

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 Богосян Олег Альбертович
(ПІБ)

академічної групи 123-20-1
(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 Адміністрації Амур-Нижньодніпровського району м. Дніпро з детальним опрацюванням побудови та налаштування корпоративної мережі»
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Цвіркун Л.І.			
спеціальної частини	проф. Цвіркун Л.І.			
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)

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

ЗАВДАННЯ**на кваліфікаційну роботу ступеня бакалавр**

студента Богосян О.А. академічної групи 123-20-1
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

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

на тему: «Комп'ютерна система Адміністрації Амур-Нижньодніпровського району м. Дніпро з детальним опрацюванням побудови та налаштування корпоративної мережі» 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)

Цвіркун Л.І.
(прізвище, ініціали)

Дата видачі 12.02.2024

Дата подання до екзаменаційної комісії 03.06.2024 р.

Прийнято до виконання _____
(підпис студента)

Богосян О.А.
(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 69 с., 33 рис., 5 табл., 2 дод., 9 джерел.

КОМП'ЮТЕРНА СИСТЕМА, МЕРЕЖА, WAN, LAN, CISCO, КОРПОРАТИВНА МЕРЕЖА

Об'єкт розробки – комп'ютерна система Адміністрації Амур-Нижньодніпровського району м. Дніпро з детальним опрацюванням побудови та налаштування корпоративної мережі та створення телеграм боту.

Мета роботи – налаштування та розробка корпоративної мережі для Адміністрації Амур-Нижньодніпровського району м. Дніпро, що ключає в себе інтеграцію Telegram-боту для покращення взаємодії з громадянами та забезпечення швидкого доступу до адміністративних послуг.

В першому розділі було розглянуто галузь розробки системи, моделі за допомогою яких було творено систему, визначено завдання та обрані параметри для дослідження різних компонентів системи.

У другому розділі були визначені усі необхідні технічні вимоги розробки системи, було розроблено топологічну схему ААНР, обрані та обгрунтовані використані технічні засоби, розраховано інтенсивність вихідного трафіку.

В третьому розділі було розраховано схему адресації мережі, маршрутизацію, виконано налаштування та перевірка системи. Розроблена топологічна схема за допомогою програмного забезпечення Cisco.

В четвертому розділі було розроблено телеграм бота, для покращення та зручності робочого процесу в ААНР.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Стан питання та постановка завдання.....	8
1.1 Стисла характеристика галузі розробки програмного забезпечення	8
1.2 Характеристика Адміністрації Амур-Нижньодніпровського району м. Дніпро	8
1.3 Принципи, технічні способи та математичні методи інформаційного забезпечення підприємства	9
1.4 Організаційна структура ААНР	10
1.5 Топологічне розміщення структурних підрозділів	11
1.6 Постановка завдання.....	12
2 Розробка апаратної частини комп'ютерної або кіберфізичної системи	15
2.1 Технічні вимоги до КС.....	15
2.1.1 Вимоги до системи в цілому.....	15
2.1.1.1 Вимоги до структури і функціонуванню системи	15
2.1.1.2 Вимоги до способів і засобів зв'язку між компонентами комп'ютерної системи	16
2.1.1.3 Вимоги до режимів функціонування системи	17
2.1.1.4 Перспективи розвитку, модернізації системи	17
Рисунок 2.1 – Топологія КМ ААНР	19
2.1.1.5 Вимоги до діагностування системи.....	19
2.1.2 Вимоги до системи в цілому.....	19
2.1.2.1 Вимоги до технічного забезпечення	19
2.1.2.2 Вимоги до програмного забезпечення	20
2.2 Розробка апаратної частини комп'ютерної системи	21
2.2.1 Розробка структурної схеми мережі.....	21
2.2.2 Розробка специфікації апаратних засобів КС	23
2.2.2 Розрахунок інтенсивності трафіку найбільшої локальної мережі підприємства	25
3 Розробка корпоративної мережі підприємства.....	27
3.1 Розрахунок схеми адресації мережі	27
3.2 Розрахунок схеми адресації пристроїв	29

3.3 Розробка топологічної мережі	31
На основі отриманих даних, створимо модель мережі компанії у середовищі Cisco Packet Tracer. Логічна топологія, представлена на рисунку 3.1.	31
3.4 Налаштування конфігурації та роботи КС	31
3.4.1 Базове налаштування конфігурації пристроїв	31
3.4.2 Налаштування маршрутизації	33
3.4.3 Налаштування служби AAA.....	34
3.4.3 Налаштування служби Інтернет	35
3.4.4 Налаштування служби VLAN	37
3.4.5 Перевірка роботи комп'ютерної системи	40
4 Розробка компонента системи.....	45
4.1 Розробка телеграм боту для ААНР	45
4.2 Бібліотека SQLite	45
4.2 Розробка програмної частини.....	46
4.2.1 Імпорт необхідних бібліотек	46
4.2.2 Створення токену	48
4.3 Перевірка роботи розробленого телеграм боту.....	53
Висновки.....	56
Перелік посилань	58
Додаток А – налаштування шлюзового маршрутизатора.....	59
Додаток Б – розробка телеграм боту.....	644

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КС – комп'ютерна система.

ААНР - Адміністрація Амур-Нижньодніпровського району м. Дніпр

ПК – персональний комп'ютер. Ethernet – протокол кабельних комп'ютерних мереж.

VPN – віртуальна приватна мережа.

LAN – локальна (обчислювальна) мережа із групи комп'ютерів та/або інших пристроїв.

ПЗ – програмне забезпечення

ВСТУП

У сучасному світі діджиталізація, комп'ютерні системи відіграють ключову роль у забезпеченні ефективної роботи адміністративних структур. Комп'ютерні системи та корпоративні мережі стали невід'ємною частиною щоденної діяльності будь-якої організації, зокрема адміністративних органів місцевого самоврядування. Вони дозволяють автоматизувати процеси, підвищувати продуктивність праці, забезпечувати надійний обмін інформацією та захищати дані від несанкціонованого доступу.

Адміністрація Амур-Нижньодніпровського району міста Дніпро не є виключенням. З метою покращення управлінських процесів та підвищення якості обслуговування громадян, необхідно створити сучасну та надійну комп'ютерну систему, яка включає в себе ефективно побудовану та налаштовану корпоративну мережу.

У даній роботі буде розглянуто процес створення комп'ютерної системи для Адміністрації Амур-Нижньодніпровського району м. Дніпро. Особлива увага буде приділена побудові та налаштуванню корпоративної мережі, яка забезпечить безперебійний та безпечний обмін даними між різними підрозділами адміністрації. Також будуть розглянуті питання вибору відповідного апаратного та програмного забезпечення, методи захисту інформації та оптимізації мережевих ресурсів.

Метою цієї роботи є створення надійної, безпечної та ефективної комп'ютерної системи, яка сприятиме покращенню функціонування адміністративних структур та забезпечить високий рівень обслуговування мешканців району.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі розробки програмного забезпечення

Галузь розробки програмного забезпечення для системи адміністрування Амур-Нижньодніпровського району м. Дніпро відображає сучасні тенденції у сфері ІТ, враховуючи розвиток технологій та потреби користувачів. Розробка програмного забезпечення в цій галузі відбувається з використанням передових методів розробки, що дозволяє швидко реагувати на зміни та забезпечувати постійну відповідність до потреб користувачів.

Ключовими пріоритетами в розробці програмного забезпечення для системи адміністрування району є забезпечення безпеки та конфіденційності даних, зручність та доступність користувачам, ефективність роботи системи та забезпечення масштабованості для врахування потенційного збільшення обсягу роботи.

Високий рівень професіоналізму та експертизи в галузі розробки програмного забезпечення дозволяє створювати продукти, що відповідають найвищим стандартам якості та надійності. Крім того, активний діалог з користувачами та залучення їхнього фідбеку у процесі розробки дозволяє створювати програмне забезпечення, яке дійсно відповідає їхнім потребам і очікуванням.

Розробка програмного забезпечення для системи адміністрування району є складним та відповідальним процесом, який вимагає тісного співробітництва між розробниками, адміністрацією та користувачами для досягнення успішних результатів [1].

1.2 Характеристика Адміністрації Амур-Нижньодніпровського району м. Дніпро

Адміністрація Амур-Нижньодніпровського району м. Дніпро використовуватиме комп'ютерну систему для вирішення різноманітних завдань:

– комп'ютерна система дозволить зберігати великі обсяги даних, такі як інформація про мешканців, бюджетні дані, документація про проекти та інше. Це

дасть змогу адміністрації легко отримувати доступ до необхідної інформації, швидко обробляти її та аналізувати;

– КС забезпечить засоби комунікації між різними відділами та співробітниками адміністрації. Це можуть бути електронна пошта, системи миттєвих повідомлень, спільні ресурси для спільної роботи над документами тощо. Це дозволить покращити координацію робіт та збільшити ефективність командної роботи;

– КС дозволить автоматизувати багато рутинних операцій, таких як ведення обліку, створення звітів, обробка запитів тощо. Це дозволить зменшити людські помилки, підвищити швидкість роботи та звільнити час для виконання більш важливих завдань;

– комп'ютерна система дозволить забезпечити захист конфіденційної інформації та здійснювати контроль доступу до даних. Це включає в себе встановлення антивірусного програмного забезпечення, налаштування прав доступу до файлів та документів, а також резервне копіювання важливих даних для запобігання їх втраті.

Отже, комп'ютерна система є невід'ємною частиною роботи адміністрації та допомагає забезпечити ефективне управління та розвиток регіону [1].

1.3 Принципи, технічні способи та математичні методи інформаційного забезпечення підприємства

Принципи, технічні способи та математичні методи інформаційного забезпечення підприємства для системи адміністрування Амур-Нижньодніпровського району м. Дніпро орієнтовані на забезпечення безпеки, надійності та ефективності обміну та обробки інформації. Це включає в себе застосування таких принципів, як конфіденційність, цілісність, доступність, аутентичність та неденесення збитків. Технічні засоби включають в себе шифрування даних для забезпечення конфіденційності, контроль доступу для забезпечення

аутифікації та авторизації користувачів, а також застосування систем захисту мережі та резервне копіювання даних для забезпечення доступності та цілісності інформації. Математичні методи, такі як криптографія та хеш-функції, використовуються для розробки безпечних криптографічних алгоритмів та перевірки цілісності даних [1].

1.4 Організаційна структура ААНР

Структура робітників в адміністрації Амур-Нижньодніпровського району м. Дніпро може включати наступні категорії працівників (рис.1.1):

Керівництво: голова адміністрації району; заступник голови адміністрації з питань соціального розвитку; заступник голови адміністрації з питань економічного розвитку; члени виконавчого комітету адміністрації.

Відділи та служби:

- відділ земельних ресурсів;
- відділ містобудування та архітектури;
- відділ економічного розвитку;
- відділ соціального захисту населення;
- відділ культури та туризму;
- відділ освіти;
- відділ охорони здоров'я;
- відділ молоді та спорту;
- відділ інфраструктури та благоустрою;
- служба протипожежної безпеки;
- служба цивільного захисту;
- служба у справах дітей та сім'ї.

Спеціалісти:

- юристи;
- фінансисти та бухгалтери;
- економісти;
- архітектори;

- інженери-містобудівники;
- соціальні працівники;
- педагоги;
- лікарі та медичні працівники;
- спеціалісти з реклами та зв'язків з громадськістю;
- спортивні тренери та інструктори;
- інші фахівці з різних сфер.

Схема організаційної структури ДП «УАМЦ» (з 01.12.2021р.)

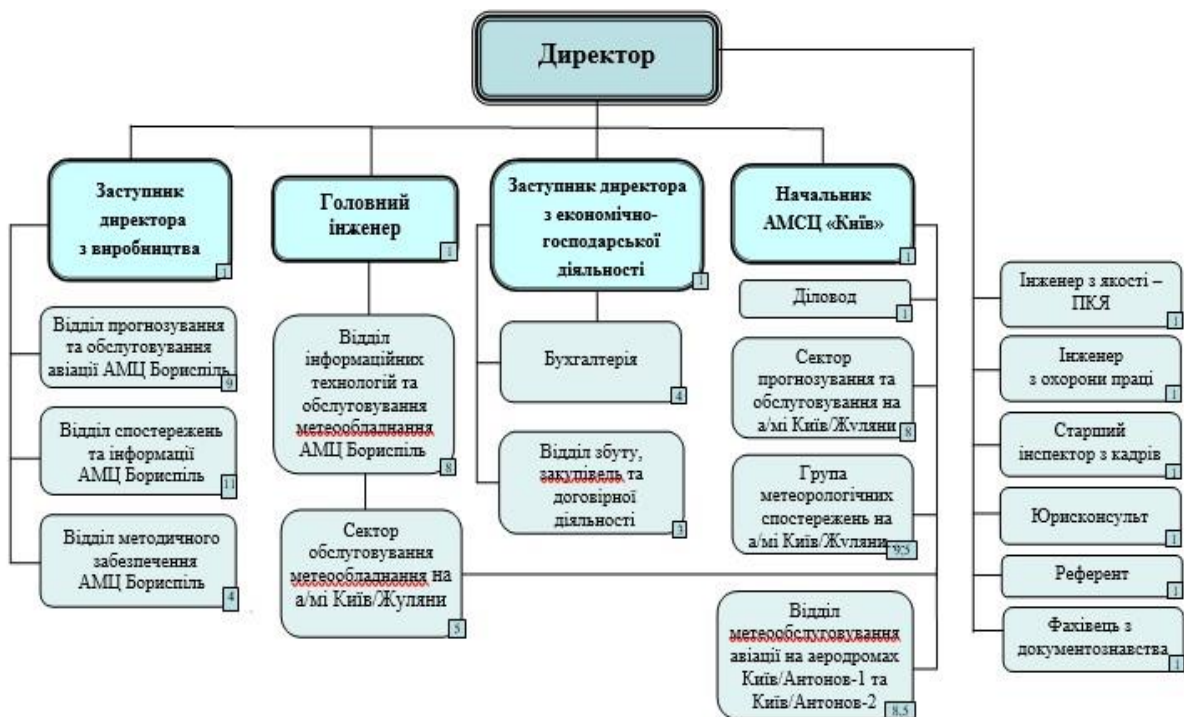


Рисунок 1.1 – Організаційна структура ААНР [1]

Структура на рис.1.1 може змінюватися в залежності від конкретних потреб та організаційної моделі адміністрації району.

1.5 Топологічне розміщення структурних підрозділів

ААНР розташована за адресою проспект Мануйлівський, 31, Дніпро, Дніпропетровська область, 49000. Будівля складається з 4-ох поверхів на яких розташовано 65 кабінетів (рис.1.2) [1].

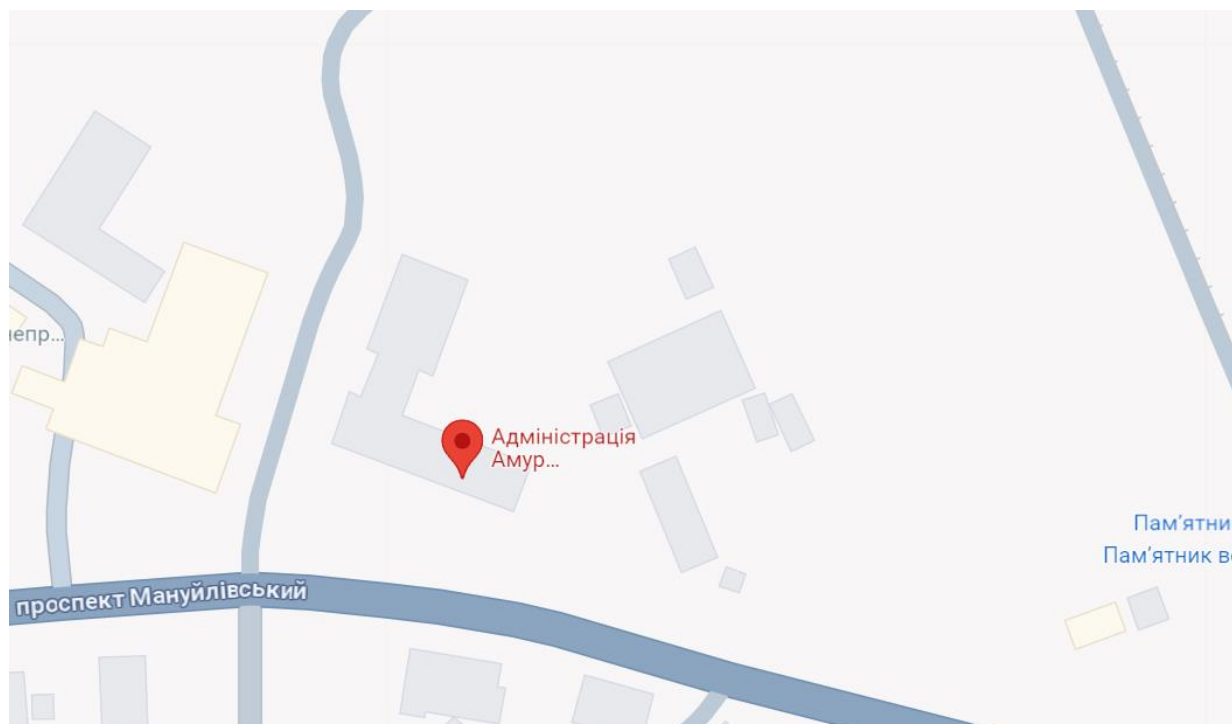


Рисунок 1.2 – Топографічна схема розташування ААНР

1.6 Постановка завдання

Завданням кваліфікаційної роботи є розробка «Комп'ютерної системи Адміністрації Амур-Нижньодніпровського району м. Дніпро з детальним опрацюванням побудови та налаштування корпоративної мережі».

Для вирішення цього завдання необхідно виконати низку конкретних дій.

По-перше, потрібно ознайомитися зі структурою організації. Це включає аналіз існуючої організаційної структури адміністрації та визначення потреб і вимог до комп'ютерної системи з урахуванням специфіки роботи підрозділів.

По-друге, слід розробити технічні вимоги для комп'ютерної системи (КС). На цьому етапі визначаються основні функціональні та нефункціональні вимоги до КС, створюється список необхідних технічних характеристик та параметрів.

По-третє, необхідно обрати комплекс технічних засобів для КС. Підбір відповідного апаратного забезпечення здійснюється з урахуванням технічних вимог. Після оцінки можливих варіантів вибирається оптимальне рішення.

Далі потрібно обрати топологію корпоративної мережі. На основі аналізу можливих топологій мережі вибирається найбільш підходяща, з урахуванням майбутнього розвитку мережі.

Наступним кроком є розрахунок налаштувань для заданої топології мережі. Це включає вибір відповідних інтерфейсів каналів зв'язку та протоколів обміну даними, що забезпечать оптимальну роботу мережі.

Після цього створюється топологічна (логічна) схема комп'ютерної системи. Розробка схеми відображає логічну структуру мережі та взаємозв'язки між різними її елементами.

Далі здійснюється розрахунок налаштувань маршрутизації комп'ютерної мережі. Це включає визначення оптимальних маршрутів передачі даних та налаштування протоколів маршрутизації для забезпечення ефективного обміну інформацією.

Потім відбувається налаштування корпоративної мережі та її тестування. Реалізуються налаштування мережі відповідно до розробленої схеми, після чого проводиться тестування для перевірки працездатності та стабільності мережі.

На наступному етапі здійснюється підбір обладнання та програмного забезпечення. Вибирається відповідне обладнання для реалізації КС та підбирається програмне забезпечення, необхідне для роботи мережі та виконання завдань адміністрації.

Останнім етапом є розробка телеграм-бота для Адміністрації Амур-Нижньодніпровського району м. Дніпро. Створюється телеграм-бот для поліпшення взаємодії з громадянами та інтегрується з існуючою комп'ютерною системою для забезпечення автоматизації певних процесів.

Виконання вищеописаних завдань дозволить створити надійну, ефективну та безпечну комп'ютерну систему, що відповідатиме сучасним вимогам і сприятиме

підвищенню продуктивності роботи Адміністрації Амур-Нижньодніпровського району м. Дніпро.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ АБО КІБЕРФІЗИЧНОЇ СИСТЕМИ

2.1 Технічні вимоги до КС

2.1.1 Вимоги до системи в цілому

2.1.1.1 Вимоги до структури і функціонуванню системи

Об'єктом розробки є комп'ютерна система (надалі КС) для Адміністрації Амур-Нижньодніпровського району м. Дніпро (надалі просто ААНД). Основним призначенням КС є забезпечення ефективного функціонування адміністративних процесів, забезпечення надійного обміну інформацією між підрозділами адміністрації, а також надання громадянам можливості отримання інформації та взаємодії з адміністрацією через телеграм-бота.

КС має складатися з наступних функціональних блоків [2]:

- мережа адміністративних підрозділів, дана частина КС виконує функцію забезпечення роботи адміністративних підрозділів, включаючи комп'ютери, принтери, сканери та інше офісне обладнання. Також ця мережа повинна включати бездротові точки доступу до мережі Інтернет для використання співробітниками адміністрації;

- мережа обслуговування та моніторингу - ця частина КС має складатися з комп'ютерів, призначених для використання технічним персоналом. З цього функціонального блоку має здійснюватись моніторинг роботи всієї мережі та керування її налаштуваннями. Також через цей функціональний блок має здійснюватись обмін даними з інтернет-провайдером;

- мережа відділу зв'язків з громадськістю має складатися з обладнаних комп'ютерами робочих місць, призначених для фахівців, що займаються аналізом запитів громадян, розробкою інновацій та підтримкою телеграм-бота.

Підсистема для моніторингу та керування мережею на основі мережевого протоколу SNMP (Simple Network Management Protocol). Підсистема має складатися з трьох основних компонентів:

- агенти моніторингу встановлюються на кожному пристрої мережі та збирають дані про стан пристрою та мережі;
- сервер моніторингу збирає дані від агентів моніторингу та будує карту мережі, що дозволяє адміністратору бачити стан мережі в режимі реального часу;
- інтерфейс адміністратора забезпечує керування мережею та налаштування параметрів моніторингу;
- телеграм-бот для взаємодії з громадянами має забезпечувати швидкий та зручний доступ громадян до інформації про діяльність адміністрації, прийом звернень та запитів, а також автоматизацію деяких процесів, таких як реєстрація на прийом, отримання довідок тощо.

Виконання цих пунктів дозволить створити надійну, ефективну та безпечну комп'ютерну систему, що відповідатиме сучасним вимогам і сприятиме підвищенню продуктивності роботи Адміністрації Амур-Нижньодніпровського району м. Дніпро.

2.1.1.2 Вимоги до способів і засобів зв'язку між компонентами комп'ютерної системи

Для створення каналів зв'язку в КС між мережевими пристроями Адміністрації Амур-Нижньодніпровського району м. Дніпро необхідно використовувати кабелі вита пара категорії 6а для забезпечення пропускної здатності каналу до 10 Гбіт/с на відстані до 100 метрів. Це дозволить забезпечити високу швидкість передачі даних та стабільність роботи мережі.

Також слід додати до мережі точки доступу Wi-Fi для надання доступу до мережі мобільним пристроям співробітників та відвідувачів адміністрації. Це дозволить забезпечити зручний доступ до необхідних ресурсів та покращити комунікацію між різними відділами.

Додатково, для інтеграції з телеграм-ботом, мережеві пристрої повинні підтримувати технології захисту даних та забезпечувати безперебійний доступ до Інтернету. Це включає налаштування VPN-з'єднань для захищеного обміну даними та використання протоколів шифрування для забезпечення конфіденційності інформації, яка передається через мережу.

2.1.1.3 Вимоги до режимів функціонування системи

КС повинна працювати в декількох різних режимах роботи:

– режим робочого функціонування - цей режим застосовується в робочі години адміністрації щодня протягом тижня, з 8:00 до 18:00. Він передбачає одночасну роботу всіх функціональних блоків КС, включаючи мережу адміністративних підрозділів, мережу обслуговування та моніторингу, мережу відділу зв'язків з громадськістю, а також інтеграцію з телеграм-ботом для взаємодії з громадянами;

– режим обмеженого функціонування, у цей режим мережа переводиться у неробочі години адміністрації, з 18:00 до 8:00. У цьому режимі працюють лише ті функціональні блоки КС, які використовуються для внутрішніх процесів функціонування адміністрації, такі як мережа обслуговування та моніторингу, і базові функції мережі адміністративних підрозділів. Відділ зв'язків з громадськістю та інтеграція з телеграм-ботом залишаються активними для забезпечення обслуговування запитів громадян у будь-який час;

– режим технічного обслуговування може застосовуватись замість режиму обмеженого функціонування у неробочі години адміністрації, коли виникає необхідність проведення технічного обслуговування вузлів всередині функціональних блоків або під час планового технічного обслуговування один раз на тиждень. У цьому режимі частково або повністю обмежується робота мережі для проведення діагностики, оновлення програмного забезпечення та перевірки обладнання. Інтеграція з телеграм-ботом також може бути тимчасово відключена для оновлення та технічного обслуговування.

2.1.1.4 Перспективи розвитку, модернізації системи

Необхідно мережу розділити на п'ять підмереж відповідно до організаційної структури за допомогою VLAN. Комутатори з'єднати в кільце для покращення надійності мережі шляхом зменшення точок відмов та забезпечення резервних маршрутів для передачі даних. У процесі розвитку важливою є питання ефективності

та надійності мережевої інфраструктури. Одним із ключових напрямків у цьому аспекті є модернізація мережі, що полягає у розширенні її функціоналу.

Для досягнення цієї мети можна розглянути можливість додання п'яти підмереж нових закладів геймінг-кафе та їх з'єднання через множинні маршрутизатори. Розширити мережу шляхом додавання 5 підмереж організації та їх з'єднання через множинні маршрутизатори, налаштування маршрутизації.

IP-адресацію мереж потрібно використовувати із запропонованого завдання та відповідно до варіанту. Завданий адресний простір мережі – 10.24.16.0/21. Кількість вузлів в певному сегменті мережі:

LAN1 – адміністративний відділ для роботи основних адміністративних підрозділів та забезпечення адміністративних функцій має 59 хостів;

LAN2 – відділ обслуговування та моніторингу для технічного персоналу, який відповідає за обслуговування мережі та моніторинг її стану має 124 хостів;

LAN3 – відділ зв'язків з громадськістю: для фахівців, що займаються комунікаціями, PR та підтримкою телеграм-бота має 236 хостів;

LAN4 – архівний відділ для зберігання та обробки архівних даних та документів має 42 хости;

LAN5 – IT-підтримка та розробка для IT-фахівців, що займаються розробкою, підтримкою програмного забезпечення та системного адміністрування 81хост.

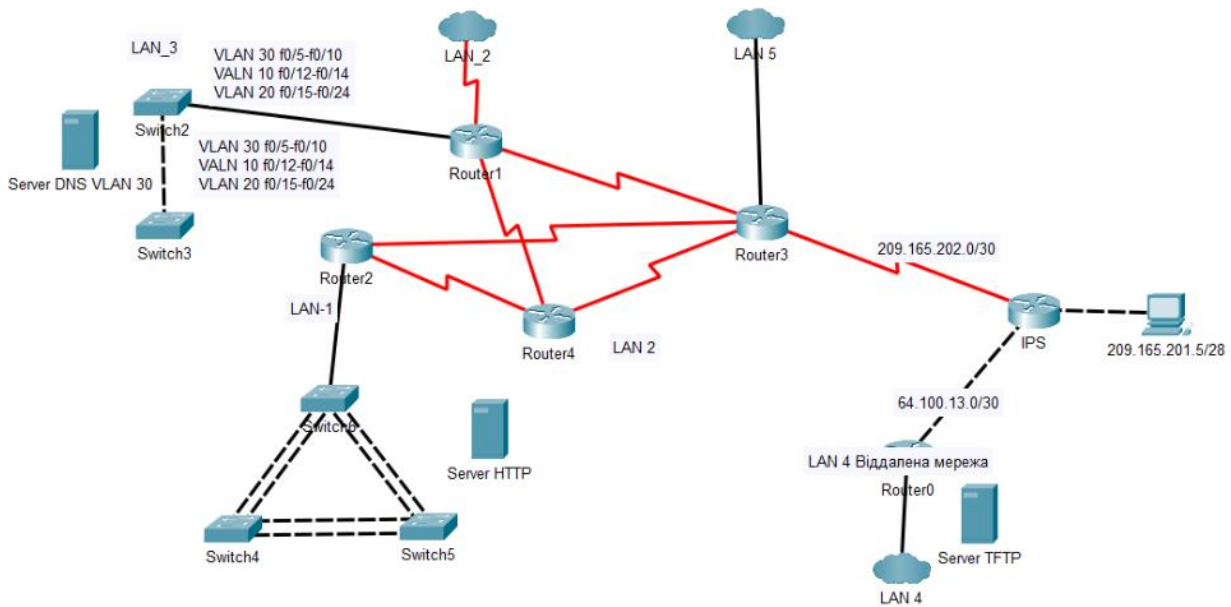


Рисунок 2.1 – Топологія КМ ААНР

2.1.1.5 Вимоги до діагностування системи

Під час всіх режимів функціонування системи Адміністрації Амур-Нижньодніпровського району м. Дніпро, необхідно забезпечити постійний моніторинг мережевих пристроїв, таких як маршрутизатори та комутатори, використовуючи SNMP. Це дозволить відстежувати навантаження на різних ділянках мережі та своєчасно виявляти будь-які помилки в їх роботі. Моніторинг також має забезпечувати збір і аналіз даних щодо використання мережевих ресурсів та інфраструктури для підтримки оптимальної продуктивності та надійності системи в усіх умовах експлуатації.

2.1.2 Вимоги до системи в цілому

2.1.2.1 Вимоги до технічного забезпечення

Мережеві пристрої, такі як маршрутизатори та комутатори, повинні забезпечувати наступне [3]:

- високу пропускну здатність каналу для забезпечення швидкого доступу до мережних ресурсів;

- підтримку протоколу SNMP для збирання даних про стан мережі та віддаленого управління.
- Потужний сервер моніторингу для обробки великої кількості даних, що надходять від мережевих пристроїв, має мати наступні характеристики:
 - багатоядерний процесор (не менше 4 ядер) для ефективною обробки паралельних процесів;
 - мінімум 16 ГБ RAM для швидкого доступу до оброблюваних даних;
 - мінімум 1 Тб жорстких дисків (рекомендується SSD) для зберігання великих обсягів даних про мережу;
 - пропускна здатність не менше 1 Гбіт/с для швидкого доступу до даних з мережевих пристроїв;
 - підтримка протоколів шифрування, таких як SSH або SSL/TLS, для захисту конфіденційності даних;
 - підтримка протоколу SNMP для надійного моніторингу та управління мережею.
 - Комутатори повинні відповідати наступним вимогам:
 - мати не менше 24 портів гігабітної мережі для підключення до різних вузлів мережі;
 - забезпечувати швидкість передачі даних до 16 Гбіт/с для ефективного обміну даними між пристроями;
 - відповідати стандарту 100BASE-TX для сумісності з різними мережними пристроями.

2.1.2.2 Вимоги до програмного забезпечення

На сервері також необхідно налаштувати систему для зберігання резервних копій даних з мережевих пристроїв, що використовують SNMP. Це дозволить забезпечити відновлення даних у випадку втрати або пошкодження основних даних або конфігурацій мережевих пристроїв.

Для забезпечення безпеки і конфіденційності даних, система повинна підтримувати механізми шифрування, такі як SSH або SSL/TLS, для захисту передачі і зберігання SNMP-даних. Це важливо для запобігання несанкціонованому доступу до мережевої інфраструктури і захисту від зловживання чи перехоплення даних.

Крім того, сервер повинен мати достатні ресурси для ефективної обробки великого обсягу даних, що надходять від мережевих пристроїв, зокрема:

- потужний багатоядерний процесор (рекомендовано не менше 4 ядер);
- мінімум 16 ГБ оперативної пам'яті RAM для швидкої обробки та аналізу даних;
- жорсткий диск ємністю не менше 1 ТБ, з перевагою на SSD для швидкого доступу до даних і високої швидкості обробки;
- пропускна здатність мережевого інтерфейсу мінімум 1 Гбіт/с для забезпечення швидкості передачі даних між мережевими пристроями і сервером моніторингу.

2.2 Розробка апаратної частини комп'ютерної системи

2.2.1 Розробка структурної схеми мережі

На основі загальної характеристики системи з підтримкою чат-бота, її організаційної структури на рис.2.2 та поставлених завдань була створена структурна схема технічних засобів комп'ютерної системи.

Розроблена схема включає в себе наступні компоненти: розташування маршрутизаторів та комутаторів дозволяє ефективно керувати мережевим трафіком, забезпечуючи високу доступність, розподіл навантаження між вузлами мережі та забезпечення високої пропускної здатності.

Сегментація мережі за допомогою Virtual LAN (VLAN): використання VLAN допомагає ізолювати трафік всередині мережі, що забезпечує підвищену захищеність

мережі та ефективного керування нею. Віртуальні мережі розділені за функціональним призначенням для полегшення керування та моніторингу:

- VLAN10 – віртуальна мережа для користувачів чат-бота. Включає 50 комп'ютерів та мобільні пристрої з можливістю бездротового доступу;
- VLAN20 – призначена для команди підтримки чат-бота. Включає 20 робочих станцій;
- VLAN30 – мережа для внутрішнього тестування та розвитку чат-бота. Включає 10 робочих станцій для розробників;
- VLAN40 – мережа для аналізу та обробки даних, включаючи аналітичні інструменти. Включає 5 робочих станцій з можливістю додавання нових пристроїв для експериментів;
- VLAN100 – віртуальна мережа для адміністрування мережним обладнанням та системою моніторингу.

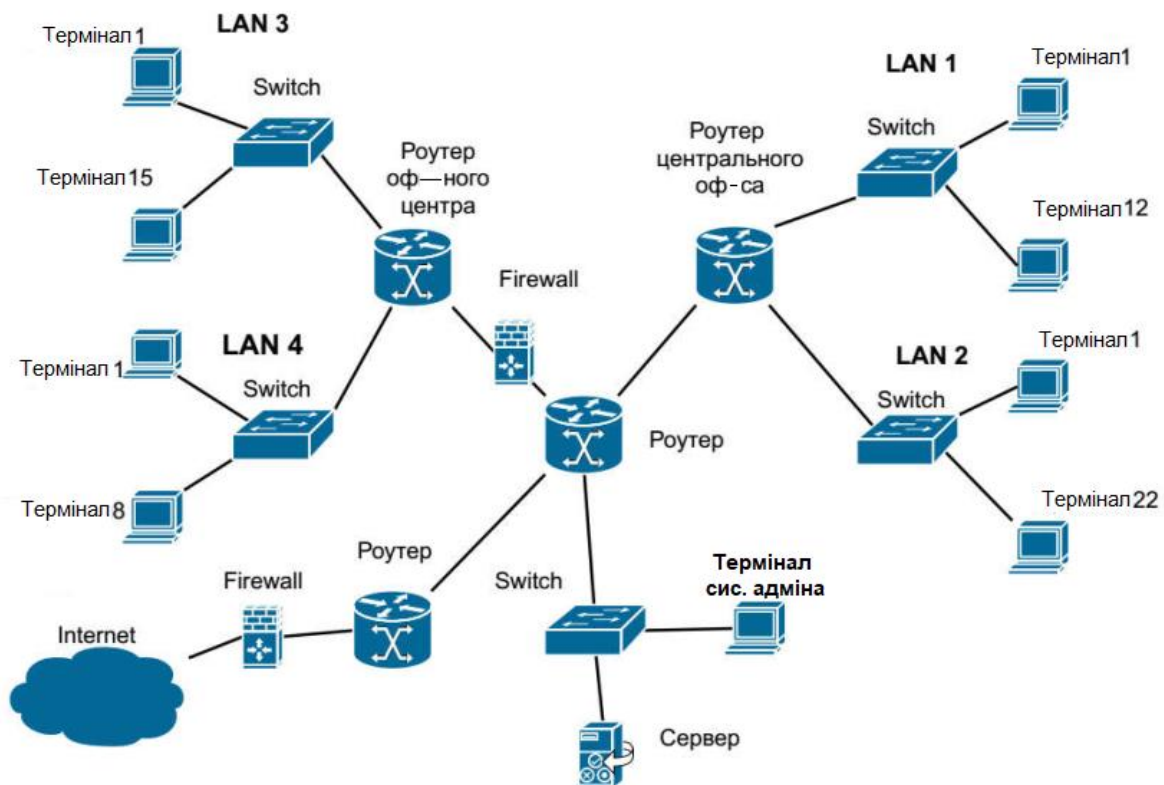


Рисунок 2.2 – Структурна схема комплексу технічних засобів КС з підтримкою чат-бота

Структурна схема (рис.2.2) дозволяє ефективно організувати і керувати мережевими ресурсами системи з підтримкою чат-бота, забезпечуючи необхідний рівень безпеки, доступності та продуктивності.

2.2.2 Розробка специфікації апаратних засобів КС

Комутатор на 24 порти було обрано з можливістю додавання нових робочих місць для співробітників ААНР (табл.2.1).

Таблиця 2.1 – Специфікація обладнання [4]

№	Найменування і технічна характеристика	Тип , марка	Кількість одиниць
1	Маршрутизатор MicroTik RJ4 Код товару CCR2004-1G-12S+2XS Архітектура ARM 64 bit ЦП AL32400 Кількість ядер ЦП 4 Номінальна частота процесора - 1700 МГц Розміри 443 x 224 x 44 мм Ліцензія -RouterOS 6 Операційна система RouterOS v7 Розмір оперативної пам'яті 4 Гб Розмір пам'яті 128 Мб Тип накопичувача NAND MTBF Приблизно 200 000 годин при 25С Випробувана температура навколишнього середовища від -20°С до 60°С	CCR2004 -1G- 12S+2XS	3
2	Інтерфейсний модуль MicroTik Швидкість LAN портів - 1,25 Гбіт/с Дальність передавання, км - 1 Тип портів- 1 гігабітний Ethernet Тип волокна - Багатомодовий	S- 85DLC05 D	3
3	Коммутатор MicroTik Архітектура: ARM 32 біти Процесор:98DX3236A1-BTD4C000	CRS328- 24P-4S + RM	3

	<p>Кількість ядер процесора/входів змінного струму: 1</p> <p>Номінальна частота процесора: 800 МГц</p> <p>Об'єм оперативної пам'яті: 512 МБ</p> <p>Об'єм накопичувача: 16 МБ</p> <p>Тип пам'яті: FLASH</p> <p>Підтримка Jumbo-фреймів: до 10218 байтів</p> <p>Неблокувальна пропускна здатність: 64 Гбіт/сек</p> <p>Комутаційна здатність: 128 Гбіт/сек</p> <p>Швидкість пересилання: 95.2 Mpps</p> <p>Випробувана температура довкілля: від -20 до 60 °С</p> <p>Автодетект PoE стандарту на кожному порту</p> <p>Діапазон вхідного змінного струму: 100 - 240 В</p>		
4	<p>Комп'ютер Відеокарта: GeForce RTX4090</p> <p>Процесор: AMD Ryzen 5 5500</p> <p>Обсяг оперативної пам'яті: 32ГБ</p> <p>Обсяг SSD: 16 ГБ</p>	<p>Nvidia</p> <p>Ryzen</p>	50
5	<p>Монітор 23.8" Samsung S24C310</p> <p>Частота оновлення 75 Гц</p> <p>Максимальна роздільна здатність дисплея</p> <p>1920x1080 (FullHD)</p> <p>Час реакції матриці 5 мс</p> <p>Вбудований тюнер Немає</p> <p>Яскравість дисплея 250 кд/м²</p> <p>Тип матриці IPS</p> <p>Інтерфейси VGA HDMI</p> <p>Споживана потужність</p> <p>Енергоспоживання: 25 Вт</p> <p>Енергоспоживання у режимі очікування: 0.3 Вт</p>	<p>LS24C31</p> <p>0EAIXCI</p>	50
6	<p>Клавіатура та мишка (набір)</p> <p>Розкладка: Eng / Ru / Ukr</p> <p>Максимальна роздільна здатність сенсора, dpi: 1400</p> <p>Комплектація набору: 2 в 1</p> <p>Тип клавіатури: Мембранні</p> <p>Комплектація: Клавіатура Миша, USB адаптер</p>	<p>Hator</p> <p>Logitech</p> <p>A4Tech</p>	50

2.2.2 Розрахунок інтенсивності трафіку найбільшої локальної мережі підприємства

В корпоративній мережі, яка використовується, встановлено обладнання компанії Cisco, а саме комутатори та маршрутизатори.

Трафік пересилається в лінію з пропускною здатністю, яка дорівнює 1000 мб/с.

Під час обчислення будемо вважати, що мережею користуються 100% користувачів. Середня інтенсивність трафіку $\mu=154$ (кадрів/с), а середня довжина повідомлення – 650 байт. В найбільшій мережі 23 пк. Пропускную здатність мережі необхідно обчислювати за наступною формулою [5]:

$$F_{p.p} = \mu * 1 * N * 8 = 154 * 650 * 23 * 8 = 18,41 \left(\frac{\text{Мбіт}}{\text{с}} \right) \quad (2.1)$$

Комутатор в мережі пересилає трафік до маршрутизатора, через лінію зі швидкістю 1000 Мбіт/с. Розрахуємо кількість пакетів на секунду, яка не повина перевищувати на комутаторі:

Розрахуємо кількість пакетів на секунду, яка не повина перевищувати на комутаторі [5]:

$$\mu_{\text{вих}} = \frac{1\,000\,000\,000}{650 * 8} = 20830 \text{ пак/с} \quad (2.2)$$

Навантаження максимальне знаходимо так:

$$N = \frac{20830}{154} = 135 \text{ джерел} \quad (2.3)$$

Розрахуємо також інтенсивність вихідного трафіку від всіх користувачів, що дорівнює:

$$\lambda = N * \mu = 23 * 154 = 3542 \text{ пак/с} \quad (2.4)$$

Розраховуємо коефіцієнт затримки:

$$\rho = \frac{\lambda}{\mu_{\text{вих}}} = \frac{3542}{20830} = 0,17 \quad (2.5)$$

Коефіцієнт зайнятості комутатора розрахуємо наступним чином:

$$r = \frac{p}{1-p} = \frac{0,17}{1-0,17} = 0,2 \quad (2.6)$$

Середня затримка кадру, яка пов'язана з чергою М/М/1 буде дорівнювати:

$$T = \frac{1}{\mu-\lambda} = \frac{1}{20830-3542} = 5,8 * 10^{-6} \quad (2.7)$$

Розрахуємо середню довжину черги:

$$L_{\text{чер}} = \frac{p^2}{1-p} = \frac{0,17^2}{1-0,17} = 0.0348 \quad (2.8)$$

Середній час, який перебуває пакет у черзі:

$$T_{\text{оч}} = \frac{L_{\text{чер}}}{\lambda} = \frac{0.0348}{3542} = 9.8 \text{ мкм} \quad (2.9)$$

Розрахуємо пропускну здатність для каналу:

λ = пропускну здатність / довжина кадру = b / l

$$b = \lambda * l = 3542 * 650 * 8 = 18,41 \frac{\text{Мбіт}}{\text{с}} \quad (2.10)$$

За розрахунками , система повністю відповідає технічним вимогам й не буде перенавантажена.

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ ПІДПРИЄМСТВА

3.1 Розрахунок схеми адресації мережі

Для конфігурування підмереж обрано початкову мережу: 10.24.16.0/21 (табл.3.1).

Таблиця 3.1 – Розрахована кількість вузлів для підмереж

Підмережа	LAN 1	LAN 2	LAN 3	LAN 4	LAN 5
Кількість вузлів	59	124	236	42	81

Для розрахунку VLSM (Variable Length Subnet Masking) [6] для мережі 10.24.16.0/21 з вказаними хостами потрібно спочатку визначити кількість бітів, необхідних для кожної підмережі відповідно до кількості хостів у кожній LAN.

LAN1 – 59 хостів:

- необхідна кількість хостів: 59;
- блок IP-адрес: 64 (бо $2^6 = 64$, включаючи адресу мережі та broadcast);
- кількість бітів підмережі: 6 ($32 - 26 = 6$);
- CIDR підмережа: /26;
- IP-адреси використовується: 10.24.16.0/26.

LAN2 – 124 хостів:

- необхідна кількість хостів: 124;
- блок IP-адрес: 128 (бо $2^7 = 128$, включаючи адресу мережі та broadcast);
- кількість бітів підмережі: 7 ($32 - 27 = 7$);
- CIDR підмережа: /27;
- IP-адреси використовується: 10.24.16.64/27.

LAN3 – 236 хостів:

- необхідна кількість хостів: 236;
- блок IP-адрес: 256 (бо $2^8 = 256$, включаючи адресу мережі та broadcast);
- кількість бітів підмережі: 8 ($32 - 28 = 8$);
- CIDR підмережа: /28;
- IP-адреси використовується: 10.24.16.128/28.

LAN4 – 42 хостів:

- необхідна кількість хостів: 42;
- блок IP-адрес: 64 (бо $2^6 = 64$, включаючи адресу мережі та broadcast);
- кількість бітів підмережі: 6 ($32 - 26 = 6$);
- CIDR підмережа: /26;
- IP-адреси використовується: 10.24.16.192/26.

LAN5 – 81 хостів:

- необхідна кількість хостів: 81;
- блок IP-адрес: 128 (бо $2^7 = 128$, включаючи адресу мережі та broadcast);
- кількість бітів підмережі: 7 ($32 - 27 = 7$);
- CIDR підмережа: /27.

Результат розподілу адрес представлено в табл.3.2.

Таблиця 3.2 – Схема розподілу адресації мережі

LAN	IP-адреса мережі	CIDR	Діапазон допустимих адрес
LAN1	10.24.16.0	/26	10.24.16.1 - 10.24.16.62
LAN2	10.24.16.64	/27	10.24.16.65 - 10.24.16.94
LAN3	10.24.16.128	/28	10.24.16.129 - 10.24.16.142
LAN4	10.24.16.192	/26	10.24.16.193 - 10.24.16.254
LAN5	10.24.16.160	/25	10.24.16.161 - 10.24.16.222

3.2 Розрахунок схеми адресації пристроїв

Наступним кроком є визначення адрес, які будуть призначені відповідним інтерфейсам маршрутизаторів у мережі. Згідно з правилом, перша доступна адреса у кожній підмережі присвоюється інтерфейсу маршрутизатора (шлюзу). Результати цього призначення наведені у таблиці таблиці 3.3.

Таблиця 3.3 – Схема адресації маршрутизаторів

Пристрій	Інтерфейс	IP-адреса	Маска
Bogosyan_Router_0	Gig0/0	10.24.16.193	255.255.255.240
	Gig0/1	64.100.13.2	255.255.255.252
Bogosyan_Router_1	Gig0/0	10.24.16.65	255.255.255. 224
	Se0/2/1	10.1.2.10	255.255.255.252
	Se0/3/0	10.1.2.18	255.255.255.252
	Se0/3/1	10.1.2.14	255.255.255.252
Bogosyan_Router_2	Gig0/0	10.24.16.1	255.255.255.192
	Se0/3/0	10.1.2.17	255.255.255.252
	Se0/3/1	10.1.2.2	255.255.255.252
	Gig0/0.25	10.24.16.1	255.255.255.224
	Gig0/0.35	10.24.16.33	255.255.255.224
	Gig0/0.45	10.24.16.65	255.255.255.224
	Gig0/0.99	10.24.16.97	255.255.255.240
Bogosyan_Router_3	Se0/2/0	10.1.2.5	255.255.255.252
	Se0/2/1	10.1.2.9	255.255.255.252
	Se0/3/0	209.165.202.2	255.255.255.252
	Se0/3/1	10.1.2.1	255.255.255.252
Bogosyan_Router_4	Gig0/0	10.24.16.129	255.255.255.240
	Se0/3/0	10.1.2.6	255.255.255.252

Пристрій	Інтерфейс	ІР-адреса	Маска
	Se0/3/1	10.1.2.13	255.255.255.252
Bogosityanv_Router_ISP	Gig0/0	209.165.201.1	255.255.255.240
	Gig0/1	64.100.13.1	255.255.255.252
	Se0/3/0	209.165.202.1	255.255.255.252

Другі з доступних адрес у підмережі необхідно призначити SVI-інтерфейсам комутаторів мережі. Результати наведено у таблиці 3.4.

Таблиця 3.4 – ІР-адреси комутаторів у підмережах

Підмережа	Пристрій	ІР-адреса SVI інтерфейсу	Маска підмережі	Адреса шлюзу
LAN1	Bogosityan_Switch_4	10.24.16.195	255.255.255.192	10.24.16.193
	Bogosityan_Switch_5	10.24.16.196	255.255.255.192	10.24.16.193
	Bogosityan_Switch_6	10.24.16.194	255.255.255.192	10.24.16.193
LAN2	Bogosityan_Switch_7	10.24.16.67	255.255.255.224	10.24.16.65
LAN3	Bogosityan_Switch_3	10.24.16.135	255.255.255.224	10.24.16.129
LAN4	Bogosityan_Switch_0	10.24.16.59	255.255.255.192	10.24.16.1
	Bogosityan_Switch_1	10.24.16.58	255.255.255.192	10.24.16.1
	Bogosityan_Switch_2	10.24.16.60	255.255.255.192	10.24.16.1
LAN5	Bogosityan_Switch_8	10.24.16.34	255.255.255.224	10.24.16.1

3.3 Розробка топологічної мережі

На основі отриманих даних, створимо модель мережі компанії у середовищі Cisco Packet Tracer. Логічна топологія, представлена на рисунку 3.1.

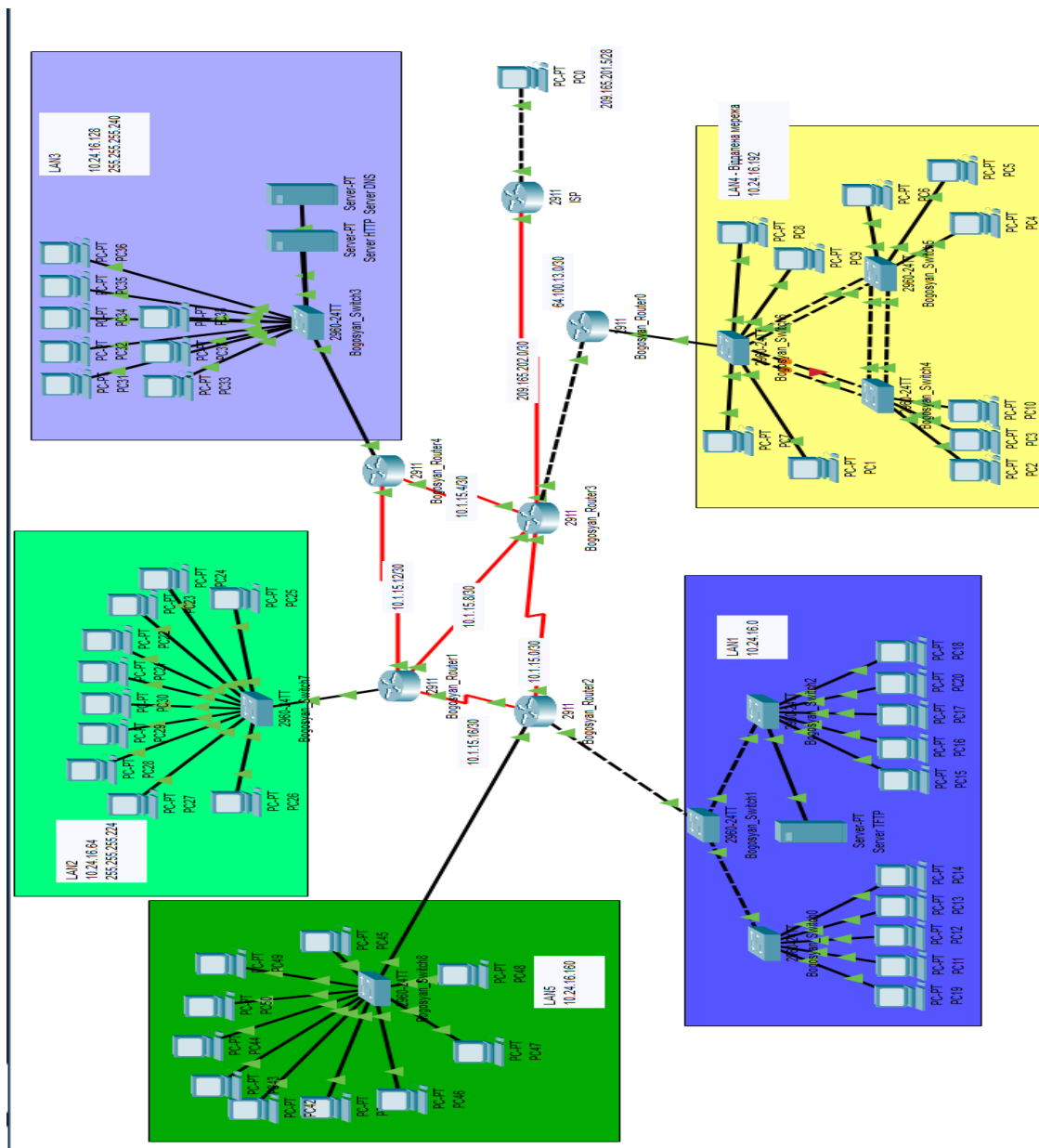


Рисунок 3.1 - Топологія мережі для ААНР

3.4 Налаштування конфігурації та роботи КС

3.4.1 Базове налаштування конфігурації пристроїв

На мережевих пристроях КС необхідно налаштувати базову конфігурацію, яка включає призначення імені пристрою, налаштування банера MOTD, створення

користувача та доменного імені, призначення паролів для привілейованого режиму, ліній console та vty, шифрування паролів, створення зашифрованого ключа довжиною 1024 біти за допомогою алгоритму RSA [7] та налаштування захищеного протоколу SSH. Розглянемо налаштування базової конфігурації на прикладі маршрутизатора Bogosyan_Router_1 (рис.3.2).

```
Enable // Вмикаємо привілейований режим EXEC
Conf t // Переходимо в режим конфігурування
Hostname Bogosyan_Router1 // Призначаємо пристрою ім'я
Line console 0 // Обираємо лінію console 0
Password cisco // Встановлюємо пароль cisco
Login // Вмикаємо авторизацію
Line vty 0 15 // Обираємо лінії vty 0 15
Password cisco // Встановлюємо пароль cisco
Login // Вмикаємо авторизацію
Enable secret cisco // Встановлюємо пароль для привілейованого режиму EXEC
Service password-encryption // Шифруємо всі створені паролі
Banner motd 'Bogosyan_Router1' // Встановлюємо банер MOTD
ip domain-name Bogosyan_Router1 // Створюємо доменне ім'я
Crypto key generate rsa // Генеруємо ключ RSA
1024 // Вказуємо довжину ключа
Username 123120_Bogosyan password admincisco // Створюємо користувача з паролем admincisco
Line vty 0 15 // Обираємо лінії vty 0 15
Transport input ssh // Вмикаємо роботу протоколу ssh
Login local // Вмикаємо локальну авторизацію
```

Рисунок 3.2 – Фрагмент базового налаштування Bogosyan_Router_1

Також необхідно виконати об'єднання фізичних портів комутаторів у мережі LAN1 для забезпечення більш стабільної роботи мережі та більшої швидкості передачі даних. Розглянемо налаштування на прикладі комутатора Bogosyan_Switch1.

```
en
conf t
interface range fa0/1-2 // Вибір діапазону інтерфейсів
channel-group 1 mode active // Активація групування каналів на обраних інтерфейсах
interface port-channel 1 // Увімкнення об'єднаного інтерфейсу
switchport mode trunk // Вмикання режиму trunk
switchport trunk allowed vlan all // Дозвіл на передачу даних для всіх VLAN
interface range fa0/5-6 // Вибір іншого діапазону інтерфейсів
channel-group 2 mode active // Активація групування каналів на обраних інтерфейсах
interface port-channel 2 // Увімкнення об'єднаного інтерфейсу
switchport mode trunk // Вмикання режиму trunk
switchport trunk allowed vlan all // Дозвіл на передачу даних для всіх VLAN
```

Рисунок 3.3 – Фрагмент команд базового налаштування Bogosyan_Switch1

Аналогічно налаштуємо решту обладнання.

3.4.2 Налаштування маршрутизації

Після початкового налаштування обладнання наступним кроком є конфігурація всіх маршрутизаторів у корпоративній мережі. Для забезпечення обміну даними між підмережами необхідно налаштувати маршрутизацію. Використовуватимемо протокол OSPF для динамічної маршрутизації. Налаштувавши цей протокол, маршрутизатори зможуть встановлювати зв'язки з сусідніми пристроями та розповсюджувати інформацію про свої мережі. Важливо відключити розсилку оновлень на внутрішні інтерфейси підмереж, щоб запобігти зайвому навантаженню на мережу.

Розглянемо налаштування на прикладі маршрутизатора Bogosyan_Router_1 (рис.3.4). Після налаштування динамічної маршрутизації потрібно додати статичні маршрути на маршрутизаторі з прямим підключенням до ISP. Потрібно створити маршрут за замовчуванням, який буде надсилати весь невідомий трафік до провайдера.

Далі створимо статичний маршрут безпосередньо до мережі провайдера.

Також необхідно налаштувати DCE-інтерфейси маршрутизаторів в мережі. Для цього встановлюємо частоту 128000 і пропускну здатність 128.

```
router ospf 1 // Створюємо нову схему маршрутизації
passive-interface GigabitEthernet0/0 // Вказуємо пасивний інтерфейс
network 10.1.15.8 0.0.0.3 area 0 // Оголошуємо мережу для розповсюдження
network 10.1.15.16 0.0.0.3 area 0 // Оголошуємо мережу для розповсюдження
network 10.1.15.12 0.0.0.3 area 0 // Оголошуємо мережу для розповсюдження
network 10.24.16.64 0.0.0.31 area 0 // Оголошуємо мережу для розповсюдження

ip route 0.0.0.0 0.0.0.0 209.165.202.1 // Створюємо маршрут за замовчуванням

ip route 209.165.201.0 255.255.255.240 209.165.202.1 // Створюємо статичний маршрут до мережі провайдера

interface Serial0/3/0 // Обираємо потрібний DCE-інтерфейс
bandwidth 128 // Встановлюємо пропускну здатність
clock rate 128000 // Встановлюємо частоту
```

Рисунок 3.4 – Фрагмент команд для маршрутизації Bogosyan_Router_1

3.4.3 Налаштування служби AAA

Потрібно налаштувати доступ до маршрутизаторів мережі через службу AAA з використанням протоколу Radius. В якості сервера, на якому працюватиме ця служба, буде використовуватися DNS-сервер компанії. Налаштування повинні забезпечити автентифікацію користувачів через консоль за допомогою протоколу Radius. Додатково слід створити локальну базу користувачів для використання у разі відсутності зв'язку з Radius-сервером. Налаштування складається з (рис.3.5 та рис.3.6):

- створення нової AAA-моделі;
- вказівка адреси Radius-сервера;
- увімкнення автентифікації користувачів на консолі за протоколом Radius;
- налаштування лінії консолі;
- увімкнення використання локальної бази користувачів;
- створення нового користувача;
- налаштування ліній vty.

```
aaa new-model
radius-server host 10.24.16.133 auth-port 1645 key radius123
aaa authentication login CONSOLE group radius local

line console 0
login authentication CONSOLE
aaa authentication login default local
Створення нового користувача:
username 123120_Bogosyan password admin123
line vty 0 15
login authentication default
```

Рисунок 3.5 – Команди для AAA

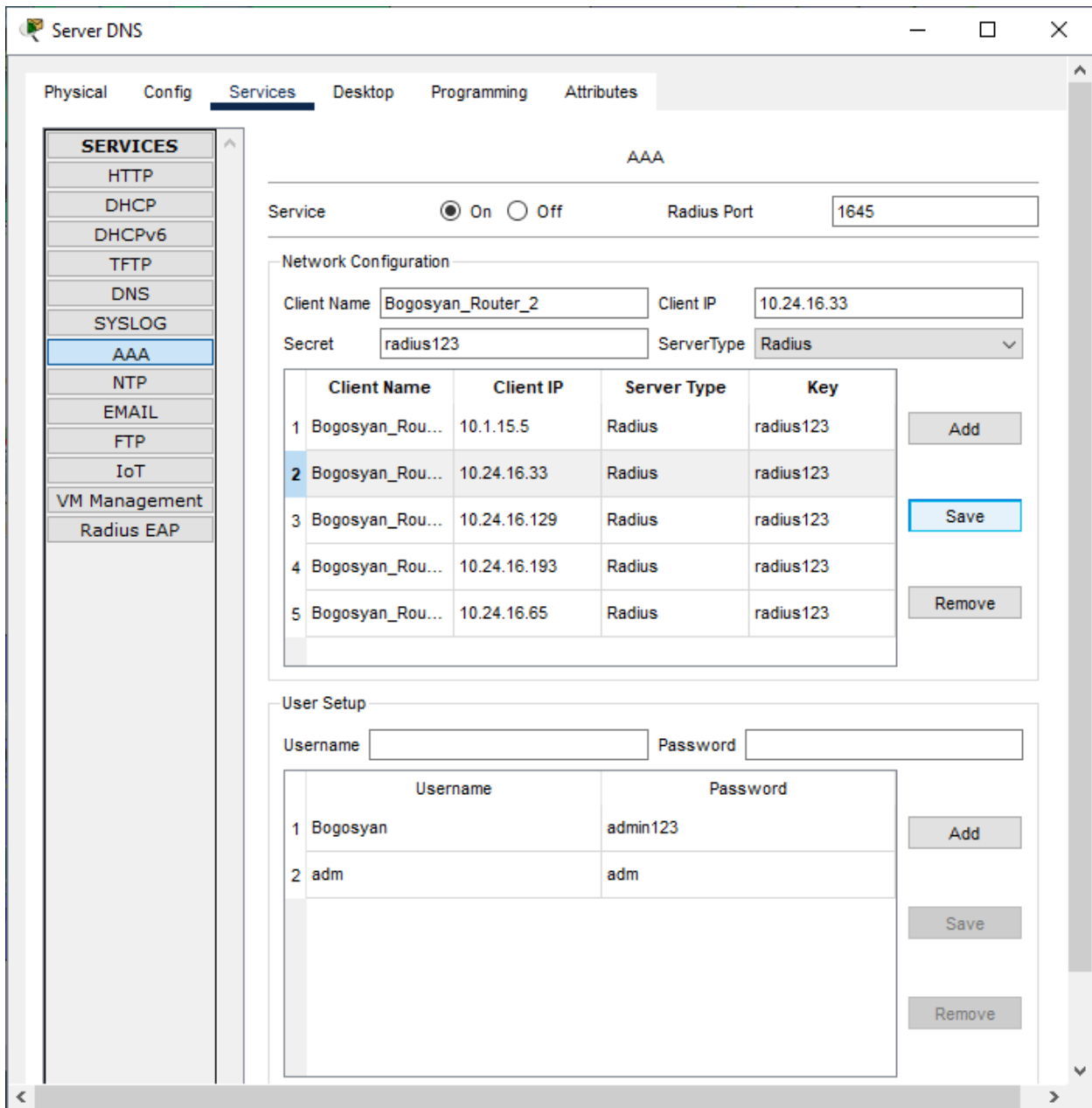


Рисунок 3.6 – Налаштування RADIUS-сервер

3.4.3 Налаштування служби Інтернет

Після налаштування маршрутизаторів у мережі необхідно забезпечити доступ до інтернету для пристроїв головного та віддаленого офісів компанії. Для цього слід налаштувати технологію NAT, яка буде перетворювати локальні адреси на публічні. Ми використаємо динамічний NAT, який автоматично обирає та присвоює публічні адреси з виділеного пулу. Для нашої компанії виділено наступний пул адрес: від 209.165.200.5 до 209.165.200.30.

Щоб визначити, який трафік потрібно транслювати, а який ні, необхідно створити ACL-списки на прикордонних маршрутизаторах. Важливо заборонити трансляцію адрес для трафіку, що надходить з мережі головного офісу до віддаленого офісу, оскільки цей трафік буде проходити через VPN-канал. Розглянемо налаштування ACL-списку на прикладі маршрутизатора Bogosyan_Router_3 (рис.3.7). Треба виконати:

- створення нового розширеного ACL-списку;
- заборона трафіку з кожної підмережі головного офісу до віддаленого офісу;
- дозвіл всього іншого трафіку.

```
ip access-list extended NAT15
deny ip 10.24.16.0 0.0.0.31 10.24.16.192 0.0.0.63
deny ip 10.24.16.128 0.0.0.63 10.24.16.192 0.0.0.63
deny ip 10.24.16.32 0.0.0.31 10.24.16.192 0.0.0.63
deny ip 10.24.16.0 0.0.0.127 10.24.16.192 0.0.0.63
deny ip 10.1.16.0 0.0.0.255 10.24.16.192 0.0.0.63
permit ip 10.16.73.0 0.0.0.31 any
permit ip 10.16.72.128 0.0.0.63 any
permit ip 10.16.73.32 0.0.0.31 any
permit ip 10.16.72.0 0.0.0.127 any
permit ip 10.1.15.0 0.0.0.255 any
```

Рисунок 3.7 – Налаштування ACL-списку

Після налаштування списку доступу, перейдемо до конфігурації технології NAT. Потрібно створити пул адрес з ім'ям Internet, увімкнути NAT, вказавши ACL-список та пул адрес, і налаштувати інтерфейси маршрутизатора як внутрішні та зовнішній. Налаштування на маршрутизаторі Bogosyan_Router_3 наведено на рис.3.8.

```
ip nat pool Internet 209.165.200.5 209.165.200.30 netmask 255.255.255.224
ip nat inside source list NAT15 pool Internet
interface Serial0/3/0
ip nat outside
interface Serial0/2/0
ip nat inside
interface Serial0/2/1
ip nat inside
interface Serial0/3/1
ip nat inside
```

Рисунок 3.8 – Налаштування NAT

Після цього необхідно створити доменне ім'я та прив'язати до нього публічну IP-адресу HTTP-сервера. Налаштування показано на рисунку 3.9.

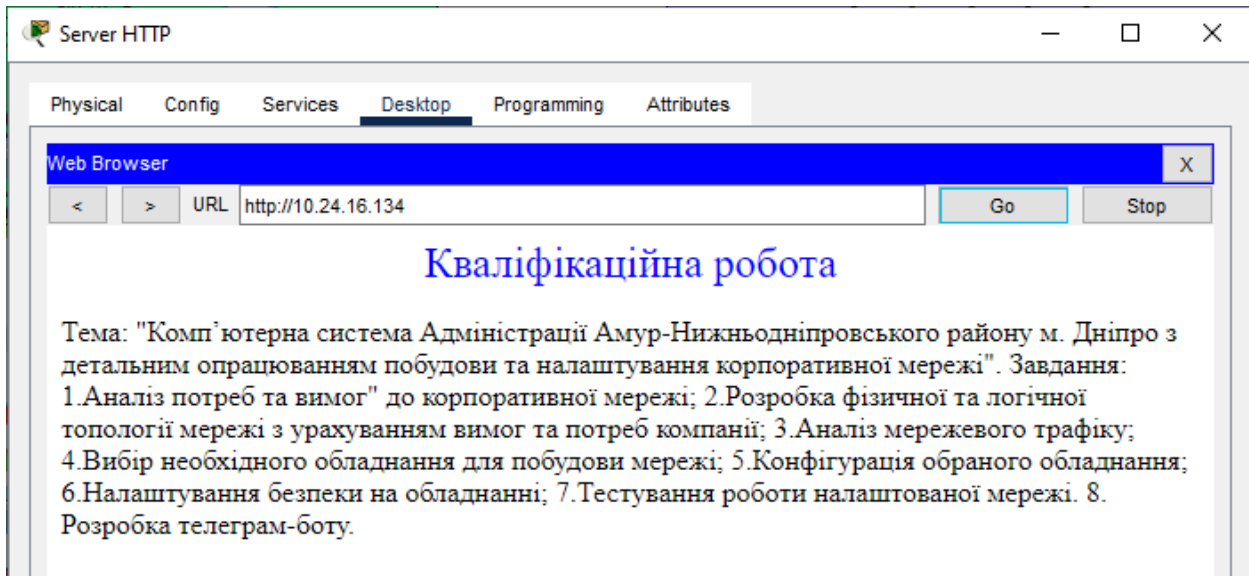


Рисунок 3.9 – Налаштування HTTP-сервера

3.4.4 Налаштування служби VLAN

Мережі LAN2 I LAN3 під'єднані до одного комутатора, тому для їх маршрутизації на комутаторі створено відповідні VLAN (див. рис. 3.10).

```
interface Vlan12
  no ip address
  !
interface Vlan22
  no ip address
  .
```

Рисунок 3.10 – Створення VLAN

Інтерфейси, до яких під'єднано кінцеві пристрої налаштовані в режим доступу відповідного VLAN (див. рис 3.11).

```

interface FastEthernet0/8
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/9
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/10
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/11
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/12
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/13
  switchport access vlan 12
  switchport mode access
!
interface FastEthernet0/14
  switchport access vlan 22
  switchport mode access
!
interface FastEthernet0/15
  switchport access vlan 22
  switchport mode access
!
interface FastEthernet0/16
  switchport access vlan 22
  switchport mode access
!
interface FastEthernet0/17
  switchport access vlan 22
  switchport mode access
!
interface FastEthernet0/18
  switchport access vlan 22
  switchport mode access
.

```

Рисунок 3.11 – Налаштування режиму доступу

Тепер потрібно виділити підмережі для зв'язку між маршрутизаторами. Оскільки для з'єднання між трьома маршрутизаторами потрібно три підмережі (по одній на кожне з'єднання між двома маршрутизаторами). А інтерфейс, яким комутатор під'єднано до маршрутизатора налаштовано в режим транк на передачу обох VLAN (див. рис. 3.12).

```

interface FastEthernet0/1
  switchport trunk allowed vlan 12,22
  switchport mode trunk
.

```

Рисунок 3.12 – Інтерфейс під'єднання комутаторів

На маршрутизаторі створено підінтерфейси, для кожної локальної підмережі, які приймають пакети з відповідної підмережі та виступають шлюзом за замовчуванням (див. рис. 3.13).

```
interface GigabitEthernet0/0.12
  encapsulation dot1Q 12
  ip address 10.24.18.1 255.255.255.192
!
interface GigabitEthernet0/0.22
  encapsulation dot1Q 22
  ip address 10.24.17.1 255.255.255.128
.
```

Рисунок 3.13 – Створення підінтерфейсів

У таблиці 3.4 перераховано номери і назви VLAN, які будуть використовуватися в LAN4.

Таблиця 3.4 – Мережі VLAN

Номер VLAN	Ім'я VLAN	Примітка
25	VLAN25	По роботі з персоналом
35	VLAN35	Фінансовий
45	VLAN45	Call-центр
1	Default	Не використовується
99	Management	Для керування пристроями
100	Native	Власна

3.4.5 Перевірка роботи комп'ютерної системи

У нашій мережі, яка має значну кількість кінцевих користувачів, використання DHCP є більш доцільним порівняно зі статичним призначенням IP-адрес, оскільки:

- DHCP автоматично призначає IP-адреси кінцевим пристроям, що суттєво спрощує процес налаштування. Адміністратору не потрібно вручну призначати IP-адреси кожному пристрою;
- автоматизація зменшує ризик людських помилок, таких як дублювання IP-адрес або неправильні конфігурації;
- DHCP динамічно виділяє IP-адреси з доступного пулу, що дозволяє уникнути ситуацій, коли IP-адреси залишаються не використаними через зарезервовані або непідключені пристрої;
- зміни в конфігурації мережі, такі як зміна маски підмережі, шлюза за замовчуванням або DNS-серверів, можна легко внести на DHCP-сервері, і всі клієнти автоматично отримають нові налаштування без необхідності вручну змінювати конфігурації на кожному пристрої.
- кінцеві користувачі отримують IP-адреси за допомогою DHCP, для чого на маршрутизаторі створено пули адрес для кожної підмережі (див. рисунок 3.14).

```
Bogosyan_Router_0# configure terminal
Bogosyan_Router_0(config)# ip dhcp excluded-address 10.24.16.193 10.24.16.202
Bogosyan_Router_0(config)# ip dhcp pool LAN1
Bogosyan_Router_0(dhcp-config)# network 10.24.16.192 255.255.255.240
Bogosyan_Router_0(dhcp-config)# default-router 10.24.16.193
Bogosyan_Router_0(dhcp-config)# end
```

Рисунок 3.14 – Створення DHCP-пулів адрес

Тепер виконуємо налаштування DHCP на ПК як показано на рис.3.15.

The image shows a screenshot of a Cisco configuration window titled "PC31". The window has tabs for "Physical", "Config", "Desktop", "Programming", and "Attributes". The "Config" tab is active, and the "IP Configuration" section is highlighted in blue. Below the tabs, the "Interface" is set to "FastEthernet0".

The "IP Configuration" section is expanded, showing the following settings:

- DHCP (selected) and Static
- IPv4 Address: 10.24.16.135
- Subnet Mask: 255.255.255.240
- Default Gateway: 10.24.16.129
- DNS Server: 10.24.16.130

The "IPv6 Configuration" section is also expanded, showing the following settings:

- Automatic and Static
- IPv6 Address: (empty field) / (empty field)
- Link Local Address: FE80::2D0:FFFF:FE41:8C52
- Default Gateway: (empty field)
- DNS Server: (empty field)

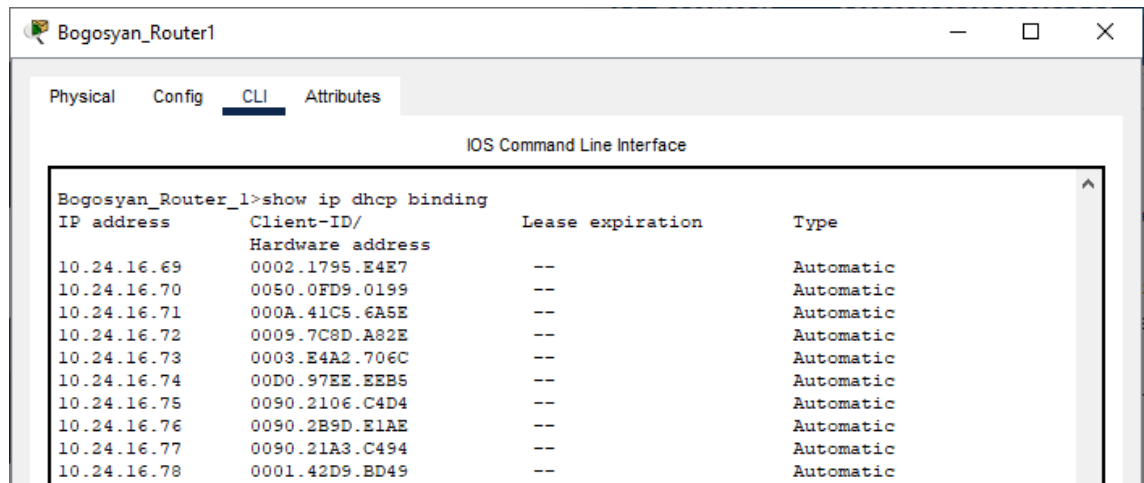
The "802.1X" section is expanded, showing the following settings:

- Use 802.1X Security
- Authentication: MD5 (selected in a dropdown menu)
- Username: (empty field)
- Password: (empty field)

At the bottom left of the window, there is a "Top" button.

Рисунок 3.15 – Налаштування DHCP протоколу

Для перевірки налаштувань DHCP на маршрутизаторі Cisco вводимо команду `show ip dhcp pool` в CLI (рис.3.16).

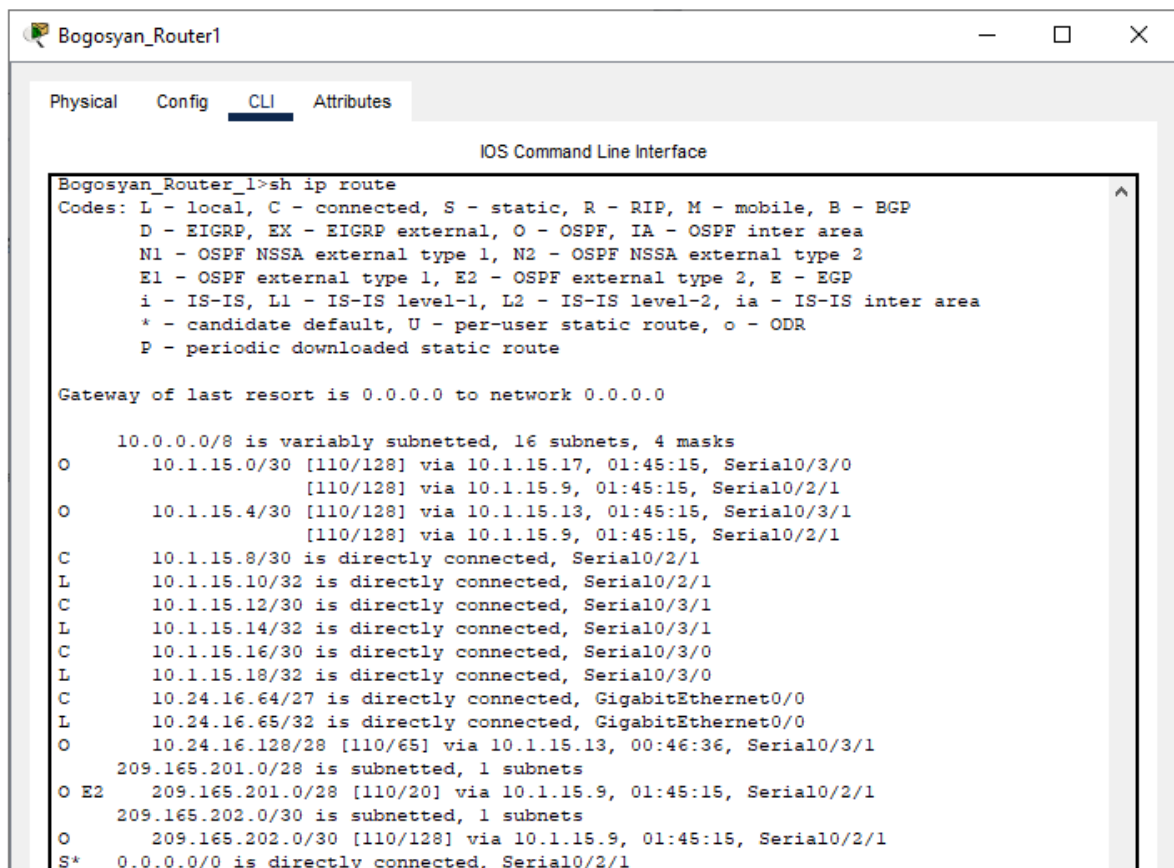


The screenshot shows the CLI of a router named 'Bogosityan_Router1'. The 'CLI' tab is selected. The command 'show ip dhcp binding' has been executed, resulting in a table of DHCP bindings.

IP address	Client-ID/ Hardware address	Lease expiration	Type
10.24.16.69	0002.1795.E4E7	--	Automatic
10.24.16.70	0050.0FD9.0199	--	Automatic
10.24.16.71	000A.41C5.6A5E	--	Automatic
10.24.16.72	0009.7C8D.A82E	--	Automatic
10.24.16.73	0003.E4A2.706C	--	Automatic
10.24.16.74	00D0.97EE.EEB5	--	Automatic
10.24.16.75	0090.2106.C4D4	--	Automatic
10.24.16.76	0090.2B9D.E1AE	--	Automatic
10.24.16.77	0090.21A3.C494	--	Automatic
10.24.16.78	0001.42D9.BD49	--	Automatic

Рисунок 3.16 – Перевірка налаштувань DHCP на маршрутизаторі

На рисунку 3.17 наведено таблицю маршрутизацію, в якій бачимо налаштований динамічний протокол OSPF.



The screenshot shows the CLI of a router named 'Bogosityan_Router1'. The 'CLI' tab is selected. The command 'sh ip route' has been executed, displaying the routing table and gateway information.

```

Bogosityan_Router1>sh ip route
Codes: L - local, C - connected, S - static, 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 0.0.0.0 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 16 subnets, 4 masks
O    10.1.15.0/30 [110/128] via 10.1.15.17, 01:45:15, Serial0/3/0
     [110/128] via 10.1.15.9, 01:45:15, Serial0/2/1
O    10.1.15.4/30 [110/128] via 10.1.15.13, 01:45:15, Serial0/3/1
     [110/128] via 10.1.15.9, 01:45:15, Serial0/2/1
C    10.1.15.8/30 is directly connected, Serial0/2/1
L    10.1.15.10/32 is directly connected, Serial0/2/1
C    10.1.15.12/30 is directly connected, Serial0/3/1
L    10.1.15.14/32 is directly connected, Serial0/3/1
C    10.1.15.16/30 is directly connected, Serial0/3/0
L    10.1.15.18/32 is directly connected, Serial0/3/0
C    10.24.16.64/27 is directly connected, GigabitEthernet0/0
L    10.24.16.65/32 is directly connected, GigabitEthernet0/0
O    10.24.16.128/28 [110/65] via 10.1.15.13, 00:46:36, Serial0/3/1
209.165.201.0/28 is subnetted, 1 subnets
O E2  209.165.201.0/28 [110/20] via 10.1.15.9, 01:45:15, Serial0/2/1
209.165.202.0/30 is subnetted, 1 subnets
O    209.165.202.0/30 [110/128] via 10.1.15.9, 01:45:15, Serial0/2/1
S*   0.0.0.0/0 is directly connected, Serial0/2/1

```

Рисунок 3.16 – Перевірка таблиці маршрутизації

Перевіримо наявність зв'язку між вузлами в різних підмережах за допомогою ping між LAN2-LAN3 (рис.3.17).

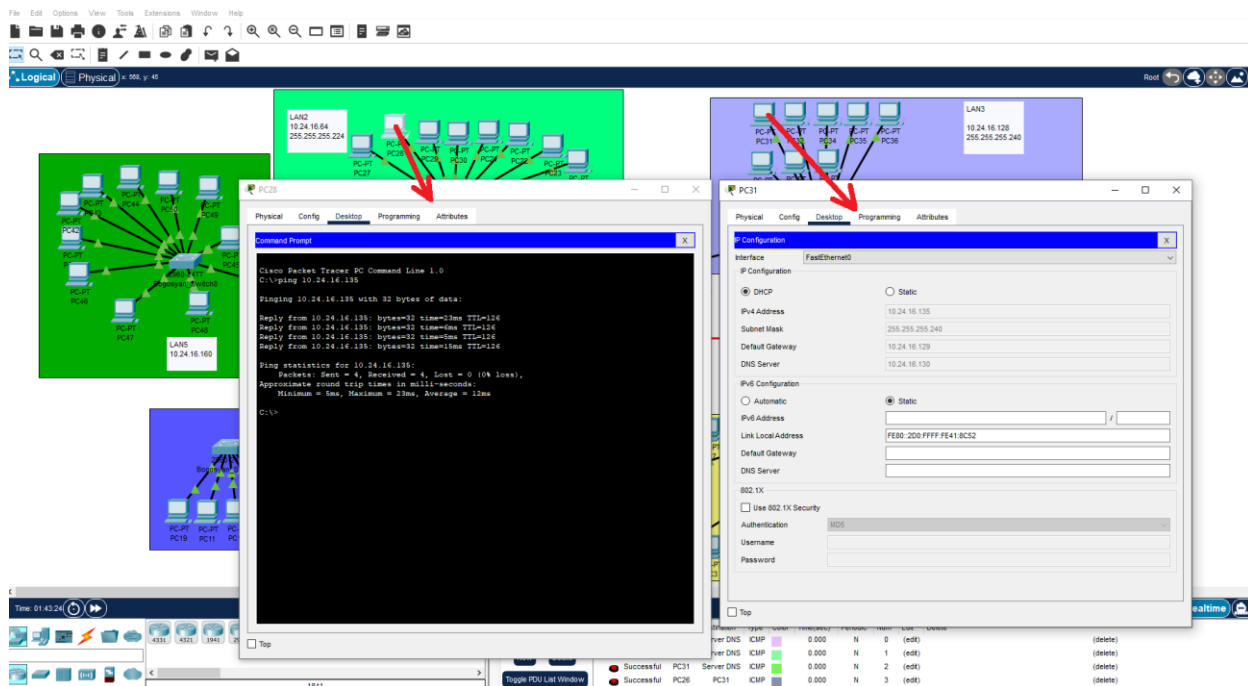


Рисунок 3.17 – Перевірка зв'язку між вузлами

Для забезпечення безпеки комутаторів та маршрутизаторів було встановлено паролі на консольний доступ та на привілейований режим. Крім того, на кожному з мережевих пристроїв налаштовано банер MOTD з назвою відповідного пристрою (див. рис. 3.18).

```

Bogosyan_Switch_4
User Access Verification

Username: Bogosyan
Password:
Bogosyan Switch 4>
  
```

Рисунок 3.18 – Встановлення паролів

Для забезпечення віддаленого доступу до всіх мережевих пристроїв створено користувачів, налаштовано доменні імена та сконфігуровано лінії віддаленого доступу з використанням протоколу SSH. (див. рис. 3.19):

```
C:\>ssh -l 1231_bogosyan 10.24.16.1

Password:

Welcome to bogosyan_R1

bogosyan_R1#en
bogosyan_R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
bogosyan_R1(config)#
```

Рисунок 3.19 – Перевірка можливості віддаленого доступу до маршрутизатора

Перевірка можливості віддаленого доступу до комутатора (рис. 3.20).

```
C:\>ssh -l 1231_bogosyan 10.24.18.14

Password:

Welcome to bogosyan_S2

bogosyan_S2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
bogosyan_S2(config)#ex
bogosyan_S2#ex

[Connection to 10.24.18.14 closed by foreign host]
C:\>
```

Рисунок 3.20 – Перевірка можливості віддаленого доступу до комутатора

Система повністю відповідає вимогам , працює коректно , та може використовуватися у роботі.

4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ

4.1 Розробка телеграм боту для ААНР

Чат-боти класифікуються за декількома критеріями: платформа впровадження, технологія розробки, спосіб спілкування з користувачами та функціональність. Вони можуть бути простими, такими як чат-боти в Telegram, що виконують одне завдання за заданим сценарієм. Існують учні чат-боти, які використовують штучний інтелект: вони ставлять уточнюючі питання, пропонують товари на основі відповідей користувача і зберігають унікальні пошукові запити для покращення відповідей.

Чат-боти сприяють ефективній комунікації компаній з клієнтами, дозволяючи швидко отримувати необхідні інструкції без очікування на оператора. Вони також заощаджують час співробітників, виконуючи прості комунікаційні завдання.

Telegram-бот, розроблений на Python [8] з використанням бібліотек `sqlite3` для роботи з базою даних, `datetime` для обробки часу і дати, а також бібліотек `telegram` і `telegram.ext` для інтеграції з Telegram, автоматизує процес обміну документами всередині організації. Він приймає файли від співробітників, зберігає їх у базі даних SQLite і передає для подальшого розгляду керівництву. Керівники можуть приймати, відхиляти або коментувати файли, і відповідні повідомлення надсилаються співробітникам, щоб забезпечити ефективну обробку і зворотний зв'язок..

4.2 Бібліотека SQLite

SQLite (вимовляється як «еск'юлайт») — це компактна вбудована реляційна база даних з відкритим кодом, визнана світовим співтовариством. Вона використовується широко в додатках та системах, де потрібно ефективно зберігати та управляти даними. SQLite не є окремим сервером баз даних, але є вбудовуваною бібліотекою, що дозволяє легко інтегрувати її безпосередньо у додаток без потреби у встановленні окремого сервера. Основні особливості включають низькі системні вимоги, підтримку стандарту SQL, можливість використання через бібліотеки API різних мов програмування, підтримку транзакцій для забезпечення цілісності даних та простоту у використанні.

SQLite підходить для мобільних застосунків та інших середовищ з обмеженими ресурсами, де необхідна надійна та легка у використанні база даних. Вона забезпечує простий механізм взаємодії з даними, зберігання всієї інформації в одному файлі бази даних, що спрощує резервне копіювання та перенесення даних. Хоча SQLite має свої переваги, такі як легкість використання і вбудованість, вона не підходить для великих обсягів даних та високонавантажених систем через обмежену підтримку одночасних записів та відсутність можливості віддаленого доступу через мережу [9].

4.2 Розробка програмної частини

4.2.1 Імпорт необхідних бібліотек

Імпортуємо усі необхідні бібліотеки у програмний код , який ми використовуємо для написання боту (див. рис. 4.1) [8].

```
import sqlite3
from datetime import datetime

from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
from telegram.ext import Updater, MessageHandler, Filters, CallbackQueryHandler
```

Рисунок 4.1 – Імпорт необхідних бібліотек

Обрані бібліотеки для роботи в цьому прикладі включають:

- `sqlite3` надає інтерфейс для роботи з базою даних SQLite в Python. Вона дозволяє створювати з'єднання з базою даних, виконувати SQL-запити, керувати даними та забезпечувати взаємодію програми зі збереженими даними;
- `datetime` в Python використовується для роботи з датами та часом. Вона надає класи та функції для представлення та маніпулювання даними про час, що є важливим для обробки часових маркерів у програмі;
- `telegram` це набір бібліотек для розробки Telegram-ботів у Python. Конкретно використовуються класи та функції з `telegram` і `telegram.ext` для створення ботів, обробки повідомлень, створення відповідей на дії користувачів через кнопки і клавіатуру.

Ці бібліотеки разом дозволяють створювати Telegram-бота, який може взаємодіяти з користувачами, зберігати та обробляти дані в базі даних SQLite, а також працювати з часовими маркерами для правильної обробки даних.

Схема отримання запитів від Telegram API представлена на рис.4.2. Опишемо функціонал. Спершу потрібно створити бота у Telegram за допомогою спеціального бота @BotFather. Цей крок включає надання унікального імені бота та отримання токена доступу, який потрібно використовувати для аутентифікації запитів до API.

Розробник повинен налаштувати середовище розробки, щоб бот міг приймати та обробляти повідомлення від користувачів Telegram. Це включає використання фреймворків, таких як python-telegram-bot, для спрощення обробки повідомлень та інших подій.

Telegram API передає повідомлення боту у форматі JSON через HTTP-запити. Бот повинен мати встановлені обробники (handlers) для різних типів повідомлень, наприклад, текстових повідомлень, файлів, команд і т.д.

Після отримання повідомлення бот обробляє його, генерує відповідь та відправляє її назад користувачеві через Telegram API. Відповідь також може містити текст, медіа-файли, кнопки, клавіатуру і т.д.

Telegram-боти використовуює стан для керування сесіями користувачів. Це дозволяє зберігати стан обміну повідомленнями і реагувати на введення користувача залежно від поточного контексту.

Важливо дотримуватись політики безпеки Telegram API, включаючи обмеження на частоту запитів і забезпечення безпеки передачі даних.

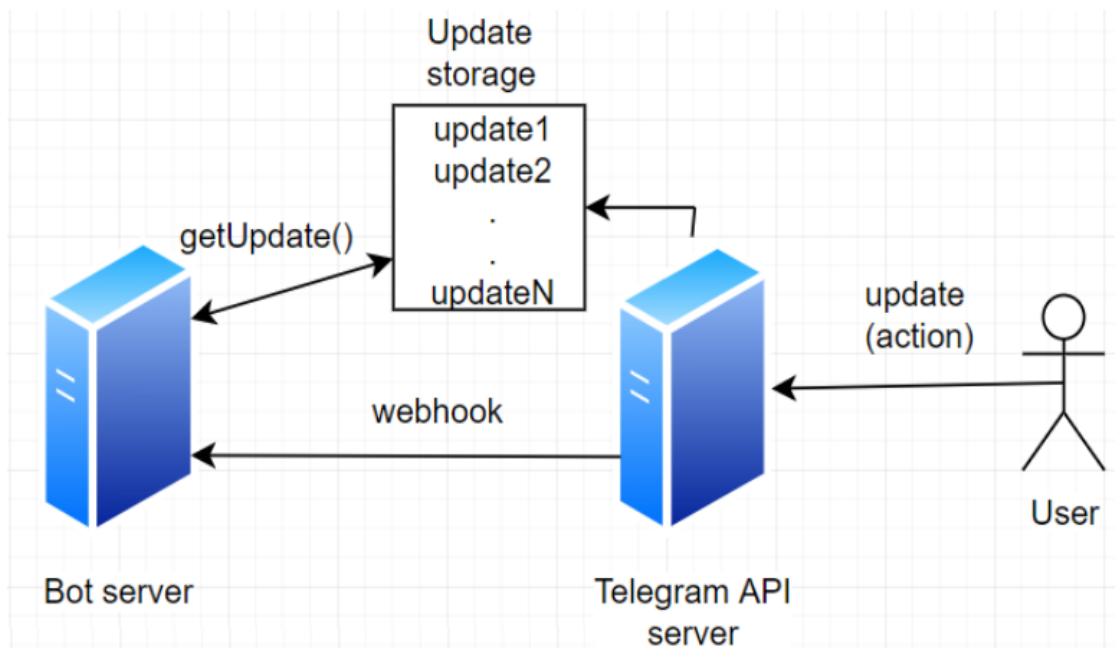


Рисунок 4.2 – Схема отримання запитів від Telegram API

4.2.2 Створення токена

Токен є унікальним ключем, складеним з символів, необхідним для зв'язку між webhook (веб-посилання, через яке відбувається обмін повідомленнями з сервером Telegram) та сервером Telegram. Цей ключ використовується для ідентифікації вашого бота та забезпечення безпеки комунікацій. (рис 4.3).

```
TOKEN = '6539864350:AAGfdFXqmQYUUQyJUjPU0QVS3Wqt31V4keY'
```

Рисунок 4.3 – Створення токена

Після створення токена ми формуємо список ідентифікаторів менеджерів, які відповідають за прийняття або відхилення надісланих файлів. Кожен ідентифікатор пов'язаний з інформацією про співробітника, його посаду та час надсилання файлу керівництву.

Для зберігання метаданих файлів ми використовуємо базу даних SQLite, створюючи таблицю "files", де зберігаються дані про файли, відправлені

співробітниками. Після підключення до бази даних ми можемо виконувати SQL-запити для збереження і отримання інформації про файли.

Функція "handle_file" призначена для обробки отриманих файлів, збереження їх у базі даних та надсилення на розгляд менеджером з використанням кнопок для прийняття чи відхилення файлу. Вона також створює повідомлення з метаданими файлу для кожного менеджера.

Функція "handle_button" обробляє натискання кнопок керівництва, що використовуються для прийняття або відхилення файлів. В залежності від вибраної кнопки вона змінює статус файлу в базі даних та інформує співробітника про прийняте рішення.

Для обробки коментарів, які залишаються менеджерами при відхиленні файлу, використовується функція "handle_comment". Вона дозволяє додати коментар до відхилених файлів у базі даних та сповістити співробітника про причину відхилення.

Функція "notify_employee" відповідає за надсилення сповіщень співробітникам про рішення щодо їхніх файлів, використовуючи ідентифікатор співробітника, збережений у таблиці бази даних.

```
def handle_file(update: Update, context):
    if update.message.document:
        file = update.message.document
        sender_id = update.effective_user.id
        sender_name = update.effective_user.full_name
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        # Save the file data in the database
        file_data = file.get_file().download_as_bytearray()
        cursor.execute(
            "INSERT INTO files (employee, employee_id, file_name, file_data, timestamp, accepted) VALUES (?, ?, ?, ?, ?, "
            "? , ?)",
            (sender_name, sender_id, file.file_name, sqlite3.Binary(file_data), timestamp, False))
        file_id = cursor.lastrowid
        conn.commit()

        # Send the file to management with metadata
        caption = f"Файл: {file.file_name}\nБув відправлен: {sender_name}\nЧас: {timestamp}"
        for manager_id in management:
            sent_file = context.bot.send_document(chat_id=manager_id,
            document=bytes(file_data),
            filename=file.file_name, caption=caption)
            context.bot.send_message(chat_id=manager_id, text="Оберіть дію:",
            reply_markup=get_action_buttons(file_id, sent_file.message_id))
```

Рисунок 4.4 – Створення функцій керівництва

Всі ці компоненти разом створюють функціональну систему для обробки файлів у Telegram боті, забезпечуючи ефективний обмін документами та зручність в управлінні файловими процесами в організації.

Загальна схема роботи телеграм боту наведена на рис.4.5. У цій схемі задіяно декілька основних компонентів: NoSQL та SQL бази даних, створення користувача, логіка меню, підготовка та відправка даних, Telegram сервер та користувач.

NoSQL та SQL бази даних використовуються для зберігання даних додатку. Методи `checkParams()`, `saveAndUpdateMenu()`, `handle(menuType)` працюють з цими базами даних для забезпечення коректної роботи додатку. Ці методи дозволяють перевіряти параметри, зберігати та оновлювати меню, а також обробляти типи меню.

Процес взаємодії починається з перевірки параметрів за допомогою методу `checkParams()`. Залежно від результатів перевірки, додаток може виконувати збереження або оновлення меню за допомогою `saveAndUpdateMenu()`, або обробляти тип меню за допомогою `handle(menuType)`. Далі йде блок створення користувача. Якщо користувача ще не існує в системі, він створюється, інакше продовжується взаємодія з існуючим користувачем.

Додаток надсилає запити до Telegram серверу через `webhook` або `getUpdate()`. Цей сервер відповідає користувачам, що дозволяє додатку отримувати та надсилати повідомлення користувачам через різні пристрої, такі як смартфон, комп'ютер чи сервер.

Після цього додаток обробляє логіку меню на основі даних, отриманих від користувача, у блоці "логіка меню". Підготовка та відправка відповідних даних здійснюється у наступному блоці, де додаток готує необхідну інформацію і надсилає її користувачеві через Telegram сервер.

Ця схема демонструє основні етапи взаємодії додатку з користувачами через Telegram, використовуючи бази даних для зберігання та обробки даних. Вона показує, як додаток забезпечує створення нових користувачів, обробку запитів і повідомлень, а також управління меню та даними.

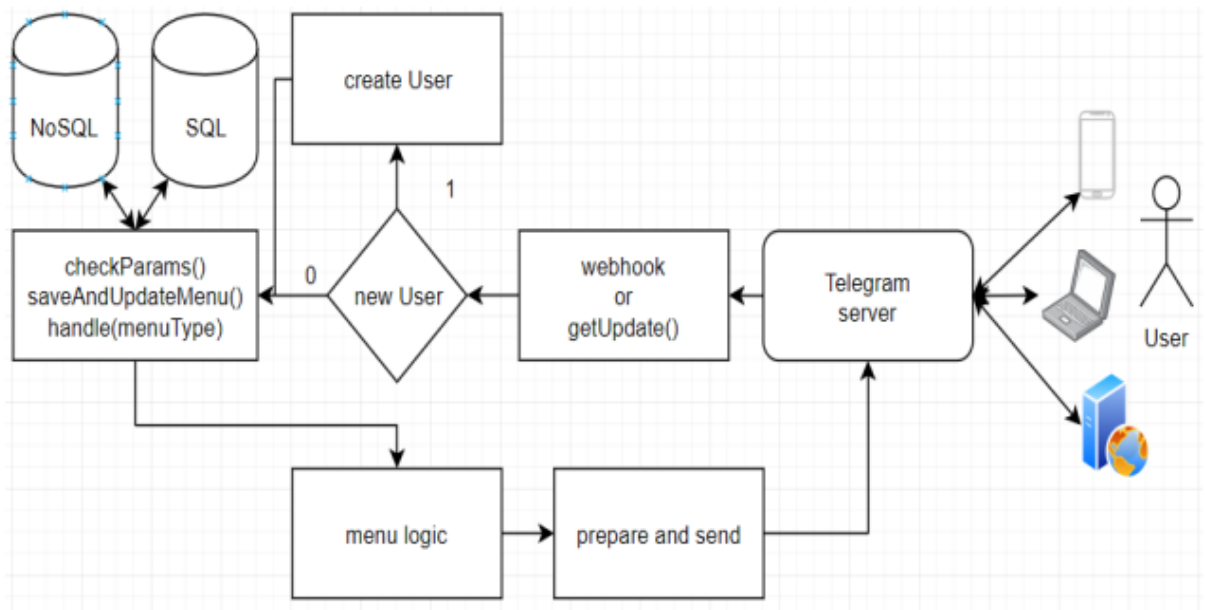


Рисунок 4.5 – Загальна структурна схема телеграм боту

Алгоритм роботи додатку, представлено на рис.4.6, складається з декількох послідовних етапів, починаючи з ініціації додатку. Перший етап — це початок процесу, де відбувається ініціація додатку. Далі користувач надсилає запит до додатку, який потрапляє до контролера. Контролер обробляє початкові дані та передає запит на подальшу обробку.

На наступному етапі додаток перевіряє, чи існує користувач у системі. Якщо користувача не існує, система створює нового користувача. Якщо користувач вже існує, процес переходить до обробки та збереження параметрів, наданих користувачем. Параметри обробляються та зберігаються у системі для подальшої роботи.

Після цього додаток викликає відповідне меню для взаємодії з користувачем. На основі вибору користувача обробляється логіка роботи меню. Під час цього етапу додаток локалізує меню та готує відповідь для відправки користувачеві. Готова відповідь надсилається до Telegram серверу, який передає її користувачеві. Після завершення всіх етапів процес завершується, і додаток готовий до обробки наступного запиту.

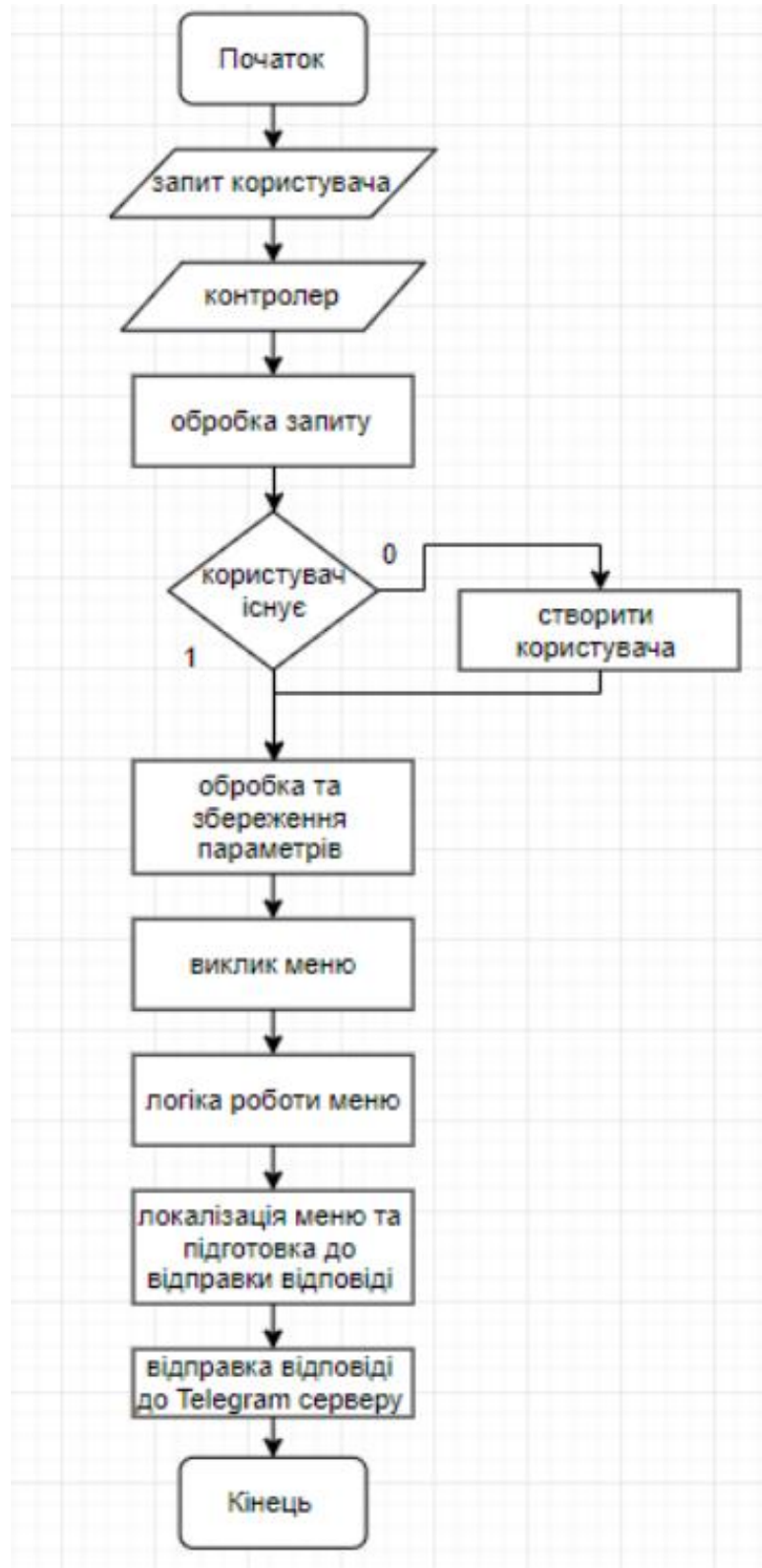


Рисунок 4.6 – Алгоритм роботи додатку

4.3 Перевірка роботи розробленого телеграм боту

Тепер, коли бота повністю налаштовано, перевіримо його роботу з обох сторін — керівництва й персоналу. Для цього спочатку протестуємо основні функції та можливості бота, переконавшись, що всі запити обробляються коректно і відповіді надаються вчасно. Зі сторони персоналу надсилається певний файл, який розглядається керівництвом (рис.4.8). Після огляду повертається повідомлення від керівників: прийнято файл або ні. Щодо сторони керівництва: вони отримують повідомлення, у якому вказано файл, ім'я файлу, ким він був відправлений, та час надсилання. Їм надається 3 кнопки, а саме: «Погодити», «Повернути з коментарем» або «Відмовити». Після натиску на певну кнопку відправляється повідомлення(рис.4.8).

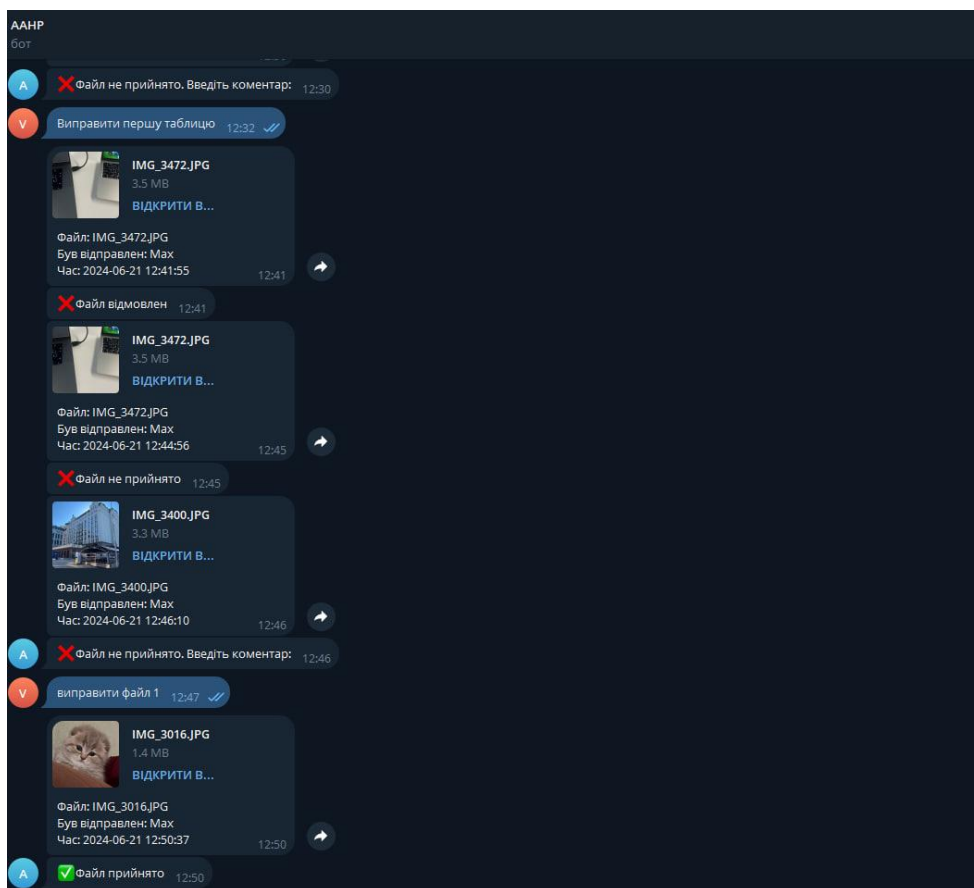


Рисунок 4.7 – Функціонал бота зі сторони керівництва

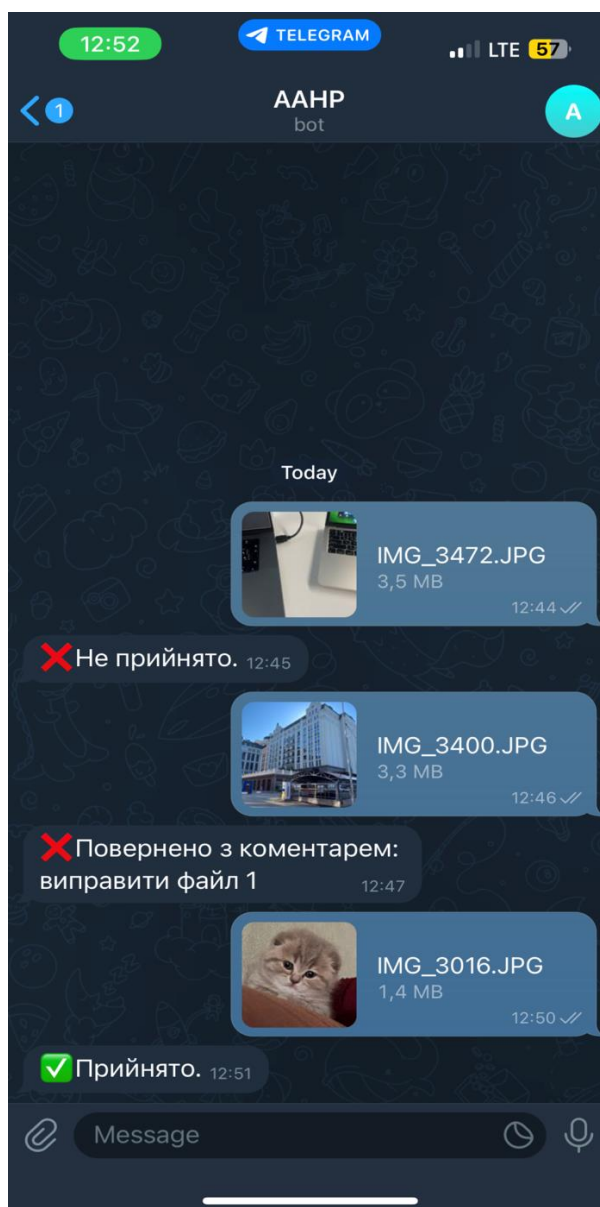


Рисунок 4.8 – Функціонал боту зі сторони персоналу

Розробка та впровадження Telegram-бота для обробки файлів від співробітників значно підвищує ефективність внутрішніх процесів в організації. Бот автоматизує прийом, зберігання та передачу файлів керівництву, що дозволяє зменшити обсяг ручної роботи та мінімізувати ризик помилок. Використання бази даних SQLite забезпечує надійне зберігання файлів, полегшує їх пошук та відстеження історії взаємодій. Інтерфейс Telegram-бота зручний та інтуїтивно зрозумілий, що спрощує роботу як для співробітників, так і для керівництва. Можливість додавання коментарів до відхилених файлів надає зворотний зв'язок, сприяючи покращенню якості роботи. Використання кнопок дій значно полегшує процес прийняття рішень.

Бот також автоматично повідомляє співробітників про статус їхніх файлів, підвищуючи прозорість процесу. Розробка такого бота демонструє, як сучасні технології можуть покращити взаємодію в колективі. Загалом, впровадження цього бота сприяє підвищенню продуктивності та організаційної культури. Подальший розвиток бота може включати розширення функціоналу та інтеграцію з іншими системами для ще більшої ефективності.

ВИСНОВКИ

У кваліфікаційній роботі "Комп'ютерна система Адміністрації Амур-Нижньодніпровського району м. Дніпро з детальним опрацюванням побудови та налаштування корпоративної мережі" було виконано завдання, які сприяли розробці та впровадженню сучасної комп'ютерної системи для адміністрації району.

Перш за все, було проведено детальне ознайомлення зі структурою організації. Це включало аналіз існуючої організаційної структури адміністрації та визначення специфічних потреб і вимог до комп'ютерної системи з урахуванням особливостей роботи різних підрозділів. Такий підхід дозволив чітко окреслити функціональні вимоги до майбутньої системи.

На основі проведеного аналізу було розроблено технічні вимоги для комп'ютерної системи. Визначено основні функціональні та нефункціональні вимоги, складено список необхідних технічних характеристик і параметрів. Це забезпечило чітке розуміння необхідних ресурсів та критеріїв для відбору обладнання та програмного забезпечення.

Для забезпечення належної роботи системи було здійснено підбір комплексу технічних засобів. Вибір апаратного забезпечення проводився з урахуванням технічних вимог та оптимальних варіантів, що гарантує стабільну та ефективну роботу системи.

Окремо було приділено увагу вибору топології корпоративної мережі. Проаналізовано різні можливі топології та обрано найбільш підходящу для потреб адміністрації, враховуючи перспективи майбутнього розвитку мережі. Після цього проведено розрахунок налаштувань для обраної топології мережі, що включало вибір відповідних інтерфейсів каналів зв'язку та протоколів обміну даними.

Розроблена топологічна (логічна) схема комп'ютерної системи відображає логічну структуру мережі та взаємозв'язки між її елементами. Це забезпечує зрозумілість та зручність в управлінні мережею.

Далі було проведено розрахунок налаштувань маршрутизації комп'ютерної мережі, що включало визначення оптимальних маршрутів передачі даних та

налаштування протоколів маршрутизації для забезпечення ефективного обміну інформацією.

Налаштування корпоративної мережі та її тестування здійснювалися відповідно до розробленої схеми. Проведено тестування мережі для перевірки працездатності та стабільності її роботи, що дозволило виявити та усунути можливі недоліки.

Підбір обладнання та програмного забезпечення був здійснений з метою реалізації КС та забезпечення виконання завдань адміністрації. Обрано відповідне обладнання та програмне забезпечення, яке відповідає встановленим технічним вимогам.

Останнім етапом стала розробка телеграм-бота для Адміністрації Амур-Нижньодніпровського району м. Дніпро. Бот створено для покращення взаємодії з громадянами та інтегровано з існуючою комп'ютерною системою для автоматизації певних процесів.

Виконання цих завдань дозволило створити надійну, ефективну та безпечну комп'ютерну систему, що відповідає сучасним вимогам і сприяє підвищенню продуктивності роботи Адміністрації Амур-Нижньодніпровського району м. Дніпро. Подальший розвиток системи може включати розширення функціоналу та інтеграцію з іншими системами для досягнення ще більшої ефективності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Адміністрації Амур-Нижньодніпровського району м. Дніпро. URL: <https://dniprorada.gov.ua/uk/page/administraciya-amur-nizhnodniprovsкого-rajonu-dniprovs-koi-miskoi-radi>
2. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 інформаційні технології спеціальності 123 комп'ютерна інженерія / Л.І. Цвіркун, В.В. Гнатушенко, С.М. Ткаченко ; М-во освіти і науки України, НТУ “ДП” – Дніпро: 2023. – 29 с
3. Network Warrior 2e Paperback – 14 Jun. 2011 English edition by Gary A. Donahue - 650 сторінок.
4. Технічні характеристики обладнання Cisco – Режим доступу: <https://stack-systems.com.ua/> (дата звернення 24.05.2024)
5. Networking All-in-One for Dummies Paperback – 10 April 2018 English edition by Doug Lowe – 948 сторінок.
6. Pernul, G. Database Security / G. Pernul // In: Advances in Computers, Vol. 38. ed. by MC Yovits. - Academic Press. - 1994. - pp.
7. 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, НТУ «Дніпровська політехніка», 2018. – Ч. 1. – 41 с.
8. Python. Головна сторінка [Електронний ресурс] – Режим доступу до ресурсу. URL: <https://www.python.org/> [дата звернення 24.05.2024]
9. Що таке SQLITE URL:<https://freehost.com.ua/ukr/faq/wiki/chto-takoe-sqlite/> [дата звернення 2.06.2024]

ДОДАТОК А

Конфігураційний файл шлюзового маршрутизатора Bogosyan_Router_4

```
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Bogosyan_Router_4
!
!
!
enable secret 5 $1$mERr$9cTjUIEqNGurQiFU.ZeCi1
!
!

ip dhcp excluded-address 10.24.16.129 10.24.16.134
ip dhcp excluded-address 10.24.16.129 10.24.16.134
ip dhcp excluded-address 10.24.16.1 10.24.16.128
!
ip dhcp pool LAN3
network 10.24.16.128 255.255.255.240
default-router 10.24.16.129
dns-server 10.24.16.130
!
!
aaa new-model
!
aaa authentication login CONSOLE group radius local
```

```
aaa authentication login default local
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
no ip cef
```

```
no ipv6 cef
```

```
!
```

```
!
```

```
!
```

```
username 123120_Bogosityan password 7 082048430017544541
```

```
!
```

```
!
```

```
license udi pid CISCO2911/K9 sn FTX15241914-
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
ip domain-name Bogosityan_Router_4
```

```
!
```

```
!
```

```
spanning-tree mode pvst
```

```
!  
!  
!  
!  
!  
!  
interface GigabitEthernet0/0  
  ip address 10.24.16.129 255.255.255.240  
  duplex auto  
  speed auto  
!  
interface GigabitEthernet0/1  
  no ip address  
  duplex auto  
  speed auto  
  shutdown  
!  
interface GigabitEthernet0/2  
  no ip address  
  duplex auto  
  speed auto  
  shutdown  
!  
interface Serial0/3/0  
  ip address 10.1.15.6 255.255.255.252  
!  
interface Serial0/3/1  
  ip address 10.1.15.13 255.255.255.252  
  clock rate 2000000  
!
```

```
interface Vlan1
  no ip address
  shutdown
!
router ospf 1
  log-adjacency-changes
  passive-interface GigabitEthernet0/0
  network 10.23.73.0 0.0.0.31 area 0
  network 10.1.15.4 0.0.0.3 area 0
  network 10.1.15.12 0.0.0.3 area 0
  network 10.24.16.128 0.0.0.15 area 0
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/3/0
!
ip flow-export version 9
!
!
!
banner motd 'Bogosyan_Router_4'
!
no radius server 10.23.73.25
  no address ipv4 10.23.73.25 auth-port 1645
  no key radius123
radius server 10.24.16.131
  address ipv4 10.24.16.131 auth-port 1645
  key radius123
!
!
!
```

```
line con 0
  password 7 0822455D0A16
  login authentication CONSOLE
!
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 телеграм-боту

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

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
РОЗРОБКА ТЕЛЕГРАМ-БОТУ

Текст програми

804.02070743.24002-01 12 01

Листів 4

2024

АНОТАЦІЇ

Додаток містить програмний код мовою Python який реалізує телеграм-бота. Бот автоматизує процес обміну документами всередині організації. Функціональність бота включає прийняття файлів від співробітників, які зберігаються у базі даних та надсилаються керівникам для розгляду. Керівництво, у свою чергу, може прийняти файл або відхилити. Також можна надати коментар щодо файлу, який буде надіслано співробітнику.

3MICT

main.py.....68

Текст програми

```
import sqlite3
from datetime import datetime

from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
from telegram.ext import Updater, MessageHandler, Filters, CallbackQueryHandler

# Set up the Telegram bot token
TOKEN = '6539864450:AAGfdFXqmQYUUQyJUjPU0QVS7Wqt31V4keY'

# Define arrays for management and employees
management = ['7165527638', '833385840']

# Create a SQLite3 database connection
conn = sqlite3.connect('filess.db', check_same_thread=False)
cursor = conn.cursor()

# Create a table to store file metadata
cursor.execute("""CREATE TABLE IF NOT EXISTS files
                (id INTEGER PRIMARY KEY AUTOINCREMENT, employee TEXT, employee_id
INTEGER, file_name TEXT, file_data BLOB, timestamp TEXT, accepted BOOLEAN, comment
TEXT)""")
conn.commit()

def handle_file(update: Update, context):
    if update.message.document:
        file = update.message.document
        sender_id = update.effective_user.id
        sender_name = update.effective_user.full_name
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

        # Save the file data in the database
        file_data = file.get_file().download_as_bytearray()
```

```

cursor.execute(
    "INSERT INTO files (employee, employee_id, file_name, file_data, timestamp, accepted)
VALUES (?, ?, ?, ?, "
    "?, ?)",
    (sender_name, sender_id, file.file_name, sqlite3.Binary(file_data), timestamp, False))
file_id = cursor.lastrowid
conn.commit()

# Send the file to management with metadata
caption = f"Файл: {file.file_name}\nБув відправлен: {sender_name}\nЧас: {timestamp}"
for manager_id in management:
    sent_file = context.bot.send_document(chat_id=manager_id, document=bytes(file_data),
                                          filename=file.file_name, caption=caption)
    context.bot.send_message(chat_id=manager_id, text="Оберіть дію:",
                              reply_markup=get_action_buttons(file_id, sent_file.message_id))

# Define a function to handle button clicks from management
def handle_button(update: Update, context):
    query = update.callback_query
    data = query.data.split('_')
    action = data[0]
    file_id = int(data[1])
    message_id = int(data[2])

    # Handle the action based on the button clicked
    if action == 'accept':
        cursor.execute("UPDATE files SET accepted = ? WHERE id = ?", (True, file_id))
        conn.commit()
        query.edit_message_text(text="✅ Файл погоджен")
        notify_employee(file_id, "✅ Ваш файл був погоджен.", context)
    elif action == 'reject-with-comment':
        query.edit_message_text(text="❌ Файл відмовлен. Введіть коментар:")

```

```

# Add a handler for the next message (comment)

context.user_data['comment_handler'] = MessageHandler(Filters.text & ~Filters.command,
handle_comment)

context.user_data['file_id'] = file_id

context.user_data['message_id'] = message_id

context.dispatcher.add_handler(context.user_data['comment_handler'])

elif action == 'reject':

    cursor.execute("UPDATE files SET accepted = ? WHERE id = ?", (False, file_id))

    conn.commit()

    query.edit_message_text(text=" ❌ Файл відмовлен")

    notify_employee(file_id, " ❌ Ваш файл був відмовлен.", context)

# Define a function to handle comments from management

def handle_comment(update: Update, context):

    comment = update.message.text

    file_id = context.user_data.get('file_id')

    cursor.execute("UPDATE files SET accepted = ?, comment = ? WHERE id = ?", (False, comment,
file_id))

    conn.commit()

    notify_employee(file_id, f" ❌ Ваш файл був відмовлен. Коментар: {comment}", context)

    context.dispatcher.remove_handler(context.user_data['comment_handler'])

    del context.user_data['comment_handler']

    del context.user_data['file_id']

    del context.user_data['message_id']

# Define a function to notify the employee of the decision

def notify_employee(file_id, message_text, context):

    cursor.execute("SELECT employee_id FROM files WHERE id = ?", (file_id,))

    employee_id = cursor.fetchone()[0]

    context.bot.send_message(chat_id=employee_id, text=message_text)

```

```

# Define a function to create the action buttons

def get_action_buttons(file_id, message_id):

    keyboard = [[InlineKeyboardButton("✅ Погодити",
callback_data=f'accept_{file_id}_{message_id}'),

                [InlineKeyboardButton("❌ Повернути з коментарем",
callback_data=f'reject-with-comment_{file_id}_{message_id}'),

                [InlineKeyboardButton("❌ Відмовити", callback_data=f'reject_{file_id}_{message_id}'), ]]

    return InlineKeyboardMarkup(keyboard)

# Set up the bot and handlers

updater = Updater(TOKEN, use_context=True)
dispatcher = updater.dispatcher

dispatcher.add_handler(MessageHandler(Filters.document, handle_file))
dispatcher.add_handler(CallbackQueryHandler(handle_button))

# Start the bot

updater.start_polling()

```