**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине: **«Программирование мобильных информационных систем»**

на тему: **«Использование шаблона проектирование MVC в разработке Android приложений»**

Выполнил: студент группы ИТП-31

Пронуза М.Ю.

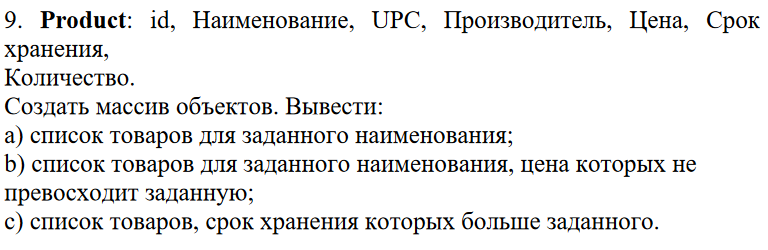
Проверил ст. преподаватель:

Стефановский И.Л.

Гомель 2024

**Цель работы**: изучить архитектуру "Модель-Представление-Контроллер" в Android.

**Задание:**

Разработать Android приложение с использованием архитектуры "Модель-Представление-Контроллер", выполняющее ввод данных, вывод и редактирование в соответствии с вариантом. Для выполнения каждого пункта задания (a-d) использовать отдельную Activity и модель. Выполнить запуск приложения на эмуляторе.

**Ход выполнения и результаты работы:**

На рисунке 1 представлен полученный макет главного экрана.

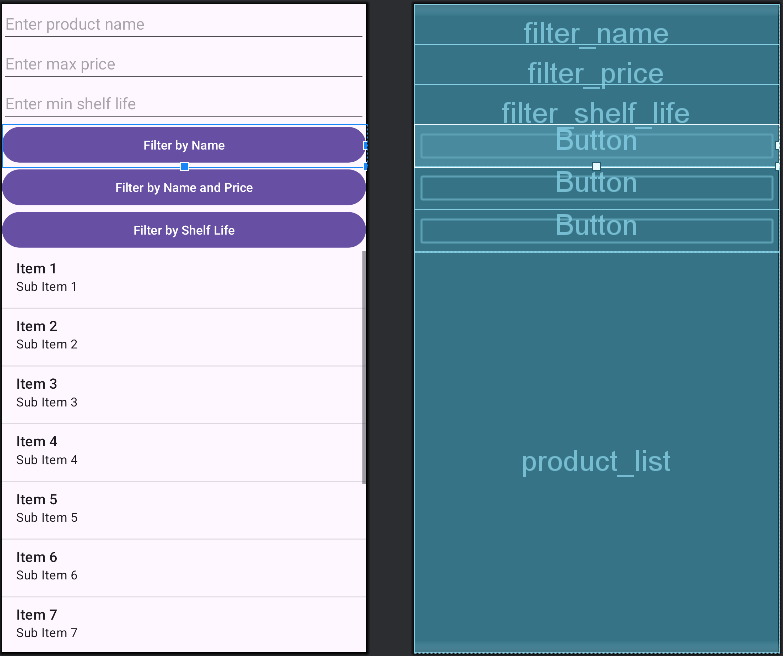


Рисунок 1 – Макет приложения

На рисунке 2 представлен полученный макет экрана редактирования.

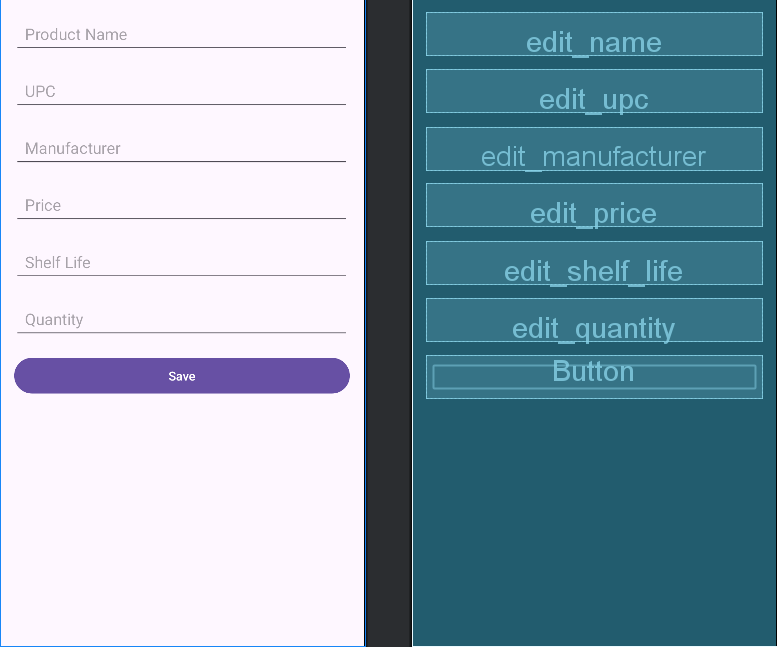


Рисунок 2 – макет экрана редактирования

На рисунке 3 и 4 представлен результат выполнения программы на устройстве Android.

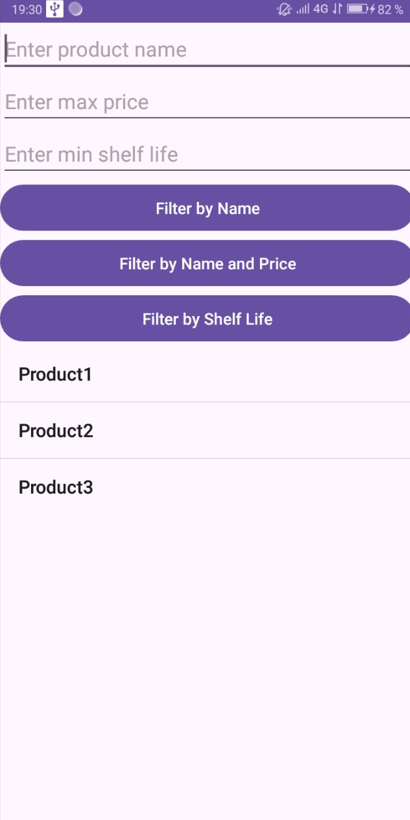


Рисунок 3 – Выполнение программы на устройстве Android

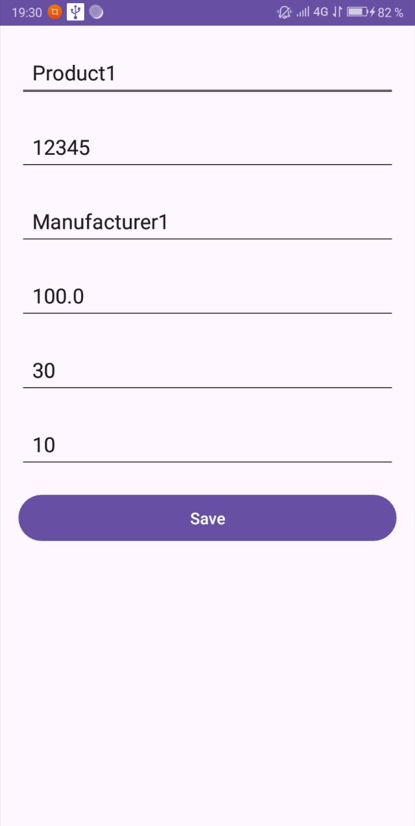


Рисунок 3 – редактирование продукта

Текст *MainActivity.java* представлен в приложении А.

Текст *EditProductActivity.java* представлен в приложении Б.

Текст *Product.java* представлен в приложении В.

Текст *ProductRepository.java* представлен в приложении Г.

**Вывод:** в ходе выполнения лабораторной работы изучена архитектура "Модель-Представление-Контроллер" в Android..

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст *MainActivity***

package com.example.lab;

import android.os.Bundle;

import android.util.Log;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.graphics.Insets;

import androidx.core.view.ViewCompat;

import androidx.core.view.WindowInsetsCompat;

import android.content.Intent;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.EditText;

import android.widget.ListView;

import java.util.List;

public class MainActivity extends AppCompatActivity {

private ProductRepository productRepository;

private ListView listView;

private EditText filterName, filterPrice, filterShelfLife;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

productRepository = new ProductRepository();

listView = findViewById(R.id.product\_list);

filterName = findViewById(R.id.filter\_name);

filterPrice = findViewById(R.id.filter\_price);

filterShelfLife = findViewById(R.id.filter\_shelf\_life);

// Отображаем все продукты при запуске

showProducts(productRepository.products);

// Обрабатываем нажатие на элемент списка

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

Product selectedProduct = (Product) listView.getAdapter().getItem(position);

Log.e("MainActivity", "Product hash: " + selectedProduct.hashCode());

Intent intent = new Intent(MainActivity.this, EditProductActivity.class);

// Передаем сам объект продукта

intent.putExtra("selectedProduct", selectedProduct);

startActivityForResult(intent, 1); // Запускаем активность с ожиданием результата

}

});

}

// Метод отображения продуктов в ListView

private void showProducts(List<Product> products) {

ArrayAdapter<Product> adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, products);

listView.setAdapter(adapter);

}

// Обработка фильтрации по наименованию

public void filterByName(View view) {

String name = filterName.getText().toString();

showProducts(productRepository.getProductsByName(name));

}

// Обработка фильтрации по наименованию и цене

public void filterByNameAndPrice(View view) {

String name = filterName.getText().toString();

double price = Double.parseDouble(filterPrice.getText().toString());

showProducts(productRepository.getProductsByNameAndPrice(name, price));

}

// Обработка фильтрации по сроку хранения

public void filterByShelfLife(View view) {

int shelfLife = Integer.parseInt(filterShelfLife.getText().toString());

showProducts(productRepository.getProductsByShelfLife(shelfLife));

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == 1 && resultCode == RESULT\_OK) {

// Получаем обновленный продукт

Product updatedProduct = (Product) data.getSerializableExtra("updatedProduct");

if (updatedProduct != null) {

// Обновляем продукт в репозитории

productRepository.updateProduct(updatedProduct);

showProducts(productRepository.products);

}

}

}

}

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Текст *EditProductActivity***

package com.example.lab;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class EditProductActivity extends AppCompatActivity {

private Product selectedProduct;

private EditText nameField, upcField, manufacturerField, priceField, shelfLifeField, quantityField;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_product);

// Получаем переданный продукт

selectedProduct = (Product) getIntent().getSerializableExtra("selectedProduct");

Log.e("EditProductActivity", "Editing Product: " + selectedProduct);

// Связывание полей с элементами интерфейса

nameField = findViewById(R.id.edit\_name);

upcField = findViewById(R.id.edit\_upc);

manufacturerField = findViewById(R.id.edit\_manufacturer);

priceField = findViewById(R.id.edit\_price);

shelfLifeField = findViewById(R.id.edit\_shelf\_life);

quantityField = findViewById(R.id.edit\_quantity);

// Заполнение полей данными о продукте

if (selectedProduct != null) {

nameField.setText(selectedProduct.name);

upcField.setText(selectedProduct.upc);

manufacturerField.setText(selectedProduct.manufacturer);

priceField.setText(String.valueOf(selectedProduct.price));

shelfLifeField.setText(String.valueOf(selectedProduct.shelfLife));

quantityField.setText(String.valueOf(selectedProduct.quantity));

}

}

public void saveProduct(View view) {

if (selectedProduct != null) {

// Сохраняем изменения

selectedProduct.name = nameField.getText().toString();

selectedProduct.upc = upcField.getText().toString();

selectedProduct.manufacturer = manufacturerField.getText().toString();

selectedProduct.price = Double.parseDouble(priceField.getText().toString());

selectedProduct.shelfLife = Integer.parseInt(shelfLifeField.getText().toString());

selectedProduct.quantity = Integer.parseInt(quantityField.getText().toString());

// Возвращаем обновленный продукт в MainActivity

Intent resultIntent = new Intent();

resultIntent.putExtra("updatedProduct", selectedProduct);

setResult(RESULT\_OK, resultIntent); // Возвращаем результат OK

Log.e("EditProductActivity", "Product hash: " + selectedProduct.hashCode());

}

finish(); // Закрываем активность

}

}

**ПРИЛОЖЕНИЕ В**

(обязательное)

**Текст *Product***

package com.example.lab;

import java.io.Serializable;

public class Product implements Serializable {

public int id;

public String name;

public String upc;

public String manufacturer;

public double price;

public int shelfLife;

public int quantity;

public Product(int id, String name, String upc, String manufacturer, double price, int shelfLife, int quantity) {

this.id = id;

this.name = name;

this.upc = upc;

this.manufacturer = manufacturer;

this.price = price;

this.shelfLife = shelfLife;

this.quantity = quantity;

}

@Override

public String toString() {

return name;

}

}

**ПРИЛОЖЕНИЕ Г**

(обязательное)

**Текст *ProductRepository***

package com.example.lab;

import java.util.ArrayList;

import java.util.List;

public class ProductRepository {

public List<Product> products;

public ProductRepository() {

products = new ArrayList<>();

products.add(new Product(1, "Product1", "12345", "Manufacturer1", 100.0, 30, 10));

products.add(new Product(2, "Product2", "67890", "Manufacturer2", 150.0, 60, 5));

products.add(new Product(3, "Product3", "54321", "Manufacturer1", 200.0, 20, 2));

}

// Фильтрация по наименованию

public List<Product> getProductsByName(String name) {

List<Product> result = new ArrayList<>();

for (Product product : products) {

if (product.name.equalsIgnoreCase(name)) {

result.add(product);

}

}

return result;

}

// Фильтрация по наименованию и цене

public List<Product> getProductsByNameAndPrice(String name, double maxPrice) {

List<Product> result = new ArrayList<>();

for (Product product : products) {

if (product.name.equalsIgnoreCase(name) && product.price <= maxPrice) {

result.add(product);

}

}

return result;

}

// Фильтрация по сроку хранения

public List<Product> getProductsByShelfLife(int minShelfLife) {

List<Product> result = new ArrayList<>();

for (Product product : products) {

if (product.shelfLife > minShelfLife) {

result.add(product);

}

}

return result;

}

public Product findProductById(int id) {

for (Product product : products) {

if (product.id == id) {

return product;

}

}

return null;

}

// Обновление продукта

public void updateProduct(Product updatedProduct) {

for (int i = 0; i < products.size(); i++) {

if (products.get(i).id == updatedProduct.id) {

products.set(i, updatedProduct);

break;

}

}

}

}