**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное

учреждение высшего образования

**Национальный исследовательский**

**Томский политехнический университет**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Отчет по лабораторной работе №15 по дисциплине

**«Язык Kotlin и основы разработки»**

Диалоги

****

Выполнил:

Студент группы 1А22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.К. Кравцов

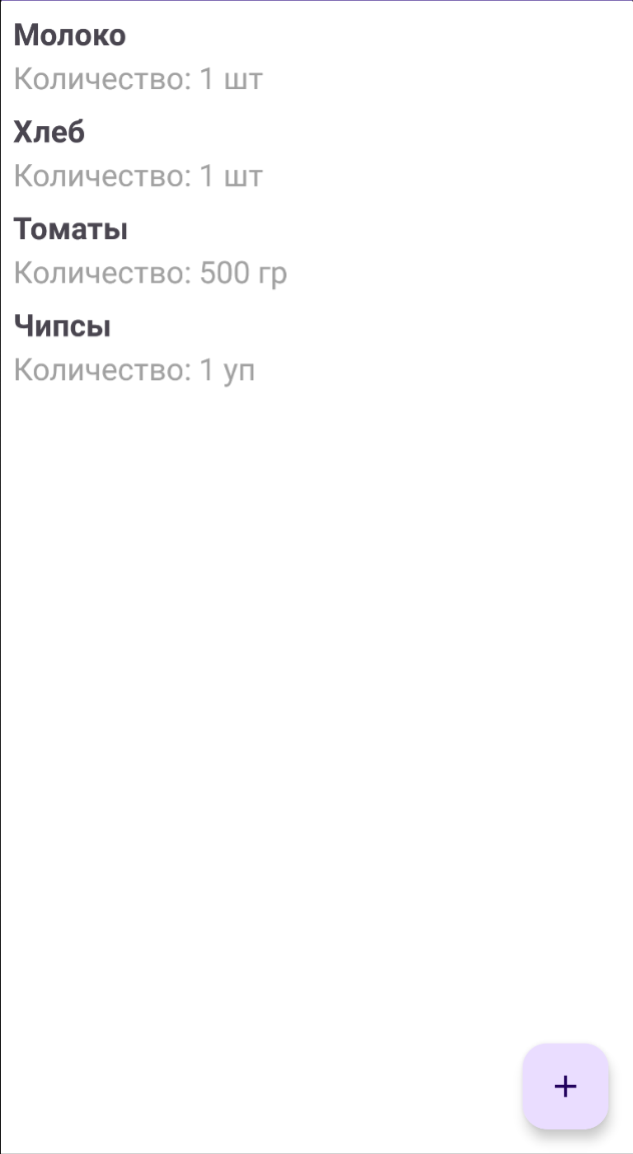
Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

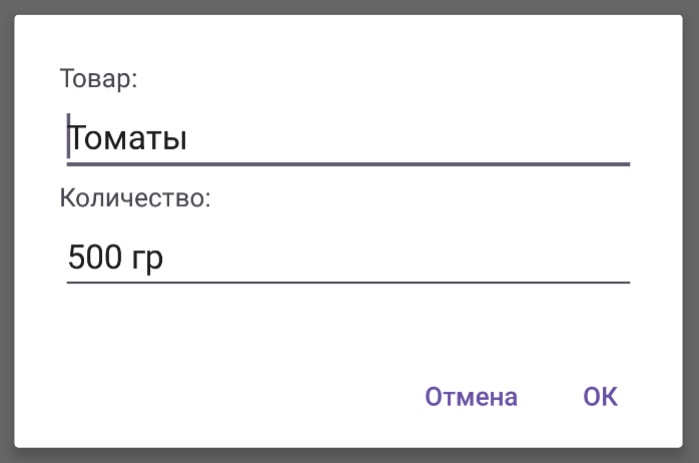
Томск 2025

# Задание

Создайте приложение для ведения списка покупок. В основе должен лежать RecyclerView, который отображает список примерно в следующем виде:



При нажатии плавающей кнопки должен появляться диалог с пустыми полями, а при его положительном завершении – в список должна добавляться новая покупка:



При касании записи она должна открываться на редактирование: появляется такой же диалог, но уже с заполненными полями, а при положительном завершении – этот элемент в списке обновляется.

При жесте смахивания на какой-либо записи (проведении по ней пальцем влево или вправо) она должна удаляться из списка.

# Ход работы

1. Создан проект Lab15 на основе Empty Views Activity.
2. Настроены ресурсы приложения:

* dimens.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <!-- Отступы -->  
 <dimen name="margin\_small">8dp</dimen>  
 <dimen name="margin\_medium">16dp</dimen>  
  
 <!-- Размеры текста -->  
 <dimen name="text\_size\_small">14sp</dimen>  
 <dimen name="text\_size\_medium">16sp</dimen>  
 <dimen name="text\_size\_large">18sp</dimen>  
  
 <!-- Радиус скругления -->  
 <dimen name="corner\_radius">8dp</dimen>  
  
 <!-- Высота тени -->  
 <dimen name="elevation">4dp</dimen>  
</resources>

* strings.xml

<resources>  
 <string name="app\_name">Lab15</string>  
  
 <!-- Диалог -->  
 <string name="dialog\_title\_add">Добавить товар</string>  
 <string name="dialog\_title\_edit">Редактировать товар</string>  
 <string name="dialog\_product\_label">Товар:</string>  
 <string name="dialog\_quantity\_label">Количество:</string>  
 <string name="dialog\_product\_hint">Введите название товара</string>  
 <string name="dialog\_quantity\_hint">Например: 1 шт или 500 гр</string>  
 <string name="dialog\_button\_ok">OK</string>  
 <string name="dialog\_button\_cancel">Отмена</string>  
  
 <!-- Элемент списка -->  
 <string name="item\_quantity\_format">Количество: %s</string>  
  
 <!-- FAB -->  
 <string name="fab\_content\_description">Добавить новый товар</string>  
</resources>

1. Реализована модель данных:

* Purchase.kt

package ru.olegkravtsov.lab15  
  
import java.io.Serializable  
  
data class Purchase(  
 val id: Long = System.currentTimeMillis(),  
 var name: String,  
 var quantity: String  
) : Serializable

1. Создана основная активность MainActivity:

package ru.olegkravtsov.lab15  
  
import android.os.Bundle  
import androidx.activity.enableEdgeToEdge  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.view.ViewCompat  
import androidx.core.view.WindowInsetsCompat  
import androidx.recyclerview.widget.ItemTouchHelper  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.google.android.material.floatingactionbutton.FloatingActionButton  
  
class MainActivity : AppCompatActivity(), PurchaseDialogFragment.PurchaseDialogListener {  
  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var fabAdd: FloatingActionButton  
 private lateinit var adapter: PurchaseAdapter  
  
 private val purchases = mutableListOf<Purchase>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 enableEdgeToEdge()  
 setContentView(R.layout.activity\_main)  
  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
 val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 }  
  
 initViews()  
 setupRecyclerView()  
 setupSwipeToDelete()  
 }  
  
 private fun initViews() {  
 recyclerView = findViewById(R.id.recyclerView)  
 fabAdd = findViewById(R.id.fabAdd)  
  
 fabAdd.setOnClickListener {  
 showAddPurchaseDialog()  
 }  
 }  
  
 private fun setupRecyclerView() {  
 adapter = PurchaseAdapter(  
 purchases = purchases,  
 onItemClick = { purchase ->  
 showEditPurchaseDialog(purchase)  
 }  
 )  
  
 recyclerView.layoutManager = LinearLayoutManager(this)  
 recyclerView.adapter = adapter  
 }  
  
 private fun setupSwipeToDelete() {  
 val swipeCallback = object : ItemTouchHelper.SimpleCallback(  
 0,  
 ItemTouchHelper.LEFT or ItemTouchHelper.RIGHT  
 ) {  
 override fun onMove(  
 recyclerView: RecyclerView,  
 viewHolder: RecyclerView.ViewHolder,  
 target: RecyclerView.ViewHolder  
 ): Boolean = false  
  
 override fun onSwiped(viewHolder: RecyclerView.ViewHolder, direction: Int) {  
 val position = viewHolder.absoluteAdapterPosition  
 adapter.removePurchase(position)  
 }  
 }  
  
 val itemTouchHelper = ItemTouchHelper(swipeCallback)  
 itemTouchHelper.attachToRecyclerView(recyclerView)  
 }  
  
 private fun showAddPurchaseDialog() {  
 val dialog = PurchaseDialogFragment.newInstance()  
 dialog.setListener(this)  
 dialog.show(supportFragmentManager, "add\_purchase")  
 }  
  
 private fun showEditPurchaseDialog(purchase: Purchase) {  
 val dialog = PurchaseDialogFragment.newInstance(purchase)  
 dialog.setListener(this)  
 dialog.show(supportFragmentManager, "edit\_purchase")  
 }  
  
 override fun onPurchaseAdded(purchase: Purchase) {  
 adapter.addPurchase(purchase)  
 }  
  
 override fun onPurchaseUpdated(purchase: Purchase) {  
 val position = adapter.getPurchasePosition(purchase)  
 if (position != -1) {  
 adapter.updatePurchase(position, purchase)  
 }  
 }  
}

1. Реализован адаптер PurchaseAdapter для отображения списка покупок:

package ru.olegkravtsov.lab15  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
  
class PurchaseAdapter(  
 private var purchases: MutableList<Purchase>,  
 private val onItemClick: (Purchase) -> Unit  
) : RecyclerView.Adapter<PurchaseAdapter.PurchaseViewHolder>() {  
  
 class PurchaseViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val tvName: TextView = itemView.findViewById(R.id.tvName)  
 val tvQuantity: TextView = itemView.findViewById(R.id.tvQuantity)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PurchaseViewHolder {  
 val view = LayoutInflater.from(parent.context)  
 .inflate(R.layout.item\_purchase, parent, false)  
 return PurchaseViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: PurchaseViewHolder, position: Int) {  
 val purchase = purchases[position]  
 holder.tvName.text = purchase.name  
 holder.tvQuantity.text = holder.itemView.context.getString(R.string.item\_quantity\_format, purchase.quantity)  
  
 holder.itemView.setOnClickListener {  
 onItemClick(purchase)  
 }  
 }  
  
 override fun getItemCount(): Int = purchases.size  
  
 fun addPurchase(purchase: Purchase) {  
 purchases.add(purchase)  
 notifyItemInserted(purchases.size - 1)  
 }  
  
 fun updatePurchase(position: Int, purchase: Purchase) {  
 purchases[position] = purchase  
 notifyItemChanged(position)  
 }  
  
 fun removePurchase(position: Int) {  
 purchases.removeAt(position)  
 notifyItemRemoved(position)  
 }  
  
 fun getPurchasePosition(purchase: Purchase): Int {  
 return purchases.indexOfFirst { it.id == purchase.id }  
 }  
}

1. Создан диалог PurchaseDialogFragment для добавления/редактирования:

package ru.olegkravtsov.lab15  
  
import android.app.AlertDialog  
import android.app.Dialog  
import android.os.Bundle  
import android.widget.EditText  
import androidx.fragment.app.DialogFragment  
  
class PurchaseDialogFragment : DialogFragment() {  
  
 interface PurchaseDialogListener {  
 fun onPurchaseAdded(purchase: Purchase)  
 fun onPurchaseUpdated(purchase: Purchase)  
 }  
  
 companion object {  
 private const val ARG\_PURCHASE = "purchase"  
  
 fun newInstance(purchase: Purchase? = null): PurchaseDialogFragment {  
 val args = Bundle().apply {  
 putSerializable(ARG\_PURCHASE, purchase)  
 }  
 return PurchaseDialogFragment().apply {  
 arguments = args  
 }  
 }  
 }  
  
 private var listener: PurchaseDialogListener? = null  
 private var existingPurchase: Purchase? = null  
  
 override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {  
 existingPurchase = arguments?.getSerializable(ARG\_PURCHASE, Purchase::class.java)  
  
 val builder = AlertDialog.Builder(requireContext())  
 val inflater = requireActivity().layoutInflater  
 val view = inflater.inflate(R.layout.dialog\_purchase, null)  
  
 val etName = view.findViewById<EditText>(R.id.etName)  
 val etQuantity = view.findViewById<EditText>(R.id.etQuantity)  
  
 // Если передан существующий товар - заполняем поля  
 existingPurchase?.let {  
 etName.setText(it.name)  
 etQuantity.setText(it.quantity)  
 }  
  
 val title = if (existingPurchase == null) {  
 R.string.dialog\_title\_add  
 } else {  
 R.string.dialog\_title\_edit  
 }  
  
 builder.setView(view)  
 .setTitle(title)  
 .setPositiveButton(R.string.dialog\_button\_ok) { dialog, which ->  
 val name = etName.text.toString().trim()  
 val quantity = etQuantity.text.toString().trim()  
  
 if (name.isNotEmpty() && quantity.isNotEmpty()) {  
 if (existingPurchase == null) {  
 // Добавление нового товара  
 listener?.onPurchaseAdded(Purchase(name = name, quantity = quantity))  
 } else {  
 // Обновление существующего товара  
 val updatedPurchase = existingPurchase!!.copy(  
 name = name,  
 quantity = quantity  
 )  
 listener?.onPurchaseUpdated(updatedPurchase)  
 }  
 }  
 }  
 .setNegativeButton(R.string.dialog\_button\_cancel) { dialog, which ->  
 dialog.dismiss()  
 }  
  
 return builder.create()  
 }  
  
 fun setListener(listener: PurchaseDialogListener) {  
 this.listener = listener  
 }  
}

1. Настроены макеты:

* activity\_main.xml - основной layout с RecyclerView и FAB

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/fabAdd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|end"  
 android:layout\_margin="@dimen/margin\_medium"  
 android:contentDescription="@string/fab\_content\_description"  
 android:src="@android:drawable/ic\_input\_add" />  
  
</FrameLayout>

* item\_purchase.xml - элемент списка на основе CardView

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/margin\_small"  
 app:cardCornerRadius="@dimen/corner\_radius"  
 app:cardElevation="@dimen/elevation">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="@dimen/margin\_medium">  
  
 <TextView  
 android:id="@+id/tvName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="@dimen/text\_size\_large"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/tvQuantity"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="@dimen/margin\_small"  
 android:textSize="@dimen/text\_size\_small" />  
  
 </LinearLayout>  
  
</androidx.cardview.widget.CardView>

* dialog\_purchase.xml - разметка диалогового окна

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="@dimen/margin\_medium">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/dialog\_product\_label"  
 android:textSize="@dimen/text\_size\_medium" />  
  
 <EditText  
 android:id="@+id/etName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="@dimen/margin\_medium"  
 android:hint="@string/dialog\_product\_hint"  
 android:inputType="textCapWords"  
 android:autofillHints="name" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/dialog\_quantity\_label"  
 android:textSize="@dimen/text\_size\_medium" />  
  
 <EditText  
 android:id="@+id/etQuantity"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/dialog\_quantity\_hint"  
 android:inputType="text"  
 android:autofillHints="none" />  
  
</LinearLayout>

**Результат работы**

Приложение успешно запускается и отображает интерфейс списка покупок (рис. 1). При нажатии на плавающую кнопку появляется диалоговое окно для добавления нового товара (рис. 2). После заполнения полей и нажатия кнопки "OK" товар добавляется в список. При нажатии на существующий товар в списке открывается диалоговое окно с заполненными полями для редактирования (рис. 3). После изменения данных и подтверждения товар обновляется в списке. При смахивании товара влево или вправо он удаляется из списка с анимацией (рис. 4).

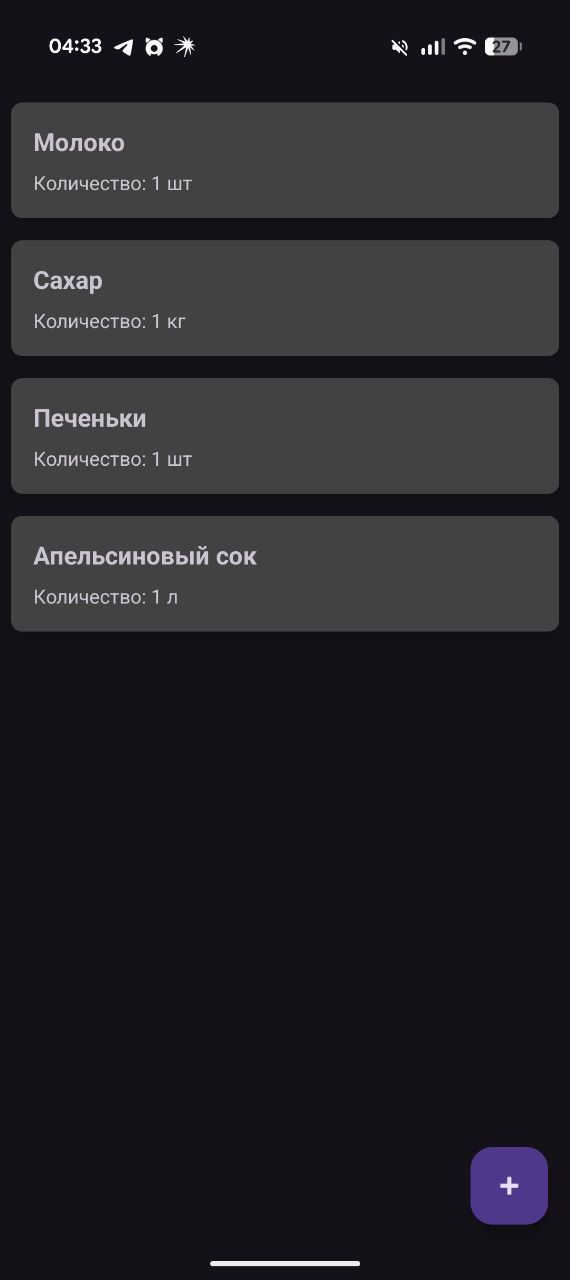


Рисунок 1 – Главный экран приложения с 4 добавленными товарами в список

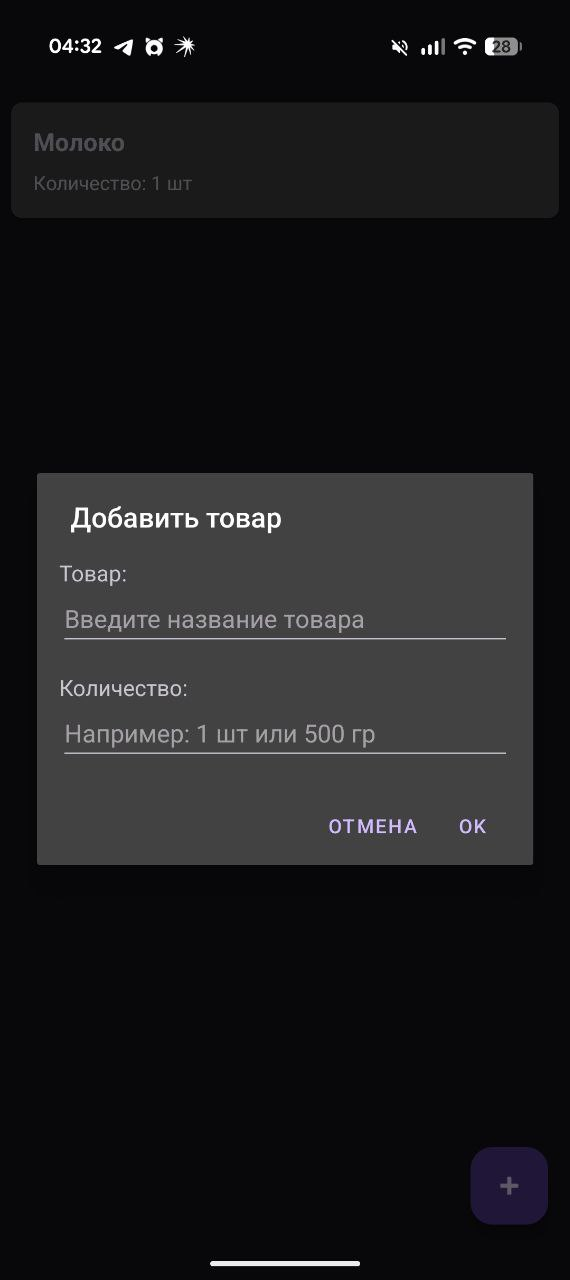


Рисунок 2 – Диалоговое окно добавления нового товара с пустыми полями ввода

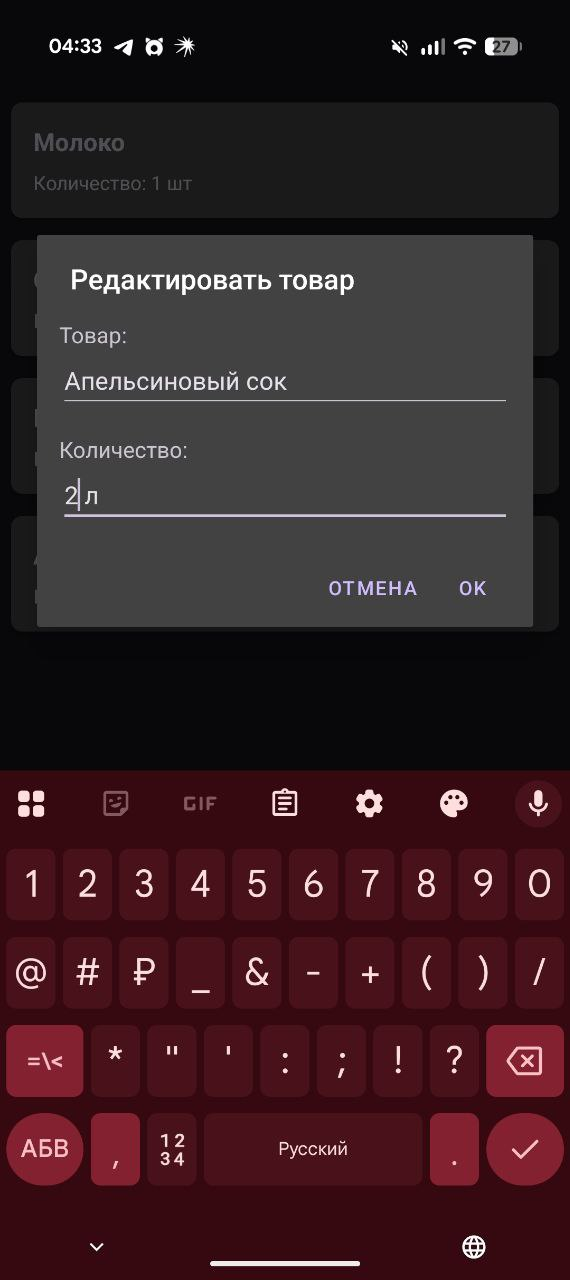


Рисунок 3 – Диалоговое окно редактирования существующего товара из списка

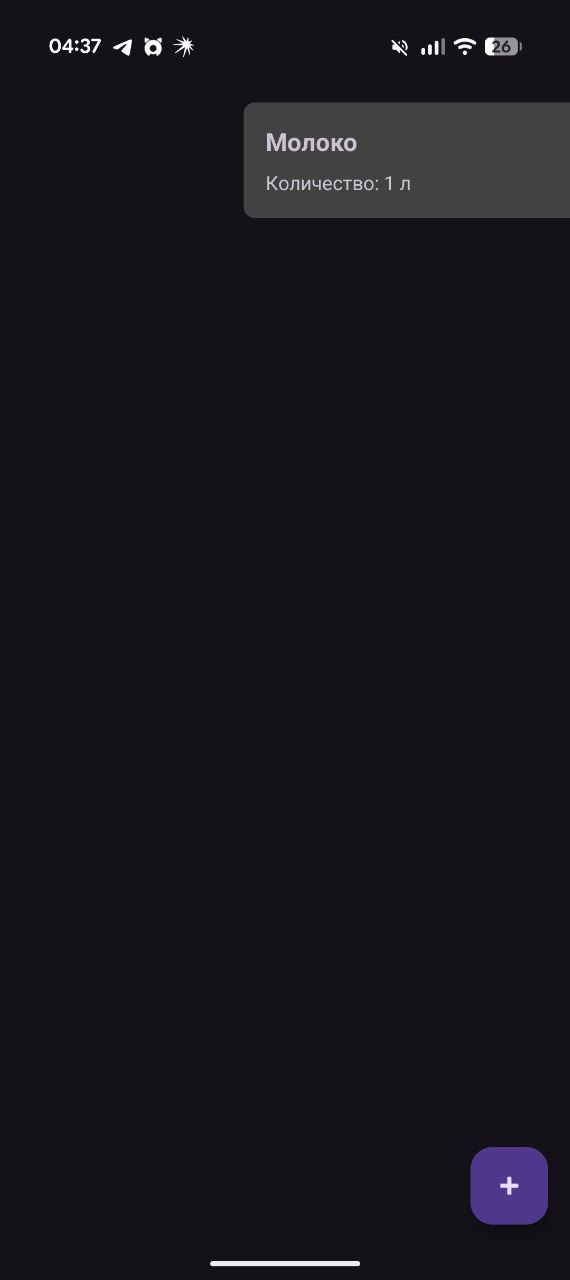


Рисунок 4 – Анимация удаления элемента списка жестом смахивания

# Выводы

В ходе выполнения лабораторной работы было разработано приложение для ведения списка покупок, использующее современные компоненты Android. RecyclerView с CardView обеспечивает отображение списка с современным дизайном. DialogFragment используется для показа диалоговых окон добавления и редактирования. Реализовано удаление элементов списка жестом смахивания с помощью ItemTouchHelper. Все строковые ресурсы и размеры вынесены в ресурсы, что облегчает поддержку и локализацию приложения. Приложение демонстрирует правильное использование архитектурных компонентов и соответствует требованиям задания.