

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

“Мобільний додаток для перегляду навчального розкладу МК СумДУ”

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Будник М.М.

Студент групи ІН–73–9

Тимошенко Д.О.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 р.

Завдання
до випускної роботи

Студента четвертого курсу, групи ІН-73-9 спеціальності “Комп'ютерні науки”
денної форми навчання Тимошенко Дмитра Олександровича.

**Тема: “ Мобільний додаток для перегляду навчального розкладу МК
СумДУ”**

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) аналітичний огляд призначення мобільного додатку для розкладу; 2) постановка завдання й формування завдань дослідження; 3) огляд і опис засобів для розробки; 4) розробка мобільного додатку для перегляду навчального розкладу МК СумДУ; 5) аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Будник М.М.

Завдання прийняв до виконання _____ Тимошенко Д.О.

РЕФЕРАТ

Записка: 38 стор., 7 рис., 9 додатків, 10 джерел.

Об'єкт дослідження — Мобільний додаток для перегляду навчального розкладу МК СумДУ.

Мета роботи — розробити мобільний додаток, який повинен бути швидким при великій кількості інформації, мати малий розмір (до 10 мегабайт), простий в дизайні і зрозумілий для користувача. Додаток повинен бути адаптивним до всіх пристроїв на базі android 4.1 і вище.

Результати — розроблено мобільний додаток для платформи android 4.1+ на мові Java для МК СумДУ котрий надає можливість переглядати стандартний розклад, розклад замін занять та лабораторних робіт.

ANDROID, JAVA, SQLITE, ANDROID STUDIO, ANDROID SDK

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	6
1.1 Призначення мобільного додатку	6
1.2 Аналіз аналогів	6
1.3 Аналіз пов'язаних систем	7
1.4 Постановка задачі	7
2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ	8
2.1 Вибір засобів реалізації	8
2.2 Вибір середовища розробки	9
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	10
3.1 Проектування структури файлів додатку	10
3.2 Реалізація зовнішнього інтерфейсу	12
ВИСНОВКИ	17
СПИСОК ЛІТЕРАТУРИ	18
ДОДАТКИ	19

ВСТУП

Розвиток інформаційних технологій, освіти, технічних засобів та форм навчання нав'язують необхідність використання нових технологій у сфері освіти.

Незалежно від цілі автоматизації, будь то студентський склад або адміністрація інституту, в освітньому закладі такі системи впроваджують заради однієї кінцевої мети - підвищення якості освіти.

Заклад вищої освіти, як і будь-яке підприємство, неодмінно проходить процес автоматизації і, незважаючи на те, що поняття освітньої діяльності єдине для всіх освітніх установ, в кожному ЗВО цей процес проходить по-різному. Значний вплив на процеси автоматизації надає як наявність грошових коштів, так і готовність використання пропонованих ринком інформаційних послуг програмних продуктів.

На даний момент у закладі вищої освіти можна використовувати інформаційний простір, у тому числі компоненти з різних напрямків: кадровий облік співробітників, облік студентів, служба безпеки, рейтинг викладачів, контроль успішності студентів та інші. Робота співробітників в єдиному просторі організована на основі доменів Active Directory, що дозволяє виконувати централізоване управління, забезпечує єдиний процес входу в систему і масштабованість мережі.

Створення і впровадження мобільного розкладу в будь-який навчальний заклад спрощує освітнє життя студентів так як їм не потрібно фотографувати чи запам'ятовувати розклад. Також однією з позитивних сторін є можливе збільшення відвідуваності студентами занять і значне зменшення помилкового відвідування занять.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Призначення мобільного додатку

Одна з найважливіших задач організації навчального процесу у ЗВО є створення автоматизованого навчального розкладу. Правильно створений розклад надає можливість рівномірного розподілення завантаження студентських груп і викладацького складу.

Використання інформаційних систем у вищих навчальних закладах є звичністю. Діапазон їх використання широкий і варіюється від автоматизації окремо взятих робочих місць до повної автоматизації діяльності ЗВО.

Наразі ефективність використання педагогічного потенціалу залежить від декількох факторів, одним з яких є рівень організації навчального процесу. Важливою складовою є розклад занять, що установлює трудовий ритм і мотивованість як студентів, так і викладацького складу. [2]

Саме тому ведення розкладу має бути зручним і надійним. Можливість студентам в режимі онлайн дізнаватися розклад може призвести до зменшення кількості пропусків занять без поважних причин за рахунок того що студентам не потрібно запам'ятовувати розклад, так як при цьому є ймовірність помилки, а навпаки, вони завжди можуть дізнатися точний розклад на найближчі дні, в тому числі й тимчасові заміни занять.

В Сучасному світі в багатьох престижних закладах освіти є можливість використання розкладу за допомогою мобільного додатку, наприклад СумДУ.

Станом на весну 2021 року в Машинобудівному коледжі СумДУ присутній лише сайт. Тому було вирішено розробити мобільний додаток для коледжу.

1.2 Аналіз аналогів

На даний момент немає аналогів котрі можуть бути використані Машинобудівним коледжем так як останній має особливості в веденні розкладу.

Для проектування як за приклад було взято мобільний додаток розклад СумДУ так як він має достатньо простий дизайн котрий задовольняє потреби в дизайні і розкладу для МК СумДУ.

1.3 Аналіз пов'язаних систем

За умовою представника МК СумДУ мобільних додаток має бути пов'язаний с новим розкладом котрий наразі розроблюється, а саме отримувати дані з нього.

Серверна частина представлена в вигляді сайту, бази даних та API що дозволяє додатку отримувати дані в форматі JSON.

Мобільний додаток в разі необхідності може отримувати дані з API як замість розкладу та лабораторних робіт так і весь основний розклад.

1.4 Постановка задачі

У даній роботі необхідно розробити мобільний додаток з наступним функціоналом:

- отримання розкладу за групою, аудиторією чи викладачем;
- отримання розкладу за комбінацією групи, аудиторії чи викладача;
- отримання розкладу замість занять;
- отримання розкладу лабораторних робіт;
- можливість перегляду даних при відсутності інтернет підключення, тобто збереження даних.

Результатом роботи має стати мобільний додаток, який повинен бути швидким, простим в дизайні і зрозумілим для користувача. Також повинен бути адаптивним до всіх пристроїв і повинен працювати у всіх сучасних смартфонах.

2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Вибір засобів реалізації

Програми для Android можна писати на мовах Kotlin, Java та C++. Інструменти SDK для Android компілюють код разом з будь-якими файлами даних та ресурсів у файл .apk, пакет Android, який є архівним файлом із .apk суфіксом. Один файл APK містить весь вміст програми для Android і є файлом, який пристрої на базі Android використовують для встановлення програми. [1]

Кожна програма для Android живе у власній пісочниці безпеки, захищеній такими функціями безпеки Android:

- Операційна система Android - це багатокористувацька система Linux, в якій кожна програма є іншим користувачем.

- За замовчуванням система присвоює кожному додатку унікальний ідентифікатор користувача Linux (ідентифікатор використовується лише системою і невідомий додатку). Система встановлює дозволи для всіх файлів у програмі, так що доступ до них може отримати лише ідентифікатор користувача, призначений цій програмі.

- Кожен процес має власну віртуальну машину (VM), тому код програми працює ізольовано від інших програм.

- За замовчуванням кожна програма працює у своєму власному процесі Linux. Система Android запускає процес, коли потрібно виконати будь-який із компонентів програми, а потім вимикає процес, коли він більше не потрібен або коли система повинна відновити пам'ять для інших програм.[1]

Іншими альтернативами були Flutter, Ionic, ReactNative.

Серед описаних вище засобів вибір був зроблений на користь Java так як додатки реалізовані за допомогою нього мають високу швидкодію і достатню простоту використання.

Також для розробки під платформу android необхідно використати такий інструмент як Android SDK котрий дозволяє створювати додатки на основі коду Java.

Java + Android SDK дозволяють без підключення додаткових інструментів та засобів використовувати СУБД SQLite котра була використана для збереження даних на випадок відсутності інтернет підключення для оновлення даних з серверу.

2.2 Вибір середовища розробки

З середовищ розробки котрі доступні безкоштовно є Android Studio та IntelliJ IDEA Community Edition. Вибір пав на користь останнього за рахунок того що IDEA має весь функціонал що і Android Studio.

Для тестування було використано емулятор Android 11 з Android API версії 30, а також смартфон Samsung Galaxy M21 з Android версії 11.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проектування структури файлів додатку

Структура файлів проекту — це логічне розділення ресурсів додатку за призначення чи типом. Неправильна побудова структури проекту суттєво ускладнює підтримку і модернізацію додатку.

Структура проекту має наступний вигляд:

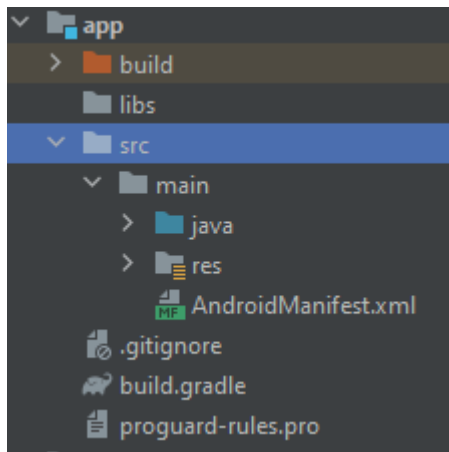


Рисунок 3.1 — Структура файлів

В випадку даного проекту варта уваги лише директорія `main` так як інші файли і папки є службовими і використовуються Android SDK при компіляції додатку.

Папка `main` містить ще дві папки, `java` та `res`.

Директорія `java` містить код проекту на мові Java та задає всю бізнес логіку проекту і описує всі процеси що проходять в додатку.

Директорія `res` містить `xml` файли що описують частину конфігурацій проекту та містить макети що оформлюють зовнішній інтерфейс мобільного додатку.

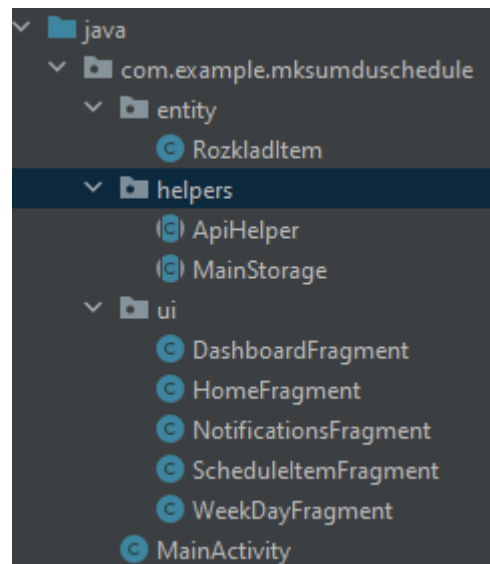


Рисунок 3.2 — Структура директорії java

В директорії java містяться наступні піддиректорії і файли:

– entity

Entity директорія що містить сутності що описують дані котрі використовуються в додатку. В даному випадку є лише одна сутність котра описує запис в розкладі.

– helpers

Helpers директорія описує абстрактні класи що мають лише статичні методи і властивості і котрі виконують допоміжні функції в бізнес логіці. ApiHelper реалізує отримання даних з серверу та їх первинну обробку. MainStorage реалізує збереження даних як на час роботи додатку так і на постійний час.

– ui

Ui директорія описує функціональність частин зовнішнього інтерфейсу. DashboardFragment, HomeFragment та Notificationfragment описують функціонал кожної з сторінок додатку. ScheduleItemFragment описує функціонал запису розкладу.

– MainActivity

MainActivity містить основний функціонал запуску додатку.

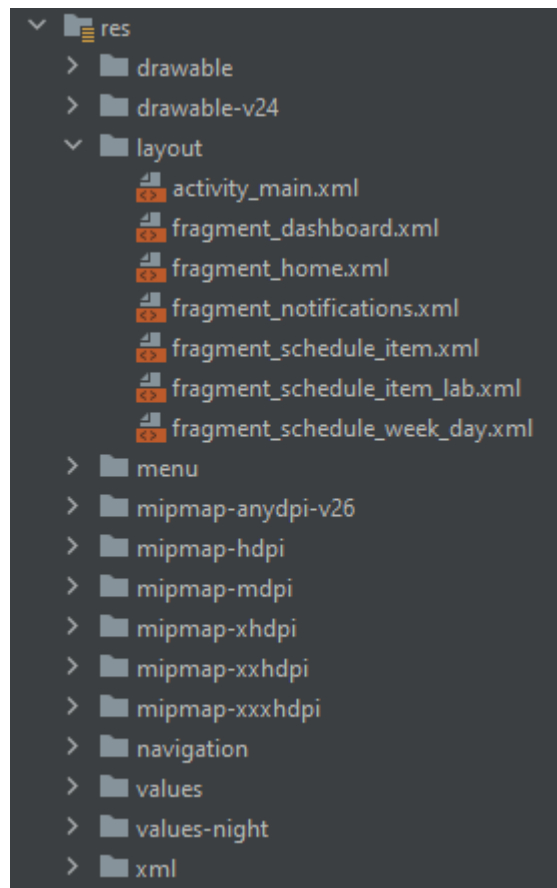


Рисунок 3.3 — Структура директорії res

В директорії res містяться наступні піддиректорії на котрих варто заголосити увагу:

- drawable

drawable директорія що містить іконки що використовуються для означення фільтрів та відображення додатку в списку додатків в меню android.

- layout

layout директорія що містить шаблони для створення зовнішнього інтерфейсу додатка. Fragment_schedule_item_lab.xml є альтернативою для відображення запису розкладу для випадку якщо це запис про лабораторну роботу.

3.2 Реалізація зовнішнього інтерфейсу

Через те що API серверної частини станом на 10.06.2021 не закінчене і

не у всіх розділах є дані то для презентації всі дані взяті с одної частини API, а саме "http://rozklad.mksumdu.info/api/schedule/4".

activity_main.xml відображує головне вікно додатку, і відповідає за нижню панель навігації, верхню панель та місце в центрі для інших фрагментів:

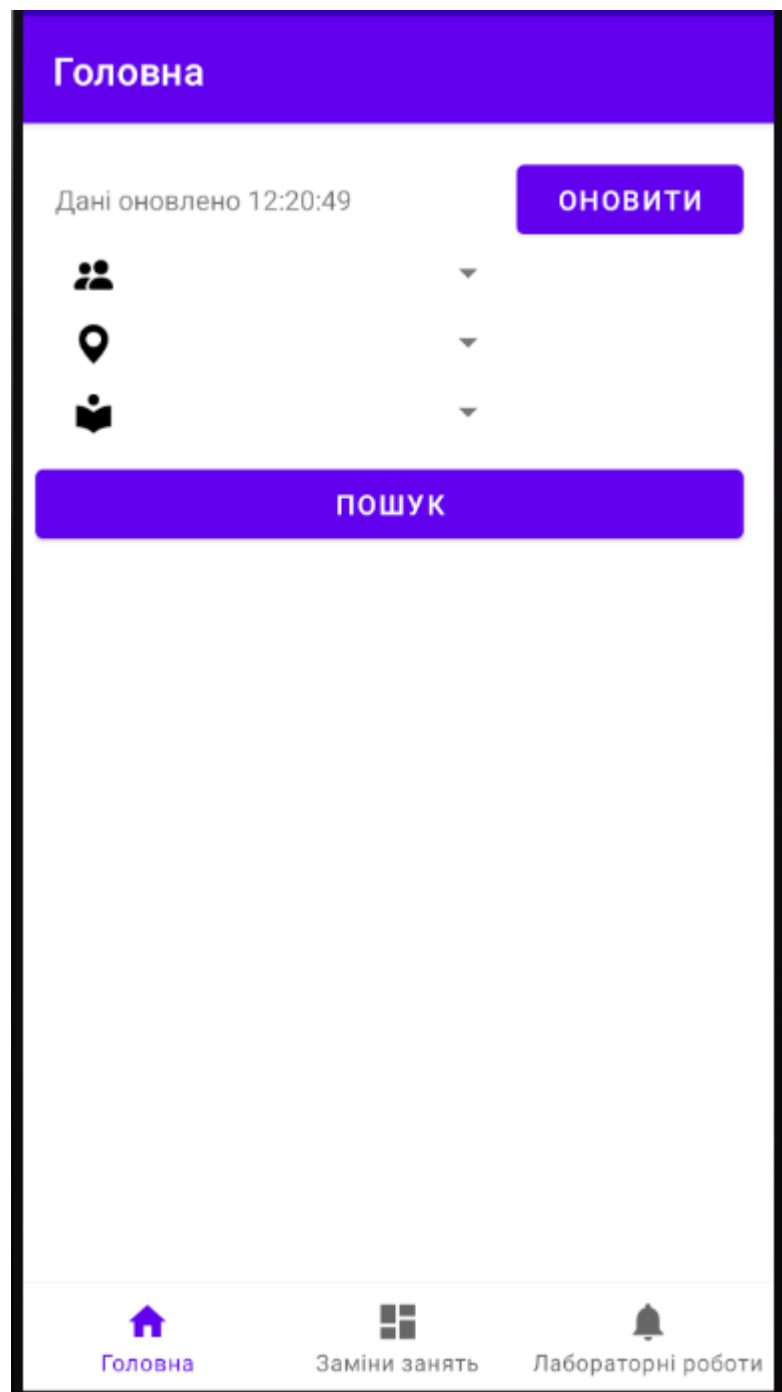


Рисунок 3.4 — Інтерфейс додатку після старту

fragment_home.xml відображує головний розділ додатку в котрому знаходяться фільтри для запиту розкладу по критеріям:

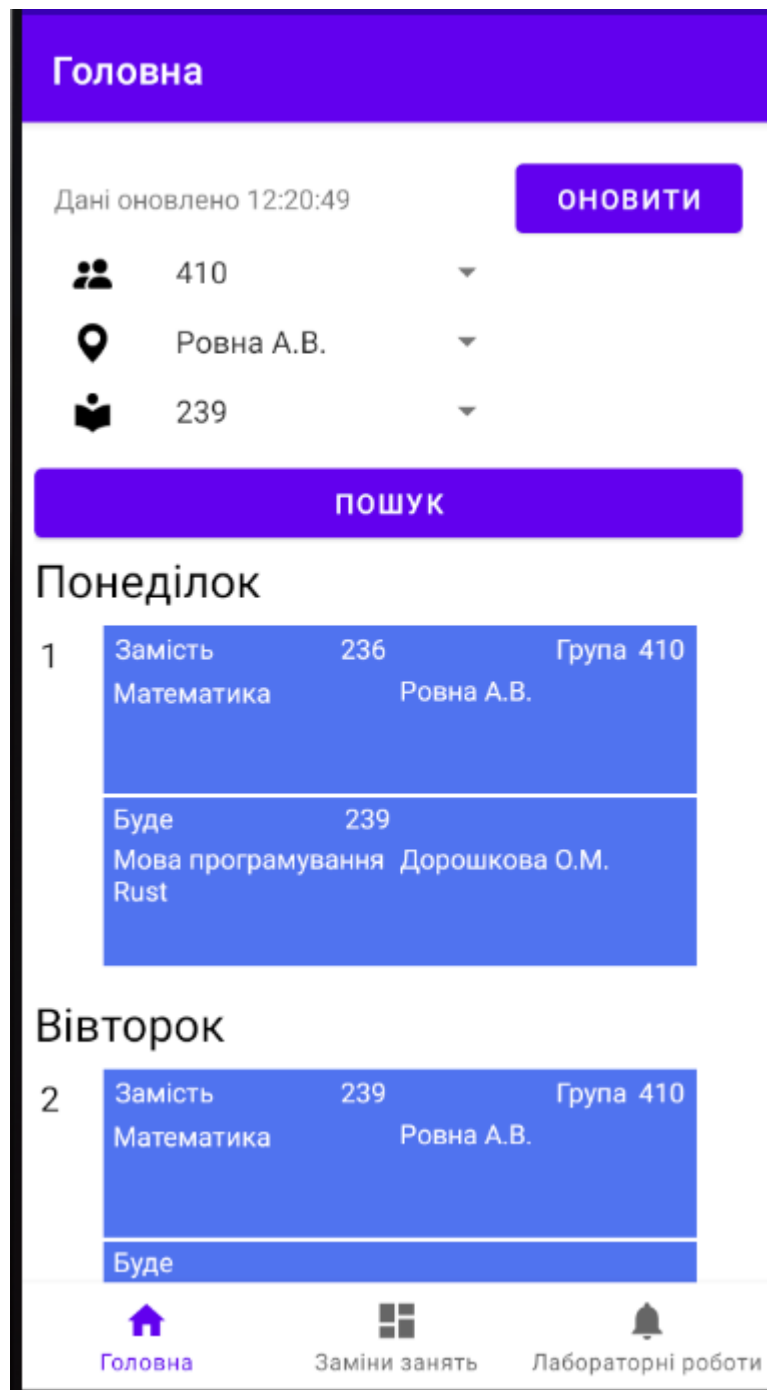


Рисунок 3.5 — Інтерфейс головного розділу

Після вибору параметрів і натиснення кнопки пошук (рис. 3.5) під кнопкою відобразиться список записів в розкладі котрі задовольняють умови пошуку. Дана сторінка як і дві інші можуть прогортатися вверх і вниз якщо

міста на екрані не вистачає.

Кнопка “Оновити” запускає процес оновлення даних з серверу і у випадку успіху оновлює показник часу останнього оновлення. Ця ж функція запускається також і після запуску додатку.

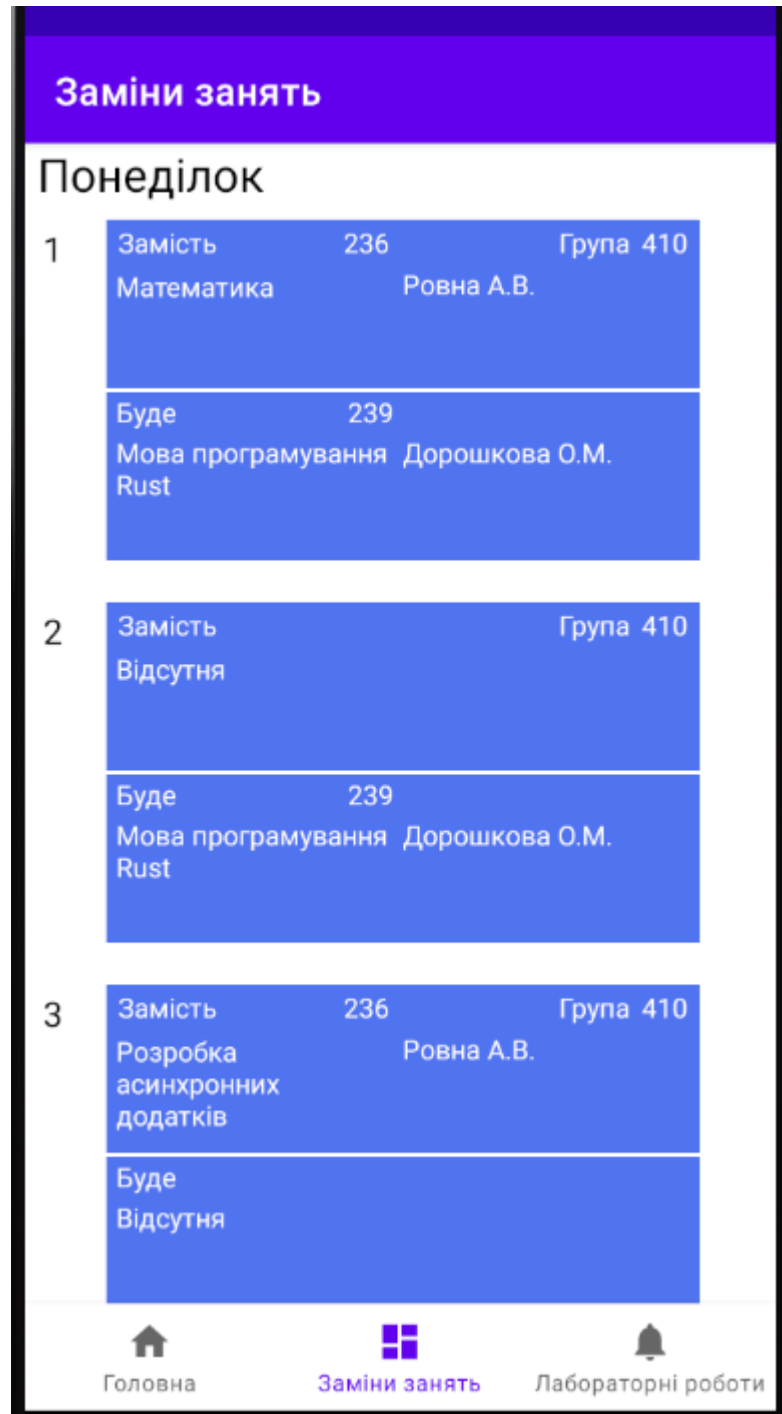


Рисунок 3.6 — Інтерфейс розділу замін занять

Після перемикання в нижньому меню на вкладку замін занять

відкривається сторінка як на рис.3.6. В даному розділі надається інформація про тимчасові зміни в стандартному розкладі.



Рисунок 3.7 — Інтерфейс розділу лабораторних робіт

Після перемикання в нижньому меню на вкладку лабораторних робіт відкривається сторінка як на рис.3.7. В даному розділі надається інформація про розклад найближчих лабораторних робіт. В даному випадку один із записів відображується некоректно так як дані призначені для головної

вклады.

ВИСНОВКИ

Була сформульована мета створити мобільний додаток для МК СумДУ що дозволить студентам зручніше отримувати інформацію про розклад занять в коледжі.

Під час створення дипломного проекту був проведений детальний аналіз предметної області. Було проведено вибір засобів реалізації та середовища розробки.

Завдяки використанню сучасних технологій вдалося створити функціональний, зручний та інтуїтивно-зрозумілий інтерфейс для різних категорій користувачів, яким легко користуватися у сучасних смартфонах та при різних розширеннях екранів. Також було здійснено тестування системи.

В результаті роботи було створено мобільний додаток, який безумовно полегшить процес освіти для студентів та дасть можливість легко дізнаватися актуальну інформацію про зміни в розкладі.

СПИСОК ЛІТЕРАТУРИ

1. Основи застосування | Розробники Android // <https://developer.android.com/guide/components/fundamentals>
2. «Система автоматизованого управління розкладом МК СумДУ»
Тимошенко Д.О. 2019р.
3. App resources overview | Android Developers // <https://developer.android.com/guide/topics/resources/providing-resources>
4. App Manifest Overview | Android Developers // <https://developer.android.com/guide/topics/manifest/manifest-intro>
5. Java и Android | Введение. Установка Android Studio и Android SDK // <https://metanit.com/java/android/1.1.php>
6. Java и Android | Первый проект в Android Studio // <https://metanit.com/java/android/1.2.php>
7. Java и Android | Создание графического интерфейса // <https://metanit.com/java/android/1.3.php>
8. Java. Экспресс-курс // <http://developer.alexanderklimov.ru/android/java/java.php>
9. Java. Экспресс-курс: Учебная программа // <http://developer.alexanderklimov.ru/android/java/sampleapp.php>
10. Android: Теория // <http://developer.alexanderklimov.ru/android/theory/>

ДОДАТКИ

Додаток А

MainActivity.java:

```
package com.example.mksumduschedule;

import android.os.Bundle;

import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

import com.example.mksumduschedule.helpers.MainStorage;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import androidx.appcompat.app.AppCompatActivity;

import androidx.navigation.NavController;

import androidx.navigation.Navigation;

import androidx.navigation.ui.AppBarConfiguration;

import androidx.navigation.ui.NavigationUI;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    public static MainActivity mainActivity;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);

        setContentView(R.layout.activity_main);

        BottomNavigationView navView = findViewById(R.id.nav_view);
```

```

        AppBarConfiguration appBarConfiguration = new
AppBarConfiguration.Builder(
        R.id.navigation_home, R.id.navigation_dashboard,
        R.id.navigation_notifications)
        .build();

        NavController navController = Navigation.findNavController(this,
        R.id.nav_host_fragment);

        NavigationUI.setupActionBarWithNavController(this, navController,
        appBarConfiguration);

        NavigationUI.setupWithNavController(navView, navController);

        MainActivity = this;

        MainStorage.getDataFromCache();
    }
}

```

Додаток Б

У даному додатку приведено код моделі RozkladItem.

```

package com.example.mksumduschedule.entity;

public class RozkladItem {

    public int number;

    public int numberOfDay;

    public String predmet1;

    public String predmet2;

    public String audit1;

    public String audit2;

    public String prepod1;

    public String prepod2;
}

```

```
public String group;

public String type;

    static public RozkladItem newMain(int number, int numberOfDay, String
predmet1, String predmet2, String audit1, String audit2, String prepod1, String
prepod2, String group) {

        RozkladItem rozkladItem = new RozkladItem();

        rozkladItem.number = number;

        rozkladItem.numberOfDay = numberOfDay;

        rozkladItem.predmet1 = predmet1;

        rozkladItem.predmet2 = predmet2;

        rozkladItem.audit1 = audit1;

        rozkladItem.audit2 = audit2;

        rozkladItem.prepod1 = prepod1;

        rozkladItem.prepod2 = prepod2;

        rozkladItem.group = group;

        rozkladItem.type = "1";

        return rozkladItem;

    }

    static public RozkladItem newReplacement(int number, int numberOfDay,
String predmet1, String predmet2, String audit1, String audit2, String prepod1,
String prepod2, String group) {

        RozkladItem rozkladItem = new RozkladItem();

        rozkladItem.number = number;

        rozkladItem.numberOfDay = numberOfDay;

        rozkladItem.predmet1 = predmet1;

        rozkladItem.predmet2 = predmet2;

        rozkladItem.audit1 = audit1;
```

```
rozkladItem.audit2 = audit2;

rozkladItem.prepod1 = prepod1;

rozkladItem.prepod2 = prepod2;

rozkladItem.group = group;

rozkladItem.type = "2";

return rozkladItem;

}

static public RozkladItem newLab(int number, int numberOfDay, String
predmet1, String audit1, String prepod1, String group) {

    RozkladItem rozkladItem = new RozkladItem();

    rozkladItem.number = number;

    rozkladItem.numberOfDay = numberOfDay;

    rozkladItem.predmet1 = predmet1;

    rozkladItem.audit1 = audit1;

    rozkladItem.prepod1 = prepod1;

    rozkladItem.group = group;

    rozkladItem.type = "3";

    return rozkladItem;

}

}
```

Додаток В

У даному додатку приведено код MainStorage:

```
package com.example.mksumduschedule.helpers;

import android.content.Context;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.widget.TextView;
import com.example.mksumduschedule.MainActivity;
import com.example.mksumduschedule.R;
import com.example.mksumduschedule.entity.RozkladItem;
import com.example.mksumduschedule.ui.HomeFragment;
import org.json.JSONException;

import java.io.IOException;
import java.sql.Time;
import java.util.ArrayList;
import java.util.HashSet;

abstract public class MainStorage {
    static public long timeOfUpdate;

    static public ArrayList<RozkladItem> data = new ArrayList<>();

    static public ArrayList<RozkladItem> cachedLab = new ArrayList<>();

    static public ArrayList<RozkladItem> cachedReplacement = new
ArrayList<>();

    static public String[] groups = new String[0];

    static public String[] audits = new String[0];

    static public String[] prepods = new String[0];

    static public SQLiteDatabase db;

    static public SharedPreferences preferences;

    static public void getDataFromCache() {
        Context context = MainActivity.mainActivity.getBaseContext();

        preferences = context.getSharedPreferences("dateOfUpdate",
Context.MODE_PRIVATE);

        timeOfUpdate = Long.parseLong(preferences.getString("time", "0"));

        if (context.databaseList().length == 0) {
            db = context.openOrCreateDatabase("app.db", Context.MODE_PRIVATE,
null);
            db.execSQL(
                "CREATE TABLE IF NOT EXISTS data_m (number_a int,
number_of_day int, `group` text, predmet1 text, predmet2 text, audit1 text,
audit2 text, prep1 text, prep2 text)"
            );
            db.execSQL(
```



```

        "CREATE TABLE IF NOT EXISTS data_rep (number_a int,
number_of_day int, `group` text, predmet1 text, predmet2 text, audit1 text,
audit2 text, prep1 text, prep2 text)"
    );
    db.execSQL(
        "CREATE TABLE IF NOT EXISTS data_lab (number_a int,
number_of_day int, `group` text, predmet1 text, audit1 text, prep1 text)"
    );
    } else {
        db = context.openOrCreateDatabase("app.db", Context.MODE_PRIVATE,
null);
        loadCacheData();
    }

    new Thread(() -> {
        try {
            getDataFromServer();
        } catch (IOException | JSONException e) {
            e.printStackTrace();
        }
    }).start();
}

static public void loadCacheData() {
    data = new ArrayList<>();
    cachedReplacement = new ArrayList<>();
    cachedLab = new ArrayList<>();

    HashSet<String> setGroups = new HashSet<>();
    HashSet<String> setAudit = new HashSet<>();
    HashSet<String> setPrep = new HashSet<>();

    Cursor query = db.rawQuery("SELECT * FROM data_m ORDER BY
number_of_day, number_a;", null);
    while(query.moveToNext()){
        RozkladItem rozkladItem = new RozkladItem();
        rozkladItem.number = query.getInt(0);
        rozkladItem.numberOfDay = query.getInt(1);
        rozkladItem.group = query.getString(2);
        rozkladItem.predmet1 = query.getString(3);
        rozkladItem.predmet2 = query.getString(4);
        rozkladItem.audit1 = query.getString(5);
        rozkladItem.audit2 = query.getString(6);
        rozkladItem.prepod1 = query.getString(7);
        rozkladItem.prepod2 = query.getString(8);
        rozkladItem.type = "1";
        data.add(rozkladItem);

        setGroups.add(rozkladItem.group);
        if (!rozkladItem.audit1.equals("null") &&
!rozkladItem.audit1.equals("") && !rozkladItem.audit1.equals("Відсутня")) {
            setAudit.add(rozkladItem.audit1);
        }
        if (!rozkladItem.audit2.equals("null") &&
!rozkladItem.audit2.equals("") && !rozkladItem.audit2.equals("Відсутня")) {
            setAudit.add(rozkladItem.audit2);
        }
        if (!rozkladItem.prepod1.equals("null") &&
!rozkladItem.prepod1.equals("") && !rozkladItem.prepod1.equals("Відсутня")) {
            setPrep.add(rozkladItem.prepod1);
        }
    }
}

```

```

        if (!rozkladItem.prepod2.equals("null") &&
!rozkladItem.prepod2.equals("") && !rozkladItem.prepod2.equals("Відсутня")) {
            setPrep.add(rozkladItem.prepod2);
        }
    }
    query.close();

    groups = new String[setGroups.size() + 1];
    int c = 1;
    groups[0] = "";
    for(String group : setGroups) groups[c++] = group;

    audits = new String[setAudit.size() + 1];
    c = 1;
    audits[0] = "";
    for(String audit : setAudit) audits[c++] = audit;

    prepods = new String[setPrep.size() + 1];
    c = 1;
    prepods[0] = "";
    for(String prep : setPrep) prepods[c++] = prep;

    query = db.rawQuery("SELECT * FROM data_rep ORDER BY number_of_day,
number_a;", null);
    while(query.moveToNext()){
        RozkladItem rozkladItem = new RozkladItem();
        rozkladItem.number = query.getInt(0);
        rozkladItem.numberOfDay = query.getInt(1);
        rozkladItem.group = query.getString(2);
        rozkladItem.predmet1 = query.getString(3);
        rozkladItem.predmet2 = query.getString(4);
        rozkladItem.audit1 = query.getString(5);
        rozkladItem.audit2 = query.getString(6);
        rozkladItem.prepod1 = query.getString(7);
        rozkladItem.prepod2 = query.getString(8);
        rozkladItem.type = "2";
        cachedReplacement.add(rozkladItem);
    }
    query.close();

    query = db.rawQuery("SELECT * FROM data_lab ORDER BY number_of_day,
number_a;", null);
    while(query.moveToNext()){
        RozkladItem rozkladItem = new RozkladItem();
        rozkladItem.number = query.getInt(0);
        rozkladItem.numberOfDay = query.getInt(1);
        rozkladItem.group = query.getString(2);
        rozkladItem.predmet1 = query.getString(3);
        rozkladItem.predmet2 = "";
        rozkladItem.audit1 = query.getString(4);
        rozkladItem.audit2 = "";
        rozkladItem.prepod1 = query.getString(5);
        rozkladItem.prepod2 = "";
        rozkladItem.type = "3";
        cachedLab.add(rozkladItem);
    }
    query.close();
}

static public void getDataFromServer() throws IOException, JSONException
{
    ArrayList<RozkladItem> main = ApiHelper.getMainData();
}

```

```

        ArrayList<RozkladItem> rep = ApiHelper.getRepData();
        ArrayList<RozkladItem> lab = ApiHelper.getLabData();
        db.execSQL("Delete from data_m");
        db.execSQL("Delete from data_rep");
        db.execSQL("Delete from data_lab");
        for (RozkladItem item : main) {
            db.execSQL("INSERT INTO `data_m` (`number_a`, `number_of_day`,
`group`, `predmet1`, `predmet2`, `audit1`, `audit2`, `prep1`, `prep2`) \n" +
                "VALUES ('"+item.number+"', '"+item.numberOfDay+"',
 '"+item.group+"', '"+item.predmet1+"', '"+item.predmet2+"',
 '"+item.audit1+"', '"+item.audit2+"', '"+item.prepod1+"',
 '"+item.prepod2+"');");
        }
        for (RozkladItem item : rep) {
            db.execSQL("INSERT INTO `data_rep` (`number_a`, `number_of_day`,
`group`, `predmet1`, `predmet2`, `audit1`, `audit2`, `prep1`, `prep2`) \n" +
                "VALUES ('"+item.number+"', '"+item.numberOfDay+"',
 '"+item.group+"', '"+item.predmet1+"', '"+item.predmet2+"',
 '"+item.audit1+"', '"+item.audit2+"', '"+item.prepod1+"',
 '"+item.prepod2+"');");
        }
        for (RozkladItem item : lab) {
            db.execSQL("INSERT INTO `data_lab` (`number_a`, `number_of_day`,
`group`, `predmet1`, `audit1`, `prep1`) \n" +
                "VALUES ('"+item.number+"', '"+item.numberOfDay+"',
 '"+item.group+"', '"+item.predmet1+"', '"+item.audit1+"',
 '"+item.prepod1+"');");
        }
        updateTime(System.currentTimeMillis());
        loadCacheData();
    }

    static public void updateTime(long time) {
        timeOfUpdate = time;
        TextView textView =
HomeFragment.root.findViewById(R.id.time_of_update);
        textView.setText("Дані оновлено " + (new Time(time)).toString());
        preferences.edit().putString("time", String.valueOf(time)).commit();
    }

    static public void updateTimeFromCache() {
        updateTime(timeOfUpdate);
    }
}

```

Додаток Г

У даному додатку приведено код ApiHelper:

```
package com.example.mksumduschedule.helpers;

import com.example.mksumduschedule.entity.RozkladItem;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

abstract public class ApiHelper {
    static public ArrayList<RozkladItem> getMainData() throws IOException,
    JSONException {
        ArrayList<RozkladItem> result = new ArrayList<>();

        String res =
getContent("http://rozklad.mksumdu.info/api/schedule/4");// TODO after fix
api

        JSONArray jsonArray = new JSONObject(res).getJSONArray("data");

        for (int i=0; i < jsonArray.length(); i++)
        {
            try {
                JSONObject oneObject = jsonArray.getJSONObject(i);
                RozkladItem item = new RozkladItem();
                item.number = oneObject.getInt("order");
                item.numberOfDay = oneObject.getInt("dayOfWeek");
                item.group = "410"; // TODO after fix api
                item.type = "1";
                item.predmet1 = oneObject.getString("numeratorSubject");
                item.predmet2 = oneObject.getString("denominatorSubject");
                item.audit1 = oneObject.getString("numeratorAuditory");
                item.audit2 = oneObject.getString("denominatorAuditory");
                item.prepod1 = oneObject.getString("numeratorTeacher");
                item.prepod2 = oneObject.getString("denominatorTeacher");

                if (item.predmet1.equals("null")) {
                    item.predmet1 = "Відсутня";
                    item.prepod1 = "";
                    item.audit1 = "";
                }
                if (item.predmet2.equals("null")) {
                    item.predmet2 = "Відсутня";
                    item.prepod2 = "";
                    item.audit2 = "";
                }

                result.add(item);
            } catch (JSONException e) {
                // Oops
            }
        }
    }
}
```

```

        }
    }

    return result;
}

static public ArrayList<RozkladItem> getRepData() throws IOException,
JSONException {
    ArrayList<RozkladItem> result = new ArrayList<>();

    String res =
getContent("http://rozklad.mksumdu.info/api/schedule/4");// TODO after fix
api

    JSONArray jsonArray = new JSONObject(res).getJSONArray("data");

    for (int i=0; i < jsonArray.length(); i++)
    {
        try {
            JSONObject oneObject = jsonArray.getJSONObject(i);
            RozkladItem item = new RozkladItem();
            item.number = oneObject.getInt("order");
            item.numberOfDay = oneObject.getInt("dayOfWeek");
            item.group = "410"; // TODO after fix api
            item.type = "2";
            item.predmet1 = oneObject.getString("numeratorSubject");
            item.predmet2 = oneObject.getString("denominatorSubject");
            item.audit1 = oneObject.getString("numeratorAuditory");
            item.audit2 = oneObject.getString("denominatorAuditory");
            item.prepod1 = oneObject.getString("numeratorTeacher");
            item.prepod2 = oneObject.getString("denominatorTeacher");

            if (item.predmet1.equals("null")) {
                item.predmet1 = "Відсутня";
                item.prepod1 = "";
                item.audit1 = "";
            }
            if (item.predmet2.equals("null")) {
                item.predmet2 = "Відсутня";
                item.prepod2 = "";
                item.audit2 = "";
            }

            result.add(item);
        } catch (JSONException e) {
            // Oops
        }
    }

    return result;
}

static public ArrayList<RozkladItem> getLabData() throws JSONException,
IOException {
    ArrayList<RozkladItem> result = new ArrayList<>();

    String res =
getContent("http://rozklad.mksumdu.info/api/schedule/4");// TODO after fix
api

    JSONArray jsonArray = new JSONObject(res).getJSONArray("data");

```

```

for (int i=0; i < jsonArray.length(); i++)
{
    try {
        JSONObject oneObject = jsonArray.getJSONObject(i);
        RozkladItem item = new RozkladItem();
        item.number = oneObject.getInt("order");
        item.numberOfDay = oneObject.getInt("dayOfWeek");
        item.group = "410"; // TODO after fix api
        item.type = "3";
        item.predmet1 = oneObject.getString("numeratorSubject");
        item.audit1 = oneObject.getString("numeratorAuditory");
        item.prepod1 = oneObject.getString("numeratorTeacher");

        result.add(item);
    } catch (JSONException e) {
        // Oops
    }
}

return result;
}

static private String getContent(String path) throws IOException {
    BufferedReader reader=null;
    InputStream stream = null;
    HttpURLConnection connection = null;
    try {
        URL url=new URL(path);
        connection =(HttpURLConnection)url.openConnection();
        connection.setRequestMethod("GET");
        connection.setReadTimeout(10000);
        connection.connect();
        stream = connection.getInputStream();
        reader= new BufferedReader(new InputStreamReader(stream));
        StringBuilder buf=new StringBuilder();
        String line;
        while ((line=reader.readLine()) != null) {
            buf.append(line).append("\n");
        }
        return(buf.toString());
    }
    finally {
        if (reader != null) {
            reader.close();
        }
        if (stream != null) {
            stream.close();
        }
        if (connection != null) {
            connection.disconnect();
        }
    }
}
}

```

Додаток Д

У даному додатку приведено код HomeFragment:

```
package com.example.mksumduschedule.ui;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProvider;
import com.example.mksumduschedule.MainActivity;
import com.example.mksumduschedule.R;
import com.example.mksumduschedule.entity.RozkladItem;
import com.example.mksumduschedule.helpers.MainStorage;
import com.example.mksumduschedule.ui.ScheduleItemFragment;
import com.example.mksumduschedule.ui.WeekDayFragment;
import org.json.JSONException;

import java.io.IOException;
import java.util.ArrayList;

public class HomeFragment extends Fragment {
    static public ViewGroup root;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState)
    {
        root = (ViewGroup)inflater.inflate(R.layout.fragment_home, container,
false);

        MainStorage.updateTimeFromCache();

        Button button = root.findViewById(R.id.button_search);
        button.setOnClickListener(v -> onSearch());

        button = root.findViewById(R.id.update_data);
        button.setOnClickListener(v -> {
            new Thread(() -> {
                try {
                    MainStorage.getDataFromServer();
                } catch (IOException | JSONException e) {
                    e.printStackTrace();
                }
            }).start();
        });

        Spinner spinner = root.findViewById(R.id.select_group);
        ArrayAdapter<String> adapter = new
ArrayAdapter<>(MainActivity.mainActivity,
android.R.layout.simple_spinner_item, MainStorage.groups);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
);
    }
}
```

```

        spinner.setAdapter(adapter);

        spinner = root.findViewById(R.id.select_prepod);
        adapter = new ArrayAdapter<>(MainActivity.mainActivity,
        android.R.layout.simple_spinner_item, MainStorage.prepods);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
        );
        spinner.setAdapter(adapter);

        spinner = root.findViewById(R.id.select_audit);
        adapter = new ArrayAdapter<>(MainActivity.mainActivity,
        android.R.layout.simple_spinner_item, MainStorage.audits);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
        );
        spinner.setAdapter(adapter);

        return root;
    }

    private void onSearch() {
        LinearLayout layout = root.findViewById(R.id.content_main_core);
        layout.removeAllViews();

        Spinner spinner = root.findViewById(R.id.select_group);
        String filterGroup = (String) spinner.getSelectedItem();
        spinner = root.findViewById(R.id.select_audit);
        String filterAudit = (String) spinner.getSelectedItem();
        spinner = root.findViewById(R.id.select_prepod);
        String filterPrepod = (String) spinner.getSelectedItem();

        ArrayList<RozkladItem> filteredItems = new ArrayList<>();

        for (RozkladItem rozkladItem : MainStorage.cachedReplacement) {
            if (!filterGroup.equalsIgnoreCase("")) {
                if (!rozkladItem.group.equalsIgnoreCase(filterGroup)) {
                    continue;
                }
            }
            if (!filterAudit.equalsIgnoreCase("")) {
                if (!rozkladItem.audit1.equalsIgnoreCase(filterAudit) &&
                !rozkladItem.audit2.equalsIgnoreCase(filterAudit)) {
                    continue;
                }
            }
            if (!filterPrepod.equalsIgnoreCase("")) {
                if (!rozkladItem.prepod1.equalsIgnoreCase(filterPrepod) &&
                !rozkladItem.prepod2.equalsIgnoreCase(filterPrepod)) {
                    continue;
                }
            }
            filteredItems.add(rozkladItem);
        }

        int currentDay = 0;
        for (RozkladItem rozkladItem : filteredItems) {
            if (rozkladItem.numberOfDay != currentDay) {
                currentDay = rozkladItem.numberOfDay;
            }
        }

        getParentFragmentManager().beginTransaction().add(layout.getId(), new
        WeekDayFragment(currentDay), "someTag1").commit();
    }

```



```
        }
        getParentFragmentManager().beginTransaction().add(layout.getId(),
ScheduleItemFragment.newReplacement(rozkladItem), "someTag1").commit();
    }
}
```

Додаток Е

У даному додатку приведено код ScheduleItemFragment:

```
package com.example.mksumduschedule.ui;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.fragment.app.Fragment;
import com.example.mksumduschedule.R;
import com.example.mksumduschedule.entity.RozkladItem;
import org.jetbrains.annotations.NotNull;

public class ScheduleItemFragment extends Fragment {
    public static ScheduleItemFragment newMain(RozkladItem rozkladItem) {

        ScheduleItemFragment f = new ScheduleItemFragment();

        Bundle b = new Bundle();
        b.putString("item_para", String.valueOf(rozkladItem.number));
        b.putString("item_group", String.valueOf(rozkladItem.group));
        b.putString("item_audit_1", String.valueOf(rozkladItem.audit1));
        b.putString("item_audit_2", String.valueOf(rozkladItem.audit2));
        b.putString("item_lesson_1", String.valueOf(rozkladItem.predmet1));
        b.putString("item_lesson_2", String.valueOf(rozkladItem.predmet2));
        b.putString("item_prep_1", String.valueOf(rozkladItem.prepod1));
        b.putString("item_prep_2", String.valueOf(rozkladItem.prepod2));
        b.putString("item_type", "1");
        f.setArguments(b);
        return f;
    }

    public static ScheduleItemFragment newReplacement(RozkladItem
rozkladItem) {

        ScheduleItemFragment f = new ScheduleItemFragment();

        Bundle b = new Bundle();
        b.putString("item_para", String.valueOf(rozkladItem.number));
        b.putString("item_group", String.valueOf(rozkladItem.group));
        b.putString("item_audit_1", String.valueOf(rozkladItem.audit1));
        b.putString("item_audit_2", String.valueOf(rozkladItem.audit2));
        b.putString("item_lesson_1", String.valueOf(rozkladItem.predmet1));
        b.putString("item_lesson_2", String.valueOf(rozkladItem.predmet2));
        b.putString("item_prep_1", String.valueOf(rozkladItem.prepod1));
        b.putString("item_prep_2", String.valueOf(rozkladItem.prepod2));
        b.putString("item_type", "2");
        f.setArguments(b);
        return f;
    }

    public static ScheduleItemFragment newLab(RozkladItem rozkladItem) {
```

```

ScheduleItemFragment f = new ScheduleItemFragment();

Bundle b = new Bundle();
b.putString("item_para", String.valueOf(rozkladItem.number));
b.putString("item_group", String.valueOf(rozkladItem.group));
b.putString("item_audit_1", String.valueOf(rozkladItem.audit1));
b.putString("item_audit_2", String.valueOf(rozkladItem.audit2));
b.putString("item_lesson_1", String.valueOf(rozkladItem.predmet1));
b.putString("item_lesson_2", String.valueOf(rozkladItem.predmet2));
b.putString("item_prep_1", String.valueOf(rozkladItem.prepod1));
b.putString("item_prep_2", String.valueOf(rozkladItem.prepod2));
b.putString("item_type", "3");
f.setArguments(b);
return f;
}

@SuppressLint("SetTextI18n")
@Override
public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {

    View v;
    if (getArguments().getString("item_type").equals("3")) {
        v = inflater.inflate(R.layout.fragment_schedule_item_lab,
container, false);
    } else {
        v = inflater.inflate(R.layout.fragment_schedule_item, container,
false);
    }

    ((TextView)
v.findViewById(R.id.item_para)).setText(getArguments().getString("item_para"
));
    ((TextView)
v.findViewById(R.id.item_group)).setText(getArguments().getString("item_group
"));
    ((TextView)
v.findViewById(R.id.item_audit_1)).setText(getArguments().getString("item_aud
it_1"));
    ((TextView)
v.findViewById(R.id.item_lesson_1)).setText(getArguments().getString("item_le
sson_1"));
    ((TextView)
v.findViewById(R.id.item_prep_1)).setText(getArguments().getString("item_prep
_1"));

    if (!getArguments().getString("item_type").equals("3")) {
        ((TextView)
v.findViewById(R.id.item_audit_2)).setText(getArguments().getString("item_aud
it_2"));
        ((TextView)
v.findViewById(R.id.item_lesson_2)).setText(getArguments().getString("item_le
sson_2"));
        ((TextView)
v.findViewById(R.id.item_prep_2)).setText(getArguments().getString("item_prep
_2"));

        if (getArguments().getString("item_type").equals("2")) {
            ((TextView)
v.findViewById(R.id.item_chis)).setText("Замість");
            ((TextView) v.findViewById(R.id.item_znam)).setText("Буде");
        } else {

```

```
        ((TextView)
v.findViewById(R.id.item_chis)).setText("Чисельник");
        ((TextView)
v.findViewById(R.id.item_znam)).setText("Знаменник");
    }
}
return v;
}
}
```

Додаток Є

У даному додатку приведено код DashboardFragment:

```
package com.example.mksumduschedule.ui;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.*;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.ViewModelProvider;
import com.example.mksumduschedule.MainActivity;
import com.example.mksumduschedule.R;
import com.example.mksumduschedule.entity.RozkladItem;
import com.example.mksumduschedule.helpers.MainStorage;
import com.example.mksumduschedule.ui.ScheduleItemFragment;
import com.example.mksumduschedule.ui.WeekDayFragment;
import org.jetbrains.annotations.NotNull;

public class DashboardFragment extends Fragment {
    private View root;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState)
    {
        root = inflater.inflate(R.layout.fragment_dashboard, container,
            false);
        createTestList();

        return root;
    }

    private void createTestList() {
        int currentDay = 0;
        LinearLayout layout = root.findViewById(R.id.content_replacement);
        for (RozkladItem rozkladItem : MainStorage.cachedReplacement) {
            if (rozkladItem.numberOfDay != currentDay) {
                currentDay = rozkladItem.numberOfDay;

                getParentFragmentManager().beginTransaction().add(layout.getId(), new
                    WeekDayFragment(currentDay), "someTag1").commit();
            }
            getParentFragmentManager().beginTransaction().add(layout.getId(),
                ScheduleItemFragment.newReplacement(rozkladItem), "someTag1").commit();
        }
    }
}
```

Додаток Ж

У даному додатку приведено код NotificationFragment:

```
package com.example.mksumduschedule.ui;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProvider;
import com.example.mksumduschedule.R;
import com.example.mksumduschedule.entity.RozkladItem;
import com.example.mksumduschedule.helpers.MainStorage;
import com.example.mksumduschedule.ui.ScheduleItemFragment;
import com.example.mksumduschedule.ui.WeekDayFragment;

public class NotificationsFragment extends Fragment {
    private View root;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState)
    {
        root = inflater.inflate(R.layout.fragment_notifications, container,
false);
        createTestList();
        return root;
    }

    private void createTestList() {
        int currentDay = 0;
        LinearLayout layout = root.findViewById(R.id.content_lab);
        for (RozkladItem rozkladItem : MainStorage.cachedLab) {
            if (rozkladItem.numberOfDay != currentDay) {
                currentDay = rozkladItem.numberOfDay;

                getParentFragmentManager().beginTransaction().add(layout.getId(), new
WeekDayFragment(currentDay), "someTag1").commit();
            }
            getParentFragmentManager().beginTransaction().add(layout.getId(),
ScheduleItemFragment.newLab(rozkladItem), "someTag1").commit();
        }
    }
}
```

Додаток Ж

У даному додатку приведено код `WeekDayFragment`:

```
package com.example.mksumduschedule.ui;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.fragment.app.Fragment;
import com.example.mksumduschedule.R;
import org.jetbrains.annotations.NotNull;

public class WeekDayFragment extends Fragment {
    private final int dayNumber;

    public WeekDayFragment(int dayNumber) {
        super();
        this.dayNumber = dayNumber;
    }

    @Override
    public View onCreateView(@NotNull LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {

        View v;
        v = inflater.inflate(R.layout.fragment_schedule_week_day, container,
false);
        String dayName;
        switch (dayNumber) {
            case 1: {dayName = "Понеділок";break;}
            case 2: {dayName = "Вівторок";break;}
            case 3: {dayName = "Середа";break;}
            case 4: {dayName = "Чертерг";break;}
            case 5: {dayName = "П'ятниця";break;}
            default: {dayName = "";}
        }
        ((TextView) v.findViewById(R.id.day_name)).setText(dayName);
        return v;
    }
}
```