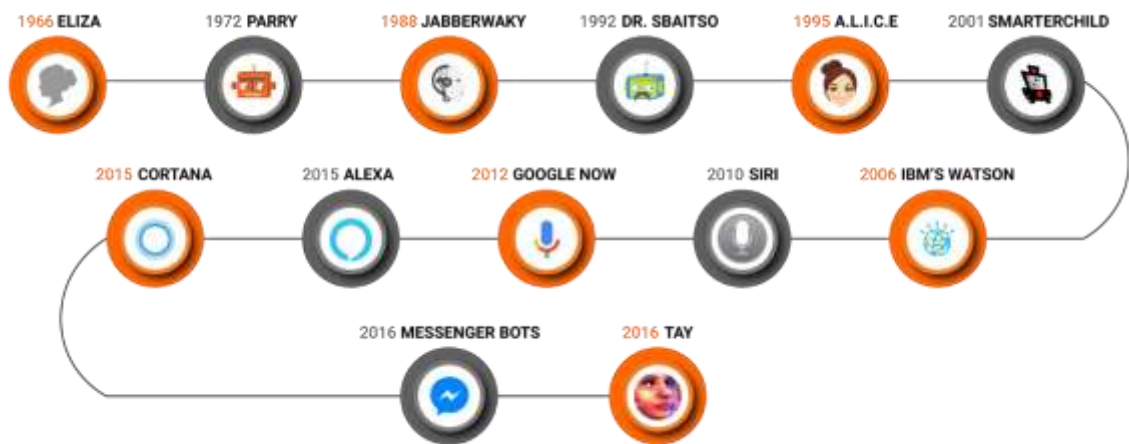


Рисунок 1.2 – Хронологія створення віртуальних співрозмовників

На рисунку 1.2 можна спостерігати етапи розвитку ЧБ. З часом, ця технологія стане ще популярнішою та матиме більше інструментів, оскільки боти продовжують досліджуватись і створюватись.

## 1.2 Диверсифіковані платформи для інтегрування чат-бота

Месенджер – це програмне забезпечення, що надає можливість обмінюватись повідомленнями [10]. Це дозволяє обмінюватися

повідомленнями, що є різновидом онлайн-чату, який пропонує передачу тексту в реальному часі через Інтернет.

Існує багато видів месенджерів: Facebook Messenger, Telegram, WhatsApp, Viber та інші. Ці додатки зручні і вони стали звичним засобом для комунікацій. Кожного дня вони все більше проникають у наше щоденне життя, коли ми спілкуємося з друзями, родичами і колегами в чатах. Згідно з [11] двадцять п'ять відсотків користувачів, які хочуть зробити замовлення на певну продукцію чи послугу, роблять це за допомогою мобільного телефону та доступу до Інтернету. Більше половини споживачів хотіли б отримувати сповіщення про послуги текстовим способом. Користувачі смартфонів по всій країні дзвонять і отримують у п'ять разів менше дзвінків, ніж надсилають та отримують повідомлення, а статистично кажучи, 30% користувачів смартфонів повністю відмовляються здійснювати та приймати дзвінки, а спілкуються лише через повідомлення [12].

Згідно з даними сайту [13], станом на січень 2021 року у світі лідирує месенджер WhatsApp, який налічує два мільярди користувачів. На рисунку 1.3 зображено, що за популярністю йде Facebook Messenger, Weixin, QQ, Telegram.

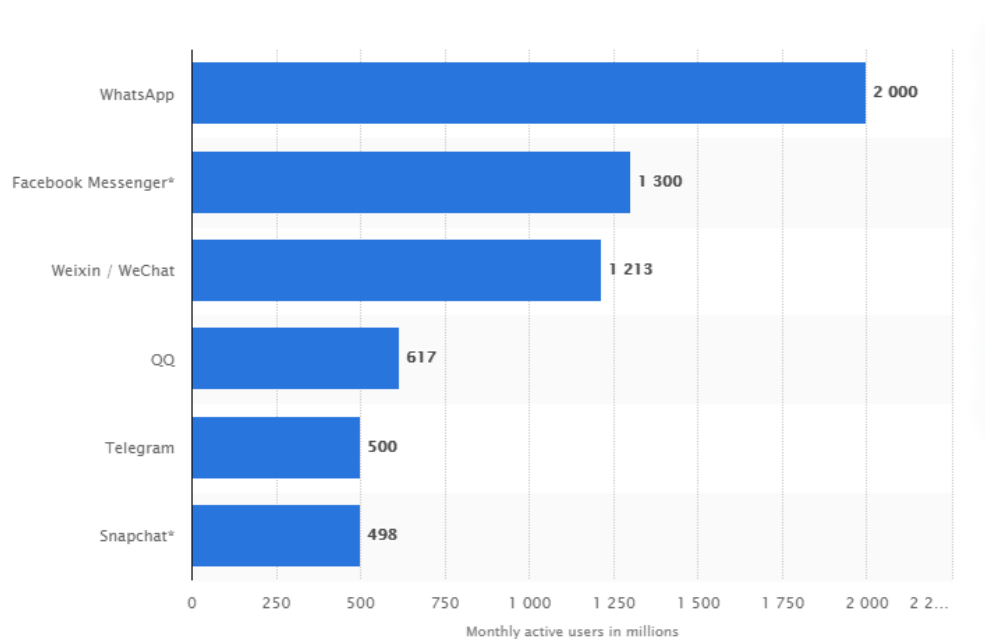


Рисунок 1.3 – Популярність месенджерів у світі

Згідно дослідженню, що провели AIN.UA [14], у якому взяли участь 1229 людей, в Україні лідирує Telegram. Близько п'ятдесяти відсотків респондентів вибирають його в якості основного месенджера та лише тридцять відсотків використовують його як робоче середовище. На другому місці опинився Viber, близько дев'ятнадцяти відсотків людей використовують його для спілкування, а Facebook Messenger зайняв лише третю позицію, де в якості основного месенджера його використовують вісім відсотків опитаних, що зображено на рисунку 1.4.

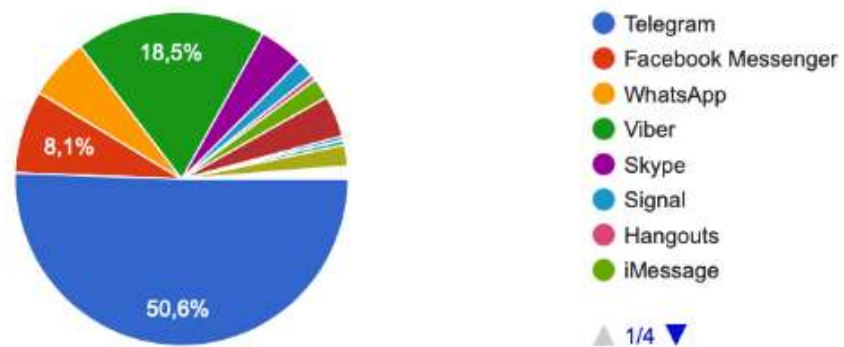


Рисунок 1.4 – Вибір месенджера в Україні

Бот-платформа – це програма на API з якою можна створити віртуальний діалог. Чат-боти підтримують найпопулярніші месенджери, такі як: Viber, Facebook Messenger, Telegram, Whatsapp, Wechat.

Функціонал чат-боту на платформі не поступається іншим. Розмова відбувається через повідомлення, що надіслане з загальнодоступного акаунту. Також месенджер містить стандартний набір функцій. Він може робити розсилки, тобто відправляти повідомлення всім контактам, навіть тим, що не підписані на канал та оформлювати пости у вигляді «каруселі» з товарами [15]. Чат-боти Viber доступні у всьому світі, і багато брендів



почали використовувати їх для спілкування зі своїми клієнтами. На рисунку 1.5 зображено чат-бота компанії YASNO, який надає перелік дій, що можна виконати за допомогою чат-боту.

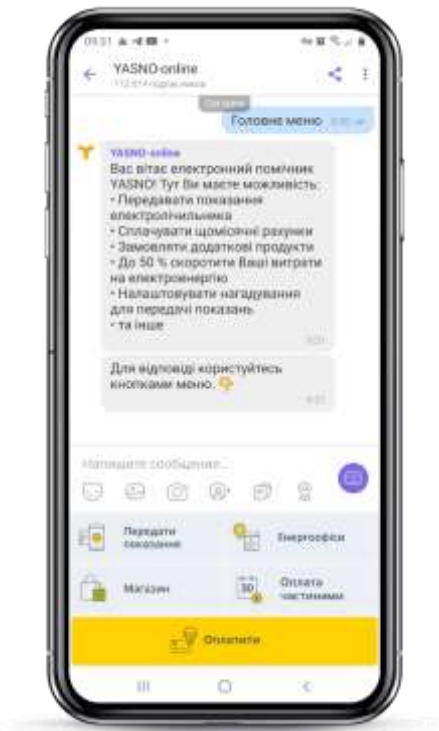


Рисунок 1.5 – Чат-бот в Viber

Viber не має обмежень API. Розділ з відкритим кодом від спільноти Viber API пропонує деяку досить витончену функціональність, якої інші месенджери не містять. Наприклад, ви можете повністю розробити власну кнопку, шпалери тощо.

Facebook Messenger має 1,3 мільярда користувачів і більше 100 000 активних ботів. Кожен день ці боти збирають дані, рекомендують продукти, приймають замовлення, надають підтримку користувачам [15].

Крім текстових повідомлень, боти мають можливість спілкуватись за допомогою зображень, списків, аудіо файлів.

Користувачі можуть відповідати за допомогою тексту, смайликів, GIF-файлів, зображень, аудіо, відео, а також підтягувати чат-ботів в групові чати.

Незважаючи на те, що Facebook Messenger має широкий спектр інструментів і функцій, в ньому немає команд у вигляді кнопок, що дивно і незручно. На рисунку 1.6 зображено чат-бота Facebook Messenger компанії «Галицький Пекар».

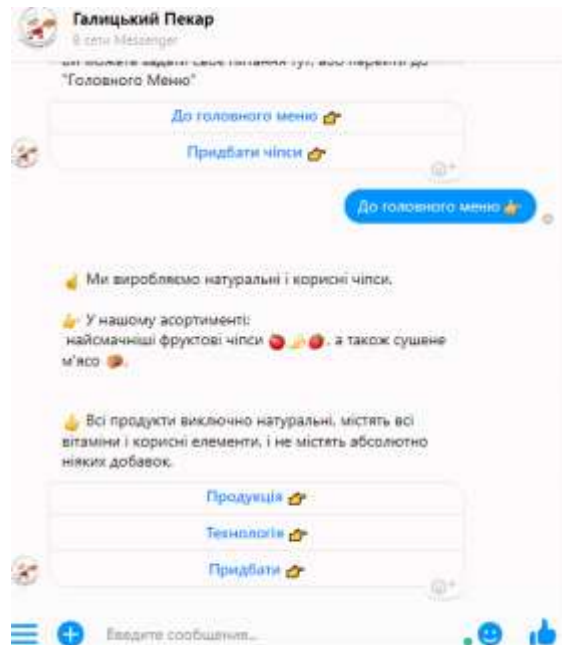


Рисунок 1.6 – Чат-бот у Facebook Messenger

Якщо говорити про чат-бота в Telegram, то основною перевагою цього програмного забезпечення над Facebook Messenger є його API, який відкритий для всіх, пропонує найбільше можливостей і дозволяє реалізувати безліч творчих ідей.

Telegram – це чудова захищена платформа для створення креативного та вишуканого ЧБ, наприклад, він буде мати можливість для інтеграції з гаманцями для різних валют, зможе проводити фінансові операції та обробляти банківські запити. [15]. Команда Telegram також підкреслює просту інтеграцію вбудованих запитів, ігрових ботів HTML5 і призначених для користувача клавіатур. На рисунку 1.7 буде зображено Telegram чат-бот RailwayBot, що допомагає шукати квитки на потяг.

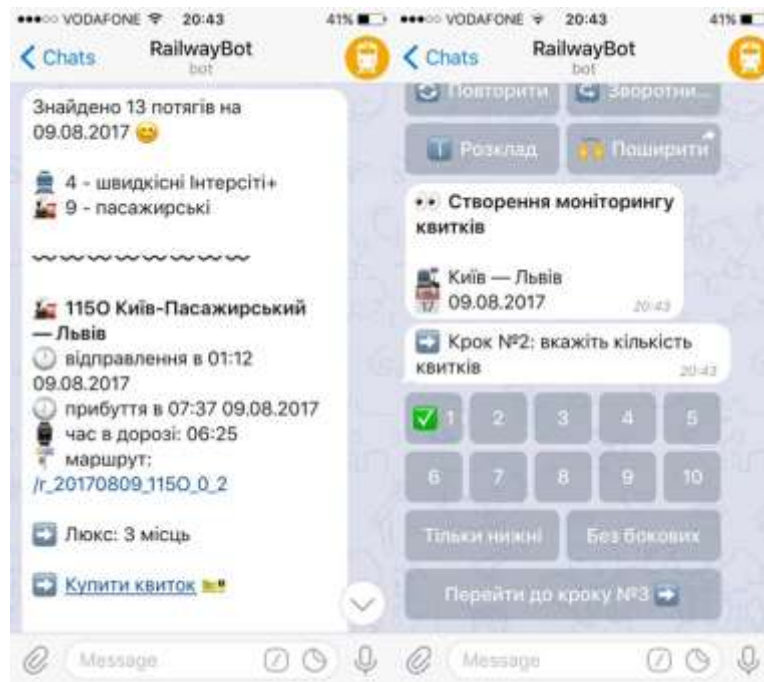


Рисунок 1.7 – Чат-бот у Telegram

Оскільки ролетерія «Soloway» спрямована більш на молодшу аудиторію, доречно використовувати Telegram як платформу для інтегрування чат-боту.

### 1.3 Дослідження методів створення чат-ботів

#### 1.3.1 Диверсифіковані платформи для створення чат-ботів

Сьогодні нам пропонується великий вибір платформ для створення ЧБ, які можуть вести діалог з користувачем за написаним сценарієм або ж з використанням ШІ. Ці платформи також дозволяють інтеграцію з багатьма месенджерами, розглянутими в попередньому пункті, а також з власними веб-сайтами та додатками.

Відбираючи сервіси для огляду, було акцентовано увагу на наступних:

1. Aimylogic [16] – конструктор чат-ботів на основі машинного навчання та ШІ. Також в сервісі можна створювати сценарії для розумних дзвінків, голосових ботів і навички для голосових асистентів.

2. Flow XO [17] – конструктор чат-ботів для різних бізнес-завдань: для вітання відвідувачів сайту, збору і кваліфікації лідів, відповідей на найчастіші запитання та т. д. Працює в браузері і інтегрується з сотнею інших сервісів і додатків, включаючи Trello, Gmail, CRM, Basecamp 2.

3. BroBot [18] є інструментом автоматизації роботи в соцмережах. Крім чат-ботів тут можна створювати розклад для постів, запрошення стати друзями.

4. Чат-боти в BotKits [19] створюються на основі дерева сценаріїв, яке включає основні запити і варіанти розвитку діалогу. Працює в браузері. Дозволяє відстежувати статистику взаємодії з ботом. Також в сервісі можна створювати тригерні розсилки по базі користувачів, які взаємодіяли з чат-ботом, і працювати з цією базою у вбудованій міні-CRM.

5. Botmother [20] – конструктор чат-ботів для соцмереж і месенджерів. Можна перенести базу користувачів з інших конструкторів, а також експортувати її з Botmother. Є можливість відстежувати статистику і налаштовувати тригерні розсилки по базі користувачів, які взаємодіяли з ботом. У додатку Б наведено порівняльну таблицю популярних платформ для створення чат-ботів, які не потребують навиків програмування.

### **1.3.2 Мови програмування для створення чат-ботів**

Мова програмування [21] – це набір правил, які визначають, як виглядає написана комп'ютерна програма і що комп'ютер може робити під її контролем. Програма – це код, написаний відповідно до правил певної мови програмування. Код, який стає програмою, що називається «вихідним кодом».

Існує ще велика кількість різних мов програмування, і кожна з них застосовується в певній сфері. Необхідно вибрати кілька найпопулярніших

мов, щоб між ними проводити аналіз, і дізнатись яку з мов програмування обрати для написання чат-боту.

Проаналізувавши рейтинг PYPL та TIOBE від Github за 2021 рік, можна скласти список десяти найбільш затребуваних мов програмування. PYPL та TIOBE мають однаковий принцип роботи, і полягає у тому, що чим більш раз мова програмування згадується, тим популярнішою вона вважається. Відрізняються ці рейтинги лише за підходами. TIOBE вимірюється кількістю звернень пошукових системи, серед них Google, Yahoo !, YouTube, Bing, Baidu, Wikipedia, Amazon. У той час, як PYPL враховує як часто навчальні матеріали по мовах вивчаються Google Trends.

Згідно з TIOBE, що зображено на рисунку 1.8, попереду Python тільки C, і творці TIOBE вважають, що ці мови програмування можуть помінятися місцями вже в другому півріччі, тому що C, як і Java, втрачає популярність. У минулому травні Java вперше за попередні п'ять років поступилась першим місцем у списку рейтингу.

May 2021	May 2020	Change	Programming Language	Ratings	Change
1	1		C	13.38%	-3.68%
2	3	▲	Python	11.87%	+2.75%
3	2	▼	Java	11.74%	-4.54%
4	4		C++	7.81%	+1.69%
5	5		C#	4.41%	+0.12%
6	6		Visual Basic	4.02%	-0.16%
7	7		JavaScript	2.45%	-0.23%
8	14	▲	Assembly language	2.43%	+1.31%
9	8	▼	PHP	1.86%	-0.63%
10	9	▼	SQL	1.71%	-0.38%

Рисунок 1.8 – Рейтинг мов програмування згідно з TIOBE [22]

Слід пам'ятати, що індекс TIOBE – це не найкраща мова програмування або мова, на якій написано найбільше коду.

Рейтинг PYPL віддає перше місце Python, а друге та третє місце посідає Java та Javascript. C# посідає 4-те місце, TypeScript – 10-те, показано на рисунку 1.9.

Rank	Change	Language	Share	Trend
1		Python	29.9 %	-1.2 %
2		Java	17.72 %	-0.0 %
3		JavaScript	8.31 %	+0.4 %
4		C#	6.9 %	-0.1 %
5	↑	C/C++	6.62 %	+0.9 %
6	↓	PHP	6.15 %	+0.1 %
7		R	3.93 %	+0.0 %
8		Objective-C	2.52 %	+0.1 %
9		Swift	1.96 %	-0.2 %
10	↑	TypeScript	1.89 %	+0.0 %

Рисунок 1.9 – Рейтинг мов програмування згідно з PYPL [23]

Індекс PYPL базується на необроблених даних, що надходять від Google Trends, тому будемо використовувати даний рейтинг як основу. Далі розглянемо три найпопулярніші мови програмування, такі, як Python, Java та Javascript, щоб визначити яка з них буде найкращим рішенням для написання телеграм-боту.

JavaScript [24] посідає високе місце в рейтингах протягом декількох років, і у 2021 році він знову потрапив до провідних мов програмування, оскільки, по-суті, є основою інтерфейсної розробки.

JavaScript зазвичай використовується, щоб збагатити веб-сторінки та зробити їх інтерактивними. За допомогою JS можна додавати спливаючі вікна, ефекти та навіть невеликі ігри до веб-додатків.

Найголовніше в JavaScript:

1. Доступ до декількох фреймворків;
2. Функціональність перевірки даних;
3. Сумісний з декількома мовами програмування;

4. Підтримує бібліотеку для створення телеграм-ботів.

Java посіла друге місце за популярністю і нею користуються велика кількість програмістів протягом багатьох років поспіль.

Найбільші світові бренди використовують Java [25] для створення веб-додатків та серверних веб-систем, наприклад, такі як: eBay, LinkedIn, Twitter, Amazon.

Java також є офіційною мовою для створення програм для Android. Він має найбільшу підтримку від Google, і більшість програм у магазині Google Play написані на Java.

Найголовніше в Java:

1. Портативна в операційних системах;
2. Стабільне середовище;
3. Якісна компіляція коду;
4. Висока стійкість;
5. Підтримує бібліотеку для створення телеграм-ботів.

Далі розглянемо першу позицію у списку PYPL та одну з найкращих серверних мов на 2021 рік – Python.

Вона пройшла великий розвиток, оскільки перетворилась з мови для написання сценаріїв автоматизації або швидкого прототипування додатків на першокласну мову для веб- і мобільних розробок.

На сьогодні, Python [26] – це мова з послідовним синтаксисом, єдиною стандартною бібліотекою, найкращою документацією, легкими фреймворками, включаючи надзвичайно популярний Django.

Найважливіше в Python:

1. Мова з відкритим кодом, об'єктно-орієнтована;
2. Крос-платформні рішення;
3. Дизайн асинхронного кодування;
4. Використовується для ШІ та ММ, настільних та веб-програм;
5. Доступ до декількох модулів;

б. Підтримує бібліотеку для створення телеграм-ботів.

У підсумок можна сказати, у кожної мови програмування є свої плюси і мінуси, кожна мова хороший варіант для конкретних цілей, наприклад якщо створювати сайт, то за описаним критерієм потрібно вибрати JavaScript, а для написання ЧБ в месенджері Telegram буде найкращим рішенням Python, так як він більш простий в написанні, має багато документації і великий вибір бібліотек, які допоможуть створити хороший сервіс.

Для реалізації ЧБ був обраний інтерактивний Python IDE Pycharm з повним набором інструментів для ефективною розробки в Python. Це вправний та функціональний редактор для коду, що володіє підсвічуванням синтаксису, надає можливість автоматичного форматування та автоматичного відступу для тих мов, що підтримуються.

#### **1.4 Аналіз існуючих телеграм-ботів**

Безкоштовні та доступні інтерфейси програмування програм (API) полегшують життя, якщо потрібно збирати дані або автоматизувати процеси. Telegram має можливість показувати статус партнера по спілкуванню в Інтернеті та передавати тексти, фотографії, відео, аудіофайли, інформацію про місцезнаходження, контакти та документи. Telegram Bot – це програма, яка поводить як звичайний користувач у чаті з додатковими функціями. Він виконує заздалегідь визначені завдання самостійно та без участі людини.

Ще однією особливістю Telegram є його безпека. Усі заходи, включаючи чати, групи та засоби масової інформації, якими діляться учасники, шифруються. Це означає, що їх не видно без попереднього розшифрування. Telegram також володіє шифруванням між користувачами в чаті за допомогою функції "секретні чати".

Боти стають дедалі популярніші. Однак бот може бути не тільки розвагою, але і важливим інструментом для компаній, інтернет-магазинів,



великих ЗМІ і власників сторінок в соцмережах. Далі розглянемо декілька телеграм-ботів, які є корисними для української аудиторії, зображених на таблиці 1.1.

Таблиця 1.1 – Існуючі рішення телеграм-ботів

Назва боту	Вид	Можливості
1	2	3
City Guide	Пошуковий	Допомагає знайти будь-яку організацію за ключовими словами та тегами.
The Unexplored City	Розважальний	Прокладає пішохідні маршрути для відвідування згідно геолокації.
OpenDataBot	Консультант	Допомагає знайти всю відкриту інформацію про компанію, ФОП, транспортні засоби за декількома критеріями.
WolframBot	Помічник	Допоможе знайти розв’язок завдання та зробити розрахунки

Як можемо спостерігати, телеграм-боти мають не лише розважальний характер. Бот Telegram є корисним доповненням до інтелектуального спілкування, особливо у професійних сферах. У порівнянні з найпопулярнішими службами обміну повідомленнями, Telegram пропонує додаткові переваги в обміні інформацією за допомогою спеціальних запитів та дій бота. Однак, як і всі програми, Telegram зазвичай вимагає конкретних знань з програмування для широкого використання.

## 1.5 Висновок до першого розділу

В першому розділі кваліфікаційної роботи було розглянуто історію виникнення та актуальність чат-ботів, проаналізовано платформи для інтеграції і вибрано платформу – Telegram, через поєднання швидкості, безпеки та популярності.

Використання чат-ботів можливе у будь-якій сфері і у наш час широко використовується завдяки розвитку технологій та спрощення їх створення за допомогою спеціальних платформ. Також у першому розділі було досліджено мови програмування для створення телеграм-боту та обрано найкраще рішення для даного проекту – Python.

Завдяки аналізу існуючих ботів зроблено висновки, що телеграм-бот є хорошим інструментом у розвитку бізнесу, оскільки він економічно вигідний для компанії, зменшує кількість черг та навантаження на працівників та робить можливість комунікації між організацією та клієнтом цілодобово.

## **2 СТВОРЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РОСТЕРІЇ «SOLOWAY»**

### **2.1 Вимоги замовника**

Як тільки чат-боти з'явилися у мережі, люди були зацікавлені та одночасно стурбовані впливом цієї технології та тим, як складеться майбутнє їхнього бізнесу.

При створенні нового проекту компаніям завжди потрібна реклама, яка не завжди окупається. При цьому ж варто розуміти, що реклама вартує чималих коштів, якщо потрібно досягнути правильного результату.

Для просування кав'ярні, кращим рішенням буде створення телеграм-боту для просування, оскільки він має автоматизовану підтримку клієнтів для різних запитів, економить людські ресурси, має певні шаблони для відповідей на найчастіші питання, створює власний бренд, зручний та простий для використання.

Телеграм-бот для просування ростерії «Soloway» має володіти наступним функціоналом:

1. Володіти фірмовими кольорами та власним логотипом;
2. Бронювання за допомогою боту;
3. Мати можливість перегляду меню;
4. Мати систему лояльності для клієнтів у вигляді гри;
5. Можливість зробити замовлення;
6. Проведення оплати за допомогою бота.

### **2.2 Вибір фірмових кольорів та створення логотипу**

Колір підсвідомо впливає на людину, може викликати необхідні емоції, такі як: впевненість, прихильність, спокій або, протилежні, такі, як тривогу та бажання до дії. Колір компанії може бути як про місію компанії, так і про

слоган компанії, іншими словами: що цінує компанія в першу чергу, на якій основі вона побудована. Нарешті, кожен колір відображає певні асоціації, які в нашій свідомості пов'язані з людськими рисами. Крім того, першою інформацією про все, що зберігається в людській пам'яті, є колір. Таким чином, кольори компанії – це яскрава ідентичність компанії. Таким чином, при розробці фірмового стилю це рішення виконує три основні функції: воно створює бажані емоції у замовника, визначає цінності компанії та допомагає зробити компанію унікальною. Ось чому важливо добре розуміти компанію, вибираючи фірмові кольори.

Для телеграм-боту для просування ростерії «Soloway» було вибрано наступні кольори: #3e391b, #10bc10, #98c510, #d8f014, що зображено на рисунку 2.1.

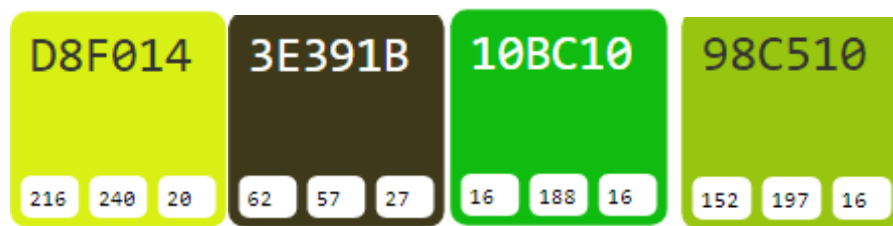


Рисунок 2.1 – Фірмові кольори

Логотип – це тип дизайну, який представляє компанію, бренд чи особу.

Сам термін «логотип» не обов'язково відображати назву, хоча маркетингова мудрість (і здоровий глузд) передбачає, що він, мабуть, має певним чином відповідати бренду, про який йде мова, і має бути досить універсальним, щоб мати змогу його використовувати у різних кампаніях.

Його побудова та вбудовані елементи можуть відрізнятись залежно від призначення.

При розробці логотипу для телеграм-боту я використала Adobe Photoshop [27], програму для редагування зображень та ретушування фотографій.

Pс пропонує користувачам можливість створювати, вдосконалювати чи іншим чином редагувати зображення та ілюстрації.

З її допомогою можна змінювати фон, імітувати зображення з реального життя або створити альтернативне уявлення про Всесвіт. Це найбільш широко використовуваний програмний інструмент для редагування фотографій, маніпуляцій із зображеннями та ретуші зображень та відео різних форматів.

За допомогою Photoshop можна редагувати одиничні зображення, так і великі партії фотографій. На рисунку 2.2 зображено логотип чат-боту «Soloway bot».



Рисунок 2.2 – Логотип чат-боту «Soloway bot»

При розробці логотипу для телеграм-боту зображення пташки було виконане за допомогою інструментів «Фігура» та «Ласо», а також використано шрифт «Sequoie UI» для надпису «Soloway». Всі фігури були залиті градієнтною заливкою фірмових кольорів ростерії.

## 2.3 Розробка гри на HTML, JS та CSS

Для просування ростерії «Soloway» було продумано розробити гру в телеграм-боті, що в свою чергу виконувала б роль програми-лояльності для заохочення клієнтів. Дана гра написана на HTML, з використанням JS та CSS.

HTML – це мова розмітки гіпертексту, що застосовується для створення веб-застосунків. Вона обробляється браузером і відображається у вигляді документа, є невід'ємною складовою і основою для багатьох веб-сторінок.

Хоча HTML є потужним інструментом, створити професійний та повноцінний веб-сайт неможливо. У більшості випадків він використовується для додавання елементів тексту та для створення структур вмісту.

Мова розмітки добре працює з CSS та JavaScript, двома іншими інтерфейсами. CSS використовується для створення стилю, тобто тла, вибору кольорових елементів, макетів та анімації. JS надає можливість додавати динамічні функції.

Зміст гри полягає у наступному – користувачу на вхід подаються числа зліва від знаку «=» та одне число справа. Потрібно встановити оператори «+» та «-», щоб ліва частина була рівна правій.

Маркетингова стратегія передбачає, що відвідувачі зможуть скоротити час очікування на замовлення за допомогою гри, а також будуть мати можливість набравши 50 балів у грі отримати безкоштовну каву, що є у меню.

На рисунку 2.3 буде зображено інтерфейс гри, що також виконаний у фірмових кольорах.

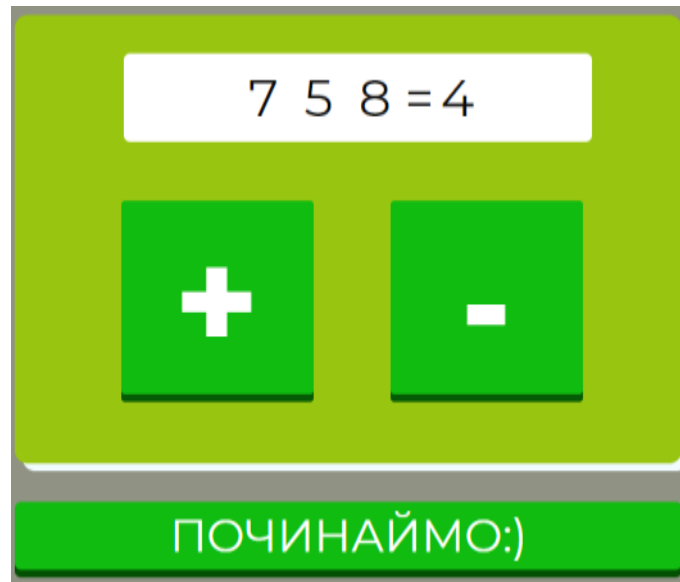


Рисунок 2.3 – Система лояльності для клієнтів у вигляді гри

На рисунку 2.4 можна спостерігати результат роботи. У полі з цифрами бачимо не вірний оператор, що приводить до помилково обчислення, який буде не рівний правій частині рівності. У блоці з текстом можна спостерігати поточний рахунок та максимальний рахунок за весь час гри користувача.



Рисунок 2.4 – Можливий результат гри системі лояльності

Програми лояльності клієнтів є важливою частиною маркетингової кампанії і при правильному підході можна перетворити нових клієнтів на постійних і тих, що любитимуть бренд, що, в свою чергу, призведе до зростання бізнесу.

## 2.4 Розробка телеграм-бота для просування ростерії «Soloway»

Для того, щоб створити телеграм-бота найперше потрібно розробити план його роботи, який включає всі пункти озвучені замовником. Даний чат-бот буде ділитись на дві частини: адміністративну та клієнтську, що показано на рисунку 2.5.



Рисунок 2.5 – Структура телеграм-боту «Soloway»



Як бачимо з рисунку 2.5 чат-бот наділений багатьма функціями, такими як робота з меню, оброблення замовлень, проведення транзакцій.

#### 2.4.1 Створення чат-боту

Для того, щоб зробити ЧБ у Telegram [28] потрібно отримати токен API [29], що є інтерфейсом на основі HTTP. Для цього нам потрібно звернутись до The BotFather, як показано на рисунку 2.6.



Рисунок 2.6 – Створення нового бота в BotFather

За допомогою команди `/newbot` створюється телеграм-бот. Йому потрібно обрати назву та ім'я користувача, і далі отримується унікальний токен для керування ботом, адже він обов'язково повинен бути у кодї, щоб програма чи платформа розробки розуміли з яким саме ботом вони працюють.

Оскільки токен для кожного чат-бота унікальний, його потрібно зберегти і не афішувати, щоб хтось не зміг його викрасти і використовувати бота у власних цілях.

The BotFather також надсилає низку команд для керування і налаштування бота. Серед них є зміна імені, опису, зображення. Також можна видалити бот або згенерувати новий токен.

Щоб взаємодіяти з телеграм-ботами користувачам потрібно надсилати їм певні команди, наприклад «/star», «/cancel», що володіють певними параметрами або ж використовувати вбудовану клавіатуру (ІК).

Щоразу, коли бот надсилає повідомлення, він в своєму інтерфейсі має спеціальну клавіатуру із заздалегідь визначеними параметрами відповіді. У той час, коли натискається певна кнопка, що була заздалегідь визначена, програма надсилає відповідну команду, це може бути текстове повідомлення, картинка, геолокація та інші види даних. У такий спосіб спрощується комунікація між клієнтом та ботом.

Telegram підтримує текст та смайлики у вбудованих клавіатурах.

Для того, щоб створити проект потрібно використати PyCharm та встановити необхідні бібліотеки, такі як «python-telegram-bot», що забезпечує інтерфейс Python для API Telegram Bot та «sqlalchemy» – бібліотека абстракції БД. Для початку створимо меню менеджера у стилі RelpyKeyboardMarkup [30], де кнопки знаходяться під полем для введення повідомлення, що показано на рисунку 2.7, який буде містити наступні пункти:

1. Страви;
2. Замовлення;
3. Створити транзакцію;
4. Список транзакцій;
5. .csv файл з проведеними транзакціями;
6. Редагувати менеджерів;

## 7. Режим замовника.

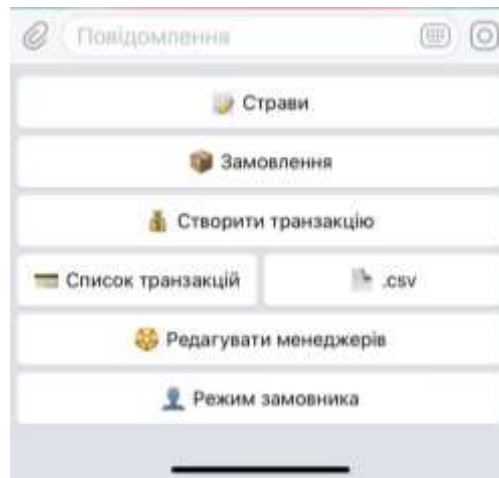
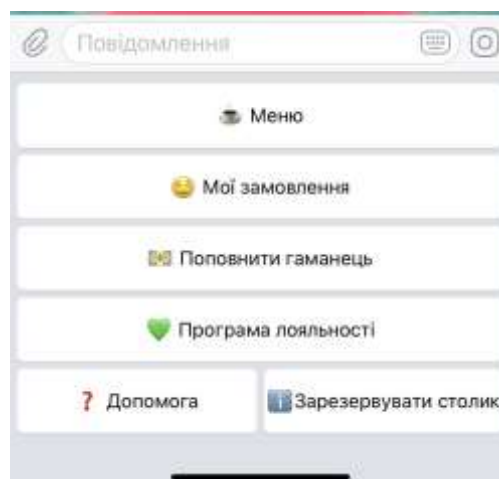


Рисунок 2.7 – Меню адміністратора чат-боту ростерії «Soloway»

Далі створимо меню для клієнта, також у стилі `RelpyKeyboardMarkup`, що показано на рисунку 2.8, який буде містити наступні пункти:

1. Страви;
2. Замовлення;
3. Створити транзакцію;
4. Список транзакцій;
5. .csv файл з проведеними транзакціями;
6. Редагувати менеджерів;
7. Режим замовника.



## Рисунок 2.8 – Меню клієнта чат-боту ростерії «Soloway»

У додатку В буде представлено лістинг файлу «admin\_user.py» з реалізацією меню для адміністратора та користувачів.

Наступним завданням була реалізація кожного пункту з меню, тобто написання логіки програми, а також дозволяти користувачам вибирати інформацію, яку їм потрібно отримати. На рисунку 2.9 буде показано схему папок та файлів реалізованих на Python.

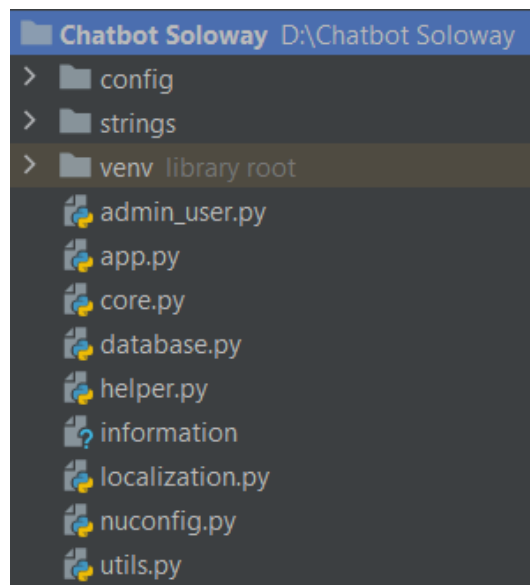


Рисунок 2.9 – Структура проекту

Папка config використовується для налаштування параметрів мови, параметрів баз даних, підключення токена для боту Telegram, також для налаштування часу очікування перед відправкою повідомлення ботом. При розвитку боту, коли буде доступна платіжна система, то у цій папці потрібно вказати токен для кредитної картки, мінімальні та максимальні значення зарахування коштів. Оскільки в даному чат-боті не була під'єднана платіжна система через неможливість укладання договору, для альтернативи було створено тестову систему для транзакцій. Код з папки config буде вказано у додатку Г.

Папка `string` надає можливість зміни мов. У даному випадку під'єднана українська.

Папка `venv`, містить всі необхідні файли що потрібні для проекту Python.

Файл `admin_user.py` містить в собі меню для адміністратора та клієнта, включає функції, що проводять обробку повідомлень, відправлення відповідей ботом, створення замовлень, та інших дій.

Файл `app.py` відповідає за бронювання. За допомогою нього створюється, скасовується, переглядається бронювання. Ця частина боту допомагає уникнути конфлікту і не очікувати на телефонну відповідь адміністратора. Всі дані вносять в окремий файл, в майбутньому також можна буде під'єднати до власної системи ростерії.

Файл `helper.py` надає можливість записатись в список очікування і якщо на певний день та час звільниться місце, бот про це повідомить і можна буде здійснити бронювання, що показано на рисунку 2.10.

```
def subscribe_user_to_waiting_list(context, user_id):
    chosen_day = date(YEAR, int(context.user_data['month']), int(context.user_data['day']))
    waitinglist = sess.query(WaitingList).filter(
        WaitingList.user_id == user_id,
        WaitingList.day == chosen_day,
    ).first()
    if waitinglist:
        return 'Ви вже в списку очікування на ' + chosen_day.strftime("%Y-%m-%d") + '!'
    new_waitinglist = WaitingList(
        user_id=user_id,
        day=chosen_day,
    )
    sess.add(new_waitinglist)
    sess.commit()
    return 'Ви були успішно додані до списку очікування на ' + chosen_day.strftime("%Y-%m-%d") + \
        '\n'. Бот нагадає Вам, якщо резерв на цей день буде відмінено'
```

Рисунок 2.10 – Частина коду, що відповідає за додавання в список очікування

Файл `database.py` створює файл бази даних, використовуючи SQLite, кросплатформу вбудовану БД, що підтримує певний набір команд SQL.

У файлах `nunconfig.py` та `util.py` реалізовано `logging`, тобто модуль для реєстрації подій для бібліотек та програм.

## 2.5 Тестування телеграм-боту

Після того як було розроблено сценарій роботи та програму чат-боту протестуємо його вручну в самому месенджері, використавши додаток на мобільному пристрої з ОС – IOS.

Першим кроком відкриваємо додаток, знаходимо телеграм-бот за псевдонімом «`@your_coffee_shop_bot`», щоб почати чат, як адміністратор пишемо команду «`/start`», адже до цього моменту наш діалог був пустий, що зображено на рисунку 2.11.

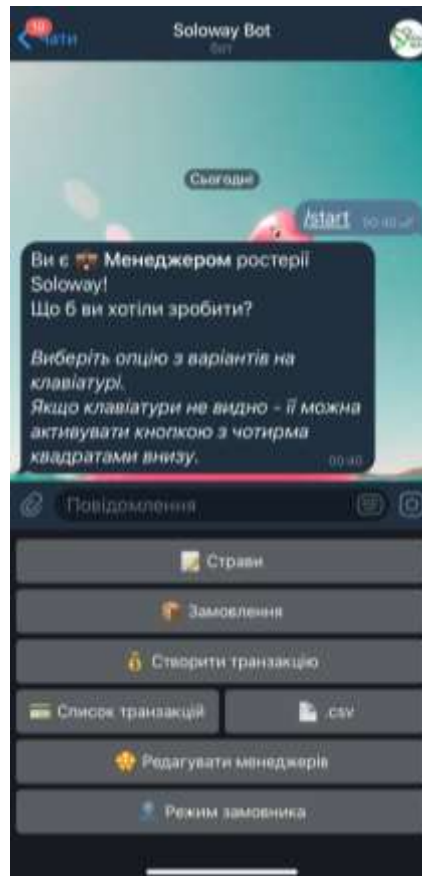


Рисунок 2.11 – Початок діалогу з ботом з боку адміністратора

Далі, якщо вибрати пункт «Страви», відкриється ще одне меню у якому можна додавати нові страви, видаляти їх, а також редагувати страви, які вже

були створені раніше. Розглянемо детальніше додавання нових страв, що зображено на рисунку 2.12.

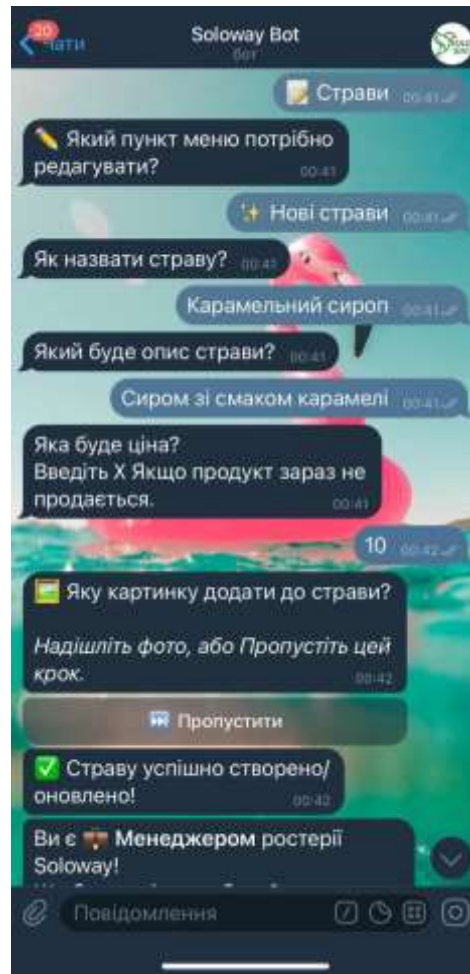


Рисунок 2.12 – Додавання нового пункту меню

Як бачимо з рисунку 2.12 у пункту «Нові страви», потрібно надати назву, також можна описати нову страву, додати зображення або ж пропустити останній пункт. Нам повідомляється про успіх операції та кава або ж їжа додається в БД.

Якщо натиснути клавішу «Замовлення», менеджер або ж люди, які також мають доступ до адміністративної частини можуть приймати замовлення, що надійшли від клієнта, поставивши їх у статус. У відповідь клієнту прийде сповіщення про процес приготування страви.

На рисунку 2.13 зображено виконання клавіші «Створити транзакцію».

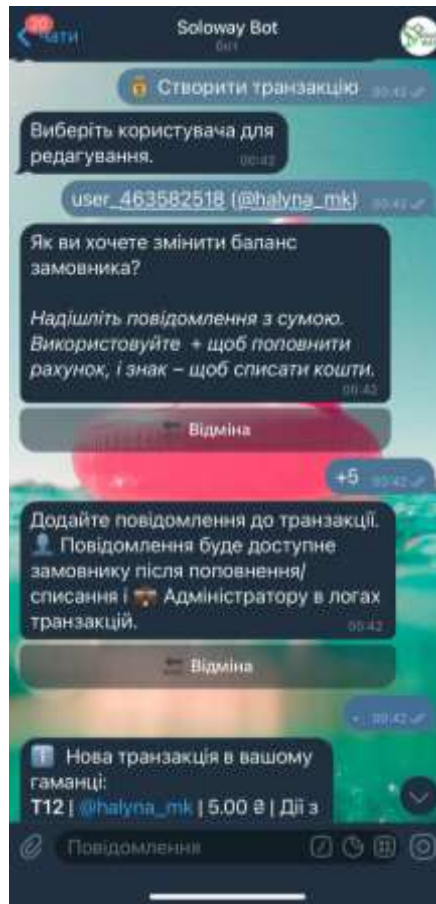


Рисунок 2.13 – Виконання дії «Створити транзакцію»

Щоб повернути кошти або ж списати з рахунку клієнта потрібно ввести потрібну суму, з вибраним знаком, також можна залишити коментар клієнту. У разі успішного виконання отримаємо повідомлення про зарахування чи списання коштів.

Усі операції з переказами можна дізнавшись після натиснення «Список транзакцій», що підтягує дані з БД, або ж скачати .csv файл для зручності перегляду.

«Редагувати менеджерів» дозволить людині, яка має доступ до адміністративної частини додавати інших співробітників, у кожного з яким буде свій ID Telegram, щоб могли проводити всі зазначені вище операції.

На рисунку 2.14 розглянемо тепер «Режим замовника» та виберемо «Меню», щоб переглянути відображення страв.



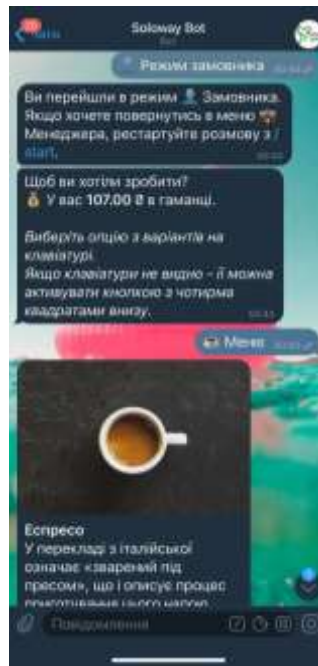


Рисунок 2.14 – Режим замовника

Бачимо обрану адміністратором картинку, назву та опис страви, щоб мати змогу додати або прибрати страву з кошика, а також оформити замовлення, потрібно скористатись InlineKeyboardMarkup, що прив'язується до конкретного повідомлення, як показано на рисунку 2.15.

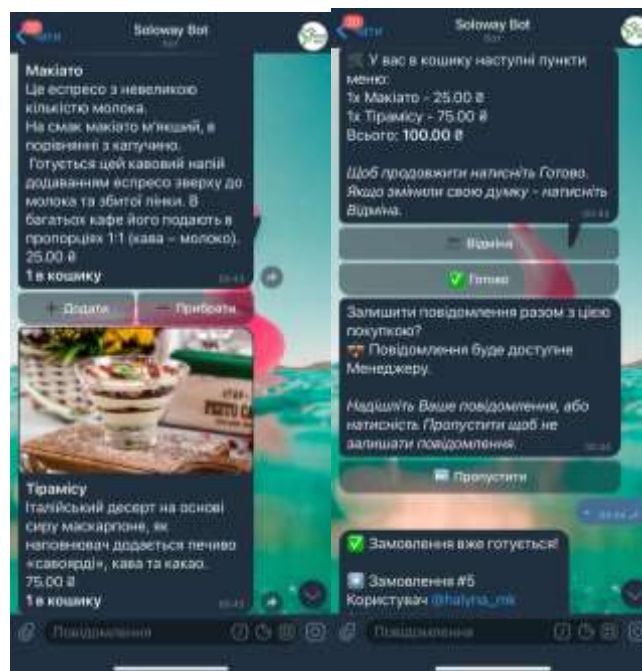


Рисунок 2.15 – Оформлення замовлення

Також можемо спостерігати з рисунку 2.15, що після оформлення замовлення нам з'являється кошик у якому можна оформити замовлення або його відмінити. Після успішної оплати відбудеться повідомлення про номер та ID замовлення, а менеджер у свою підтвердить приготування страви

У додатку Д буде зображено роботу клавiш «Поповнення гаманця», «Програма лояльності» та «Зарезервувати столик».

## **2.6 Висновок до другого розділу**

В другому розділі кваліфікаційної роботи було розглянуто вимоги замовника для розробки чат-боту на платформі Telegram, визначено фірмові кольори та створено логотип.

Програма лояльності є важливим маркетинговим інструментом, тому для телеграм-боту було розроблено гру, що надає можливість залучити більшу кількість клієнтів. Також у другому розділі було створено телеграм-бота «Soloway Bot», та написано програмний код для його роботи на мові Python.

Завдяки тестуванню розробленого бота зроблено висновки, що середовище є робочим і виконує всі вимоги замовника.

## **3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ**

### **3.1 Маркетингова діяльність на підприємстві**

Маркетингова діяльність – планування, організація, контроль і реалізація маркетингових програм, політик, стратегій і тактик, покликаних створити і задовольнити попит на пропозиції товарів або послуг фірм, як засіб отримання прийняттого прибутку [31].

Важливим компонентом є комплексне дослідження ринку, тобто дослідження, споживачів, виявлення незадоволених потреб, знайомство з діяльністю конкурентів.

Одним з найважливіших пунктів маркетингової діяльності є розробка маркетингової стратегії, де визначаються маркетингові цілі, розробляється відповідний підхід до клієнта, визначається цінова політика (ціноутворення на товари та послуги, розвиток цін), створюються стратегії або рішення щодо зміни ціни.

До товарної політики входить рішення щодо розробки нового товару, зняття з виробництва, модифікація товарів, планування масштабу виробництва. Після цього визначається політика розподілу – вибір системи розподілу, вибір оптимальних каналів та спосіб поширення, знаходження каналів.

Комунікаційна політика включає в себе вибір ефективних форм просування продукції, організація реклами, стимулювання збуту, PR-кампанія. Потрібно також пам'ятати про маркетинговий аналіз та зовнішній аудит маркетингу, маркетингові цілі та стратегії, аналіз результатів. В результаті аналізу можна створити ефективні рішення на ринку підприємства.

Потрібно зробити аналіз споживачів, як існуючих, так і потенційних. Суть полягає у аналізі демографічних, економічних, географічних та соціальних аспектів тих людей, які здійснили рішення про покупку.

До аналізу та планування продукції входить процес створення нових продуктів, вдосконалення старих або відкинення тих, що вже є застарілі.

Маркетингова діяльність орієнтована не лише на покупців, а й на агентів та прямих продавців. Реалізація вдалої цінової політики, тобто створення систем планування ціноутворення допоможе в експорті товарів, вигідних умовах кредитування та задоволені всіх сторін.

В даний час існує безліч видів маркетингу і маркетингових стратегій, кожне підприємство виходячи зі своїх потреб вибирає те, що найбільше підходить для досягнення поставленої мети.

Як зазначив Говард Шульц, споживачі очікують набагато більше від бренду, вони готові платити скоріше за враження ніж продукцію [32].

Ростерія «Soloway» – компанія харчової промисловості України, виробник кави та засновниця мережі кав'ярень-ростерій.

Оскільки компанія на ринку України доволі давно, її основною метою є збільшити довіру та вподобання людей. Компанія має низку продуктів та пропозицій, що підходять як для людей, що не мають великої фінансової спроможності, так і для тих, хто хоче отримати більше за свої кошти.

Ростерія «Soloway» позиціонує себе, як «місце з великою концентрацією любові до кави» [33].

Цілі компанії:

- покращити комунікацію з клієнтами;
- зробити ростерію місцем, куди будуть хотіти заходити справжні поціновувачі кави;
- ефективно адаптуватися до змін на ринку, постійно вдосконалювати маркетингову стратегію;

– збільшити прибуток компанії за допомогою відповідальних бізнес-підходів, спрямованих на збільшення клієнтів та просування мережі кав'ярень.

Персонал ростерії «Soloway» має спеціальну освіту, або пройшов професійну підготовку з метою оволодіння знаннями, навичками і кваліфікацією для виконання конкретних функцій. Прийом на роботу персоналу здійснюється на конкурсній основі за результатами кваліфікаційних випробувань і тестування.

Для кожної конкретної посади персоналу розроблена посадова інструкція, що встановлює функції, обов'язки, права і відповідальність працівника, обсяги, порядок і вимоги до якості виконуваних робіт, до професійної освіти, технічних знань і досвіду роботи.

Організаційна структура ростерії «Soloway» представлена на рисунку 3.1:

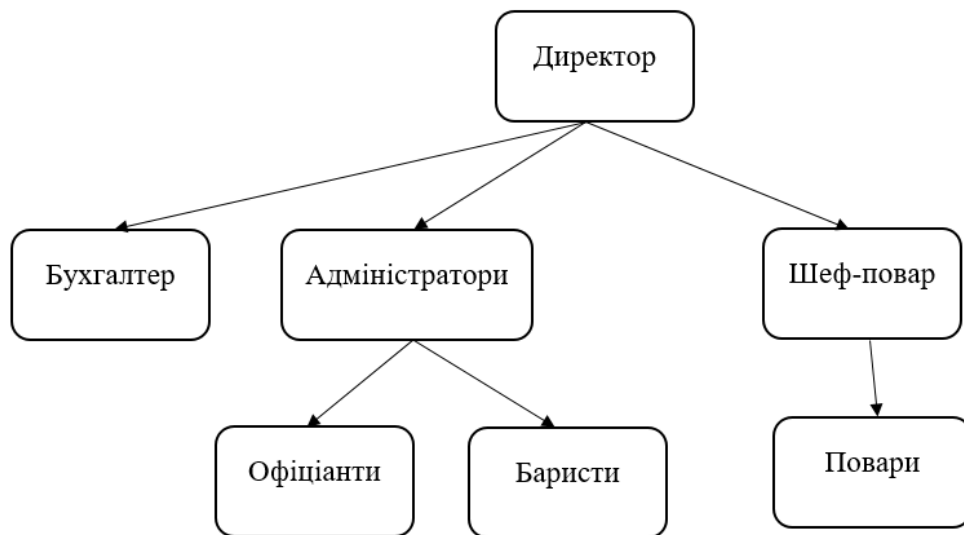


Рисунок 3.1 – Організаційна структура

Маркетингова діяльність в ростерії «Soloway» здійснюється безпосередньо директором – саме він вирішує, як, коли, які і яким чином

будуть використані інструменти маркетингу, спрямовані на підвищення привабливості та конкурентоспроможності підприємства. В окремих випадках пропозиції можуть бути запропоновані адміністратором, але рішення про прийняття та впровадження будь-якого заходу завжди стоїть за керівником ростерії «Soloway». Посилаючись на джерело [34], можу сказати, що більшість людей погано розуміє, що таке маркетинг і плутають її з рекламою. Для керівництва важливо вміти показати, що собою являє насправді даний заклад.

Ростерія «Soloway» використовує такі методи просування:

- зовнішня реклама – вивіска, покажчики, інформаційні таблички.
- інтернет-реклама. Як показує статистика, цільова аудиторія для такого типу бізнесу – це, в основному, активні інтернет-користувачі.– POS-матеріали - фірмові візитки, флаєра, буклети, що нагадують про заклад.
- введення чат-боту на платформі телеграм для просування та заохочення клієнтів.
- внутрішній маркетинг – внутрішні акції і свята, компліменти і подарунки, програми лояльності та дисконтні програми. такі заходи можна умовно розділити на календарні і некалендарний. Календарні – це коли до якогось свята або події прив'язується цікавий захід або акція.

До не календарних програм лояльності відноситься:

- картки зі знижкою 7% для постійних гостей;
- знижки іменинникам 25%;
- безкоштовний напій при проходженні гри в телеграм-боті.

У висновок слід зазначити, що будь-які ефективні методи просування, які б використовувала ростерія, якість послуг, що надаються і продукції має постійно вдосконалюватися і розвиватися. Фактично для того, щоб забезпечити успіх закладу протягом тривалого періоду, його бренд повинен мати цінність на ринку, а також важливо присутність постійних клієнтів.

### 3.2 Суть та зміст управління охороною праці.

Управління охороною праці – підготовка, прийняття і реалізація рішень щодо здійснення організаційних, технічних, санітарно-гігієнічних і лікувально-профілактичних заходів, спрямованих на забезпечення безпеки, збереження здоров'я та працездатності людини в процесі праці [35].

В умовах сучасного виробництва окремі заходи щодо поліпшення умов праці виявляються недостатніми, тому вони здійснюються комплексно, утворюючи систему управління охороною праці (СУОП) – сукупність об'єкта управління та керуючої частини, пов'язаних каналами передачі інформації. Об'єктом управління служить безпека праці на робочому місці і характеризується впливом людей з предметами і знаряддями праці.

Основним принципом державної політики в галузі охорони праці є забезпечення пріоритету життя та здоров'я працівників по відношенню до результатів виробничої діяльності [36].

СУОП – частина загальної системи управління (менеджменту) підприємств, що забезпечує управління ризиками в областях охорони здоров'я та безпеки праці, пов'язаних з діяльністю підприємств [37].

Мета СУОП в ростерії «Soloway» – забезпечити безпечні і нормальні умови праці для працівників на всіх стадіях виробничого процесу; створити умови, при яких забезпечується не тільки своєчасне усунення будь-яких порушень норм з охорони праці, а й попередження можливості їх виникнення.

Воно полягає у:

- зменшенні шкідливих або небезпечних чинників;
- збільшенні кількості часу на навчання щодо безпеки та умов праці;
- обов'язком звітувати про персональну відповідальність з питань охорони праці;

– контролю з боку керівництва щодо виконання нормативно-правових актів з питань охорони праці з боку роботодавця.

Одним з найважливіших елементів системи управління охороною праці є контроль. Основним призначенням контролю є аналіз результатів, одержуваних внаслідок керуючого впливу на об'єкт управління. Тільки чітко організований контроль в процесі управління створює в керуючій системі атмосферу оперативності виконання та забезпечує високу ефективність, що у результаті надає потрібний результат.



## ВИСНОВКИ

У ході виконання кваліфікаційної роботи освітнього рівня «Бакалавр» було створено телеграм-бота для просування ростерії «Soloway». Бот розроблений на мові програмування Python з використанням баз даних для зберігання інформації.

У першому розділі розглянуто актуальність питання створення чат-боту, досліджено історію-виникнення перших програмних застосунків з використанням штучного інтелекту.

Проаналізовано платформи для розміщення чат-боту, такі як Telegram, Viber, WhatsApp, а також методи створення ботів за допомогою спеціалізованих програмних застосунків або мов програмування. Обґрунтовано вибір платформи для інтеграції та мови програмування.

У другому розділі визначено вимоги замовника, спроектовано адміністративну та клієнтську частину телеграм-боту. У ролі менеджера користувач може проводити операції з пунктами меню, транзакціями, отримувати замовлення та інформацію про здійснені транзакції.

Клієнт у свою чергу може переглянути меню, зробити замовлення або ж його відмінити, поповнити рахунок для замовлення та зробити бронювання у закладі на певну годину.

Також було проведено тестування телеграм-боту, підтверджено його справність.

У розділі «Безпека життєдіяльності, основи охорони праці» розглянуто маркетингову стратегію та досліджено суть та зміст охорони праці у ростерії «Soloway».

## ПЕРЕЛІК ДЖЕРЕЛ

- 1 Gregory A. 9 Tips for Providing Excellent Customer Service [Електронний ресурс] / Alyssa Gregory. – 2019. – Режим доступу до ресурсу: <https://www.thebalancesmb.com/providing-excellent-customer-service-2951744>
- 2 MacDonald S. 5 WAYS TO DELIVER EXCELLENT CUSTOMER SERVICE (WITH EXAMPLES) [Електронний ресурс] / Steven MacDonald. – 2021. – Режим доступу до ресурсу: <https://www.superoffice.com/blog/five-ways-to-deliver-excellent-customer-service/>.
- 3 Акуліч М. Чат-боты и маркетинг / Маргарита Акуліч., 2018. – 82 с.
- 4 The History Of Chatbots – From ELIZA to ALEXA [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://onlim.com/en/the-history-of-chatbots/>.
- 5 Sean B. Chatbot History: What is Dr. Sbaitso [Електронний ресурс] / Sean. – 2020. – Режим доступу до ресурсу: <https://yakbots.com/chatbot-history-what-is-dr-sbaitso/>.
- 6 Rodrigues A. A History of SmarterChild [Електронний ресурс] / Ashwin Rodrigues. – 2016. – Режим доступу до ресурсу: <https://www.vice.com/en/article/jpgpey/a-history-of-smarterchild>.
- 7 DORMEHL L. Today in Apple history: Siri debuts on iPhone 4s [Електронний ресурс] / LUKE DORMEHL. – 2020. – Режим доступу до ресурсу: <https://www.cultofmac.com/447783/today-in-apple-history-siri-makes-its-public-debut-on-iphone-4s/>.
- 8 Perez S. Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Електронний ресурс] / Sarah Perez. – 2016. – Режим доступу до ресурсу: <https://techcrunch.com/2016/03/24/microsoft-silences-its-new-a-i-bot-tay-after-twitter-users-teach-it-racism/>.
- 9 Rouhiainen L. How AI and Data Could Personalize Higher Education [Електронний ресурс] / Lasse Rouhiainen. – 2019. – Режим доступу до

ресурсу: <https://hbr.org/2019/10/how-ai-and-data-could-personalize-higher-education>.

10 Юдин А. ТОП МЕССЕНДЖЕРОВ В УКРАИНЕ И МИРЕ 2020 [Электронный ресурс] / Антон Юдин. – 2020. – Режим доступа до ресурсу: <https://marketer.ua/top-messengers-in-ukraine-and-the-world/>.

11 Pope L. 45 SMS Marketing Statistics Your Customers Wish You Knew [Электронный ресурс] / Lauren Pope. – 2020. – Режим доступа до ресурсу: <https://learn.g2.com/sms-marketing-statistics>.

12 WHY CUSTOMERS PREFER TEXT MESSAGES TO PHONE CALLS [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.brightlink.com/why-customers-prefer-text-messages-to-phone-calls/>.

13 Most popular global mobile messaging apps 2021 Published by H. Tankovska, Feb 10, 2021 As of January 2021, two billion users were accessing the WhatsApp messenger on a monthly basis. The app's reach is usage penetration is particularly strong in markets [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>.

14 Татьяна Д. Для 50,6% Telegram — основной мессенджер. Результаты опроса AIN.UA [Электронный ресурс] / Грицьук Татьяна. – 2021. – Режим доступа до ресурсу: <https://ain.ua/2021/01/26/dlya-506-telegram-osnovnoj-messendzher-rezultaty-oprosa-ain-ua/>.

15 Коцофанэ О. Все о чат-ботах: типы и примеры, какому бизнесу подойдет, список конструкторов для создания [Электронный ресурс] / Ольга Коцофанэ. – 2019. – Режим доступа до ресурсу: <https://web-promo.ua/blog/vse-o-chat-botah-tipy-i-primery-kakomu-biznesu-podojdet-spisok-konstruktorov-dlya-sozdaniya/>.

16 Создаем бота на Aimylogic для автоматизации интернет-магазина [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: [https://habr.com/ru/company/just\\_ai/blog/416245/](https://habr.com/ru/company/just_ai/blog/416245/).

17 Wouters J. Flow XO Review [Электронный ресурс] / Joren Wouters. – 2021. – Режим доступа до ресурсу: <https://chatimize.com/reviews/flowxo/>.

18 BroBot – бот для продвижения в Инстаграм со встроенным автоответчиком [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://instagramsoft.ru/brobot-programma-dlya-prodvizheniya-instagram.html>.

19 BotKit — создание бота для мессенджера ТамТам без программирования [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://vc.ru/tribuna/71759-botkit-sozdanie-bota-dlya-messendzhera-tamtam-bez-programmirovaniya>.

20 28 сервисов по созданию чат-ботов в Telegram [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://vc.ru/services/110661-28-servisov-po-sozdaniyu-chat-botov-v-telegram>.

21 Programming language [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.computerhope.com/jargon/p/programming-language.htm>.

22 PYPL PopularitY of Programming Language [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://pypl.github.io/PYPL.html>.

23 TIOBE Index for May 2021 [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.tiobe.com/tiobe-index/>.

24 Rauschmayer D. JavaScript for impatient programmers (ES2021 edition) / Dr. Axel Rauschmayer., 2021. – 319 с.

25 Schildt H. Java: A Beginner's Guide, Eighth Edition / Herbert Schildt., 2018. – (McGraw-Hill Education).

26 Shaw Z. Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code / Zed A. v Shaw., 2016. – 306 с.

27 What is Photoshop [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://www.agitraining.com/design-news/photoshop-training-news/what-photoshop>.

28 Malizia N. How To Create Your First Telegram Bot [Електронний ресурс] / Nicola Malizia. – 2018. – Режим доступу до ресурсу: <https://unnikked.ga/how-to-create-your-first-telegram-bot-9005c08a5aa5>.

29 Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.

30 Урок 5. Клавиатуры и кнопки [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://surik00.gitbooks.io/aioqram-lessons/content/chapter5.html>.

31 Bhardawaj A. What is Marketing Management? [Електронний ресурс] / Bhardawaj. – 2019. – Режим доступу до ресурсу: <https://www.economicdiscussion.net/marketing-management/what-is-marketing-management/31788>.

32 Мілліган З. Міфи про брендинг / З. Мілліган, С. Бейлі., 2020. – 256 с. – (Фабула).

33 Soloway. Ростерія "Soloway" [Електронний ресурс] / Soloway. – 2021. – Режим доступу до ресурсу: <https://solowaycoffee.com/>.

34 Хармон К. Что я знаю о работе кофейни / Колин Хармон., 2019. – 270 с. – (Азбука).

35 Банько В. Д. Охорона праці в туристському комплексі [Електронний ресурс] / В. Д. Банько. – 2017. – Режим доступу до ресурсу: <https://ukrtextbook.com/oxorona-praci-v-turistskomu-kompleksi-banko-v-g/oxorona-praci-v-turistskomu-kompleksi-banko-v-g-2-1-sut-zmist-i-struktura-upravlinnya-oxoronoju-praci-v-turistskomu-kompleksi.html>.

36 Управление охраной труда [Електронний ресурс] – Режим доступу до ресурсу: <https://stroilogik.ru/bezopasnost-truda/112-upravlenie-ohranoi-truda.html>.

37 Управление охраной труда в организации [Електронний ресурс] – Режим доступу до ресурсу: [https://studme.org/12461220/bzhd/upravlenie\\_ohranou\\_truda\\_organizatsii](https://studme.org/12461220/bzhd/upravlenie_ohranou_truda_organizatsii).



# ДОДАТКИ

## Тези конференції

*IV Міжнародна студентська науково - технічна конференція  
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"*

Міністерство освіти і науки України,  
Тернопільський національний технічний університет  
імені Івана Пулюя  
Маріборський університет (Словенія)  
Технічний університет в Кошице (Словаччина)  
Каунаський технологічний університет (Литва)  
Львівський національний університет  
імені Івана Франка,  
Гірничо-металургійна академія ім. Станіслава Сташиця  
(Польща)  
Луцький національний технічний університет,  
Чернівецький національний університет  
імені Юрія Федьковича,  
Вроцлавський економічний університет (Польща)  
Донбаська державна машинобудівна академія



*Студентське наукове товариство*



**IV МІЖНАРОДНА**  
студентська науково - технічна конференція  
**"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ**  
**НАУКИ.**

**АКТУАЛЬНІ ПИТАННЯ"**

28-29 квітня 2021 р.

*(збірник тез конференції)*

*Тернопіль 2021*



IV Міжнародна студентська науково - технічна конференція  
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

## З М І С Т

Секція: Інформаційні технології

Величко Д. <b>ПРОБЛЕМИ НАКОПИЧЕННЯ ЕЛЕКТРОННИХ ВІДХОДІВ</b>	<b>3</b>
Гірша Ю. <b>ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В КОНТЕКСТІ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ</b>	<b>4</b>
Кузьо М. <b>ЗАСТОСУВАННЯ СТЕКУ ELK В ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ ДЛЯ КІБЕРБЕЗПЕКИ</b>	<b>5</b>
Гніздох В., Притоцький О., Масвський Т. <b>ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В СИСТЕМАХ ДЛЯ ОПРАЦЮВАННЯ ВІДОМОСТЕЙ ЩОДО COVID-19</b>	<b>7</b>
Данильців О., Хом'як А., Назаревич Т. <b>ВИКОРИСТАННЯ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ ОЦІНЮВАННІ СТАНУ РОСЛИН В РОЗУМНИХ ТЕПЛИЦЯХ</b>	<b>8</b>
Kashosi Aser, Nazarevych T. <b>HEART RATE VARIABILITY ANALYSIS TOOLKIT FOR FURTHER ANALYSIS OF HUMAN STRESS</b>	<b>10</b>
Тригубець Б. <b>ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ В ЕЛЕКТРОННІЙ КОМЕРЦІЇ</b>	<b>11</b>
Крамаров Ю. <b>ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ APPLE XCODE</b>	<b>13</b>
Крамаров Ю. <b>СТВОРЕННЯ ВЛАСНОЇ КАРТИ ЗА ДОПОМОГОЮ APPLE МАРКІТ</b>	<b>14</b>
Мушинська Г. <b>АКТУАЛЬНІСТЬ ЧАТ-БОТУ У СФЕРІ БІЗНЕСУ</b>	<b>16</b>
Павлюс В., Мацюк А., Слободян П., Яскілка О. <b>ВИБІР КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМ МІСТА</b>	<b>17</b>
Пясецький В., Масвський Т. <b>АУТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ НА ОСНОВІ ВІДБИТКІВ ПАЛЬЦІВ</b>	<b>19</b>
Пясецький В., Масвський Т. <b>БІОМЕТРИЧНІ ЗАСОБИ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ</b>	<b>20</b>
Шевченко Н., Горбуляк Ю., Масвський Т. <b>АНАЛІЗ ПРОТОКОЛУ OSPF</b>	<b>21</b>

УДК 621.326

Мушинська Г. – ст.гр. СН-41

*Тернопільський національний технічний університет імені Івана Пулюя*

## **АКТУАЛЬНІСТЬ ЧАТ-БОТУ У СФЕРІ БІЗНЕСУ**

Науковий керівник: д.е.н, проф. Струтинська І.В.

Mushynska H.

*Ternopil Ivan Puluj National Technical University*

## **RELEVANCE OF CHATBOT IN THE FIELD OF BUSINESS**

Supervisor: Iryna Strutynska

Ключові слова: Чат-бот, бізнес, клієнт

Keywords: Chatbot, business, client

У сучасному світі бізнес-організації різних форм власності часто зустрічаються з диверсифікованими проблемами, зокрема, з понаднормовими витратами, недостатньою підтримкою, супроводом клієнтів та зменшенням ефективності діяльності тощо. Для того, щоб покращити надання послуг та у свою чергу зробити підприємство економічно прогресивним, конкурентоспроможним не достатньо покращити виробничі процеси, а потрібно зробити новий крок для розвитку бізнесу, залучити та інтегрувати у власні бізнес-процеси дієві цифрові маркетингові інструменти.

Одне з найбільш поширених питань у бізнесі – обслуговування клієнтів. Чат-боти у цьому випадку допомагають не лише скоротити витрати, але також збільшити точність даних та зменшити затримку у відповіді клієнту на його запит. Згідно з даними американського журналу Forbes, 80% маркетологів планують використовувати чат-боти у своїх бізнес-процесах вже у 2021 році, що підтверджує актуальність даного цифрового інструменту. Тому, можна впевнено спрогнозувати, що в майбутньому чат-боти витіснять звичні інструменти, телефонні розмови та e-mail-листування.

На разі, основною проблемою введення чат-боту бізнесом є недовіра клієнтів до даного інструменту, адже вітчизняне суспільство все ще надає перевагу людському ресурсу понад штучним інтелектом. І хоча ця технологія ще не скоро зможе замінити електронну пошту та телефонний зв'язок повністю, оскільки деякі питання потребують людського втручання, чат-боти є технологією, яка допомагає подолати «вузькі місця» у каналах комунікації та покращити роботу організації.

Щоб створити ефективний чат-бот потрібно визначити бізнес-вимоги компанії, зрозуміти основні та допоміжні бізнес-процеси. Він має не лише їм відповідати, але й вирішувати поставлену проблему. Не менш важливим є вибір правильної платформи для реалізації та інтегрування чат-боту. Інтерфейс повинен бути зрозумілим для клієнтів, а платформа бути надійною з точки зору інфраструктурного забезпечення.

Боти зі штучним інтелектом стали необхідною частиною маркетингової стратегії, що допомагає раціонально налагодити комунікаційну складову бізнесу, забезпечити взаємодію із цільовою аудиторією 24/7. В найближчому майбутньому, за допомогою штучного інтелекту, боти будуть більш вдосконалені, а люди в свою чергу знайдуть інші способи для їхнього використання і застосування не лише у бізнесі, але і у всіх сферах життя. Таким чином, можна впевнено стверджувати, що технологія чат-ботів радикально трансформує методи взаємодії бізнесу з клієнтами.

## Додаток Б

### Порівняння платформ для створення ЧБ

Назва сервісу	Безкоштовний період	Можливості	Інтеграція з CRM	Інші інтеграції
1	2	3	4	5
Aimylogic	14 днів	Розумні дзвінки, голосові боти, конструктор навичок для голосових асистентів	Бітрікс24	WhatsApp, Telegram, Viber, Facebook, VK, Instagram, онлайн-чати, голосові помічники
BroBot	Без терміну для одного акаунту та соцмережі	Автоматизація лайків, репостів, запрошень у групи і друзі, т.д.	Немає	VK, Facebook, Instagram, Mamba, LovePlanet
Flow XO	Без терміну з обмеженнями	Аналітика взаємодій з користувачами	Так	Facebook Messenger, Telegram, Twilio SMS, Clack, сайт
BotKits	7 днів	Тригерні розсилки по базі користувачів	Своя міні-CRM	VK, Telegram, Facebook, Viber, Яндекс.Аліса
Botmother	Без терміну для одного боту	Аналітика взаємодій, тригерні розсилки по базі користувачів	Через сервіс-інтегратори Zapier, Albato, ApiX-Drive	Telegram, Viber, WhatsApp, Facebook, VK, сайт

## Додаток В

### Лістинг з файлу admin\_user.py

```
import datetime
import logging
import os
import queue as queueem
import re
import sys
import threading
import traceback
import uuid
from html import escape
from typing import *

import requests
import sqlalchemy.orm
import telegram

import database as db
import localization
import nuconfig

log = logging.getLogger(name)

def user_menu(self):
    log.debug("Displaying user_menu")
    while True:
        # Створення клавіатури головного меню
        keyboard =
[[telegram.KeyboardButton(self.loc.get("menu_order"))],
telegram.KeyboardButton(self.loc.get("menu_order_status"))],
telegram.KeyboardButton(self.loc.get("menu_add_credit"))],
telegram.KeyboardButton(self.loc.get("menu_booking"))],
telegram.KeyboardButton(self.loc.get("menu_help")),
telegram.KeyboardButton(self.loc.get("menu_bot_info"))]]
        self.bot.send_message(self.chat.id,
credit=self.Price(self.user.credit),
reply_markup=telegram.ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True))
        # Очікувати відповідь від користувача
        selection = self.wait_for_specific_message([
            self.loc.get("menu_order"),
            self.loc.get("menu_order_status"),
            self.loc.get("menu_add_credit"),
            self.loc.get("menu_booking"),
            self.loc.get("menu_help"),
            self.loc.get("menu_bot_info"),
        ])
        # Після відповіді користувача очікувати дані
```

```

self.update_user()
if selection == self.loc.get("menu_order"):
    self.order_menu()
elif selection == self.loc.get("menu_order_status"):
    self.order_status()
elif selection == self.loc.get("menu_add_credit"):
    self.add_credit_menu()
elif selection == self.loc.get("menu_booking"):
    self.booking_menu()
elif selection == self.loc.get("menu_bot_info"):
    self.bot_info()
elif selection == self.loc.get("menu_help"):
    self.help_menu()

def order_menu(self):
    """Меню, щоб замовити страву."""
    log.debug("Displaying order_menu")
    # Отримання списку страв з БД
    products =
self.session.query(db.Product).filter_by(deleted=False).all()
        cart: Dict[List[db.Product, int]] = {}
        continue
        message =
product.send_as_message(w=self, chat_id=self.chat.id)
        cart[message['result']['message_id']] = [product, 0]
        inline_keyboard = telegram.InlineKeyboardMarkup(
[[telegram.InlineKeyboardButton(self.loc.get("menu_add_to_cart")
, callback_data="cart_add")]]
        )
        if product.image is None:
            self.bot.edit_message_text(chat_id=self.chat.id,
message_id=message['result']['message_id'],
text=product.text(w=self),
reply_markup=inline_keyboard)
        else:
self.bot.edit_message_caption(chat_id=self.chat.id,
message_id=message['result']['message_id'],
caption=product.text(w=self),
reply_markup=inline_keyboard)
            # Створення кнопки скачування
            inline_keyboard =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_cancel"),

```

```

callback_data="cart_cancel"]])
    final_msg = self.bot.send_message(self.chat.id,

self.loc.get("conversation_cart_actions"),

reply_markup=inline_keyboard)
    while True:
        callback = self.wait_for_inlinekeyboard_callback()

        if callback.data == "cart_cancel":
            return

total_cost=str(self.get_cart_value(cart)),
            reply_markup=final_inline_keyboard)

        elif callback.data == "cart_done":
            break

def get_cart_summary(self, cart):
    # Створити суму на картці
    product_list = ""
    for product_id in cart:
        if cart[product_id][1] > 0:
            product_list += cart[product_id][0].text(w=self,

style="short",

cart_qty=cart[product_id][1]) + "\n"
    return product_list

def order_transaction(self, order, value):
    # Створення нової транзакції
    transaction = db.Transaction(user=self.user,
                                value=value,
                                order_id=order.order_id)

    self.session.add(transaction)
    # Підтвердження всіх змін
    self.session.commit()
    # Оновлення грошей юзера
    self.user.recalculate_credit()
    # Підтвердження змін
    self.session.commit()
    # Нагадування адміну за нові транзакції
    self.order_notify_admins(order=order)

def order_notify_admins(self, order):
    # Нагадування юзеру за статус замовлення
    self.bot.send_message(self.chat.id,
self.loc.get("success_order_created", order=order.text(w=self,

```

```

session=self.session,

user=True)))
    # Нагадування адміну за нове замовлення
    admins =
self.session.query(db.Admin).filter_by(live_mode=True).all()
    # Створення меню для замовлення
    order_keyboard = telegram.InlineKeyboardMarkup(
        [

[telegram.InlineKeyboardButton(self.loc.get("menu_complete"),
callback_data="order_complete")],

[telegram.InlineKeyboardButton(self.loc.get("menu_refund"),
callback_data="order_refund")]
        ])

    def order_status(self):
        """Відображення статусу замовлення
        log.debug("Displaying order_status")
        orders = self.session.query(db.Order) \
            .filter(db.Order.user == self.user) \
            .order_by(db.Order.creation_date.desc()) \
            .limit(20) \
            .all()

        if len(orders) == 0:
            self.bot.send_message(self.chat.id,
self.loc.get("error_no_orders"))

        for order in orders:
            self.bot.send_message(self.chat.id,
order.text(w=self, session=self.session, user=True))

    def make_payment(self, amount):
        # Ствоєрння оплати
        self.invoice_payload = str(uuid.uuid4())
        # Створення масиву оплати
        prices =
[telegram.LabeledPrice(label=self.loc.get("payment_invoice_label
"), amount=int(amount))]

        # Створення клавіатури
        inline_keyboard =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_pay"),

pay=True)],

[telegram.InlineKeyboardButton(self.loc.get("menu_cancel"),

```

```

callback_data="cmd_cancel"]])
    # Підтвердження оплати
    self.bot.send_invoice(self.chat.id,

title=self.loc.get("payment_invoice_title"),

description=self.loc.get("payment_invoice_description",
amount=str(amount)),
                                payload=self.invoice_payload,

provider_token=self.cfg["Payments"]["CreditCard"]["credit_card_t
oken"],
                                start_parameter="tempdeeplink",

currency=self.cfg["Payments"]["currency"],
                                prices=prices,

need_name=self.cfg["Payments"]["CreditCard"]["name_required"],

need_email=self.cfg["Payments"]["CreditCard"]["email_required"],

need_phone_number=self.cfg["Payments"]["CreditCard"]["phone_requ
ired"],
                                reply_markup=inline_keyboard)
                                precheckoutquery =
self.wait_for_precheckoutquery(cancellable=True)
    if isinstance(precheckoutquery, CancelSignal):
        return

    self.bot.answer_pre_checkout_query(precheckoutquery.id,
ok=True)

    successfulpayment =
self.wait_for_successfulpayment(cancellable=False)
    # Створення нової транзакції в БД
    transaction = db.Transaction(user=self.user,

value=int(successfulpayment.total_amount) - fee,
                                provider="Credit Card",

telegram_charge_id=successfulpayment.telegram_payment_charge_id,

provider_charge_id=successfulpayment.provider_payment_charge_id)

    if successfulpayment.order_info is not None:
        transaction.payment_name =
successfulpayment.order_info.name
        transaction.payment_email =
successfulpayment.order_info.email
        transaction.payment_phone =
successfulpayment.order_info.phone_number

```



```

self.user.recalculate_credit()
self.session.commit()

def admin_menu(self):
    """Викликається коли юзер є адміністратором"""
    log.debug("Displaying admin_menu")
    while True:
        keyboard = []
        if self.admin.edit_products:
            keyboard.append([self.loc.get("menu_products")])
        if self.admin.receive_orders:
            keyboard.append([self.loc.get("menu_orders")])
        if self.admin.create_transactions:

keyboard.append([self.loc.get("menu_edit_credit")])

keyboard.append([self.loc.get("menu_transactions"),
self.loc.get("menu_csv")])
            if self.admin.is_owner:

keyboard.append([self.loc.get("menu_edit_admins")])
            keyboard.append([self.loc.get("menu_user_mode")])
            self.bot.send_message(self.chat.id,
self.loc.get("conversation_open_admin_menu"),

reply_markup=telegram.ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True))
            selection =
self.wait_for_specific_message([self.loc.get("menu_products"),
self.loc.get("menu_orders"),
self.loc.get("menu_user_mode"),
self.loc.get("menu_edit_credit"),
self.loc.get("menu_transactions"),
self.loc.get("menu_csv"),
self.loc.get("menu_edit_admins")])
            if selection == self.loc.get("menu_products"):
                self.products_menu()
            elif selection == self.loc.get("menu_orders"):
                self.orders_menu()
            elif selection == self.loc.get("menu_edit_credit"):
                self.create_transaction()
            elif selection == self.loc.get("menu_user_mode"):
                self.bot.send_message(self.chat.id,
self.loc.get("conversation_switch_to_user_mode"))
                self.user_menu()

```

```

        elif selection ==
self.loc.get("menu_edit_admins"):
    self.add_admin()

    elif selection == self.loc.get("menu_transactions"):
        # Open the transaction pages
        self.transaction_pages()

    elif selection == self.loc.get("menu_csv"):

        self.transactions_file()

    def products_menu(self):
        log.debug("Displaying products_menu")
        # Список страв з БД
        products =
self.session.query(db.Product).filter_by(deleted=False).all()
        # Створення назви
        product_names = [product.name for product in products]
            product_names.insert(0,
self.loc.get("menu_cancel"))
            product_names.insert(1,
self.loc.get("menu_add_product"))
            product_names.insert(2,
self.loc.get("menu_delete_product"))

            keyboard = [[telegram.KeyboardButton(product_name)] for
product_name in product_names]
                self.bot.send_message(self.chat.id,
self.loc.get("conversation_admin_select_product"),

reply_markup=telegram.ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True))
            selection =
self.wait_for_specific_message(product_names, cancellable=True)

            if isinstance(selection, CancelSignal):
                return

            elif selection == self.loc.get("menu_add_product")
self.edit_product_menu()
                elif selection ==
self.loc.get("menu_delete_product"):
                    # Видалення страв
                    self.delete_product_menu()
                else:

                    product =
self.session.query(db.Product).filter_by(name=selection,
deleted=False).one()

                    self.edit_product_menu(product=product)

```

```

def edit_product_menu(self, product: Optional[db.Product] =
None):
    """Додавання до ДБ або регування
    log.debug("Displaying edit_product_menu")

    cancel =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_skip"),

callback_data="cmd_cancel")]])
        while True:
self.bot.send_message(self.chat.id,
self.loc.get("ask_product_name"))
        if product:
            self.bot.send_message(self.chat.id,
self.loc.get("edit_current_value", value=escape(product.name)),
                reply_markup=cancel)
            name = self.wait_for_regex(r"(.*)",
cancellable=bool(product))
            if (product and isinstance(name, CancelSignal)) or \

self.session.query(db.Product).filter_by(name=name,
deleted=False).one_or_none() in [None, product]:
                break
            self.bot.send_message(self.chat.id,
self.loc.get("error_duplicate_name"))
            # Запит на опис
            self.bot.send_message(self.chat.id,
self.loc.get("ask_product_description"))
            # Відображення вже існуючого опису, якщо редагуємо
            if product:
                self.bot.send_message(self.chat.id,

self.loc.get("edit_current_value",
value=escape(product.description)),
                reply_markup=cancel)

            # Очікування відповіді
            description = self.wait_for_regex(r"(.*)",
cancellable=bool(product))
            # Запит на ціну
            self.bot.send_message(self.chat.id,
                self.loc.get("ask_product_price"))
            # Відображення назви, якщо редагуємо
            if product:
                self.bot.send_message(self.chat.id,

self.loc.get("edit_current_value",

value=(str(self.Price(product.price))

product.price is not None else 'Non in vendita')),
                if

```

```

                                reply_markup=cancel)
    price = self.wait_for_regex(r"([0-9]+(?:[.],[0-9]{1,2})?[Xx])",
                                cancellable=True)
    # Якщо ціна пропущена
    if isinstance(price, CancelSignal):
        pass
    elif price.lower() == "x":
        price = None
    else:
        price = self.Price(price)
    if not isinstance(price, CancelSignal) and price is not
None:
        price = int(price)
    # Запит на зображення продукта
    self.bot.send_message(self.chat.id,
self.loc.get("ask_product_image"), reply_markup=cancel)
    photo_list = self.wait_for_photo(cancellable=True)
    if not product:
        # Створення запису в ДБ
        product = db.Product(name=name,
                                description=description,
                                price=price,
                                deleted=False)
        # Додавання запису до ДБ
        self.session.add(product)
    # Якщо продукт редаговано
    else:
        # редагуємо з новими значеннями
        product.name = name if not isinstance(name,
CancelSignal) else product.name
        product.description = description if not
isinstance(description, CancelSignal) else product.description
        product.price = price if not isinstance(price,
CancelSignal) else product.pric
        if isinstance(photo_list, list):
            largest_photo = photo_list[0]
            for photo in photo_list[1:]:
                if photo.width > largest_photo.width:
                    largest_photo = photo
            photo_file =
self.bot.get_file(largest_photo.file_id)
            self.bot.send_message(self.chat.id,
self.loc.get("downloading_image"))
            self.bot.send_chat_action(self.chat.id,
action="upload_photo")
            product.set_image(photo_file)
            self.session.commit()
            self.bot.send_message(self.chat.id,
self.loc.get("success_product_edited"))

    def delete_product_menu(self):

```

```

        log.debug("Displaying delete_product_menu")
        products =
self.session.query(db.Product).filter_by(deleted=False).all()
        product_names = [product.name for product in products]
        product_names.insert(0, self.loc.get("menu_cancel"))
        keyboard = [[telegram.KeyboardButton(product_name)] for
product_name in product_names]
        self.bot.send_message(self.chat.id,
self.loc.get("conversation_admin_select_product_to_delete"),

reply_markup=telegram.ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True))
        selection =
self.wait_for_specific_message(product_names, cancellable=True)
        if isinstance(selection, CancelSignal):
            return
        else:
            product =
self.session.query(db.Product).filter_by(name=selection,
deleted=False).one()
            product.deleted = True
            self.session.commit()
            self.bot.send_message(self.chat.id,
self.loc.get("success_product_deleted"))

    def orders_menu(self):
        """Відображення нових замовлень
        log.debug("Displaying orders_menu")
        stop_keyboard =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_stop"),

callback_data="cmd_cancel")]])
        cancel_keyboard =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_cancel"),

callback_data="cmd_cancel")]])
        self.bot.send_message(self.chat.id,

self.loc.get("conversation_live_orders_start"),

reply_markup=telegram.ReplyKeyboardRemove())
        self.bot.send_message(self.chat.id,

self.loc.get("conversation_live_orders_stop"),
                                reply_markup=stop_keyboard)
        order_keyboard =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_complete"),

callback_data="order_complete")]],

```



```

        self.bot.delete_message(self.chat.id,
reason_msg.message_id)
        continue
        # Позначити замовлення повернутим
order.refund_date = datetime.datetime.now()
order.refund_reason = reply
order.transaction.refunded = True
order.user.recalculate_credit()
self.session.commit()
self.bot.edit_message_text(order.text(w=self,
session=self.session),
                                chat_id=self.chat.id,
message_id=update.message.message_id)
        self.bot.send_message(order.user_id,

self.loc.get("notification_order_refunded",
order=order.text(w=self,

session=self.session,

user=True)))
        self.bot.send_message(self.chat.id,
self.loc.get("success_order_refunded", order_id=order.order_id))

def create_transaction(self):
    log.debug("Displaying create_transaction")
    user = self.user_select()
    if isinstance(user, CancelSignal):
        return
    cancel =
telegram.InlineKeyboardMarkup([[telegram.InlineKeyboardButton(se
lf.loc.get("menu_cancel"),

callback_data="cmd_cancel")]])
        self.bot.send_message(self.chat.id,
self.loc.get("ask_credit"), reply_markup=cancel)
        reply = self.wait_for_regex(r"(-? ?[0-9]{1,3})(?:[.,][0-
9]{1,2})?", cancellable=True)
        if isinstance(reply, CancelSignal):
            return
        price = self.Price(reply)
        self.bot.send_message(self.chat.id,
self.loc.get("ask_transaction_notes"), reply_markup=cancel)
        reply = self.wait_for_regex(r"(.*)", cancellable=True)
        if isinstance(reply, CancelSignal):
            return
        transaction = db.Transaction(user=user,
                                value=int(price),
                                provider="Дії з коштами",
                                notes=reply)
        self.session.add(transaction)

```

```

        user.recalculate_credit()
        self.bot.send_message(user.user_id,

self.loc.get("notification_transaction_created",

    def help_menu(self):
        log.debug("Displaying help_menu")
        keyboard =
[[telegram.KeyboardButton(self.loc.get("menu_guide"))],

[telegram.KeyboardButton(self.loc.get("menu_contact_shopkeeper"
)],

[telegram.KeyboardButton(self.loc.get("menu_cancel"))]]
        self.bot.send_message(self.chat.id,

self.loc.get("conversation_open_help_menu"),

reply_markup=telegram.ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True))
        selection = self.wait_for_specific_message([
            self.loc.get("menu_guide"),
            self.loc.get("menu_contact_shopkeeper")
        ], cancellable=True)
        if selection == self.loc.get("menu_guide"):
            self.bot.send_message(self.chat.id,
self.loc.get("help_msg"))
            elif selection ==
self.loc.get("menu_contact_shopkeeper"):
                shopkeepers =
self.session.query(db.Admin).filter_by(display_on_help=True).joi
n(db.User).all()
                shopkeepers_string = "\n".join([admin.user.mention()
for admin in shopkeepers])
                self.bot.send_message(self.chat.id,
self.loc.get("contact_shopkeeper",
shopkeepers=shopkeepers_string))

```



## Додаток Г

### Код з файлу config.toml

```
default_language = "uk"
fallback_language = "en"

[Database]
engine = "sqlite:///database.sqlite"

# Telegram bot параметри
[Telegram]
token = "1648358535:AAG5j375pCSvw6Mme76h2HV1EgYCBD105FM"
# Час в секундах перед повідомленнями
conversation_timeout = 7200
# Час очікування для відправки наступних повідомлень
long_polling_timeout = 30

# Налаштування оплати
[Payments]
# ISO код валюти
currency = "UAH"
currency_symbol = "₴"

# Logging налаштування
[Logging]
# Повідомлення що друкуються в консолі
format = "{asctime} | {threadName} | {name} | {message}"
```

## Додаток Д

### Приклад виклику клавіші «Поповнення гаманця»

Оскільки раніше ми вже розглянули основні аспекти роботи телеграм-бота, поглянемо на поповнення електронно гаманця. Це можна зробити двома способами, заплатити готівкою адміністратору і він зарахує кошти на рахунок користувача або поповнити за допомогою кредитної картки, що показано на рисунку Д.1.

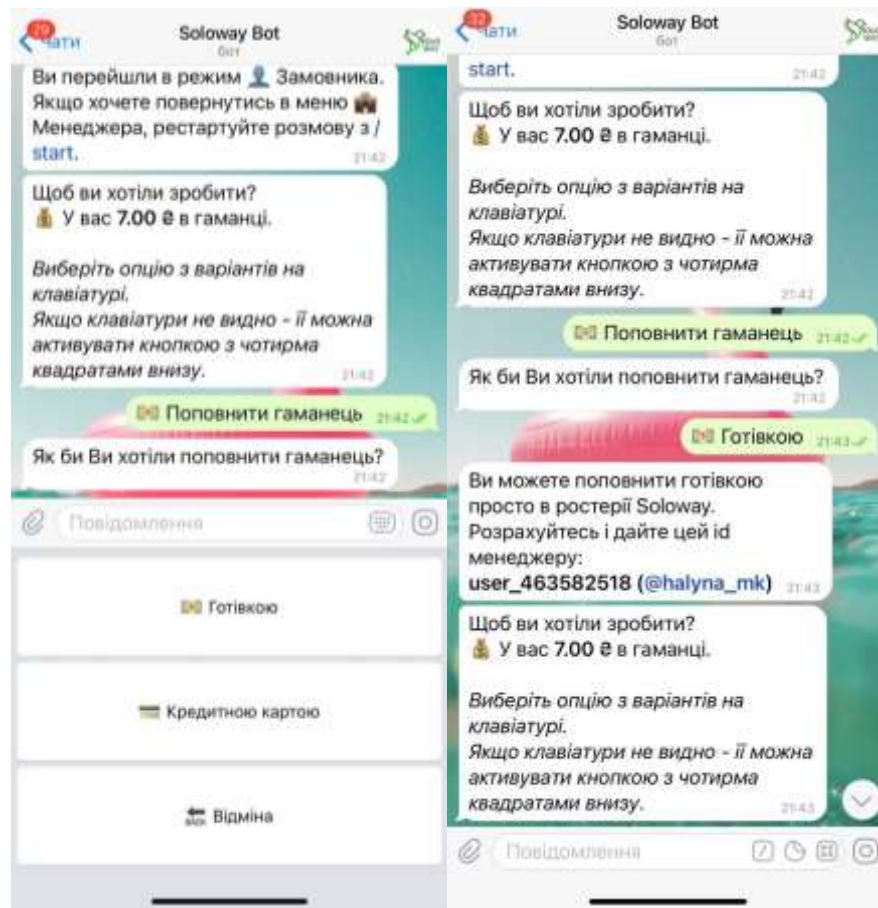


Рисунок Д.1 – Поповнення гаманця клієнта

На рисунку Д.2 будемо спостерігати за зарахуванням коштів зі сторони адміністратора, який отримав вказаний ідентифікатор клієнта.

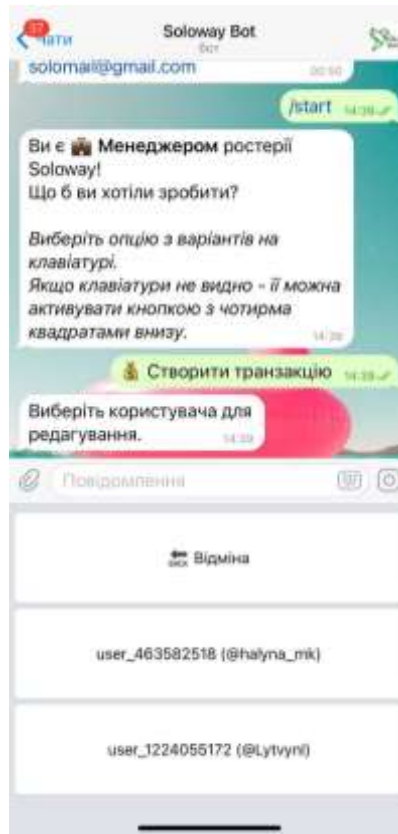


Рисунок Д.2 – Зарахування коштів клієнту зі сторони адміністратора

Після вкладання договору, можна буде поповнити гаманець за допомогою кредитної картки.

### **Приклад виклику клавіші «Програма лояльності»**

Про роботу програми лояльності було розказано у другому розділі, на рисунку Д.3 показано, що при виклику клавіші «Програма лояльності» клієнтові підтягує онлайн-гру, яку він може відкрити натиснувши на посилання.

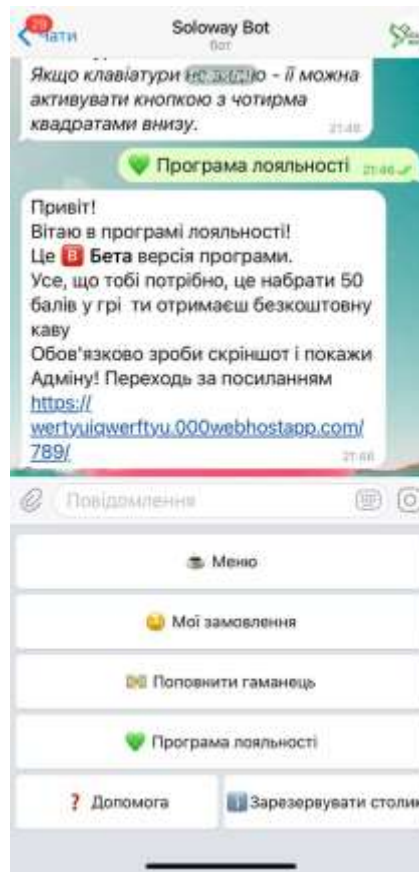


Рисунок Д.3 – Програма лояльності ростерії

На рисунку Д.4 буде показано, як відбувається гра та можливий результат клієнта.

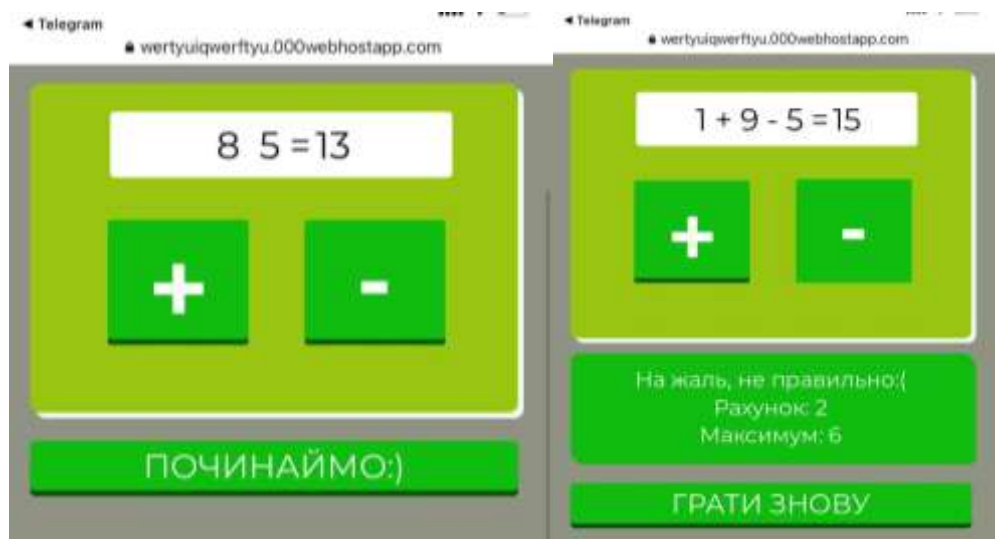


Рисунок Д.4 – Програма лояльності у вигляді гри

## Приклад виклику клавiші «Зарезервувати столик»

Резервація відбувається наступним чином: користувач обирає місяць, день та годину, щоб зробити бронь, якщо інший користувач не зарезервував цієї дати, то резервація підтверджується. У іншому випадку користувача додається у список очікування.

Щоб скасувати свій резерв потрібно обрати ідентифікаційний номер та видалити запис, як зображено на рисунку Д.5.



Рисунок Д.5 – Приклад створення та скасування резерву