Донецкий Национальный Технический Университет

Лабораторная работа № 4

«Работа с адресной книгой»

Выполнил:

ст. группы ПИ-13

Лысенко А. С.

Проверил:

ст. препод. Костин В.И.

Покровск 2017

Задание. Создать приложение, которое добавляет, редактирует и удаляет контакт из адресной книги. При добавлении контакта необходимо указать фамилию, имя, номер мобильного телефона, E-mail, место работы, рабочий телефон, а также позволить пользователю выбрать для контакта изображение и рингтон.

Для создания приложения и хранения данных используется встроенная база данных SQLite.

В программе реализованы 2 класса MainActivity и DBHelper.

В DBHelper описываем подключение к БД и создание таблицы.

В MainActivity описываем реализацию работы с БД.

MainActivity.java

**package** com.example.sudo.adressbookextended;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
  
**import** android.app.Activity;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.view.View.OnClickListener;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
  
**public class** MainActivity **extends** Activity **implements** OnClickListener {  
  
 **final** String **LOG\_TAG** = **"myLogs"**;  
  
 Button **btnAdd**, **btnRead**, **btnClear**, **btnUpdate**, **btnDelete**;  
 EditText **etName**, **etEmail**, **etId**, **etSurname**, **etWork**, **etPhoneWork**, **etPhoneNumber**;  
  
 DBHelper **dbHelper**;  
  
 */\*\* Called when the activity is first created. \*/* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **btnAdd** = (Button) findViewById(R.id.***btnAdd***);  
 **btnAdd**.setOnClickListener(**this**);  
  
 **btnRead** = (Button) findViewById(R.id.***btnRead***);  
 **btnRead**.setOnClickListener(**this**);  
  
 **btnClear** = (Button) findViewById(R.id.***btnClear***);  
 **btnClear**.setOnClickListener(**this**);  
  
 **btnDelete** = (Button) findViewById(R.id.***btnDelete***);  
 **btnDelete**.setOnClickListener(**this**);  
  
 **btnUpdate** = (Button) findViewById(R.id.***btnUpdate***);  
 **btnUpdate**.setOnClickListener(**this**);  
  
 **etId** = (EditText) findViewById(R.id.***etId***);  
 **etName** = (EditText) findViewById(R.id.***etName***);  
 **etEmail** = (EditText) findViewById(R.id.***etEmail***);  
 **etSurname** = (EditText) findViewById(R.id.***etSurname***);  
 **etWork** = (EditText) findViewById(R.id.***etWork***);  
 **etPhoneNumber** = (EditText) findViewById(R.id.***etPhoneNumber***);  
 **etPhoneWork** = (EditText) findViewById(R.id.***etPhoneWork***);  
  
 *// создаем объект для создания и управления версиями БД* **dbHelper** = **new** DBHelper(**this**);  
 }  
  
  
 @Override  
 **public void** onClick(View v) {  
  
 *// создаем объект для данных* ContentValues cv = **new** ContentValues();  
  
 *// получаем данные из полей ввода* String name = **etName**.getText().toString();  
 String email = **etEmail**.getText().toString();  
 String id = **etId**.getText().toString();  
 String surname = **etSurname**.getText().toString();  
 String work = **etWork**.getText().toString();  
 String phoneNumber = **etPhoneWork**.getText().toString();  
 String phoneWork = **etPhoneWork**.getText().toString();  
  
 *// подключаемся к БД* SQLiteDatabase db = **dbHelper**.getWritableDatabase();  
  
  
 **switch** (v.getId()) {  
 **case** R.id.***btnAdd***:  
 Log.*d*(**LOG\_TAG**, **"--- Insert in mytable: ---"**);  
 *// подготовим данные для вставки в виде пар: наименование столбца - значение* cv.put(**"name"**, name);  
 cv.put(**"email"**, email);  
 cv.put(**"surname"**, surname);  
 cv.put(**"phoneNumber"**, phoneNumber);  
 cv.put(**"workPhoneNumber"**, phoneWork);  
 cv.put(**"work"**, work);  
  
 *// вставляем запись и получаем ее ID* **long** rowID = db.insert(**"mytable"**, **null**, cv);  
 Log.*d*(**LOG\_TAG**, **"row inserted, ID = "** + rowID);  
 **break**;  
 **case** R.id.***btnRead***:  
 Log.*d*(**LOG\_TAG**, **"--- Rows in mytable: ---"**);  
 *// делаем запрос всех данных из таблицы mytable, получаем Cursor* Cursor c = db.query(**"mytable"**, **null**, **null**, **null**, **null**, **null**, **null**);  
  
 *// ставим позицию курсора на первую строку выборки  
 // если в выборке нет строк, вернется false* **if** (c.moveToFirst()) {  
  
 *// определяем номера столбцов по имени в выборке* **int** idColIndex = c.getColumnIndex(**"id"**);  
 **int** nameColIndex = c.getColumnIndex(**"name"**);  
 **int** emailColIndex = c.getColumnIndex(**"email"**);  
 **int** surnameColIndex = c.getColumnIndex(**"surname"**);  
 **int** workColIndex = c.getColumnIndex(**"work"**);  
 **int** phoneNumberColIndex = c.getColumnIndex(**"phoneNumber"**);  
 **int** workPhoneNumberColIndex = c.getColumnIndex(**"workPhoneNumber"**);  
  
  
  
 **do** {  
 *// получаем значения по номерам столбцов и пишем все в лог* Log.*d*(**LOG\_TAG**,  
 **"ID = "** + c.getInt(idColIndex) +  
 **", name = "** + c.getString(nameColIndex) +  
 **", email = "** + c.getString(emailColIndex)  
 + **", surname = "** + c.getString(surnameColIndex)  
 + **", work = "** + c.getString(workColIndex)  
 + **", phoneNumber = "** + c.getString(phoneNumberColIndex)  
 + **", workPhoneNumber = "** + c.getString(workPhoneNumberColIndex));  
 *// переход на следующую строку  
 // а если следующей нет (текущая - последняя), то false - выходим из цикла* } **while** (c.moveToNext());  
 } **else** Log.*d*(**LOG\_TAG**, **"0 rows"**);  
 c.close();  
 **break**;  
 **case** R.id.***btnClear***:  
 Log.*d*(**LOG\_TAG**, **"--- Clear mytable: ---"**);  
 *// удаляем все записи* **int** clearCount = db.delete(**"mytable"**, **null**, **null**);  
 Log.*d*(**LOG\_TAG**, **"deleted rows count = "** + clearCount);  
 **break**;  
 *// удаляем запись с указанным id* **case** R.id.***btnDelete***:  
 **if**(id.equalsIgnoreCase(**""**))  
 {  
 **break**;  
 }  
 **int** delCount = db.delete(**"mytable"**, **"id"** + **"="** + id, **null**);  
 Log.*d*(**LOG\_TAG**, **"delete rows count = "** + delCount);  
 *//изменяем запись с указанным id* **case** R.id.***btnUpdate***:  
 **if**(id.equalsIgnoreCase(**""**))  
 {  
 **break**;  
 }  
 cv.put(**"name"**, name);  
 cv.put(**"email"**, email);  
 cv.put(**"surname"**, surname);  
 cv.put(**"phoneNumber"**, phoneNumber);  
 cv.put(**"workPhoneNumber"**, phoneWork);  
 cv.put(**"work"**, work);  
  
  
 **int** updCount = db.update(**"mytable"**, cv, **"id"** + **"= ?"**, **new** String[] {id});  
  
 Log.*d*(**LOG\_TAG**, **"updates rows count = "** + updCount);  
  
 }  
 *// закрываем подключение к БД* **dbHelper**.close();  
 }  
  
  
  
 **class** DBHelper **extends** SQLiteOpenHelper {  
  
 **public** DBHelper(Context context) {  
 *// конструктор суперкласса* **super**(context, **"myDB"**, **null**, 1);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 Log.*d*(**LOG\_TAG**, **"--- onCreate database ---"**);  
 *// создаем таблицу с полями* db.execSQL(**"create table mytable ("** + **"id integer primary key autoincrement,"** + **"surname text,"** + **"name text,"** + **"phoneNumber text,"** + **"email text,"** + **"work text,"** + **"workPhoneNumber text"** + **");"**);  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 Log.*d*(**LOG\_TAG**, **"--- Drop table if exists ---"**);  
  
 onCreate(db);  
 }  
 }  
  
}

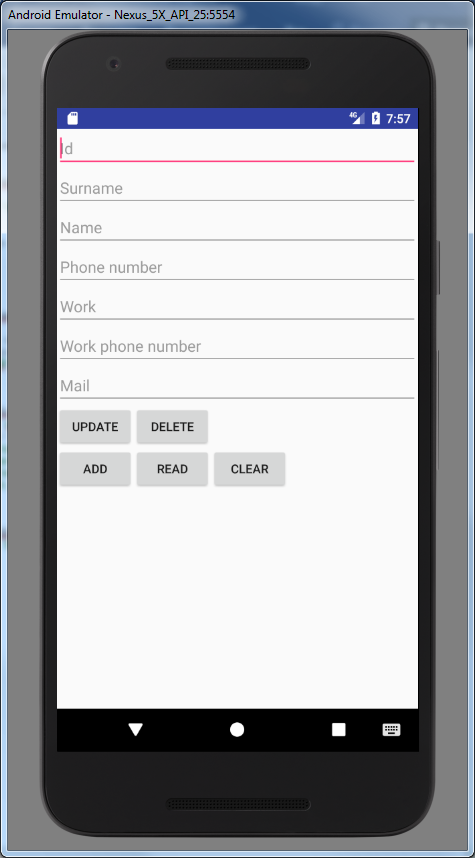
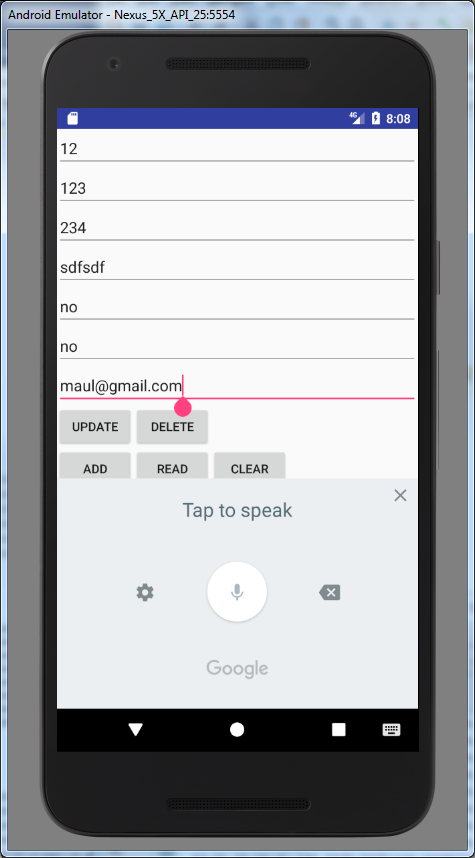
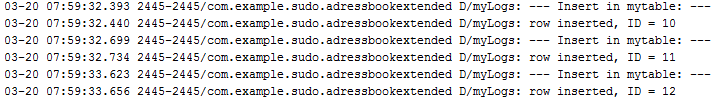
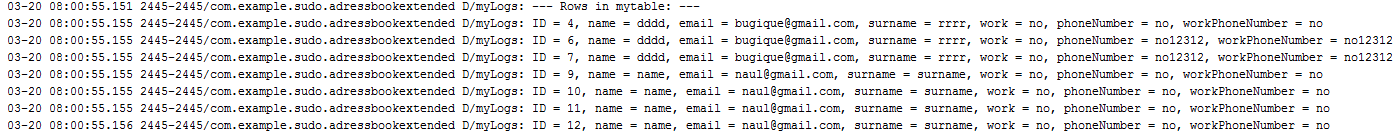


Рис. 1 вид приложения

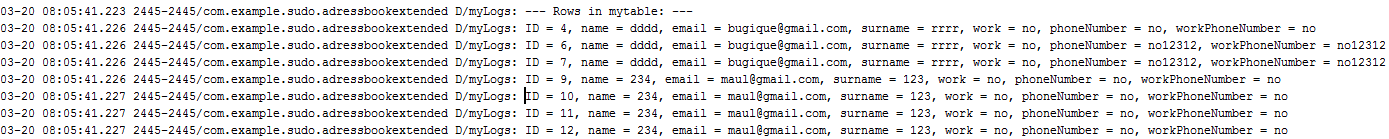




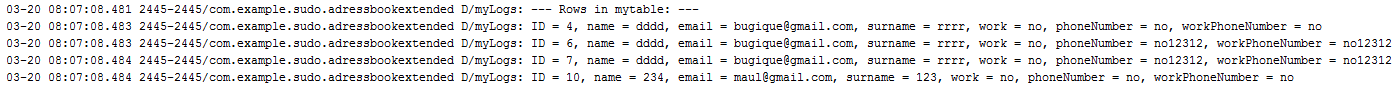
Добавление записей в БД



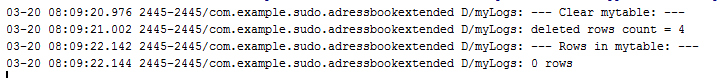
Чтение записей из БД



Обновление БД



Удаление данных из БД



Очистка БД