

ТЕХНИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ UNIVERSITY HELPER

В данной статье представлено описание процесса разработки мобильного приложения для университетов University Helper. Разработка начиналась с установки программного обеспечения необходимого для написания кода и его отладки: Android Studio (IDE-среда), Genymotion (эмулятор для тестирования приложения на различных устройствах).

Приложение работает с web-сервером, где в базе данных хранится вся необходимая информация: пользователи, списки групп и студентов, предметы, информация о посещаемости. Обмен данными происходит с помощью Api-запросов на тестовый адрес http://tatyanaast.com/uh_api/api.php. После успешного обращения к серверу приложение получает ответ в формате json, который оно разбирает на компоненты с помощью библиотеки Gson, отображает пользователю данные, а также сохраняет их в базе данных мобильного устройства. Описанные выше действия представлены на диаграмме (рисунок 1).

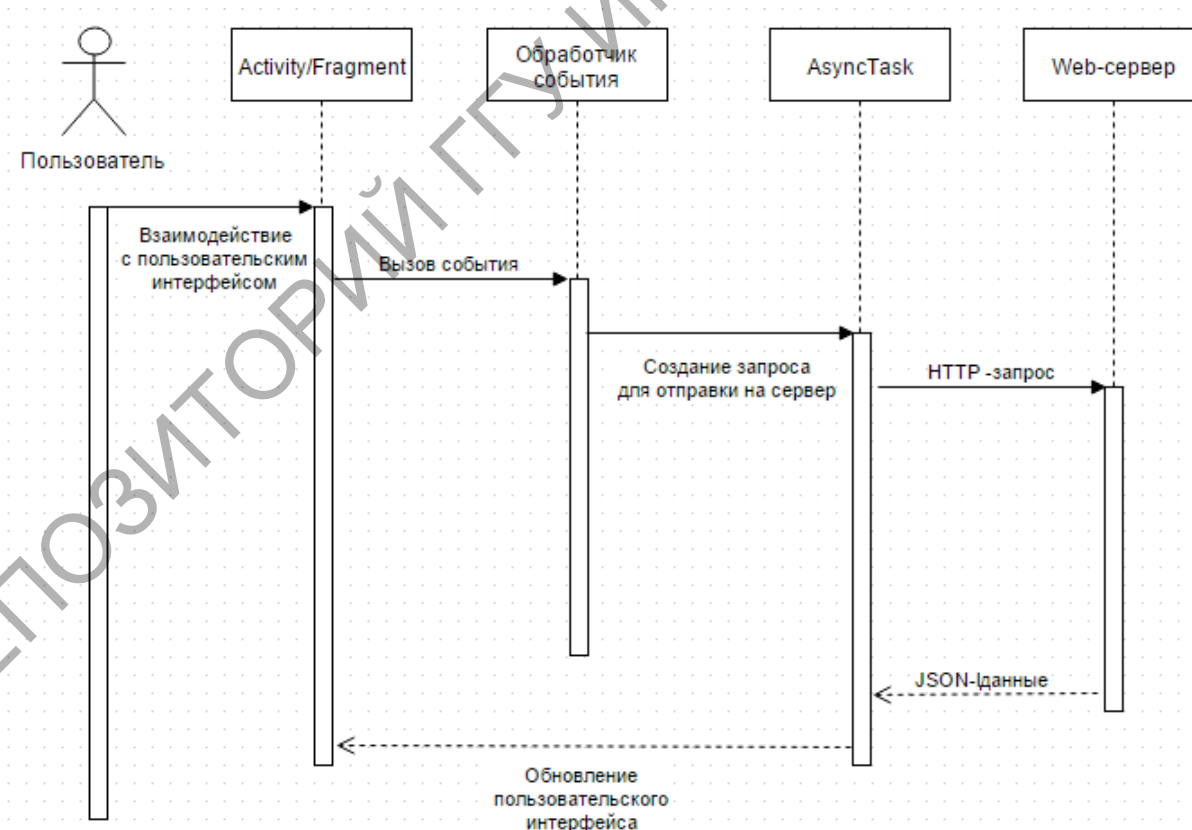


Рисунок 1 – Диаграмма последовательностей

Создана локальная база данных на устройстве для работы с приложением в offline-режиме, схема которой представлена на рисунке 2. Использовалась SQLite Database. Она является реляционной, открытой и легковесной.

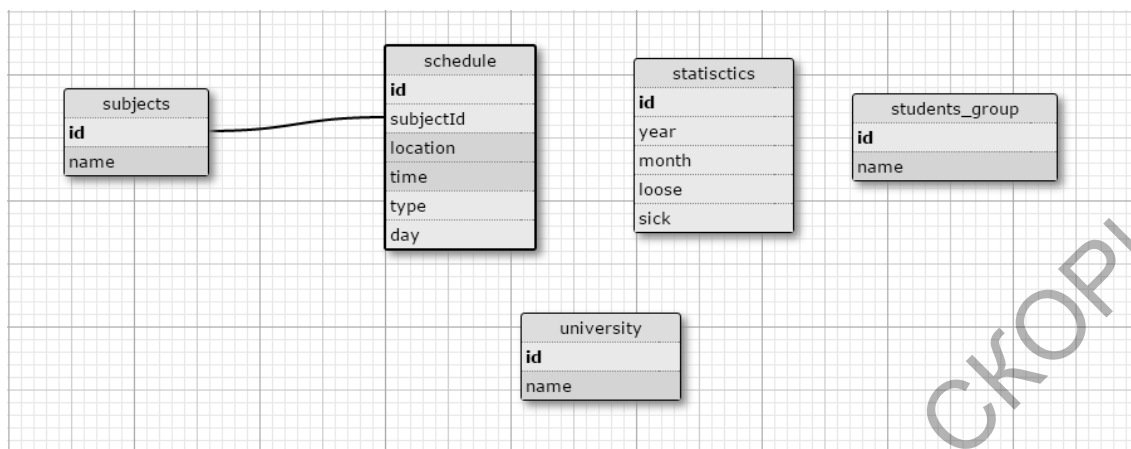


Рисунок 2 – Схема локальной базы данных на мобильном устройстве

База данных состоит из пяти таблиц:

- Таблица subjects хранит список предметов, которые используются таблицей с расписанием.
- Таблица schedule хранит расписание. Она представлена следующими полями: предмет, тип (над чертой, под чертой), время проведения, аудитория, а также день недели.
- Таблица statistics содержит статистику посещаемости студента: количество пропусков по уважительной и неуважительной причине.
- Таблица students_group хранит список студентов в группе.
- Таблица university содержит список университетов, с которыми может работать администратор.

После разработки базы данных и создания соответствующих java-классов для работы с ней, началась разработка графического интерфейса и логики приложения.

Для отображения данных в удобном для пользователя виде использовались xml-файлы. С их помощью задавалось расположение элементов на экране, цвета, стили. Фрагмент xml-кода представлен ниже.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/tabs">
    <ImageView
        android:id="@+id/tab_image"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_margin="5dp"
        android:layout_centerInParent="true" />
</RelativeLayout>
    
```

Экраны приложения представлены java-классами-потомками Android SDK класса Activity. В нем написаны обработчики событий, вызванных действиями пользователя, обращение к вспомогательным классам, и подключение xml-файла для отображения графического интерфейса. У потомка Activity имеется цикл жизни. Главный из них onCreate, который вызывается первым. Код переопределенного метода одного из экранов представлен ниже.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login);
    bar = getActionBar();
    bar.hide();

    loginButton = (Button) findViewById(R.id.login_button);
    webButton = (Button) findViewById(R.id.website_button);
    editLogin = (EditText) findViewById(R.id.email_edit);
    editPass = (EditText) findViewById(R.id.pass_edit);
    webButton.setOnClickListener(this);
    loginButton.setOnClickListener(this);
}
```

Разработаны вспомогательные классы для работы с локальной базой данных, для взаимодействия с web-сервером через http-протокол. Благодаря использованию библиотеки GSON сериализация и десериализация json-строк упростилась во много раз. Пример десериализации представлен ниже.

```
@Override
public Object getResponse(String json) {
    Gson gson = new Gson();
    Schedule shedule = gson.fromJson(json, Schedule.class);
    return shedule;
}
```

После написания кода, мобильное приложение University Helper было протестировано на самых популярных устройствах под управлением операционной системы Android с помощью эмулятора Genymotion.

Т.С. Герасимович (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Е.Е. Пугачёва**, ассистент

АВТОМАТИЗАЦИЯ ПРИЕМА И УЧЕТА СЖИЖЕННОГО ГАЗА НАСЕЛЕНИЮ

В разработке любого приложения, предназначенного для хранения информации, большую роль играет использование базы данных в качестве