Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

Факультет: информационных технологий и управления

Кафедра: систем автоматизированного проектирования и управления

Направление подготовки:09.03.03 Прикладная информатика в химии

Форма обучения: очная

Уровень подготовки: бакалавр

Учебная дисциплина: разработка программного обеспечения для мобильных систем

Группа: 405

**ОТЧЕТ**

**О ЛАБОРАТОРНОЙ РАБОТЕ№3**

ТЕМА: Управление данными локального хранилища мобильного устройства на основе интегрированной СУБД SQLite и графического компонента RecyclerView

Вариант № 10

Преподаватели Д.Н. Петров

К.А. Захаренко

Исполнители М. А. Шахов

К. И. Емельянова

Санкт-Петербург

2023

**Задание:**

Задание на лабораторную работу представлено в таблице 1.

Таблица 1 – Задание на лабораторную работу

|  |  |
| --- | --- |
| № | Наименование |
| 10 | Карточки преподавателей кафедры с указанием ФИО, должности, степени, звания и фото преподавателя. Фото доступно с официального сайта или Медиапортала по прямой ссылке при загрузке его в виджет ImageView. Сортировку установить по званию и ФИО преподавателей. |

**Этапы выполнения**

1) Изучение специфики языка запросов и типов данных SQLite, механизма подключения и использования интегрированной СУБД SQLite для управления данными в памяти мобильного устройства. Ознакомление с принципами настройки адаптера данных, связывания данных, особенностями доступа к данным с использованием объекта cursor.

2) Разработка мобильного приложения:

- создание и настройка нового проекта приложения;

- работа с ресурсами приложения: строковыми константами, цветами, шрифтами, стилями, пиктограммами, установка иконки приложения;

- разметка макета графического интерфейса для двух активностей и элемента списка;

- создание управляющей логики (класса для работы с SQLite, адаптера данных, кода главной активности, кода активности для создания/редактирования элемента списка);

- тестирование мобильного приложения.

**Описание выполнения лабораторной работы**

**1) Изучение специфики языка запросов и типов данных SQLite**

SQLite – набор интегрированных библиотек для работы со структурированным локальным хранилищем данных. SQLite называют системой управления базами данных (СУБД) для локальных баз данных. SQLite входит в Android SDK и может быть использована для организации локального хранилища данных. Как СУБД SQLite поддерживает основные конструкции языка запросов SQL. Но «чистым» SQL разработчики в Android Studio пользуются редко. Для создания и выполнения SQL-запросов используется класс-помощник SQLiteOpenHelper или наследованный от него класс.

SQLite поддерживает языковые группы DDL, DQL, DML.

Пример DDL (создание таблицы): CREATE TABLE tblNotes (id INTEGER PRIMARY KEY AUTOINCREMENT, note\_title TEXT, note\_text TEXT, note\_prior INTEGER);

Пример DQL (оператор выборки данных): SELECT id, note\_title, note\_text, note\_prior FROM tblNotes ORDER BY note\_prior;

Пример DML (оператор удаления): DELETE FROM tblNotes WHERE id=1;

Как видно из примеров выше, SQL, используемый для SQLite имеет сильное сходство с SQL, используемым для серверных СУБД, таких, как MS SQL Server, MySQL, PostgreSQL.

Основные типы данных SQLite:

INTEGER и NUMERIC – для целых чисел;

REAL – для десятичных чисел;

TEXT – для текстовых данных;

NONE – не определенный тип данных.

Для сопряжения внутренних списков данных с данными из БД, а также этих списков с виджетами для визуализации данных используют адаптеры.

Например, виджет RecyclerView mainList, как и ListView, имеют собственный адаптер, который можно переназначать на свой (кастомный): mainList.setAdapter(customAdapter). customAdapter настраивается таким образом, чтобы виджеты элемента списка mainList были сопряжены прямо или косвенно с каким-либо внутренним списком ArrayList<string>.

Связывание (bind) осуществляется для каждого элемента внутреннего списка и элемента виджета через «держатель» представления (или макета) элемента виджета, например:

holder.note\_id\_txt.setText(String.valueOf(note\_id.get(position))), где position – позиция элемента внутреннего списка. Получается, что адаптер автоматически создает ровно столько элементов в виджете списка, сколько элементов во внутреннем списке.

Создание «держателя» представления выполняется в методе:

public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)

Связывание данных внутренних списков с виджетами, а также назначение действий (обработчиков) для виджетов (например, нажатие на виджет) происходит в методе:

public void onBindViewHolder(@NonNull final MyViewHolder holder, final int position)

Для загрузки данных из БД SQLite в Android-проектах используется класс Cursor.

В созданный объект data класса Cursor можно вернуть массив строк с нумерованными колонками (полями), в котором колонки нумеруются начиная с нулевой:

Cursor data = DBH.loadData(), где loadData – метод, возвращающий структуру данных, полученную вследствие выполнения запроса SELECT: cursor = db.rawQuery(Query,null). Query – текст SQL-запроса на выборку данных.

* data.getCount() – определяет количество записей в структуре;
* data.moveToNext() – переход к следующей записи структуры;
* data.getString(0) – получение значения первой колонки текущей строки;
* data.getString(1) – получение значения второй колонки текущей строки и т.д.

**2) Разработка мобильного приложения**

**2.1 Создание и настройка нового проекта приложения**

В среде разработки мобильных приложений Android Studio Giraffe 17.0.6 для Android был создан проект Lab3. На рисунках 1-4 представлен процесс создания проекта.

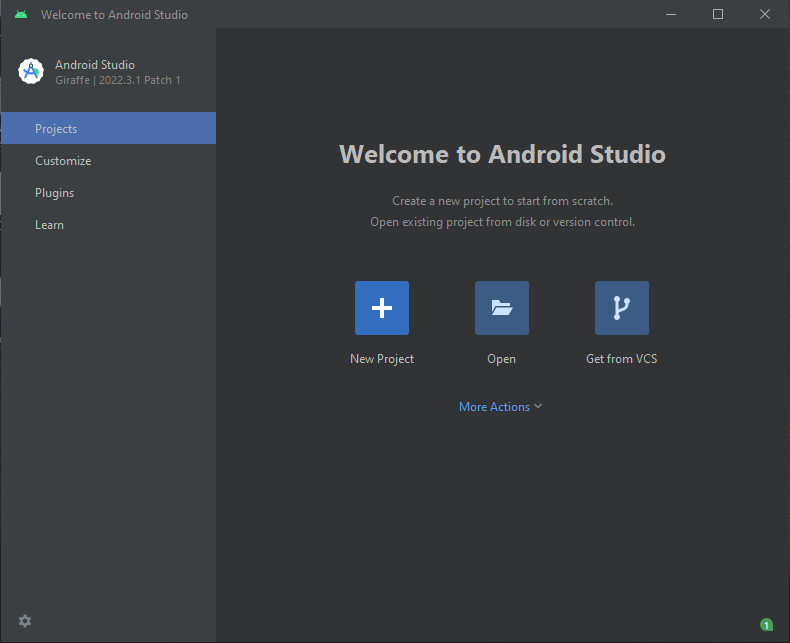


Рисунок 1 – Создание нового проекта

Создать новый проект также возможно из рабочего интерфейса по команде File->New->New Project (рисунок 2)

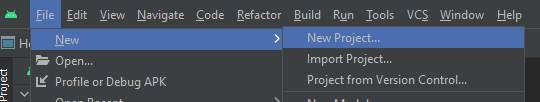


Рисунок 2 – Создание нового проекта

На появившейся форме выбран тип целевого мобильного устройства (Phone and Tablet) и шаблон приложения (Empty Activity до версии Android Studio Giraffe, а начиная с Giraffe, Empty Views Activity, т.к. требуется поддержка языка Java) (рисунок 3).

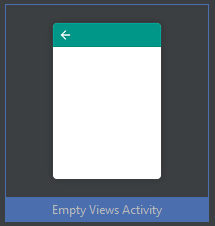
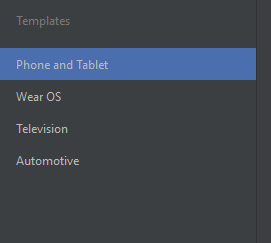


Рисунок 3 – Выбор типа целевого устройства и шаблона приложения

Далее указано наименование приложения, пакета, выбрана директория для сохранения файлов проекта и выбрана минимальную версию SDK, от которой зависит поддержка приложения операционными системами (рисунок 4).

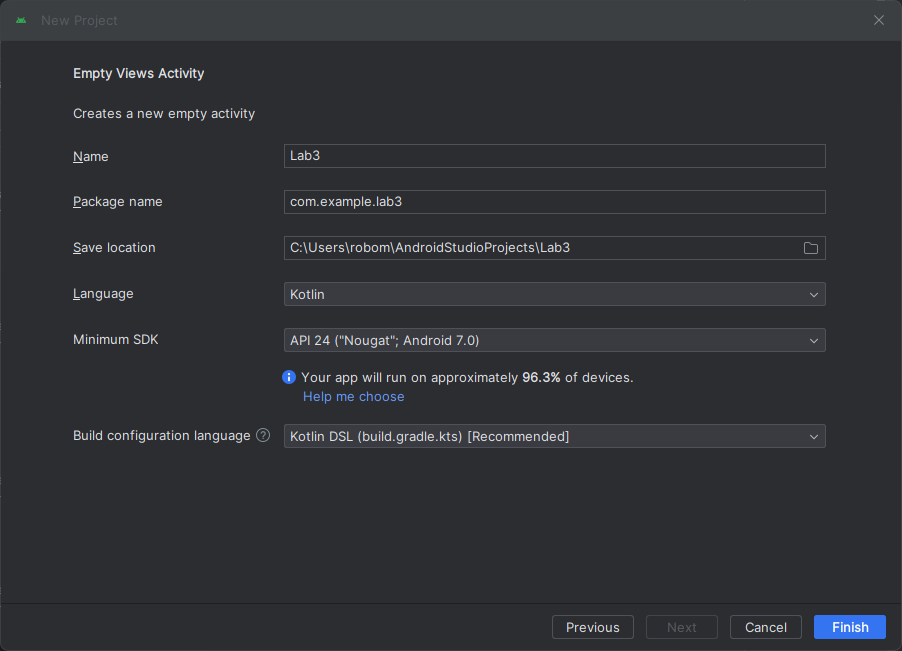


Рисунок 4 – Настройка нового приложения

Минимальная версия SDK – 24, минимальная версия ОС Android – 7.

**2.2 Разметка макета графического интерфейса главной активности**

**Структура проекта**

В таблице 2 представлена структура мобильного приложения.

Таблица 2 – Структура проекта мобильного приложения

|  |  |  |
| --- | --- | --- |
| Каталог | Файлы | Описание |
| manifests | AndroidManifest.xml | Основная информация о программе |
| java/com.lab3 | MainActivity.java | Основной управляющий код приложения |
| CustomAdapter.java | Управляющий код взаимодействия с записями в главной активности приложения |
| DBHelper.java | Управляющий код взаимодействия с базой данных |
| AddEditActivity.java | Управляющий код добавления, редактирования и удаления записей |
| res/drawable | add.xml | Логотип кнопки добавления записи |
| icon.xml | Логотип приложения |
| del.xml | Логотип кнопки удаления записи |
| delall.xml | Логотип кнопки удаления всех записей |
| circle.xml | Подложка кнопок добавления и удаления всех записей |
| default\_photo.jpg | Стандартная картинка для отображения фотографии преподавателя |
| res/layout | activity\_main.xml | Главная активность приложения |
| activity\_add\_edit.xml | Активность добавления, редактирования и удаления записи |
| my\_row.xml | Макет представления записи в главной активности |
| res/values | colors.xml | Используемые цвета в приложении |
| strings.xml | Строковые константы приложения |
| res/values/themes | themes.xml | Файл светлой темы приложения |
| themes.xml (night) | Файл темной темы приложения |
| res/xml | backup\_rules.xml | Настройка резервной копии приложения |
| data\_extraction\_rules.xml | Настройка извлечения данных |

При разработке приложения для запуска и отладки было использовано физическое устройство – мобильный телефон realme 10. Минимальная версия OC Android – 7, целевая – 13. Минимальная версия SDK – 24, целевая – 31.

Листинг файла манифеста проекта (файл AndroidManifest.xml):

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.INTERNET" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@drawable/icon"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Teachers"  
 tools:targetApi="31">  
 <activity  
 android:name=".AddEditActivity"  
 android:exported="false">  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 </application>  
</manifest>

Листинг файла строковых констант strings.xml:

<resources>  
 <string name="app\_name">Лабораторная работа №3</string>  
 <string name="main\_label">Реестр преподавателей</string>  
 <string name="c\_id">id</string>  
 <string name="c\_degree">degree</string>  
 <string name="c\_name">name</string>  
 <string name="c\_job\_title">job\_title</string>  
 <string name="c\_rank">rank</string>  
 <string name="c\_photo">photo</string>  
 <string name="rank">Звание</string>  
 <string name="name">ФИО</string>  
 <string name="job\_title">Должность</string>  
 <string name="degree">Степень</string>  
 <string name="photo">Ссылка на фото преподавателя</string>  
 <string name="adding">Добавление преподавателя</string>  
 <string name="editing">Редактирование преподавателя</string>  
 <string name="delete">Удалить</string>  
 <string name="deleting">Удаление</string>  
 <string name="save">Сохранить</string>  
 <string name="delete\_teacher">Удалить преподавателя?</string>  
 <string name="delete\_all\_teachers">Удалить всех преподавателей?</string>  
 <string name="yes">Да</string>  
 <string name="no">Нет</string>  
 <string name="no\_teachers">Преподавателей нет</string>  
 <string name="success\_add">Новый преподаватель успешно добавлен</string>  
 <string name="success\_edit">Преподаватель успешно обновлен</string>  
 <string name="success\_delete">Преподаватель успешно удален</string>  
 <string name="success\_delete\_all">Все преподаватели успешно удалены</string>  
 <string name="error\_add">Ошибка при добавлении нового преподавателя</string>  
 <string name="error\_edit">Ошибка обновления</string>  
 <string name="error\_delete">Ошибка удаления</string>  
 <string name="error">Ошибка</string>  
 <string name="ok">Ok</string>  
 <string name="cancel">Отмена</string>  
 <string name="errormessage1">Заполните поля</string>  
 <string name="row\_teacher">Преподаватель:</string>  
 <string name="row\_name">ФИО:</string>  
 <string name="row\_job\_title">Должность:</string>  
 <string name="row\_rank">Звание:</string>  
 <string name="row\_degree">Степень:</string>  
 <string name="row\_photo">Фото:</string>  
 <string name="\_jpg">.jpg</string>  
</resources>

Листинг файла цветовых констант colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="dark\_purple">#5e239d</color>  
 <color name="purple">#7e2ed3</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
</resources>

Используемые изображения:

Логотипы приложения и оформления кнопок размещены в res/drawable (таблица 3)

Таблица 3 – Логотипы для оформления кнопок в приложении

|  |  |
| --- | --- |
| Код разметки | Графический вид |
| add.xml | |
| <vector  android:height="48dp"  android:tint="?attr/colorOnSecondary"  android:viewportHeight="24"  android:viewportWidth="24"  android:width="48dp"  xmlns:android="http://schemas.android.com/apk/res/android">  <path  android:fillColor="?attr/colorOnSecondary"  android:pathData="M19,13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z" />  </vector> |  |
| icon.xml | |
| <vector android:height="48dp" android:tint="@color/purple" android:viewportHeight="24" android:viewportWidth="24" android:width="48dp" xmlns:android="http://schemas.android.com/apk/res/android"> <path android:fillColor="@color/purple" android:pathData="M12,12.75c1.63,0 3.07,0.39 4.24,0.9c1.08,0.48 1.76,1.56 1.76,2.73L18,18H6l0,-1.61c0,-1.18 0.68,-2.26 1.76,-2.73C8.93,13.14 10.37,12.75 12,12.75zM4,13c1.1,0 2,-0.9 2,-2c0,-1.1 -0.9,-2 -2,-2s-2,0.9 -2,2C2,12.1 2.9,13 4,13zM5.13,14.1C4.76,14.04 4.39,14 4,14c-0.99,0 -1.93,0.21 -2.78,0.58C0.48,14.9 0,15.62 0,16.43V18l4.5,0v-1.61C4.5,15.56 4.73,14.78 5.13,14.1zM20,13c1.1,0 2,-0.9 2,-2c0,-1.1 -0.9,-2 -2,-2s-2,0.9 -2,2C18,12.1 18.9,13 20,13zM24,16.43c0,-0.81 -0.48,- |  |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | Код разметки | Графический вид |   icon.xml | |
| 1.53 -1.22,-1.85C21.93,14.21 20.99,14 20,14c-0.39,0 -0.76,0.04 -1.13,0.1c0.4,0.68 0.63,1.46 0.63,2.29V18l4.5,0V16.43zM12,6c1.66,0 3,1.34 3,3c0,1.66 -1.34,3 -3,3s-3,-1.34 -3,-3C9,7.34 10.34,6 12,6z"/> </vector> |  |
| circle.xml | |
| <vector  android:height="48dp"  android:tint="?attr/colorPrimary"  android:viewportHeight="24"  android:viewportWidth="24"  android:width="48dp"  xmlns:android="http://schemas.android.com/apk/res/android">  <path  android:fillColor="?attr/colorPrimary"  android:pathData="M12,2C6.47,2 2,6.47 2,12s4.47,10 10,10 10,-4.47 10,-10S17.53,2 12,2z"/> |  |
| del.xml | |
| <vector  android:autoMirrored="true"  android:height="30dp"  android:tint="?attr/colorOnPrimary"  android:viewportHeight="24"  android:viewportWidth="24"  android:width="30dp"  xmlns:android="http://schemas.android.com/apk/res/android">  <path  android:fillColor="?attr/colorOnPrimary"  android:pathData="M19,6.41L17.59,5 12,10.59 6.41,5 5,6.41 10.59,12 5,17.59 6.41,19 12,13.41 17.59,19 19,17.59 13.41,12z"/>  </vector> |  |
| delall.xml | |
| <vector  android:height="48dp"  android:tint="?attr/colorOnSecondary"  android:viewportHeight="24"  android:viewportWidth="24"  android:width="48dp"  xmlns:android="http://schemas.android.com/apk/res/android">  <path  android:fillColor="?attr/colorOnSecondary"  android:pathData="M6,19c0,1.1 0.9,2 2,2h8c1.1,0 2,-0.9 2,-2V7H6v12zM19,4h-3.5l-1,-1h-5l-1,1H5v2h14V4z"/>  </vector> |  |

Темы оформления:

Листинг файла светлой темы (файл themes.xml):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <style name="Theme.Teachers" parent="Theme.AppCompat.DayNight.DarkActionBar">  
 <item name="colorPrimary">@color/purple</item>  
 <item name="colorOnPrimary">@color/black</item>  
 <item name="colorSecondary">@color/purple</item>  
 <item name="colorOnSecondary">@color/white</item>  
 <item name="android:statusBarColor">@color/purple</item>  
 <item name="android:textColor">@color/black</item>  
 </style>  
</resources>

Листинг файла темной темы (themes.xml(night)):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <style name="Theme.Teachers" parent="Theme.AppCompat.DayNight.DarkActionBar">  
 <item name="colorPrimary">@color/dark\_purple</item>  
 <item name="colorOnPrimary">@color/white</item>  
 <item name="colorSecondary">@color/dark\_purple</item>  
 <item name="colorOnSecondary">@color/white</item>  
 <item name="android:statusBarColor">@color/dark\_purple</item>  
 <item name="android:textColor">@color/white</item>  
 </style>  
</resources>

**2.3 Разметка макетов графических интерфейсов**

На рисунке 5 представлена разработанная структура макета главной активности приложения в режиме дизайна.

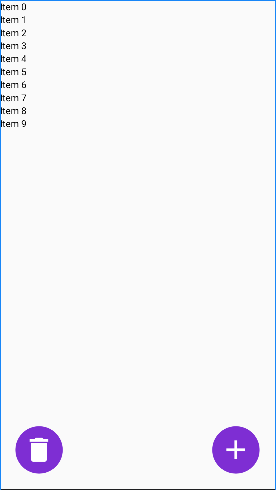


Рисунок 5 – Структура макета activity\_main

На макет графического интерфейса были добавлены следующие графические компоненты:

* recyclerView – прокручиваемый список (MainList);
* button – кнопка (AddBtn, DelAllBtn).

Листинг файла разметки макета экрана главной активности (файл activity\_main.xml):

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/MainList"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Button  
 android:id="@+id/AddBtn"  
 android:layout\_width="85dp"  
 android:layout\_height="85dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_margin="16dp"  
 android:background="@drawable/circle"  
 android:backgroundTint="?attr/colorSecondary"  
 android:clickable="true"  
 android:focusable="true"  
 android:foreground="@drawable/add"  
 android:foregroundGravity="center"  
 android:foregroundTint="?attr/colorOnSecondary"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:maxImageSize="70dp"  
 tools:ignore="SpeakableTextPresentCheck" />  
  
 <Button  
 android:id="@+id/DelAllBtn"  
 android:layout\_width="85dp"  
 android:layout\_height="85dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_margin="16dp"  
 android:layout\_marginStart="16dp"  
 android:background="@drawable/circle"  
 android:backgroundTint="?attr/colorSecondary"  
 android:clickable="true"  
 android:focusable="true"  
 android:foreground="@drawable/delall"  
 android:foregroundGravity="center"  
 android:foregroundTint="?attr/colorOnSecondary"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:maxImageSize="70dp"  
 tools:ignore="SpeakableTextPresentCheck" />  
</androidx.constraintlayout.widget.ConstraintLayout>

На рисунке 6 представлена разработанная структура макета активности добавления, редактирования и удаления записи в режиме дизайна.

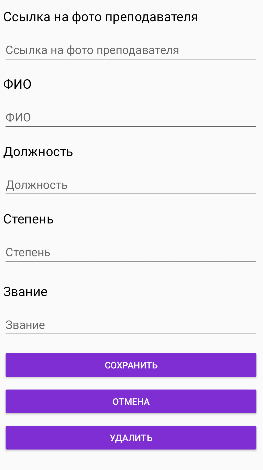


Рисунок 6 – Структура макета add\_edit\_activity

На макет графического интерфейса были добавлены следующие графические компоненты:

* scrollView – вертикальная полоса прокрутки (scrollView);
* linearLayout – выравнивает все дочерние объекты в одном направлении – вертикально или горизонтально (linearLayout);
* editText – предназначен для ввода текста пользователем (rank, name, photo, jobTitle, degree);
* textView – предназначен для отображения текста без возможности редактирования его пользователем (title\_rank, title\_name, title\_photo, title\_job\_title, title\_degree);
* button – кнопка (saveBtn, delBtn, cancelBtn).

Листинг файла разметки макета экрана активности для добавления, редактирования и удаления записи (файл activity\_add\_edit.xml):

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".AddEditActivity">  
  
 <ScrollView  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <EditText  
 android:id="@+id/rank"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:hint="@string/rank"  
 android:inputType="textMultiLine"  
 app:layout\_constraintBottom\_toTopOf="@id/saveBtn"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/title\_rank" />  
  
 <EditText  
 android:id="@+id/name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:hint="@string/name"  
 android:inputType="textMultiLine"  
 app:layout\_constraintBottom\_toTopOf="@id/title\_job\_title"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/title\_name" />  
  
 <EditText  
 android:id="@+id/photo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:hint="@string/photo"  
 android:inputType="textMultiLine"  
 app:layout\_constraintBottom\_toTopOf="@+id/title\_name"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/title\_photo" />  
  
 <EditText  
 android:id="@+id/jobTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:hint="@string/job\_title"  
 android:inputType="textMultiLine"  
 app:layout\_constraintBottom\_toTopOf="@id/title\_degree"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/title\_job\_title" />  
  
 <Button  
 android:id="@+id/saveBtn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:backgroundTint="?attr/colorSecondary"  
 android:onClick="saveBtnClick"  
 android:text="@string/save"  
 android:textColor="?attr/colorOnSecondary"  
 app:layout\_constraintBottom\_toTopOf="@id/cancelBtn"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/rank" />  
  
 <EditText  
 android:id="@+id/degree"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="16dp"  
 android:hint="@string/degree"  
 android:inputType="textMultiLine"  
 app:layout\_constraintBottom\_toTopOf="@+id/title\_rank"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/title\_degree" />  
  
 <Button  
 android:id="@+id/cancelBtn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:backgroundTint="?attr/colorSecondary"  
 android:onClick="delBtnClick"  
 android:text="@string/cancel"  
 android:textColor="?attr/colorOnSecondary"  
 app:layout\_constraintBottom\_toTopOf="@id/delBtn"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/saveBtn" />  
  
 <Button  
 android:id="@+id/delBtn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:backgroundTint="?attr/colorSecondary"  
 android:onClick="delBtnClick"  
 android:text="@string/delete"  
 android:textColor="?attr/colorOnSecondary"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/cancelBtn" />  
  
 <TextView  
 android:id="@+id/title\_rank"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/rank"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/rank"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/degree"  
 tools:text="@string/rank" />  
  
 <TextView  
 android:id="@+id/title\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/name"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@id/name"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/photo" />  
  
 <TextView  
 android:id="@+id/title\_photo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/photo"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/photo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="parent" />  
  
 <TextView  
 android:id="@+id/title\_job\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/job\_title"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@id/jobTitle"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/name" />  
  
 <TextView  
 android:id="@+id/title\_degree"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/degree"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@id/degree"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/jobTitle" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

На рисунке 7 представлена разработанная структура макета представления записи в главной активности в режиме дизайна.

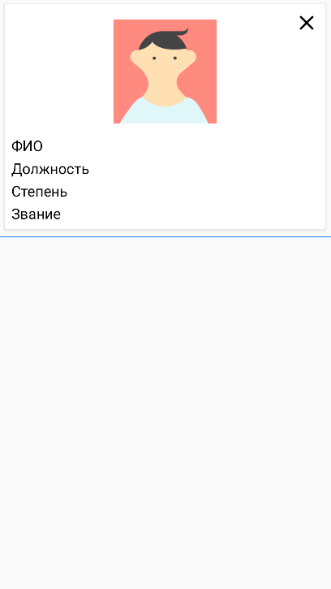


Рисунок 7 – Структура макета my\_row

На макет графического интерфейса были добавлены следующие графические компоненты:

* textView – предназначен для отображения текста без возможности редактирования его пользователем (jobTitle, name, degree, rank);
* imageView – предназначен для отображения изображений (imageViewPhoto);
* imageButton – предназначен для отображения изображений (delBtnRV).

Листинг файла разметки макета представления записи в главной активности (файл my\_row.xml):

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/mainLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/name"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="2dp"  
 android:text="@string/name"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/jobTitle"  
 app:layout\_constraintEnd\_toStartOf="@id/delBtnRV"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageViewPhoto" />  
  
 <TextView  
 android:id="@+id/jobTitle"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="2dp"  
 android:text="@string/job\_title"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/degree"  
 app:layout\_constraintEnd\_toStartOf="@id/delBtnRV"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/name" />  
  
 <TextView  
 android:id="@+id/degree"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="2dp"  
 android:text="@string/degree"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/rank"  
 app:layout\_constraintEnd\_toStartOf="@id/delBtnRV"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/jobTitle"  
 app:layout\_constraintVertical\_bias="0.1" />  
  
 <TextView  
 android:id="@+id/rank"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/rank"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@id/delBtnRV"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/degree" />  
  
 <ImageButton  
 android:id="@+id/delBtnRV"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:background="@null"  
 android:contentDescription="@string/delete"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@drawable/del"  
 tools:ignore="SpeakableTextPresentCheck,TouchTargetSizeCheck" />  
  
 <ImageView  
 android:id="@+id/imageViewPhoto"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="150dp"  
 android:layout\_marginTop="8dp"  
 android:contentDescription="@string/c\_photo"  
 app:layout\_constraintBottom\_toTopOf="@+id/name"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="ImageContrastCheck"  
 tools:srcCompat="@tools:sample/avatars" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </androidx.cardview.widget.CardView>  
</LinearLayout>

**2.4 Создание управляющей логики**

Листинг кода MainActivity.java:

package com.lab3;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.Toast;  
import androidx.activity.result.ActivityResultLauncher;  
import androidx.activity.result.contract.ActivityResultContracts;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
import java.util.Objects;  
  
public class MainActivity extends AppCompatActivity {  
 RecyclerView mainList;  
 Button AddBtn, DelAllBtn;  
 DBHelper DBH;  
 ArrayList<String> id, name, job\_title, degree, rank, photo;  
 CustomAdapter customAdapter;  
 ActivityResultLauncher<Intent> addEditActivityLauncher = registerForActivityResult(  
 new ActivityResultContracts.StartActivityForResult(),  
 result -> mainListUpdate()  
 );  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 mainList = findViewById(R.id.*MainList*);  
 AddBtn = findViewById(R.id.*AddBtn*);  
 DelAllBtn = findViewById(R.id.*DelAllBtn*);  
 Objects.*requireNonNull*(getSupportActionBar()).setTitle(R.string.*main\_label*);  
 AddBtn.setOnClickListener(view -> {  
 Intent intent = new Intent(MainActivity.this, AddEditActivity.class);  
 addEditActivityLauncher.launch(intent);  
 });  
  
 DelAllBtn.setOnClickListener(view -> {  
 if(customAdapter.getItemCount() >0 ) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
 builder.setTitle(R.string.*deleting*)  
 .setMessage(R.string.*delete\_all\_teachers*)  
 .setCancelable(true)  
 .setNegativeButton(R.string.*no*,  
 (dialog, id) -> dialog.cancel())  
 .setPositiveButton(R.string.*yes*,  
 (dialog, id) -> {  
 DBH = new DBHelper(MainActivity.this);  
 DBH.delAllTeachers();  
 mainListUpdate();  
 dialog.cancel();  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 }  
 });  
  
 DBH = new DBHelper(MainActivity.this);  
 id = new ArrayList<>();  
 name = new ArrayList<>();  
 job\_title = new ArrayList<>();  
 degree = new ArrayList<>();  
 rank = new ArrayList<>();  
 photo = new ArrayList<>();  
 fillArrays();  
 customAdapter = new CustomAdapter(MainActivity.this,this, id, name, job\_title, degree, rank, photo);  
 mainList.setAdapter(customAdapter);  
 mainList.setLayoutManager(new LinearLayoutManager(MainActivity.this));  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode,resultCode, data);  
 if (requestCode == 1) {  
 mainListUpdate();  
 }  
 }  
  
 void mainListUpdate(){  
 fillArrays();  
 mainList.setAdapter(customAdapter);  
 }  
  
 void fillArrays(){  
 Cursor cursor = DBH.loadData();  
 id.clear();  
 name.clear();  
 job\_title.clear();  
 degree.clear();  
 rank.clear();  
 photo.clear();  
 if(cursor.getCount() == 0){  
 Toast.*makeText*(this, R.string.*no\_teachers*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 while(cursor.moveToNext()){  
 id.add(cursor.getString(0));  
 name.add(cursor.getString(1));  
 job\_title.add(cursor.getString(2));  
 degree.add(cursor.getString(3));  
 rank.add(cursor.getString(4));  
 photo.add(cursor.getString(5));  
 }  
 }  
 }  
}

Листинг кода AddEditActivity.java:

package com.lab3;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.Objects;  
  
public class AddEditActivity extends AppCompatActivity {  
  
 EditText Name, Job\_title, Degree, Rank, Photo;  
 TextView Title\_name, Title\_job\_title, Title\_degree, Title\_rank, Title\_photo;  
 Button saveBtn, delBtn, cancelBtn;  
 String id, name, jobTitle, degree, rank, photo;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_add\_edit);  
  
 Name = findViewById(R.id.*name*);  
 Job\_title = findViewById(R.id.*jobTitle*);  
 Degree = findViewById(R.id.*degree*);  
 Rank = findViewById(R.id.*rank*);  
 Photo = findViewById(R.id.*photo*);  
  
 saveBtn = findViewById(R.id.*saveBtn*);  
 delBtn = findViewById(R.id.*delBtn*);  
 cancelBtn = findViewById(R.id.*cancelBtn*);  
  
 id = getIntent().getStringExtra(getString(R.string.*c\_id*));  
 Title\_name = findViewById(R.id.*title\_name*);  
 Title\_job\_title = findViewById(R.id.*title\_job\_title*);  
 Title\_degree = findViewById(R.id.*title\_degree*);  
 Title\_rank = findViewById(R.id.*title\_rank*);  
 Title\_photo = findViewById(R.id.*title\_photo*);  
  
 ActionBar ab = getSupportActionBar();  
  
 if (getIntent().hasExtra(getString(R.string.c\_id))) {  
 Objects.requireNonNull(ab).setTitle(R.string.editing);  
 delBtn.setVisibility(View.VISIBLE);  
 getAndSetIntentData();  
 } else {  
 Objects.requireNonNull(ab).setTitle(R.string.adding);  
 delBtn.setVisibility(View.INVISIBLE);  
 }  
  
 saveBtn.setOnClickListener(view -> {  
 DBHelper dbh = new DBHelper(AddEditActivity.this);  
 AlertDialog.Builder builder = new AlertDialog.Builder(AddEditActivity.this);  
 id = getIntent().getStringExtra(getString(R.string.c\_id));  
 if (getIntent().hasExtra(getString(R.string.c\_id))) {  
 id = getIntent().getStringExtra(getString(R.string.c\_id));  
 if (  
 Name.getText().toString().isEmpty() ||  
 Job\_title.getText().toString().isEmpty() ||  
 Degree.getText().toString().isEmpty() ||  
 Rank.getText().toString().isEmpty()  
 ){  
 builder.setTitle(R.string.error)  
 .setMessage(R.string.errormessage1)  
 .setPositiveButton(R.string.ok, null)  
 .create();  
 builder.show();  
 }  
 else {  
 dbh.editData(id,  
 Name.getText().toString().trim(),  
 Job\_title.getText().toString().trim(),  
 Degree.getText().toString().trim(),  
 Rank.getText().toString().trim(),  
 Photo.getText().toString().trim()  
 );  
 finish();  
 }  
 }  
 else {  
 if (  
 Name.getText().toString().isEmpty() ||  
 Job\_title.getText().toString().isEmpty() ||  
 Degree.getText().toString().isEmpty() ||  
 Rank.getText().toString().isEmpty()  
 ){  
 builder.setTitle(R.string.error)  
 .setMessage(R.string.errormessage1)  
 .setPositiveButton(R.string.ok, null)  
 .create();  
 builder.show();  
 }  
 else {  
 dbh.addTeacher(  
 Name.getText().toString().trim(),  
 Job\_title.getText().toString().trim(),  
 Degree.getText().toString().trim(),  
 Rank.getText().toString().trim(),  
 Photo.getText().toString()  
 );  
 finish();  
 }  
 }  
 setResult(1, null);  
 });  
  
 delBtn.setOnClickListener(view -> {  
 if (getIntent().hasExtra(getString(R.string.c\_id))) {  
 String nid = id;  
 AlertDialog.Builder builder = new AlertDialog.Builder(AddEditActivity.this);  
 builder.setTitle(R.string.deleting)  
 .setMessage(R.string.delete\_teacher)  
 .setCancelable(true)  
 .setNegativeButton(R.string.cancel,  
 (dialog, id) -> dialog.cancel())  
 .setPositiveButton(R.string.ok,  
 (dialog, id) -> {  
 DBHelper dbh = new DBHelper(AddEditActivity.this);  
 dbh.delOneTeacher(nid);  
 dialog.cancel();  
 setResult(1, null);  
 finish();  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 }  
 });  
 cancelBtn.setOnClickListener(v -> finish());  
 }  
 void getAndSetIntentData(){  
 if(getIntent().hasExtra(getString(R.string.c\_id))){  
 id = getIntent().getStringExtra(getString(R.string.c\_id));  
 name = getIntent().getStringExtra(getString(R.string.c\_name));  
 jobTitle = getIntent().getStringExtra(getString(R.string.c\_job\_title));  
 degree = getIntent().getStringExtra(getString(R.string.c\_degree));  
 rank = getIntent().getStringExtra(getString(R.string.c\_rank));  
 photo = getIntent().getStringExtra(getString(R.string.c\_photo));  
  
 Name.setText(name);  
 Job\_title.setText(jobTitle);  
 Degree.setText(degree);  
 Rank.setText(rank);  
 Photo.setText(photo);  
 }  
 else Toast.makeText(this, R.string.error, Toast.LENGTH\_SHORT).show();  
 }  
}

Листинг кода DBHelper.java:

package com.lab3;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.widget.Toast;  
import androidx.annotation.Nullable;  
  
public class DBHelper extends SQLiteOpenHelper {  
 private final Context context;  
 private static final String *DB\_NAME* = "Teachers.db";  
 private static final int *DB\_VERSION* = 1;  
 private static final String *TBL\_NAME* = "tblTeachers";  
 private static final String *C\_ID* = "id";  
 private static final String *C\_NAME* = "name";  
 private static final String *C\_JOB\_TITLE* = "jobTitle";  
 private static final String *C\_DEGREE* = "degree";  
 private static final String *C\_RANK* = "rank";  
 private static final String *C\_PHOTO* = "photo";  
 public DBHelper(@Nullable Context context){  
 super(context,*DB\_NAME*,null,*DB\_VERSION*);  
 this.context = context;  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 //создание новой таблицы в БД  
 String Query = "CREATE TABLE " + *TBL\_NAME* + "( " +  
 *C\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  
 *C\_NAME* + " TEXT, " +  
 *C\_JOB\_TITLE* + " TEXT, " +  
 *C\_DEGREE* + " TEXT, " +  
 *C\_RANK* + " TEXT, " +  
 *C\_PHOTO* + " TEXT)";  
 db.execSQL(Query);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int i, int i1) {  
 //удаление таблицы в БД, если она есть  
 db.execSQL("DROP TABLE IF EXISTS " + *TBL\_NAME* + ";");  
 onCreate(db);  
 }  
  
 void addTeacher(String name, String job\_title, String degree, String rank, String photo){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues cv = new ContentValues();  
 cv.put(*C\_NAME*, name);  
 cv.put(*C\_JOB\_TITLE*, job\_title);  
 cv.put(*C\_DEGREE*, degree);  
 cv.put(*C\_RANK*, rank);  
 cv.put(*C\_PHOTO*, photo);  
 long res = db.insert(*TBL\_NAME*, null,cv);  
 if(res == -1){  
 Toast.*makeText*(context, R.string.*error\_add*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(context, R.string.*success\_add*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 Cursor loadData(){  
 String Query = "SELECT " + *C\_ID* + "," + *C\_NAME* + "," + *C\_JOB\_TITLE* + "," + *C\_DEGREE* + "," + *C\_RANK* + "," + *C\_PHOTO* +  
 " FROM " + *TBL\_NAME* +  
 " ORDER BY " + *C\_RANK* + ", " + *C\_NAME* + ";";  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = null;  
 if(db != null){  
 cursor = db.rawQuery(Query,null);  
 }  
 return cursor;  
 }  
  
 void editData(String id, String name, String job\_title, String rank, String degree, String photo){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues cv = new ContentValues();  
 cv.put(*C\_ID*, id);  
 cv.put(*C\_NAME*, name);  
 cv.put(*C\_JOB\_TITLE*, job\_title);  
 cv.put(*C\_DEGREE*, degree);  
 cv.put(*C\_RANK*, rank);  
 cv.put(*C\_PHOTO*, photo);  
 long result = db.update(*TBL\_NAME*, cv, *C\_ID*+"=?", new String[]{id});  
 if(result == -1){  
 Toast.*makeText*(context, R.string.*error\_edit*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(context, R.string.*success\_edit*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 void delOneTeacher(String id){  
 SQLiteDatabase db = this.getWritableDatabase();  
 long result = db.delete(*TBL\_NAME*, *C\_ID*+"=?", new String[]{id});  
 if(result == -1){  
 Toast.*makeText*(context, R.string.*error\_delete*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(context, R.string.*success\_delete*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 void delAllTeachers(){  
 SQLiteDatabase db = this.getWritableDatabase();  
 long result = db.delete(*TBL\_NAME*,"",new String[]{});  
 if(result == -1){  
 Toast.*makeText*(context, R.string.*error\_delete*, Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(context, R.string.*success\_delete\_all*, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

Листинг кода CustomAdapter.java:

package com.lab3;  
  
import android.annotation.SuppressLint;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.text.SpannableStringBuilder;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.StringRes;  
import androidx.appcompat.app.AlertDialog;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.bumptech.glide.Glide;  
import com.bumptech.glide.request.RequestOptions;  
  
import java.util.ArrayList;  
  
public class CustomAdapter extends RecyclerView.Adapter<CustomAdapter.MyViewHolder>{  
 private final Activity activity;  
 private final Context context;  
 private final ArrayList<String> id, name, job\_title, degree, rank, photo;  
 CustomAdapter(  
 Activity activity, Context context, ArrayList<String> id, ArrayList<String> name,  
 ArrayList<String> jobTitle, ArrayList<String> degree, ArrayList<String> rank, ArrayList<String> photo){  
 this.activity = activity;  
 this.context = context;  
 this.id = id;  
 this.name = name;  
 this.job\_title = jobTitle;  
 this.degree = degree;  
 this.rank = rank;  
 this.photo = photo;  
 }  
  
 @NonNull  
 @Override  
 public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType){  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 View view = inflater.inflate(R.layout.*my\_row*, parent, false);  
 return new MyViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull final MyViewHolder holder, @SuppressLint("RecyclerView") final int position) {  
 holder.name.setText(buildSpannableString(R.string.*row\_name*, String.*valueOf*(name.get(position))));  
 holder.jobTitle.setText(buildSpannableString(R.string.*row\_job\_title*, String.*valueOf*(job\_title.get(position))));  
 holder.degree.setText(buildSpannableString(R.string.*row\_degree*, String.*valueOf*(degree.get(position))));  
 holder.rank.setText(buildSpannableString(R.string.*row\_rank*, String.*valueOf*(rank.get(position))));  
  
 String photoURL = photo.get(position);  
 Glide.*with*(context)  
 .applyDefaultRequestOptions(new RequestOptions())  
 .load(photoURL)  
 .error(R.drawable.*default\_photo*)  
 .into(holder.imageViewPhoto);  
  
  
 holder.mainLayout.setOnClickListener(view -> {  
 Intent intent = new Intent(context,AddEditActivity.class);  
 intent.putExtra(activity.getString(R.string.*c\_id*),String.*valueOf*(id.get(position)));  
 intent.putExtra(activity.getString(R.string.*c\_name*),String.*valueOf*(name.get(position)));  
 intent.putExtra(activity.getString(R.string.*c\_job\_title*),String.*valueOf*(job\_title.get(position)));  
 intent.putExtra(activity.getString(R.string.*c\_degree*),String.*valueOf*(degree.get(position)));  
 intent.putExtra(activity.getString(R.string.*c\_rank*),String.*valueOf*(rank.get(position)));  
 intent.putExtra(activity.getString(R.string.*c\_photo*),String.*valueOf*(photo.get(position)));  
 activity.startActivityForResult(intent, 1);  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return id.size();  
 }  
  
 class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{  
 TextView name, jobTitle, degree, rank, photo;  
 ImageView imageViewPhoto;  
 ImageButton btnDelRV;  
 LinearLayout mainLayout;  
 MyViewHolder(@NonNull View itemView) {  
 super(itemView);  
 name = itemView.findViewById(R.id.*name*);  
 jobTitle = itemView.findViewById(R.id.*jobTitle*);  
 degree = itemView.findViewById(R.id.*degree*);  
 rank = itemView.findViewById(R.id.*rank*);  
 photo = itemView.findViewById(R.id.*photo*);  
 imageViewPhoto = itemView.findViewById(R.id.*imageViewPhoto*);  
 mainLayout = itemView.findViewById(R.id.*mainLayout*);  
 btnDelRV = itemView.findViewById(R.id.*delBtnRV*);  
 btnDelRV.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View view) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(activity);  
 builder.setTitle(R.string.*deleting*)  
 .setMessage(R.string.*delete\_teacher*)  
 .setCancelable(true)  
 .setNegativeButton(R.string.*cancel*,  
 (dialog, id) -> dialog.cancel())  
 .setPositiveButton(R.string.*ok*,  
 (dialog, id) -> {  
 DBHelper dbh = new DBHelper(activity);  
 int itemPos = getAdapterPosition();  
 String nid = CustomAdapter.this.id.get(itemPos);  
 dbh.delOneTeacher(nid);  
 activity.recreate();  
 dialog.cancel();  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 }  
 }  
  
 private SpannableStringBuilder buildSpannableString(@StringRes int stringResId, String value) {  
 String rowText = context.getResources().getString(stringResId);  
 SpannableStringBuilder spannableStringBuilder = new SpannableStringBuilder(rowText);  
 spannableStringBuilder.append(" ").append(value);  
 return spannableStringBuilder;  
 }  
}

**2.5 Тестирование мобильного приложения**

Тестирование выполнялось на физическом устройстве (мобильный телефон Xiaomi Redmi Note 9 Pro).

Результат работы мобильного приложения представлен на рисунках 8-19.

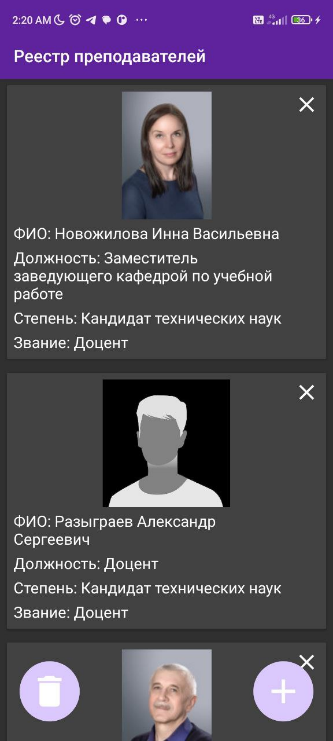


Рисунок 8 – Результат работы главной активности мобильного приложения на темной теме, портретная ориентация

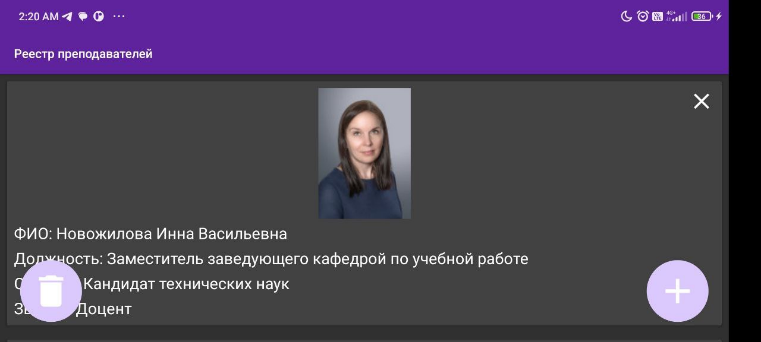


Рисунок 9 – Результат работы главной активности мобильного приложения на темной теме, альбомная ориентация

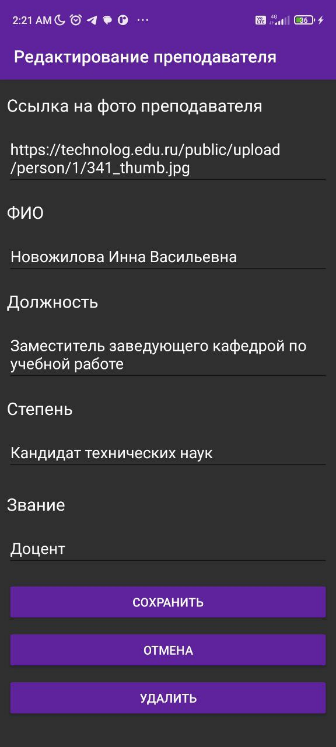


Рисунок 10 – Результат работы активности добавления/редактирования записи на темной теме, портретная ориентация

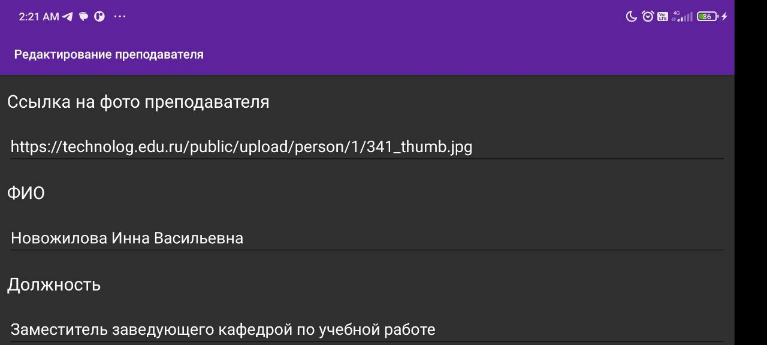


Рисунок 11 – Результат работы активности добавления/редактирования записи на темной теме, альбомная ориентация

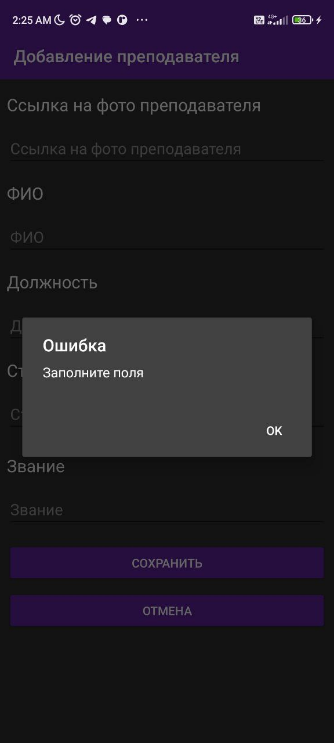


Рисунок 12 – Диалоговое окно мобильного приложения на темной теме, портретная ориентация

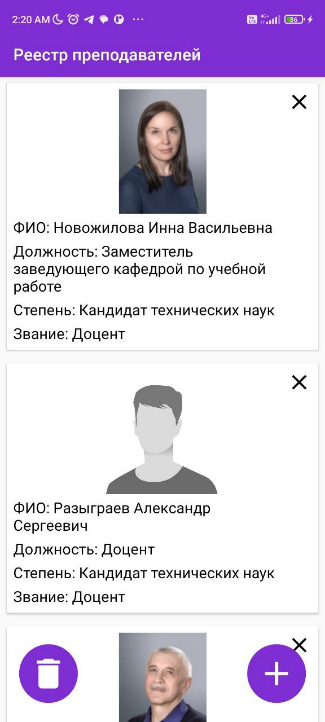


Рисунок 13 – Результат работы главной активности мобильного приложения на светлой теме, портретная ориентация

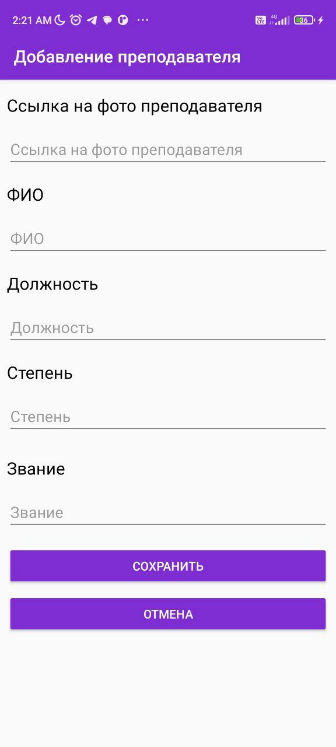


Рисунок 14 – Результат работы активности добавления/редактирования записи на светлой теме, портретная ориентация

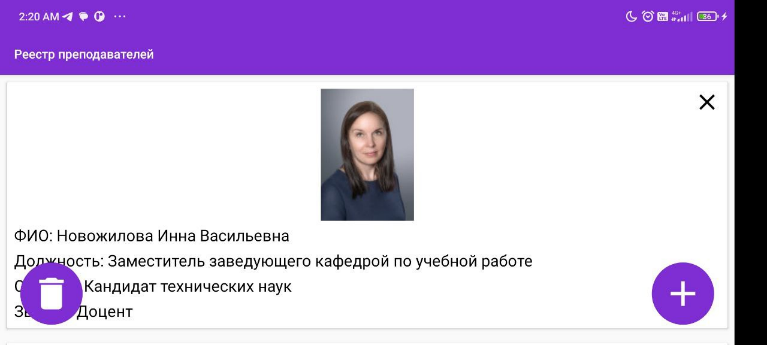


Рисунок 15 – Результат работы главной активности мобильного приложения на светлой теме, альбомная ориентация

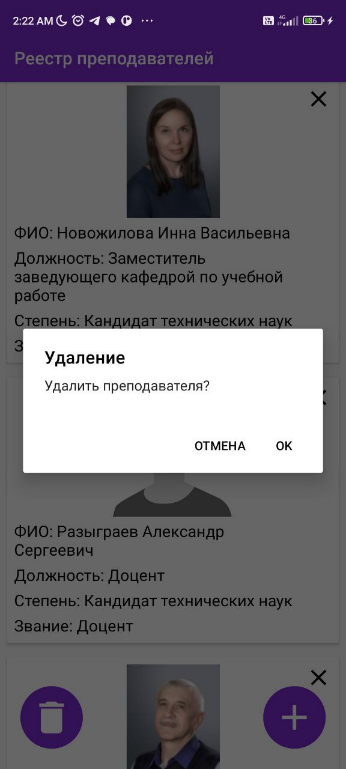


Рисунок 16 – Диалоговое окно мобильного приложения на светлой теме, портретная ориентация

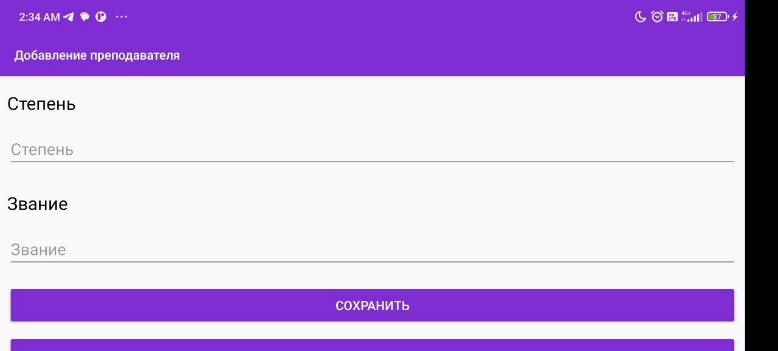


Рисунок 17 – Результат работы активности добавления/редактирования записи на светлой теме, альбомная ориентация

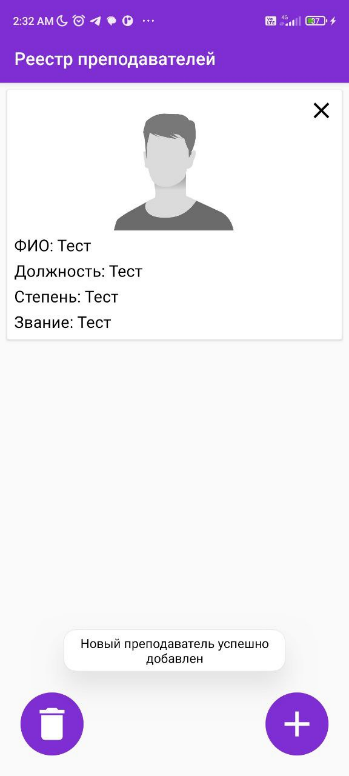


Рисунок 18 – Сообщение о действии с заметкой на главной активности мобильного приложения на светлой теме, портретная ориентация

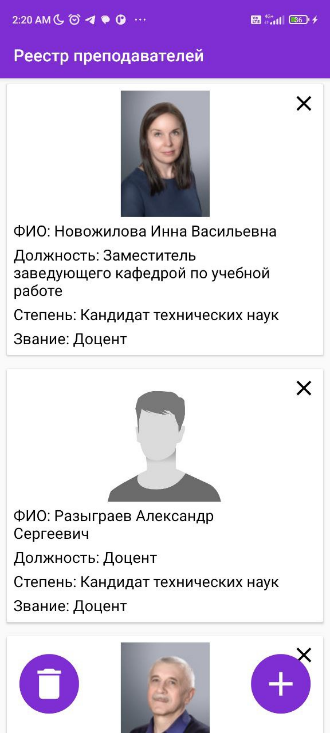


Рисунок 19 – Результат работы сортировки записей по ФИО на главной активности мобильного приложения на светлой теме, портретная ориентация

**Выводы**

В ходе выполнения лабораторной работы было разработано мобильное приложения для Android для управления данными локального хранилища мобильного устройства на основе интегрированной СУБД SQLite и графического компонента RecyclerView. Для достижения поставленной цели были выполнены следующие задачи:

1) Изучена специфика языка запросов и типов данных SQLite, механизма подключения и использования интегрированной СУБД SQLite для управления данными в памяти мобильного устройства. Ознакомились с принципами настройки адаптера данных, связывания данных, особенностями доступа к данным с использованием объекта cursor.

2) Разработано мобильное приложение для Android в среде Android Studio Giraffe 2022.3.1:

- создан и настроен новый проект приложения;

- указано наименование приложения и выбраны версии SDK: минимальная – 24, целевая – 31, а также версии ОС Android: минимальная – 7, целевая – 13;

- выбран язык программирования – Java.

- создана разметка макета графического интерфейса для двух активностей и элемента списка;

- создана управляющая логика (класс для работы с SQLite, адаптер данных, код главной активности, код активности для создания/редактирования/удаления элемента списка);

- работа с ресурсами приложения: строковыми константами, цветами, шрифтами, стилями, пиктограммами, установка иконки приложения;

- разметка макета графического интерфейса для двух активностей и элемента списка;

- создание управляющей логики (класса для работы с SQLite, адаптера данных, кода главной активности, кода активности для создания/редактирования элемента списка);

- проведено тестирование мобильного приложения. Тестирование показало работоспособность мобильного приложения в портретной и альбомной ориентации, а также светлой и темной теме.

Характеристики аппаратного обеспечения разработчика мобильного приложения приведены в таблице 4.

Таблица 4 – Характеристики аппаратного обеспечения

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Ноутбук |
| Тип и тактовая частота (ν) процессора | Intel Core i7, ν = 2,71 ГГц |
| Состав и технические характеристики периферийных устройств ЭВМ | Мобильный телефон realme 10;  Кабель USB-A – USB-C |
| Операционная система ЭВМ | Windows 10 |
| Версия Android Studio | Android Studio Giraffe 2022.3.1 |