|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** - **Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5** | |
| **по дисциплине** | |
| «Разработка программных приложений» | |
|  | |
| Выполнил студент группы ИКБО-02-18 | Юркевич Г.А. |
| Принял ассистент кафедры ИППО | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_2020 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. |  |

Москва 2020

**Содержание**

[1. Цель лабораторной работы 3](#_heading=h.26in1rg)

2. Общие требования [3](#_heading=h.2et92p0)

3. Задание [3](#_heading=h.3dy6vkm)

4. Поэтапное выполнение лабораторной работы [4](#_heading=h.4d34og8)

5. Результат работы [4](#_heading=h.lnxbz9)

6.Вывод [4](#_heading=h.17dp8vu)

[Список использованных источников 9](#_heading=h.35nkun2)

1. Цель лабораторной работы

Целью данной лабораторной работы является ознакомление со средой разработки программных приложений Android Studio, а также создание приложения моделирующего e-commerce решение.

**2. Общие требования**

Среда разработки по умолчанию - Android Studio. Язык программирования - Java.

По своему желанию, можно использовать Kotlin, а также XCode/Swift при наличии соответствующего оборудования.

Решения должны быть размещены в виде проекта в git репозитории (github/bitbucket).

Ссылку на git-репозиторий и отчет по выполненной работе прислать на почту: strogankova-mirea@yandex.ru.

Минимальная версия API – 19/

**3. Задание**

Необходимо создать приложение моделирующее e-commerce решение. Приложение должно состоять из двух частей располагающихся на разных экранах/активити:

● store-front — клиент, где осуществляются просмотр и покупка товаров  
● back-end — административная часть, позволяющая добавлять новые товары или редактировать существующие

Информация о товаре должна содержать наименование, стоимость и количество (единиц в наличии).  
Store-front предоставляет возможности просмотра полной информации об одной товарной позиции и перехода к следующей или предыдущей позиции. Товар отображается только в том случае если его количество больше нуля. Переход от одного товара к другому происходит с анимированным сдвигом экрана влево/вправо при горизонтальном «перелистывании».  
Back-end предоставляет возможности просмотра, редактирования и добавления продуктов. В любой момент пользователь может переходить от одного экрана к другому.

**4. Поэтапное выполнение лабораторной работы**

Создается первое активити Store-front на котором размещен ViewPager, после чего создается адаптер для ViewPager, а затем и второе активити back-end. Для хранения данных о товаре создается база данных взаимодействие (добавление и изменение записей в базе данных) с которой осуществляется через back-end. Записи базы данных в которых значение в колонке количество равно 0 не отображаются в ViewPager, остальные отображаются.

**5. Результат работы**

**MainActivity**

package com.example.ecomerce6;

import androidx.appcompat.app.AppCompatActivity;

import androidx.viewpager.widget.ViewPager;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.FrameLayout;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

private ViewPager pager = null;

private ViewAdapter pagerAdapter = null;

public DBHelper DBH;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

pager=(ViewPager)findViewById(R.id.pager);

pagerAdapter = new ViewAdapter();

pager.setAdapter (pagerAdapter);

DBH= new DBHelper(this);

final SQLiteDatabase database = DBH.getWritableDatabase();

// Create an initial view to display; must be a subclass of FrameLayout.

LayoutInflater inflater = getLayoutInflater();

FrameLayout v1 = (FrameLayout) inflater.inflate (R.layout.fragment\_page, null);

Cursor cursor = database.query(DBHelper.TABLE\_CONTACTS, null, null, null, null, null, null);

if (cursor.moveToFirst()) {

int idIndex = cursor.getColumnIndex(DBH.KEY\_ID);

int nameIndex1 = cursor.getColumnIndex(DBH.COST);

int nameIndex2 = cursor.getColumnIndex(DBH.QUANTITY);

int nameIndex3 = cursor.getColumnIndex(DBH.NAME);

do {

if (cursor.getInt(nameIndex2)!=0)

{FrameLayout v0 = (FrameLayout) inflater.inflate (R.layout.fragment\_page, null);

TextView tvName = (TextView) v0.findViewById(R.id.displayText);

tvName.setText(cursor.getString(nameIndex3)+ "\n"+cursor.getString(nameIndex1)+ "\n"+cursor.getString(nameIndex2));

pagerAdapter.addView (v0, 0);

pagerAdapter.notifyDataSetChanged();}

} while (cursor.moveToNext());

} else

Log.d("mLog","0 rows");

cursor.close();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_settings:

Intent intent = new Intent(this,BackFront.class);

startActivity(intent);

this.finish();

// Toast.makeText(getApplicationContext(),

// "You selected Settings", Toast.LENGTH\_LONG).show();

// return true;

}

return super.onOptionsItemSelected(item);

}

}

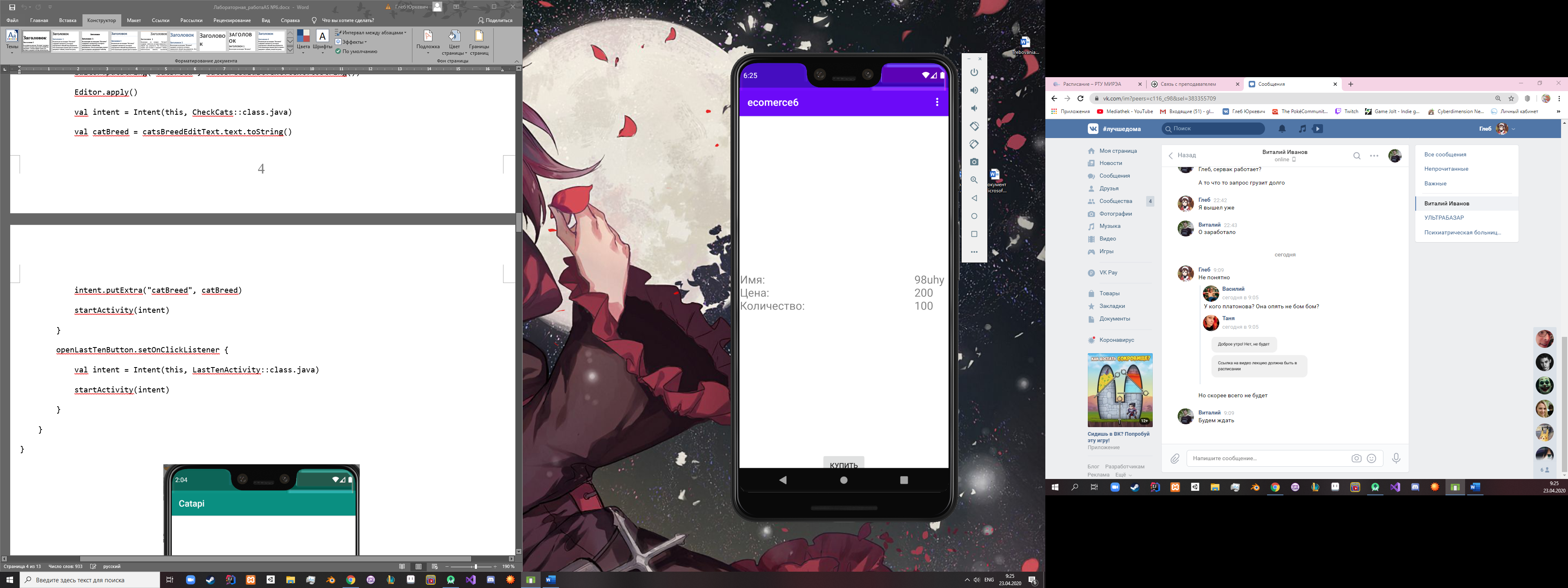


Рисунок 1. Store-front

**DBHelper**

package com.example.ecomerce6;

import android.content.Context;

import android.database.sqlite.SQLiteOpenHelper;

import android.database.sqlite.SQLiteDatabase;

public class DBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 1;

public static final String DATABASE\_NAME = "DBitem";

public static final String TABLE\_CONTACTS = "Item";

public static final String KEY\_ID = "id";

public static final String COST = "Cost";

public static final String QUANTITY = "Quantity";

public static final String NAME = "Name";

public DBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + TABLE\_CONTACTS + "(" + KEY\_ID + " integer primary key," + COST + " text," + QUANTITY + " text,"+ NAME + " text" + ")");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("drop table if exists " + TABLE\_CONTACTS);

onCreate(db);

}

public void getcost(SQLiteDatabase db,int id)

{

db.execSQL("SELECT MAX(" +KEY\_ID +") FROM"+TABLE\_CONTACTS);

}

}

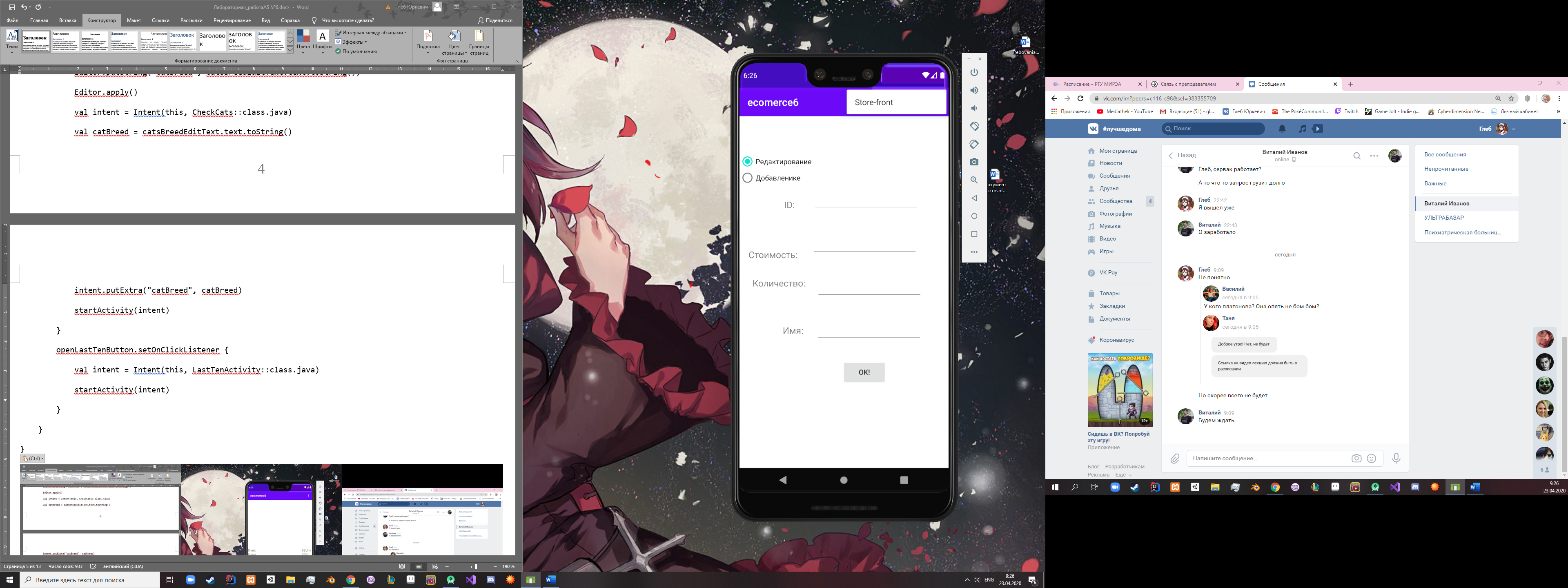


Рисунок 2. Back-end

**ViewAdapter**

package com.example.ecomerce6;

import android.view.View;

import android.view.ViewGroup;

import androidx.viewpager.widget.PagerAdapter;

import androidx.viewpager.widget.ViewPager;

import java.util.ArrayList;

public class ViewAdapter extends PagerAdapter

{

private ArrayList<View> views = new ArrayList<View>();

@Override

public int getItemPosition (Object object)

{

int index = views.indexOf (object);

if (index == -1)

return POSITION\_NONE;

else

return index;

}

@Override

public Object instantiateItem (ViewGroup container, int position)

{

View v = views.get (position);

container.addView (v);

return v;

}

@Override

public void destroyItem (ViewGroup container, int position, Object object)

{

container.removeView (views.get (position));

}

@Override

public int getCount ()

{

return views.size();

}

@Override

public boolean isViewFromObject (View view, Object object)

{

return view == object;

}

public int addView (View v)

{

return addView (v, views.size());

}

public int addView (View v, int position)

{

views.add (position, v);

return position;

}

public int removeView (ViewPager pager, View v)

{

return removeView (pager, views.indexOf (v));

}

public int removeView (ViewPager pager, int position)

{

pager.setAdapter (null);

views.remove (position);

pager.setAdapter (this);

return position;

}

public View getView (int position)

{

return views.get (position);

}

}

**BackFront**

package com.example.ecomerce6;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;

import android.content.Intent;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.TextView;

public class BackFront extends AppCompatActivity {

Button button1;

EditText editText1;

EditText editText2;

EditText editText3;

EditText editText4;

RadioButton radioButtonAdd;

RadioButton radioButtonRed;

TextView textView;

DBHelper DBH;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

DBH= new DBHelper(this);

final SQLiteDatabase database = DBH.getWritableDatabase();

final ContentValues contentValues = new ContentValues();

setContentView(R.layout.activity\_back\_front);

editText1=(EditText) findViewById(R.id.editText);

editText2=(EditText) findViewById(R.id.editText2);

editText3=(EditText) findViewById(R.id.editText3);

editText4=(EditText) findViewById(R.id.editText4);

button1=(Button) findViewById(R.id.button);

radioButtonAdd=(RadioButton)findViewById(R.id.radioButtonAdd);

radioButtonRed=(RadioButton)findViewById(R.id.radioButtonRed);

radioButtonRed.setOnClickListener(radioButtonClickListener);

radioButtonAdd.setOnClickListener(radioButtonClickListener);

textView=(TextView) findViewById(R.id.textView);

View.OnClickListener oclBtnOk = new View.OnClickListener() {

@Override

public void onClick(View v) {

ContentValues cv=new ContentValues();

if (radioButtonAdd.isChecked())

{

cv.put(DBHelper.COST,(editText2.getText().toString()));

cv.put(DBHelper.QUANTITY,(editText3.getText().toString()));

cv.put(DBHelper.NAME,(editText4.getText().toString()));

database.insert(DBHelper.TABLE\_CONTACTS,null,cv);

Log.d("mLog","+1");

}

else

{

cv.put(DBHelper.COST,(editText2.getText().toString()));

cv.put(DBHelper.QUANTITY,(editText3.getText().toString()));

cv.put(DBHelper.NAME,(editText4.getText().toString()));

database.update(DBHelper.TABLE\_CONTACTS,cv,"id = ?",new String[] { (editText1.getText().toString())});

Log.d("mLog","re");

}

}

};

button1.setOnClickListener(oclBtnOk);

}

View.OnClickListener radioButtonClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

RadioButton rb = (RadioButton)v;

switch (rb.getId()) {

case R.id.radioButtonAdd: editText1.setVisibility(View.INVISIBLE);textView.setVisibility(View.INVISIBLE);

break;

case R.id.radioButtonRed:editText1.setVisibility(View.VISIBLE);textView.setVisibility(View.VISIBLE);

break;

}

}

};

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_back, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_settings2:

Intent intent2 = new Intent(this,MainActivity.class);

startActivity(intent2);

// Toast.makeText(getApplicationContext(),

// "You selected Settings", Toast.LENGTH\_LONG).show();

// return true;

finish();

}

return super.onOptionsItemSelected(item);

}

}

**5. Вывод**

В данной работе было разработано приложение моделирующее e-commerce решение. Приложение состоит из двух частей располагающихся на разных экранах/активити.

**Список использованных источников**

1. Главный сайт по Android: <https://developer.android.com/>
2. Канал Google на YouTube:

<http://www.youtube.com/channel/UC_x5XG1OV2P6uZZ5FSM9Ttw>

1. Онлайн-курсы по Android:

– Udacity:

<https://eu.udacity.com/course/new-android-fundamentals--ud851>

1. <https://stackoverflow.com/>
2. <https://github.com>
3. <https://google.com/>