

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**
Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

**Кваліфікаційна робота
на здобуття ступеня бакалавра**

за спеціальністю 122 Комп'ютерні науки

на тему:

**РОЗРОБКА АНДРОЇД ДОДАТКУ ДЛЯ ВІДОБРАЖЕННЯ
НАВЧАЛЬНОГО РОЗКЛАДУ**

Виконав студент 4-го курсу
Кирило РИМАНОВ



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Тетяна КАРНАУХ



(підпис)

Засвідчую, що в цій роботі немає запозичень з праць
інших авторів без відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту на засіданні
кафедри теоретичної кібернетики

« 01 » ч е р в н я 2022 р., протокол № 11

Завідувач кафедри

Юрій КРАК

(підпис)

РЕФЕРАТ

Обсяг роботи 39 сторінок, 28 ілюстрацій, 23 джерела посилань.

РОЗКЛАД, ANDROID ДОДАТОК, GOOGLE API, GOOGLE SHEETS

Об'єктом роботи є відображення навчального розкладу. Предметом роботи є реалізація додатка з навчальним розкладом на базі Android за допомогою Android Studio.

Метою роботи є розроблення та програмна реалізація мобільного додатку для відображення навчального розкладу, а також підбір необхідних засобів та технологій.

Інструменти розроблення: інтегроване середовище розробки: Android Studio, IntelliJ Idea, Google API services, Google play API, Google Sheets API.

Результати роботи: виконано огляд IDE Android Studio, розроблено логіку реалізації додатку та виконано його базовий функціонал. Реалізовано та порівняно два способи витягнення даних з Google Sheets, який грав роль бази даних у проекті.

Розроблене програмне забезпечення може використовуватись навчальними закладами середньої та вищої освіти. Може бути додатково змінено та використано в усіх галузях, де існує динаміка в робочому процесі та існує кінцевий список варіантів зайнятостей.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
1 ВИКОРИСТОВУВАНІ ПРОГРАМНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ	8
1.1 Мова програмування Java	8
1.2 Android studio	8
1.3. Google Sheets	9
1.4 Порівняння Google Sheets та Microsoft Excel	10
1.5 Google API	11
2 РОБОТА З ЗОВНІШНІМИ ДЛЯ ПРОЄКТУ API	13
2.1 Аутентифікація для використання Google API	13
2.2 Використання Google API	15
2.3 Використання Google Sheets API	17
2.3.1 Запис даних у JSON файл	19
3 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ РОЗРОБЛЮВАНОВОГО ДОДАТКУ	21
3.1 Загальна структура додатку	21
3.1.1 Паттерн Observer	23
3.2 Загальна конфігурація проєкту в AD	24
3.3 Підготовка даних	26
3.4 Отримання даних	29
3.5 Графічний інтерфейс	30
3.5 Програмно-апаратні вимоги та розгортання	34
3.6 Інструкція користувача	34
ВИСНОВКИ	36
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	38

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Android – Мобільна операційна система

AD (*Android Studio*) – IDE, який дозволяє працювати над програмами для запуску на мобільних пристроях з програмним забезпеченням Android

API (*application programming interface*) – програмний посередник, який дозволяє двом додаткам спілкуватися один з одним.

IDE (*Integrated development environment*) - загальна назва програмного забезпечення для будування програм, який має в собі редактор коду, налагоджувач та локальну автоматизацію коду

SDK (*software Development Kit*) – деяка сукупність інструментів та документації, який дозволяє програмістам створювати програми та продукти за даною технологією або для зазначеної платформи

ВСТУП

Оцінка сучасного стану об'єкта розробки. Модернізація типових процесів у кожній сфері життя необхідна для росту, ефективності та прогресу суспільства та збільшення його досягнень. Це універсальний шаг, який треба робити в будь-якому разі, якщо на це є можливість. Тому діджиталізація навчального розкладу це дуже логічний висновок, до якого ми би дійшли раніше або пізніше. Не має ніякого сенсу в сучасному світі носити з собою будь-яку бумагу, або мати доступ до розкладу тільки в одній точці закладу, тому найліпше надати змогу кожному знайти актуальний розклад у своєму мобільному пристрої.

Кожен сьогодні має з собою смартфон, яким користується кожен день у місцях, в яких несприятливо використовувати стаціонарні або мобільні ПК – розробка додатку на смартфон також має ідентичний прагматичний задум.

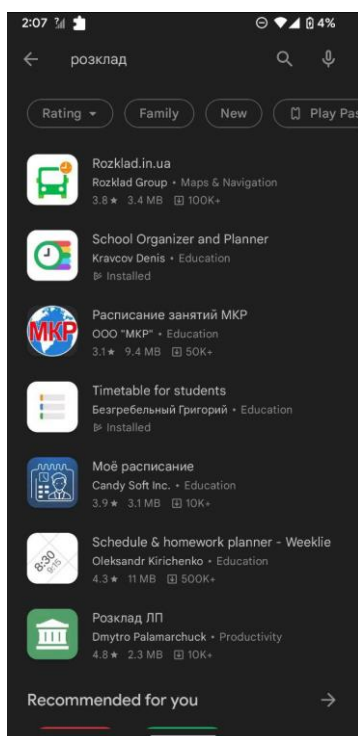


Рисунок 1.1 – Доступні рішення додатка “розклад”

На рис. 1.1 зображені найбільш популярні додатки за запитом “розклад”. Шість з них для навчального розкладу, два — для локального користування учнів

в інших закладах та чотири для самовлаштування деякого розкладу. Розроблюване в цій роботі рішення дещо інше і концентрується на функціоналі, який ніяка з цих програм не може дати нашому університету.

Актуальність роботи. Навіть якщо існують варіанти реалізації схожого задуму в загальному доступі, розробка локального додатку водночас позбавить від багатьох речей, які можуть стати фатальними в незвичайних обставинах, будь то приватність, побудова системи на програмному забезпеченні, яке можуть відключити в майбутньому через нерентабельність, або коли стандартними засобами неможливо отримати необхідну гнучкість.

Якщо ми розглянемо Play Store додатки для знаходження потенційно потрібного для нас рішення, то побачимо, що варіанти не зможуть нас повністю влаштувати.

Мета й завдання роботи. Мета кваліфікаційної роботи полягає у створенні програмного засобу для відображення навчального розкладу. Розроблюваний засіб має працювати на платформі Android та використовувати онлайн базу даних.

Для досягнення цієї мети поставлено такі завдання.

- Ознайомитись з Google API, Google Sheets API.
- Виконати порівняльний аналіз Microsoft Excel та Google Sheets.
- Ознайомитись з Android Studio.
- Підібрати надійні та зручні засоби та технології для зчитування даних, що мають відображатися.
- Розробити програмний продукт під Android для відображення навчального розкладу.

Об'єкт, методи й засоби розроблення. Об'єктом розроблення є програмний засіб для відображення навчального розкладу. Розробці передувало ознайомлення з необхідними засобами та технологіями та порівняльний аналіз Microsoft Excel та Google Sheets, на підставі якого було вирішено використовувати Google Sheets.

Використовувались такі інструментальні засоби, бібліотеки та технології: Android Studio, IntelliJ Idea, Google API services, Google play API, Google Sheets API.

Можливі сфери застосування. Розроблене програмне забезпечення може використовуватись навчальними закладами середньої та вищої освіти. Може бути додатково змінено та використано в усіх галузях, де існує динаміка в робочому процесі та існує кінцевий список варіантів зайнятостей.

1 ВИКОРИСТОВУВАНІ ПРОГРАМНІ ЗАСОБИ ТА ТЕХНОЛОГІЇ

1.1 Мова програмування Java

Java це одна з найпоширеніших мов для розробки програм в усьому світі. Її універсальність, підтримка та релевантність є однією зі складових, чому я вирішив співпрацювати з нею у цьому проєкті.

Для розробки проєкту нам потрібні інструменти для роботи з різними API і в Java є дуже сильний, надійний та простий інструмент для інтегрування API, який називається Gradle. Gradle — це система, яка фокусується на автоматизації конструкції, має багатомовну підтримку, серед яких є:

- Java
- Scala
- Android
- Kotlin
- C/C++
- Groovy

Для розробки програми можливості Gradle необхідні. Також як Java, так і Gradle є повністю інтегрованими та стандартним рішенням для розробки в AD.

1.2 Android studio

Більше ніж три мільярди пристроїв мають операційну систему Android[17]. Android пристрої — це стандарт, який велика частина людей має при собі, тому робити додаток для цієї операційної системи є дуже прагматичною ідеєю. Android

Studio має всі можливості для розробки для цієї операційної системи, тому вибір пал на неї.

Функціонал AD SDK дуже широкий, у наявності присутній емулятор з усіма версіями операційної системи та лінійкою пристроїв різних років та потужностей. Внутрішній графічний дизайнер, який має змогу конструювати користувацького інтерфейсу через інструменти та код, має швидко синхронізоване прев'ю.

Величезна спільнота розробників та порівняно низький поріг для успішного початку створення проекту з AD також не можна не згадати. AD є стандартом та навіть якщо ринок андроїд пристроїв не зростає як раніше, існуюча аудиторія можливо найбільша з яких один проект може дотягнутися до.

1.3. Google Sheets

Google Sheets – це веб-додаток, який дозволяє користувачам створювати, оновлювати та змінювати електронні таблиці та обмінюватися даними в режимі онлайн в режимі реального часу.

Продукт Google пропонує типові функції електронних таблиць, такі як можливість додавати, видаляти та сортувати рядки та стовпці. Але на відміну від деяких інших програм для роботи з електронними таблицями, Google Sheets також дає змогу кільком користувачам водночас працювати над електронною таблицею та спілкуватися за допомогою вбудованої програми обміну миттєвими повідомленнями. Користувачі можуть завантажувати електронні таблиці безпосередньо зі своїх комп'ютерів або мобільних пристроїв. Програма автоматично зберігає кожну зміну, і користувачі можуть бачити зміни інших користувачів під час їх внесення.[18]

Таблиці Google включені як частина набору безкоштовних веб-програм «Редактори документів Google». Цей пакет також включає Google Docs, Google

Slides, Google Drawings, Google Forms, Google Sites і Google Keep.

Google Sheets зазвичай використовуються для спільної роботи з електронними таблицями в різних географічних місцях. Декілька користувачів можуть змінювати документ Google Sheets у режимі реального часу, при цьому зміни відстежуються для кожного окремого користувача.

Онлайн-додаток дає змогу користувачам створювати, редагувати та формувати електронні таблиці онлайн, щоб упорядковувати й аналізувати інформацію. Google Sheets часто порівнюють з Microsoft Excel, оскільки обидві програми використовуються для подібних цілей. Google Sheets – це, по суті, хмарна версія основних функцій Microsoft Excel від Google.

1.4 Порівняння Google Sheets та Microsoft Excel

Google Sheets створювався свідомо як аналог Microsoft Excel, тому базовий функціонал, як формули, розрахунки, загальний інтерфейс ідентичні. По недолікам та плюсам можна швидко зрозуміти для чого кожен з них більше підходить.

- **Ціна.** Google Sheets безкоштовний для особистого користування, а для Excel потрібна підписка на Office 365.
- **Хмарність.** Google Sheets — це хмарна веб-програма, а доступ до Microsoft 365 здійснюється через Інтернет за допомогою веб-браузера.
- **Співпраця.** Як правило, Google Sheets має переваги для спільної роботи, оскільки він повністю базується у веб. Хоча Excel має онлайн-версію, вона має бідніший функціонал.
- **Обробка даних.** Швидкість і продуктивність Google Sheets можуть бути обмежені підключенням до Інтернету або наближенням максимального обсягу пам'яті. Excel може зберігати 17 мільйонів клітинок, на відміну від обмеження в Google Sheets, яке становить 5 мільйонів клітинок.
- **Особливості.** Хоча Google Sheets мають основні функції програми

електронних таблиць, її пропозиції є досить простими. Excel має більший набір функцій і функцій для спеціалізованих цілей, а також більше параметрів налаштування та вбудованих формул.

- Інтеграція. Excel інтегрується з іншими програмами Microsoft, такими як Power BI. Хоча Google Sheets інтегрується з веб-програмами Google, такими як Google Drive, вони менш поширені в корпоративних налаштуваннях.
- Підтримка. І Sheets, і Excel мають спільноти підтримки. Google пропонує довідкові статті та інтерактивний форум спільноти для Таблиць. Microsoft пропонує довідковий форум спільноти та спеціальний навчальний центр для Excel.
- Доповнення функціонала. Google надає спільноті самій додавати те, чого не вистачає, через розробку плагінів та поширення в спеціальному централізованому хабі.[18]

Найбільш корисно виконувати задачі в Excel, коли вони потребують математичних операцій, розрахунків великих масштабів або великої статичної бази даних. Google Sheets, у свою чергу, більш привабливий для роботи над малими та середніми базами даних та заощадженням коштів.

1.5 Google API

Google API — це набір інтерфейсу програмного забезпечення, розробленого Google, який дозволяє спілкуватися зі службами Google. API дотримуються певних правил і методів, щоб чітко передавати запити та відповіді.

Можливість доступу до даних і обчислювальних ресурсів значно підвищує ефективність розробника. Набагато простіше використовувати API, ніж створювати кожен програму, метод чи набір даних з нуля. API створено з урахуванням розробників і часто не пропонує графічний інтерфейс користувача

(GUI). Google пропонує API, які можна застосувати до багатьох різних областей і секторів. API часто використовуються у веб-розробці, машинному навчанні, науці про дані та робочих процесах системного адміністрування.

Однією з найбільших проблем для компаній і розробників є інтеграція власної системної інфраструктури з цими хмарними платформами або сервісами. API це дуже корисний варіант для цих задач. В основному існують чотири можливі сценарії для компаній:

1. PaaS API: Платформа як сервіс, ці API призначені для забезпечення доступу та функціональності для хмарного середовища. Це означає інтеграцію з базами даних, системами обміну повідомленнями, порталами та компонентами зберігання.
2. SaaS API: програмне забезпечення як сервіс, ці сервісні API створюють зв'язок між хмарним програмним забезпеченням та усією інфраструктурою системи. CRM та ERP є справжніми прикладами цього.
3. IaaS API: інфраструктура як сервіс, ці API допомагають контролювати конкретні хмарні ресурси та їх розподіл. Конфігурація мережі та керування робочим навантаженням також можуть бути областю, де використовуються ці API.[11]

API надають широкий функціонал, такий як аналітика, машинне навчання як сервіс (API передбачення) або доступ до даних користувача (коли надано дозвіл на читання даних). Іншим важливим прикладом є вбудована у веб сайти Google Map, яка може бути досягнута за допомогою Static Maps API, Places API або API Google Earth.

2 РОБОТА З ЗОВНІШНІМИ ДЛЯ ПРОЄКТУ API

У програмному проєкті використовуються Google API та Google Sheets API. Розглянемо використовувані елементи.

2.1 Аутентифікація для використання Google API

Аутентифікація з Google OAuth 2.0 це інструмент, який є базисом для роботи з Google API.

Для того, щоб почати працювати з Google API треба зареєструвати додаток у Google Cloud та отримати потрібні компоненти від гугл для успішної аутентифікації додатком. Взаємін Google віддасть файл credentials, який використовує дані даного акаунту Google та дозволяє використовувати приватну інформацію, яка йому належить, наприклад, документ у Google Sheets. Нижче представлена діаграма комунікації додатку з Google серверами.

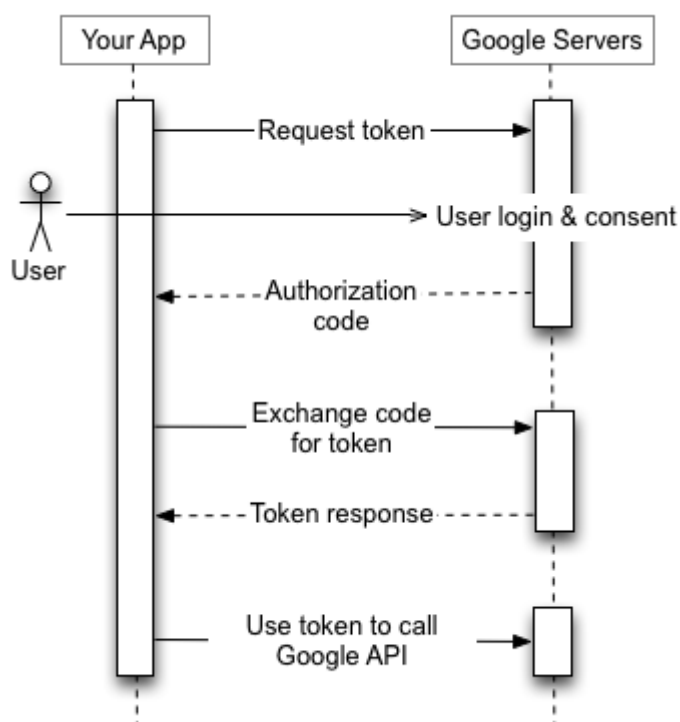


Рисунок 2.1 – Ініціалізація ідентифікатора клієнта.[10]

Перш за все додаток, маючи потрібні методи з API, з'єднується з сервером Google задля отримання маркера доступу (token). Google віддає назад пересилання яке просить нас одноразово зареєструватися через браузер у даний акаунт. Слідуючим кроком процесу є деякий ідентифікатор клієнта (Client ID) та в нашому випадку Client Secrets – файл з даними, як Client ID та іншими важливими OAuth2 даними.

```
GoogleClientSecrets clientSecrets = GoogleClientSecrets.load(JSON_FACTORY, new InputStreamReader(in));
```

Рисунок 2.2 – Ініціалізація ідентифікатора клієнта. [9]

Авторизаційний процес йде, коли додаток перенаправляє браузер на URL-адресу Google; сама адреса містить параметри запиту, які вказують на тип запитуваного доступу. У цей час відбувається згода та аутентифікація користувачів, також вибір сесії (сам конкретний запит). Результатом є код авторизації, який програма може обміняти на маркер доступу (token) та маркер оновлення (refresh token).

```
GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.Builder(
    HTTP_TRANSPORT, JSON_FACTORY, clientSecrets, SCOPES)
    .setDataStoreFactory(new FileDataStoreFactory(tokenFolder))
    .setAccessType("offline")
    .build();
```

Рисунок 2.3 – Код авторизації. [9]

Програма повинна зберігати маркер оновлення для подальшого використання та використовувати маркер доступу для права користування Google API. Після закінчення терміну дії маркера доступу програма використовує маркер оновлення для отримання нового.

Для опрацювання цього запиту та подальшої реалізації коду програми існує шаблон Observer, який реагує на успішне виконання однієї частини коду, щоб

запустити іншу.

2.2 Використання Google API

```
GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.Builder(
    HTTP_TRANSPORT, JSON_FACTORY, clientSecrets, SCOPES)
    .setDataStoreFactory(new FileDataStoreFactory(tokenfolder))
    .setAccessType("offline")
    .build();
```

Рисунок 2.4 – Код авторизації.

Програма повинна зберігати маркер оновлення для подальшого використання та використовувати маркер доступу для доступу до Google API. Після закінчення терміну дії маркера доступу програма використовує маркер оновлення для отримання нового.

Для подальшої роботи з маркером, його зберігання на сучасних пристроях Android нам потрібен дозвіл користувача до External Storage. В класичному виді ця процедура має йти через паттерн Observer.

TedPermission дозволяє скористатися його методом .create, який значно скорочує процес створення та модифікування коду з дозволами та паттерном Observer. Докладніше питання роботи з TedPermission висвітлено в [13].

```
TedPermission.create()
    .setPermissionListener(permissionlistener)
    .setDeniedMessage("If you reject permission,you can not use this service\n\nPlease turn on permi
    .setPermissions(Manifest.permission.READ_CONTACTS, Manifest.permission.ACCESS_FINE_LOCATION)
    .check();
```

Рисунок 2.5 – Метод дозволу користувачем. [13]

Метод OnPermissionGranted буде спрацьовувати кожен раз як користувач

буде погоджуватися з дозволом на модифікацію зовнішнього сховища і дозволяє нам створити кореневу папку для зберігання токена та подальшої роботи програми.

```
@Override
public void onPermissionGranted() {
    Toast.makeText(context: MainActivity.this, text: "Permission Granted", Toast.LENGTH_SHORT).show();

    File tokenFolder = new File(pathname: Environment.getExternalStorageDirectory() +
        File.separator + "tokens");
    if (!tokenFolder.exists()) {
        tokenFolder.mkdirs();
    }
}
```

Рисунок 2.6 – Метод продовження реалізації програми.

Після успішного повернення токена для авторизації ми можемо користуватися Sheets API та сконструювати наш об'єкт Sheets, наш запит до документу.

```
service = new Sheets.Builder(HTTP_TRANSPORT, SheetsImpl.getJsonFactory(), SheetsImpl.getCredentials(HTTP_TRANSPORT, tokenFolder))
    .setApplicationName(SheetsImpl.getApplicationName())
    .build();
```

Рисунок 2.7 – Ініціалізація ідентифікатора клієнта.[8]

Та метод для використання таблиці вказаної в range.

```
response = service.spreadsheets().values()
    .get(sheetId, range)
    .execute();
```

Рисунок 2.8 – Ініціалізація ідентифікатора клієнта.[8]

І остання процедура з API це витягування даних з таблиці у матрицю.

```
List<List<Object>> values = response.getValues();
```

Рисунок 2.9 – Ініціалізація ідентифікатора клієнта.[8]

Після цього нам потрібно тільки написати функцію зображення даних, будь то через консоль або через графічний об'єкт в дизайнері AD.

2.3 Використання Google Sheets API

Після успішної авторизації в Google API ми починаємо працювати з самим Sheets API, котрий посилається на наш ID (який є частиною посилання на документ в браузері) та для виконання більшості операцій зчитування та писання використовує діапазон (range) – покажчик масштабу дії коду над даними.

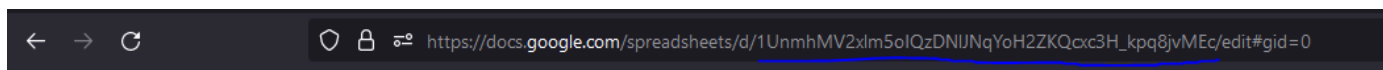


Рисунок 2.10 – Посилання та ID на тестовий документ.

```
final String spreadsheetId = "1UnmhMV2xlm5oIQzDNIJNqYoH2ZKQcxc3H_kpq8jvMEc";
final String range = "Sheet1!A2:C";
Sheets service = null;
```

Рисунок 2.11 – Головні параметри Sheets API .

Тілом запиту на зчитування має бути range, в якому можна скоротити позначку шуканого, хоча єдиним обов'язковим полем є вкладка документу(в цьому випадку Sheet1). Якщо вказано діапазон(A2:C), він повинен відповідати діапазону в URL-адресі. У діапазоні range можна за бажанням вказати його основний вимір. За замовчуванням використовується rows. Якщо вказано columns, кожен внутрішній масив записується в стовпець замість рядка.

Діапазон введення використовується для пошуку наявних даних і пошуку підтаблиці в цьому діапазоні. Значення додаються до наступного рядка підтаблиці, починаючи з першого стовпця таблиці. Наприклад, розглянемо таблицю у вкладці Sheet1, яка виглядає так:

	A	B	C	D	E
1	x	y	z		
2	x	y	z		
3					
4		x	y		
5			y	z	
6		x	y	z	
7					

Рисунок 2.12 – Діаграма – таблиця. [8]

На рис. 2.12 є дві таблиці: A1:C2 і B4:D6. Значення будуть додаватися з B7 для всіх наступних вхідних даних діапазону:

- Sheet 1, оскільки він перевіряє всю вкладку, визначає, що таблиця в B4:D6 є останньою таблицею.
- B4 або C5:D5, тому що вони обидва знаходяться в таблиці B4:D6.
- B2:D4, тому що останньою таблицею в діапазоні є таблиця B4:D6 (незважаючи на те, що вона також містить таблицю A1:C2).
- A3:G10, тому що останньою таблицею в діапазоні є таблиця B4:D6 (незважаючи на те, що починається перед і закінчується після неї).

Наступні вхідні дані діапазону не починатимуть запис з B7:

- A1 почне писати з A3, тому що це в таблиці A1:C2.
- E4 почне писати з E4, тому що його немає в жодній таблиці. (A4 також почне писати на A4 з тих же причин.)

2.3.1 Запис даних у JSON файл

Google Sheets має можливість публікації документу у веб та серед підтримуваних форматів присутня класична веб сторінка. Якщо в налаштуваннях надати таку можливість документу, то можна написати метод для зчитування та запису даних в JSON, що і зробив opensheet API.

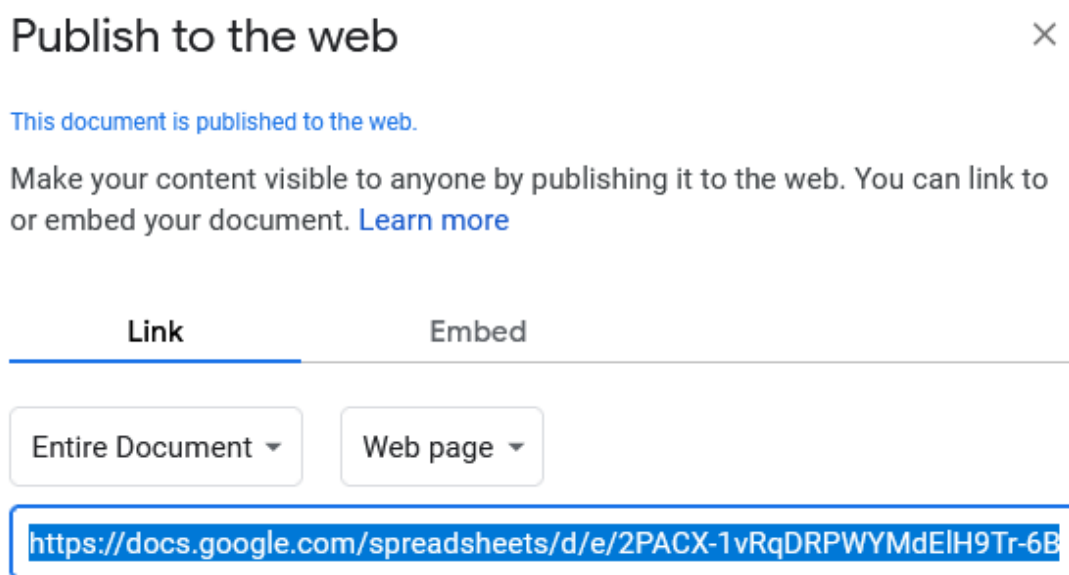
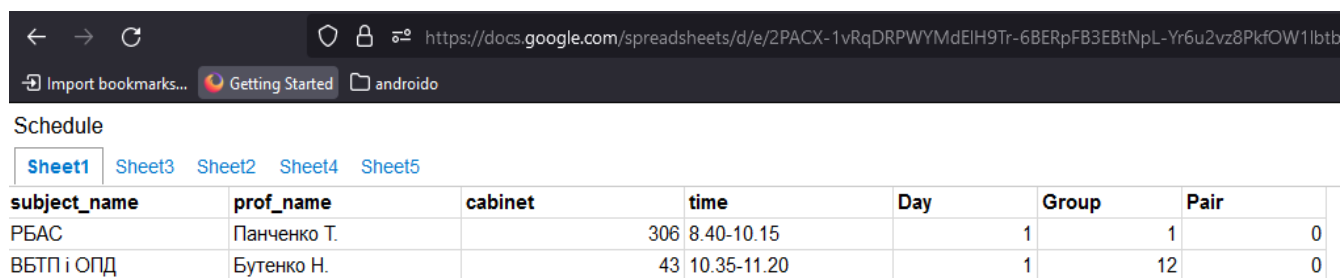


Рисунок 2.13 – Публікація документа.

Можна помітити, що цей етап лімітує наші таблиці, оскільки для успішної серіалізації у JSON нам потрібно структурувати дані через заголовки для кожної колонки. Також ця процедура змінює посилання на дані, тому в цьому методі додаток навіть не звертається безпосередньо до головного документа.

Opensheet API використовує посилання на документ та формує JSON масив JSON файлів з рядків даної таблиці. На рис 2.14 показано один документ, який має п'ять таблиць та сім однакових заголовків в кожному з них. Як виглядають ці ж дані у JSON можна побачити на рис. 3.8 у підрозділі 3.3 “Підготовка даних”.



Schedule

Sheet1 Sheet3 Sheet2 Sheet4 Sheet5

subject_name	prof_name	cabinet	time	Day	Group	Pair
РБАС	Панченко Т.	306	8.40-10.15	1	1	0
ВБТП і ОПД	Бутенко Н.	43	10.35-11.20	1	12	0

Рисунок 2.14 – Приклад таблиці Google Sheets.

3 АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ РОЗРОБЛЮВАНОВОГО ДОДАТКУ

Модернізація процесу це найголовніша мета цього проекту. Тому в даному випадку зробити тільки доступний та зручний варіант для відображення розкладу для студентів буде замало, треба також реалізувати більш привабливий метод заповнення та модифікації розкладу персоналом.

Google Sheets це онлайн аналог Microsot Excel з дуже схожим, але не ідентичним інтерфейсом. Цікава його особливість у тому, що він дає змогу ділитися між користувачами (Google акаунтами) правом для редагування документу та має перманентне посилання для доступу кожному, хто її відкриє. Якщо в університеті декілька осіб редагують розклад, то це не є проблемою ні для синхронізації усіх змін, ні для паралельної роботи або доступу до документа.

Алгоритм реалізації полягає у тому, щоб вирішити дві головні проблеми:

1. Надання додатку доступу до інформації з даного документу в Google Sheets.
2. Реалізації та синхронізації даних з додатком в графічному виді.

Виконавши цей проєкт двома способами, я буду розглядати обидва з них в цієї роботі.

3.1 Загальна структура додатку

Розробники програми для Android завжди віддають перевагу розробкам із застосуванням шаблону архітектури програмного забезпечення. Шаблон архітектури надає модульність файлам проекту . Це полегшує розробникам задачу підтримувати програмне забезпечення та розширювати можливості програми в майбутньому. Існують деякі архітектури, які дуже популярні серед розробників, і одна з них — шаблон Model—View—Controller (MVC). Шаблон MVC пропонує

розділити код на три компоненти. Під час створення класу/файлу програми розробник повинен розділити його на один із трьох наступних рівнів:

- Model(Модель): Цей компонент зберігає дані програми. Він не взаємодіє з інтерфейсом. Модель відповідає за обробку логіки домену і зв'язку з базою даних і мережевими рівнями.
- View(Перегляд): це Інтерфейс користувача, який містить компоненти, видимі на екрані. Крім того, він забезпечує візуалізацію даних, що зберігаються в Model, і пропонує взаємодію з користувачем.
- Controller(Контролер): цей компонент встановлює зв'язок між View і Model. Він містить основну логіку програми та отримує інформацію про поведінку користувача та оновлює Model відповідно до потреб.[21]

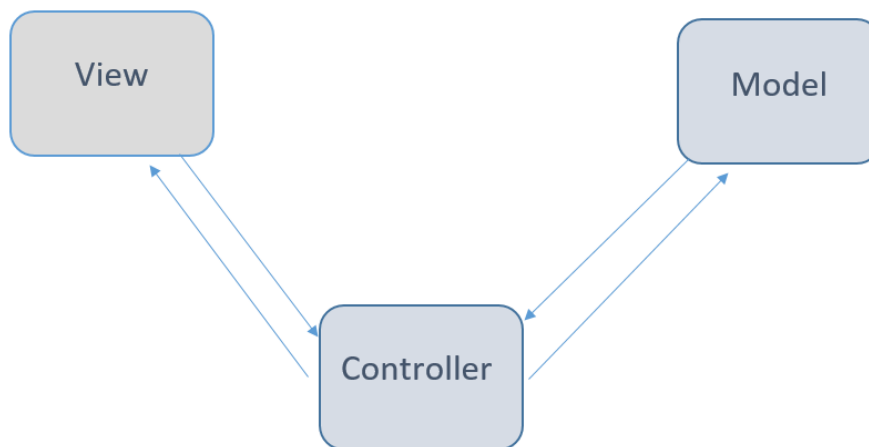


Рисунок 3.1 - Класична діаграма MVC [21]

Незважаючи на застосування схеми MVC для модульного дизайну програми, кодові рівні все ще залежать один від одного. У цьому шаблоні View і Controller залежать від Model. Для застосування шаблону MVC у проекті можна використовувати кілька підходів:

Підхід 1: дії та фрагменти можуть виконувати Controller роль та відповідати за оновлення View.

Підхід 2: використовувати Activity або його фрагменти як View і Controller, тоді Model буде окремим класом, який не розширює жодного класу Android.[22]

Даний додаток матиме 3 кнопки, і кожна з них відображає розклад для даної групи спеціальності на факультеті, коли користувач натиснув цю кнопку. Для розробки цієї програми код був розділений таким чином:

Controller і View будуть оброблятися за допомогою MainActivity. Щоразу, коли користувач натискає кнопки, активність спрямовує Model на виконання подальших операцій. MainActivity буде виконувати роль Observer'a.

Model буде окремим класом, який містить дані для відображення. Операції над даними будуть виконуватися функціями цього класу, і після оновлення значень даних цей клас повідомляє Observer'a (MainActivity) про зміну.

3.1.1 Паттерн Observer

Observer (Спостерігач) – це шаблон публікації / підписки в АД, який дозволяє кільком об'єктам спостерігача реагувати на подію.

Цей паттерн використовується для того, щоб зменшити зв'язність між компонентами програми та забезпечити розширюваність та модифікованість коду. Потреба у використанні певної варіації “спостерігача” з'являється при розробці графічного інтерфейсу користувача. Зазвичай, розробники графічної бібліотеки або фреймворку пропонують свій варіант цього шаблону програмування.

На Android є кілька способів перехопити події, які відбуваються під час взаємодії користувача з програмою. Під час розгляду подій в інтерфейсі користувача підхід полягає в тому, щоб фіксувати події з конкретного об'єкта View, з яким взаємодіє користувач. Клас View надає засоби для цього. Цей клас можна розширити для написання власних користувацьких компонентів.

При розробці програми упродовж її життєвого циклу може знадобитися розширювати функціонал(наприклад додати додаткову поведінку при натисканні на кнопку). У такому випадку треба “підписатися” на слухача подій натискання на кнопку.

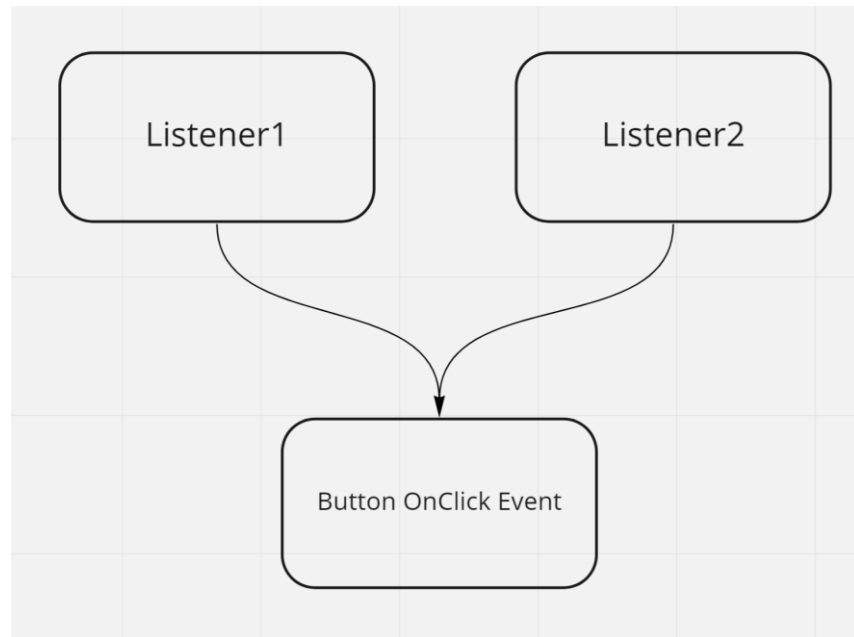


Рисунок 3.2 – Observer приклад [13]

Слухач подій (event listener) — це інтерфейс у класі View, який містить один метод зворотного виклику. Ці методи будуть викликатися фреймворком Android, коли представлення (view), в якому зареєстровано слухача, ініціюється взаємодією користувача з елементом в інтерфейсі користувача.

Для обробки натискання кнопки слід використовувати `onClick()` з `View.OnClickListener`. ”Це викликається, коли користувач торкається елемента (у режимі дотику), або фокусується на елементі за допомогою навігаційних клавіш або трекбола і натискає відповідну клавішу «enter» або натискає на трекбол.” [1]

3.2 Загальна конфігурація проєкту в AD

З початку ми створюємо проєкт у AD за даними настройками:
No activity -> Language: Java -> Minimum SDK: API 21

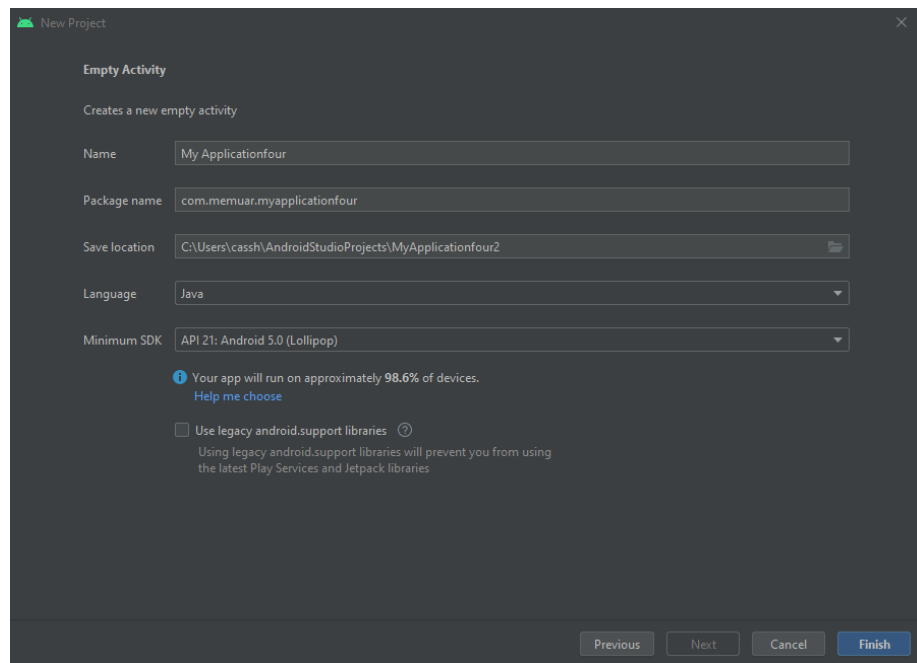


Рисунок 3.3 – Налаштування AD при створенні проєкту.

Потім у `build.gradle(app)` доповнюємо “dependencies“ важливими для нас залежностями, які грають роль додаткових бібліотек, завантажених з вебу. Нам потрібні такі залежності, як :

- 1) `google-api-client:1.33.0` - Базова Google API
- 2) `google-oauth-client-jetty:1.32.1` - Google OAuth2, апі для роботи з Google акаунтами, верифікацією тощо
- 3) `google-api-services-sheets:v4-rev20210629-1.32.1` - Google Sheets API
- 4) `tedpermission-normal:3.3.0` - Невелика залежність для AD, яка дозволяє інакше працювати з дозволом на додатковий функціонал, що грає важливу роль у розробці під Android.

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.6.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.api-client:google-api-client:1.33.0'
    implementation 'com.google.oauth-client:google-oauth-client-jetty:1.32.1'
    implementation 'com.google.apis:google-api-services-sheets:v4-rev20210629-1.32.1'
    implementation 'io.github.ParkSangGwon:tedpermission-normal:3.3.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

Рисунок 3.4 - Залежності Gradle

У девайс менеджері створюємо два девайси – один за мінімальними вимогами, а інший за настройками:

Phone->Pixel 2->Android 9-> на останньому екрані не виконуємо змін та підтверджуємо створення.

Далі прописуємо увесь потрібний додатковий функціонал пристрою з яким нам потрібно працювати в AndroidManifest.xml, а це:

- Дозвіл READ(WRITE)_EXTERNAL_STORAGE на читання та запис у зовнішню пам'ять
- Дозвіл на вихід в інтернет з пристрою
- Дозвіл ACCESS_NETWORK_STATE з-за для можливості дебагеру розуміти стан причетності до мережі
- GET_ACCOUNTS дозволяє працювати з акаунтами у акаунт сервісі, загальний google account якого є частиною.

3.3 Підготовка даних

Розглянемо вхідні дані з Google Sheets документу та Microsoft Excel

таблицею розкладу КНУ Шевченка кібернетики 2021-2022 другого семестру задля з'ясування, яка різниця в інформації та оформленні існують між ними.

На рис.3.5 ілюстрований тестовий варіант таблиці, де виділені жирним шрифтом заголовки диктують структуру документу та є найважливішими компонентами успішного форматування в JSON файл:

- 1) subject_name - назва предмету
- 2) prof_name - ім'я викладача
- 3) cabinet - номер аудиторії
- 4) time - початок та кінець пари
- 5) Day - день тижня
- 6) Group - підгрупа
- 7) Pair - парність/непарність тижня

	A	B	C	D	E	F	G
1	subject_name	prof_name	cabinet	time	Day	Group	Pair
2	Test Subject	Test Name		0 00.00		0 0	0

Рисунок 3.5 – Результат тестового запиту.

1	subject_name	prof_name	cabinet	time	Day	Group	Pair
2	Нейроні мережі	Пашко А.		308 8.40-10.15		1	1 0
3	ВБТП і ОПД	Бутенко Н.		43 10.35-11.20		1	12 0
4	Інформаційні технології	Федорус О.		15 12.20 - 13.05		1	2 0
5	Нейроні мережі	Кондратюк С.		9 8.40-10.15		2	12 0
6	Інтелектуальні системи	Глибовець М.		1 10.35-11.20		2	12 1
7	Операційні системи	Коваль Ю.В		233 12.20 - 13.05		2	12 2
8	Розпізнавання образів	Трохимчук Р.		221 8.40-10.15		3	12 0
9	Розпізнавання образів	Трохимчук Р.		221 10.35-11.20		3	12 1
10	ПЛІ	Крак Ю.		42 10.35-11.20		3	12 2
11	ПВМ	Вергунова І.		10 8.40-10.15		4	1 0
12	Телекомунікаційні технології	Зарембовський		704 10.35-11.20		4	1 0
13	Інформаційні технології	Вергунова І.		204 12.20 - 13.05		4	12 0
14	Інформаційні технології	Вергунова І.		204 10.35-11.20		4	12 1
15	Операційні системи	Коваль Ю.		3 12.20 - 13.05		4	1 2

Рисунок 3.6 - Розклад ТК42 в Google Sheets на тиждень.

4 КУРС

TK-41	TK-42	ТПП	МІ
Інформатика			
Нейроні мережі та нейрообчислення (лек.,20год) 306 п. Пашко А.О.	Розробка бізнес аналітичних систем (лек.,10год) 308 д. Панченко Т.В.	Проблеми рефакторингу (лек.,22год) 705 д. Кулябко П.П.	
Вибрані розділи трудового права і основ підприємницької діяльності (лек., 20год) 43 проф. Бутенко Н.В., ас. Богдан І.А.			
Інформаційні технології (лаб.,18год) *** а. Федорус О.М. *** а. Тарануха В.Ю.	Вибрані розділи трудового права і основ підприємницької діяльності (с.,8год) 311 проф. Бутенко Н.В., ас. Богдан І.А.	Коректність програм та логіки програмування (с.,8год) 311 а. Ткаченко	Інформаційні технології в менеджменті (с.,8год) 704 п. Вергунова І.М.
	Операційні системи з розподілом часу (с.,8год) 311 а. Веклич	Інформаційні технології в менеджменті (с.,8год) 704 п. Вергунова І.М.	Вибрані розділи трудового права і основ підприємницької діяльності (с.,8год) 312 проф. Бутенко Н.В., ас. Богдан І.А.

Рисунок 3.7 - Приклад Excel розкладу факультету 2021-2022

Задля того, щоб зобразити дані розкладу з Excel розкладу в Sheets потрібно зробити важливе рішення щодо структури документу:

1) Створити таблицю для кожної спеціальності та кафедри на тиждень в одній таблиці, тоді кількість таблиць буде дорівнювати кількості існуючих спеціальностей, років навчання та кафедр в них.

2) Створити таблицю для кожного дня тижня і мати там усі спеціальності, роки, кафедри та додати ще один параметр, який буде розділяти кожен при зчитуванням додатком.

Вибір між нагромадженням параметрів або таблиць робить кожний з варіантів компромісним.

```

JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
time: "10.35-11.20"
Day: "3"
Group: "12"
Pair: "2"
9:
  subject_name: "ITBM"
  prof_name: "Вергунова І."
  cabinet: "10"
  time: "8.40-10.15"
  Day: "4"
  Group: "1"
  Pair: "0"
10:
  subject_name: "Телекомунікаційні технології"
  prof_name: "Зарембовський К."
  cabinet: "704"
  time: "10.35-11.20"
  Day: "4"
  Group: "1"
  Pair: "0"
11:
  subject_name: "Інформаційні технології"
  prof_name: "Вергунова І."
  cabinet: "204"
  time: "12.20 - 13.05"
  Day: "4"
  Group: "12"
  Pair: "0"
12:
  subject_name: "Інформаційні технології"
  prof_name: "Вергунова І."
  cabinet: "204"
  time: "10.35-11.20"
  Day: "4"
  Group: "12"
  Pair: "1"
13:
  subject_name: "Операційні системи "
  prof_name: "Коваль Ю."
  cabinet: "3"
  time: "12.20 - 13.05"
  Day: "4"
  Group: "1"
  Pair: "2"

```

Рисунок 3.8 - JSON формат даних з Sheet1.

3.4 Отримання даних

Ініціалізація request як JsonArrayRequest надає змогу звернутися до JSON масиву з посилання на веб сторінку конвертованого документу.

```
JSONArrayRequest request = new JSONArrayRequest(Request.Method.GET, url, jsonRequest: null, new Response.Listener<JSONArray>() {
    @Override
    public void onResponse(JSONArray response) {
```

Рисунок 3.9 – Запит даних.[19]

Далі відбувається створення нового об'єкта json і отримання кожного об'єкта з нашого масиву json. Наступним кроком ми отримуємо відповідь від API у форматі об'єкта json, у нижньому рядку ми витягуємо рядок з його значення ключа з нашого об'єкта. Аналогічно так ми витягуємо всі рядки.

```
public void onResponse(JSONArray response) {
    for (int i = 0; i < response.length(); i++) {

        try {
            JSONObject responseObj = response.getJSONObject(i);

            String subjectName = responseObj.getString( name: "subject_name");
            String prof_name = responseObj.getString( name: "prof_name");
            int cabinet = responseObj.getInt( name: "cabinet");
            String time = responseObj.getString( name: "time");
```

Рисунок 3.10 – Отримання даних.

3.5 Графічний інтерфейс

Для тестування я зробив базу даних щоденного розкладу спеціальності 121 КНУ факультету комп'ютерних наук та кібернетики для четвертого курсу.

Програма дозволяє вибрати розклад для заданої групи.

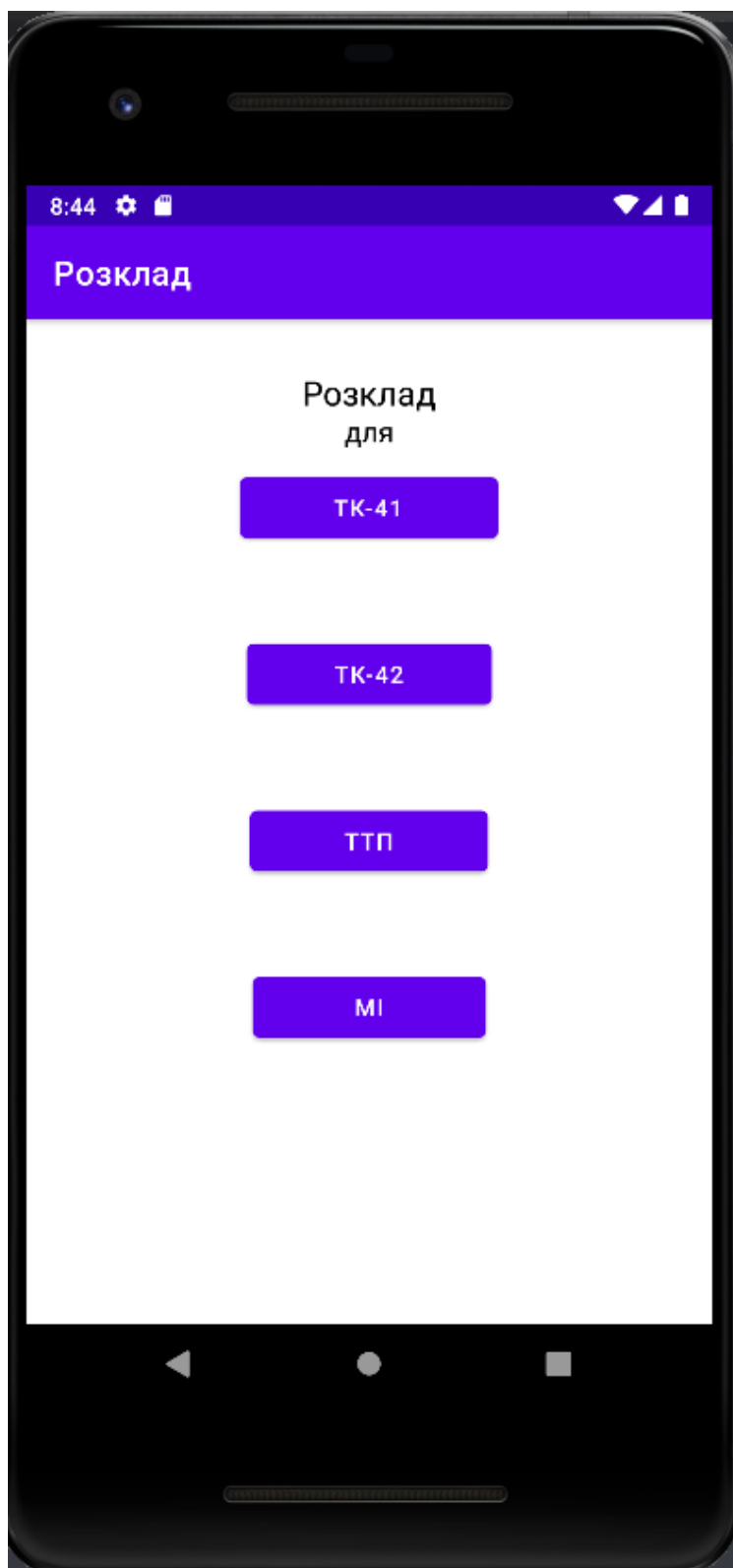


Рисунок 3.11 - Головний екран додатка

Після вибору групи можна вибрати розклад для дня тижня.

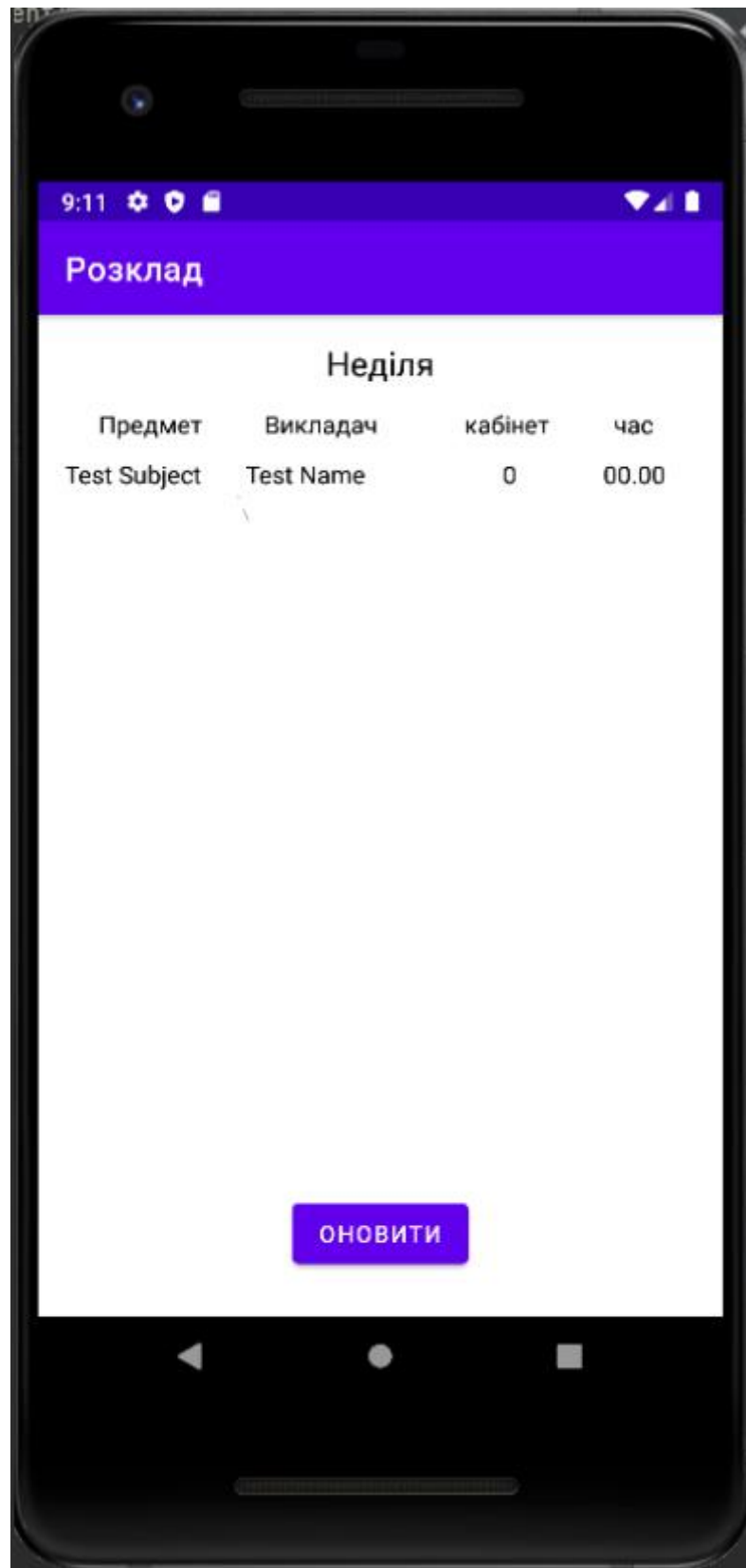


Рисунок 3.12 - Результат тестового запиту

8:49

Розклад

Четвер

Предмет	Викладач	кабінет	час
ITVM	Вергунова І.	10	8.40-10.15
Телекомунікаційні технології	Зарембовський К.	704	10.35-11.20
Інформаційні технології	Вергунова І.	204	12.20 - 13.05
Операційні системи	Коваль Ю.	3	12.20 - 13.05

ОНОВИТИ

Рисунок 3.13 - Результат ТК42 розкладу на четвер.

3.5 Програмно-апаратні вимоги та розгортання

Мінімальні вимоги для використання додатку пристроєм:

- Android version 5.0 (Lollipop)
- 64 bit Quad core (Minimum A53, 1.0GHz),
- 512MB RAM
- 2GB storage

Рекомендовані моделі:

- Google Pixel
- Google Pixel 2
- Google Pixel 2 XL
- Google Pixel 3
- Google Pixel 3 XL
- Google Pixel 3A
- Google Pixel 4A

Додаток використовує стандартний функціонал Android та не вимагає спеціальних дозволів для користування, будь то root або інше. Для завантаження потрібно лише отримати копію файлу .apk та дати дозвіл йому на встановлення пристроєм.

На даний момент стабільне та коректне виконання можливо лише на смартфонах.

3.6 Інструкція користувача

1. Завантажити додаток з копії .apk файлу
2. Надати дозвіл пристрою встановити .apk файл
3. Мати доступ до мережі Інтернет

4. Запустити додаток
5. Обрати потрібні настройки (Рік, спеціальність, групу)
6. Оновити результати

ВИСНОВКИ

В якості бази даних у проєкті використаний Google Sheets, аналог Microsoft Excel від Google. Здебільшого перевагою є централізованість сервісу - Google Sheets тісно пов'язан з Google Drive та Google акаунтом, що утворюють єдине місце для зберігання, поширення та модифікації документа. Хоча Microsoft теж використовує схожу систему з One Drive, але сервіс який надає Google безкоштовний, коли Microsoft просить щомісячну плату.

Основним результатом виконання роботи стало створення базового функціоналу програмного засобу для відображення навчального розкладу. Під час виконання роботи було виявлено та порівняно методи для добування даних з необхідного документа Google Sheets до додатку. Також було розглянуто базовий робочий процес з AD.

Класичний метод добування даних через Google API є більш комплексним, ніж проєкт цього потребує. Так, у випадку опрацювання таблиці він має декілька більш гнучких рішень, якщо робити ці таблиці менш централізованими, але для наших задач це можна опустити.

Ахіллесовою п'ятою роботи через Google API є аутентифікація та проблеми з нею. Вже через тиждень в моєму проєкті були знайдені проблеми з токеном та його оновленням.

Усі методи в Google API вимагають доступ до Google акаунту юзера, будь то мастер акаунта з документом або іншого. Це недоцільно. Прив'язка акаунта не дає змоги через API використовувати це для наших задач (наприклад, зробити що прив'язаний акаунт буде пам'ятати з якого він курсу та звертатися до частини документа з курсом), а тільки ускладнює реалізацію такого функціонала, якщо ми вирішимо використовувати цей параметр. Google Client Secrets не дає змоги так користуватися і давати йому нові параметри.

Рішення з JSON є менш гнучким, але більш простим та надійним. Ця

надійність виражається у можливості налаштувати сервер і не мати потреби брати до уваги застарілість рішення, його відключення, коли на іншому кінці робота з Google API змінилася за останній рік - закrywся старий бранч spreadsheets v3, яким я користувався на початку навчального року.

Google API шлях :

- + Класичний варіант, який завжди буде мати доступний шлях реалізації
- + Найбільша гнучкість роботи з таблицями, будь то зчитування або оновлення даних
- + Найбільша свобода для корекції таблиць, структуризації
- Волатильний, документація змінюється та технологія з проекту колишнього року застаріла та вже немає підтримки
- Більш комплексний для роботи, більше можливих проблем на кожному рівні навіть після успішної конфігурації

JSON шлях:

- + Простіший варіант, менше документації для реалізації функціоналу
- + Надійніший варіант, працює з самими базовими варіантами веб сторінок
- + Не потребує авторизації
- Тільки одна структура таблиці
- Лімітована утилізація таблиць, тільки зчитування

Google API добре підійде для проєктів, де є задача модифікувати дані з обох сторін (джерела та пристрою), коли JSON може добре підійти для даного проєкту – тільки зчитування дати з обмеженої та впорядкованої бази даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software – Addison Wesley, 1994. ISBN 0-201-63361-2.
2. B. Phillips Android Programming: The Big Nerd Ranch Guide – Big Nerd Ranch Guides, 2019. ISBN 978-0135245125.
3. N. Smith Android Studio 3.0 Development Essentials – CreateSpace Independent Publishing Platform, 2017. ISBN 978-1977540096.
4. Переваги та недоліки розробки додатків для Android [Електронний ресурс] – Режим доступу до ресурсу : <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-android-app-development/>
5. Gradle Dependencies [Електронний ресурс] – Режим доступу до ресурсу : https://www.tutorialspoint.com/gradle/gradle_dependency_management.html
6. ACCESS_NETWORK and INTERNET [Електронний ресурс] – Режим доступу до ресурсу : <https://stackoverflow.com/questions/19642032/whats-the-difference-between-access-network-state-and-internet>
7. Початок роботи Java з Google API [Електронний ресурс] – Режим доступу до ресурсу : <https://developers.google.com/sheets/api/quickstart/java>
8. Читання та запис даних [Електронний ресурс] – Режим доступу до ресурсу : <https://developers.google.com/sheets/api/guides/values>
9. OAuth 2.0 in Google APIs [Електронний ресурс] – Режим доступу до ресурсу : <https://developers.google.com/identity/protocols/oauth2>
10. OAuth 2.0 for Mobile & Desktop Apps[Електронний ресурс] – Режим доступу до ресурсу : <https://developers.google.com/identity/protocols/oauth2/native-app>
11. AppAuth for Android [Електронний ресурс] – Режим доступу до ресурсу : <https://openid.github.io/AppAuth-Android/>

12. Input events overview [Электронный ресурс] – Режим доступа до ресурсу :
<https://developer.android.com/guide/topics/ui/ui-events>
13. TedPermission [Электронный ресурс] – Режим доступа до ресурсу :
<https://github.com/ParkSangGwon/TedPermission>
14. Introducing JSON[Электронный ресурс] Режим доступа до ресурсу:
<https://www.json.org/json-en.html>
15. Meet Android Studio[Электронный ресурс] Режим доступа до ресурсу:
<https://developer.android.com/studio/intro>
16. Gradle[Электронный ресурс] Режим доступа до ресурсу:
<https://github.com/gradle/gradle>
17. Android devices[Электронный ресурс] Режим доступа до ресурсу:
<https://www.theverge.com/2021/5/18/22440813/android-devices-active-number-smartphones-google-2021>
18. Excel and Google Sheets[Электронный ресурс] Режим доступа до ресурсу:
https://www.simplilearn.com/tutorials/excel-tutorial/google-sheets-vs-excel#key_differences
19. OpensheetInk[Электронный ресурс] Режим доступа до ресурсу:
<https://benborgers.com/posts/google-sheets-json>
20. Google Spreadsheets[Электронный ресурс] Режим доступа до ресурсу:
<https://www.techtarget.com/whatis/definition/Google-Spreadsheets>
21. Mvc Architecture[Электронный ресурс] Режим доступа до ресурсу:
<https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1>.
22. MVC in Android[Электронный ресурс] Режим доступа до ресурсу:
<https://www.geeksforgeeks.org/mvc-model-view-controller-architecture-pattern-in-android-with-example/>
23. Meet Android Studio[Электронный ресурс] Режим доступа до ресурсу:
<https://developer.android.com/studio/intro>