

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

студента Силаков Микита Вікторович
(ПІБ)

академічної групи 123-20-2
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему Комп'ютерна система компанії «RokSave» з детальною розробкою функціоналу та сервісів інтерактивного телеграм-боту
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Бешта Д.О.			
спеціальної частини	доц. Бешта Д.О.			
розділів:				
розробка апаратної частини	доц. Ткаченко С.М.			
розробка корпоративної мережі	ас. Бешта Л.В.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2024

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ (підпис) _____ Гнатушенко В.В.
(прізвище, ініціали)
« _____ » _____ 2024 року

ЗАВДАННЯ**на кваліфікаційну роботу ступеня бакалавр**

студента Силаков М.В. академічної групи 123-20-2
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»
(офіційна назва)

на тему: «Комп'ютерна система компанії «RokSave» з детальною розробкою функціоналу та сервісів інтерактивного телеграм-боту» затверджену наказом ректора НТУ «Дніпровська політехніка» від 23.05.2024 р. № 469-с.

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел конкретизується предмет та мету роботи та виконується постанова завдання	26.02.2024
Розробка апаратної частини	На основі аналізу підприємства формулюються технічні вимоги до комп'ютерної системи та розробляється апаратна частина системи	19.03.2024
Розробка корпоративної мережі	Виконується розрахунок налаштувань корпоративної мережі та перевірка роботи системи, розробляються методи та налаштування обладнання для захисту інформації в системі	03.05.2024
Розробка компонента системи	Виконується детальна розробка компонента системи	31.05.2024

Завдання видано _____
(підпис керівника)

_____ Бешта Д.О.
(прізвище, ініціали)

Дата видачі 12.02.2024

Дата подання до екзаменаційної комісії 03.06.2024 р.

Прийнято до виконання _____
(підпис студента)

_____ Силаков М.В.
(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 87 с., 36 рис., 5 табл., 1 дод., 13 джерел.

Об'єкт розробки: комп'ютерна система управління Головне управління Пенсійного фонду України в Дніпропетровській області

Мета: розробка комп'ютерної системи для Головного управління Пенсійного фонду України в Дніпропетровській області. Основний акцент буде зроблено на детальному опрацюванні процесу побудови та налаштування корпоративної мережі, а також реалізації програмного забезпечення на мові програмування Python з використанням бази даних PostgreSQL.

Комп'ютерна система, розроблена в рамках даної роботи, спрямована на оптимізацію та покращення інформаційно-технічного забезпечення Головного управління Пенсійного фонду України в Дніпропетровській області. Основні завдання цієї системи полягають у забезпеченні ефективного збору, збереження та обробки даних, а також у створенні надійної та безперебійної мережевої інфраструктури для забезпечення потреб управління пенсійними справами в регіоні. Реалізація цієї системи спрямована на поліпшення якості обробки та доступу до інформації, що дозволить збільшити ефективність та прозорість роботи управління пенсійними справами та підвищити задоволеність користувачів послугами фонду.

Відповідно до вимог кваліфікаційної роботи бакалавра була проведена розробка комп'ютерної мережі. Схема цієї мережі була втілена у формі логічної топології за допомогою програми Cisco Packet Tracer. Після цього була проведена перевірка спроектованої мережі, результати якої були документовані у вигляді таблиць та графіків. Ці результати були докладно описані та включені до пояснювальної записки або додатків.

Ключові слова: корпоративна мережа, база даних, програмне забезпечення, безпека даних, надійність, інтеграція.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....
Ошибка! Закладка не определена.	
ВСТУП.....	7
1 Стан питання і постановка завдання	8
1.1 Опис об’єкта розробки та умов застосування КС	8
1.2 Характеристика і структура об’єкта впровадження.....	9
1.3 Огляд інженерних рішень	12
1.4 Постановка завдання і мети роботи	14
1.5 Визначення напрямків рішення поставлених завдань	16
2 Розробка апаратної частини комп’ютерної системи підприємства	18
2.1 Вимоги до комп’ютерної системи	18
2.1.1 Вимоги до структури і функціональних можливостей інтерактивного телеграм-боту.....	18
2.1.2 Вимоги до чисельності та кваліфікації персоналу, який обслуговує систему і режим її роботи.....	19
2.1.3 Вимоги до показників призначення	20
2.1.4 Вимоги до захисту інформації від несанкціонованого доступу	21
2.1.5 Вимоги до технічної експлуатації та обслуговування компанії «RokSave»	22
2.2 Вимоги до видів забезпечення.....	23
2.2.1 Вимоги до клієнтської частини.....	23
2.2.2 Вимоги до серверної частини.....	23
2.2.3 Вимоги до програмного забезпечення КС	24
2.3 Розробка апаратної частини комп’ютерної системи	26
2.4 Розрахунок інтенсивності вихідного трафіку найбільшої локальної мережі підприємства.....	30
3 Проектування корпоративної мережі та перевірка роботи комп’ютерної системи підприємства	32
3.1 Розрахунок схеми адресації корпоративної мережі	32
3.2 Розрахунок схеми адресації пристроїв для компанії «RokSave»	34
3.3 Налаштування роботи комп’ютерної системи	37
3.3.1 Базове налаштування конфігурації пристроїв КС	37
3.3.2 Налаштування маршрутизаторів в КС компанії «RokSave»	41
3.3.3 Налаштування служби агрегування каналів LACP.....	46
3.3.4 Налаштування VLAN Головного управління Пенсійного фонду України .	46
3.3.5 Налаштування динамічного NAT	48
3.3.6 Налаштування списків доступу	48
3.3.7 Налаштування аутентифікації службою AAA	49

3.4 Перевірка роботи комп'ютерної системи.....	51
4 Розробка компонента системи	53
4.1 Розробка функціоналу чат-бота замовлення таксі	53
4.2 Розробка чат-боту замовлення таксі	57
4.2.1 Node.js.....	57
4.2.2 Telegram bot API	58
4.2.3 Розміщення чат-боту на віддаленому сервері	59
4.3 Тестування роботи чат-бота замовлення таксі	60
Додаток А. Текст програми налаштувань мережі комп'ютерної	Ошибка! Закладка не определена.
Додаток Б. Текст програми чат-боту таксі	Ошибка! Закладка не определена.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ ТА ТЕРМІНІВ

КС – комп'ютерна система

VLSM - Маскування підмережі змінної довжини

ISAKMP - Internet Security Association and Key Management Protocol, протокол управління ключами

NAT - Трансляція мережевих адрес

VLAN - Virtual Local Area Network, віртуальна локальна мережа

БД– база даних

ВСТУП

Комп'ютерні технології займають важливе місце у нашому повсякденному житті, сприяючи автоматизації рутинних процесів та покращенню ефективності різних видів діяльності. Одним з таких інструментів є чат-боти, які забезпечують швидку та зручну комунікацію між користувачами та системами. Чат-боти стали невід'ємною частиною багатьох сфер, включаючи бізнес, обслуговування клієнтів, навчання та розваги. Вони допомагають забезпечити безперервний зв'язок, оперативно відповідати на запити користувачів та надавати необхідну інформацію.

Метою даної кваліфікаційної роботи є створення інтерактивного Telegram-бота для компанії «RokSave», який буде виконувати функції виклику таксі та управління даними користувачів. Бот має забезпечити зручний інтерфейс для водіїв та пасажирів, автоматизувати процеси реєстрації та замовлення послуг, а також забезпечити безперебійний доступ до інформації.

Важливою частиною роботи є розробка та налаштування комп'ютерної мережі, яка забезпечить стабільну роботу бота та інших сервісів компанії. Впровадження хмарних технологій, таких як платформа Heroku, дозволить забезпечити високу доступність та масштабованість системи, що є ключовим для надання якісних послуг клієнтам.

Актуальність дослідження зумовлена зростаючою популярністю та необхідністю автоматизації бізнес-процесів у сучасних компаніях. Використання чат-ботів дозволяє значно підвищити ефективність обслуговування клієнтів, зменшити навантаження на службу підтримки та забезпечити швидкий доступ до необхідної інформації. Розробка інтерактивного бота для виклику таксі дозволить компанії «RokSave» покращити якість послуг, забезпечити зручність для користувачів та підвищити конкурентоспроможність на ринку.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Опис об'єкта розробки та умов застосування КС

Компанія «RokSave» спеціалізується на наданні послуг перевезень та таксі, що вимагає ефективного управління замовленнями, оперативної комунікації з клієнтами та оптимізації внутрішніх процесів. Розробка комп'ютерної системи з інтерактивним телеграм-ботом є ключовим елементом стратегії компанії, спрямованої на покращення якості обслуговування та підвищення конкурентоспроможності [1].

Об'єктом розробки є комплексна комп'ютерна система, яка складається з декількох компонентів:

- центральна база даних (БД), що зберігає інформацію про замовлення, клієнтів, водіїв та транспортні засоби;
- серверна частина, яка обробляє запити від клієнтів та водіїв, керує взаємодією з базою даних і забезпечує роботу телеграм-боту;
- інтерактивний телеграм-бот, який є інтерфейсом для взаємодії з клієнтами та водіями через мобільний додаток Telegram.

Комп'ютерна система компанії «RokSave» призначена для:

- автоматизації обробки замовлень на перевезення;
- оперативного інформування клієнтів про статус замовлення;
- підтримки ефективною комунікації між клієнтами, водіями та диспетчерами;
- збирання та аналізу даних для покращення бізнес-процесів.

Система повинна забезпечувати:

- високу доступність та надійність: робота системи 24/7 без простоїв;
- безпеку даних: захист персональної інформації клієнтів та водіїв;
- масштабованість: можливість обробки великої кількості запитів з мінімальними затримками;
- інтеграцію з існуючими системами: взаємодію з бухгалтерськими програмами, системами навігації та іншими інструментами.

Основні функції телеграм-боту:

- реєстрація та авторизація користувачів: клієнти та водії можуть зареєструватися та увійти до системи через бот, використовуючи свої мобільні номери та інші ідентифікатори;
- оформлення замовлень: клієнти можуть швидко і зручно оформити замовлення на перевезення, вказавши місце відправлення та призначення, час та особливі вимоги;
- сповіщення про статус замовлення: бот інформує клієнтів про прийняття замовлення, призначення водія, під'їзд автомобіля та завершення поїздки;
- зворотний зв'язок та оцінка послуг: клієнти можуть залишати відгуки про поїздки, оцінювати водіїв та повідомляти про проблеми;
- адміністрування та управління: диспетчери можуть використовувати бот для розподілу замовлень, моніторингу активності водіїв та управління розкладом.

Для забезпечення безпеки комп'ютерної системи буде реалізовано наступні заходи:

- використання шифрування даних при передачі та зберіганні;
- аутентифікація та авторизація користувачів через двофакторну верифікацію;
- регулярне оновлення програмного забезпечення та патчів безпеки;
- моніторинг системи на наявність підозрілої активності та автоматичне виявлення загроз.

1.2 Характеристика і структура об'єкта впровадження

Для ефективного впровадження комп'ютерної системи з інтерактивним телеграм-ботом у компанії «RokSave», важливо зрозуміти її організаційну структуру. Це дозволить краще врахувати потреби кожного підрозділу та забезпечити інтеграцію нової системи з існуючими процесами. Компанія «RokSave» має наступну організаційну структуру: керівництво, операційний відділ, підрозділ ІТ, маркетингу та продажів, обслуговування клієнтів. Розглянемо детально [1].

Компанія «RokSave» знаходиться в Україні, у місті Львів, яке є адміністративним центром Львівської області. Точна адреса компанії: вулиця Антоновича, будинок 5 А. Львів є одним з найбільших та найважливіших міст Західної України, відомим своїм культурним та історичним значенням, а також розвиненою інфраструктурою та сприятливим бізнес-кліматом (рис.1.1).

Розташування компанії у Львові забезпечує доступ до основних транспортних артерій, що сприяє ефективній організації перевезень та наданню послуг таксі. Місто є важливим логістичним центром, що полегшує зв'язок з іншими регіонами України та країнами Європи. Це розташування також надає компанії переваги у доступі до кваліфікованих кадрів та сучасних технологій, що є важливими для успішної діяльності в сфері перевезень.

Антоновича вулиця розташована у зручному місці міста, що забезпечує легкий доступ для клієнтів та співробітників компанії. Будинок 5А, де знаходиться офіс «RokSave», обладнаний усіма необхідними зручностями для ефективної роботи компанії, включаючи сучасні офісні приміщення, засоби комунікації та безпеку.



Рисунок 1.1 – Географічне розміщення організації [1]

Керівництво компанії складається з:

– генеральний директор, який відповідає за загальне управління компанією, прийняття стратегічних рішень та контроль за виконанням планів розвитку. Генеральний директор також здійснює нагляд за впровадженням нових

технологій і забезпечує підтримку інноваційних проєктів, таких як розробка телеграм-боту;

– фінансовий директор, який відповідає за фінансове планування, бюджетування та контроль витрат. Забезпечує фінансування проєкту впровадження нової комп'ютерної системи.

Операційний підрозділ складається з:

– операційний директор - керує щоденною діяльністю компанії, включаючи організацію роботи водіїв та диспетчерів. Забезпечує впровадження ефективних бізнес-процесів та оптимізацію операцій;

– диспетчерська служба - відповідає за прийом замовлень від клієнтів, розподіл завдань між водіями, моніторинг виконання замовлень та надання оперативної інформації клієнтам. Диспетчери будуть активно взаємодіяти з новим телеграм-ботом для покращення процесу обробки замовлень;

– водії виконують перевезення клієнтів за замовленнями, отриманими через диспетчерську службу та телеграм-бот. Водії також надають зворотний зв'язок щодо якості системи та процесів.

Підрозділ інформаційних технологій складається з:

– ІТ-директора, який відповідає за стратегічне управління ІТ-ресурсами компанії, включаючи розробку та впровадження нових технологій. Забезпечує технічну підтримку та інфраструктуру для впровадження нової системи;

– команда розробників - займається створенням та підтримкою програмного забезпечення, включаючи розробку телеграм-боту, інтеграцію з базою даних та іншими системами компанії. Команда розробників забезпечує тестування та запуск нової системи;

– системні адміністратори відповідають за підтримку серверів, баз даних та мережевої інфраструктури. Забезпечують надійну та безперебійну роботу комп'ютерної системи та захист даних.

Підрозділ маркетингу та продажів складається з:

– маркетинговий директор відповідає за розробку та реалізацію маркетингових стратегій, просування нових послуг та взаємодію з клієнтами.

Маркетинговий відділ також займається аналізом даних, отриманих через телеграм-бот, для покращення послуг компанії;

- команда продажів працює над залученням нових клієнтів та утриманням існуючих, використовуючи дані та інструменти, надані новою системою.

Підрозділ обслуговування клієнтів складається з:

- керівник служби підтримки клієнтів відповідає за забезпечення високого рівня обслуговування клієнтів, вирішення проблем та отримання зворотного зв'язку. Служба підтримки активно використовуватиме телеграм-бот для оперативної комунікації з клієнтами та водіями;

- Оператори служби підтримки надають консультації клієнтам, обробляють звернення та вирішують проблемні ситуації. Забезпечують підтримку клієнтів через різні канали зв'язку, включаючи телеграм-бот.

1.3 Огляд інженерних рішень

Для забезпечення ефективної роботи компанії «RokSave», яка спеціалізується на перевезеннях та наданні послуг таксі, необхідно впровадити комплексну комп'ютерну систему. Ця система повинна включати в себе різноманітні інженерні рішення, спрямовані на автоматизацію бізнес-процесів, покращення якості обслуговування клієнтів, а також підвищення загальної ефективності та безпеки. У цьому розділі розглянемо основні інженерні рішення, які можуть бути застосовані у комп'ютерній системі компанії «RokSave».

Використання хмарних сервісів. Інфраструктура як послуга (IaaS) забезпечує гнучкість та масштабованість обчислювальних ресурсів, дозволяючи компанії легко адаптуватися до змін у попиті на послуги. Платформа як послуга (PaaS) дозволяє розробникам швидко створювати, тестувати та впроваджувати нові функції, знижуючи витрати на управління інфраструктурою. Програмне забезпечення як послуга (SaaS) забезпечує доступ до різноманітних програм, таких як системи управління замовленнями та CRM, через інтернет без необхідності їх локальної установки.

База даних. Реляційні бази даних (SQL) використовуються для зберігання структурованих даних про клієнтів, водіїв та замовлення. Рішення на базі MySQL або PostgreSQL забезпечують надійність і масштабованість. Нереляційні бази даних (NoSQL) використовуються для зберігання великих обсягів неструктурованих даних, таких як журнали активності або відгуки клієнтів. MongoDB або Cassandra можуть бути використані для цієї мети.

Інтерактивний телеграм-бот. Telegram Bot API використовується для створення та інтеграції телеграм-боту з основною системою. Забезпечує всі необхідні функції для взаємодії з користувачами. Вебхуки та довготривалі опитування: Використовуються для отримання повідомлень та обробки запитів в реальному часі, що забезпечує миттєву реакцію на запити клієнтів.

Розпізнавання природної мови (NLP) – інтеграція з сервісами NLP, такими як Dialogflow або Wit.ai, для покращення взаємодії бот-користувач та забезпечення більш природної комунікації.

Безпека. Шифрування даних використання SSL/TLS для шифрування даних при передачі між клієнтами, серверами та ботом. Аутентифікація та авторизація - впровадження методів двофакторної автентифікації (2FA) для підвищення рівня безпеки доступу до системи. Захист від DDoS атак - використання хмарних сервісів захисту, таких як Cloudflare, для запобігання та мінімізації впливу DDoS атак.

Аналітика та звітність. Системи бізнес-аналітики (BI): інструменти, такі як Power BI або Tableau, для аналізу даних та створення візуалізацій, що допомагають керівництву приймати обґрунтовані рішення.

Машинне навчання та штучний інтелект використання моделей машинного навчання для прогнозування попиту, оптимізації маршрутів та покращення обслуговування клієнтів.

Інтеграція та масштабованість. API-шлюзи - використання API Gateway для управління трафіком, автентифікації та моніторингу API запитів. Мікросервісна архітектура - розробка системи у вигляді набору незалежних сервісів, що взаємодіють через API, забезпечує гнучкість та можливість масштабування.

Системи контейнеризації та оркестрації. Docker - використання контейнерів для ізоляції додатків та забезпечення їх портативності. Kubernetes - оркестрація контейнерів для автоматизації розгортання, управління та масштабування додатків.

1.4 Постановка завдання і мети роботи

У сучасному конкурентному середовищі компанії, що надають послуги перевезення та таксі, постійно шукають нові способи підвищення ефективності своїх операцій та покращення якості обслуговування клієнтів. Компанія «RokSave» не є винятком. З метою вдосконалення процесів обслуговування клієнтів, автоматизації управління замовленнями та підвищення загальної ефективності роботи, виникла потреба у розробці нової комп'ютерної системи. Основні завдання, які ставляться перед цією системою, включають:

- розробка та впровадження топологічної схеми для компанії «RokSave» яка представлена на рис.1.2;
- забезпечення швидкої та надійної обробки замовлень на перевезення, зменшення часу від прийому замовлення до його виконання;
- створення зручного і доступного інструменту для комунікації з клієнтами, який дозволить швидко отримувати інформацію про послуги, робити замовлення та отримувати статус їх виконання;
- надання інструментів для ефективного управління розкладом водіїв, моніторингу їх роботи та оптимізації маршрутів;
- захист персональної інформації клієнтів та конфіденційних даних компанії від несанкціонованого доступу та втрат;
- забезпечення безперебійної роботи нової системи разом з іншими інформаційними системами, що вже використовуються у компанії.

Метою даної роботи є розробка та впровадження комп'ютерної системи для компанії «RokSave», що включає інтерактивний телеграм-бот, який дозволить автоматизувати процеси обробки замовлень, покращити взаємодію з клієнтами,

оптимізувати роботу водіїв та диспетчерів, а також забезпечити високий рівень безпеки даних.

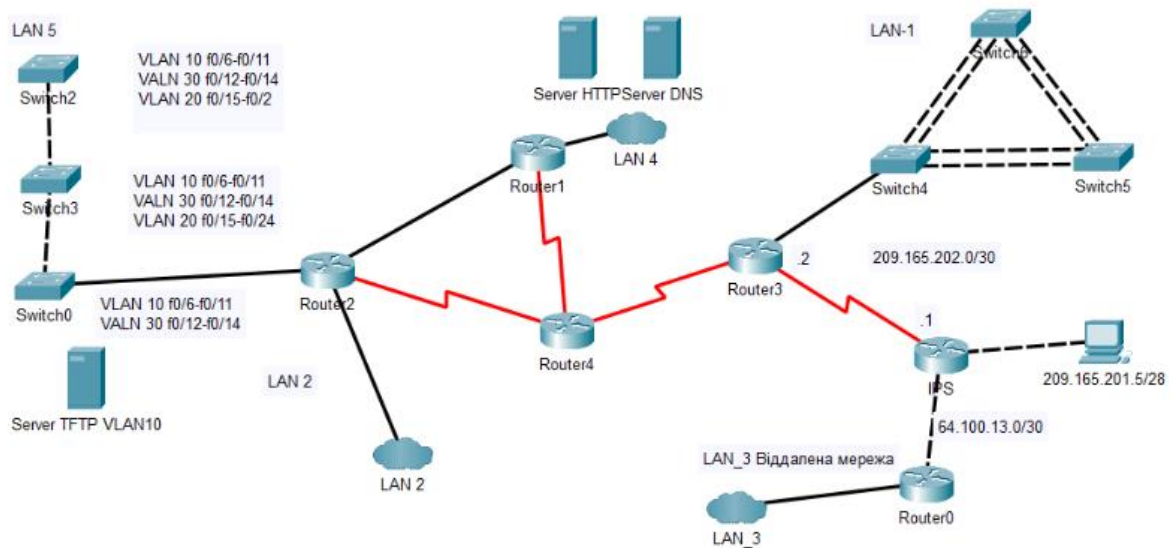


Рисунок 1.2 – Топологічна схема для компанії «RokSave»

Досягнення цієї мети передбачає вирішення наступних завдань:

- дослідити поточні процеси компанії «RokSave», визначити вимоги та особливості роботи у сфері перевезень та таксі;
- визначити необхідні функціональні та нефункціональні вимоги до нової системи, враховуючи потреби всіх зацікавлених сторін;
- вибрати оптимальну архітектуру для забезпечення надійної роботи телеграм-боту, інтеграції з основною системою та масштабованості;
- визначити необхідні апаратні та програмні засоби для забезпечення стабільної та ефективної роботи комп'ютерної системи;
- дослідити поточний стан мережевого трафіку компанії, визначити потенційні вузькі місця та забезпечити оптимізацію трафіку для роботи нової системи;
- створити детальну модель комп'ютерної системи, включаючи телеграм-бот, серверну частину та базу даних, та налаштувати відповідне обладнання;
- впровадити рішення для забезпечення безпеки даних та захисту інфраструктури компанії;

– перевірити роботу комп'ютерної системи в цілому та окремих її компонентів, включаючи телеграм-бот, серверну частину, базу даних та мережеве обладнання, для забезпечення їхньої надійності та відповідності технічним вимогам.

1.5 Визначення напрямків рішення поставлених завдань

Зважаючи на потреби та характеристики компанії «RokSave», яка спеціалізується на перевезеннях та наданні послуг таксі, а також на вимоги щодо побудови та налаштування інтерактивного телеграм-боту і комп'ютерної системи, необхідно ретельно розглянути можливі напрямки розв'язання цих завдань.

Першим кроком може бути аналіз об'єкта. Це включає дослідження поточних процесів компанії, визначення вимог та особливостей роботи у сфері перевезень та таксі. Для цього можна використовувати документальний аналіз, інтерв'ю та опитування співробітників і клієнтів, а також безпосереднє спостереження за роботою компанії.

Після аналізу об'єкта необхідно сформулювати технічні вимоги для розробки комп'ютерної системи. Це включає визначення необхідних функціональних та нефункціональних вимог до системи, враховуючи потреби всіх зацікавлених сторін, таких як керівництво, диспетчери, водії та клієнти.

Наступним кроком є обґрунтування вибору архітектури для інтерактивного телеграм-боту. Можливі варіанти включають монолітну архітектуру, мікросервісну архітектуру та серверлесс архітектуру. Вибір оптимальної архітектури повинен враховувати надійність, масштабованість та інтеграцію з основною системою компанії.

Після вибору архітектури необхідно розробити специфікацію апаратних та програмних засобів. Це включає визначення необхідних апаратних та програмних компонентів для забезпечення стабільної та ефективної роботи комп'ютерної системи.

Для забезпечення надійності та оптимізації роботи мережі компанії слід провести аналіз мережевого трафіку. Використовуючи інструменти моніторингу та

аналізу мережевого трафіку, можна визначити потенційні вузькі місця та забезпечити оптимізацію трафіку для роботи нової системи.

Далі необхідно розробити модель комп'ютерної системи та виконати конфігурування мережевого обладнання. Це включає створення детальної моделі системи, налаштування серверів, бази даних та інтерактивного телеграм-боту, а також налаштування мережевого обладнання.

Для забезпечення безпеки необхідно реалізувати інтеграцію IoT-систем для безпеки. Це включає вибір IoT-платформи, розробку та впровадження IoT-пристроїв для моніторингу та забезпечення безпеки, а також інтеграцію цих пристроїв з основною системою компанії.

Після впровадження всіх компонентів необхідно провести тестування системи. Це включає функціональне тестування всіх функцій системи, стрес-тестування для визначення стійкості системи до навантажень та тестування безпеки для виявлення вразливостей та забезпечення захисту даних.

Нарешті, після успішного тестування можна переходити до налагодження системи та підготовки до її впровадження. Це включає навчання персоналу компанії, забезпечення технічної підтримки та реагування на зворотний зв'язок від користувачів.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДПРИЄМСТВА

2.1 Вимоги до комп'ютерної системи

2.1.1 Вимоги до структури і функціональних можливостей інтерактивного телеграм-боту

Основним призначенням КС є забезпечення автоматизації процесу замовлення і перевезення, а також удосконалення взаємодії з клієнтами через інтерактивний телеграм-бот. КС повинна забезпечувати швидке та ефективне оброблення запитів, зберігання даних про замовлення та клієнтів, а також покращення якості послуг на основі зібраної інформації.

Об'єктом розробки є комп'ютерна система, яке включає інтерактивний телеграм-бот, систему обробки даних, базу даних для зберігання інформації, а також механізми для авторизації, аналізу роботи і забезпечення безпеки даних.

Комп'ютерна система компанії «RokSave» з інтерактивним телеграм-ботом має надавати наступний функціонал [2]:

- збір інформації. Користувачі мають можливість через телеграм-бот робити замовлення на перевезення, передавати інформацію про місце призначення, вибір типу транспорту та додаткові побажання;
- обробка інформації. ПЗ повинне обробляти інформацію, яку передає користувач через телеграм-бот. це включає валідацію введених даних, розрахунок оптимального маршруту, визначення вартості послуги та підтвердження замовлення;
- зберігання даних. Зберігання інформації про замовлення, маршрути, дані клієнтів та водіїв. використання бази даних для зберігання історії замовлень та оброблених даних;
- авторизація змін. Кожна дія користувача в системі повинна бути підтверджена авторизацією. це включає вхід до системи, підтвердження замовлення та зміни у профілі користувача;
- самостійний аналіз роботи. Система повинна аналізувати виконані замовлення, маршрути та відгуки клієнтів для покращення якості послуг.

використання алгоритмів машинного навчання для оптимізації маршрутів та підвищення ефективності роботи;

- безперервна робота. Сервіс повинен бути доступний користувачам у будь-якому місці та у будь-який час, забезпечуючи безперебійну роботу 24/7;
- інформаційна підтримка. Телеграм-бот повинен надавати користувачам актуальну інформацію про доступні послуги, статус замовлення, а також можливість зв'язку з оператором для отримання додаткової допомоги;
- безпека даних. Забезпечення високого рівня безпеки при обробці та зберіганні персональних даних користувачів, захист від несанкціонованого доступу та витоку інформації;
- гнучкість налаштувань. Користувачі повинні мати можливість налаштовувати свій профіль, вибирати пріоритетні способи комунікації та оплати, а також отримувати персоналізовані пропозиції на основі їхньої історії замовлень.

2.1.2 Вимоги до чисельності та кваліфікації персоналу, який обслуговує систему і режим її роботи

Компанія «RokSave» має спеціалізований відділ технічної підтримки, який відповідає за безперебійну роботу інтерактивного телеграм-боту та комп'ютерної системи в цілому. Цей відділ забезпечує найвищий рівень технічної підтримки, зокрема в разі виникнення нештатних ситуацій або проблем з мережевим обладнанням. Співробітники відділу мають кваліфікацію, необхідну для швидкого аналізу та вирішення проблем та негайного відновлення роботи системи.

Оскільки компанія «RokSave» має офісні приміщення на різних поверхах будівлі, ефективне обслуговування користувачів вимагає належної організації робочого часу та достатньої кількості працівників в технічній підтримці. Тому, для забезпечення нормованого робочого процесу і запобігання перевтоми, було призначено 6 співробітників на цей відділ. Робочий графік передбачає роботу з понеділка до суботи з 9:00 до 21:00, з одним вихідним у неділю, що забезпечує покриття усіх годин роботи та можливість оперативного реагування на потреби користувачів та компанії.

Компанія «RokSave» забезпечує свою діяльність через спеціалізований відділ технічної підтримки з чисельністю 20 співробітників. Цей відділ здійснює безперебійну роботу інтерактивного телеграм-боту та комп'ютерної системи в цілому, зокрема в разі виникнення нештатних ситуацій або проблем з мережевим обладнанням.

Окрім того, для забезпечення ефективності обслуговування водіїв компанії, що складає 50 осіб, використовуються спеціальні інтерфейси із системою прийому замовлень, моніторингу маршрутів та взаємодії з клієнтами. Це дозволяє компанії оптимізувати процеси надання послуг та забезпечити оперативне виконання завдань водіями.

2.1.3 Вимоги до показників призначення

Інтерактивний телеграм-бот повинен бути доступним для користувачів цілодобово, без відмов у сервісі, щоб забезпечити неперервну взаємодію з клієнтами у будь-який час доби. Це включає здатність до автоматичного відновлення після можливих збоїв та забезпечення стабільної роботи під час високих навантажень.

Система повинна мати можливість обробляти не менше ніж 100 запитів в секунду на читання та відповідь на повідомлення від користувачів. Це важливо для оперативної взаємодії з клієнтами і швидкого надання відповідей на їх запити.

Компанія «RokSave» повинна мати можливість масштабувати систему в разі потреби. Це означає, що інфраструктура бота має бути гнучкою та легко розширюватися для відповіді на зростаючі потреби в обсягах взаємодії з користувачами та обробки даних.

Бот повинен працювати в хмарному середовищі для забезпечення високої доступності та масштабованості. Використання платформи, такої як Amazon Web Services (AWS) або Google Cloud Platform, дозволяє ефективно керувати навантаженням та автоматично масштабувати ресурси.

Для обробки запитів від користувачів і управління бізнес-логікою бота необхідно розробити потужну бекенд-систему. Вона повинна бути побудована на

мові програмування з високою продуктивністю, наприклад, Python чи Node.js, щоб забезпечити швидке виконання запитів.

Важливо забезпечити високий рівень надійності і захисту конфіденційності даних користувачів. Система повинна мати вбудовані заходи безпеки для захисту від несанкціонованого доступу та збереження цілісності інформації.

2.1.4 Вимоги до захисту інформації від несанкціонованого доступу

Всі користувачі телеграм-боту повинні проходити процедуру автентифікації для підтвердження своєї ідентичності перед отриманням доступу до системи. Це може включати використання двофакторної аутентифікації (2FA) або інших методів валідації.

Всі дані, що передаються між користувачем і ботом, повинні бути зашифровані для захисту від перехоплення третіми особами. Використання сучасних алгоритмів шифрування (наприклад, AES-256) дозволяє забезпечити конфіденційність інформації під час трансляції через мережу.

Важливо забезпечити безпеку фізичного доступу до серверних приміщень та обмежити доступ до баз даних лише авторизованим персоналом за допомогою строгих прав доступу і ролей користувачів.

Система повинна мати вбудовані засоби моніторингу активності і аудиту, щоб вчасно виявляти підозрілі дії і несанкціоновані спроби доступу. Це дозволяє оперативно реагувати на потенційні загрози безпеці інформації.

Компанія повинна розробити та впровадити політику захисту конфіденційної інформації, яка включає в себе навчання персоналу з питань кібербезпеки, контроль за використанням USB-пристроїв та інших засобів зберігання даних, а також заходи проти соціальної інженерії.

Використання реляційних або нереляційних баз даних з механізмами захисту, такими як доступ через SSL (Secure Sockets Layer) або TLS (Transport Layer Security), аудит доступу до даних і захищений зберігання паролів з використанням солей і хешування.

Регулярне створення резервних копій даних і процесів в разі втрати даних або інциденту безпеки для швидкого відновлення роботи системи.

2.1.5 Вимоги до технічної експлуатації та обслуговування компанії «RokSave»

На компанію «RokSave» покладаються високі вимоги щодо експлуатації, технічного обслуговування, ремонту і збереження інфраструктури, необхідної для операцій інтерактивного телеграм-боту. Основними аспектами є забезпечення безперебійної роботи мережевого обладнання та програмного забезпечення, що здійснюється завдяки системному підходу до технічної підтримки і регулярного моніторингу стану обладнання.

Ключовим елементом є персональні комп'ютери, маршрутизатори, комутатори і сервери, які працюють в режимі 24/7. Щоденна перевірка функціональності та стану обладнання дозволяє вчасно виявляти і усувати можливі несправності. Щомісячне обслуговування включає заходи з профілактики для забезпечення максимальної ефективності роботи [3].

Робочий час технічної підтримки обмежено восьмома годинами на добу, що включає в себе систематичне перевірвання стану мережевого обладнання і оперативне реагування на будь-які технічні проблеми. Планові обслуговування виконуються ізольовано для кожного типу обладнання, щоб забезпечити його найкращу продуктивність.

На складі знаходиться запасне мережеве обладнання, яке включає три жорстких диска, два комутатори, один маршрутизатор, одне джерело безперебійного живлення (БП) та десять дисків з ліцензійним програмним забезпеченням. Все запасне обладнання зберігається в упаковках, що запобігають пилу та іншим факторам, що можуть вплинути на їх ефективність.

Ці міри забезпечують, що компанія «RokSave» може мінімізувати можливість виробничих зупинок та зберігати оптимальну робочу продуктивність свого інтерактивного телеграм-боту, гарантуючи безперебійну та ефективну роботу на довгостроковій основі.

2.2 Вимоги до видів забезпечення

2.2.1 Вимоги до клієнтської частини

Клієнтська частина боту повинна мати інтуїтивно зрозумілий і привабливий інтерфейс, який легко навчити і використовувати навіть новачкам. Взаємодія з ботом має бути простою та зрозумілою, з можливістю швидкої навігації і доступу до основних функцій:

- обробка запитів. Бот повинен ефективно обробляти різноманітні запити користувачів, включаючи запити на інформацію, оновлення статусу послуг або замовлень, а також взаємодію з іншими інтегрованими сервісами;

- персоналізація. Можливість налаштовувати індивідуальні налаштування та отримувати персоналізовані рекомендації і повідомлення відповідно до історії взаємодії користувача з ботом;

- мультимовність. Підтримка різних мовних версій для забезпечення доступності користувачів з різних регіонів і культур;

- захист особистих даних. Забезпечення високого рівня захисту особистих даних користувачів, включаючи зашифрування і конфіденційність інформації.

- автентифікація і авторизація. Використання механізмів автентифікації для підтвердження ідентифікації користувача і авторизації доступу до конфіденційної інформації.

- швидкодія. Забезпечення миттєвої відповіді на запити користувачів і мінімізація часу очікування відповіді від бота.

- стабільність. Інтерактивний телеграм-бот повинен бути стабільним і працездатним без перебоїв часу на будь-який час доби;

- масштабованість. Здатність до масштабування для підтримки зростаючої кількості користувачів і розширення функціональності за потреби компанії «RokSave».

2.2.2 Вимоги до серверної частини

Технічні характеристики серверної частини.

Обчислювальна потужність:

- процесор мінімум Intel Xeon або еквівалентний з можливістю гіпертрейдингу;
- оперативна пам'ять мінімум 16 ГБ DDR4 ECC RAM;
- вільний об'єм на жорсткому диску не менше 1 ТБ SSD для операційної системи та додаткового програмного забезпечення;
- підтримка Gigabit Ethernet для швидкого обміну даними з користувачами і базою даних;
- пропускна здатність мінімум 1 Гбіт/с для забезпечення швидкого доступу користувачів до сервісів бота;
- система управління базами даних (СУБД) PostgreSQL або еквівалент з підтримкою ACID;
- простір для зберігання даних мінімум 5 ТБ на внутрішньому масиві SSD або відмінної рівності;
- використання TLS/SSL для захисту передачі даних між клієнтами і сервером;
- щоденні резервні копії баз даних і системних файлів на віддалених серверах або хмарних платформах;
- використання системи моніторингу, наприклад, Prometheus з Grafana для реального часу та збору метрик;
- доступ до консолі адміністратора через SSH з двофакторною аутентифікацією для забезпечення безпеки.

2.2.3 Вимоги до програмного забезпечення КС

Для розробки програм на мові Java необхідно мати встановлені IntelliJ IDEA або Eclipse. IntelliJ IDEA від компанії JetBrains є потужним інтегрованим середовищем розробки, яке підтримує не тільки Java, але і інші мови програмування, такі як Kotlin, Scala, Groovy, JavaScript та інші. Eclipse також забезпечує широкі можливості для розробки Java-програм і є популярним серед розробників.

Для розробки програмного забезпечення на Python необхідно встановити і налаштувати PyCharm. PyCharm є інтегрованим середовищем розробки від JetBrains, спеціально призначеним для роботи з Python. Воно підтримує всі основні функції для

ефективної розробки Python-проектів, що робить його ідеальним вибором для професійних розробників.

Для розробки програм на мові C++ використовуються Code::Blocks і MS Visual Studio. Code::Blocks є відкритим інтегрованим середовищем розробки, яке підтримує мови програмування, такі як C, C++ і Fortran. MS Visual Studio, розроблене компанією Microsoft, також надає розширені можливості для розробки різноманітних програм, включаючи десктопні застосунки, веб-сайти і мобільні додатки на платформі Windows [3].

Для управління реляційними базами даних використовуються MS SQL Server і MySQL Server. MS SQL Server, розроблений компанією Microsoft, і MySQL Server, відкрита реляційна система управління базами даних від Oracle Corporation, є одними з найпоширеніших систем для зберігання, керування і обробки даних у корпоративному середовищі.

Для зручної роботи з різними системами управління базами даних використовується DBeaver. DBeaver є універсальним клієнтом баз даних, який підтримує багато різних СУБД, таких як MySQL, PostgreSQL, Oracle, SQL Server, SQLite та інші, і дозволяє здійснювати всі необхідні операції з управління базами даних з одного інтерфейсу.

Для розробки програм на мові C# використовується MS Visual Studio, яке забезпечує інструменти для створення різноманітних програм, включаючи десктопні додатки, веб-сайти, мобільні додатки і хмарні рішення.

Для проектування і налаштування комп'ютерних мереж використовується Cisco Packet Tracer. Цей інструмент від компанії Cisco дозволяє моделювати, конфігурувати і відлагоджувати мережеві схеми, що є важливим для навчання та розробки мережевих технологій.

Для створення, редагування та обробки документів, презентацій, електронних таблиць та іншої електронної документації використовується Microsoft Office. Цей набір програмного забезпечення від компанії Microsoft забезпечує надійні інструменти для офісних завдань.

Для доступу до веб-ресурсів і забезпечення продуктивного веб-браузингу використовується Google Chrome. Цей браузер від компанії Google є одним з найпопулярніших на світі і надає широкі можливості для роботи з веб-додатками та інтернет-ресурсами. Обмін даними між компонентами системи повинен здійснюватися через єдиний інформаційний простір, що забезпечує єдність даних та їх доступність для всіх відповідних компонентів.

2.3 Розробка апаратної частини комп'ютерної системи

2.3.1 Розробка загальної структури КС компанії «RokSave»

Структурна схема системи компанії «RokSave» представлена на рис. 2.1, яка складається з п'яти під мереж [6].

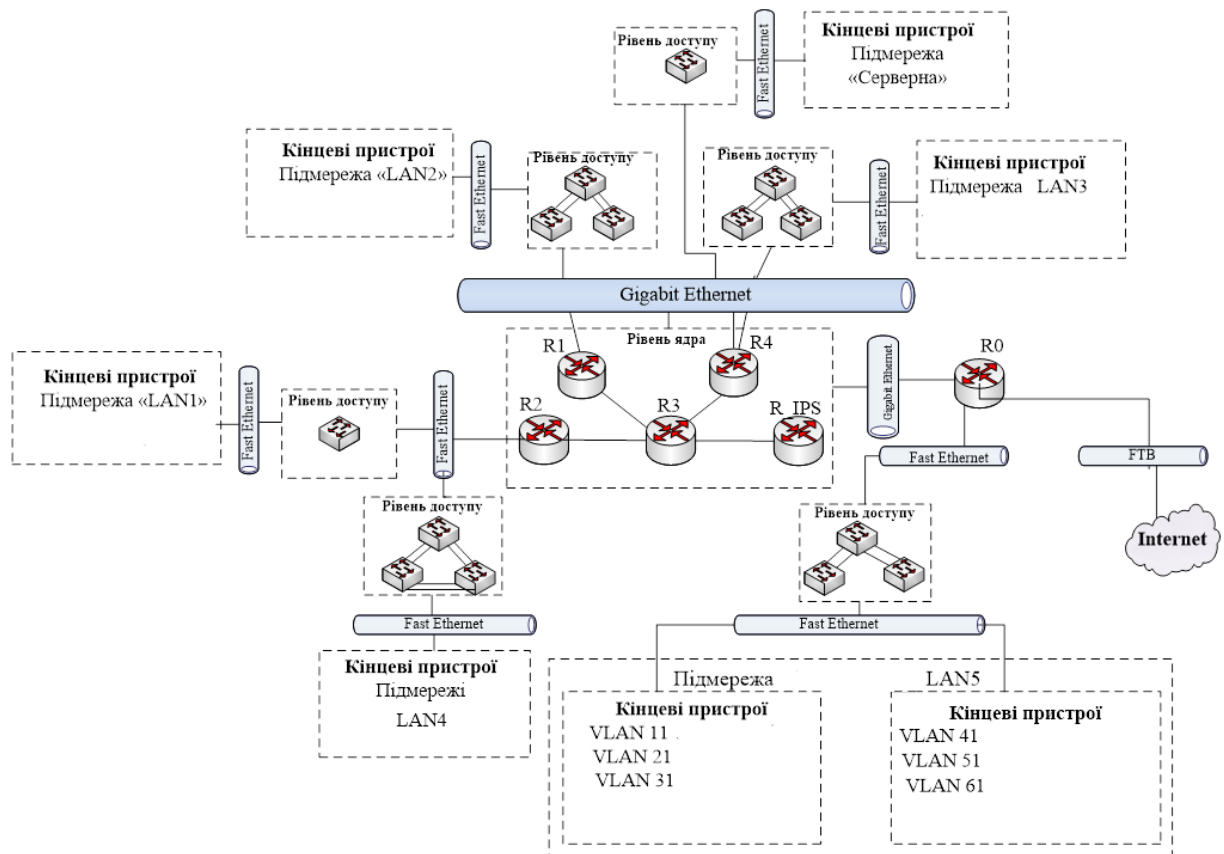


Рисунок 2.1 – Структурна схема компанії «RokSave»

LAN1 Відділ розробки програмного забезпечення. LAN1 призначений для відділу розробки програмного забезпечення. Включає робочі станції розробників,

сервери для розробки та тестування програмного забезпечення, а також обладнання для версійного контролю і спільної роботи над проектами.

LAN2 Відділ системного адміністрування і IT-підтримки. LAN2 призначений для відділу системного адміністрування і IT-підтримки. Включає сервери для управління мережею, бази даних інфраструктури, системи моніторингу і резервне копіювання даних, а також робочі місця системних адміністраторів і технічної підтримки.

LAN3 Відділ фінансів і бухгалтерії. LAN3 призначений для відділу фінансів і бухгалтерії. Включає робочі станції бухгалтерів, програмне забезпечення для фінансового обліку і звітності, сервери для зберігання фінансових даних і баз даних клієнтів.

LAN4 Відділ маркетингу і реклами. LAN4 призначений для відділу маркетингу і реклами. Включає робочі станції маркетологів, веб-сервери для розміщення корпоративного веб-сайту і онлайн-реклами, аналітичні системи для вимірювання ефективності рекламних кампаній.

LAN5 Відділ людських ресурсів. LAN5 призначений для відділу людських ресурсів. Включає робочі станції HR-менеджерів, системи для управління персоналом, бази даних з інформацією про співробітників і кандидатів на роботу.

LAN6 Відділ управління. LAN6 призначений для відділу управління компанією. Включає робочі станції топ-менеджменту, системи для прийняття управлінських рішень, веб-портали для звітності і стратегічного планування.

Мережа компанії має центральне ядро, яке складається з шести роутерів. Ці роутери забезпечують основні мережеві функції, такі як маршрутизація між різними локальними мережами (LAN), управління трафіком і забезпечення безпеки даних на рівні мережі.

Для забезпечення підключення кінцевих пристроїв до мережі використовуються сім комутаторів на рівні доступу. Ці комутатори розташовані в кожній з шести локальних мереж і відповідають за обробку і пересилання даних між робочими станціями і серверами, що підключені до них. В таблиці 2.1 наведено специфікацію обладнання.

Таблиця 2.1 – Специфікація обладнання

№	Найменування і технічна характеристика	Тип, марка, позначення документа	Одиниці виміру	Кількість
1	Cisco ASR1001-X System, Crypto, 4 built-in GE, Dual P/S, 20Gbit, 6x1000Base-X (SFP), 2x10G SFP+ інтегровані RP, SIP та ESP, 1xNIM, 1xSPA, RAM 8Gb, 2xAC	Cisco 1808	Од.	5
2	Комутатор 24 x Ethernet 10/100/1000 Мбіт/сек, RIP v1, RIP v2, OSPF, USB-порт, LAN Base, 4 SFP слоти	Cisco Nexus	Од.	23
3	Роутер з підтримкою 4G LTE, Wi-Fi	TP-Link Archer MR600	Од.	15

	802.11ac, 4 x Gigabit Ethernet порти, USB-порт для підключення зовнішнього сховища			
4	Мережева карта 10 Gigabit Ethernet, PCIe x4, підтримка RDMA over Converged Ethernet (RoCE)	Intel X550-T2	Од.	30
5	Бездротова точка доступу, Wi-Fi 6 (802.11ax), MU-MIMO, 4x4:4, 2.4GHz/5GHz, до 4.8Gbps, PoE підтримка	Ubiquiti UniFi 6 LR	Од.	50
6	Маршрутизатор корпоративного класу, 8 x Gigabit Ethernet порти, підтримка VPN, вбудований	MikroTik RB4011iGS+RM	Од.	10

	міжмережевий екран, USB-порт			
7	Оптичний трансивер, 10GBase-SR, Multi-Mode, LC connector, до 300 метрів	Cisco SFP-10G-SR	Од.	40

2.4 Розрахунок інтенсивності вихідного трафіку найбільшої локальної мережі підприємства

В підмережі встановлений комутатор Cisco2960, який об'єднує 188 кінцевих користувачів. Вихідний трафік з комутатора передається до роутера через лінію з пропускнуою здатністю 1000 Мбіт/с. Щоб уникнути перенасичення комутатора, швидкість надходження пакетів повинна відповідати швидкості їх відправлення. Припускається, що всі 188 користувачів підмережі одночасно користуються мережею. Середня інтенсивність трафіку $\mu=217$ кадрів на секунду, а середня довжина повідомлення складає 1150 байтів.

$$P_{p.p.} = \mu \times L_{нов} \times N \times 8 = 15.84 \text{ Мбіт/с} \quad (2.1)$$

Розрахунок загального навантаження на комутатор за формулою (2.2):

$$\mu_{вих} = \mu / L_{нов} \times 8 = 0.0213 \text{ пакетів/с} \quad (2.2)$$

Розрахунок кількості джерел, які може обслуговувати маршрутизатор за формулою (2.3):

$$N = \mu_{вих} / \mu \approx 0.000247 \quad (2.3)$$

Отже, кількість джерел, які може обслуговувати маршрутизатор, приблизно 0.000247, що означає, що в маршрутизатора є достатньо ресурсів для обслуговування всіх джерел у мережі.

Розрахунок інтенсивності вихідного трафіку за формулою (2.4):

$$A = N * \mu = 22,932 \text{ пакетів на секунду} \quad (2.4)$$

Коефіцієнт затримки ρ обчислюється як відношення інтенсивності трафіку вихідного потоку до пропускної здатності маршрутизатора:

$$\rho \approx 1076.15 \quad (2.5)$$

Коефіцієнт зайнятості маршрутизатора:

$$\rho \approx 1076.15 \quad (2.6)$$

Середня затримка кадру, пов'язана з чергою M/M/1, обчислюється за формулою:

$$T = 1 / (\mu_{вих} - \lambda) \approx 0.0437 \quad (2.7)$$

$$\text{Середня довжина черги } L_{чер} \approx 1075.75 \quad (2.8)$$

$$\text{Середній час перебування пакета у черзі } T_{оч} = 0.46 \text{ мкс} \quad (2.9)$$

3 ПРОЕКТУВАННЯ КОРПОРАТИВНОЇ МЕРЕЖІ ТА ПЕРЕВІРКА РОБОТИ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДПРИЄМСТВА

3.1 Розрахунок схеми адресації корпоративної мережі

Використання VLSM (Variable Length Subnet Masking) для корпоративних мереж є критично важливим з точки зору ефективного управління IP-адресами і оптимізації ресурсів мережі. У корпоративних мережах часто потрібно оптимально використовувати доступні IP-адреси. За допомогою VLSM можна точно визначити маску підмереж для кожної частини мережі в залежності від потреб в кількості хостів. Наприклад, для великих підрозділів мережі може використовуватись більша підмережа з великою кількістю доступних IP-адрес, тоді як для менших підрозділів - менша підмережа. Протоколи маршрутизації, такі як OSPF або EIGRP, підтримують VLSM і можуть автоматично адаптуватися до різних масок підмереж. Це забезпечує ефективне використання мережевих ресурсів і зменшує навантаження на мережеві пристрої. Завдяки VLSM адміністратори мережі можуть ефективно керувати IP-адресацією, забезпечуючи масштабованість і простоту управління великими корпоративними мережами.

Щоб розробити VLSM адресацію для мережі з IP 10.25.104.0/22 та заданими підмережами (LAN1=40, LAN2=121, LAN3=188, LAN4=23, LAN5=81), виконаємо наступні кроки.

Крок 1. Визначимо кількість хостів у кожній підмережі та знайдемо найменшу кількість бітів, необхідних для кожної підмережі:

- LAN1 = 40 хостів $\rightarrow 2^6 = 64$ адреси (6 біт для хостів, маска /26);
- LAN2 = 121 хостів $\rightarrow 2^8 = 256$ адрес (8 біт для хостів, маска /24);
- LAN3 = 188 хостів $\rightarrow 2^8 = 256$ адрес (8 біт для хостів, маска /24);
- LAN4 = 23 хостів $\rightarrow 2^5 = 32$ адреси (5 біт для хостів, маска /27);
- LAN5 = 81 хостів $\rightarrow 2^7 = 128$ адрес (7 біт для хостів, маска /25).

Розподілимо підмережі з найбільшої до найменшої, щоб ефективно використовувати доступні IP-адреси.

Крок 2. Розрахуємо адреси для кожної підмережі. Початкова мережа: 10.25.104.0/22 (діапазон адрес: 10.25.104.0 - 10.25.107.255, загалом 1024 адреси).

Підмережа LAN3 (188 хостів, маска /24). Мережа: 10.25.104.0/24. Діапазон адрес: 10.25.104.0 - 10.25.104.255 (256 адрес)

Підмережа LAN2 (121 хост, маска /24). Мережа: 10.25.105.0/24. Діапазон адрес: 10.25.105.0 - 10.25.105.255 (256 адрес).

Підмережа LAN5 (81 хост, маска /25). Мережа: 10.25.106.0/25. Діапазон адрес: 10.25.106.0 - 10.25.106.127 (128 адрес)

Підмережа LAN1 (40 хостів, маска /26). Мережа: 10.25.106.128/26. Діапазон адрес: 10.25.106.128 - 10.25.106.191 (64 адреси).

Підмережа LAN4 (23 хости, маска /27). Мережа: 10.25.106.192/27. Діапазон адрес: 10.25.106.192 - 10.25.106.223 (32 адреси).

Отримуємо (табл.3.1):

- LAN3: 10.25.104.0/24 (256 адрес);
- LAN2: 10.25.105.0/24 (256 адрес);
- LAN5: 10.25.106.0/25 (128 адрес);
- LAN1: 10.25.106.128/26 (64 адреси);
- LAN4: 10.25.106.192/27 (32 адреси).

Таблиця 3.1 – Адресація КС компанії «RokSave»

Назва підмережі	Необхідна кількість вузлів	Номер мережі	Префікс мережі	Діапазон доступних адрес
LAN1	40	10.25.106.128	/26	10.25.106.129 - 10.25.106.190
LAN2	121	10.25.105.0	/24	10.25.105.1 - 10.25.105.254
LAN3	188	10.25.104.0	/24	10.25.104.1 - 10.25.104.254
LAN4	23	10.25.106.192	/27	10.25.106.193 - 10.25.106.222
LAN5	81	10.25.106.0	/25	10.25.106.1 - 10.25.106.126
VLAN 17	14	10.25.64.0	/28	10.25.64.1 - 10.25.64.14

VLAN 27	13	10.25.64.16	/28	10.25.64.17 – 10.25.64.30
VLAN 37	13	10.25.64.32	/28	10.25.64.33 – 10.25.64.46
VLAN 99	3	10.25.64.48	/29	10.25.64.49 – 10.25.64.54
WAN1	2	10.1.13.0	/30	10.1.13.1
WAN2	2	10.2.13.0	/30	10.2.13.1 - 10.2.13.254
WAN3	2	10.3.13.0	/30	10.3.13.1 - 10.3.13.254
WAN4	2	10.4.13.0	/30	10.4.13.1- 10.4.13.254
WAN5	2	10.5.13.0	/30	10.5.13.1- 10.5.13.254
WAN_IPS	2	209.165.202. 0	/30	209.165.202.1- 209.165.202.2 54

3.2 Розрахунок схеми адресації пристроїв для компанії «RokSave»

Адресація пристроїв — це процес присвоєння унікальних адрес кожному пристрою або компоненту в мережі або системі, що дозволяє їм обмінюватися даними. Адресація використовується в різних типах мереж і систем, включаючи комп'ютерні мережі, пам'ять комп'ютера та інші апаратні компоненти.

Кожен пристрій у мережі має унікальну IP-адресу (IPv4 або IPv6), що дозволяє ідентифікувати його у мережі. IP-адреси можуть бути статичними (постійними) або динамічними (присвоюються при підключенні до мережі).

Кожен мережевий адаптер має унікальну фізичну адресу (MAC-адресу), яка використовується на каналному рівні (Layer 2) для обміну даними між пристроями в локальній мережі (LAN).

Відповідно до розрахункових даних з таблиці 3.1, на основі цих вимог була розроблена схема адресації пристроїв, яка представлена у таблиці 3.2.

Таблиця 3.2 – Схема адресації маршрутизаторів

Ім'я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
Sylakov_0	Gig0/0/0	10.25.104.1	/24	-	-	Gig0/2
	Gig0/0/1	10.7.13.2	/24	-	-	Gig0/0/0
Sylakov_1	Gig3/0	10.4.7.1	/24	-	-	Gig0/0/1
	Se0/0	10.1.13.2	/24	-	-	Se0/2/1
	Gig1/0	10.25.106.19 3	/27	-	-	Gig0/2
	Gig2/0.17	10.25.64.1	/28	-	17	Gig0/1
	Gig2/0.27	10.25.64.17	/28	-	27	Gig0/1
	Gig2/0.37	10.25.64.33	/28	-	37	Gig0/1
Sylakov_2	Gig2/0.99	10.25.1.49	/29	-	99	Gig0/1
	Se0/1/1	10.3.13.1	/24	-	-	Se0/1/1
	Se0/2/0	10.2.13.1	/24	-	-	Se0/2/1
	Se0/2/1	10.1.13.1	/24	-	-	Se0/2/0
Sylakov_3	Gig0/0/1	10.25.106.12 9	/26			Gig0/0/0
	Se0/1/1	209.165.202. 1	/30	-	-	Se0/1/1
	Se0/1/1	10.8.7.2	/24	-	-	Se0/2/1
	Se0/2/1	10.9.13.2	/24	-	-	Se0/2/0
Sylakov_4	Gig0/0/1	10.5.13.2	/24	-	-	Gig0/0/0
	Se0/2/1	10.3.13.2	/24	-	-	Se0/1/1
	Se0/1/1	10.8.13.1	/24	-	-	Se0/2/0
	Se0/2/0	10.9.13.1	/24	-	-	Se0/2/1
	Gig0/0/1	10.25.106.1	/26	-	-	Gig0/2

Sylakov_5	Se0/1/1	10.2.13.2	/24	-	-	Se0/2/0
	Gig0/0/1	10.4.13.2	/24	-	-	Gig0/0/0
	Gig0/0/0	10.5.13.1	/24	-	-	Gig0/0/1
Sylakov_ISP	Se0/1/1	209.165.202. 2	/30	-	-	Se0/1/1
	Gig0/0/1	209.165.201. 1	/28	-	-	Fa0
	Gig0/0/0	10.7.13.1	/30	-	-	Gig0/0/1

В таблиці 3.3 наведена адресації проміжних пристроїв-комутаторів серії Sylakov.

Таблиця 3.3 – Схема адресації комутаторів Sylakov

Ім'я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
Sylakov_Switch_0	Fa0/5	-	-	-	27	Fa0
	Fa0/6	-	-	-	27	Fa0
	Fa0/12	-	-	-	17	Fa0
	Fa0/13	-	-	-	17	Fa0
	Fa0/15	-	-	-	37	Fa0
	VLAN99	10.25.105.3 7	/28	-	99	-
Sylakov_Switch_1	Fa0/5	-	-	-	27	Fa0
	Fa0/6	-	-	-	27	Fa0
	Fa0/12	-	-	-	17	Fa0
	Fa0/13	-	-	-	17	Fa0
	VLAN99	10.25.105.3 6	/28	-	99	-

Sylakov_Switch_2	VLAN1	10.25.106.1 50	/26	-	1	-
Sylakov_Switch_3	VLAN1	10.25.106.1 51	/26	-	1	-
Sylakov_Switch_4	VLAN1	10.25.106.1 52	/26	-	1	-
Sylakov_Switch_5	VLAN1	10.25.106.1 20	/25	-	1	-
Sylakov_Switch_6	VLAN1	10.25.106.2 00	/27	-	1	-
Sylakov_Switch_7	VLAN1	10.25.106.1 00	/25	-	1	-

3.3 Налаштування роботи комп'ютерної системи

3.3.1 Базове налаштування конфігурації пристроїв КС

Для даної мережі була обрана топологія "зірка" (star topology). Ця топологія є однією з найпоширеніших та найефективніших для локальних мереж (LAN) завдяки своїй гнучкості та надійності. Обрано топологію зірка з комутатором (Switch-based Star Topology), що є активною зіркою. Цей вибір обумовлений наступними причинами:

- комутатори (switches) забезпечують ефективне управління трафіком, що дозволяє кожному пристрою мати повну пропускну здатність мережевого з'єднання;
- легко додавати нові пристрої до мережі, просто підключаючи їх до комутатора;
- всі пристрої підключені до одного центрального комутатора, що спрощує управління мережею і діагностику несправностей;
- при виході з ладу одного з пристроїв мережа продовжує функціонувати, за винятком випадків виходу з ладу центрального комутатора.

На рисунку 3.1 представлена логічна топологія компанії «RokSave».

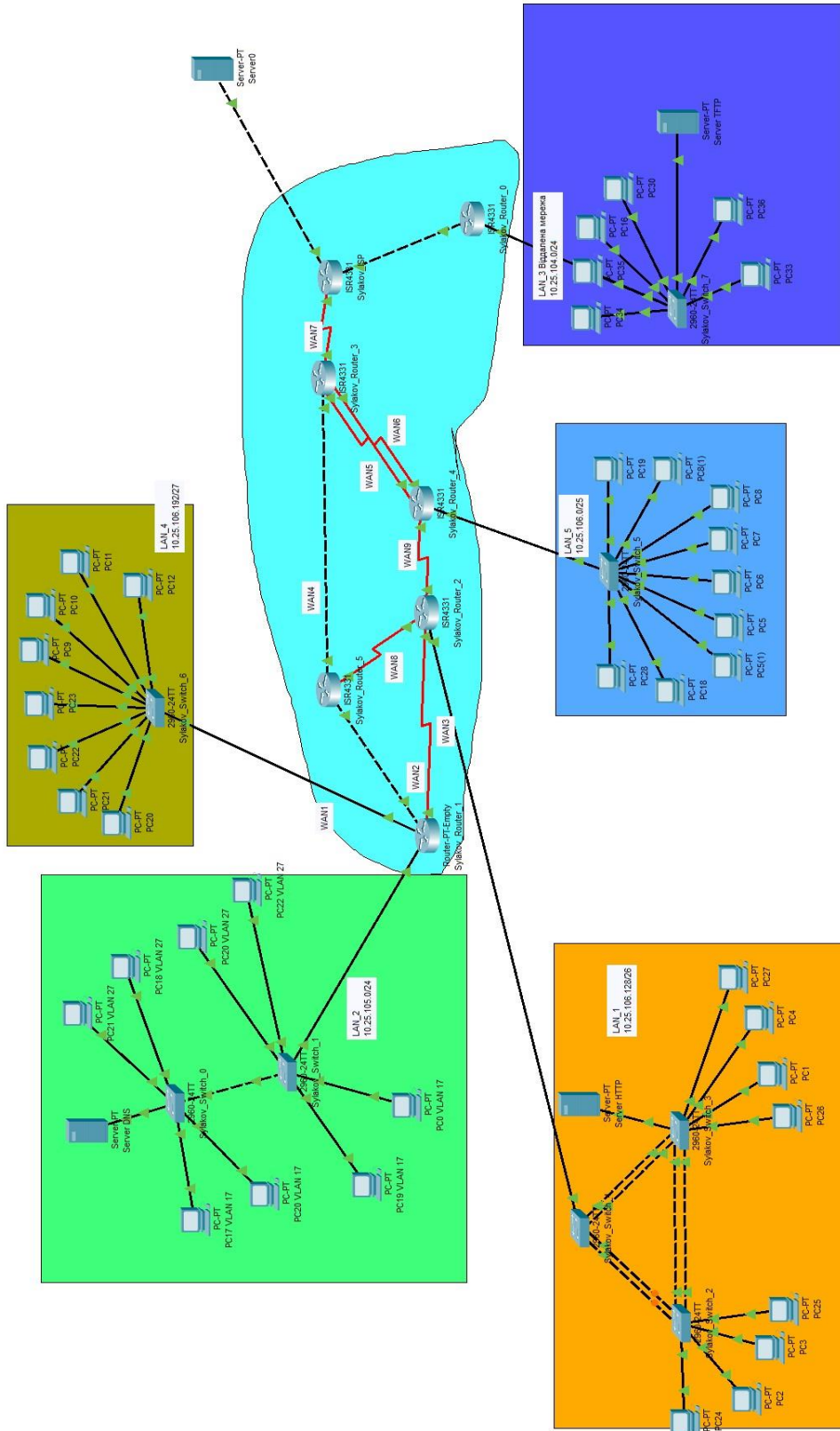


Рисунок 3.1 – Логічна топологія комп'ютерної мережі компанії «RokSave»

Було проведено базові налаштування всього мережевого обладнання, що включає такі дії:

- присвоєння унікального імені кожному пристрою для його ідентифікації в мережі;
- конфігурація доменного імені, що збігається з ім'ям пристрою для уніфікації;
- створення облікового запису локального адміністратора для управління пристроєм;
- налаштування пароля для привілейованого режиму, щоб обмежити доступ до конфігураційних налаштувань;
- установлення пароля для консольного порту, щоб забезпечити його захист;
- конфігурація пароля для vty-ліній, що використовуються для віддаленого доступу;
- створення стартового банерного повідомлення для інформування користувачів про умови користування мережею;
- увімкнення шифрування паролів для підвищення рівня безпеки;
- активування ssh для безпечного віддаленого управління пристроями.

Ось приклад команд, які були використані для цих налаштувань на Sylakov_Switch_0:

```
Switch(config)#hostname Sylakov_Switch_0
Sylakov_Switch_01 (config)#line console 0
Sylakov_Switch_0 (config-line)#password cisco
Sylakov_Switch_0 (config-line)#login
Sylakov_Switch_0 (config-line)#exit
Sylakov_Switch_0 (config)#line vty 0 15
Sylakov_Switch_0 (config-line)#password cisco
Sylakov_Switch_0 (config-line)#login
Sylakov_Switch_0 (config-line)#exit
```

```
Sylakov_Switch_0 (config)#enable secret class
Sylakov_Switch_0 (config)#service password-encryption
Sylakov_Switch_0 (config)#banner motd # Welcome to the Sylakov_Switch_0#
Sylakov__Switch_0(config)# username 12320_Sylakov privilege 15 password
```

admindisco

Приклад використаних команд для налаштування Sylakov_Router_0:

```
Router(config)#hostname Sylakov_Router_0
Sylakov_Router_0(config)#line console 0
Sylakov_Router_0(config-line)#password cisco
Sylakov_Router_0(config-line)#login
Sylakov_Router_0(config-line)#exit
Sylakov_Router_0(config)#line vty 0 15
Sylakov_Router_0(config-line)#password cisco
Sylakov_Router_0(config-line)#login
Sylakov_Router_0(config-line)#exit
Sylakov_Router_0(config)#enable secret class
Sylakov_Router_0(config)#service password-encryption
Sylakov_Router_0(config)#banner motd # Welcome to the Sylakov_Router_0!#
Sylakov_Router_0(config)#ip domain name Sylakov_Router_0
Sylakov_Router_0(config)#ip ssh version 2
Sylakov_Router_0(config)#crypto key generate rsa
How many bits in the modulus [512]: 1024
Sylakov_Router_0(config)# username 12320_Sylakov privilege 15 password
```

admindisco

```
Sylakov_Router_0(config)#line vty 0 15
Sylakov_Router_0(config-line)#transport input ssh
Sylakov_Router_0(config-line)#login local
Sylakov_Router_0(config-line)#exec-time 60 0
Sylakov_Router_0(config)#int g0/0
Sylakov_Router_0(config-if)#no sh duwn
```



```
Sylakov_Router_0(config-if)#ip addr 10.7.13.2 255.255.255.0
```

```
Sylakov_Router_0(config-if)#exit
```

```
Sylakov_2(config-if)#clock rate 128000
```

3.3.2 Налаштування маршрутизаторів в КС компанії «RokSave»

Маршрутизація є ключовим аспектом в будь-якій комп'ютерній мережі, що відповідає за направлення пакетів даних від одного пристрою до іншого через мережу. Основні завдання маршрутизації включають визначення оптимальних шляхів для доставки даних, управління трафіком і забезпечення надійності мережі.

Процес налаштування маршрутизації зазвичай включає наступні кроки:

–визначення основних параметрів маршрутизатора: встановлення імені хоста маршрутизатора; налаштування паролів доступу до консолі і привілейованого режиму;

–налаштування мережних інтерфейсів: назначення IP-адреси і маски підмережі на кожному інтерфейсі маршрутизатора; увімкнення інтерфейсів для прийому та передачі даних.

–вибір та налаштування протоколу маршрутизації: визначення протоколу маршрутизації, який буде використовуватися в мережі (в нашій роботі обрано протокол OSPF); налаштування параметрів протоколу маршрутизації, таких як області OSPF, метрики маршрутизації тощо;

–перевірка налаштувань і моніторинг маршрутизації: використання команд моніторингу для перевірки стану мережі і таблиць маршрутизації; виявлення проблем і вирішення їх шляхом оптимізації налаштувань маршрутизації або коригування маршрутів.

```

Router0(config)# hostname Sylakov_Router_0
Sylakov_Router_0(config)# enable password 7 0822404F1A0A
Sylakov_Router_0(config)# username 12320_Sylakov password 7 082048430017061E010803
Sylakov_Router_0(config)# ip dhcp excluded-address 10.25.104.0 10.25.104.4
Sylakov_Router_0(config)# no ip dhcp excluded-address 172.24.64.128 172.24.64.133
Sylakov_Router_0(config)# ip dhcp pool polllan3
Sylakov_Router_0(dhcp-config)# network 10.25.104.0 255.255.255.0
Sylakov_Router_0(dhcp-config)# default-router 10.25.104.1
Sylakov_Router_0(dhcp-config)# dns-server 209.165.201.5
Sylakov_Router_0(config)# ip nat inside source list 101 interface GigabitEthernet0/0/1 overload
Sylakov_Router_0(config)# crypto isakmp policy 1
Sylakov_Router_0(config-isakmp)# encr aes 256
Sylakov_Router_0(config-isakmp)# authentication pre-share
Sylakov_Router_0(config)# crypto isakmp key cisco address 209.165.202.1
Sylakov_Router_0(config)# crypto ipsec transform-set VPN-IPSEC-SET esp-aes esp-sha-hmac
Sylakov_Router_0(config)# crypto map MAP 1 ipsec-isakmp
Sylakov_Router_0(config-crypto-map)# set peer 209.165.202.1
Sylakov_Router_0(config-crypto-map)# set transform-set VPN-IPSEC-SET
Sylakov_Router_0(config-crypto-map)# match address VPN
Sylakov_Router_0(config)# interface GigabitEthernet0/0/0
Sylakov_Router_0(config-if)# ip address 10.25.104.1 255.255.255.0
Sylakov_Router_0(config-if)# ip nat inside
Sylakov_Router_0(config-if)# duplex auto
Sylakov_Router_0(config-if)# speed auto
Sylakov_Router_0(config)# interface GigabitEthernet0/0/1
Sylakov_Router_0(config-if)# ip address 10.7.7.2 255.255.255.0
Sylakov_Router_0(config-if)# ip access-group 100 in
Sylakov_Router_0(config-if)# ip nat outside
Sylakov_Router_0(config-if)# duplex auto
Sylakov_Router_0(config-if)# speed auto
Sylakov_Router_0(config-if)# crypto map MAP
Sylakov_Router_0(config)# router ospf 1
Sylakov_Router_0(config-router)# log-adjacency-changes
Sylakov_Router_0(config-router)# passive-interface GigabitEthernet0/0/0
Sylakov_Router_0(config-router)# network 10.7.7.0 0.0.0.255 area 0
Sylakov_Router_0(config-router)# network 10.25.104.0 0.0.0.255 area 0
Sylakov_Router_0(config-router)# no network 172.24.64.128 0.0.0.127 area 0
Sylakov_Router_0(config)# ip route 209.165.200.0 255.255.255.0 10.7.7.1
Sylakov_Router_0(config)# banner motd #Welcome to the Sylakov_Router_0#
Sylakov_Router_0(config)# line con 0
Sylakov_Router_0(config-line)# password 7 0822455D0A16
Sylakov_Router_0(config-line)# login
Sylakov_Router_0(config)# line vty 0 4
Sylakov_Router_0(config-line)# password 7 0822455D0A16
Sylakov_Router_0(config-line)# login local
Sylakov_Router_0(config-line)# transport input ssh
Sylakov_Router_0(config)# line vty 5 15
Sylakov_Router_0(config-line)# password 7 0822455D0A16
Sylakov_Router_0(config-line)# login local
Sylakov_Router_0(config-line)# transport input ssh

```

Рисунок 3.2 – Команди для маршрутизатора Sylakov_Router_0

На рис.3.2 встановлено ім'я хоста маршрутизатора: Sylakov_Router_0. Встановлено пароль для режиму привілейованого доступу (enable password). Створено локального користувача з іменем 12320_Sylakov і паролем для входу. Визначено діапазони IP-адрес, які будуть виключені з розподілу DHCP. Створено пул DHCP з назвою polllan3 для сегмента мережі 10.25.104.0/24. Налаштовано параметри пулу DHCP, включаючи мережеву адресу, шлюз за замовчуванням та DNS-сервер 209.165.201.5. Встановлено правило NAT для перетворення IP-адрес внутрішньої мережі на зовнішню адресу інтерфейсу GigabitEthernet0/0/1. Налаштовано політику шифрування для інтерфейсу ISAKMP з використанням AES-256 і методом аутентифікації передачі попереднього ключа.

Встановлено ключ ISAKMP для безпеки IPsec з адресою піринга 209.165.202.1. Створено набір трансформацій IPsec з шифруванням AES та аутентифікацією SHA-НМАС. Створено криптографічне відображення MAP з ідентифікатором 1 для встановлення IPsec тунелю до піринга 209.165.202.1.

Налаштовано IP-адресу та параметри для інтерфейсів GigabitEthernet0/0/0 і GigabitEthernet0/0/1. Встановлено режими duplex auto та speed auto для автоматичного визначення параметрів дуплексу і швидкості.

```
Router1# configure terminal
Sylakov_Router_1(config)# hostname Sylakov_Router_1
Sylakov_Router_1(config)# enable secret cisco
Sylakov_Router_1(dhcp-config)# network 10.25.64.0 255.255.255.240
Sylakov_Router_1(dhcp-config)# default-router 10.25.64.1
Sylakov_Router_1(dhcp-config)# dns-server 10.25.64.35
Sylakov_Router_1(config)# ip dhcp pool pollvlan27
Sylakov_Router_1(dhcp-config)# network 10.25.64.32 255.255.255.240
Sylakov_Router_1(dhcp-config)# default-router 10.25.64.33
Sylakov_Router_1(dhcp-config)# dns-server 10.25.64.35
Sylakov_Router_1(config)# ip dhcp pool LAN2
Sylakov_Router_1(dhcp-config)# network 10.25.105.0 255.255.255.0
Sylakov_Router_1(dhcp-config)# default-router 10.25.105.1
Sylakov_Router_1(dhcp-config)# dns-server 10.25.105.2
Sylakov_Router_1(config)# ip dhcp pool LAN4
Sylakov_Router_1(dhcp-config)# network 10.25.106.192 255.255.255.224
Sylakov_Router_1(dhcp-config)# default-router 10.25.106.193
Sylakov_Router_1(dhcp-config)# dns-server 10.25.106.194
Sylakov_Router_1(config)# aaa new-model
Sylakov_Router_1(config)# aaa authentication login default local
Sylakov_Router_1(config)# aaa authentication login radius-auth group radius local
Sylakov_Router_1(config)# username 12320_Sylakov password 082048430017061E010803
Sylakov_Router_1(config)# ip domain-name Sylakov_Router_1
Sylakov_Router_1(config)# interface Serial0/0
Sylakov_Router_1(config-if)# ip address 10.1.7.2 255.255.255.0
Sylakov_Router_1(config-if)# ip ospf cost 7500
Sylakov_Router_1(config-if)# clock rate 128000
Sylakov_Router_1(config)# interface GigabitEthernet1/0
Sylakov_Router_1(config-if)# ip address 10.25.106.193 255.255.255.224
Sylakov_Router_1(config-if)# duplex auto
Sylakov_Router_1(config-if)# speed auto
Sylakov_Router_1(config)# interface GigabitEthernet2/0
Sylakov_Router_1(config-if)# no ip address
Sylakov_Router_1(config-if)# duplex auto
Sylakov_Router_1(config-if)# speed auto
Sylakov_Router_1(config)# interface GigabitEthernet2/0.17
Sylakov_Router_1(config-if)# encapsulation dot1Q 17
Sylakov_Router_1(config-if)# ip address 10.25.64.1 255.255.255.240
Sylakov_Router_1(config)# interface GigabitEthernet2/0.27
Sylakov_Router_1(config-if)# encapsulation dot1Q 27
Sylakov_Router_1(config-if)# ip address 10.25.64.97 255.255.255.240
Sylakov_Router_1(config)# interface GigabitEthernet2/0.37
Sylakov_Router_1(config-if)# encapsulation dot1Q 37
Sylakov_Router_1(config-if)# ip address 10.25.64.33 255.255.255.240
Sylakov_Router_1(config)# interface GigabitEthernet2/0.99
Sylakov_Router_1(config-if)# encapsulation dot1Q 99
Sylakov_Router_1(config-if)# ip address 10.25.64.49 255.255.255.248
Sylakov_Router_1(config)# interface GigabitEthernet3/0
Sylakov_Router_1(config-if)# ip address 10.4.7.1 255.255.255.0
Sylakov_Router_1(config-if)# duplex auto
Sylakov_Router_1(config-if)# speed auto
Sylakov_Router_1(config)# banner motd #Welcome to the Sylakov_Router_1#
Sylakov_Router_1(config)# line con 0
```

Рисунок 3.3 – Команди для маршрутизатора Sylakov_Router_1

На маршрутизаторі Cisco з іменем Sylakov_Router_1 було налаштовано протокол OSPF (Open Shortest Path First) за допомогою кількох кроків. Спочатку було

введено команди для переходу у режим конфігурації та визначення номера процесу OSPF. Встановлено журналювання змін стану з'єднань (log-adjacency-changes) для відстеження змін. Деякі інтерфейси були визначені як пасивні для OSPF (passive-interface), що означає, що через них OSPF не відправлятиме анонси, але буде приймати їх. Після цього були визначені мережі, які OSPF [4] має анонсувати. Це включало мережу 10.0.0.0/8, всі адреси в діапазоні 10.25.105.0/24 та адреси в діапазоні 10.25.106.192/27. Останнім кроком було збереження налаштувань у постійну пам'ять маршрутизатора, щоб зміни застосувалися після перезавантаження. Ці налаштування дозволяють маршрутизатору ефективно обмінюватися маршрутною інформацією в мережі за допомогою OSPF і забезпечують належне функціонування маршрутизаційних процесів в заданому середовищі (рис.3.4).

```
Sylakov_Router_1# configure terminal
Sylakov_Router_1(config)# router ospf 1
Sylakov_Router_1(config-router)# log-adjacency-changes
Sylakov_Router_1(config-router)# passive-interface GigabitEthernet1/0
Sylakov_Router_1(config-router)# passive-interface GigabitEthernet2/0
Sylakov_Router_1(config-router)# network 10.0.0.0 0.255.255.255 area 0
Sylakov_Router_1(config-router)# network 10.25.105.0 0.0.0.255 area 0
Sylakov_Router_1(config-router)# network 10.25.106.192 0.0.0.31 area 0
Sylakov_Router_1(config-router)# end
Sylakov_Router_1# write memory
```

Рисунок 3.4 – Приклад налаштування протоколу OSPF для Sylakov_Router_1

Після налаштування протоколу OSPF важливо перевірити його коректність та ефективність. На рисунку 3.5 представлена таблиця маршрутизації для Sylakov_Router_1, де символ «O» позначає використання протоколу OSPF.

Routing Table for Sylakov_Router_1

Type	Network	Port	Next Hop IP	Metric
C	10.25.64.0/28	GigabitEthernet2/0.17	---	0/0
C	10.25.64.32/28	GigabitEthernet2/0.37	---	0/0
C	10.25.64.48/29	GigabitEthernet2/0.99	---	0/0
C	10.25.64.96/28	GigabitEthernet2/0.27	---	0/0
O	10.25.104.0/24	GigabitEthernet3/0	10.4.7.2	110/7504
O	10.25.106.0/25	GigabitEthernet3/0	10.4.7.2	110/7503
O	10.25.106.128/26	Serial0/0	10.1.7.1	110/7501
C	10.25.106.192/27	GigabitEthernet1/0	---	0/0
O	209.165.201.0/28	GigabitEthernet3/0	10.4.7.2	110/7503
O	209.165.202.0/30	GigabitEthernet3/0	10.4.7.2	110/7502

Рисунок 3.5 – Таблиця маршрутизації Sylakov_1

На рис.3.6 наведено результат команди `show ip route ospf` для Sylakov_1, щоб переглянути таблицю маршрутизації OSPF та переконатися, що всі необхідні маршрути додані до таблиці.

```

Sylakov_Router_1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Sylakov_Router_1(config)#ex
Sylakov_Router_1#
*SYS-S-CONFIG_I: Configured from console by console

Sylakov_Router_1#disable
Sylakov_Router_1#show ip route ospf
 10.0.0.0/8 is variably subnetted, 16 subnets, 6 masks
O   10.2.7.0 [110/7501] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.3.7.0 [110/15000] via 10.1.7.1, 4294967274:4294967289:4294967244, Serial0/0
O   10.5.7.0 [110/2] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.7.7.0 [110/7503] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.8.7.0 [110/7502] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.9.7.0 [110/7502] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.25.104.0 [110/7504] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.25.106.0 [110/7503] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
O   10.25.106.128 [110/7501] via 10.1.7.1, 4294967274:4294967289:4294967244,
Serial0/0
 209.165.201.0/28 is subnetted, 1 subnets
O   209.165.201.0 [110/7503] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
 209.165.202.0/30 is subnetted, 1 subnets
O   209.165.202.0 [110/7502] via 10.4.7.2, 4294967274:4294967288:4294967264,
GigabitEthernet3/0
Sylakov_Router_1>

```

Рисунок 3.6 – Таблиця маршрутизації Sylakov_1

3.3.3 Налаштування служби агрегування каналів LACP

В даній роботі було налаштовано EtherChannel (або Port Channel). Налаштуємо інтерфейси FastEthernet0/21-24 на режим trunk (рис.3.7) – це означає, що ці порти будуть працювати в режимі, придатному для передачі міжкомпанійного трафіка, тобто кадрів даних которые проходят через порт будет содержать VLAN иначе будет игнорироваться данный VLAN.

```
Sylakov_Switch__2(config)#interface range FastEthernet0/21-24
Sylakov_Switch__2(config-if-range)#switchport mode trunk
Sylakov_Switch__2(config-if-range)#interface range FastEthernet0/21-22
Sylakov_Switch__2(config-if-range)#channel-group 1 mode active
Sylakov_Switch__2(config-if-range)#interface range FastEthernet0/23-24
Sylakov_Switch__2(config-if-range)#channel-group 2 mode active

Sylakov_Switch__3(config)#interface range FastEthernet0/21-22
Sylakov_Switch__3(config-if-range)#channel-group 1 mode passive
Sylakov_Switch__3(config-if-range)#interface range FastEthernet0/23-24
Sylakov_Switch__3(config-if-range)#switchport mode trunk
Sylakov_Switch__3(config-if-range)#channel-group 3 mode active

Sylakov_Switch__4(config)#interface range FastEthernet0/23-24
Sylakov_Switch__4(config-if-range)#channel-group 2 mode passive
Sylakov_Switch__4(config-if-range)#interface range FastEthernet0/21-22
Sylakov_Switch__4(config-if-range)#channel-group 3 mode passive
```

Рисунок 3.7 – Налаштування channel-group2 для Sylakov_Switch__4

3.3.4 Налаштування VLAN Головного управління Пенсійного фонду України

Для налаштування VLAN спочатку необхідно налаштувати під-інтерфейси на маршрутизаторі, який буде виступати в якості шлюза для цієї мережі. Згідно з логічною топологією, це Sylakov_Router__1. Для цього маршрутизатора було створено 3 під-інтерфейси, номери після крапки були вибрані відповідно до номеру VLAN, до якого буде призначено цей порт.

Конфігураційний код для комутатора Sylakov_Switch__0 включає налаштування VLAN та розподіл портів для різних відділень компанії "RokSave". Спочатку створюються VLAN для адміністрування, маркетингу, розробки, інтерактивних сервісів та нативного VLAN. Порти FastEthernet0/12-14 призначені для відділу адміністрації, порти FastEthernet0/5-10 - для відділу маркетингу, а порти FastEthernet0/15-24 - для відділу розробки. Інтерфейс VLAN99 налаштований з IP-

адресою 10.25.106.140/26, забезпечуючи зв'язок з іншими підмережами. Встановлено шлюз за замовчуванням на 10.25.106.129 для доступу до зовнішніх мереж. Порт GigabitEthernet0/2 налаштований як транкований з native VLAN 100 і дозволеними VLAN 17, 27, 37, 99, 100, щоб обмінюватися трафіком між комутаторами і мережами. (рис.3.8)

```
Sylakov_Switch__2(config)#interface range FastEthernet0/21-24
Sylakov_Switch__2(config-if-range)#switchport mode trunk
Sylakov_Switch__2(config-if-range)#interface range FastEthernet0/21-22
Sylakov_Switch__2(config-if-range)#channel-group 1 mode active
Sylakov_Switch__2(config-if-range)#interface range FastEthernet0/23-24
Sylakov_Switch__2(config-if-range)#channel-group 2 mode active

Sylakov_Switch__3(config)#interface range FastEthernet0/21-22
Sylakov_Switch__3(config-if-range)#channel-group 1 mode passive
Sylakov_Switch__3(config-if-range)#interface range FastEthernet0/23-24
Sylakov_Switch__3(config-if-range)#switchport mode trunk
Sylakov_Switch__3(config-if-range)#channel-group 3 mode active

Sylakov_Switch__4(config)#interface range FastEthernet0/23-24
Sylakov_Switch__4(config-if-range)#channel-group 2 mode passive
Sylakov_Switch__4(config-if-range)#interface range FastEthernet0/21-22
Sylakov_Switch__4(config-if-range)#channel-group 3 mode passive

Sylakov_Switch__0(config)#vlan 17
Sylakov_Switch__0(config-vlan)#name Administration
Sylakov_Switch__0(config-vlan)#exit
Sylakov_Switch__0(config)#vlan 27
Sylakov_Switch__0(config-vlan)#name Marketing
Sylakov_Switch__0(config-vlan)#exit
Sylakov_Switch__0(config)#vlan 37
Sylakov_Switch__0(config-vlan)#name Development
Sylakov_Switch__0(config-vlan)#exit
Sylakov_Switch__0(config)#vlan 99
Sylakov_Switch__0(config-vlan)#name Interactive Services
Sylakov_Switch__0(config-vlan)#exit
Sylakov_Switch__0(config)#vlan 100
Sylakov_Switch__0(config-vlan)#name Native
Sylakov_Switch__0(config-vlan)#exit

Sylakov_Switch__0(config)#interface range FastEthernet0/12-14
Sylakov_Switch__0(config-if-range)#switchport mode access
Sylakov_Switch__0(config-if-range)#switchport access vlan 17
Sylakov_Switch__0(config-if-range)#exit

Sylakov_Switch__0(config)#interface range FastEthernet0/5-10
Sylakov_Switch__0(config-if-range)#switchport mode access
Sylakov_Switch__0(config-if-range)#switchport access vlan 27
Sylakov_Switch__0(config-if-range)#exit

Sylakov_Switch__0(config)#interface range FastEthernet0/15-24
Sylakov_Switch__0(config-if-range)#switchport mode access
Sylakov_Switch__0(config-if-range)#switchport access vlan 37
Sylakov_Switch__0(config-if-range)#exit

Sylakov_Switch__0(config)#interface Vlan99
Sylakov_Switch__0(config-if)#ip address 10.25.106.140 255.255.255.192
Sylakov_Switch__0(config-if)#exit

Sylakov_Switch__0(config)#ip default-gateway 10.25.106.129

Sylakov_Switch__0(config)#interface GigabitEthernet0/2
Sylakov_Switch__0(config-if)#switchport trunk native vlan 100
```

Рисунок 3.8 – VLAN для Sylakov_Switch__0

3.3.5 Налаштування динамічного NAT

Налаштування Network Address Translation (NAT) є важливим аспектом мережевої конфігурації, особливо в корпоративних та домашніх мережах, де потрібно забезпечити зв'язок між локальною мережею і Інтернетом. NAT перетворює IP-адреси із внутрішніх локальних мереж (Private IP) на публічні IP-адреси (Public IP), що дозволяє пристроям з внутрішньої мережі виходити в Інтернет через одну або кілька публічних IP-адрес [5].

Коли пакети даних з внутрішньої мережі намагаються вийти в Інтернет, NAT перетворює їхні локальні IP-адреси на одну з публічних IP-адрес, які налаштовані на маршрутизаторі або фаїрволі. При отриманні відповіді від Інтернету NAT знову перетворює зовнішню публічну IP-адресу на внутрішню локальну IP-адресу того пристрою, що запитував дані.

NAT використовується для забезпечення безпеки мережі, приховуючи реальну адресу мережі в Інтернеті і запобігаючи нападам, таким як прямі атаки на IP-адресу. Він також дозволяє економити публічні IP-адреси, так як кілька пристроїв може використовувати одну зовнішню IP-адресу для доступу в Інтернет (рис.3.9).

```
Sylakov_Router_3(config)#ip nat pool Internet 209.165.200.5 209.165.200.30 netmask 255.255.255.0
Sylakov_Router_3(config)#access-list 1 permit 10.25.105.0 0.0.0.255
Sylakov_Router_3(config)#interface Serial0/1/1
Sylakov_Router_3(config-if)#ip nat outside
Sylakov_Router_3(config-if)#exit
Sylakov_Router_3(config)#interface Serial0/2/1
Sylakov_Router_3(config-if)#ip nat inside
Sylakov_Router_3(config-if)#exit
Sylakov_Router_3(config)#interface Serial0/2/0
Sylakov_Router_3(config-if)#ip nat inside
Sylakov_Router_3(config-if)#exit
Sylakov_Router_3(config)#interface GigabitEthernet0/0/1
Sylakov_Router_3(config-if)#ip nat inside
Sylakov_Router_3(config-if)#exit
Sylakov_Router_3(config)#ip nat inside source list 1 pool Internet
```

Рисунок 3.9 – NAT для Sylakov_Router_3

3.3.6 Налаштування списків доступу

На маршрутизаторі Sylakov_Router_0 налаштовані розширені списки доступу (ACL) для керування трафіком у мережі. ACL "VPN" включає декілька правил, що дозволяють конкретним підмережам комунікувати з іншими за певними умовами, такими як IP-адреса і маска підмережі. Наприклад, правила дозволяють IP-трафіку з підмережі 10.25.105.0/24 до 10.25.104.0/24, а також з інших підмереж до визначених

цілей. Додатково, ACL 100 використовується для управління загальним доступом до мережі, дозволяючи або блокуючи специфічні види трафіку, такі як OSPF або IP-адреси до певних підмереж. Командою "no access-list 100 permit ip any 10.25.105.0 0.0.0.255" видаляється конкретне правило з цього списку доступу для підмережі 10.25.105.0/24, що раніше дозволяло такому трафіку. (рис.3.10)

```
Sylakov_Router_0(config)#ip access-list extended VPN
Sylakov_Router_0(config-ext-nacl)#permit ip 10.25.105.0 0.0.0.255 10.25.104.0 0.0.0.255
Sylakov_Router_0(config-ext-nacl)#permit ip 10.25.106.192 0.0.0.31 10.25.106.128 0.0.0.63
Sylakov_Router_0(config-ext-nacl)#permit ip 10.24.64.128 0.0.0.127 172.24.24.0 0.0.0.127
Sylakov_Router_0(config-ext-nacl)#permit ip 10.25.106.192 0.0.0.31 10.25.104.0 0.0.0.255
Sylakov_Router_0(config-ext-nacl)#permit ip 10.24.64.128 0.0.0.127 10.25.104.0 0.0.0.255
Sylakov_Router_0(config-ext-nacl)#permit ip 10.25.64.0 0.0.0.15 10.25.104.0 0.0.0.255
Sylakov_Router_0(config)#access-list 100 permit ip any 10.7.7.0 0.0.0.255
Sylakov_Router_0(config)#access-list 100 permit ospf any any
Sylakov_Router_0(config)#no access-list 100 permit ip any 10.25.105.0 0.0.0.255
```

Рисунок 3.10 – Приклад команд для ACL на Sylakov_Router_0

3.3.7 Налаштування аутентифікації службою AAA

Налаштування аутентифікації, авторизації та обліку (AAA) є важливою складовою будь-якої мережі, що дозволяє ефективно керувати доступом користувачів до ресурсів мережі. На прикладі конфігурації маршрутизатора або комутатора Cisco можна побачити, що використовуються сервери аутентифікації, такі як TACACS+ або RADIUS, для перевірки ідентифікації користувачів перед наданням доступу до мережних послуг. Налаштування наведено на рис.3.11 [6].

```

Sylakov_Router_4(config)# hostname Sylakov_Router_4
Sylakov_Router_4(config)# aaa new-model
Sylakov_Router_4(config)# radius-server host 10.25.106.133 auth-port 1645 key radius123

Sylakov_Router_4(config)# interface GigabitEthernet0/0
Sylakov_Router_4(config-if)# ip address 10.25.108.1 255.255.255.0
Sylakov_Router_4(config-if)# duplex auto
Sylakov_Router_4(config-if)# speed auto

Sylakov_Router_4(config)# interface Serial0/0
Sylakov_Router_4(config-if)# ip address 192.168.1.1 255.255.255.252
Sylakov_Router_4(config-if)# clock rate 64000

Sylakov_Router_4(config)# interface Vlan1
Sylakov_Router_4(config-if)# no shutdown
Sylakov_Router_4(config-if)# ip address 10.10.10.1 255.255.255.0

Sylakov_Router_4(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.2

Sylakov_Router_4(config)# access-list 100 permit ip any 10.7.7.0 0.0.0.255
Sylakov_Router_4(config)# access-list 100 permit ospf any any
Sylakov_Router_4(config)# no access-list 100 permit ip any 10.25.105.0 0.0.0.255

Sylakov_Router_4(config)# line vty 0 4
Sylakov_Router_4(config-line)# login authentication default
Sylakov_Router_4(config-line)# transport input telnet ssh

Sylakov_Router_4(config)# end
Sylakov_Router_4# write memory

```

Рисунок 3.11 – AAA на Sylakov_Router__4

Налаштування параметрів та користувачі RADIUS-сервера представлено на рис.3.12.

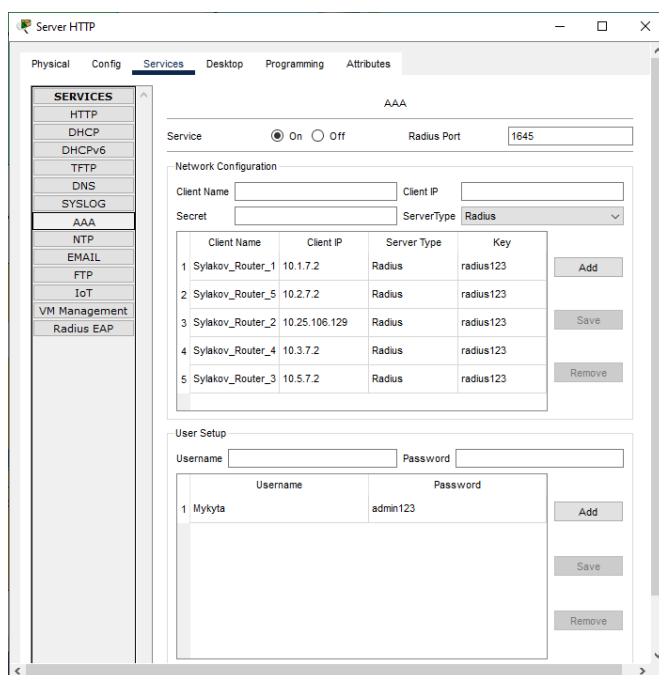
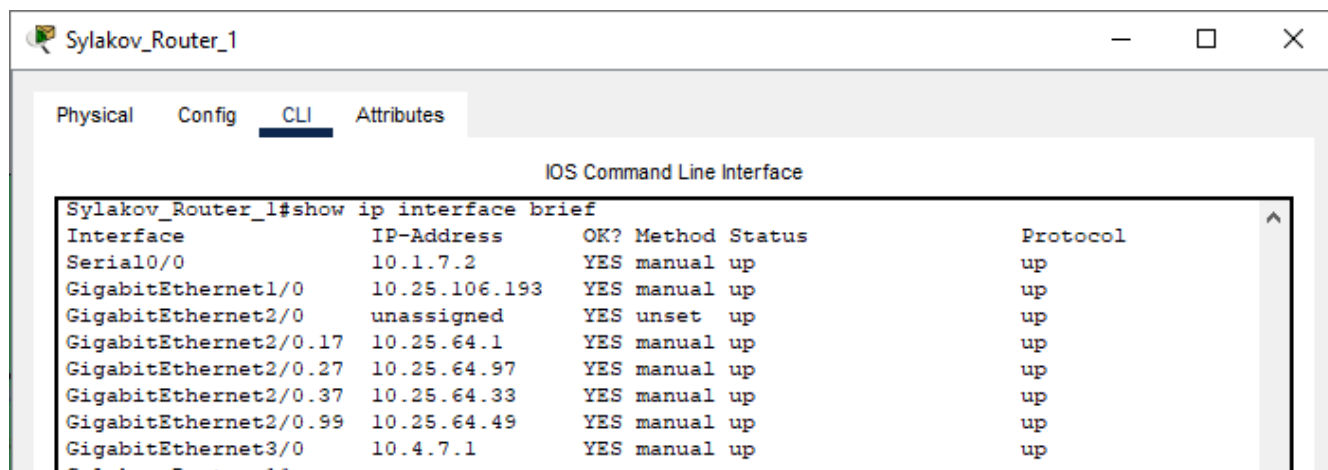


Рисунок 3.12 – RADIUS-сервер

3.4 Перевірка роботи комп'ютерної системи

Здійснення ефективної перевірки налаштування комп'ютерної мережі є критичним етапом для забезпечення її надійності та оптимального функціонування. Першим кроком у цьому процесі є перевірка фізичного підключення. Важливо переконатися, що всі мережеві кабелі належним чином підключені до пристроїв і не пошкоджені, що дозволяє уникнути потенційних проблем зі з'єднанням.

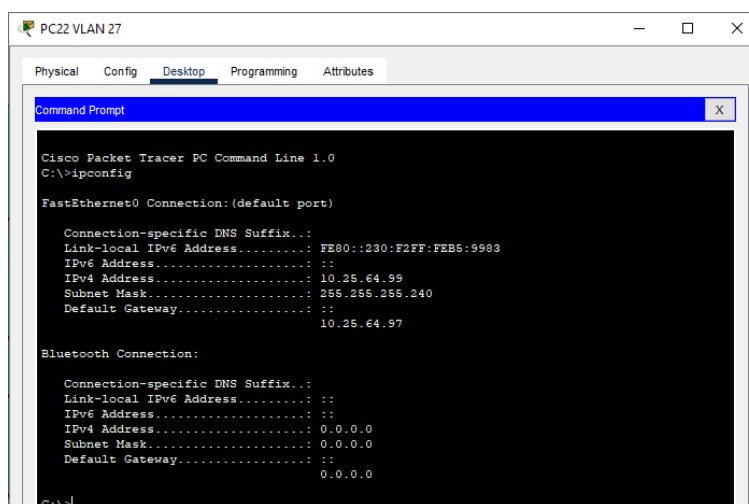
Далі слід перевірити IP-налаштування. Це можна зробити за допомогою команди `show ip interface brief` (рис.3.13) на маршрутизаторах або комутаторах Cisco, щоб впевнитися в правильності присвоєних IP-адрес і стану мережевих інтерфейсів. На комп'ютерах під управлінням Windows використовують команду `ipconfig` для перевірки IP-адреси (рис.3.14).



```

Sylakov_Router_1#show ip interface brief
Interface                IP-Address      OK? Method Status        Protocol
Serial0/0                 10.1.7.2        YES manual  up            up
GigabitEthernet1/0       10.25.106.193   YES manual  up            up
GigabitEthernet2/0       unassigned      YES unset   up            up
GigabitEthernet2/0.17    10.25.64.1      YES manual  up            up
GigabitEthernet2/0.27    10.25.64.97     YES manual  up            up
GigabitEthernet2/0.37    10.25.64.33     YES manual  up            up
GigabitEthernet2/0.99    10.25.64.49     YES manual  up            up
GigabitEthernet3/0       10.4.7.1        YES manual  up            up
  
```

Рисунок 3.13 – Результат команди `show ip interface brief`



```

Cisco Packet Tracer PC Command Line 1.0
C:\>ipconfig

FastEthernet0 Connection: (default port)

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : FE80::230:F2FF:FE85:9983
IPv6 Address . . . . . : ::
IPv4 Address . . . . . : 10.25.64.99
Subnet Mask . . . . . : 255.255.255.240
Default Gateway . . . . . :
                               10.25.64.97

Bluetooth Connection:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . :
IPv6 Address . . . . . : ::
IPv4 Address . . . . . : 0.0.0.0
Subnet Mask . . . . . : 0.0.0.0
Default Gateway . . . . . :
                               0.0.0.0
  
```

Рисунок 3.14 – Результат команди `ipconfig`

Не менш важливим є перевірка маршрутизації. Команда `show ip route` (рис.3.15) дозволяє переглянути таблиці маршрутизації на маршрутизаторах, а `route print` (Windows) або `ip route` (Linux) - на комп'ютерах. Це допомагає впевнитися, що дані правильно направляються до їхніх призначень.

```

Sylakov_Router_1#
Sylakov_Router_1#
Sylakov_Router_1#
Sylakov_Router_1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 16 subnets, 6 masks
C    10.1.7.0/24 is directly connected, Serial0/0
O    10.2.7.0/24 [110/7501] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.3.7.0/24 [110/15000] via 10.1.7.1, 00:11:11, Serial0/0
C    10.4.7.0/24 is directly connected, GigabitEthernet3/0
O    10.5.7.0/24 [110/2] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.7.7.0/24 [110/7503] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.8.7.0/24 [110/7502] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.9.7.0/24 [110/7502] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
C    10.25.64.0/28 is directly connected, GigabitEthernet2/0.17
C    10.25.64.32/28 is directly connected, GigabitEthernet2/0.37
C    10.25.64.48/29 is directly connected, GigabitEthernet2/0.99
C    10.25.64.96/28 is directly connected, GigabitEthernet2/0.27
O    10.25.104.0/24 [110/7504] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.25.106.0/25 [110/7503] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    10.25.106.128/26 [110/7501] via 10.1.7.1, 00:11:11, Serial0/0
C    10.25.106.192/27 is directly connected, GigabitEthernet1/0
O    209.165.201.0/28 is subnetted, 1 subnets
O    209.165.201.0 [110/7503] via 10.4.7.2, 00:10:36, GigabitEthernet3/0
O    209.165.202.0/30 is subnetted, 1 subnets
O    209.165.202.0 [110/7502] via 10.4.7.2, 00:10:36, GigabitEthernet3/0

Sylakov_Router_1#

```

Рисунок 3.15 – Результат команди `show ip route`

Для завершення перевірки слід також використовувати інструменти для тестування з'єднання (рис.3.16), такі як `ping`, щоб переконатися в доступності інших пристроїв у мережі. Важливо також відстежувати стан мережевих служб, використовуючи відповідні команди, які дозволяють перевірити їхню належність та функціональність.

Fire	Last Status	Destination	Source	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	LAN3_PC34	LAN2...	ICMP		0.000	N	0	(edit)	
	Successful	LAN1_PC27	LAN2...	ICMP		0.000	N	1	(edit)	
	Successful	LAN5_PC28	LAN2...	ICMP		0.000	N	2	(edit)	
	Successful	LAN4_PC20	LAN2...	ICMP		0.000	N	3	(edit)	

Рисунок 3.16 – Тестування LAN2 з LAN1, LAN3, LAN4 та LAN5

4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ

4.1 Розробка функціоналу чат-бота замовлення таксі

Оскільки вже відомий процес взаємодії між ботом та сервером, потрібно розробити функціонал для цього бота, який призначений для замовлення таксі. Користувач може вказати потрібну адресу для забору та пункт призначення через чат-інтерфейс. Після отримання цієї інформації, бот автоматично передає замовлення до партнерської служби таксі для подальшої обробки.

Основні можливості чат-бота для замовлення таксі включають:

- замовлення таксі – користувач може вказати потрібну адресу забору і пункт призначення;
- сповіщення про статус – користувач отримує повідомлення про статус свого замовлення та приблизний час прибуття таксі;
- відміна замовлення – користувач може відмінити замовлення таксі через чат-бота, якщо це необхідно.

Цей функціонал забезпечує зручний та простий спосіб замовлення транспорту безпосередньо через чат-інтерфейс, спрощуючи процес для користувачів і покращуючи їхнє досвід використання сервісу таксі.

Відповідно до контекстної діаграми, варто створити діаграму декомпозиції моделі. Ця діаграма дозволяє більш детально розглянути процес, описаний у контекстній діаграмі, шляхом розбиття головного процесу на додаткові підпроцеси. Створення діаграми декомпозиції дозволяє описати процес розробки бота для замовлення таксі. Це допоможе встановити чітке розуміння та порядок виконання дій, що буде корисно як у майбутньому процесі розробки, так і для замовника, щоб краще визначити терміни виконання завдань.



Рисунок 4.1 – Діаграма контексту розробки чат-бота

Оскільки в контекстній діаграмі зазначено процес розробки бота для замовлення таксі, діаграма декомпозиції буде детально описувати, як різні фактори та механізми впливають на окремі етапи під час виконання роботи. Діаграму декомпозиції показано на рис. 4.1.

Ця діаграма розбиває процес розробки бота на такі основні етапи:

- реєстрація бота у telegram;
- розробка;
- тестування бота.

Після успішного тестування проект вважається завершеним. Оскільки діаграма декомпозиції вже побудована та визначено ключові етапи розробки, було створено діаграму взаємодії.

Діаграма варіантів використання ілюструє акторів, які діють у системі, та функції, якими вони можуть користуватися. Це допомагає встановити зв'язок між функціоналом інтерфейсу бота та його користувачами, полегшуючи розуміння навігації та встановлення обмежень під час розробки. Діаграму наведено на рис. 4.2.

Існує два типи дозволених користувачів: звичайний користувач і адміністратор. Адміністратор має доступ до всіх функцій звичайного користувача та отримує додаткову інформацію про взаємодії користувачів із системою.

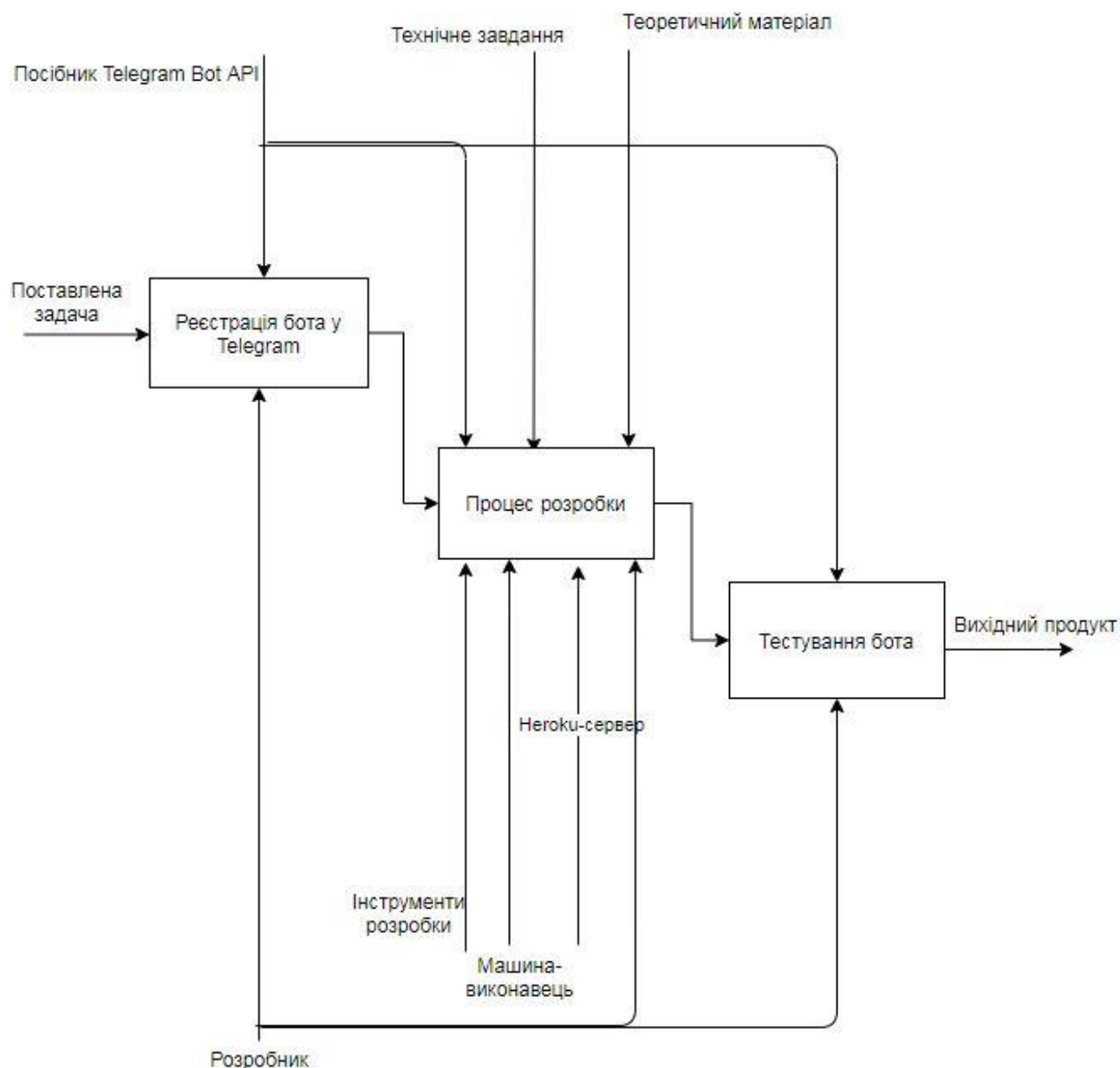


Рисунок 4.2 – Модель декомпозиції створення бота

Діаграма варіантів використання ілюструє різноманітні сценарії взаємодії користувачів з ботом для замовлення таксі. Вона включає акторів, які взаємодіють з системою, та функції, які доступні через інтерфейс бота (рис.4.3). Основні компоненти діаграми варіантів використання подробно описані нижче.

Реєстрація і авторизація. Реєстрація пасажера має можливість створити особистий профіль користувача для замовлення таксі.

Особистий профіль користувача має можливість управління особистою інформацією і налаштуваннями акаунта.

Управління автомобілями. Вибір категорії авто має можливість вибору типу автомобіля для замовлення (економ, комфорт, бізнес тощо).

Додавання інформації про автомобіль має можливість додавання та зміни інформації про автомобіль водія.

Замовлення та обробка замовлень. Виклик авто має можливість пасажирів замовити таксі через бота. Вибір авто має можливість вибору конкретного автомобіля з наявних водієм.

Адміністративні функції. Адміністратор має доступ до управління користувачами, водіями і замовленнями, може переглядати статистику та вносити зміни в систему.

Цей функціонал забезпечує зручний і ефективний процес замовлення таксі для пасажирів, дозволяє водіям керувати своїм профілем і автомобілями, а також надає адміністратору необхідні інструменти для ефективного управління системою.

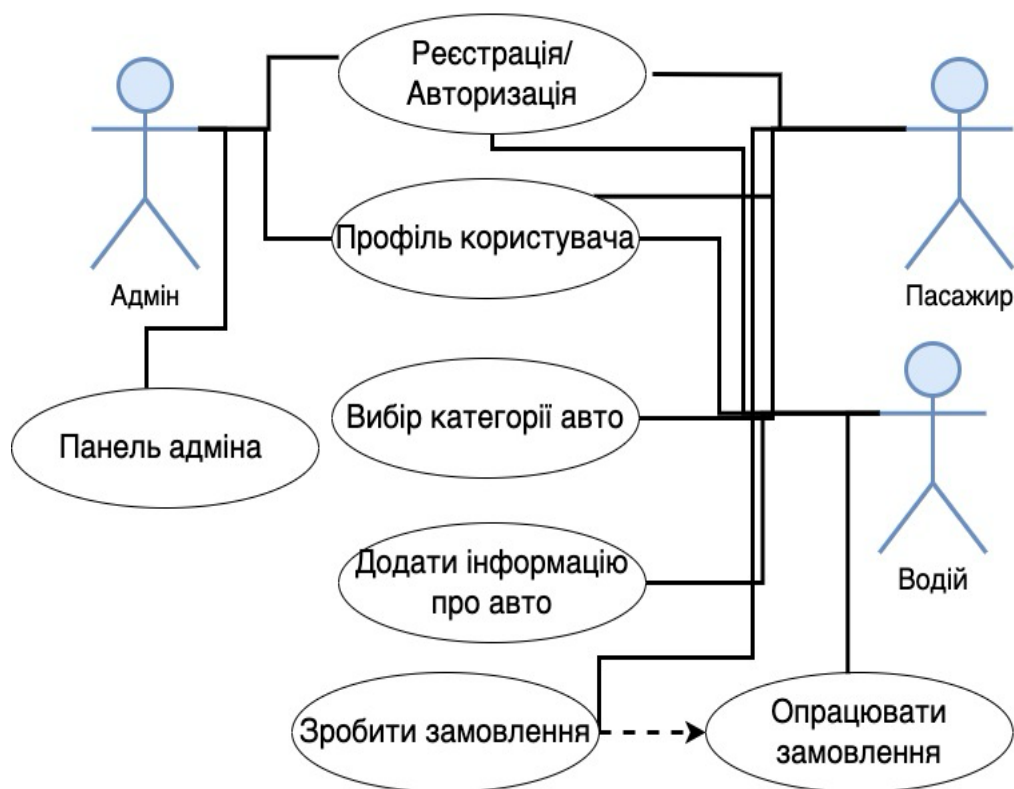


Рисунок 4.3 – UML діаграма варіантів використання

4.2 Розробка чат-боту замовлення таксі

4.2.1 Node.js

Node.js - це програмна платформа, яка використовує двигун V8 для виконання JavaScript-коду в машинний код. Цей двигун перетворює спеціалізовану мову JavaScript у загального призначення. Node.js надає можливість взаємодіяти з пристроями вводу-виводу через свій API, написаний на C++. Також підтримується використання сторонніх бібліотек, створених за допомогою різних мов програмування, що можуть бути викликані з JavaScript-коду [7].

Node.js застосовується переважно як back-end мова програмування, виконуючи роль сервера. Однак існує можливість розробки віконних десктопних додатків за допомогою спеціальних фреймворків, таких як NW.js, AppJS або Electron. Відзначається можливість використання Node.js для розробки додатків в області Інтернету речей (IoT). Сучасні технології все частіше використовують Node.js для програмування "розумної" техніки, такої як смарт-браслети, голосові асистенти та інші пристрої.

На рис.4.4 показано приклад коду, який реалізує простий сервер за допомогою Node.js.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Рисунок 4.4 – Реалізація просто сервера на Node.js

4.2.2 Telegram bot API

Bot API є HTTP-інтерфейсом для взаємодії з ботами у Telegram. Кожен бот представляє собою спеціальний обліковий запис, створений для автоматичної обробки та відправлення повідомлень. Існують два протилежні за логікою способи отримання оновлень від бота:

- long polling – програма періодично опитує сервери Telegram для виявлення нових оновлень для бота. За замовчуванням це відбувається кожні 100 мілісекунд.
- webhook – сервери Telegram автоматично сповіщають про додаток на сервері, як тільки з'являються нові оновлення.

Вхідні оновлення зберігаються на сервері, поки вони не будуть оброблені, але не довше ніж 24 години. Незалежно від способу отримання оновлень, відповідь надсилається у форматі JSON у вигляді об'єкта Update.

Усі запити до Telegram Bot API повинні здійснюватися через HTTPS за такою адресою: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ. Наприклад:

- getUpdates – використовується для отримання оновлень методом long polling.
- setWebhook – прив'язує URL домену, де запусканий бот, до бота.
- sendMessage – надсилає текстове повідомлення користувачу в Telegram.
- sendLocation – відправляє повідомлення з географічними координатами користувачу в Telegram.
- getFile – повертає інформацію про завантажений файл за його ідентифікатором та іншими параметрами.
- Для передачі параметрів у Bot API використовуються чотири способи:
- Запит в URL.
- application/x-www-form-urlencoded.
- application/json.
- multipart/form-data.

4.2.3 Розміщення чат-боту на віддаленому сервері

Для постійного доступу до чат-бота після його реалізації необхідно розмістити його на віддаленому сервері. Для цього була обрана хмарна PaaS-платформа Heroku.

Розгортання на Heroku подібне до роботи з системою контролю версій Git і включає три основні методи [8]:

- heroku Git та Container Registry, використовуючи Heroku CLI;
- підключення облікового запису GitHub з автоматичним розгортанням;
- підключення облікового запису Dropbox з автоматичним розгортанням.

У даному випадку для розгортання чат-бота використовувався Heroku CLI.

Процес включав наступні кроки:

- реєстрація на хмарній SaaS-платформі Heroku;
- завантаження та встановлення Heroku CLI;
- виконання входу до облікового запису Heroku у терміналі або вбудованому в IDE за допомогою команди "heroku login";
- клонування репозиторію з віддаленого Git-сервера Heroku на локальну машину через команду "heroku git -a <APP_NAME>";
- фіксація змін в коді за допомогою команд "git add ." та "git commit -am <COMMIT_NAME>";
- відправлення всіх фіксованих змін на віддалений сервер хмарних обчислень Heroku командою "git push heroku master".

Якщо розгортання виконано успішно, в терміналі з'явиться інформація про стан програми та доступ до неї. Це забезпечить безперебійний доступ користувачів до чат-бота, що тепер розгорнутий на віддаленому сервері хмарних обчислень Heroku.

Результат розгорнення чат-боту на віддаленому сервері наведено на рисунку 4.5.

```

remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:   Procfile declares types   -> (none)
remote:   Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:   Done: 19.4M
remote: -----> Launching...
remote:   Released v3
remote:   https://sfutelegrambotdip.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sfutelegrambotdip.git
 * [new branch]   master -> master

```

Рисунок 4.5 – Результат розгорнення чат-боту на віддаленому сервері

4.3 Тестування роботи чат-бота замовлення таксі

Чат-бот призначений для замовлення таксі у місті Дніпро. Він має функціонал, який дає можливість обирати роль пасажир або водія. При вході до бота користувач обирає, чи хоче він зареєструватися як водій і розпочати роботу, чи ж стати пасажиром і замовити таксі (рис.4.6).

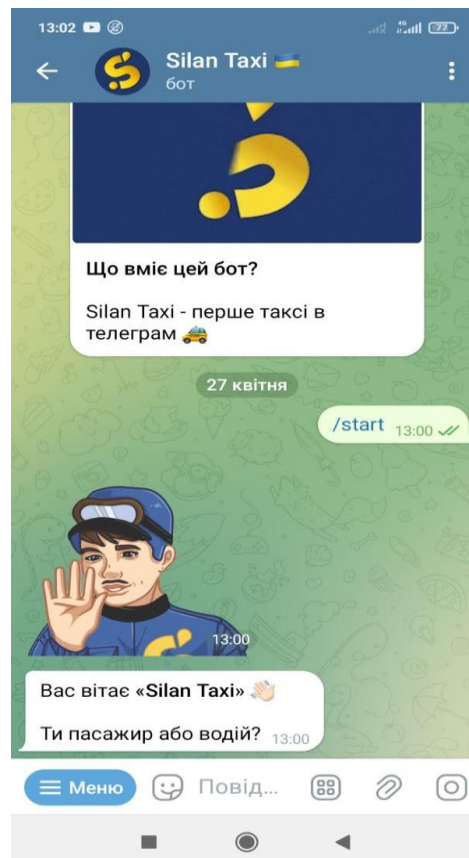


Рисунок 4.6 – Первинний екран користувача

При виборі категорії "Стандарт" водію пропонується ввести марку автомобіля, після чого система просить завантажити фото авто для перевірки адміністрацією. Після підтвердження можливості роботи водій може виходити на лінію (рис.4.7 та рис.4.8).

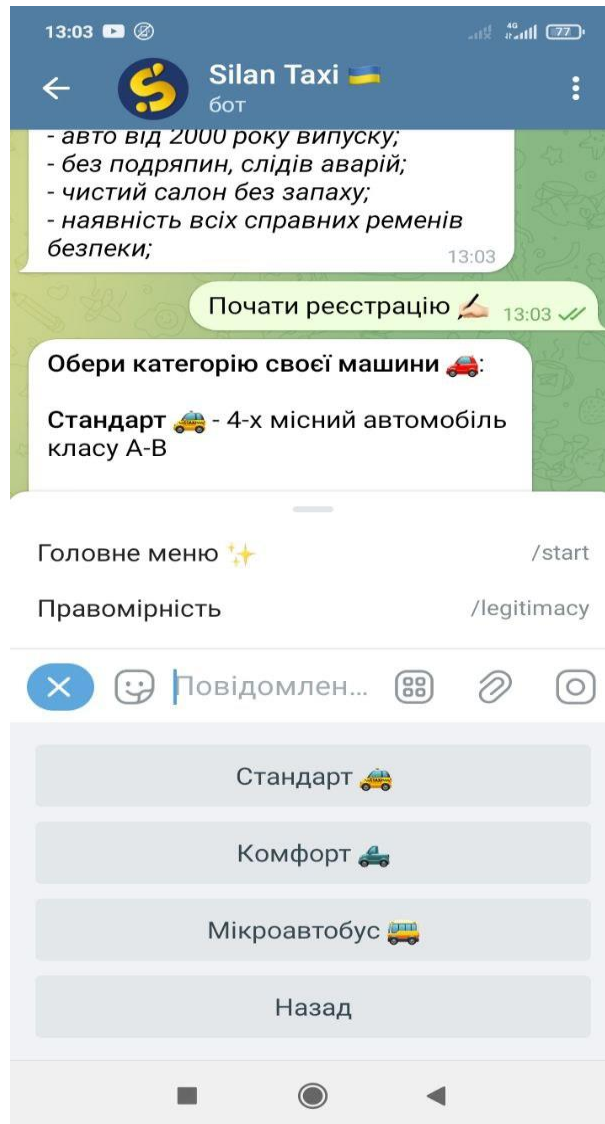


Рисунок 4.7 – Вибір категорії авто

Якщо користувач обирає роль пасажир, йому потрібно поділитися своїм контактом для реєстрації у системі. Далі він отримує повідомлення з проханням вказати місце посадки/висадки пасажир та обрати категорію авто. Бот надає можливість використовувати текстовий ввід адреси або використовувати технологію виявлення геолокації через сервіс Google Maps (рис.4.9).

Після вибору місця і категорії авто, бот показує користувачеві дві адреси та відстань до місця призначення. Користувач може обрати найвигіднішу для нього категорію за авто та ціною.

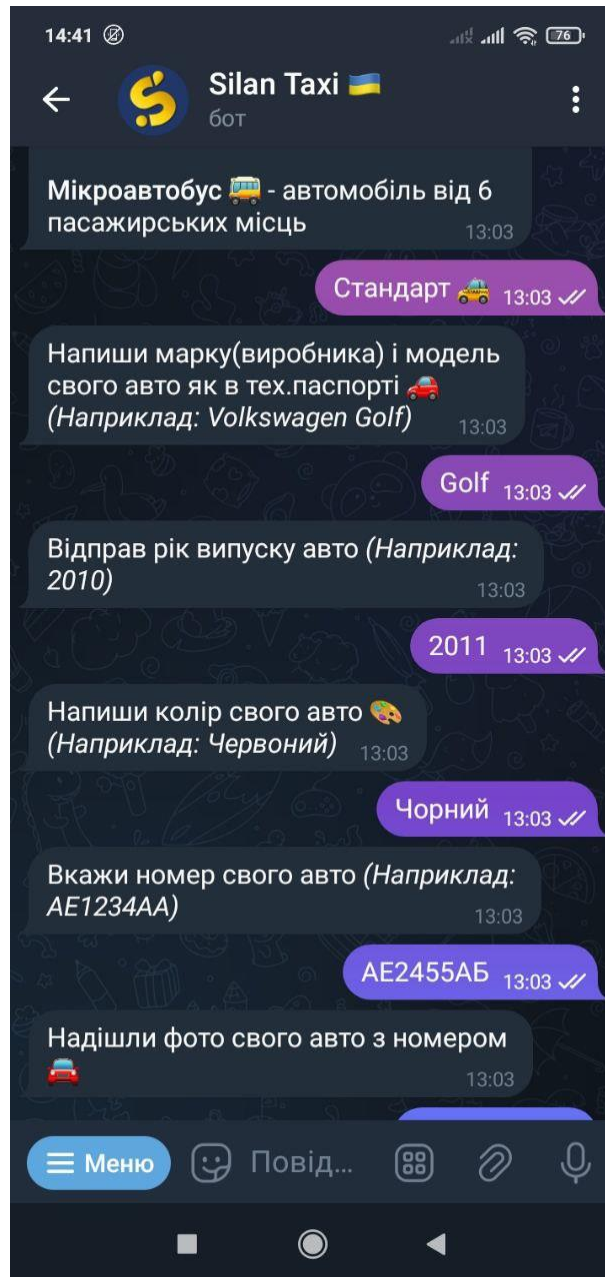


Рисунок 4.8 – Додавання інформації про автомобіль

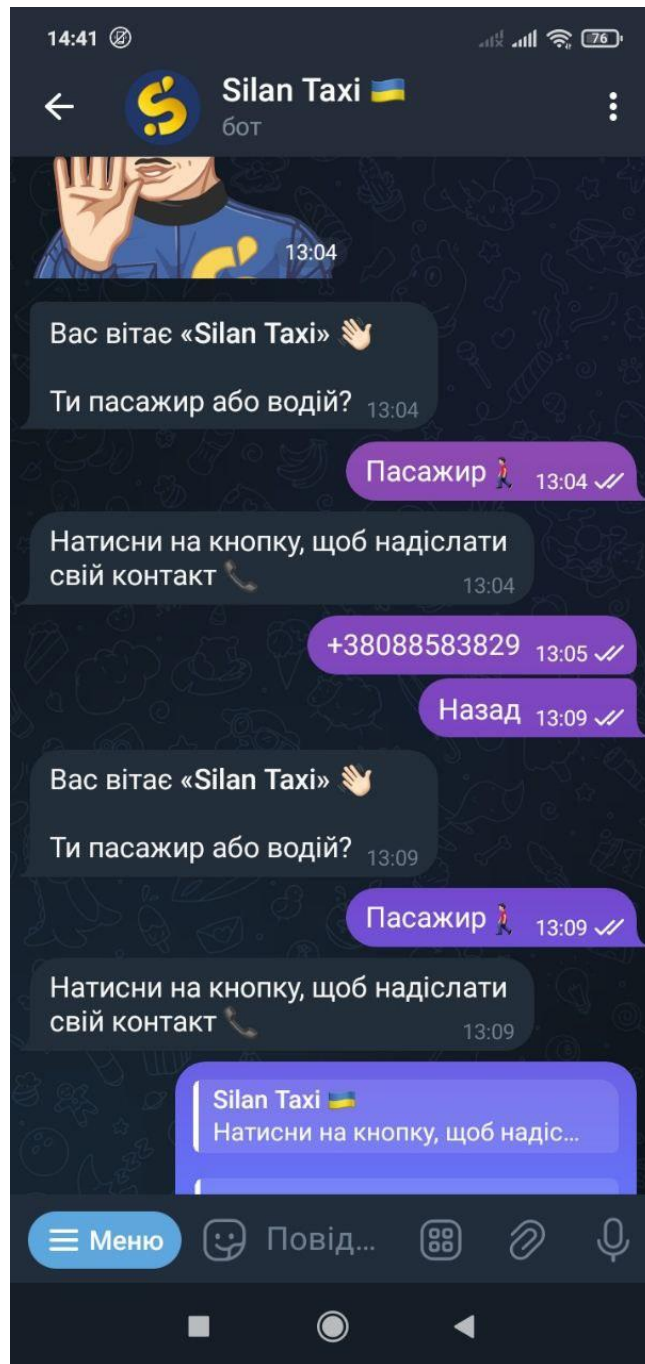


Рисунок 4.9 – Реєстрація пасажирів

Крім того, користувач має можливість переглядати особистий кабінет, де доступні такі розділи:

- мої дані: користувач може переглянути свій номер телефону та інші дані, які він вводив при реєстрації;
- знижка: інформація про доступні знижки або програми лояльності, які користувач може використовувати;

- підтримка: можливість звернутися за підтримкою або допомогою, якщо виникли проблеми чи питання;
- статистика: відображення кількості здійснених поїздок, їх вартості та деталей кожної поїздки для відстеження та аналізу.

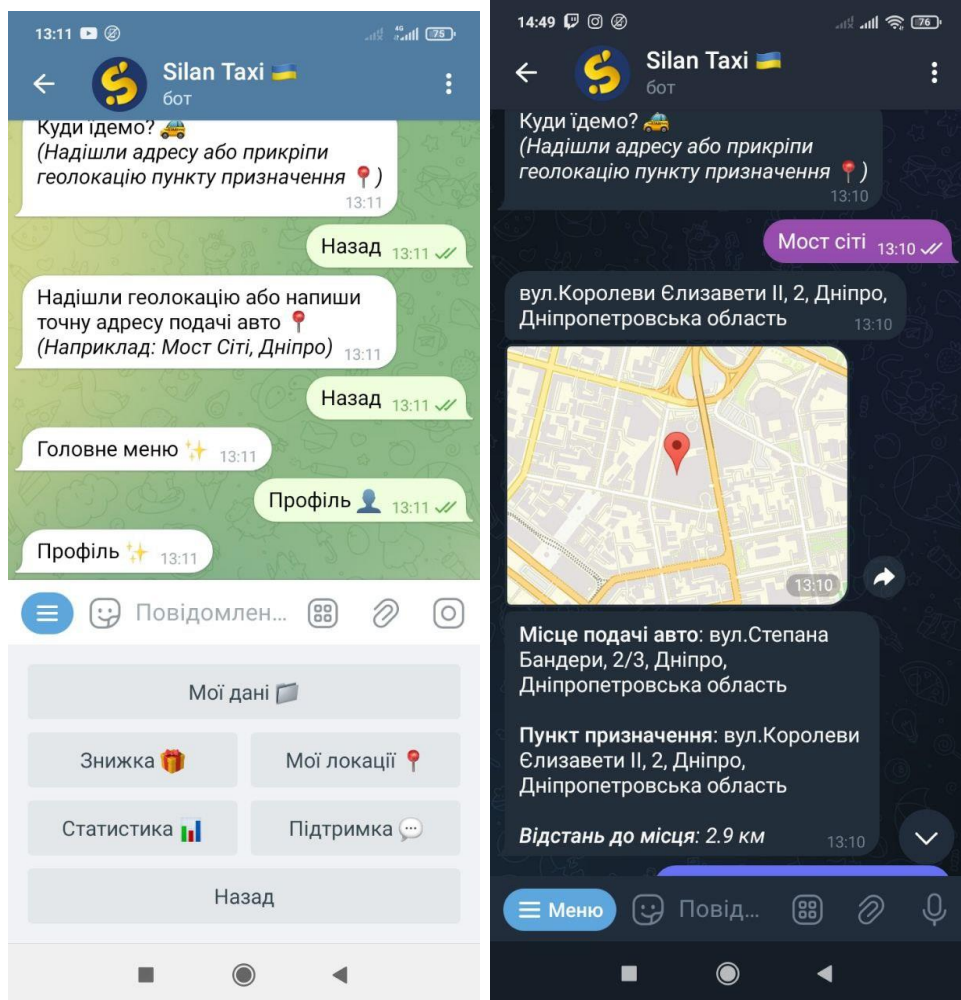


Рисунок 4.10 – Особистий профіль користувача

Ці функції дозволяють користувачеві керувати своїми особистими даними, отримувати інформацію про додаткові можливості та зручно відслідковувати статистику своїх поїздок.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра на тему "Комп'ютерна система компанії «RokSave» з детальною розробкою функціоналу та сервісів інтерактивного телеграм-боту" було виконано завдання, які сприяли створенню ефективної та надійної комп'ютерної системи для компанії.

У рамках цієї роботи був створений Telegram-бот, призначений для автоматизації процесів у компанії «RokSave» та надання зручного інтерфейсу для користувачів. Бот забезпечує користувачам, таким як водії та пасажирів, доступ до низки важливих функцій. Для водіїв реалізовані можливості реєстрації, вибору категорії авто, завантаження фото транспортного засобу та отримання підтвердження від адміністрації. Для пасажирів передбачено функції виклику таксі, поділу геолокацією, введення адреси посадки та висадки, а також вибору категорії автомобіля. Крім того, користувачі мають доступ до особистого кабінету, де можуть переглядати свої дані, отримувати знижки, звертатися за підтримкою та відстежувати статистику поїздок. Реалізовані функції дозволяють користувачам зручно взаємодіяти з системою, надаючи швидкий доступ до необхідних послуг та інформації.

Одним з ключових досягнень даної роботи стало створення та налаштування комп'ютерної мережі для забезпечення стабільної роботи бота та інших сервісів компанії. Мережа забезпечує надійну комунікацію між різними компонентами системи, гарантуючи безперебійне функціонування сервісів та захист даних користувачів. Інтеграція мережевої інфраструктури з хмарними платформами, такими як Heroku, дозволила досягти високої доступності та масштабованості системи. Це, в свою чергу, забезпечує надійність та ефективність роботи бота, роблячи його доступним для користувачів у будь-який час.

Впровадження інтерактивного бота значно покращило операційну ефективність компанії «RokSave». Завдяки автоматизації процесів, пов'язаних з обробкою запитів клієнтів, знизилася навантаження на службу підтримки, а час обробки замовлень значно скоротився. Це дозволило підвищити швидкість та якість

обслуговування клієнтів, зменшивши кількість помилок та забезпечивши більш високий рівень задоволеності користувачів. Автоматизація управління даними також сприяла підвищенню ефективності внутрішніх процесів компанії.

ПЕРЕЛІК ПОСИЛАНЬ

1. Компанія «RokSave» [Електронний ресурс] – Режим доступу до ресурсу: https://youcontrol.com.ua/ua/catalog/company_details/38287055/
2. Комп'ютерні мережі. Методичні рекомендації до виконання лабораторних робіт студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія: у 2 ч. / Л.І. Цвіркун, Я.В. Панферова ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2018. – Ч. 2. – 39 с.
3. Human: Підтримка корпоративних систем [Електронний ресурс] – Режим доступу до ресурсу: id.human.ua(дата звернення 2.06.2024р.).
4. OSPF [Електронний ресурс] – Режим доступу до ресурсу: <https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest-path-first-ospf>
5. NAT [Електронний ресурс] – Режим доступу до ресурсу: <https://voipzeker.nl/alles-over-bellen-met-voip/nat>
6. AAA. FreeRADIUS Documentation – NetworkRADIUS)? [Електронний ресурс] – Режим доступу до ресурсу: <https://networkradius.com/doc/current/concepts/introduction/AAA.html>
7. The Complete Guide to UML Diagram Types with Examples [Електронний ресурс] – Режим доступу до ресурсу: <http://creately.com/blog/diagrams/uml->
8. Widman J. The Most Popular Programming Languages of 2019 [Електронний ресурс] / Jake Widman // New Relic. – 2019. – Режим доступу до ресурсу: <https://blog.newrelic.com/technology/most-popular-programminglanguages-of-2019/>.

ДОДАТОК А

Текст програми налаштувань мережі комп'ютерної системи

Конфігураційний файл шлюзового маршрутизатора Sylakov_4

```
!  
version 15.4  
no service timestamps log datetime msec  
no service timestamps debug datetime msec  
service password-encryption  
!  
hostname Sylakov_Router_4  
!  
!  
!  
enable password 7 0822404F1A0A  
!  
!  
ip dhcp excluded-address 10.25.106.1 10.25.106.4  
!  
ip dhcp pool LAN5  
network 10.25.106.0 255.255.255.128  
default-router 10.25.106.1  
dns-server 10.25.106.2  
!  
!  
aaa new-model  
!  
aaa authentication login default local  
aaa authentication login radius-auth group radius local  
!
```

```
!  
!  
!  
!  
!  
!  
ip cef  
no ipv6 cef  
!  
!  
!  
username 12320_Sylakov password 7 082048430017061E010803  
  
!  
!  
!  
!  
!  
!  
!  
!  
!  
ip domain-name Sylakovk_Router_4  
!  
!  
spanning-tree mode pvst  
!  
!  
!  
!  
!
```

```
!  
interface GigabitEthernet0/0/0  
no ip address  
duplex auto  
speed auto  
shutdown  
!  
interface GigabitEthernet0/0/1  
ip address 10.25.106.1 255.255.255.128  
duplex auto  
speed auto  
!  
interface GigabitEthernet0/0/2  
no ip address  
duplex auto  
speed auto  
shutdown  
!  
interface Serial0/1/0  
bandwidth 128  
no ip address  
ip ospf cost 7500  
clock rate 128000  
shutdown  
!  
interface Serial0/1/1  
bandwidth 128  
ip address 10.8.7.1 255.255.255.0  
ip ospf cost 7500  
!
```

```
interface Serial0/2/0
  bandwidth 128
  ip address 10.9.7.1 255.255.255.0
  ip ospf cost 7500
!
interface Serial0/2/1
  bandwidth 128
  ip address 10.3.7.2 255.255.255.0
  ip ospf cost 7500
  clock rate 128000
!
interface Vlan1
  no ip address
  shutdown
!
router ospf 4
  log-adjacency-changes
  passive-interface GigabitEthernet0/0/1
  network 10.0.0.0 0.255.255.255 area 0
  network 172.24.64.128 0.0.0.127 area 0
  network 10.25.106.0 0.0.0.127 area 0
!
ip classless
!
ip flow-export version 9
!
!
!
no cdp run
!
```

```
banner motd #Welcome to the Sylakov_Router_4#
!
radius server radius-server
  address ipv4 10.25.106.133 auth-port 1645
  key radius123

!
!
!
line con 0
  password 7 0822455D0A16
  login authentication radius-auth
!
line aux 0
!
line vty 0 4
  password 7 0822455D0A16
  login authentication default
  transport input ssh
line vty 5 15
  password 7 0822455D0A16
  login authentication default
  transport input ssh
!
!
!
end
```


ДОДАТОК Б

Текст програми чат-боту таксі

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
РОБКА ЧАТ-БОТУ ТАКСІ

Текст програми

804.0207070743.24013-01 12 01

Листів 9

2024

АНОТАЦІЯ

Даний додаток містить програмний код для реалізації телеграм-бота, який забезпечує інтерактивний інтерфейс для користувачів. Телеграм-бот розроблений з метою оптимізації процесу замовлення та управління перевезеннями, а також покращення комунікації з клієнтами.

Основна функціональність бот включає в себе можливість користувачам через простий і зручний інтерфейс у месенджері Telegram розміщувати замовлення на перевезення. Користувачі можуть вводити інформацію про місце призначення, вибирати тип транспорту та додаткові побажання, що дозволяє зменшити час на оформлення замовлення і підвищити точність обробки запитів.

3MICT

passenger_search.py4

Текст програми

```

import asyncio

from aiogram.dispatcher import FSMContext

from aiogram.types import Message, CallbackQuery,
InlineKeyboardMarkup, InlineKeyboardButton, ReplyKeyboardRemove

from bot import dp, bot

from config import dbname, user, password, host, port

from data import Form, Passenger_menu, Passenger_order, Driver_menu,
car_category_dict2

from functions import get_drivers, choose_driver, check_pass_rating,
check_driver_rating, \
    get_categories_with_prices_keyboard, get_geolocation_keyboard,
insert_into_passenger_demand, get_category_full_price

from keyboards import passenger_menu_keyboard,
cancel_request_keyboard, driver_main_menu_keyboard

from middleware import rate_limit

from scheduler import scheduler

@dp.message_handler(state=Form.type_of_car, text_startswith=["Стандарт
🚗", "Комфорт 🚗", "Мікроавтобус 🚐"])
@rate_limit(limit=2)
async def set_passenger_order(message: Message, state: FSMContext):
    data = await state.get_data()
    pass_location_text = data.get("pass_location_text")
    pass_longitude = data.get("pass_longitude")
    pass_latitude = data.get("pass_latitude")
    destination_longitude = data.get("destination_longitude")
    destination_latitude = data.get("destination_latitude")

```

```

destination_location_text = data.get("destination_location_text")
dist = data.get("dist")
discount_type = data.get("discount_type")

message_array = message.text.split()
car_category      =      car_category_dict2[f"{message_array[0]}
{message_array[1]}"]
old_price = int(message_array[3])

price = await get_category_full_price(state, car_category, pass_latitude,
pass_longitude, message.from_user.id)

if price != old_price:
    await insert_into_passenger_demand(message.from_user.id, state)
    categories_with_prices_keyboard      =      await
get_categories_with_prices_keyboard(state, message, pass_latitude,
pass_longitude)
    await      message.answer("<b>Ціна      змінилась      ⚡</b>",
reply_markup=categories_with_prices_keyboard)
else:
    await      message.answer("Пошук      водія      🔍",
reply_markup=ReplyKeyboardRemove())

drivers = await get_drivers(pass_latitude, pass_longitude, car_category,
dist)

await choose_driver(message, scheduler, drivers, pass_location_text,
destination_location_text, pass_latitude,
pass_longitude, destination_latitude, destination_longitude,
dist, price, discount_type,

```

```
car_category, state)
```

```
@dp.message_handler(state=Form.type_of_car, text="Назад")
async def type_car_back(message: Message):
    geolocation_keyboard = await get_geolocation_keyboard(message, 2)
    await message.answer("Куди їдемо? 🚗\n"
        "<i>(Надішли адресу або прикріпи геолокацію пункту
призначення 📍)</i>",
        reply_markup=geolocation_keyboard)
    await Form.destination.set()
```

```
@dp.callback_query_handler(state=Passenger_order.functions,
text="cancel_order")
async def passenger_cancel_order(call: CallbackQuery):
    await call.answer("")
    await call.message.answer("<b>За скасування поїздки знижується твій
рейтинг !!</b>\n\n"
        "Скасувати поїздки?",
        reply_markup=cancel_request_keyboard)
    await Passenger_order.cancel_request.set()
```

```
@dp.callback_query_handler(state=Passenger_order.cancel_request,
text="cancel_no")
async def passenger_cancel_order_no(call: CallbackQuery):
    try:
        await call.message.delete()
```

```

except Exception as e:
    print(e)
await Passenger_order.functions.set()

@dp.callback_query_handler(state=Passenger_order.cancel_request,
text="cancel_yes")
async def passenger_cancel_order_yes(call: CallbackQuery, state:
FSMContext):
    try:
        await call.message.delete()
    except Exception as e:
        print(e)

    await state.finish()

    conn = await asyncpg.connect(database=dbname, user=user,
password=password, host=host, port=port)
    driver_id = await conn.fetchrow("SELECT driver_id FROM orders
WHERE pass_id = $1", call.from_user.id)
    await conn.execute("DELETE FROM orders WHERE pass_id = $1",
call.from_user.id)
    pass_statistic = await conn.fetchrow("SELECT rating, count_reviews
FROM passenger_statistic WHERE id = $1",
call.from_user.id)

    try:
        await bot.send_message(chat_id=driver_id[0], text="Замовлення
скасовано пасажиром 😊",
reply_markup=driver_main_menu_keyboard)

```



```

        await dp.storage.set_state(chat=driver_id[0],
state=Driver_menu.main_menu_choice)
    except Exception as e:
        print(e)

    rating = pass_statistic[0]
    count_reviews = pass_statistic[1]
    cancel_review = 40
    rating = round((rating * count_reviews + cancel_review) / (count_reviews
+ 1), 1)

    check_rating_result = await check_pass_rating(conn, call.from_user.id,
rating)
    if check_rating_result == 1 or check_rating_result == 2:
        await conn.execute("UPDATE passenger_statistic SET rating = $1,
count_reviews = $2 WHERE id = $3", rating,
count_reviews + 1, call.from_user.id)
        await call.message.answer("Замовлення скасоване 🙌",
reply_markup=passenger_menu_keyboard)
        if check_rating_result == 2:
            await call.message.answer("<b>Увага! Ваш рейтинг дуже
низький!</b>\n\n"
"Намагайся не знижувати свій рейтинг, щоб
уникнути блокування акаунта 😞")
            await Passenger_menu.choose.set()
        elif check_rating_result == 0:
            await call.message.answer("Твій акаунт заблоковано за дуже низький
рейтинг 😞")
            await conn.close()

```

```

@dp.callback_query_handler(state=Passenger_order.functions,
text="card_order")
async def passenger_card_order(call: CallbackQuery):
    await call.answer()
    conn = await asyncpg.connect(database=dbname, user=user,
password=password, host=host, port=port)
    order_data = await conn.fetchrow("SELECT driver_id, price FROM orders
WHERE pass_id = $1", call.from_user.id)
    driver_id = order_data[0]
    price = order_data[1]
    card_data = await conn.fetchrow("SELECT card_number, card_link
FROM drivers WHERE id = $1", driver_id)
    card_number = card_data[0]
    card_number = f"{card_number[:4]} {card_number[4:8]}
{card_number[8:12]} {card_number[12:]}"

    card_link = card_data[1]
    card_text = f"<b>Банківські реквізити водія 

```

```

        ]
    ]
)

```

```

await call.message.answer(card_text, reply_markup=card_keyboard)

```

```

@dp.callback_query_handler(state=Passenger_order.review,
text_startswith="review_star")
async def edit_review(call: CallbackQuery, state: FSMContext):
    await call.answer("")
    stars = int(call.data[-1])
    edit_review_keyboard = InlineKeyboardMarkup(row_width=5)
    buttons = []
    for star in range(1, 6):
        if star <= stars:
            button = InlineKeyboardButton(text="★",
callback_data=f"review_star{star}")
        else:
            button = InlineKeyboardButton(text="☆",
callback_data=f"review_star{star}")
        buttons.append(button)
    edit_review_keyboard.add(*buttons)
    button = InlineKeyboardButton(text="Підтвердити",
callback_data="accept_review")
    edit_review_keyboard.add(button)

try:

```

```

        await
call.message.edit_reply_markup(reply_markup=edit_review_keyboard)
    except:
        pass
    await state.update_data(stars=stars)

```

```

@dp.callback_query_handler(state=Passenger_order.review,
text="accept_review")

```

```

async def accept_review(call: CallbackQuery, state: FSMContext):

```

```

    data = await state.get_data()

```

```

    stars = data.get("stars")

```

```

    driver_id = data.get("driver_order_id")

```

```

    if stars is None:

```

```

        await call.answer("Ти не оцінив водія 😊", show_alert=True)

```

```

    else:

```

```

        try:

```

```

            await call.message.delete()

```

```

        except:

```

```

            pass

```

```

    conn = await asyncpg.connect(database=dbname, user=user,
password=password, host=host, port=port)

```

```

    pass_statistic = await conn.fetchrow("SELECT rating, count_reviews
FROM driver_statistic WHERE id = $1",
                                        driver_id)

```

```

    rating = pass_statistic[0]

```

```

    count_reviews = pass_statistic[1]

```

```

    review = stars * 20

```

```
rating = round((rating * count_reviews + review) / (count_reviews + 1),
```

1)

```
check_rating_result = await check_driver_rating(conn, driver_id, rating)
```

```
if check_rating_result == 1 or check_rating_result == 2:
```

```
    await conn.execute("UPDATE driver_statistic SET rating = $1,
count_reviews = $2 WHERE id = $3", rating,
```

```
        count_reviews + 1, driver_id)
```

```
    try:
```

```
        await bot.send_message(chat_id=driver_id, text=f"Тобі поставили
<b>{stars} ★ </b>за поїздки")
```

```
    except:
```

```
        pass
```

```
if check_rating_result == 2:
```

```
    try:
```

```
        await bot.send_message(chat_id=driver_id, text="<b>Увага!
Твій рейтинг дуже низький!</b>\n\n"
```

```
        "Намагайся не знижувати свій
```

```
рейтинг, "
```

```
        "щоб уникнути блокування
```

```
акаунта 😞")
```

```
    except:
```

```
        pass
```

```
elif check_rating_result == 0:
```

```
    try:
```

```
        await bot.send_message(chat_id=driver_id, text="Твій акаунт
заблоковано за дуже низький рейтинг 😞")
```

```
    except:
```

```
        pass
```

```
await conn.close()
await state.finish()
await call.message.answer("Дякуємо за відгук 👍",
reply_markup=passenger_menu_keyboard)
await Passenger_menu.choose.set()
```