Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт «**Практическая работа №22.1**»

Вариант 9. База данных фильмов.

Выполнила: Зиятдинова Алина Ленаровна

Группа: ПР-31

2023

|  |
| --- |
| Манифест |
|  |

|  |
| --- |
|  |
| package com.example.prakt22\_2\_  import android.annotation.SuppressLint import android.content.Intent import android.content.SharedPreferences import android.os.Bundle import android.view.View import android.widget.Button import android.widget.CheckBox import android.widget.EditText import androidx.appcompat.app.AppCompatActivity import com.google.android.material.snackbar.Snackbar  class MainActivity : AppCompatActivity() {  lateinit var login: EditText  lateinit var password: EditText  lateinit var checkBox: CheckBox  lateinit var button: Button  lateinit var snackbar: Snackbar  lateinit var dataStore: SharedPreferences  var prefs\_name: String = "PrefersFile"  @SuppressLint("MissingInflatedId")  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  login = findViewById(R.id.*edittext\_login*)  password = findViewById(R.id.*edittext\_password*)  checkBox = findViewById(R.id.*checkbox*)  button = findViewById(R.id.*button\_next*)  dataStore = getSharedPreferences(prefs\_name, *MODE\_PRIVATE*)  getPreferencesData()  }  fun next (view:View)  {  if (login.*text*.*isNotEmpty*() && password.*text*.*isNotEmpty*()) {  if (checkBox.*isChecked*) {  dataStore = getPreferences(*MODE\_PRIVATE*)  val ed: SharedPreferences.Editor = dataStore.edit()  ed.putString("login", login.getText().toString())  ed.putString("password", password.getText().toString())  ed.apply()   val db = DBHelper (this, null)  val isAuth = db.getUser(login.*text*.toString(), password.*text*.toString())  if (isAuth == true)  {  val intent = Intent(this, SecondActivity::class.*java*)  intent.putExtra("login", "${login.*text*.*trim*().toString()}")  startActivity(intent)  }  else  {  snackbar = Snackbar.make(view,R.string.*error\_auth*, Snackbar.*LENGTH\_LONG*)  snackbar.show()  }  }  else {  dataStore.edit().clear().apply()  val db = DBHelper (this, null)  val isAuth = db.getUser(login.*text*.toString(), password.*text*.toString())  if (isAuth == true)  {  val intent = Intent(this, SecondActivity::class.*java*)  intent.putExtra("login", "${login.*text*.*trim*().toString()}")  startActivity(intent)  }  else  {  snackbar = Snackbar.make(view,R.string.*error\_auth*, Snackbar.*LENGTH\_LONG*)  snackbar.show()  }  }  }  else  {  snackbar = Snackbar.make(view,R.string.*error\_emptyfields*, Snackbar.*LENGTH\_LONG*)  snackbar.show()  }  }   fun registration(view:View)  {  val intent = Intent(this, RegistrationPage::class.*java*)  startActivity(intent)  }   private fun getPreferencesData() {  dataStore = getPreferences(*MODE\_PRIVATE*);  login.setText(dataStore.getString("login", ""));  password.setText(dataStore.getString("password", ""));  } } |
| <?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:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@color/white"  android:orientation="vertical"  tools:context=".MainActivity">  <ImageView  android:id="@+id/icon"  android:layout\_width="100dp"  android:layout\_height="100dp"  android:layout\_marginTop="50dp"  android:layout\_gravity="center"  android:src="@drawable/icon\_1">  </ImageView>   <TextView  android:layout\_width="match\_parent"  android:layout\_height="30dp"  android:text="@string/title\_login"  android:textSize="18dp"  android:textColor="@color/gray"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="50dp">  </TextView>  <EditText  android:id="@+id/edittext\_login"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:inputType="text"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <TextView  android:layout\_width="match\_parent"  android:layout\_height="30dp"  android:text="@string/title\_password"  android:textSize="18dp"  android:textColor="@color/gray"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="10dp">  </TextView>  <EditText  android:id="@+id/edittext\_password"  android:layout\_width="match\_parent"  android:padding="10dp"  android:inputType="textPassword"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890.,"  android:numeric="integer"  android:layout\_height="40dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>   <CheckBox  android:id="@+id/checkbox"  android:layout\_width="121dp"  android:layout\_height="40dp"  android:layout\_marginLeft="260dp"  android:text="@string/remember\_me"  android:textColor="@color/gray"  android:textSize="15dp"></CheckBox>   <Button  android:id="@+id/button\_next"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:textSize="18dp"  android:onClick="next"  android:textColor="@color/blue"  android:textAllCaps="false"  android:background="@drawable/background\_view"  android:text="Далее">  </Button>  <Button  android:id="@+id/button\_registration"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:textSize="18dp"  android:onClick="registration"  android:layout\_marginTop="10dp"  android:textColor="@color/blue"  android:textAllCaps="false"  android:background="@drawable/background\_view"  android:text="@string/noaccount\_register">  </Button> </LinearLayout> |

|  |
| --- |
| <?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.READ\_EXTERNAL\_STORAGE" />  <uses-permission android:name="android.permission.INTERNET" />  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  <application  android:allowBackup="true"  android:dataExtractionRules="@xml/data\_extraction\_rules"  android:fullBackupContent="@xml/backup\_rules"  android:icon="@drawable/icon\_1"  android:label="@string/app\_name"  android:roundIcon="@drawable/icon\_1"  android:supportsRtl="true"  android:theme="@style/Theme.Prakt22\_2\_"  tools:targetApi="31">  <activity  android:name=".EditPage"  android:exported="false">  <meta-data  android:name="android.app.lib\_name"  android:value="" />  </activity>  <activity  android:name=".RegistrationPage"  android:exported="false">  <meta-data  android:name="android.app.lib\_name"  android:value="" />  </activity>  <activity  android:name=".ThirdActivity"  android:exported="false"  android:noHistory="true">  <meta-data  android:name="android.app.lib\_name"  android:value="" />  </activity>  <activity  android:name=".SecondActivity"  android:exported="false">  <meta-data  android:name="android.app.lib\_name"  android:value="" />  </activity>  <activity  android:name=".MainActivity"  android:exported="true"  android:noHistory="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> |

|  |
| --- |
|  |
| package com.example.prakt22\_2\_  import android.annotation.SuppressLint import android.content.Intent import android.os.Bundle import android.util.Log import android.view.View import android.widget.\* import androidx.appcompat.app.AppCompatActivity import com.android.volley.toolbox.StringRequest import com.android.volley.toolbox.Volley import com.example.prakt22\_2\_.databinding.ActivitySecondBinding import com.squareup.picasso.Picasso import org.json.JSONObject   class SecondActivity : AppCompatActivity() {  lateinit var movie\_name: EditText  lateinit var button\_search: Button  lateinit var image\_poster: ImageView  lateinit var toast: Toast  lateinit var binding: ActivitySecondBinding  @SuppressLint("MissingInflatedId")  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivitySecondBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)  movie\_name = findViewById(R.id.*edittext\_namemovie*)  button\_search = findViewById(R.id.*button\_search*)  image\_poster = findViewById(R.id.*image\_poster*)  val arguments = *intent*.*extras* val info\_login = arguments!!["login"].*toString*()  binding.buttonSearch.setOnClickListener()  **{** if (movie\_name.*text*.*isNotEmpty*()) {  var url =  "https://www.omdbapi.com/?apikey=36de77d8&t="+movie\_name.*text* val queue = Volley.newRequestQueue(this)  val stringR = StringRequest(  com.android.volley.Request.Method.*GET*,  url,  **{** response **->** val obj = JSONObject(response)  var title = getString(R.string.*title\_year*)  var info\_year = obj.getString("Year").toString()  var string\_info\_year = "$title $info\_year".toString()   title = getString(R.string.*title\_name*)  var info\_title = obj.getString("Title").toString()  var string\_info\_title = "$title $info\_title".toString()   title = getString(R.string.*title\_type*)  var info\_type = obj.getString("Type").toString()  var string\_info\_type = "$title $info\_type".toString()   title = getString(R.string.*title\_released*)  var info\_released = obj.getString("Released").toString()  var string\_info\_released = "$title $info\_released".toString()   title = getString(R.string.*title\_actors*)  var info\_actors = obj.getString("Actors").toString()  var string\_info\_actors = "$title $info\_actors".toString()   title = getString(R.string.*title\_plot*)  var info\_plot = obj.getString("Plot").toString()  var string\_info\_plot = "$title $info\_plot".toString()   var listview\_array = ArrayList<String>()  listview\_array.add("$string\_info\_title")  listview\_array.add("$string\_info\_type")  listview\_array.add("$string\_info\_released")  listview\_array.add("$string\_info\_actors")  listview\_array.add("${string\_info\_plot}")   val adapter = ArrayAdapter(this, R.layout.*list\_item*, R.id.*textView\_item*, listview\_array)  binding.listviewPlot.*adapter* = adapter   var info\_image = obj.getString("Poster").toString()  Picasso.get().load(info\_image).into(image\_poster)   Log.d("MyLog", "Response:$response")    val db = MovieDBHelper (this, null)  val isStock = db.getMovie(info\_title.toString(), info\_login.toString())   if (isStock == false)  {  val movie = Movie("$info\_login", "$info\_title", "$info\_year", "$info\_image",  "$info\_type", "$info\_released", "$info\_actors", "$info\_plot")  db.addMovie(movie)  } /\* Thread {  db.getDao().insertItem(item)  }.start()\*/  **}**,  **{** Log.d("MyLog", "Volley error:$**it**")  **}**)  queue.add(stringR)  }  else  {  toast = Toast.makeText(this,"Введите название фильма", Toast.*LENGTH\_LONG*)  toast.show()  }  **}** }  fun nextpage (view: View)  {  val arguments = *intent*.*extras* val info\_login = arguments!!["login"].*toString*()  val intent = Intent(this, ThirdActivity::class.*java*)  intent.putExtra("login", "${info\_login.*trim*().toString()}")  startActivity(intent)  } } |
| <?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:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@color/white"  android:orientation="vertical"  tools:context=".SecondActivity"> <ImageView  android:layout\_width="80dp"  android:layout\_height="80dp"  android:layout\_marginTop="10dp"  android:layout\_gravity="center"  android:src="@drawable/icon\_1"> </ImageView>  <TextView  android:id="@+id/titleee"  android:layout\_width="match\_parent"  android:layout\_height="30dp"  android:text="@string/title\_namemovie"  android:textSize="20dp"  android:textColor="@color/gray"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="8dp">  </TextView>  <EditText  android:id="@+id/edittext\_namemovie"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789"  android:inputType="text"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <Button  android:id="@+id/button\_search"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:textSize="18dp"  android:textColor="@color/blue"  android:layout\_marginTop="10dp"  android:textAllCaps="false"  android:background="@drawable/background\_view"  android:text="@string/textbutton\_search">   </Button>  <ImageView  android:id="@+id/image\_poster"  android:layout\_width="80dp"  android:layout\_height="130dp"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="5dp">  </ImageView>   <ListView  android:id="@+id/listview\_plot"  android:layout\_width="match\_parent"  android:layout\_height="290dp"  android:layout\_marginRight="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/title\_plot"  android:textSize="20dp" />   <Button  android:textAllCaps="false"  android:textSize="18dp"  android:textColor="@color/blue"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:background="@drawable/background\_view"  android:text="@string/next\_page\_bd"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:layout\_marginBottom="10dp"  android:onClick="nextpage"></Button> </LinearLayout> |

|  |
| --- |
| package com.example.prakt22\_2\_  import android.annotation.SuppressLint import android.content.ContentValues import android.content.Context import android.database.Cursor import android.database.sqlite.SQLiteConstraintException import android.database.sqlite.SQLiteDatabase import android.database.sqlite.SQLiteException import android.database.sqlite.SQLiteOpenHelper import android.view.View  class MovieDBHelper(context: Context, val factory: SQLiteDatabase.CursorFactory?) :  SQLiteOpenHelper(context, "movie\_database.db", null, 1) {   override fun onCreate(db: SQLiteDatabase) {  val quary = "CREATE TABLE movie\_database (id INT PRIMARY KEY, login TEXT, title TEXT, year TEXT, image TEXT, type TEXT, released TEXT, actors TEXT, plot TEXT)"  db!!.execSQL(quary)  }   override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {  "DROP TABLE IF EXISTS " + DBContract.MovieEntry.TABLE\_NAME  onCreate(db)  }   override fun onDowngrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {  onUpgrade(db, oldVersion, newVersion)  }  fun addMovie(movie: Movie)  {  val values = ContentValues()  values.put("login", movie.login)  values.put("title", movie.title)  values.put("year", movie.year)  values.put("type", movie.type)  values.put("released", movie.released)  values.put("actors", movie.actors)  values.put("plot", movie.plot)  values.put("image", movie.image)   val db = this.*writableDatabase* db.insert("movie\_database", null, values)  db.close()  }  fun getMovie(title: String, login: String):Boolean  {  val db = this.*readableDatabase* val result = db.rawQuery("SELECT \* FROM movie\_database WHERE title = '$title' AND LOGIN = '$login'",null)  return result.moveToFirst()  }  @SuppressLint("Range")  fun getInfoMovie(login: String, title: String):ArrayList<Movie> {  val movie = ArrayList<Movie>()  val db = *writableDatabase* var cursor: Cursor? = null  try {  cursor =  db.rawQuery("select \* from movie\_database where login = '$login' and title = '$title'",  null)  } catch (e: SQLiteException) {  return ArrayList()  }  if (cursor!!.moveToFirst()) {  while (cursor.*isAfterLast* == false) {  var login =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.LOGIN)).toString()  var title =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.TITLE)).toString()  var year =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.YEAR)).toString()  var image =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.IMAGE\_POSTER))  .toString()  var type =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.TYPE)).toString()  var released =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.RELEASED))  .toString()  var actors =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.ACTORS)).toString()  var plot =  cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.PLOT)).toString()  movie.add(Movie(login, title, year, image, type, released, actors, plot))  cursor.moveToNext()  }  }  return movie  }  @SuppressLint("Range")  fun readAllMovie(login: String): ArrayList<Movie> {  val movie = ArrayList<Movie>()  val db = *writableDatabase* var cursor: Cursor? = null  try {  cursor = db.rawQuery("select \* from movie\_database where login = '$login'", null)  } catch (e: SQLiteException) {  return ArrayList()  }  if (cursor!!.moveToFirst()) {  while (cursor.*isAfterLast* == false) {  var login = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.LOGIN)).toString()  var title = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.TITLE)).toString()  var year = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.YEAR)).toString()  var image = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.IMAGE\_POSTER)).toString()  var type = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.TYPE)).toString()  var released = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.RELEASED)).toString()  var actors = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.ACTORS)).toString()  var plot = cursor.getString(cursor.getColumnIndex(DBContract.MovieEntry.PLOT)).toString()  movie.add(Movie(login, title, year, image, type, released, actors, plot))  cursor.moveToNext()  }  }  return movie  }   @Throws(SQLiteConstraintException::class)  fun deleteMovie(movie\_title: String): Boolean {  val db = *writableDatabase* val selection = DBContract.MovieEntry.TITLE + " LIKE ?"  val selectionArgs = *arrayOf*(movie\_title)  db.delete(DBContract.MovieEntry.TABLE\_NAME, selection, selectionArgs)  return true  }   fun deleteAllMovie(): Boolean {  val db = *writableDatabase* db.delete("movie\_database", null, null)  return true  }   fun editMovieInfo(year: String, actors: String, plot: String, type: String, title:String, login:String): Boolean  {  val db = *writableDatabase* val result = db.rawQuery("UPDATE movie\_database SET year='$year', actors='$actors', plot='$plot', type='$type' WHERE title = '$title' AND LOGIN = '$login'",null)  return result.moveToFirst()  } } |

|  |
| --- |
| package com.example.prakt22\_2\_  import android.provider.BaseColumns  object DBContract {   class MovieEntry : BaseColumns {  companion object {  val TABLE\_NAME = "movie\_database"  val LOGIN = "login"  val TITLE = "title"  val YEAR = "year"  val IMAGE\_POSTER = "image"  val TYPE = "type"  val RELEASED = "released"  val ACTORS = "actors"  val PLOT = "plot"  }  } } |

|  |
| --- |
| package com.example.prakt22\_2\_  import android.content.ContentValues import android.content.Context import android.database.Cursor import android.database.sqlite.SQLiteConstraintException import android.database.sqlite.SQLiteDatabase import android.database.sqlite.SQLiteException import android.database.sqlite.SQLiteOpenHelper import androidx.room.Room  class DBHelper (val context: Context, val factory: SQLiteDatabase.CursorFactory?):SQLiteOpenHelper ///знак ? разрешает null  (context,"users", factory, 1) {  override fun onCreate(db: SQLiteDatabase?) {  val quary = "CREATE TABLE users (id INT PRIMARY KEY, login TEXT, password TEXT)"  db!!.execSQL(quary) /// !! необходимо для обработки возможного значения null  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db!!.execSQL("DROP TABLE IF EXISTS users")  onCreate(db)  }  fun addUser(user: User)  {  val values = ContentValues()  values.put("login", user.login)  values.put("password", user.password)  val db = this.*writableDatabase* db.insert("users", null, values)  db.close()  }  fun getUser(login: String, password: String):Boolean //в функцию передается пароль и логин  {  val db = this.*readableDatabase* // поиск через запрос  val result = db.rawQuery("SELECT \* FROM users WHERE login = '$login' and password = '$password'",null)  //выводим true или false  return result.moveToFirst()  }  fun checkUser(login: String):Boolean  {  val db = this.*readableDatabase* val result = db.rawQuery("SELECT \* FROM users WHERE login = '$login'",null)  return result.moveToFirst()  } } |

|  |
| --- |
|  |
| package com.example.prakt22\_2\_  import android.content.ContentValues import android.content.Context import android.database.Cursor import android.database.sqlite.SQLiteConstraintException import android.database.sqlite.SQLiteDatabase import android.database.sqlite.SQLiteException import android.database.sqlite.SQLiteOpenHelper import androidx.room.Room  class DBHelper (val context: Context, val factory: SQLiteDatabase.CursorFactory?):SQLiteOpenHelper ///знак ? разрешает null  (context,"users", factory, 1) {  override fun onCreate(db: SQLiteDatabase?) {  val quary = "CREATE TABLE users (id INT PRIMARY KEY, login TEXT, password TEXT)"  db!!.execSQL(quary) /// !! необходимо для обработки возможного значения null  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db!!.execSQL("DROP TABLE IF EXISTS users")  onCreate(db)  }  fun addUser(user: User)  {  val values = ContentValues()  values.put("login", user.login)  values.put("password", user.password)  val db = this.*writableDatabase* db.insert("users", null, values)  db.close()  }  fun getUser(login: String, password: String):Boolean //в функцию передается пароль и логин  {  val db = this.*readableDatabase* // поиск через запрос  val result = db.rawQuery("SELECT \* FROM users WHERE login = '$login' and password = '$password'",null)  //выводим true или false  return result.moveToFirst()  }  fun checkUser(login: String):Boolean  {  val db = this.*readableDatabase* val result = db.rawQuery("SELECT \* FROM users WHERE login = '$login'",null)  return result.moveToFirst()  } } |
| <?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:layout\_width="match\_parent"  android:orientation="vertical"  android:layout\_height="match\_parent"  tools:context=".EditPage">  <ImageView  android:layout\_width="80dp"  android:layout\_height="80dp"  android:layout\_marginTop="10dp"  android:layout\_gravity="center"  android:src="@drawable/icon\_1">  </ImageView>   <TextView  android:id="@+id/title"  android:layout\_width="match\_parent"  android:layout\_height="41dp"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="8dp"  android:layout\_marginRight="10dp"  android:textColor="@color/gray"  android:textSize="20dp"></TextView>  <TextView  android:id="@+id/text\_year"  android:layout\_width="match\_parent"  android:layout\_height="41dp"  android:layout\_marginLeft="10dp"  android:text="@string/title\_year"  android:layout\_marginTop="8dp"  android:layout\_marginRight="10dp"  android:textColor="@color/gray"  android:textSize="20dp"></TextView>  <EditText  android:id="@+id/edittext\_year"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789"  android:inputType="text"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <TextView  android:id="@+id/text\_type"  android:layout\_width="match\_parent"  android:layout\_height="41dp"  android:layout\_marginLeft="10dp"  android:text="@string/title\_type"  android:layout\_marginTop="8dp"  android:layout\_marginRight="10dp"  android:textColor="@color/gray"  android:textSize="20dp"></TextView>  <EditText  android:id="@+id/edittext\_type"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789"  android:inputType="text"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <TextView  android:id="@+id/text\_actors"  android:layout\_width="match\_parent"  android:layout\_height="41dp"  android:layout\_marginLeft="10dp"  android:text="@string/title\_actors"  android:layout\_marginTop="8dp"  android:layout\_marginRight="10dp"  android:textColor="@color/gray"  android:textSize="20dp"></TextView>  <EditText  android:id="@+id/edittext\_actors"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789"  android:inputType="text"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <TextView  android:id="@+id/text\_plot"  android:layout\_width="match\_parent"  android:layout\_height="41dp"  android:layout\_marginLeft="10dp"  android:text="@string/title\_plot"  android:layout\_marginTop="8dp"  android:layout\_marginRight="10dp"  android:textColor="@color/gray"  android:textSize="20dp"></TextView>  <EditText  android:id="@+id/edittext\_plot"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789"  android:inputType="text"  android:paddingLeft="10dp"  android:background="@drawable/background\_view"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp">  </EditText>  <Button  android:id="@+id/button\_edit"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:onClick="delAllItems"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:textSize="18dp"  android:textColor="@color/blue"  android:layout\_marginTop="10dp"  android:textAllCaps="false"  android:background="@drawable/background\_view"  android:text="@string/save"></Button> </LinearLayout> |