

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота магістра  
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА ТЕХНОЛОГІЯ НАЛАШТУВАННЯ  
МЕРЕЖІ АСИНХРОННОГО РЕЖИМУ ПЕРЕДАЧІ ДАНИХ З  
ВИКОРИСТАННЯМ МОБІЛЬНИХ ПРИСТРОЇВ**

Здобувач вищої освіти гр. ІН.м-13

Сергій ЛОГУТОВ

Науковий керівник,  
ст. викладач, к.ф.-м.н.

Дмитро ВЕЛИКОДНИЙ

В.о. завідувача кафедри  
доцент, к.т.н.

Ігор ШЕЛЕХОВ

# СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет ЕЛІТ Кафедра Комп'ютерних наук  
Спеціальність 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ:

в.о. зав. кафедри \_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Сергію Віталійовичу Логутову*

1. Тема роботи *Інформаційно-комунікаційна технологія налаштування мережі асинхронного режиму передачі даних з використанням мобільних пристроїв*

затверджую наказом по СумДУ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі здобувачем кваліфікаційної роботи \_\_\_\_\_

3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

*1) Вступ; 2) Огляд технологій; 3) Постановка завдання; 4) Вибір методу рішення; 5) Інформаційне та програмне забезпечення; 6) Висновок; 7) Література; 8) Додаток.*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р. Керівник \_\_\_\_\_

Завдання прийняв до виконання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання	Примітка
1	<i>Огляд існуючих технологій</i>		
2	<i>Вибір методу рішення</i>		
3	<i>Інформаційне та програмне забезпечення</i>		
4	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Здобувач вищої освіти \_\_\_\_\_

Науковий керівник \_\_\_\_\_

## РЕФЕРАТ

**Записка:** 78 ст., 25 рис., 1 додаток, 12 джерел.

**Об'єкт дослідження** — конфігурація роутерів в мережі асинхронного режиму передачі даних.

**Мета роботи** – розробка мобільного додатку для генерації скриптів налаштування роутерів в мережі асинхронного режиму передачі даних.

**Методи дослідження** – емпіричний метод, метод порівняння

**Результати** – розроблений мобільний додаток для платформ Android та IOS, що дозволяє згенерувати скрипт для кожного з роутерів мережі АТМ.

АТМ, JAVASCRIPT, REACT NATIVE, REACT, ANDROID, IOS, EXPO,  
ANDROID STUDIO, TELECOM, ПЕРСОНАЛЬНА ІС, РОУТЕР

## ЗМІСТ

ВСТУП.....	5
1 ОГЛЯД ТЕХНОЛОГІЙ.....	6
1.1 Технологія асинхронного режиму передавання .....	6
1.2 Операційна система Android .....	9
1.3 Операційна система IOS .....	11
1.4 Огляд мов програмування та фреймворків .....	14
1.4.1 Python.....	15
1.4.2 Kotlin.....	16
1.4.3 JavaScript .....	17
1.5 Вибір фреймворків та бібліотек JavaScript.....	19
1.5.1 React Native .....	19
1.5.2 React.js .....	21
1.6 Інструменти для візуалізації та розгортання мобільного додатку.....	24
1.6.1 Ехро.....	24
1.6.2 Android Studio .....	26
1.7 Постановка задачі .....	28
2 НАЛАШТУВАННЯ МЕРЕЖІ АСИНХРОННОГО РЕЖИМУ ПЕРЕДАВАННЯ В МЕРЕЖЕВОМУ ЕМУЛЯТОРІ GNS3 НА РЕАЛЬНОМУ ОБЛАДНАННІ CISCO .	29
2.1 Емулятор мереж GNS3 .....	29
2.2 Налаштування мережі на реальному обладнанні Cisco.....	33
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	35
3.1 Опис функціоналу .....	35
3.2 Тестування мобільного додатку.....	40
ВИСНОВОК.....	46
СПИСОК ЛІТЕРАТУРИ.....	47
ДОДАТКИ.....	49
Додаток А. Лістинг програмного коду .....	49

## ВСТУП

Сьогодні абсолютна більшість людей мають мобільні девайси (смартфони, планшети, годинники) і активно користуються ними. Пропозиція і попит на ринку смартфонів настільки великі, що кожного року продається близько мільярда одиниць. Через таку високу популярність та широкий функціонал мобільних телефонів, для все більшої кількості людей настільні комп'ютери стають не потрібними. Смартфони можуть виконувати дуже великий діапазон задач: від спілкування до розваг та від робочих до освітніх потреб.

Розробка мобільного додатку є основною метою даної роботи. Додаток повинен виводити на екран пристрою згенерований перелік команд для подальшого копіювання його в CLI роутерів в мережі ATM.

Розроблений додаток буде актуальним для людей, які вивчають телекомунікаційні технології. Необхідність розробки такого додатку викликана тим, що популярний емулятор для налаштування ATM мереж GNS3 не має графічного інтерфейсу.

## 1 ОГЛЯД ТЕХНОЛОГІЙ

### 1.1 Технологія асинхронного режиму передавання

Технологія асинхронного режиму передавання (англ. ATM – Asynchronous transfer mode) є найближчим конкурентом Ethernet. Створювалася дана технологія для мереж, що пропускають як високопродуктивний трафік, так і аудіо- та відеотрафіки в реальному часі та з незначною затримкою. Технологія забезпечує динамічну пропускну здатність, яка підходить для швидкісного трафіку. Всі дані закодовані в однакові комірки. Таким чином, передача даних є простою, рівномірною та передбачуваною. Уніфікований розмір пакета забезпечує ефективну обробку змішаного трафіку. Передається трафік за допомогою комірок (cells). Розмір комірки ATM становить 53 байти, 5 байт заголовка і 48 байт корисного навантаження (рис. 1.1). Невеликий розмір заголовка зменшує перевантаження пакетів, забезпечуючи ефективне використання пропускну здатності. Існує два різних формати комірок: інтерфейс користувач-мережа (UNI) та інтерфейс мережа-мережа (NNI) [1].



Рисунок 1.1 – Комірка

Кожен заголовок комірки ATM має VCI – ідентифікатор віртуального каналу, який є 16-бітним числовим полем, для ідентифікації логічного віртуального шляху (VPI – ідентифікатор віртуального шляху). Кожен VCI працює з VPI, щоб полегшити наскрізну передачу пакетів даних. Комбіновані цифрові ідентифікатори VCI та VPI забезпечують правильну передачу даних.

Еталонна модель ATM відрізняється від OSI та TCP/IP і складається з трьох рівнів (рис. 1.2):

- 1) Фізичний рівень – цей рівень аналогічний фізичному в моделі OSI. Тут комірки перетворюються на бітові потоки та передаються через фізичне середовище. Цей рівень має два підрівні: підрівень PMD (залежний від фізичного середовища) і підрівень TC (покриття передачі).
- 2) Рівень ATM – цей рівень можна порівняти з каналним рівнем моделі OSI. Він приймає 48-байтові сегменти з верхнього рівня, додає 5-байтовий заголовок до кожного сегмента та перетворює на 53-байтові комірки. Відповідає за маршрутизацію кожної комірки, керування трафіком, мультиплексування та комутацію.
- 3) Рівень адаптації ATM (AAL) – Цей рівень відповідає мережевому рівню моделі OSI. Він надає можливості існуючим мережам з комутацією пакетів підключатися до мережі ATM і використовувати її послуги. Він приймає дані та перетворює їх у сегменти фіксованого розміру. Передачі можуть бути з фіксованою або змінною швидкістю передачі даних. Цей рівень має два підрівні: підрівень конвергенції та підрівень сегментації та повторної збірки.
- 4) Кінцеві точки ATM – містять адаптер мережевого інтерфейсу ATM. Прикладами кінцевих точок є робочі станції, маршрутизатори, кодеки, комутатори локальної мережі тощо.
- 5) Комутатор ATM – передає клітинки через мережі ATM. Він приймає вхідні клітинки від кінцевих точок ATM (UNI) або іншого комутатора (NNI),

оновлює заголовок комірки та повторно передає клітинку до місця призначення [2].

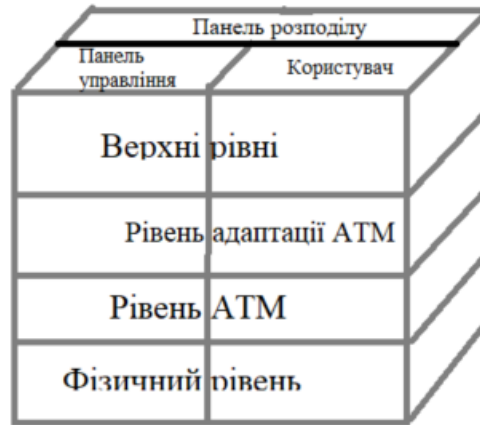


Рисунок 1.2 – Еталонна модель АТМ

Переваги технології:

- Орієнтованість на QoS і високу швидкість.
- Оптимізований для передачі голосу, даних і відео, тобто єдина мережа для всього;
- Легко інтегрувати з типами мереж LAN, WAN і MAN, тобто бездоганна інтеграція
- Використовує спрощену мережеву інфраструктуру.

Недоліки технології:

- Для досягнення QoS використовуються складні механізми;
- Комутатор АТМ дуже дорогий в порівнянні з LAN-обладнанням, крім того мережева карта (NIC) АТМ значно дорожче ніж Ethernet NIC.



## 1.2 Операційна система Android

Android – це мобільна операційна система на базі Linux, яка працює на смартфонах, планшетах тощо. ОС Android – це стек технологій з відкритим кодом, який складається з різних компонентів, які дозволяють розробникам і виробникам пристроїв працювати незалежно. Його можна розбити на п'ять основних частин — додатки, фреймворки додатків, рідні бібліотеки, середовище виконання Android і ядро Linux (рис. 1.3) [3].

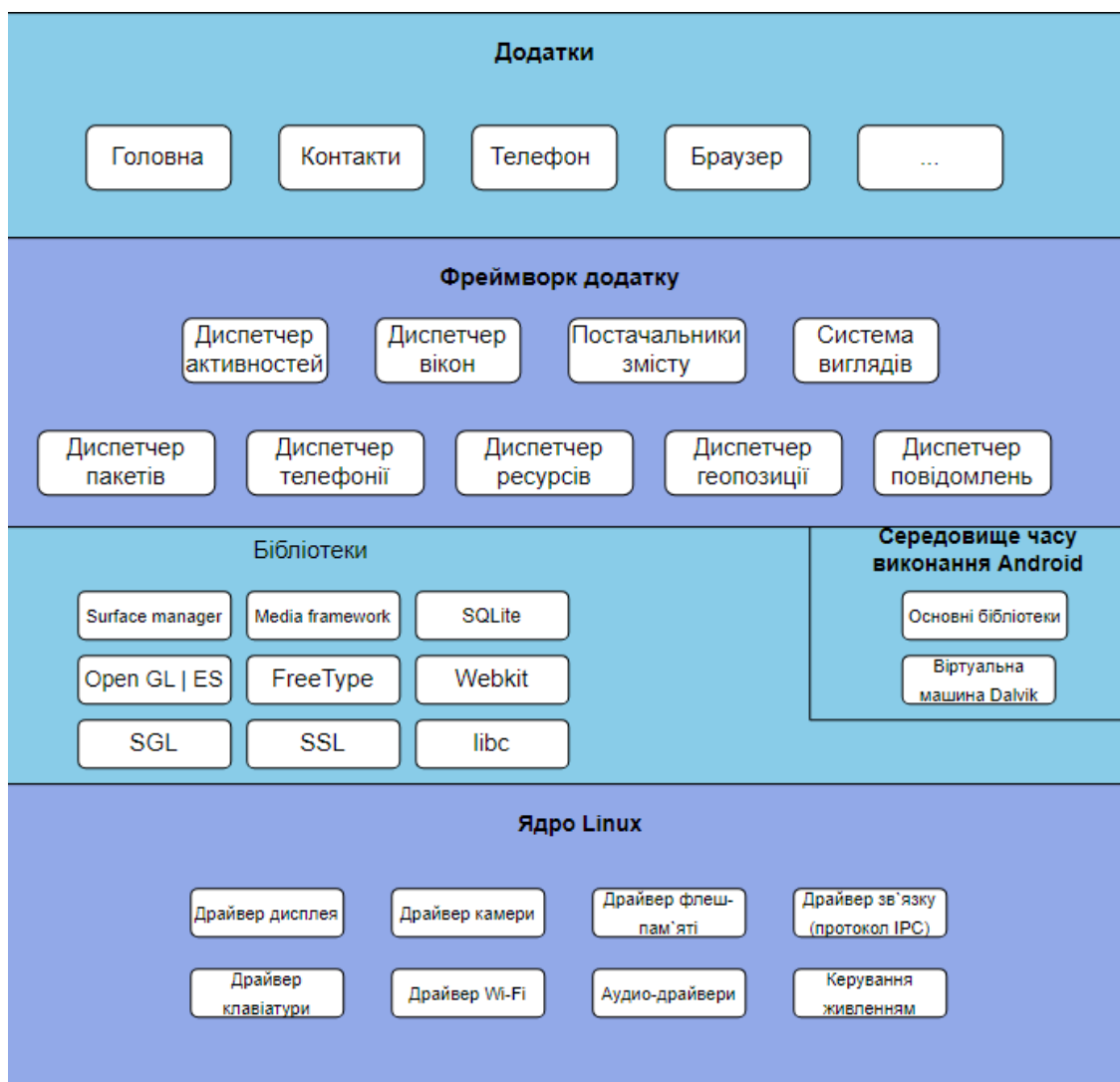


Рисунок 1.3 – Архітектура OS Android

Android відрізняється своєю невибагливістю і здатністю працювати на пристроях зі збіркою різних цінових сегментів, від флагманів з дорогою начинкою до бюджетників з більш дешевою. Саме це приваблює більшість виробників використовувати в своїх пристроях ОС Android.

Для Android найбільш вживаною мовою програмування є Java. Java – це мова ООП (об'єктно-орієнтоване програмування). Її використання є доцільним тому, що код написаний на Java працює на будь-якій платформі, що її підтримують [\[4\]](#).

### Переваги Android

- ОС Android є програмним забезпеченням з відкритим вихідним кодом, що означає, що ми можемо адаптувати його для власних потреб;
- У порівнянні з iPhone телефони Android значно дешевші та надають вам більше можливостей.
- Багато виробників мобільних телефонів обирають Android як свою операційну систему, оскільки вона має велику базу користувачів і її легко налаштувати для кожного.

### Недоліки Android

- Здатність користувача використовувати смартфон ускладнена персоналізацією. Також розробники дозволяють рекламу в індивідуальних інтерфейсах користувача.
- Незважаючи на те, що телефони Android безпечні, досвідчені хакери можуть просто зламати їх порівняно з iPhone.

### 1.3 Операційна система IOS

Операційна система IOS – це головний конкурент Android. Була розроблена компанією Apple лише для власного виробництва. Ключовими пристроями на IOS є смартфони iPhone. До 2019 року IOS також використовували планшети iPad, але потім було вирішено перевести їх на систему iPadOS. Є другою за популярністю операційною системою та займає приблизно 28% ринку згідно статистичних даних від Statcounter за 2022 рік(рис 1.4).

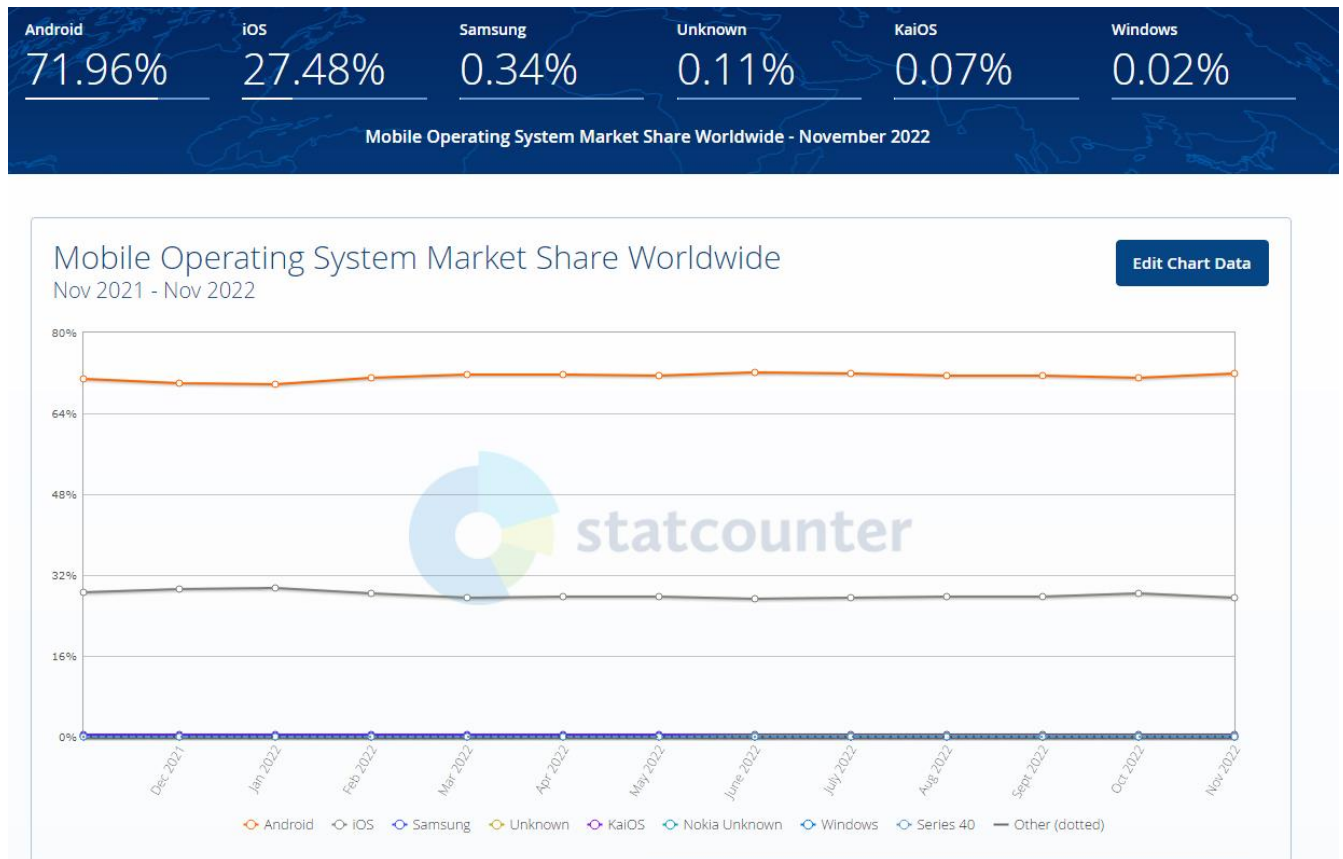


Рисунок 1.4 – Світовий ринок мобільних операційних систем

Код операційної системи є закритим для сторонніх розробників, лише деякі його частини є у вільному доступі. Код системи розроблена таким чином, що працювати вона може лише на девайсах компанії Apple і ні на чому більше, це принцип так званої пропріетарності. IOS написана мовою Swift, яка була розроблена та підтримується компанією Apple.

Архітектура IOS є багаторівневою архітектурою. На найвищому рівні iOS працює як посередник між базовим обладнанням і додатками, які ви створюєте. Програми не комунікують безпосередньо з основним обладнанням. Програми звертаються до обладнання через набір добре визначених системних інтерфейсів. Ці інтерфейси спрощують написання додатків, які постійно працюють на пристроях із різними апаратними можливостями. Нижчі рівні надають базові послуги, на які покладається вся програма, а вищий рівень надає складні графічні та інтерфейсні послуги. Apple надає більшість своїх системних інтерфейсів у спеціальних пакетах, які називаються фреймворками. Фреймворк – це каталог, який містить динамічну спільну бібліотеку, яка є файлами .a, пов'язаними ресурсами, наприклад файлами заголовків, зображеннями та допоміжними програмами, необхідними для підтримки цієї бібліотеки. Кожен рівень має набір Framework, який розробник використовує для створення програм (рис 1.5).

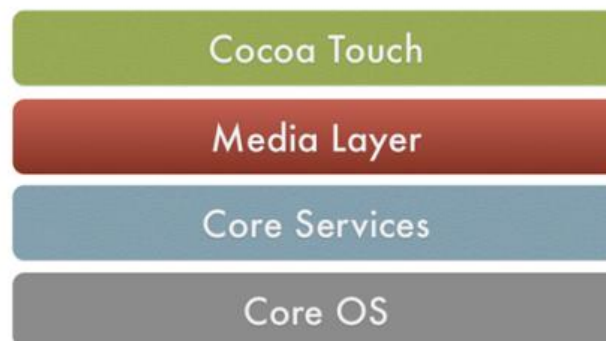


Рисунок 1.5 – Архітектура операційної системи IOS

## Переваги IOS

- IOS добре відома безпекою, яку вона забезпечує своїм клієнтам. З точки зору безпеки, це кращий варіант, ніж Android.
- Apple не робить жодних компромісів у виробництві iPhone, використовуючи лише найякісніші матеріали. Отже, має чудові технічні характеристики.
- Apple надає оновлення програмного забезпечення для всіх IOS одночасно, незалежно від того, старий телефон чи новий. Крім того, вони можуть надавати оновлення безпеки та оновлення ОС для будь-якого iPhone протягом тривалого періоду часу.

## Недоліки IOS

- Існує обмежена кількість програм для iOS, і це свого роду програмне забезпечення із закритим кодом
- В iOS є дорогі програми та відсутня підтримка віджетів.
- IOS підтримується тільки внутрішніми пристроями компанії Apple, тобто iPhone

## 1.4 Огляд мов програмування та фреймворків

Для створення мобільного додатку важливо зосередитися на виборі правильної технології і правильної мови програмування. Маючи на вибір кілька мов програмування для мобільних пристроїв, потрібно врахувати численні фактори, плюси та мінуси, а також популярність мови, перш ніж прийняти рішення створити мобільний додаток.

Мова програмування — це комп'ютерна мова, яка використовується програмістами (розробниками) для спілкування з комп'ютерами. Це набір інструкцій, написаних будь-якою конкретною мовою (C, C++, Java, Python) для виконання конкретного завдання.

Мови програмування класифікуються за рівнями:

1) Мови низького рівня:

- a) Машинна мова – різновид мови програмування низького рівня. Її також називають машинним кодом або об'єктним кодом. Комп'ютеру легше читати машинну мову, оскільки вона зазвичай відображається у двійковій або шістнадцятковій формі. Для перетворення програм не потрібен перекладач, оскільки комп'ютери безпосередньо розуміють програми машинної мови. Перевага машинної мови полягає в тому, що вона допомагає програмісту виконувати програми швидше, ніж мова програмування високого рівня.;
- b) Мова асемблера (ASM) також є різновидом мови програмування низького рівня, яка розроблена для конкретних процесорів. Він представляє набір інструкцій у символічній та зрозумілій людині формі. Він використовує асемблер для перетворення мови асемблера в машинну мову. Перевага мови асемблера полягає в тому, що для виконання програми потрібно менше пам'яті та менше часу на виконання.

- 2) Мова програмування високого рівня Мова програмування високого рівня (HLL) призначена для розробки зручних програм і веб-сайтів. Ця мова програмування потребує компілятора або інтерпретатора для переклада програми на машинну мову (виконання програми). Головна перевага мови високого рівня полягає в тому, що її легко читати, писати та підтримувати. Мова програмування високого рівня включає Python, Java, JavaScript, PHP, C#, C++, Swift та багато інших [5].

#### 1.4.1 Python

Широко визнана мова програмування Python використовується для розробки веб-додатків, настільних програм, медіа-інструментів, машинного навчання, мережевого сервера тощо. Прикладами проектів реалізованих за допомогою Python є YouTube, Instagram і Pinterest [6]. Мова забезпечує підтримку бібліотек, надійну інтеграцію та розширені можливості керування. Серед особливостей мови Python виділяють:

- Забезпечення швидкої розробки додатків;
- здатність «склеювати» існуючі сценарії один з одним;
- Читабельність і чистота коду;
- Можливість створити GUI за допомогою однієї з багатьох стандартних бібліотек.

#### Переваги Python

- Простий у використанні, легкий у навчанні та добре читається;
- Інтерактивна мова, яка використовується для створення масштабованих програм і підтримує програми GUI;
- Має величезний набір фреймворків з гнучким програмуванням.

## Недоліки Python

- Повільний час виконання.
- Не дуже добре з розробкою сучасних мобільних додатків;
- Не найкращий вибір із завданнями, що потребують пам'яті;
- Має деякі недоліки з доступом до даних.

### 1.4.2 Kotlin

Kotlin є статистично типізованою мовою програмування, яка використовується для розробки сучасних програм Android. Kotlin має потенціал впливати на інші мови програмування, такі як Java, для створення високопродуктивних і надійних програм. Деякі популярні програми, побудовані на Kotlin, це Trello, Evernote, Coursera та багато інших [6]. Kotlin це мова програмування з відкритим кодом, яка скорочує час запуску Android, має чистий і компактний синтаксис.

#### Переваги Kotlin

- Чистий, стислий і проникливий синтаксис;
- Функціонально сумісний і універсальний, і може легко усунути недоліки Java;
- Має повну підтримку від Google і інсталяційних пакетів IDE, включаючи Android і інструментарій SDK;
- Генерує компактний, простий і чистіший код порівняно з Java.



## Недоліки Kotlin

- Все ще нова мова на ринку, яку важко вивчити;
- Іноді стає повільніше;
- Ресурси для навчання обмежені.

### 1.4.3 JavaScript

Мова програмування JavaScript є текстовою і може використовуватися як на стороні клієнта, так і на стороні сервера. Вона керує мультимедіа на веб-сторінках і дозволяє їм стати інтерактивними. JavaScript дозволяє розробнику виконувати багато речей, наприклад додавати анімацію до зображень або автоматично оновлювати вміст на сторінці [7]. JavaScript є легкою, незалежною від платформи, підтримує динамічний тип і ООП, може перевіряти дані користувача та обчислення на стороні клієнта.

#### Переваги JavaScript

- Універсальна та гнучка мова, яку можна використовувати багатьма способами через Node.js;
- Проста в вивченні;
- Швидкий перегляд на стороні клієнта, не потребує компіляції;
- Простий у роботі та простий у керуванні;
- Немає встановлених стандартів і достатньо місця для варіацій.

## Недоліки JavaScript

- Є вразливим і в деяких випадках може бути використаний для зловмисних цілей;
- У деяких випадках можуть виникнути проблеми з підтримкою браузера;
- Сценарії на стороні сервера завжди забезпечують однаковий вихід, а сценарій на стороні клієнта дещо непередбачуваний.

Проаналізувавши найпопулярніші мови програмування для розробки мобільних додатків, було вирішено віддати перевагу мові JavaScript.

## 1.5 Вибір фреймворків та бібліотек JavaScript

Хоча JavaScript сам по собі достатній для створення будь-якої веб-програми з нуля. Він має всю необхідну бібліотеку, необхідну для створення будь-якої функції, необхідної для нормальної роботи програми. Однак створення цих функцій кожного разу, коли потрібно розробити програму, може бути напруженим і трудомістким, тому ідея шаблону спрощує процес розробки.

Фреймворки схожі на набір шаблонів для створення певних функцій програми. Замість того, щоб створювати функцію з нуля, можна звернутися до відповідних бібліотек та отримати бажаний результат якомога швидше і якісніше.

Крім того, фреймворк JavaScript — це набір бібліотек JavaScript для створення заздалегідь написаного коду для щоденних завдань програміста. Для розробки додатку було обрано власне мову JavaScript, фреймворк React Native та JavaScript-бібліотеку React.js [\[8\]](#).

### 1.5.1 React Native

React Native — це платформа з відкритим вихідним кодом, створена Facebook на JavaScript. Це дозволяє створювати мобільні програми як для операційних систем Android, так і для iOS. Додатки створені на React Native можуть працювати на будь-якому пристрої та використовувати лише одну кодову базу.

Фреймворк заснований на середовищі виконання JavaScriptCore і трансформаторах Babel. Завдяки такому налаштуванню RN підтримує нові функції JavaScript (ES6+), наприклад стрілочних функцій, async/wait тощо.

Цей знаменитий фреймворк для розробки мобільних додатків з'явився в 2013 році як внутрішній проєкт хакатону Facebook. Його перша публічна версія була випущена в січні 2015 року на конференції Reactjs, а в березні 2015 року Facebook зробив React Native відкритим і доступним на GitHub [\[10\]](#).

## Переваги React Native

- Швидкість і вартість розробки додатків. Що подобається розробникам у React Native, так це те, що вони можуть повторно використовувати код і переробляти компоненти, розроблені ними раніше, і широку екосистему React Native. Вони також можуть поділитися своєю кодовою базою та пришвидшити роботу. Більше того, розробники React Native займаються розробкою додатків для обох платформ – iOS та Android, тому вам не потрібно наймати дві окремі команди.
- Швидший час виходу на ринок – ви можете вийти на ринок набагато швидше, щоб протестувати свій MVP, отримати відгук і адаптувати зміни відповідно без великих інвестицій.
- Допомога за запитом – спільнота React Native величезна. Багато проблем, з якими ви можете зіткнутися під час розробки, можливо, уже виправлені членами спільноти.
- Вартість обслуговування – ви маєте справу лише з однією кодовою базою.
- Мобільні програми React Native видимі – ваші програми легко розмістити в AppStore і Play Store. Не так, як PWA.

## Недоліки React Native

- Продуктивність все ще нижча, ніж нативна – React Native не може використовувати всі переваги та потенціал певної платформи. З іншого боку, нативна програма дійсно може максимізувати функціональні можливості та забезпечити максимальну взаємодію з користувачем.
- Тим не менш, змінена архітектура докладає багато зусиль, щоб зробити RN більш продуктивним, напр. доступний новий двигун JS – Hermes, який робить програми, що працюють набагато швидше на старих пристроях Android.
- Неefективно для складних інтерфейсів.
- Відсутність деяких користувацьких модулів – багато користувацьких модулів уже доступні, але може виникнути потреба в деяких особливих компонентах, які вам доведеться створювати з нуля самостійно.
- Проблеми з оновленням – важко оновлювати програму до останньої версії React Native. Оновлення версій React Native у більшості випадків є складним процесом [\[9\]](#).

Так як потрібно розробити мобільний додаток для операційних систем Android та IOS, було вирішено обрати саме цей фреймворк, адже він не потребує якихось індивідуальних змін в коді для кожної з систем.

### 1.5.2 React.js

React.js – це відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту

вебсторінки, з якими стикаються в розробці односторінкових застосунків. Розробляється Meta (раніше Facebook) і спільнотою індивідуальних розробників.

## Переваги

- *Велика спільнота.* У комерційній розробці це одна з найважливіших речей, коли ми обираємо використовувати наші бібліотеки, інструменти тощо. Якщо певна технологія має багато шанувальників, це полегшує та прискорює процес розробки. Коли ви стикаєтеся з будь-якою помилкою, ви можете запитати на публічному форумі, і є висока ймовірність, що серед мільйонів розробників React деякі з них вирішували цю помилку раніше. Завдяки нашій великій спільноті ми також маємо безліч сторонніх бібліотек React, посібників і заходів.
- *Зворотна сумісність.* Це одна з найкращих речей React, що він не вносить критичних змін. Після кожного оновлення React API залишається майже незмінним.
- *Багаторазові компоненти.* Розробники React можуть створювати багаторазові компоненти. Це дозволяє розробникам створювати невеликі фрагменти інтерфейсу користувача та включати їх у будь-яке місце програми. Ви можете поєднати багато менших компонентів, щоб створити більший складний інтерфейс користувача. Досить часто розробники створюють додаток, використовуючи готові компоненти інтерфейсу користувача, наприклад..
- *Віртуальний DOM.* Virtual DOM (Document Object Model) — відповідає за підтримку абстракції програми React і синхронізацію в актуальному стані з реальним DOM (це те, що ми дійсно бачимо на нашому дисплеї). Це дозволяє покращити продуктивність і швидкість програми, оскільки реалізація React Virtual DOM порівнює абстракцію програми між змінами та повторно відображає лише змінені частини програми.

## Недоліки

- *Програми React засновані на сторонніх бібліотеках*

Це одночасно і перевага, і недолік. React — це «просто бібліотека», і немає офіційних бібліотек для обробки загальних функцій інтерфейсних додатків, як напр. маршрутизація, http-запити тощо. Це перевага, оскільки у вас є свобода, і якщо ви обізнаний розробник JS, ви можете вибрати найкращі інструменти для своїх потреб. Також це недолік, тому що React не має багатьох базових інструментів із коробки. Навіть якщо ви використовуєте лише найпопулярніші та найпоширеніші бібліотеки, у вас може виникнути конфлікт версій React між версією бібліотеки та вашою версією програми React (здебільшого йдеться про застарілі програми). Якщо ви хочете підтримувати свою програму в актуальному стані з версією React, майте на увазі, що може знадобитися оновлення сторонніх бібліотек. Подібно до того, як React має чудову зворотну сумісність, деякі бібліотеки (навіть дуже популярні) можуть зазнати деяких критичних змін.

- *Жодної конвенції* React не нав'язує жодних домовленостей щодо розробки.

Щоб познайомити нових розробників із застарілим проектом, потрібно більше часу, оскільки дуже часто розробники JavaScript створюють програми React різними способами, і існує висока ймовірність того, що ви зіткнетесь зі застарілим проектом із умовами кодування, яких раніше не бачили в React. Це змушує команду розробників витрачати час на обговорення деяких загальних правил розробки в проекті. Коли команда змінюється багато разів, це може призвести до безладного коду, оскільки всі мають різні звички та дотримуються різних умов [\[11\]](#).

## 1.6 Інструменти для візуалізації та розгортання мобільного додатку

Зрозуміло, що розробляти додаток безпосередньо на мобільному пристрої не зручно і складно. Також не завжди під рукою є пристрої обох операційних систем, тобто iOS та Android. Для того, щоб слідкувати за тим, як будується та виглядає мобільний додаток є спеціальні інструменти, серед них – фреймворк Expo та інтегрована середовище розробки (IDE) Android Studio.

### 1.6.1 Expo

Expo — це пакет npm, який надає набір неймовірних функцій для програм React Native. Пакет expo можна встановити майже в будь-який проєкт React Native. Він дозволяє програмісту запустити його проєкт на стадії розробки, щоб відстежувати UI, змінювати або виправляти помилки в стилі та функціоналі додатку в режимі реального часу. Створивши додаток в терміналі командою «`npm create-expo-app назва проєкту`» потрібно ввести одну з наступних команд «`npm start`» або «`expo start`» (рис. 1.6). Після виконаних дій можна побачити варіанти підключення до додатку. Зробити це можна за допомогою віртуального девайсу в Android Studio або використовуючи реальний пристрій, але для цього потрібно буде встановити офіційний додаток «Expo» для Android в Play Market (рис. 1.7) або «Expo Go» в App Store (рис. 1.8) для iOS. Приєднатись до локального серверу можна використовуючи QR-код або IP-адресу.



```
PS C:\Users\sergl\ATMGui> npm start
> atmgui@1.0.0 start
> expo start

Starting project at C:\Users\sergl\ATMGui
Starting Metro Bundler



> Metro waiting on exp://192.168.1.2:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
```

Рисунок 1.6 – Команда expo start та посилання на приєднання до локального серверу

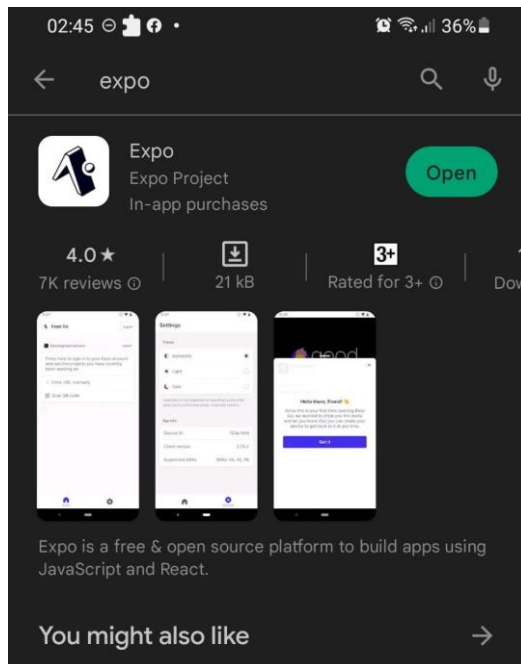


Рисунок 1.7 – сторінка Expo в Play Market

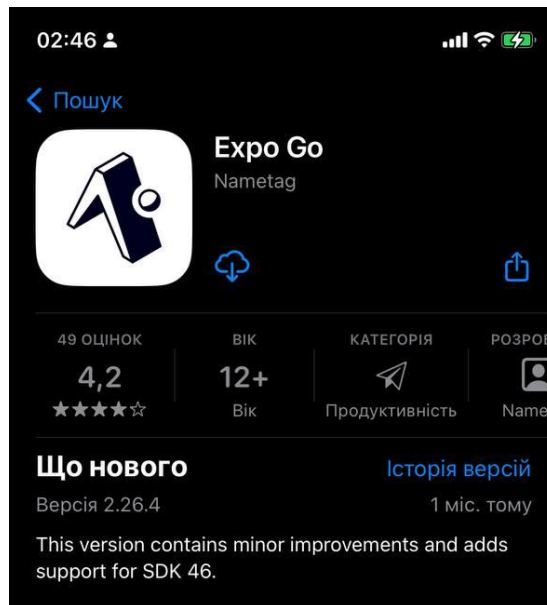


Рисунок 1.8 – сторінка Expo Go в App Store

### 1.6.2 Android Studio

Android Studio є офіційним інтегрованим середовищем розробки (IDE) для розробки додатків Android. Android Studio надає більше функцій, які підвищують нашу продуктивність під час створення програм для Android. Середовище також має набір віртуальних мобільних пристроїв (рис 1.9) та різні версії операційної системи Android.

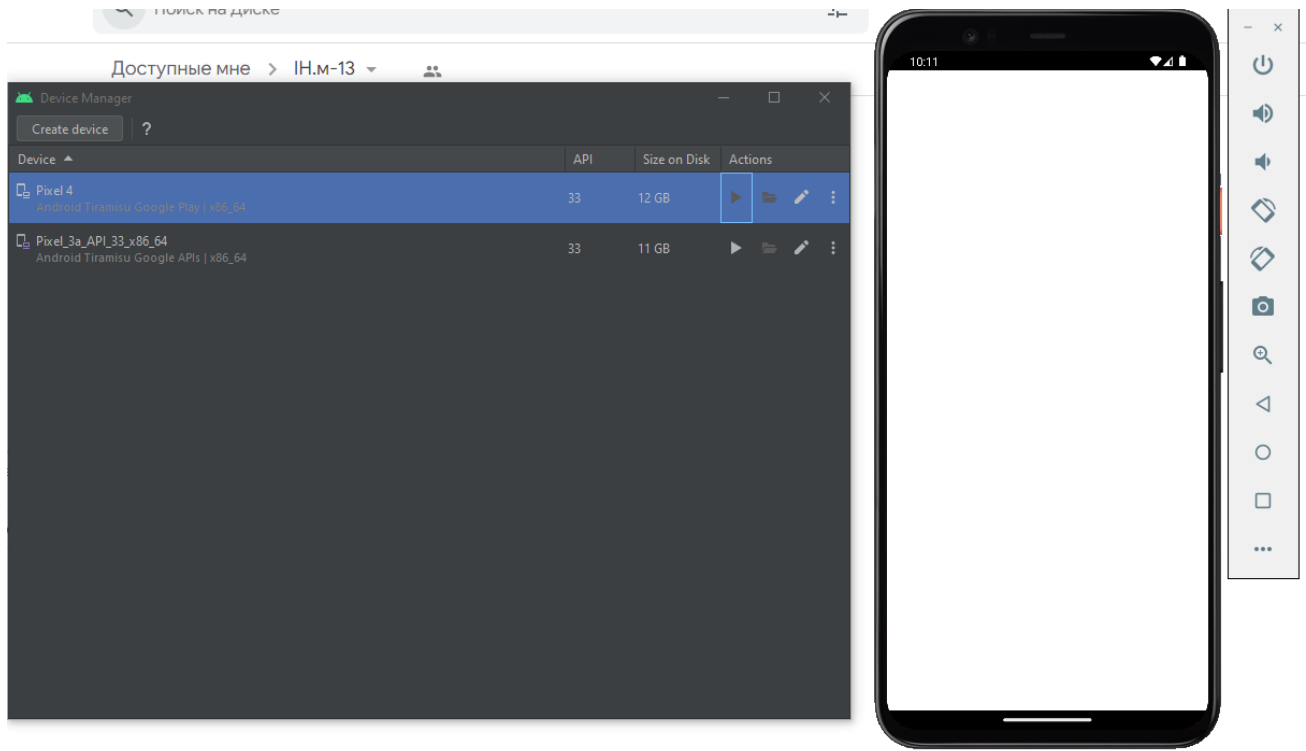


Рисунок 1.9 – Віртуальний пристрій в Android Studio

## 1.7 Постановка задачі

Метою роботи є розробка мобільного додатку для операційних систем iOS та Android, який виконує функції генератора скриптів для налаштування роутерів в мережі асинхронного режиму передавання. Основним сегментом користувачів виступатимуть студенти або люди, які самостійно вивчають телекомунікаційні-технології, а саме тему «Технологія ATM». Додаток повинен давати користувачеві змогу копіювати скрипти налаштувань після введення необхідних IP-адрес роутерів Cisco в мережі. Після копіювання внести перелік команд до CLI відповідного роутера в емуляторі мереж GNS3 або на реальному обладнанні.

Інтерфейс має бути лаконічним та зрозумілим для новачків та мати відповідну інструкцію з експлуатації.

Задля досягнення мети роботи було визначено наступні етапи:

- 1) Перегляд та аналіз доступних технологій, які можуть бути використані для розробки додатку;
- 2) Власне розробка мобільного додатку – генератора скриптів налаштування роутерів Cisco в мережі асинхронного режиму передавання;
- 3) Подальше тестування програми.

## 2 НАЛАШТУВАННЯ МЕРЕЖІ АСИНХРОННОГО РЕЖИМУ ПЕРЕДАВАННЯ В МЕРЕЖЕВОМУ ЕМУЛЯТОРІ GNS3 НА РЕАЛЬНОМУ ОБЛАДНАННІ CISCO

### 2.1 Емулятор мереж GNS3

При вивченні Telecom-технологій або розробці мереж не завжди доцільно працювати на реальному обладнанні через його високу вартість та необхідність в обслуговуванні. Для спрощення розуміння роботи мереж і були розроблені мережеві емулятори та симулятори. Симулятори та емулятори – це, програмні інструменти, які імітують роботу тієї чи іншої програми, операційної системи тощо. В даному випадку існує дві програми – це Cisco Packet Tracer та GNS3. Cisco Packet Tracer – симулятор, який надає можливість створювати макети мереж та перевіряти їх на працездатність. Проте не дивлячись на досить багатий функціонал, в ньому відсутні необхідні для мереж асинхронного режиму передавання віртуальні пристрої.

GNS3 – це, мережевий емулятор, написаний на Python та реалізований в 2008 році. Дозволяє комбінувати реальне обладнання з віртуальним та використовувати його для імітації комплексних мереж. GNS3 використовує Dynamips – програма-емулятор для емуляції маршрутизаторів Cisco, яка емулює апаратне забезпечення платформ маршрутизації серії Cisco шляхом безпосереднього завантаження фактичного образу програмного забезпечення Cisco IOS в емулятор. Dynamips емулює платформи Cisco 1700, 2600, 2691, 3600, 3725, 3745 і 7200 [\[12\]](#).

#### Переваги GNS3

- Безкоштовний та має відкритий код;
- Відсутність обмежень щодо кількості підтримуваних пристроїв

- Підтримує кілька варіантів комутації
- Підтримує пристрої багатьох брендів;
- Підтримує програми віртуалізації різних операційних систем (VirtualBox, VMware Workstation, ESXi);
- Велика та активна спільнота.

### Недоліки GNS3

- Образи iOS завантажуються користувачем самостійно;
- Потрібно вручну додавати образ маршрутизатора/комутатора, який завантажується;
- Програма працює на основі продуктивності апаратного забезпечення вашого комп'ютера.

### Моделювання мережі

Був обраний образ маршрутизатора Cisco-7200 та доданий інтерфейс RA-A1 – це портовий адаптер, який забезпечує єдиний мережевий інтерфейс ATM для підтримуваних платформ. Потім було налаштовано порти комутатора, призначено відповідні VCI (віртуальні канали) та VPI (віртуальні шляхи) (рис 2.1). Після налаштування та підключення всіх маршрутизаторів отримано схему (рис. 2.2).

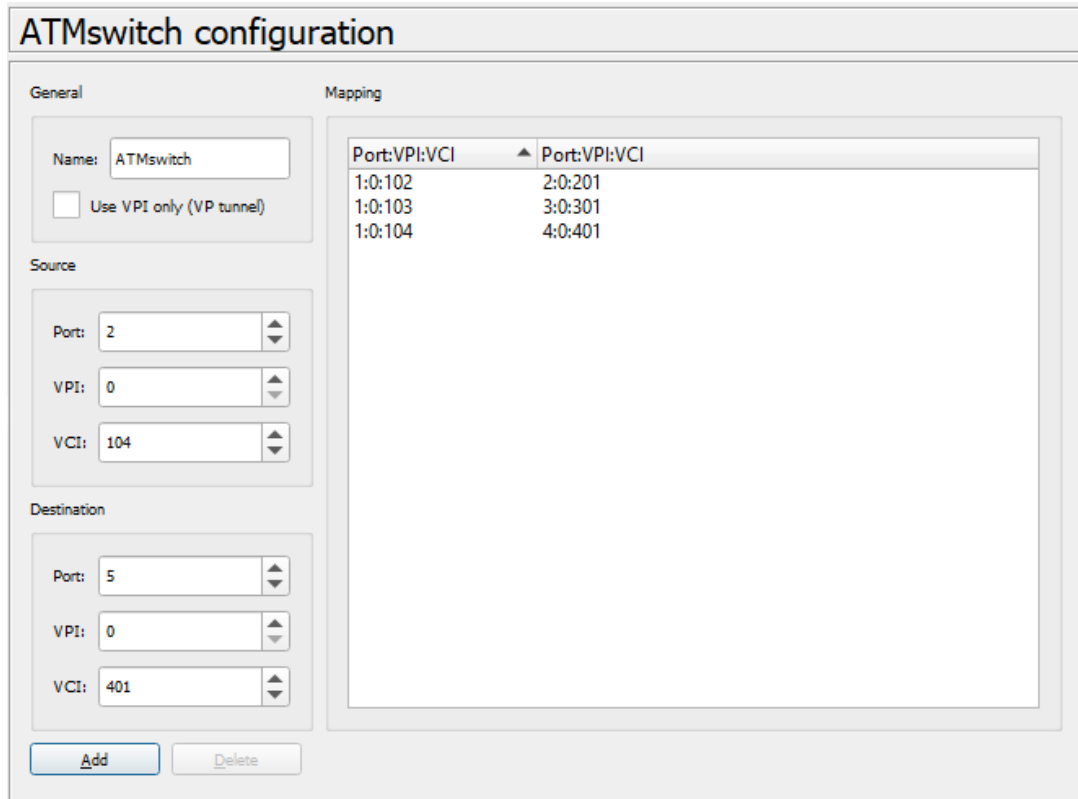


Рисунок 2.1 – Налаштування топографії портів комутатора

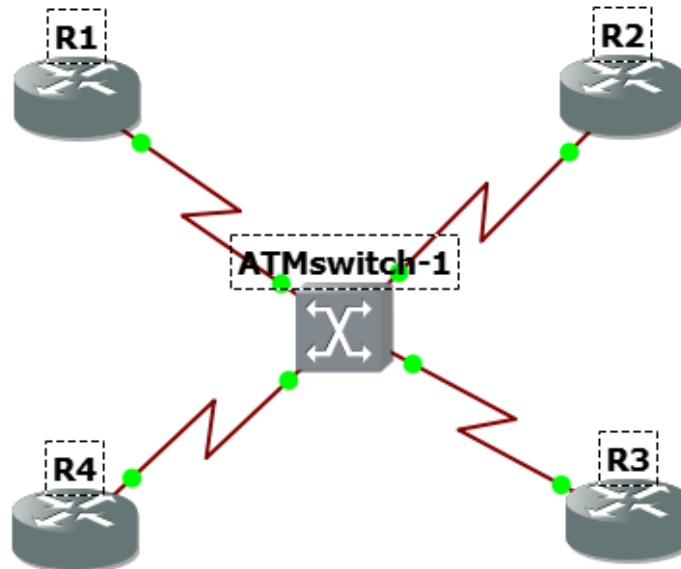


Рисунок 2.2 – мережа побудована за допомогою асинхронного режиму передавання

Тип з'єднання для роутера R1 – point-to-multipoint (точка-багатоточка), це дозволило «познайомити» його одразу з трьома роутерами R2, R3, R4 (рис. 2.3).

```

Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int atm 1/0
R1(config-if)#no sh
R1(config-if)#ex
R1(config)#int atm1/0.1 multipoint
R1(config-subif)#ip add 192.168.1.1 255.255.255.0
R1(config-subif)#pvc 0/102
R1(config-if-atm-vc)#encapsulation aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.1.2 broadcast
R1(config-if-atm-vc)#pvc 0/103
R1(config-if-atm-vc)#encap aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.1.3 broadcast
R1(config-if-atm-vc)#pvc 0/104
R1(config-if-atm-vc)#encap aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.1.4 broadcast
R1(config-if-atm-vc)#end
R1#
*Dec 12 23:32:34.967: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
R1#
*Dec 12 23:32:35.203: %SYS-5-CONFIG_I: Configured from console by console
*Dec 12 23:32:35.967: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R1#

```

Рисунок 2.3 – CLI роутера R1

Роутери R2, R3, R4 були налаштовані аналогічно один з одним методом point-to-point (точка-точка) та були з'єднані з роутером R1 (рис. 2.4, рис. 2.5, рис. 2.6).

```

Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int atm1/0
R2(config-if)#no sh
R2(config-if)#ex
R2(config)#int atm 1/0.1 point-to-point
R2(config-subif)#ip add 192.168.1.2 255.255.255.0
R2(config-subif)#pvc 0/201
R2(config-if-atm-vc)#encap aal5snap
R2(config-if-atm-vc)#end
R2#
*Dec 12 23:36:31.351: %SYS-5-CONFIG_I: Configured from console by console
R2#
*Dec 12 23:36:31.831: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
*Dec 12 23:36:32.831: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R2#

```

Рисунок 2.4 – CLI роутера R2



```

Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int atm1/0
R3(config-if)#no sh
R3(config-if)#ex
R3(config)#int atm1/0.1 point-to-point
R3(config-subif)#ip add 192.168.1.3 255.255.255.0
R3(config-subif)#pvc 0/301
R3(config-if-atm-vc)#encap aal5snap
R3(config-if-atm-vc)#end
R3#
*Dec 12 23:38:29.627: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
R3#
*Dec 12 23:38:30.499: %SYS-5-CONFIG_I: Configured from console by console
*Dec 12 23:38:30.627: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R3#

```

Рисунок 2.5 – CLI роутера R3

```

R4(config)#int atm1/0
R4(config-if)#no sh
R4(config-if)#ex
R4(config)#int atm1/0.1 point-to-point
R4(config-subif)#ip add 192.168.1.4 255.255.255.0
R4(config-subif)#pvc 0/401
R4(config-if-atm-vc)#encap aal5snap
R4(config-if-atm-vc)#end
R4#
*Dec 12 23:40:43.863: %SYS-5-CONFIG_I: Configured from console by console
R4#
*Dec 12 23:40:43.903: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
*Dec 12 23:40:44.903: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R4#

```

Рисунок 2.6 – CLI роутера R4

Отже, маємо мережу 192.168.1.0/24, яка складається з чотирьох роутерів, з'єднаних між собою з використанням технології АТМ через порти в комутаторі АТМ.

## 2.2 Налаштування мережі на реальному обладнанні Cisco

Cisco Systems – це американська компанія, яка створювалася з метою виробництва та реалізації мережного обладнання. Компанія також надає ряд послуг з обслуговування свого обладнання.

Комутатор ATM-Switch та маршрутизатор Cisco-7200, які використовуються в даній роботі також були розроблені корпорацією Cisco.

Для того щоб прискорити конфігурування роутерів в мережі ATM, було розроблено мобільний додаток, що виконує генерацію скриптів, для операційних систем Android та iOS.

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

Розробка додатку велася за наступними етапами:

- 1) Створення проєкту з використанням Ехро;
- 2) Розгортання проєкту;
- 3) Тестування додатку

#### 3.1 Опис функціоналу

Після запуску додатку можна побачити інструкцію з налаштування мережі асинхронного режиму передавання в емуляторі мереж GNS3 (рис. 3.1). Нижче можна побачити випадуючий список з варіантами кількості роутерів в мережі (рис. 3.2). Обравши одну із опцій, з'являється картинка, на якій зображено приклад налаштування мережі АТМ з відповідною кількістю маршрутизаторів. Також з'являється і поля вводу ІР-адрес для кожного з них (рис. 3.3). Кожне поле текстового вводу провалідовано регулярним виразом, який записано в змінну `regex`:

```
let regex = new RegExp(
    /^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/gi
);
```

Таким чином не дозволяється додати в перелік команд нічого окрім ІР-адрес. При невиконанні умови регулярного виразу, під кожним неправильно заповненим текстовим полем з'являється текст-попередження червоного кольору «IP address is invalid!» Для зкорочення частоти таких випадків до кожного поля була додана властивість, яка дозволяє писати тільки чисельні значення (рис 3.4).

```
keyboardType="numeric"
```

Якщо всі вихідні дані були введені без помилок можна приступити до генерації переліку команд налаштування, натиснувши на кнопку «Get Script!». Після натискання з'являється модальне вікно зі списком необхідних команд, які можна скопіювати та відправити на комп'ютер для подальшого використання скриптів (рис. 3.5). Можливість копіювати текст була реалізована властивістю тега <Text>.

```
selectable={true}
```

Якщо потрібно повернутися до головного вікна, необхідно натиснути кнопку «Close Script», або натиснути кнопку «Назад» самого пристрою.

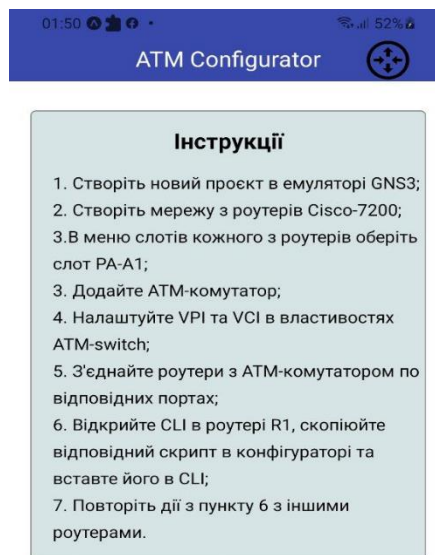


Рисунок 3.1 – Інструкція створення мережі в GNS3

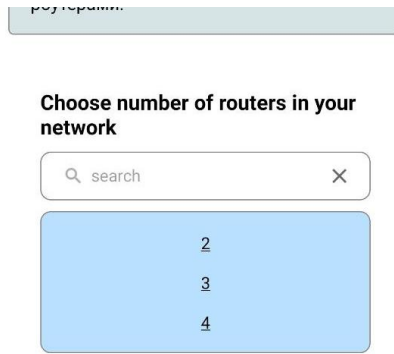


Рисунок 3.2 – Випадаючий список

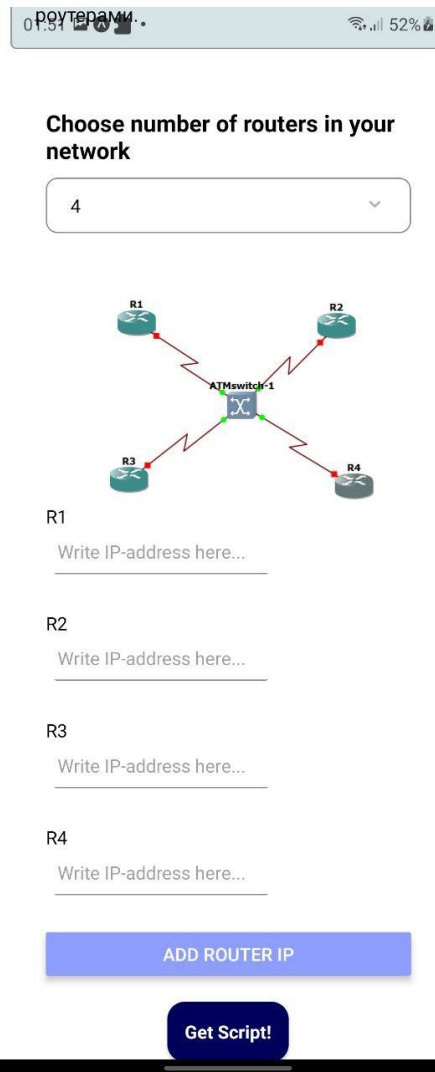


Рисунок 3.3 – Зображення мережі та поля вводу IP-адрес

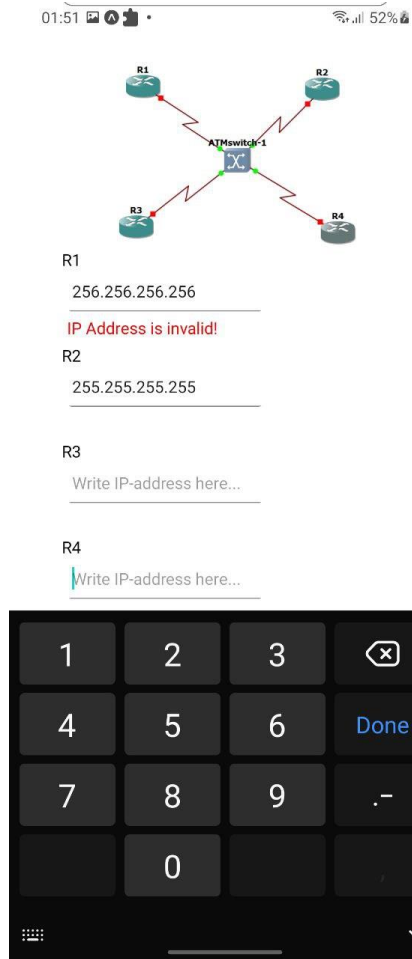


Рисунок 3.4 – Валідація текстового поля

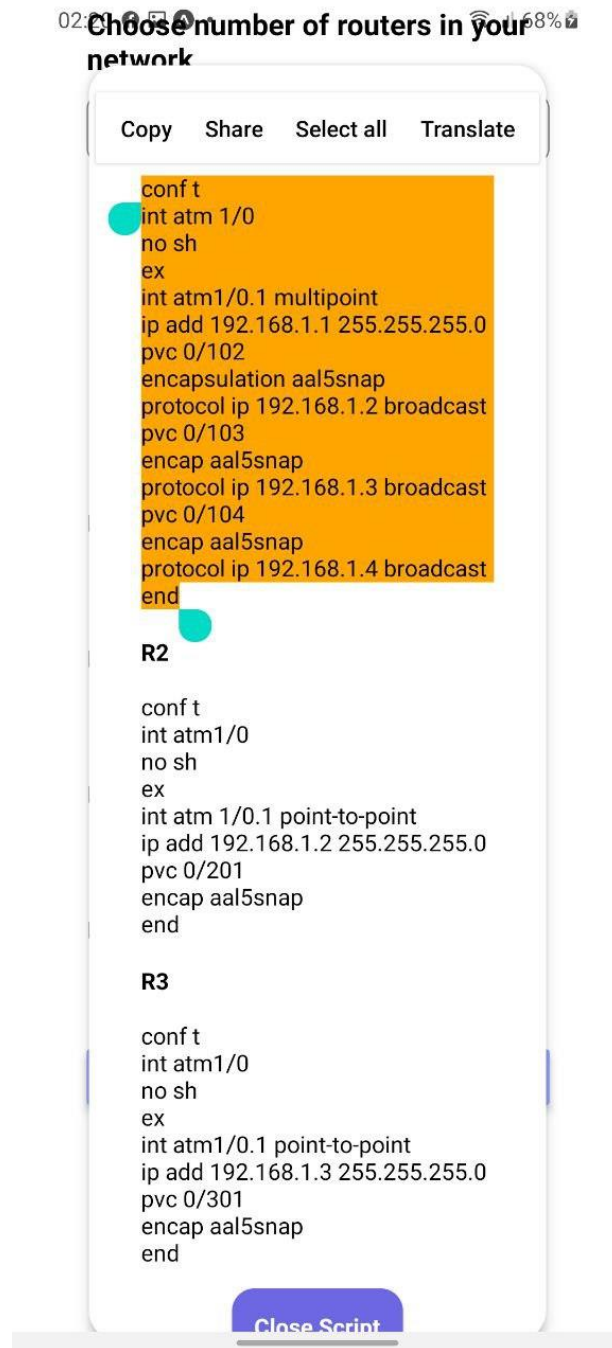


Рисунок 3.5 – Згенерований список команд для налаштування маршрутизаторів

### 3.2 Тестування мобільного додатку

Щоб впевнитись в тому, що додаток виконує поставлену задачу, потрібно побудувати мережу асинхронного режиму передавання та перевірити правильність згенерованих команд в емуляторі GNS3.

Після запуску розробленого додатку переходимо до випадючого списку <SelectList> та обираємо «4», на екрані отримуємо 4 графи введення IP-адреси для кожного окремого роутера та схематичне зображення мережі. Тепер надаємо будь-яку IP-адресу мережі, припустимо – 192.168.24.0/24, з цього виходять адреси для кожного маршрутизатора:

- R1 – 192.168.24.24
- R2 – 192.168.24.2
- R3 – 192.168.24.20
- R4 – 192.168.24.22

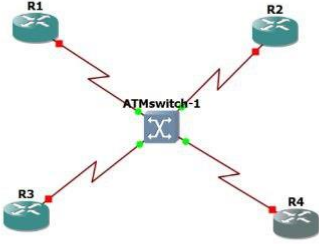
Введемо їх у відповідні поля в додатку (рис. 3.6). Натискаємо кнопку «Get Script!» та отримуємо набір команд для налаштування маршрутизаторів мережі ATM (рис.3.7). Копіюємо команди з модального вікна та відправляємо на комп'ютер з встановленим GNS3. На вже підготовленій мережі заходимо до консолі маршрутизатора R1 та вставляємо згенерований для нього скрипт (рис. 3.8).



03:27 79%

**Choose number of routers in your network**

4



R1  
192.168.24.24

R2  
192.168.24.2

R3  
192.168.24.20

R4  
192.168.24.22

ADD ROUTER IP

Get Script!

Рисунок 3.6 – Заповнені поля для кожного з роутерів

03 Choose number of routers in your network

```
R1

conf t
int atm 1/0
no sh
ex
int atm1/0.1 multipoint
ip add 192.168.24.24 255.255.255.0
pvc 0/102
encapsulation aal5snap
protocol ip 192.168.24.2 broadcast
pvc 0/103
encap aal5snap
protocol ip 192.168.24.20 broadcast
pvc 0/104
encap aal5snap
protocol ip 192.168.24.22 broadcast
end

R2

conf t
int atm1/0
no sh
ex
int atm 1/0.1 point-to-point
ip add 192.168.24.2 255.255.255.0
pvc 0/201
encap aal5snap
end

R3

conf t
int atm1/0
no sh
ex
int atm1/0.1 point-to-point
ip add 192.168.24.20 255.255.255.0
pvc 0/301
encap aal5snap
end

R4

conf t
int atm1/0
no sh
ex
int atm1/0.1 point-to-point
ip add 192.168.24.22 255.255.255.0
pvc 0/401
encap aal5snap
end
```

Close Script

Рисунок 3.7 – Отриманий перелік команд для чотирьох роутерів

```

Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by cisco Systems, Inc.
Compiled Thu 11-May-06 22:37 by evmiller
*Dec 18 03:38:49.211: %SNMP-5-COLDSTART: SNMP agent on host R1 is undergoing a c
old start
*Dec 18 03:38:50.167: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state
to administratively down
*Dec 18 03:38:50.287: %LINK-5-CHANGED: Interface ATM1/0, changed state to admini
stratively down
*Dec 18 03:38:51.167: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to down
*Dec 18 03:38:51.287: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, ch
anged state to down
R1#
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int atm 1/0
R1(config-if)#no sh
R1(config-if)#ex
R1(config)#int atm1/0.1 multipoint
R1(config-subif)#ip add 192.168.24.24 255.255.255.0
R1(config-subif)#pvc 0/102
R1(config-if-atm-vc)#encapsulation aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.24.2 broadcast
R1(config-if-atm-vc)#pvc 0/103
R1(config-if-atm-vc)#encap aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.24.20 broadcast
R1(config-if-atm-vc)#pvc 0/104
R1(config-if-atm-vc)#encap aal5snap
R1(config-if-atm-vc)#protocol ip 192.168.24.22 broadcast
R1(config-if-atm-vc)#end
R1#
*Dec 18 03:42:39.427: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
R1#
*Dec 18 03:42:40.259: %SYS-5-CONFIG_I: Configured from console by console
*Dec 18 03:42:40.427: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R1#

```

Рисунок 3.8 – Консоль роутера R1

Таким чином налаштуємо роутери, що залишились і перевіряємо поточну конфігурацію маршрутизаторів консольною командою «show run» в CLI роутера R1 (рис. 3.9). Тепер щоб перевірити цілісність мережі відправимо запит Echo-Request за допомогою команди «ping» з роутера R4 до інших (рис. 3.10).



```
!
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex half
!
interface ATM1/0
no ip address
no atm ilmi-keepalive
!
interface ATM1/0.1 multipoint
ip address 192.168.24.24 255.255.255.0
pvc 0/102
protocol ip 192.168.24.2 broadcast
encapsulation aal5snap
!
pvc 0/103
protocol ip 192.168.24.20 broadcast
encapsulation aal5snap
!
pvc 0/104
protocol ip 192.168.24.22 broadcast
encapsulation aal5snap
!
!
ip classless
no ip http server
!
!
no cdp log mismatch duplex
!
!
--More--
```

solarwinds | Solar-PuTTY free tool | © 2019 SolarWinds Worldwide, LLC. All rights reserved.

Рисунок 3.9 – Конфігурація роутера R1



```
anged state to down
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#int atm1/0
R4(config-if)#no sh
R4(config-if)#ex
R4(config)#int atm1/0.1 point-to-point
R4(config-subif)#ip add 192.168.24.22 255.255.255.0
R4(config-subif)#pvc 0/401
R4(config-if-atm-vc)#encap aal5snap
R4(config-if-atm-vc)#end
R4#
*Dec 18 03:50:37.875: %SYS-5-CONFIG_I: Configured from console by console
R4#
*Dec 18 03:50:38.531: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
*Dec 18 03:50:39.531: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
R4#P7P?
% Unrecognized command
R4#ping 192.168.24.24

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.24, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/45/56 ms
R4#ping 192.168.24.20

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 108/117/132 ms
R4#ping 192.168.24.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.24.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/103/116 ms
R4#
```

Рисунок 3.10 – Використання «ping» в консолі роутера R4

Результати виконання «ping» є задовільними, це означає, що побудована мережа АТМ є цілісною та якісною.

## ВИСНОВОК

Під час виконання роботи було детально переглянуто технології для розробки мобільного додатку для генерації скриптів налаштування роутерів в мережі асинхронного режиму передавання.

Були проаналізовані та порівняні між собою найпопулярніші в 2022 році мови програмування та їх фреймворки. Серед них було обрано найзручніше рішення – мову JavaScript з фреймворками React.js та React Native. Це дозволило зібрати мобільні додатки для операційних систем iOS і Android не змінюючи код програми.

В рамках роботи було розроблено мобільний додаток, який дозволяє швидко налаштувати мережу асинхронного режиму передавання. Для того щоб успішно згенерувати скрипти налаштування маршрутизаторів потрібно ввести їх IP-адреси в текстові поля, натиснути на кнопку та скопіювати готовий набір команд. Після чого вставити його в консоль кожного з роутерів.

Розроблений мобільний додаток відповідає всім вимогам, визначених на етапі постановки завдання.

## СПИСОК ЛІТЕРАТУРИ

1. Sen E. ATM Technology for Broadband Telecommunications Networks. – Routledge, 2018. – 304 с.
2. ATM and ATM Networks [Електронний ресурс] – Режим доступу: <https://www.tutorialspoint.com/ATM-Networks>
3. Gilski P., Stefanski J. Android os: a review //Tem Journal. – 2015. – Т. 4. – №. 1. – С. 116.
4. Why is Java ‘write once and run anywhere’? [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/why-is-java-write-once-and-run-anywhere/>
5. Programming Language [Електронний ресурс] – Режим доступу: <https://www.javatpoint.com/programming-language>
6. 16 Best Programming Languages for Mobile App Development 2022 [Електронний ресурс] – 2022. Режим доступу: <https://www.spinxdigital.com/blog/mobile-app-development-languages/>
7. Harder J. What Is JavaScript? //Graphics and Multimedia for the Web with Adobe Creative Cloud. – Apress, Berkeley, CA, 2018. – С. 881-898.
8. 9 Best JavaScript Frameworks to Use in 2022 [Електронний ресурс] – 2022. Режим доступу: <https://ninetailed.io/blog/best-javascript-frameworks/>
9. REACT NATIVE PROS AND CONS [Електронний ресурс] – 2022. Режим доступу: <https://pagepro.co/blog/react-native-pros-and-cons/>
10. What Is React Native? Is It Worth Using? [Електронний ресурс] – 2022. Режим доступу: <https://brainhub.eu/library/what-is-react-native#what-is-react-native>
11. Pros and Cons of React [Електронний ресурс] – 2022. Режим доступу: <https://thecodest.co/blog/pros-and-cons-of-react/>

12. Li T., Thain Jr W. E., Fallon T. On the use of virtualization for router network simulation //2010 Annual Conference & Exposition. – 2010. – C. 15.925. 1-15.925. 11.



## ДОДАТКИ

### Додаток А. Лістинг програмного коду

#### *App.js*

```
import React from "react";
import { StyleSheet, View, Text, ScrollView } from "react-native";
import { Navbar } from "../src/Navbar";
import { QuanSelector } from "../src/QuanSelector";

import { Instructions } from "../src/Instructions";

export default function App() {
  return (
    <ScrollView>
      <Navbar title="ATM Configurator" />
      <View style={styles.container}>
        <Instructions />
        <QuanSelector />
      </View>
    </ScrollView>
  );
}

const styles = StyleSheet.create({
  container: {
    paddingVertical: 30,
```

```

paddingHorizontal: 20,
},
inst: { marginLeft: 20, fontSize: 15, paddingBottom: 15, lineHeight:
25 },
});

```

### *Navbar.js*

```

import React from "react";
import { View, Text, StyleSheet, Image } from "react-native";

export const Navbar = (props) => {
  return (
    <View style={styles.navbar}>
      <Text style={styles.text}>{props.title}</Text>
      <Image style={styles.img} source={require("./Router.png")} />
    </View>
  );
};

const styles = StyleSheet.create({
  navbar: {
    height: 75,
    alignItems: "center",
    justifyContent: "flex-end",
    backgroundColor: "#3949ab",
  },
  text: {
    color: "#fff",
    fontSize: 20,
  }
});

```

```

    margin: -35,
    // fontFamily: "Open Sans",
  },
  img: {
    width: 42,
    height: 44,
    marginLeft: 300,
  },
});

```

### ***QuanSelector.js***

```

import React, { useState, useEffect } from "react";
import { SelectList } from "react-native-dropdown-select-list";
import {
  View,
  StyleSheet,
  Text,
  Image,
  Button,
  Alert,
  TextInput,
  Modal,
  Pressable,
} from "react-native";

export const QuanSelector = () => {
  const [selected, setSelected] = useState("");

  const data = [

```

```

    { key: "2", value: "2" },
    { key: "3", value: "3" },
    { key: "4", value: "4" },
  ];

  const [r1ip, setr1Ip] = useState("");
  const [r2ip, setr2Ip] = useState("");
  const [r3ip, setr3Ip] = useState("");
  const [r4ip, setr4Ip] = useState("");

  const [isValidr1Ip, setValidr1Ip] = useState(true);
  const [isValidr2Ip, setValidr2Ip] = useState(true);
  const [isValidr3Ip, setValidr3Ip] = useState(true);
  const [isValidr4Ip, setValidr4Ip] = useState(true);

  const verifyr1Ip = (r1ip) => {
    let regex = new RegExp(
      /^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/gi
    );
    if (regex.test(r1ip)) {
      return true;
    }
    return false;
  };

  const verifyr2Ip = (r2ip) => {
    let regex = new RegExp(
      /^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/gi
    );

```

```

    if (regex.test(r2ip)) {
      return true;
    }
    return false;
  };

const verifyr3Ip = (r3ip) => {
  let regex = new RegExp(
    /^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.) {3} ([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/gi
  );
  if (regex.test(r3ip)) {
    return true;
  }
  return false;
};

const verifyr4Ip = (r4ip) => {
  let regex = new RegExp(
    /^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.) {3} ([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$/gi
  );
  if (regex.test(r4ip)) {
    return true;
  }
  return false;
};

const [modalVisible, setModalVisible] = useState(false);

const submitIp = () => {};

```

```

// } else {
//   console.log("bye");
// }
// console.log(parseInt(selected));
// console.log(IpInput);
if (parseInt(selected) == 2) {
  return (
    <View style={styles.list}>
      <Text style={styles.txt}>Choose number of routers in your
network</Text>
      <SelectList
        data={data}
        setSelected={setSelected}
        placeholder="Select amount of routers"
        dropdownStyles={{ backgroundColor: "#B9E0FF" }}
        dropdownItemStyles={{ alignItems: "center" }}
        dropdownTextStyles={{ textDecorationLine: "underline" }}
        // onSelect={setSelected}
      />
      <View>
        <Image style={styles.img2} source={require("./2.png")} />
      </View>
      <View>
        <Text>R1</Text>
        <View>
          <View style={styles.block}>
            <TextInput
              style={styles.input}
              placeholder="Write IP-address here..."
              onChangeText={(newr1Ip) => {

```

```

        setr1Ip(newr1Ip);
        const isValid = verifyr1Ip(newr1Ip);
        isValid ? setValidr1Ip(true) : setValidr1Ip(false);
        // console.log(newr1Ip);
        return newr1Ip;
    }}
    defaultValue={r1ip}
    keyboardType="numeric"
  />
</View>
<Text style={styles.warning}>
  {isValidr1Ip ? "" : "IP Address is invalid!"}{" "}
</Text>
</View>
<Text>R2</Text>
<View>
  <View style={styles.block}>
    <TextInput
      style={styles.input}
      placeholder="Write IP-address here..."
      onChangeText={({ newr2Ip) => {
        setr2Ip(newr2Ip);
        const isValid = verifyr2Ip(newr2Ip);
        isValid ? setValidr2Ip(true) : setValidr2Ip(false);
        // console.log(newr2Ip);
        return newr2Ip;
      }}
      defaultValue={r2ip}
      keyboardType="numeric"
    />

```

```

</View>
<Text style={styles.warning}>
  {isValidr2Ip ? "" : "IP Address is invalid!"}{" "}
</Text>
</View>
<Button
  style={styles.btn}
  title="Add Router IP"
  color={"#8D9EFF"}
  disabled={isValidr1Ip & isValidr2Ip ? false : true}
  onPress={() => {
    Alert.alert("Submitted to DataBase successfully");
    createDatabase(Configuration.dbName);
    insertInto(Configuration.dbName, "r1ip", "r2ip");
    console.log(r1ip);
    console.log(r2ip);
  }}
/>
</View>
<View style={styles.centeredView}>
  <Modal
    animationType="slide"
    transparent={true}
    visible={modalVisible}
    onRequestClose={() => {
      Alert.alert("Script has been closed.");
      setModalVisible(!modalVisible);
    }}
  >
  <View style={styles.centeredView}>

```



```

<View style={styles.modalView}>
  <Text
    selectable={true}
    multiline={true}
    selectionColor="orange"
    editable={false}
    style={styles.modalText}
  >
  <Text
    style={{
      fontWeight: "bold",
    }}
  >
    R1
  </Text>
  {"\n"}
  {"\n"}
  conf t{"\n"}
  int atm 1/0{"\n"}
  no sh{"\n"}
  ex{"\n"}
  int atm1/0.1 multipoint{"\n"}
  ip add {r1ip}
  {"\n"}
  pvc 0/102{"\n"}
  encapsulation aal5snap{"\n"}
  protocol ip {r2ip} broadcast{"\n"}
  end{"\n"}
  <Text
    style={{

```

```

        fontWeight: "bold",
      }}
    >
      R1
    </Text>
    {"\n"}
    {"\n"}
    conf t{"\n"}
    int atm1/0{"\n"}
    no sh{"\n"}
    ex{"\n"}
    int atm 1/0.1 point-to-point{"\n"}
    ip add {r2ip} 255.255.255.0{"\n"}
    pvc 0/201{"\n"}
    encaps aal5snap{"\n"}
    end{"\n"}
  </Text>
  <Pressable
    style={[styles.button, styles.buttonClose]}
    onPress={() => setModalVisible(!modalVisible)}
  >
    <Text style={styles.textStyle}>Close Script</Text>
  </Pressable>
</View>
</View>
</Modal>
<Pressable
  style={[styles.button, styles.buttonOpen]}
  onPress={() => setModalVisible(true)}
>

```

```

        <Text style={styles.textStyle}>Get Script!</Text>
    </Pressable>
</View>
</View>
);
} else if (parseInt(selected) == 3) {
    return (
        <View style={styles.list}>
            <Text style={styles.txt}>Choose number of routers in your
network</Text>
            <SelectList
                data={data}
                setSelected={setSelected}
                placeholder="Select amount of routers"
                dropdownStyles={{ backgroundColor: "#B9E0FF" }}
                dropdownItemStyles={{ alignItems: "center" }}
                dropdownTextStyles={{ textDecorationLine: "underline" }}
                // onSelect={setSelected}
            />
            <View>
                <Image style={styles.img3} source={require("./3.png")} />
            </View>

            <View>
                <Text>R1</Text>
                <View>
                    <View style={styles.block}>
                        <TextInput
                            style={styles.input}
                            placeholder="Write IP-address here..."

```

```

        onChangeText={ (newr1Ip) => {
            setr1Ip(newr1Ip);
            const isValid = verifyr1Ip(newr1Ip);
            isValid ? setValidr1Ip(true) : setValidr1Ip(false);
            // console.log(newr1Ip);
            return newr1Ip;
        }}
        defaultValue={r1ip}
        keyboardType="numeric"
    />
</View>
<Text style={styles.warning}>
    {isValidr1Ip ? "" : "IP Address is invalid!"}{" "}
</Text>
</View>
<Text>R2</Text>
<View>
    <View style={styles.block}>
        <TextInput
            style={styles.input}
            placeholder="Write IP-address here..."
            onChangeText={ (newr2Ip) => {
                setr2Ip(newr2Ip);
                const isValid = verifyr2Ip(newr2Ip);
                isValid ? setValidr2Ip(true) : setValidr2Ip(false);
                // console.log(newr2Ip);
                return newr2Ip;
            }}
            defaultValue={r2ip}
            keyboardType="numeric"

```

```

    />
  </View>
  <Text style={styles.warning}>
    {isValidr2Ip ? "" : "IP Address is invalid!"}{" "}
  </Text>
</View>
<Text>R3</Text>
<View>
  <View style={styles.block}>
    <TextInput
      style={styles.input}
      placeholder="Write IP-address here..."
      onChangeText={({ newr3Ip) => {
        setr3Ip(newr3Ip);
        const isValid = verifyr3Ip(newr3Ip);
        isValid ? setValidr3Ip(true) : setValidr3Ip(false);
        // console.log(newr3Ip);
        return newr3Ip;
      }}
      defaultValue={r3ip}
      keyboardType="numeric"
    />
  </View>
  <Text style={styles.warning}>
    {isValidr3Ip ? "" : "IP Address is invalid!"}{" "}
  </Text>
</View>
<Button
  style={styles.btn}
  title="Add Router IP"

```

```

        color={"#8D9EFF"}
        disabled={isValidr1Ip & isValidr2Ip & isValidr3Ip ? false
: true}

        onPress={() => {
            Alert.alert("Submitted to DataBase successfully");
            createDatabase(Configuration.dbName);
            insertInto(Configuration.dbName, r1ip, r2ip, r3ip);
            console.log(r1ip);
            console.log(r2ip);
            console.log(r3ip);
        }}
    />
</View>
<View style={styles.centeredView}>
    <Modal
        animationType="slide"
        transparent={true}
        visible={modalVisible}
        onRequestClose={() => {
            Alert.alert("Script has been closed.");
            setModalVisible(!modalVisible);
        }}
    >
    <View style={styles.centeredView}>
        <View style={styles.modalView}>
            <Text
                selectable={true}
                multiline={true}
                selectionColor="orange"
                editable={false}

```

```
style={styles.modalText}
>
<Text
  style={{
    fontWeight: "bold",
  }}
>
  R1
</Text>
{"\n"}
{"\n"}
conf t{"\n"}
int atm 1/0{"\n"}
no sh{"\n"}
ex{"\n"}
int atm1/0.1 multipoint{"\n"}
ip add {r1ip}
{"\n"}
pvc 0/102{"\n"}
encapsulation aal5snap{"\n"}
protocol ip {r2ip} broadcast{"\n"}
pvc 0/103{"\n"}
encap aal5snap{"\n"}
protocol ip {r3ip} broadcast{"\n"}
end{"\n"}
{"\n"}
<Text
  style={{
    fontWeight: "bold",
  }}
  </Text>
```

```
>
    R2
</Text>
{"\n"}
{"\n"}
conf t{"\n"}
int atm1/0{"\n"}
no sh{"\n"}
ex{"\n"}
int atm 1/0.1 point-to-point{"\n"}
ip add {r2ip} 255.255.255.0{"\n"}
pvc 0/201{"\n"}
encap aal5snap{"\n"}
end{"\n"}
{"\n"}
<Text
    style={{
        fontWeight: "bold",
    }}
>
    R3
</Text>
{"\n"}
conf t{"\n"}
int atm1/0{"\n"}
no sh{"\n"}
ex{"\n"}
int atm1/0.1 point-to-point{"\n"}
ip add {r3ip} 255.255.255.0{"\n"}
pvc 0/301{"\n"}
```



```

        encap aal5snap{"\n"}
        end{"\n"}
    </Text>
    <Pressable
        style={ [styles.button, styles.buttonClose] }
        onPress={() => setModalVisible(!modalVisible)}
    >
        <Text style={styles.textStyle}>Close Script</Text>
    </Pressable>
</View>
</View>
</Modal>
<Pressable
    style={ [styles.button, styles.buttonOpen] }
    onPress={() => setModalVisible(true)}
>
    <Text style={styles.textStyle}>Get Script!</Text>
</Pressable>
</View>
</View>
);
} else if (parseInt(selected) == 4) {
    return (
        <View style={styles.list}>
            <Text style={styles.txt}>Choose number of routers in your
network</Text>
            <SelectList
                data={data}
                setSelected={setSelected}
                placeholder="Select amount of routers"
            >

```

```

dropdownStyles={{ backgroundColor: "#B9E0FF" }}
dropdownItemStyles={{ alignItems: "center" }}
dropdownTextStyles={{ textDecorationLine: "underline" }}
// onSelect={setSelected}
/>
<View>
  <Image style={styles.img4} source={require("./4.png")} />
</View>

<View>
  <Text>R1</Text>
  <View>
    <View style={styles.block}>
      <TextInput
        style={styles.input}
        placeholder="Write IP-address here..."
        onChangeText={({newr1Ip}) => {
          setr1Ip(newr1Ip);
          const isValid = verifyr1Ip(newr1Ip);
          isValid ? setValidr1Ip(true) : setValidr1Ip(false);
          // console.log(newr1Ip);
          return newr1Ip;
        }}
        defaultValue={r1ip}
        keyboardType="numeric"
      />
    </View>
  <Text style={styles.warning}>
    {isValidr1Ip ? "" : "IP Address is invalid!"}{" "}
  </Text>

```

```

</View>
<Text>R2</Text>
<View>
  <View style={styles.block}>
    <TextInput
      style={styles.input}
      placeholder="Write IP-address here..."
      onChangeText={ (newr2Ip) => {
        setr2Ip(newr2Ip);
        const isValid = verifyr2Ip(newr2Ip);
        isValid ? setValidr2Ip(true) : setValidr2Ip(false);
        // console.log(newr2Ip);
        return newr2Ip;
      }}
      defaultValue={r2ip}
      keyboardType="numeric"
    />
  </View>
  <Text style={styles.warning}>
    {isValidr2Ip ? "" : "IP Address is invalid!"}{" "}
  </Text>
</View>
<Text>R3</Text>
<View>
  <View style={styles.block}>
    <TextInput
      style={styles.input}
      placeholder="Write IP-address here..."
      onChangeText={ (newr3Ip) => {
        setr3Ip(newr3Ip);

```

```

        const isValid = verifyr3Ip(newr3Ip);
        isValid ? setValidr3Ip(true) : setValidr3Ip(false);
        // console.log(newr3Ip);
        return newr3Ip;
    }}
    defaultValue={r3ip}
    keyboardType="numeric"
  />
</View>
<Text style={styles.warning}>
  {isValidr3Ip ? "" : "IP Address is invalid!"}{" "}
</Text>
</View>
<Text>R4</Text>
<View>
  <View style={styles.block}>
    <TextInput
      style={styles.input}
      placeholder="Write IP-address here..."
      onChangeText={({ newr4Ip) => {
        setr4Ip(newr4Ip);
        const isValid = verifyr4Ip(newr4Ip);
        isValid ? setValidr4Ip(true) : setValidr4Ip(false);
        // console.log(newr4Ip);
        return newr4Ip;
      }}
      defaultValue={r4ip}
      keyboardType="numeric"
    />
  </View>
</View>

```

```

<Text style={styles.warning}>
    {isValidr4Ip ? "" : "IP Address is invalid!"}{" "}
</Text>
</View>
<Button
    style={styles.btn}
    title="Add Router IP"
    color={"#8D9EFF"}
    disabled={
        isValidr1Ip & isValidr2Ip & isValidr3Ip & isValidr4Ip
        ? false
        : true
    }
    onPress={() => {
        Alert.alert("Submitted to DataBase successfully");
        createDatabase(Configuration.dbName);
        insertInto(Configuration.dbName, "r1ip", "r2ip",
"r3ip", "r4ip");
        console.log(r1ip);
        console.log(r2ip);
        console.log(r3ip);
        console.log(r4ip);
    }}
/>
</View>
<View style={styles.centeredView}>
    <Modal
        animationType="slide"
        transparent={true}
        visible={modalVisible}

```

```

onRequestClose={() => {
  Alert.alert("Script has been closed.");
  setModalVisible(!modalVisible);
}}
>
<View style={styles.centeredView}>
  <View style={styles.modalView}>
    <Text
      selectable={true}
      multiline={true}
      selectionColor="orange"
      editable={false}
      style={styles.modalText}
    >
      <Text
        style={{
          fontWeight: "bold",
        }}
      >
        R1
      </Text>
      {"\n"}
      {"\n"}
      conf t{"\n"}
      int atm 1/0{"\n"}
      no sh{"\n"}
      ex{"\n"}
      int atm1/0.1 multipoint{"\n"}
      ip add {r1ip} 255.255.255.0
      {"\n"}

```

```
pvc 0/102{"\n"}
encapsulation aal5snap{"\n"}
protocol ip {r2ip} broadcast{"\n"}
pvc 0/103{"\n"}
encap aal5snap{"\n"}
protocol ip {r3ip} broadcast{"\n"}
pvc 0/104{"\n"}
encap aal5snap{"\n"}
protocol ip {r4ip} broadcast{"\n"}
end{"\n"}
{"\n"}
<Text
  style={{
    fontWeight: "bold",
  }}
>
  R2
</Text>
{"\n"}
{"\n"}
conf t{"\n"}
int atm1/0{"\n"}
no sh{"\n"}
ex{"\n"}
int atm 1/0.1 point-to-point{"\n"}
ip add {r2ip} 255.255.255.0{"\n"}
pvc 0/201{"\n"}
encap aal5snap{"\n"}
end{"\n"}
{"\n"}
```

```
<Text
  style={{
    fontWeight: "bold",
  }}
>
  R3
</Text>
{"\n"}
{"\n"}
conf t{"\n"}
int atm1/0{"\n"}
no sh{"\n"}
ex{"\n"}
int atm1/0.1 point-to-point{"\n"}
ip add {r3ip} 255.255.255.0{"\n"}
pvc 0/301{"\n"}
encap aal5snap{"\n"}
end{"\n"}
{"\n"}
<Text
  style={{
    fontWeight: "bold",
  }}
>
  R4
</Text>
{"\n"}
{"\n"}
conf t{"\n"}
int atm1/0{"\n"}
```



```

        no sh{"\n"}
        ex{"\n"}
        int atm1/0.1 point-to-point{"\n"}
        ip add {r4ip} 255.255.255.0{"\n"}
        pvc 0/401{"\n"}
        encap aal5snap{"\n"}
        end{"\n"}
    </Text>
    <Pressable
        style={[styles.button, styles.buttonClose]}
        onPress={() => setModalVisible(!modalVisible)}
    >
        <Text style={styles.textStyle}>Close Script</Text>
    </Pressable>
</View>
</View>
</Modal>
<Pressable
    style={[styles.button, styles.buttonOpen]}
    onPress={() => setModalVisible(true)}
>
    <Text style={styles.textStyle}>Get Script!</Text>
</Pressable>
</View>
</View>
);
}

return (
    <View style={styles.list}>

```

```

    <Text style={styles.txt}>Choose number of routers in your
network</Text>

    <SelectList
      data={data}
      setSelected={setSelected}
      placeholder="Select amount of routers"
      dropdownStyles={{ backgroundColor: "#B9E0FF" }}
      dropdownItemStyles={{ alignItems: "center" }}
      dropdownTextStyles={{ textDecorationLine: "underline" }}
      onSelect={setSelected}
    />
  </View>
);

// return (
//   <View style={styles.list}>
//     <Text style={styles.txt}>Choose number of routers in your
network</Text>
//     <SelectList
//       data={data}
//       setSelected={setSelected}
//       placeholder="Select amount of routers"
//       dropdownStyles={{ backgroundColor: "#B9E0FF" }}
//       dropdownItemStyles={{ alignItems: "center" }}
//       dropdownTextStyles={{ textDecorationLine: "underline" }}
//     />
//   </View>
// );
};

const styles = StyleSheet.create({

```

```
list: {
  paddingHorizontal: 30,
  paddingVertical: 50,
},

txt: {
  fontWeight: "bold",
  paddingBottom: 10,
  fontSize: 18,
},

img2: {
  width: 352.5,
  height: 124.5,
  alignItems: "center",
  justifyContent: "center",
  marginTop: 55,
},

img3: {
  width: 282.5,
  height: 260,
  alignItems: "center",
  justifyContent: "center",
  marginTop: 55,
  marginLeft: 35,
},

img4: {
  width: 247.5,
  height: 176.5,
  alignItems: "center",
```

```
    justifyContent: "center",
    marginTop: 55,
    marginLeft: 35,
  },

  input: {
    width: "60%",
    borderStyle: "solid",
    borderBottomWidth: 1,
    borderRadius: 5,
    borderColor: "grey",
    // marginTop: 20,
    marginLeft: 5,
    padding: 5,
  },

  block: {
    flexDirection: "row",
    justifyContent: "space-between",
    alignItems: "center",
  },

  warning: {
    color: "red",
    padding: 5,
    fontSize: 14,
  },

  centeredView: {
    flex: 1,
```

```
    justifyContent: "center",
    alignItems: "center",
    marginTop: 22,
  },
  modalView: {
    margin: 20,
    backgroundColor: "white",
    borderRadius: 20,
    padding: 35,
    alignItems: "center",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 4,
    elevation: 5,
  },
  button: {
    borderRadius: 15,
    padding: 15,
    elevation: 2,
  },
  buttonOpen: {
    backgroundColor: "#00005C",
  },
  buttonClose: {
    backgroundColor: "#6D67E4",
  },
```

```
textStyle: {
  color: "white",
  fontWeight: "bold",
  textAlign: "center",
},
modalText: {
  marginBottom: 15,
  // textAlign: "center",
},

rhead: {
  fontWeight: "bold",
  borderBottomColor: "black",
  borderStyle: "solid",
  borderBottomWidth: 1,
},
});
```