МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних систем та мереж

_____(Ігор ЖУКОВ)

« ___ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ "БАКАЛАВР"
ЗА СПЕЦІАЛЬНІСТЮ 123 "КОМП'ЮТЕРНА ІНЖЕНЕРІЯ"

Тема: Мережевий сервіс доставки на основі Telegram-боту

Виконавець: Святослав МАЛИК

Керівник: Ігор ТЕЛЕШКО

Нормоконтролер: Сергій ЖУРАВЕЛЬ

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж
Спеціальність 123 "Комп'ютерна інженерія"
(шифр, найменування)

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних систем та мереж

_____(Ігор ЖУКОВ)

« ____ » _____ 2023 р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Малику Святославу Васильовичу
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема роботи: Мережевий сервіс доставки на основі Telegram-боту

затверджена наказом ректора від "26" квітня 2023 року № 591/ст.

2.Термін виконання роботи: з 22.05.2023 до 25.06.2023

3.Вихідні дані до роботи:

4.Зміст пояснювальної записки (перелік питань, що підлягають розробці):

5.Перелік обов'язкового графічного матеріалу:
Презентація *PowerPoint*

6. Календарний план

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка

7. Дата отримання завдання «22» травня 2023 р.

Керівник кваліфікаційної роботи Ігор ТЕЛЕШКО
(підпис)

Завдання прийняв до виконання Святослав МАЛИК
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи “Мережевий сервіс доставки на основі Telegram-боту”: 40 с., 22 рис., 17 літературних джерел.

TELEGRAM-БОТ, МЕРЕЖЕВИЙ СЕРВІС, *БОТ*, *BOTFATHER*, *SPRING BOOT*, *JAVA*, *MYSQL*, ОБСЛУГОВУВАННЯ, *TELEGRAM*.

Мета кваліфікаційної роботи – розробити Telegram-бот для замовлення та отримання послуг доставки.

Об’єкт проєктування – мережевий сервіс.

Предмет проєктування – *Telegram*-бот як інструмент реалізації мережевого сервісу.

Метод проєктування – теоретичне ознайомлення із існуючими технологіями для розробки *Telegram*-ботів на основі *Java*, *Spring Boot*.

Прогнози припущення щодо розвитку об’єкта дослідження – створення робочого чат-боту та використання його, як мережевий сервіс доставки.

Результати кваліфікаційної роботи рекомендується використовувати при розробці нових або покращення існуючих програмних засобів, які дозволяють надавати послуги у сфері обслуговування.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

SQL – *Structured Query Language* (структурована мова запитів)

TCP/IP – *Transmission Control Protocol/Internet Protocol* (протокол управління передачею/межмережевий протокол)

URL – *Universal Resource Locator* (адреса сторінки в Інтернеті)

ВСТУП

У зростаючій цифровій епосі, де швидкість та зручність є ключовими факторами для потреб сучасних споживачів, онлайн-сервіси стають необхідною складовою сфери електронної торгівлі та бізнесу загалом. Споживачі шукають простоту та зручність використання онлайн-послуг, які дозволяють замовляти товари або послуги в будь-який зручний час та місце. Швидкість та легкість процесу замовлення, простота навігації на веб-сайтах або мобільних додатках, а також зручність способів оплати і доставки стають важливими критеріями для споживачів.

За останні роки спостерігається зростання використання цифрових додатків для замовлення товарів та послуг. Однак, існуючі методи доставки часто не задовольняють вимоги зручності та ефективності, що призводить до появи нових інноваційних рішень. Один із таких прикладів є мережевий сервіс.

Мережевий сервіс відповідає потребам зручності та ефективності у повсякденному використанні. Це надає можливість споживачам замовляти швидку та зручну доставку, що дозволяє отримати товар за найкоротший термін. Також, підприємства зможуть надавати швидку та зручну доставку своїх товарів клієнтам.

Одним із прикладів мережевих сервісіх є *Telegram*. Це одна із найпопулярніших месенджер-платформ, що надає широкі можливості для автоматизації та забезпечення зручності у взаємодії з користувачами. Використання боту в якості основного інструменту для замовлення та керування доставкою відкриває шлях до забезпечення швидкого та простого процесу замовлення, сприяючи зменшенню зусиль і часу, витрачених споживачами на взаємодію із сервісом. Також боти мають ряд наступних переваг:

- зручність та доступність: доступність для спілкування та використання 24/7 на різних платформах, таких як мобільні додатки або веб-сайти;

- автоматизація та ефективність: автоматизація багатьох рутинних завдань та запитів, що звільняє час та зусилля співробітників. Надають швидкі та точні відповіді на питання, обробляють замовлення та резервації, надають інформацію про стан доставки та багато іншого.
- покращена клієнтська підтримка: забезпечують негайну підтримку та відповіді на запитання клієнтів. Надають інформацію про товари та послуги, вирішують проблеми та скарги, а також надають підтримку під час всього процесу використання послуги.
- персоналізація та індивідуальний підхід: персоналізують рекомендації та пропозиції на основі попередніх взаємодій з клієнтом. Адаптуються до унікальних потреб та вподобань кожного клієнта, що робить комунікацію більш індивідуалізованою та приємною.
- масштабованість: можуть обслуговувати багато клієнтів одночасно, що робить їх ефективними для великих обсягів замовлень та запитів. Легко масштабуються та забезпечують швидку та якісну обробку багатьох запитів одночасно.

Мета кваліфікаційної роботи: розробка *Telegram*-боту для сервісу доставки.

Завдання кваліфікаційної роботи:

- ознайомлення із поняттям, основними складовими мережевих сервісів;
- ознайомлення із поняттям про чат-ботів, зокрема *Telegram*-ботів;
- вибір середовища розробки, мови програмування, особливостями синтаксису;
- програмування *Telegram*-бота;
- перевірка можливостей налаштованого боту;

Практичне значення роботи полягає у реалізації *Telegram*-боту, що надає можливість отримати послугу доставки.

Для розробки боту було використано:

- *Java* – мова програмування;
- *Spring Boot* – фреймворк для розробки програми;
- *IntelliJ IDEA* – середовище розробки;

- *BotFather* – інструмент для розробки *Telegram*-боту.
- *MySQL* – база даних.

Подальший розвиток роботи:

- розширення можливостей функціоналу;
- надання можливісті постійного використання, шляхом перенесення боту на сервер;
- інтегрування із іншими сервісами або додатками.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття мережевих сервісів

Мережеві сервіси – це інформаційні ресурси або програмні засоби, що забезпечують доступ до різноманітних функціональних можливостей, таких як обмін даними, комунікація, зберігання і обробка інформації тощо.

Особливістю послуг інформаційно-комунікаційних мереж є їх доступність через мережу зв'язку, що надають можливість користувачам отримувати доступ до них з будь-якого місця, де є підключення до Інтернету [1].

1.2. Основні складові мережевих сервісів

1.2.1. Комунікаційні протоколи

Комунікаційні протоколи – набір правил і стандартів, що визначають, як пристрої в мережі обмінюються даними.

Одні з найпоширеніших комунікаційних протоколів є:

- *TCP/IP* – основний протокол для передачі даних в Інтернеті. *TCP* відповідає за розбиття даних на пакети, керування передачею пакетів та перевірку цілісності отриманих даних. *IP* відповідає за маршрутизацію пакетів, визначення *IP-адрес* та адресацію пристроїв в мережі.

- *HTTP* – протокол, що використовується для передачі веб-сторінок. У його функціонал входить визначення правил обміну гіпертекстовими документами між веб-серверами та клієнтськими програмами, такими як веб-браузери. Найпопулярніші типи повідомлень: GET (рис. 1.1), POST і PUT. На рисунку 1.1 зображено приклад надсилання GET-повідомлення.

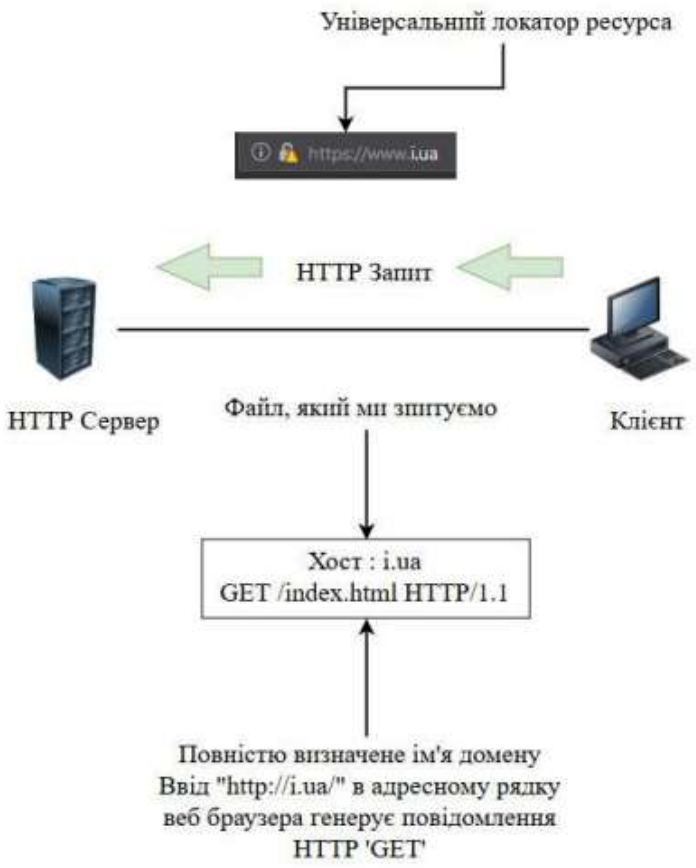

Рис. 1.1. Діаграма HTTP протоколу із використанням GET

- *SMTP* – протокол для передачі електронної пошти в мережі, що визначає правила доставки та передачі листів з одного поштового сервера на інший. Також забезпечує маршрутизацію, пересилання та доставку електронної пошти між поштовими серверами.
- *FTP* – протокол передачі файлів в мережі, що використовується для керування передачею файлів між клієнтами та серверами. Протокол *FTP* забезпечує аутентифікацію користувачів і захищене з'єднання з використанням шифрування, такого як *SSL* або *TLS*.

Окрім цих протоколів, існують інші комунікаційни протоколи, такі як *SNMP*, що використовується для керування та моніторингу мережевих пристроїв або *SSH*, що забезпечує безпечний віддалений доступ до серверів [2].

1.2.2. Комунікаційні послуги

Комунікаційні послуги є дозволяють ефективно передавати дані, спілкуватися і обмінюватися різними типами інформації.

Основні комунікаційні послуги:

- Передача даних – забезпечує передавання цифрової інформації, таку як тексти, зображення, відео та звук, між різними пристроями або мережами. Базується на комунікаційних протоколах, таких як *TCP/IP*.
- Електронна пошта – надає доступ користувачам обмінюватися повідомленнями та файлами електронним шляхом. Електронна пошта бере за основу протокол *SMTP*, який відповідає за передачу листів між поштовими серверами.
- Відеозв'язок – забезпечує візуальний контакт між користувачами, незалежно від їх географічного розташування, і базується на протоколах, таких як *SIP*.

- Голосова комунікація – передає голосові сигнали в реальному часі між різними пристроями. Протоколи, такі як *VoIP*, забезпечують передачу голосу через Інтернет та мережі пакетно вимірюваннями, а традиційні телефонні мережі використовують протоколи, такі як *PSTN*.

Крім вище вказаних послуг, мережеві послуги зв’язку включають інші форми обміну інформацією, такі як миттєві повідомлення (наприклад, за допомогою протоколів *XMPP* або *IRC*), спільне використання документів (наприклад, з використанням протоколу *WebDAV*) або передачу мультимедійних потоків (наприклад, з використанням протоколу *RTP*).

Вибір конкретних комунікаційних послуг залежить від потреб користувачів і контексту використання. У бізнес-середовищі є важливими послуги електронної пошти та відеозв'язку для спілкування з колегами та клієнтами, тоді як в особистому житті популярні послуги миттєвих повідомлень та голосового зв'язку для спілкування з друзями та родиною.

1.2.3. Системи зберігання даних

Системи зберігання даних забезпечують користувачам надійність, доступність, масштабованість ефективного зберігання, керування та отримання доступу до даних через мережу.

Приклади систем зберігання даних:

- Хмарні сховища – це сховища, що зберігають дані на віддалених серверах, які зазвичай керуються постачальниками хмарних послуг, дають можливість спільного доступу до файлів.
- Сервери файлів – спеціалізовані комп'ютери або сервери, які забезпечують централізоване зберігання та управління файлами для користувачів в мережі.
- Бази даних – надають зручний та ефективний спосіб зберігання, управління та отримання доступу до даних.

Системи зберігання даних, такі як хмарні сховища, сервери файлів і бази даних, надають користувачам зручний спосіб організації, зберігання та доступу до їх даних через мережу [3].

1.2.4. Сервіси спільної роботи

Сервіси спільної роботи надають зручні та ефективні засоби для спільної роботи над проектами, документами та іншими типами вмісту. Основна мета таких сервісів полягає в полегшенні комунікації, спільному доступі до файлів та управлінні робочим процесом.

Основна частина сервісів спільної роботи є спільний доступ до файлів. Користувачі зберігають свої файли в хмарних сховищах або на спеціальних серверах і ділитися цими файлами з іншими учасниками проекту або командою.

Також сервіси спільної роботи синхронізують дані між різними пристроями та користувачами. Тому внесені в файл або документ, автоматично оновлюються на всіх пристроях, що мають доступ до цього сервісу. Це робить можливим уникнути проблеми з несумісністю версій файлів і забезпечує гладкий процес спільної роботи.

Додатковою функціональністю є спільне редагування документів. Такий підхід полегшує колективну роботу над проектами та спільне редагування тексту, таблиць, презентацій тощо.

Сервіси також можуть надавати інструменти для управління завданнями та проектами. Користувачі можуть створювати списки завдань, встановлювати терміни виконання, призначати відповідальних осіб та відстежувати прогрес [4].

Крім того, вони забезпечують безпеку даних та контроль доступу. Користувачі можуть налаштовувати рівні доступу до файлів і документів, обмежувати права редагування або обмежувати доступ лише для певних осіб.

Приклади таких сервісів – *Google Документи, Microsoft Office 365, Dropbox.*

1.2.5. Веб-сервіси

Однією з ключових переваг веб-сервісів є їх універсальна доступність через веб-браузер та надає доступ користувачеві з будь-якого пристрою. Єдина вимога – наявність підключення до Інтернету.

Дані сервіси охоплюють різноманітні сфери діяльності. Наприклад, пошукові системи допомагають користувачам здійснювати пошук інформації в мережі Інтернет. Соціальні мережі, уможливлюють спілкуватися та обмінюватися відомостями з іншими користувачами.

Користувачі можуть шукати товари, здійснювати покупки та здійснювати оплату онлайн. Через банківські сервіси здійснюються операції переказу коштів, перегляд балансу та історію транзакцій через веб-інтерфейс.

Веб-сервіси базуються на веб-технологіях, таких як *HTML, CSS* та *JavaScript*. Вони використовуються для створення веб-сторінок, взаємодії з користувачем та передачі даних між сервером та клієнтом [5].

Забезпечення конфіденційності, цілісності та доступності даних є пріоритетним завданням для провайдерів веб-сервісів. Для запобігання несанкціонованому доступу та зловживанням використовуються різноманітні заходи безпеки, такі як шифрування даних, перевірка автентичності користувачів та механізми контролю доступу.

1.2.6. Сервіси безпеки

Основна мета сервісів безпеки полягає в забезпеченні конфіденційності, цілісності та доступності даних, а також у запобіганні несанкціонованому доступу до мережевих ресурсів. Сервіси безпеки мають на меті виявлення, запобігання та врегулювання цих загроз, забезпечуючи безпеку користувачів і їхніх даних.

Наприклад, файерволи використовуються для моніторингу трафіку в мережі і фільтрації небезпечних пакетів даних, що надходять або виходять з мережі.

Також сервіси безпеки включають антивірусне програмне забезпечення, яке сканує систему на наявність шкідливих програм і вірусів.

1.2.7. Віртуалізація мережі

Віртуалізація дозволяє створювати віртуальні мережі, що ізолюються від фізичної інфраструктури і ефективно використовує наявні ресурси, такі як пропускна здатність, обчислювальна потужність та апаратне забезпечення. Замість того, щоб мати окрему фізичну інфраструктуру для кожного сервісу, віртуалізація спільно використовує ресурси між різними віртуальними мережами [6].

Також пришвидшує конфігурацію мережевих сервісів. Відокремлення від фізичної інфраструктури дає змогу адміністраторам мережі легко налаштовувати параметри мережі, такі як маршрутизація, політики безпеки та якість обслуговування, без значних змін в фізичній інфраструктурі.

Віртуалізація мережі створює ізольовані віртуальні мережі, що забезпечують безпеку та конфіденційність даних. Кожна віртуальна мережа може мати власні політики безпеки, файерволи та механізми контролю доступу, що дозволяє управляти і обмежувати доступ до різних ресурсів в мережі [7].

Також масштабує сервіси, забезпечуючи гнучкість у розширенні та зменшенні обсягу ресурсів в залежності від потреб. Віртуалізація може додавати нові віртуальні мережі або змінювати розмір існуючих без необхідності в значних фізичних змінах. Це дає можливість підвищувати масштабованість мережі, адаптуватися до зростаючих потреб користувачів і забезпечувати швидке розгортання нових сервісів.

1.3. Поняття чат-боту

Чат-бот – це програма або штучний інтелект, який здатний автоматично взаємодіяти з людьми за допомогою текстових повідомлень. Він може бути використаний для різних цілей, таких як відповіді на запитання, надання інформації, підтримка користувачів, виконання завдань та багато іншого.

Вони можуть мати різні рівні складності і функціональності. Деякі прості чат-боти працюють за попередньо заданими правилами і надають відповіді на основі фіксованих шаблонів. Їх можна використовувати, наприклад, для автоматичних відповідей на часті запитання на веб-сайтах [8].

Також дозволяють інтегрувати технології штучного інтелекту, такі як обробка природної мови, машинне навчання і генерація тексту, щоб здійснювати більш гнучкі та контекстуальні розмови з користувачами. Вони розпізнають наміри користувачів, виконувати завдання, рекомендувати товари або послуги, а навіть імітувати розмову з реальними людьми настільки, що може бути важко відрізнити бота від реальної особи.

Застосовуються в багатьох галузях, включаючи комерцію, клієнтську підтримку, медицину, освіту, розваги та інші. Допомагають автоматизувати процеси, полегшують комунікацію з користувачами і можуть підвищити ефективність різних бізнес-процесів.

Додатково, чат-боти можуть забезпечувати постійну доступність, оскільки вони можуть працювати цілодобово без необхідності відпочинку або перерв. Це дає можливість користувачам отримувати відповіді на свої запитання у будь-який зручний для них час.

Їхня перевага полягає у здатності збирати та аналізувати дані про взаємодію з користувачами.

У майбутньому можливий подальший розвиток чат-ботів, з використанням нових технологій, таких як розпізнавання голосу, розуміння контексту та реалістичніша синтезе мови [9].

1.4. Telegram-бот

Telegram-бот є програмним агентом, який функціонує в месенджері Telegram і взаємодіє з користувачами через автоматизовані повідомлення. Принцип роботи Telegram-боту зображено на рисунку 1.2. Боти в Telegram можуть виконувати різноманітні завдання, включаючи надання інформації, виконання команд, взаємодію зі сторонніми сервісами, розповсюдження новин, генерацію відповідей на запитання та багато іншого.

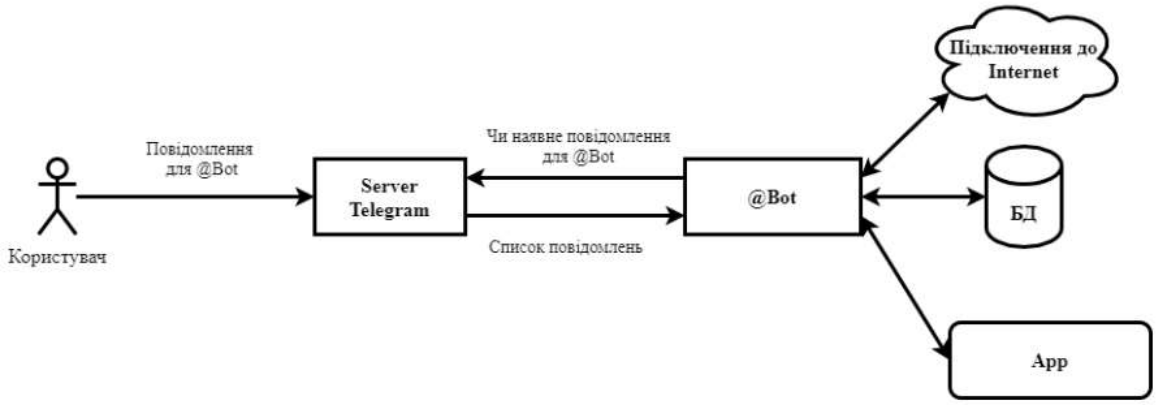

Рис. 1.2. Діаграма принципу роботи Telegram-боту

Особливості Telegram-ботів:

- комунікація з користувачами: отримування повідомлення від користувачів і надсилати їм відповіді. Наприклад, розпізнавати текстові команди, фотографії, відео, аудіо та інші типи повідомлень.
- автоматизована обробка запитів: виконання певні дії на основі отриманих запитів. Наприклад, шукати інформацію в Інтернеті, відтворювати медіафайли, проводити операції з базою даних, розраховувати і відправляти відповіді на запитання.
- інтеграція зі сторонніми сервісами: взаємодія зі сторонніми сервісами та API. Це дозволяє їм виконувати різноманітні завдання, такі як отримання погодних звітів, котирувань фінансових ринків, новин та багато іншого.

- команди та інтерфейси: мають свої власні команди та інтерфейси для взаємодії з користувачами. Наприклад, користувачі можуть викликати команди, щоб отримати певну інформацію або активувати певний функціонал бота.
- створення власного бота: *Telegram* надає спеціальний API для створення ботів. Можна використовувати мови програмування, такі як *Python*, *JavaScript*, Java або будь-яку іншу, що підтримує роботу з API. Далі потрібно створити бота в Telegram, отримати його токен і налаштувати відповідні обробники для отримання та відправки повідомлень [10].

Висновки до розділу

У даному розділі було розглянуто поняття про мережеві сервіси та основні складові, що входять до них, а саме:

- комунікаційні протоколи: *TCP/IP, HTTP, SMTP, FTP*;
- комунікаційні послуги: передача даних, електронна пошта, відеозв'язок, голосова комунікація;
- система зберігання даних: хмарні сховища, сервери файлів, бази даних;
- сервіси спільної роботи;
- веб-сервіси;
- сервіси безпеки;
- віртуалізація мережі;

Додатково було розглянуто поняття чат-боту, його можливості, переваги, застосування. Також було проведено опис використання *Telegram*-боту та виділено особливості у використанні.

В подальшій роботі потрібно проаналізувати, які засоби бот, який буде надавати необхідні користувачеві функціональні можливості та відповідати таким критеріям, як: зручність інтерфейсу, швидкість відгуку, безпека та надійність в обробленні даних про користувача, доступність на різних платформах.

РОЗДІЛ 2
ВИБІР ЗАСОБІВ ДЛЯ РОЗРОБКИ TELEGRAM-БОТА

2.1. Мова програмування

Мова програмування *Java* є об'єктно-орієнтованою мовою програмування, яка була розроблена компанією *Sun Microsystems* (пізніше *Oracle Corporation*). Відноситься до класу мов програмування "загального призначення" і використовується для розробки різноманітних типів програм та додатків.

Також програми можуть працювати на різних операційних системах без необхідності перекомпіляції. Це досягається завдяки використанню віртуальної машини *JVM*, яка виконує байт-код, отриманий після компіляції програми.

До переваг даної мови відносять вбудовану систему керування пам'яттю та механізми безпеки, які дозволяють уникнути багатьох типів помилок, таких як переповнення буфера або несанкціонований доступ до пам'яті.

Крім того є велика кількість бібліотек і фреймворків, що полегшують розробку програм. Вона підтримує різні області програмування, включаючи веб-розробку, мобільну розробку, наукові дослідження, ігрову розробку та багато інших.

Загалом, мова програмування Java є потужним інструментом для розробки програмного забезпечення, який комбінує простоту використання, переносимість і безпеку. Наприклад, її застосовують у різних галузях і є основою для багатьох технологій та платформ, таких як *Java EE* для підприємницьких додатків, *Android* для мобільних додатків та багато інших [11].

2.2. Середовище програмування

IntelliJ IDEA – IDE для розробки програмного забезпечення. Дане середовище має широкий набір інструментів та функцій, які полегшують процес програмування та збільшують продуктивність розробників.

Основні особливості та переваги [12]:

- розширена підтримка мов програмування: підтримує багато мов програмування, зокрема *Java, Kotlin, Scala, Groovy, JavaScript, TypeScript, HTML, CSS* і багато інших;
- інтелектуальне автодоповнення: *IDE* надає потужне автодоповнення коду, яке пропонує варіанти завершення, основані на контексті, типах даних, історії коду та інших факторах. Це значно прискорює процес написання коду та допомагає уникнути помилок.
- аналіз коду: аналізує код, надаючи рекомендації щодо оптимізації, виявлення помилок, створення тестів та вдосконалення стилю коду;
- вбудовані інструменти для рефакторингу: надає багато інструментів для полегшення рефакторингу коду, таких як перейменування змінних, витягування методів, оптимізація імпорту та інші;
- вбудована система керування версіями: присутня вбудована підтримка популярних систем керування версіями, таких як *Git, Subversion, Mercurial* та інші.
- інструменти для розробки веб-додатків: має розширену підтримку розробки веб-додатків, включаючи підтримку фреймворків, таких як *Spring, JavaServer Faces, Java Persistence API* та інших.
- інтеграція з іншими інструментами: підтримує інтеграцію з багатьма іншими корисними інструментами розробки, такими як системи збирання проектів (*Maven, Gradle*), сервери додатків (*Tomcat, JBoss*), бази даних (*MySQL, PostgreSQL, Oracle*) та багато інших.

2.3. Фреймворк Spring Boot

Spring Boot – це фреймворк для розробки додатків на мові Java, який спрощує процес створення самостійних, готових до виконання додатків на основі фреймворка Spring. У нього присутні стандартні конфігурації та утиліти, що допомагають швидко розгортати, налаштовувати та запускати додатки на основі Spring із мінімальними зусиллями.

Основні особливості та переваги Spring Boot [13]:

- зручна конфігурація: дозволяє розробникам зосередитися на функціональності додатків, а не на складних налаштуваннях. Автоматично налаштовує багато аспектів додатка на основі залежностей та конфігурацій, що зменшує кількість коду, який потрібно написати.
- вбудований сервер додатків: має вбудований сервер додатків, такий як *Apache Tomcat* або *Jetty*, що дозволяє запускати додатки безпосередньо без необхідності установки та налаштування додаткових серверів.
- залежності та автоконфігурація: надає широкий вибір готових модулів та бібліотек, які можна легко інтегрувати з вашим додатком. Використовує концепцію автоконфігурації, що дозволяє автоматично налаштовувати залежності та створювати біні відповідно до ваших потреб.
- управління залежностями: має вбудований інструмент для управління залежностями, відомий як *"Spring Initializr"*. Дозволяє швидкого створювати нові проєкти *Spring Boot* із необхідними залежностями та налаштуваннями.
- легкість тестування: включає зручні інструменти для тестування додатків, включаючи підтримку автоматичного створення тестових середовищ та інтеграційних тестів.
- моніторинг та керування: наявність вбудованих інструментв для моніторингу та керування додатками, включаючи вбудовані засоби збору метрик, здоров'я додатків, журналів тощо.
- підтримка мікросервісної архітектури: підходить для розробки мікросервісних архітектур, оскільки надає інструменти та бібліотеки для легкої розробки та керування мікросервісами.

Spring Boot є потужним фреймворком для швидкої розробки додатків на Java, який допомагає зосередитися на бізнес-логіці та продуктивності розробки.

2.4. Telegram-бот “BotFather”

“BotFather” – це спеціальний бот (рис. 2.1), створений командою Telegram, який дозволяє створювати та налаштовувати інші Telegram-боти. Бот надає користувачам можливість створювати нові боти, налаштовувати їх параметри та отримувати токени доступу, необхідні для використання API Telegram для бота.

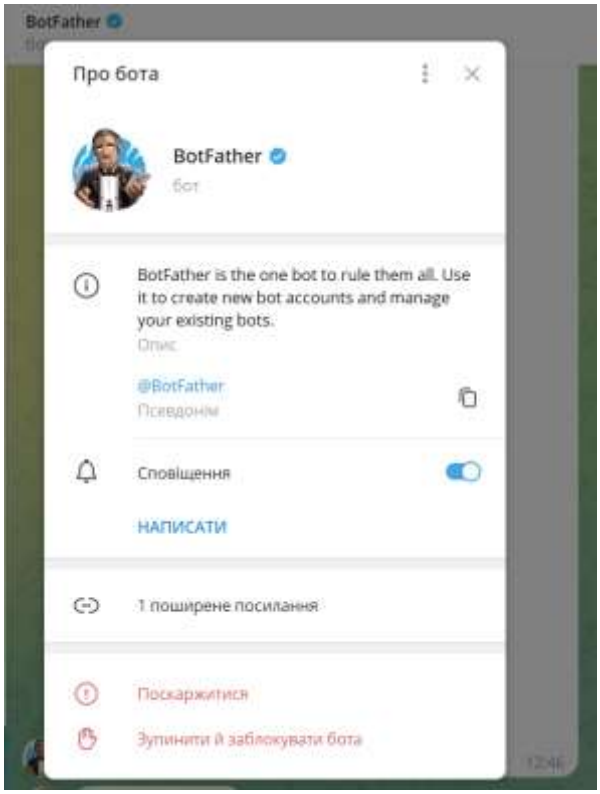

Рис. 2.1. Бот “BotFather”

Ключові особливості та можливості “BotFather” [14]:

- створення ботів: дозволяє створити нового бота, вказавши його назву та коротке ім'я. Після створення надає унікальний токен доступу, який використовується для з'єднання з API Telegram.
- налаштування ботів: дозволяє налаштувати різні параметри бота, такі як його опис, аватар, команди, сповіщення та багато інших. Користувачі можуть налаштувати бота так, як вони бажають, визначати його функціональні можливості та зовнішній вигляд.

- управління ботами: надає можливість управляти створеними ботами. Користувачі можуть змінювати налаштування, оновлювати опис, змінювати аватар та виконувати інші дії, пов'язані з керуванням ботом.
- отримання документації та допомоги: надає доступ до документації та ресурсів, пов'язаних з Telegram API та розробкою ботів. Користувачі можуть отримати детальну інформацію про можливості API, переглянути приклади коду та отримати допомогу щодо створення та налаштування своїх ботів.

Цей бот є потужним інструментом для створення та налаштування Telegram-ботів. Він дозволяє розробникам швидко створювати ботів з необхідними параметрами та функціональністю і почати їх використовувати для взаємодії з користувачами на платформі Telegram.

2.5. База даних

MySQL – одна з найпопулярніших відкритих систем управління базами даних (СУБД), яка надає широкі можливості для зберігання, керування та обробки структурованих даних. Вона використовує мову запитів *SQL* (*Structured Query Language*) для взаємодії з базою даних і забезпечує надійну, швидку та масштабовану роботу з даними.

Особливості та можливості *MySQL* [15]:

- структурована база даних: дозволяє створювати та управляти структурованою базою даних, яка складається з таблиць, стовпців і рядків. Також надає можливість визначати типи даних, обмеження цілісності, відношення між таблицями та інші атрибути;
- мова запитів *SQL*: підтримує повний набір *SQL*-операцій для створення, зчитування, оновлення та видалення даних. Наприклад, використання *SQL* для виконання складних запитів, з'єднувати таблиці, сортувати результати і багато іншого;

- транзакції та конкурентний доступ: забезпечує підтримку транзакцій, що дозволяє виконувати групу операцій як єдину атомарну одиницю. Також має механізми для управління конкурентним доступом до даних, що дозволяє кільком користувачам одночасно працювати з базою даних.
- індексування та оптимізація: надає можливості індексування, які дозволяють швидко знаходити та фільтрувати дані. Наприклад, створювати індекси на стовпцях для покращення продуктивності запитів. *MySQL* також має механізми оптимізації запитів, що дозволяють автоматично вибирати ефективний план виконання запиту;
- резервне копіювання та відновлення: дозволяє створювати резервні копії бази даних для забезпечення безпеки даних;
- розширення та підтримка: *MySQL* має велику спільноту користувачів та розробників, що надає доступ до розширень, допомоги та додаткових ресурсів.

MySQL є надійним рішенням для зберігання та обробки даних. Він використовується в багатьох мережевих сервісах, веб-додатках, системах керування вмістом, електронних комерційних платформах та інших проектах, де потрібна ефективна робота з базами даних [16].

Висновки до розділу

Цей розділ описує засоби для створення мережевого сервісу Telegram-боту доставки. Було розглянуто основні можливості, особливості та переваги до наступних засобів:

- *Java*, як мова програмування;
- *Intellij IDEA*, як середовище програмування;
- *Spring Boot*, як фреймворк для розробки програми;
- *BotFather*, як інструмент для розробки *Telegram*-боту.
- *MySQL*, як база даних.

В подальшій роботі потрібно розробити сервіс доставки на основі Telegram-боту із застосування вище згаданих засобів.

РОЗДІЛ 3
РОЗРОБКА МЕРЕЖЕВОГО СЕРВІСУ
НА ОСНОВІ TELEGRAM-БОТА

3.1. Реалізація бота «BotFather»

Для взаємодії бота із *Telegram API* потрібен токен. Це унікальний рядок символів, який використовується для ідентифікації і автентифікації в процесі взаємодії з API. Використовують його для аутентифікації бота і дозволу доступу до різних функцій, таких як отримання повідомлень, надсилання повідомлень, робота з клавіатурами і багато іншого. Токен є конфіденційною інформацією, яку необхідно зберігати у безпеці.

За допомогою бота «BotFather» можна згенерувати токен потрібно виконати наступні кроки:

1. Відкрити Telegram і знайти @BotFather в списку користувачів або ввести його у поле пошуку.
2. Натиснути на @BotFather, щоб відкрити його профіль.
3. Натиснути кнопку *"Start"* або надіслати команду */start*.
4. У списку команд, натиснути або ввести /newbot.
5. Ввести назву бота – «*DeliverMeNowBot*».
6. Далі ввести псевдонім бота – «*DeliverMeNowBot*».
7. Потім «BotFather» надсилає повідомлення, де буде згенерований токен домену (рис. 3.1).

Також бот дозволяє повторно генерувати токен, якщо невпевнені у конфіденційності та безпеці.

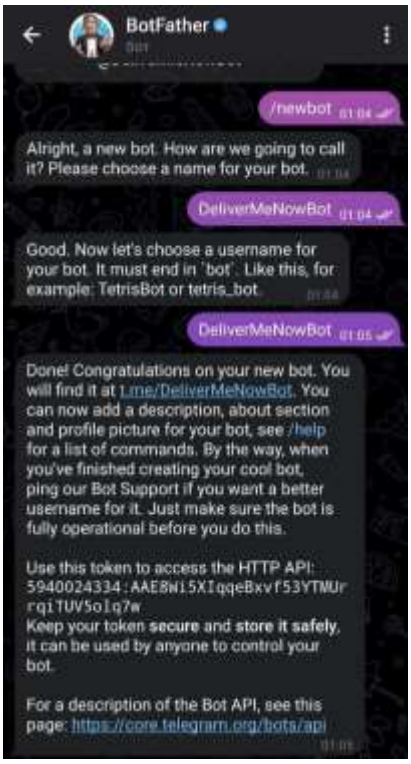

Рис. 3.1. Згенерований токен за допомогою «BotFather»

Далі у BotFather» вводиться команда «/setdescription» для встановлення текстового опису бота (рис. 3.2).

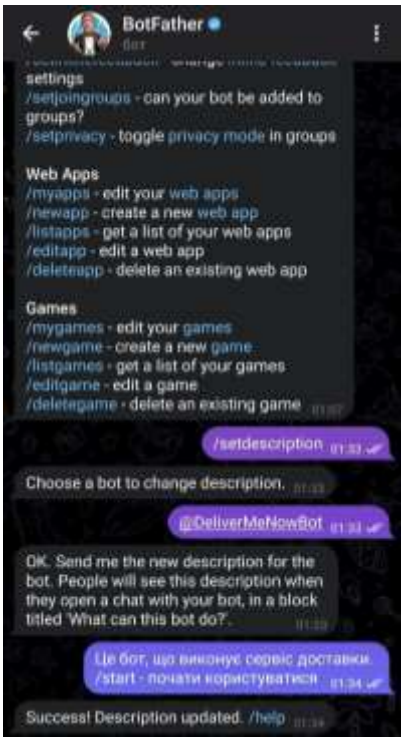

Рис. 3.2. Встановлення текстового опису бота

3.2. Реалізація MySQL

Для реалізації зберігання даних використовується MySQL. У базі даних зберігаються дані про користувача, його взаємодію із сервісом доставки та дані про заклади, із яких можна замовити товар.

Таблиця із даними про користувача створюється конфігурується за допомогою фреймворка *Spring Boot*.

Щоб зв’язати базу даних *MySQL* та *Spring Boot* виконується наступне:

1. Додається залежність у файл «build.gradle»:

implementation 'org.springframework.boot:spring-boot-starter-data-jpa';

2. Налаштовуються параметри підключення до бази даних MySQL у файлі «application.properties» (рис. 3.3).

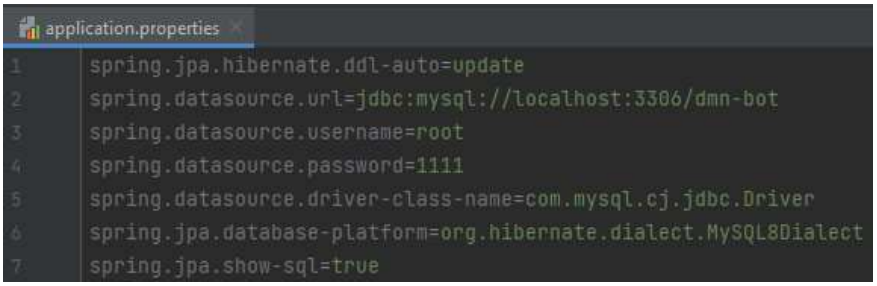

Рис. 3.3. Налаштування параметрів для зв’язку з базою даних

3. Створюється сутності для таблиці (рис. 3.4).

Рис. 3.4. Сутності для таблиці із користувачами

4. Створюється репозиторій для взаємодії із базою даних. Він дозволяє виконуати операції читання, запису, оновлення та видалення даних (рис. 3.5).

```java
import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface UserRepository extends CrudRepository<User,Long> {
    List<User> findAll();
}
```

Рис. 3.5. Реалізація інерфейсу *CrudRepository*

5. Далі використовується репозиторій у сервісах для взаємодії із *MySQL*.

Результати виконання налаштувань та зображено на рисунку 3.6 та рисунку 3.7 після створення, обробки, запису та редагування даних у них [17].

```
1 •  SELECT * FROM `dan-bot`.users_data_table;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

chat_id	place	birth_day	change_data	deliver_done	first_name	last_name	num_product	phone_number	product	registered_at	type_place	type_product	user_name
578967742	street2	NULL	578969212	NULL	NULL	NULL	35	+38096[illegible]6	NULL	2023-06-09 12:52:55.005000	dfsdfsdf	1470	NULL
578969212	street2	0	50	1	Zorro	M	35	+38096[illegible]6	NULL	2023-06-09 12:45:47.766000	dfsdfsdf	1470	Zorro_Ma
578969220	NULL	0	NULL	NULL	Zorro	M	NULL	+38096[illegible]5	dragon_sushi	2023-06-09 12:52:42.635000	NULL	NULL	NULL
578969224	NULL	0	NULL	NULL	Zorro	M	NULL	+38096[illegible]5	chiken_soup	2023-06-09 12:52:41.445000	NULL	NULL	NULL
578969225	NULL	0	NULL	NULL	Zorro	M	NULL	+38096[illegible]5	acaprese_salad	2023-06-09 12:52:40.313000	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 3.5. Таблиця із даними про користувачів

inc	type_establishment	name_establishments	type_product	products	price_for_product	price_for_delivery	location	time_for_deliver	time_for_cook	photo_of_product
1	cooked	Hek	burger	pork-burger	350	60	street1	30	45	BLOB
2	cooked	Solt	burger	mix-burger	450	50	street2	35	50	BLOB
3	cooked	Gas	burger	bekon-burger	400	55	street4	40	40	BLOB
4	cooked	Hek	pizza	dough_pizza	425	60	street1	30	25	BLOB
5	cooked	Solt	pizza	cheese_pizza	400	50	street2	35	20	BLOB
6	cooked	Gas	pizza	pizza_mix	450	55	street4	40	25	BLOB
7	cooked	Hek	sushi	booga_sushi	900	60	street1	30	50	BLOB
8	cooked	Solt	sushi	dragon_sushi	1100	50	street2	35	50	BLOB
9	cooked	Gas	sushi	ogami_sushi	1050	55	street4	40	50	BLOB
10	cooked	Hek	soup	borsh_soup	150	60	street1	30	50	BLOB
11	cooked	Hek	salad	cesar_salad	230	60	street1	30	25	BLOB
12	cooked	Solt	soup	chiken_soup	125	50	street2	35	50	BLOB
13	cooked	Solt	salad	acaprese_salad	195	50	street2	35	30	BLOB
14	cooked	Gas	salad	greece_salad	220	55	street4	40	20	BLOB
15	cooked	Gas	soup	solianka_soup	210	55	street4	40	60	BLOB
16	market	alco_market	alcohol	scotch	650	35	street3	40	2	BLOB
17	market	alco_market	alcohol	wine	540	35	street3	40	2	BLOB
18	market	dairy_market	dairy	milk	60	30	street5	20	1	BLOB
19	market	dairy_market	dairy	butter	100	30	street5	20	1	BLOB
20	pharmacy	help	pills	pills_cold	245	45	street6	15	3	BLOB
21	pharmacy	help	drops	nose_drops	300	45	street6	15	3	BLOB
22	pharmacy	number_1	pills	pills_pain	250	35	street7	25	3	BLOB
23	pharmacy	number_1	pills	pills_strain	325	35	street7	25	3	BLOB
24	pharmacy	number_1	pills	pills_cought	200	35	street7	25	3	BLOB

Рис. 3.6. Таблиця із даними про заклади

3.3. Основний функціонал

Реєстрація користувача

У даному пункті, при вводі початкової команди "*/start*", *Telegram*-бот перевіряє чи вводив користувач цю команду раніше. Якщо ні – пропонує зареєструватися (рис. 3.7), якщо так – відображає інформацію, що користувач був зареєстрований раніше і відображає основне меню сервісу (рис. 3.8).

Рис. 3.7. Частина коду для реєстрації нового користувача

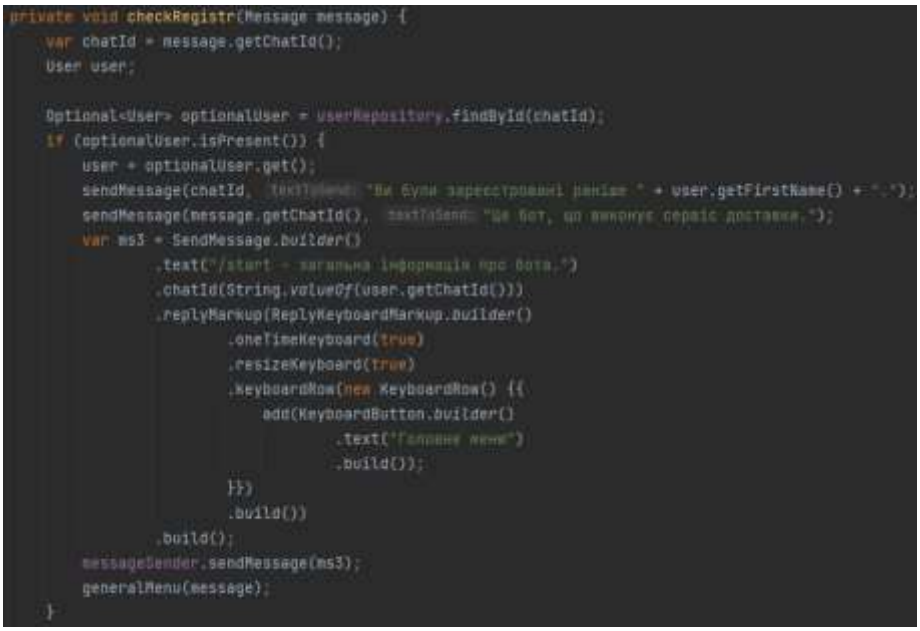

Рис. 3.8. Частина коду для раніше створених користувачів

Оформлення замовлення

Після натискання кнопки “Оформлення замовлення” відображаються доступні заклади, з яких можна виконати замовлення. При переході в бажаний заклад, бот надає всі доступні позиції, які наявні для подальшого придбання. Щоб обрати товар, потрібно натиснути на кнопку “Додати до корзини”, після чого товар буде доданий до “Корзини”. Для подальшого оформлення замовлення, показано перелік товарів, які було обрано в закладі. Потім замовник вводить через клавіатуру місце для доставки, після чого сформується і відобразиться загальна інформація про замовлення. На рисунку 3.9 зображено метод, що виконується після натискання кнопки “Оформлення замовлення”.

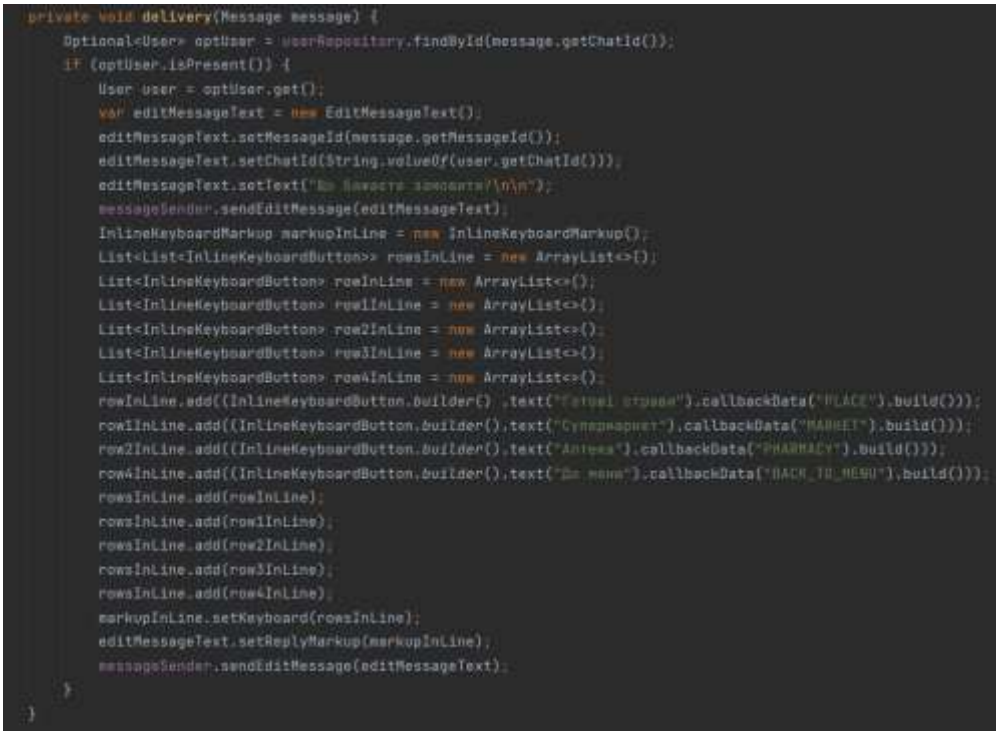

Рис. 3.9. Код для оформлення замовлення

Додатковий функціонал

На даному кроці бот відображає, інформацію про замовлення, яке було виконане раніше (рис. 3.10).

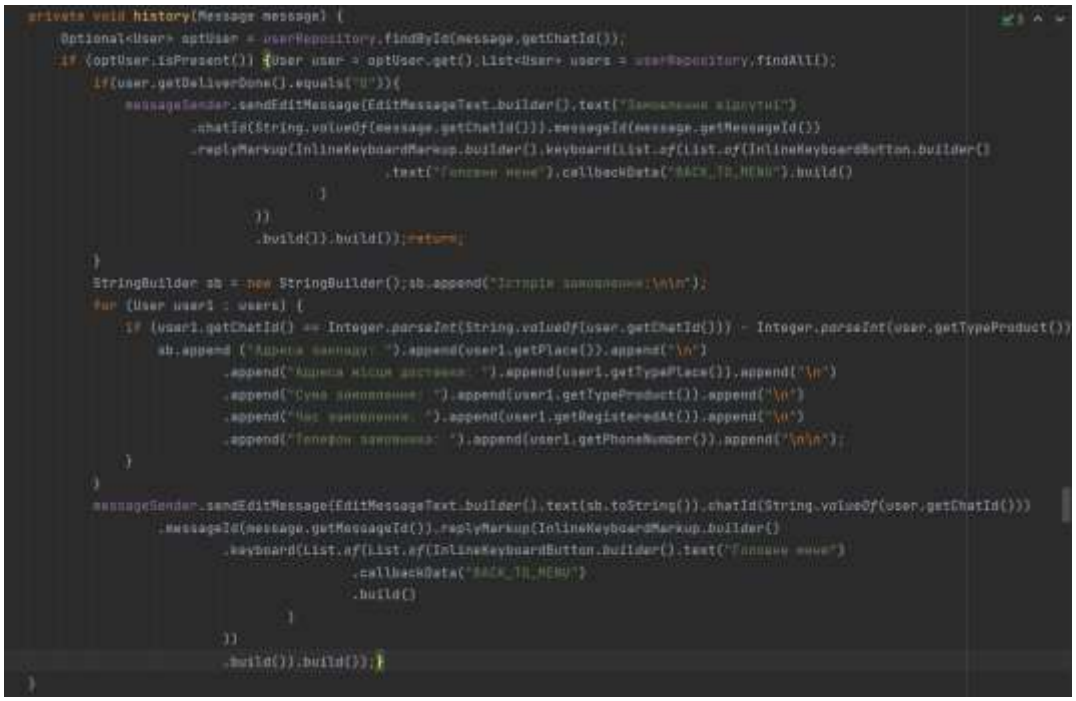

Рис. 3.10. Метод, що виконується при перегляді історії замовлення

Також можна переглянути та змінити персональну інформацію (рис. 3.11).

```
private void profile(Message message) {
    Optional<User> optUser = userRepository.findById(message.getChatId());
    if (optUser.isPresent()) {
        User user = optUser.get();
        var editMessageText = new EditMessageText();
        editMessageText.setChatId(String.valueOf(user.getChatId()));
        editMessageText.setMessageId(message.getMessageId());
        editMessageText.setText("Особистий кабінет користувача\n\n" +
                "Ім'я: " + user.getFirstName() + "\n" +
                "Прізвище: " + user.getLastName() + "\n" +
                "Номер телефону: " + user.getPhoneNumber() + "\n\n");
        messageSender.sendEditMessage(editMessageText);
        askForChangeData(editMessageText);
    }

}
```

Рис. 3.11. Метод, для перегляду та зміни особистих даних

3.4. Результати реалізації мережевого сервісу доставки

Після відкриття чат-боту “*DeliverMeNowBot*”, відображаєть текстовий опис створеного боту. Для початку роботи із сервісом потрібно ввести команду “/start” або натиснути на неї у текстовому описі (рис. 3.12).

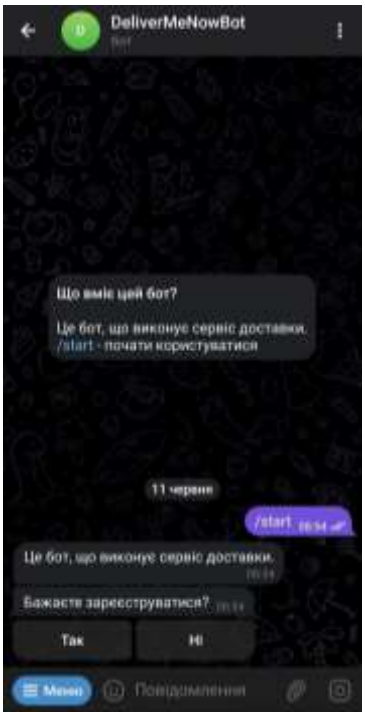

Рис. 3.12. Виконання команди “/start”

Далі чат-бот пропонує зареєструватися, оскільки вхід виконується вперше. Після натискання кнопки “Так”, він надсилає запит, для отримання номера мобільного телефону. Після чого реєструє та відображає основне меню сервісу (рис. 3.13).

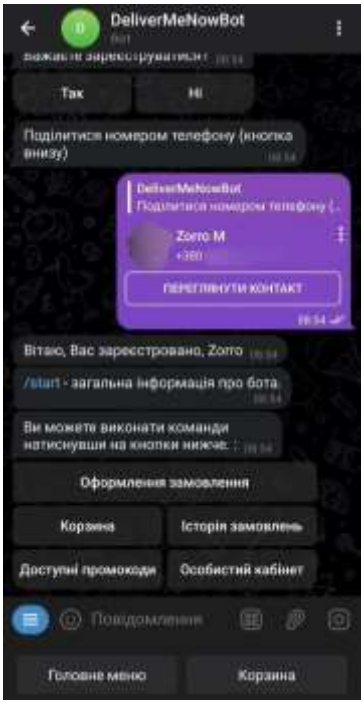

Рис. 3.13. Виконана реєстрація

Натиснувши на “Оформлення замовлення” обирається заклад “Супермаркет”, пісял чого бот відображає всі наявні позиції даного закладу, де вказується назва товару, ціна за продукт, ціна за доставку, час на приготування замовлення та час на доставку. На рисунку 3.14 показано прилад відображення товару закладу “Супермаркет”.

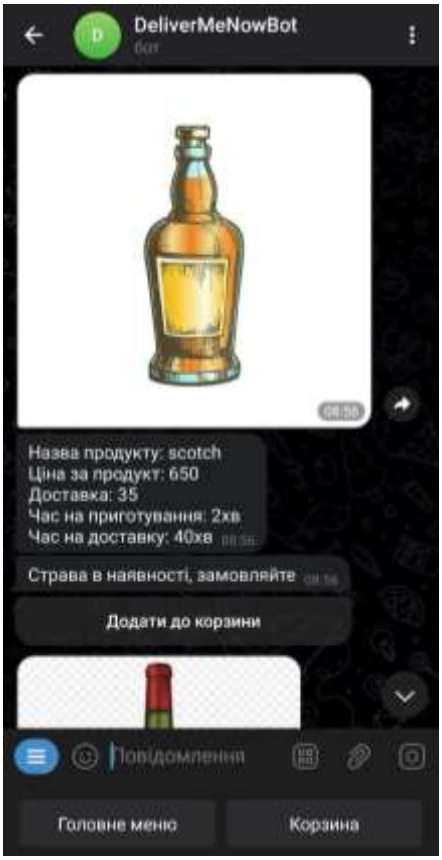

Рис. 3.14. Товар, який наявний в “Супермаркет”

Потім додаються товари за бажанням у корзину, натисненням на кнопку “Додати до корзини”. Далі, перейшовши до корзини, натиснувши кнопку “Корзина” бот надсилає повідомлення із вираними позиціями, вказуючи назву товару, ціну за товар, адресу закладу, ціну за доставку, та загальну вартість замовлення. Для кінцевого оформлення, потрібно вказати адресу для доставки і після цього замовлення буде оформлено. Нижче на рисунку 3.15 зображено повідомлення після оформлення замовлення.

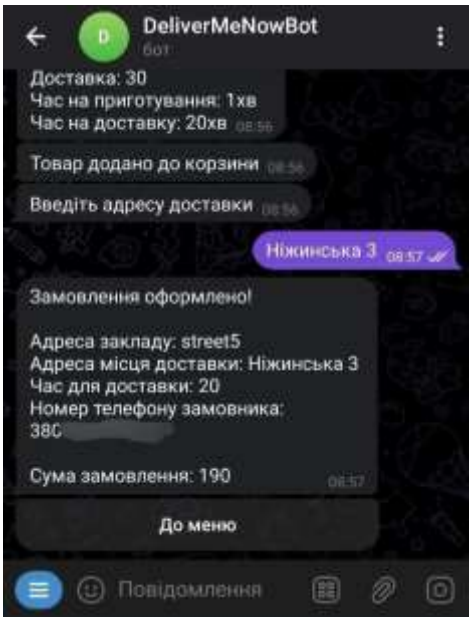

Рис. 3.15. Оформлене замовлення

Щоб подивитися історію замовлення, із головного меню потрібно натиснути на кнопку “Історія замовлення” (рис. 3.16)

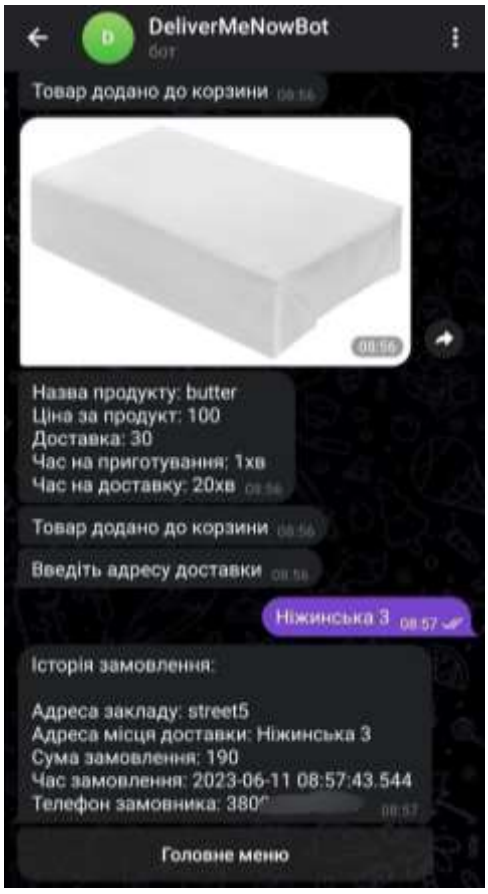

Рис. 3.16. Історія замовлення

Також, у мережевому сервісі є можливість застосування промокоду через головне меню натиснувши на кнопку “Доступні промокоди”.(рис. 3.17).

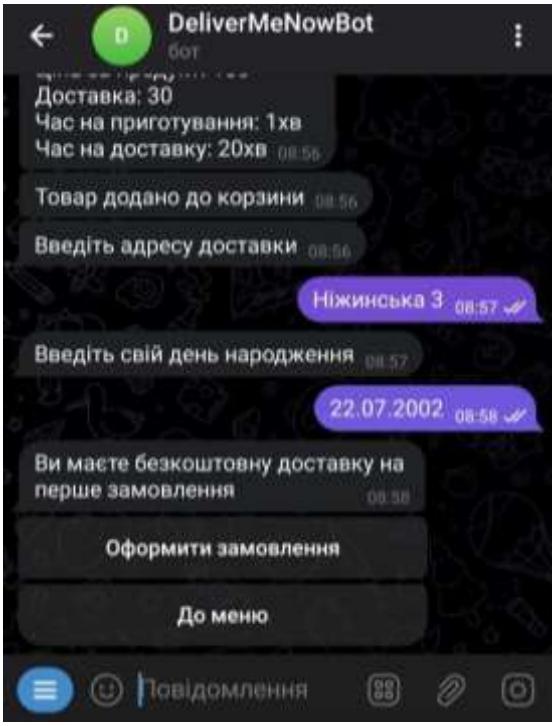

Рис. 3.17. Доступні промокоди

Для перегляду особистої інформації у сервісі доставки потрібно натиснути на кнопку “Особистий кабінет” головного меню (рис. 3.18).

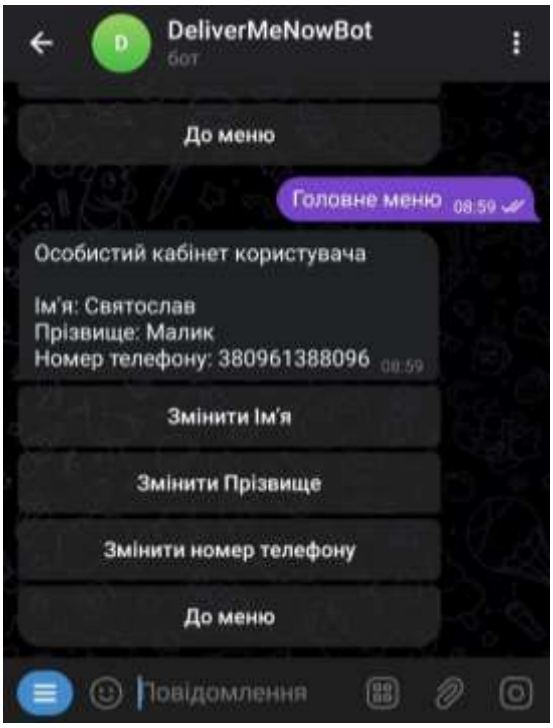

Рис. 3.18. Особистий кабінет

Висновки до розділу

У кінцевому розділі виконується процес розробки мережевого сервісу із урахуваннями особливостей функціоналу. Перед написанням коду, було розглянуто процес отримання токену для взаємодії із *Telegram*.

Крім того, було описано використання бази даних *MySQL* для зберігання інформації про користувачів та заклади, а також процес налаштування з'єднання між *MySQL* і *Spring Boot*. Було обгрунтовано використання фреймворка *Spring Boot* для створення таблиць, репозиторіїв та взаємодію з базою даних.

Далі було описано основний функціонал бота, такий як реєстрація користувача, оформлення замовлення та додаткові функції, наприклад, перегляд та зміна персональної інформації. Для кожного з цих етапів були надані інструкції та зображено результати виконання.

У підсумку, було успішно налаштовано взаємодію з *Telegram API*, збережено необхідну інформацію в базі даних та реалізовано функціонал мережевого сервісу доставки.

ВИСНОВКИ

У даній робот було розглянуто поняття мережевих сервісів та їх складові, такі як комунікаційні протоколи, комунікаційні послуги, системи зберігання даних, сервіси спільної роботи, веб-сервіси, сервіси безпеки та віртуалізація мережі. Також було детально розглянуто поняття чат-боту, його можливості та переваги. Також було проаналізовано доступні засоби для створення мережевого сервісу доставки з врахуванням критеріїв, таких як зручність інтерфейсу, швидкість відгуку, безпека та надійність в обробленні даних про користувача, доступність на різних платформах.

В додаток цьому було описано засоби, що використовуються для створення мережевого сервісу. Були визначені основні можливості та переваги засобів, таких як Java, Intellij IDEA, Spring Boot, BotFather та MySQL. Також розглянуто процес отримання токену для взаємодії з Telegram API, використання бази даних MySQL для зберігання інформації та налаштування з'єднання між MySQL та Spring Boot.

У результаті виконання кваліфікаційної роботи було створено мережевий сервіс доставки на основі Telegram-бот. Він дозволяє замовити товар із обраного закладу, відображає всі наявні товари, продукти, страви, що знаходяться у ньому. Також надає моливість зменшити загальну суму замовлення, використавши знижку. Окрім цього, можна редагувати обрані товари, а саме додавати і віднімати позиції. Сервіс забезпечує високу конфіденційність та безпеку даних користувачів.

Подальшоий розвитку даної роботи орієнтується на розширенні можливостей функціоналу, наданні можливості постійного використання, шляхом перенесення боту на сервер та інтегрування із іншими сервісами та додатками.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Understanding Network Resources and Their Types* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.sciencedirect.com/science/article/pii/B9780128096658000035*.

2. *Network Services and Their Role in Computer Networks* [Електронний ресурс] – Режим доступу до ресурсу: *https://link.springer.com/article/10.1007/s11227-019-02897-9*

3. *Introduction to Network Resources and Their Management* [Електронний ресурс] – Режим доступу до ресурсу: *https://ieeexplore.ieee.org/document/7427852*.

4. *Network Resource Allocation in Cloud Computing Environments* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.sciencedirect.com/science/article/pii/S2352864816300966*.

5. *TCP/IP Protocol Suite: An Overview of Communication Protocols* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.researchgate.net/publication/282116571_Cervical_ranula*.

6. *A Survey on Wireless Communication Protocols for Intelligent Transportation Systems* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.mdpi.com/1424-8220/17/7/1567*.

7. *An Overview of Communication Protocols in Internet of Things for Smart Grid Applications* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.mdpi.com/1996-1073/10/1/37*.

8. *A Survey on Chatbot Design Techniques in Speech Conversation Systems* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.mdpi.com/1999-5903/11/9/201*.

9. *Chatbots in Customer Service: A Systematic Literature Review* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.mdpi.com/2079-9292/9/2/270*.

10. *Chatbot for Education: A Review of Recent Advances and Challenges* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.mdpi.com/2076-3417/10/8/2776*.

11. *Chatbot Technologies in Healthcare: State-of-the-Art and Future Challenges* – Режим доступу до ресурсу: *https://www.mdpi.com/2079-9292/9/18/3808.*

12. *Introduction to Java Programming Language* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.javatpoint.com/java-tutorial*.

13. *IntelliJ IDEA: Getting Started Guide* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.jetbrains.com/idea/guide/.*

14. *Introduction to Spring Boot Framework* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.baeldung.com/spring-boot.*

15. *Creating a Telegram Bot using BotFather and Java* [Електронний ресурс] – Режим доступу до ресурсу: *https://dzone.com/articles/creating-a-telegram-bot-using-java.*

16. *MySQL Tutorial for Beginners*– Режим доступу до ресурсу: *https://www.mysqltutorial.org/introduction-to-mysql-full-text-search.aspx.*

17. *An Introduction to MySQL* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.guru99.com/mysql-tutorial.html.*