

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА ОБЩЕЙ МАТЕМАТИКИ

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ
ИНТЕРНЕТ-МАГАЗИНА ДЛЯ ОС ANDROID**

Выпускная квалификационная работа
обучающегося по направлению подготовки
01.03.02 Прикладная математика и информатика
очной формы обучения, группы 12001510
Араказя Ален Бано

Научный руководитель:
к. ф.-м. н., доцент
Флоринский В. В.

БЕЛГОРОД 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОБЗОР ТРЕБОВАНИЙ И ПРИНЦИПОВ ПОСТРОЕНИЯ АНДРОИД-ПРИЛОЖЕНИЯ ДЛЯ ИНТЕРНЕТ-МАГАЗИНА	5
1.1 Особенности разработки под ОС Android	5
1.2 Бизнес-требования	6
1.3 Среды разработки	7
1.3.1 Android Studio	9
1.3.2 Другие инструменты, необходимые для разработки	10
1.3.3 Работа с базами данных	12
1.4 Общая сема работы приложения	15
2 ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ИНТЕРНЕТ-МАГАЗИНА	16
2.1 Доменное имя и хостинг	16
2.2 Создание базы данных	17
2.2.1 Удалённая база данных MySQL	18
2.2.2 Локальная база данных SQLite	21
2.3 Создание необходимых ресурсов	23
2.3.1 Использование JSON с php	23
2.3.2 Загрузка ресурсы	25
2.3.3 Создание изображений и иконок	26
2.4 Структура разрабатываемого приложения	27
2.4.1 Манифесты	28
2.4.2 Визуальные ресурсы	30
2.4.3 Классы Java	31
3 ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ	44
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51

ВВЕДЕНИЕ

Человек всегда должен был знать о развитии информационных технологий для создания новых устройств и гаджетов. Полезность этих устройств означает, что они нам нужны постоянно, но размер компьютеров и ноутбуков создает неудобства при постоянном использовании.

Это привело к появлению различных типов мини-компьютеров: смартфоны, которые работают с операционными системами, отличными от компьютеров. Самые популярные операционные системы, используемые сегодня, являются Android и iOS. Но эти платформы могут полностью работать только при условии, что была реализована разработка мобильных приложений. Операционные системы не принесут много пользы, если они не смогут использовать другие приложения.

В данной выпускной квалификационной работе поставлена задача по разработке приложения для платформы Android: интернет-магазина приложений для всех тех, кто хочет делать покупки этом магазине, с возможностью просмотра продуктов и всей другой информации о продуктах, возможности заказа выбранного товара. Основным языком программирования на платформе Android являются Java, но есть библиотеки, которые позволяют разрабатывать на C++ или Kotlin.

Целью этой квалификационной работы является разработка приложения интернет-магазина для смартфонов, которые работают под ОС Android. Мы разработаем приложение на основе версии 4.1 Android (Jelly Bean), и которое сможет поддерживать эту версию и более поздние версии. Основное назначение приложения, которое мы собираемся создать это использование данных, расположенных на удаленном сервере, доступном для всех.

Задачи, которые необходимо выполнить: изучение и улучшение знания о создании приложения под ОС Android, и разработка программы, ознакомление с использованием многих потоков и функциями платформы. В процессе

разработки приложения, были рассмотрены и решены задачи: усовершенствованно знание о принципах разработки для платформ Android (разметка xml, элементы управления создание графики, использование баз данных, представления изображений, работа приложения, возможные функции для работы с приложениями для Android, задачи, которые работают в фоновом режиме, команды SQL и JSON, php); минимальная версия платформы была Android 4.1 (Jelly Bean).

Языком программирования приложения является язык Java, но другие языки будут использоваться для обеспечения наилучшего использования баз данных приложением.

Для создания приложения была использовано Android Studio 3.4 (бесплатная версия). В ходе выполнения работы было разработано соответствующее приложение. Это будет описано более подробно ниже. Разработанное приложение было протестировано.

1 ОБЗОР ТРЕБОВАНИЙ И ПРИНЦИПОВ ПОСТРОЕНИЯ АНДРОИД-ПРИЛОЖЕНИЯ ДЛЯ ИНТЕРНЕТ-МАГАЗИНА

В настоящее время рынок мобильных приложений в основном разделен на две части: Android и iOS. Android является наиболее используемой операционной системой в мире с долей рынка 80%, опережая Apple от iOS и Windows Phone от Microsoft. Обе платформы имеют свои особенности и способ работы. В нашем случае операционной системой, в которой мы будем работать, является Android.

1.1 Особенности разработки под ОС Android

В случае Android Google разработаны инструменты для разработчиков, адаптированные к этому интерфейсу, чтобы облегчить их работу и обеспечить полную совместимость продукта, разработанного с платформой.

Язык разработки приложений для Android - Java, а в последнее время - Kotlin. Этот относительно молодой язык позволяет завершить диапазон возможностей, предлагаемых Java до сих пор.

Помимо этих различных баз, консоль, разработанная Google, Android Studio, позволяет работать в соответствии с ожиданиями системы. Эта среда предоставляет инструменты и набор баз данных для разработки решений на основе кода, которые уже зарекомендовали себя с точки зрения эффективности и стабильности.

У Android есть несколько версий, последняя - восьмая (Android 8.0 Oreo), но есть много предыдущих версий, которые все еще работают на многих смартфонах. Поэтому важно убедиться, что в разработке учтен этот фактор обратной совместимости, чтобы не исключать его использование на любом

устройстве. Вот почему мы выбрали Android 4.1 Jelly Bean (API 19), так как приложение будет работать более чем на 99,6%.

1.2 Бизнес-требования

Наше приложение будет направлено на:

- разрешение просмотра товара;
- совершение покупки из выбранных товаров.

Но, чтобы достичь этих двух основных целей, были созданы другие функции, облегчающие использование нашего приложения:

- презентация товаров по категориям для облегчения;
- возможность выбрать понравившиеся товары и ранжировать их в избранную категорию;
- создание профиля пользователя для облегчения заказов товаров;
- предоставление помощи с использованием нашего приложения.

Характеристика разработанного приложения

Приложение, которое мы разработали, учитывая его функцию, имеет некоторые специфические для него характеристики, которые описаны ниже.

Приложение представляет собой интернет-магазин, что означает, что приложение должно отображать доступные товары в общем виде, товары обновляются при каждом изменении окна.

Поскольку наш магазин содержит товары нескольких категорий, приложение классифицирует товары по категориям. Данные о товарах не могут регистрироваться в приложении, что может усложнить обновление товаров, а также вызвать использование большего объема памяти в смартфоне.

Поэтому необходимо организовать механизм сбора данных для приложения. Этот механизм, называемый JSON (JavaScript Object Notation), представляет собой текстовый формат данных, полученный из нотации объектов на языке

JavaScript. Это позволяет представлять структурированную информацию, как, например, XML.

JSON позволит нам отправлять и получать данные, расположенные в базе данных и касающиеся товаров. Данные, полученные с помощью JSON, представлены в виде объектов, результат состоит из определенного числа объектов, и каждый объект содержит характеристики, значения которых меняются в зависимости от каждого объекта.

Этот механизм также предоставляет данные для всех подразделений приложения по запросу. Из-за характера данных для отображения этот механизм будет работать каждый раз, когда окно обновляется или изменяется, до тех пор, пока приложение полностью или вручную не остановится, что позволит получать реальные данные о товаре в любое момент.

1.3 Среды разработки

Android - самая используемая операционная система в мире. Поэтому неудивительно, что многие разработчики стремятся создавать новые приложения, работающие на этой операционной системе на основе Linux, будь то для развлечения или зарабатывания денег. Тем не менее, будь то любитель или эксперт в этой области, разработчик, для реализации своего проекта должны, несомненно:

- ✓ Знать основы языка Java или вообще объектно-ориентированного программирования.
- ✓ Установите на своем компьютере множество программ, таких как полный Android SDK или JDK.
- ✓ Выберите правильную рабочую среду.

В Java есть две платформы:

- JRE (Java Runtime Environment), которая содержит виртуальную машину Java (JavaVirtualMachine), базовые библиотеки языка и все компоненты,

необходимые для запуска приложений или апплетов Java. По сути, это набор инструментов, которые позволят вам запускать приложения Java.

- JDK (Java Development Kit), который содержит JRE (для запуска Java-приложений), а также набор инструментов для компиляции и отладки вашего кода.

SDK расшифровывается как SoftwareDevelopmentKit, это набор инструментов программирования для разработки программного обеспечения, игр, мобильных приложений и т.д. Для терминала и / или конкретной операционной системы. SDK содержит код, позволяющий создать интерфейс или часть цифрового интерфейса (веб, мобильный, игры, программное обеспечение для поиска ...). Этот код разработан с языком программирования, соответствующим терминалу (компьютер, телефон, планшет и т.д.) и целевой навигационной системе. Например, для разработки мобильного приложения (программного обеспечения) для iPhone (мобильный терминал с операционной системой iOS) необходимо использовать мобильный iOS SDK.

Чтобы сократить затраты и время, необходимые для завершения нового приложения, новых рабочих сред, которые объединяют различные инструменты,

Различные среды разработки запускаются каждый год, чтобы помочь разработчикам в их работе, и они подразделяются на две категории:

1. Таким образом, IDE (IntegratedDevelopmentEnvironment) облегчает компиляцию, написание и отладку реализованной программы и позволяет получить исполняемый файл (.apk). Вот некоторые из наиболее часто используемых на рынке:
 - a. Android Studio - это инструмент, установка которого возможна одновременно с SDK для всех типов операционных систем. Он представляет собой среду IDE, предпочитаемую Google для создания приложений Android. Благодаря своей мощности, простоте и свободе, он

- легко свергнет все остальные среды, используемые до этого. Вот почему был выбран Android Studio 3.3.1 в качестве среды разработки;
- b. Eclipse была одной из первых программных платформ для Android. Чтобы правильно использовать его на своем компьютере, вам потребуется несколько утилит, а также плагин ADT для Eclipse;
 - c. Netbeans - это интегрированная среда разработки, доступная в виде программного пакета NetbeansMobile;
2. Среда cross-platforms: Чтобы разработать приложение для Android, iOS или Windows Phone, следует использовать соответствующий язык, но особенно, соответствующую рабочую среду. Но сегодня появилась новая универсальная версия этих сред, называемая кроссплатформенной или мульти-ОС, для достижения большей экономии и обеспечения возможности выполнения одной программы в разных системах.
- a. Xamarin от Microsoft требует знания языка C #.
 - b. Среда Titanium работает с объектно-ориентированным JavaScript.

1.3.1 Android Studio

Android Studio - это среда разработки и программирования для разработки мобильных приложений Android, недавно выпущенная Google для систем Android. Он основан на IntelliJ IDEA и использует производственный механизм Gradle. Его можно загрузить в операционных системах Windows, macOS и Linux.

Android Studio также предлагает другие вещи:

- надежная среда разработки;
- простой способ проверить производительность на других типах устройств;
- мастера и шаблоны для общих предметов, найденных на всех программистов Android;

- полный редактор с набором инструментов для ускорения разработки вашего приложения.

Чтобы установить Android Studio, необходимо иметь программное обеспечение из комплекта разработки для Android (SDK), а также комплект для разработки (JDK), для которого необходимо указать его путь на жестком диске. В любом случае, Android Studio позволяет разработчикам продвигаться вперед с помощью правильных инструментов для разработки мощных решений.

1.3.2 Другие инструменты, необходимые для разработки

Редактор изображения

Помимо разработки в Android Studio необходимы другие инструменты, чтобы уточнить некоторые детали. Приложение имеет по крайней мере один конкретный значок или логотип и содержит изображения. Эти значки и изображения будут созданы с помощью Photoshop.

Клиентская программа FTP

Файлы (PHP, изображения) и данные необходимы для правильной работы приложения. Файлы должны быть доступны всем пользователям приложения, для этого они будут размещены на нашем сервере с помощью программы под названием Filezilla.

FileZilla - это программное обеспечение, которое позволяет копировать файлы с компьютера или с другого компьютера в сети, администрировать веб-сайт, переносить веб-сайт с локального компьютера в WorldWideWeb или удалять, или изменять его. файлы на этом компьютере. В этом процессе нам понадобится доменное имя и хост-сервер.

Доменное имя - это то, что позволяет легко идентифицировать веб-сайт. Он состоит из имени и расширения его домена, в зависимости от его активности.

Каждый компьютер, подключенный к Интернету, имеет IP-адрес, состоящий из последовательности цифр для его идентификации. Для удобства и простоты использования было решено сопоставлять каждый IP-адрес с другим именем: доменное имя. Соответствие между доменным именем и IP-адресом обеспечивается DNS (система доменных имен).

Веб-хостинг необходим, если вы хотите опубликовать сайт в Интернете. Веб-хостинг предоставит вам место на сервере, подключенном к Интернету, для размещения ваших файлов, после чего они будут видны всем через Интернет.

Эмулятор

Эмулятор - это программное обеспечение, роль которого заключается в имитации поведения любого компьютера. Компьютер или устройство, на котором работает эмулятор, ведет себя как машина, которую мы пытаемся имитировать. В случае микрокомпьютеров эмулятор восстанавливает ресурсы, доступные исходной машине. Он использует часть микропроцессора и оперативной памяти и воссоздает кассету или дисковод гибких дисков практически на жестком диске.

В нашем случае нас интересует эмулятор андроида. Этот эмулятор имитирует работу устройства Android. Версия операционной системы может быть изменена по нашему выбору.

Мы будем использовать самый быстрый и самый полный компьютерный эмулятор для Android: Genymotion. Этот эмулятор работает с VirtualBox. Он создает виртуальную машину в virtualbox, которая эмулирует Android в реальном времени на вашем ПК с Windows, Mac или Linux.

1.3.3 Работа с базами данных

Когда мы должны использовать много данных о копии какого-либо (объекта) желательно использовать базу данных. Данные о товарах нашего интернет-магазина будут храниться в базе данных. Есть несколько способов доступа к базе данных, но мы будем использовать JSON и SQLite.

JSON

JavaScript Object Notation (JSON) - это легкий текстовый обмен данными, полученный из объектной нотации JavaScript. Мы будем использовать JSON для передачи данных из нашей базы данных, расположенной на удаленном сервере, в наше приложение. Он используется в качестве языка передачи данных. Это позволяет представлять структурированную информацию, как, например, XML.

JSON основан на двух структурах, которые принимают следующие формы:

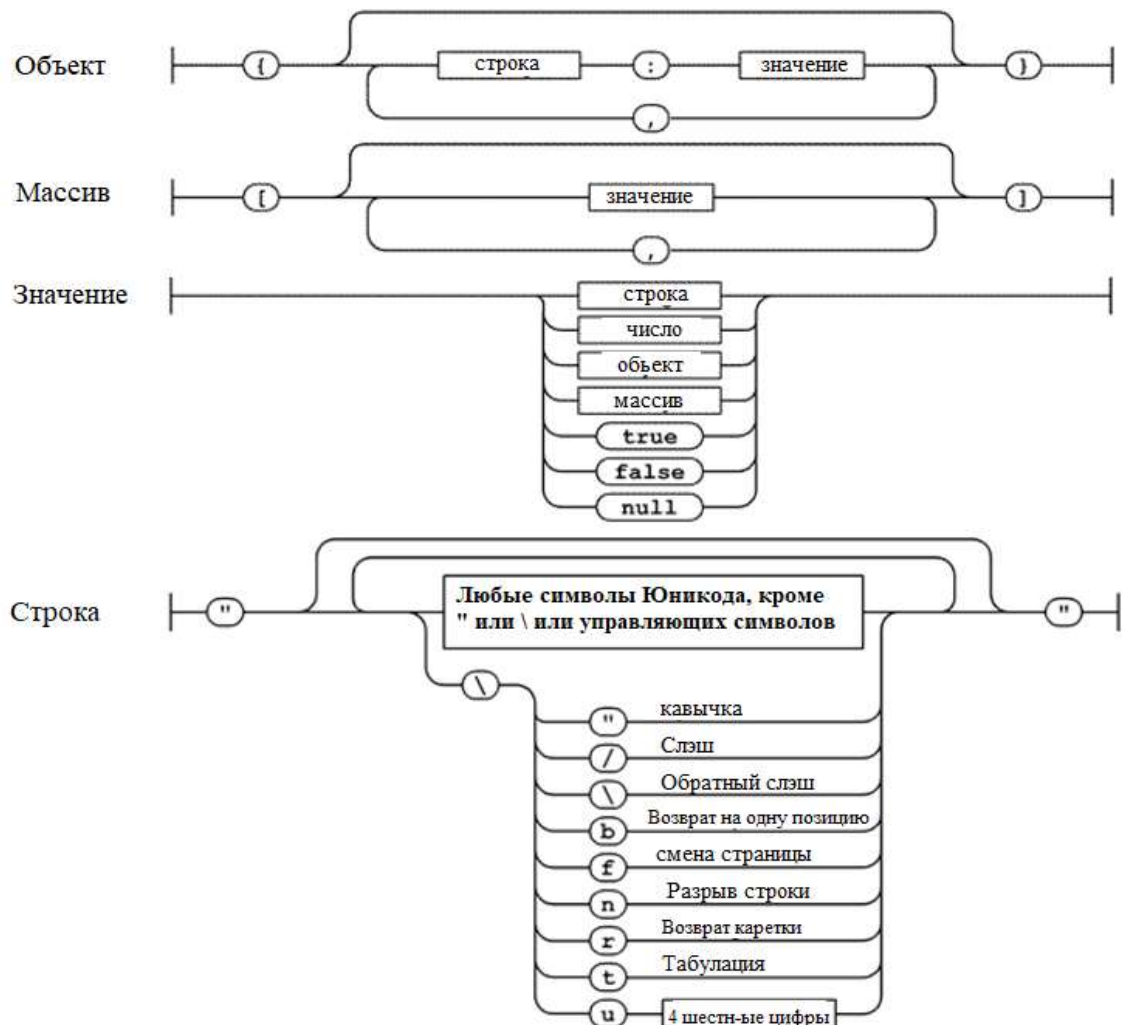


Рисунок 1.1 – Структура данных в JSON

- Объект, представляющий собой набор неупорядоченных пар имя / значение. Объект начинается с {(левая скобка) и заканчивается} (правая скобка). Каждое имя сопровождается: (двоеточие), а пары имя / значение разделяются, (запятая).
- Массив представляет собой набор упорядоченных значений. Массив начинается с [(левая скобка) и заканчивается] (правая скобка). Значения разделены, (запятая).
- Значением может быть либо строка в кавычках, число, истина или ложь, либо ноль, либо объект или массив. Эти структуры могут быть вложенными.
- Строка - это строка из нуля или более символов Юникода, заключенная в кавычки и использующая символы с обратной косой чертой. Символ представлен строкой из одного символа.

SQLite

Интерес в том, что его очень легко и быстро поставить на место. База данных, созданная SQLite, занимает очень мало памяти, база данных полностью хранится в файле, независимо от программного обеспечения. Его можно использовать для хранения данных в реальной базе данных на смартфоне (iPhone или Android). Вот почему мы собираемся использовать SQLite. Это позволит нам хранить личные данные, такие как любимые товары, заказ товаров и товаров в корзине.

1.3 Общая схема работы приложения

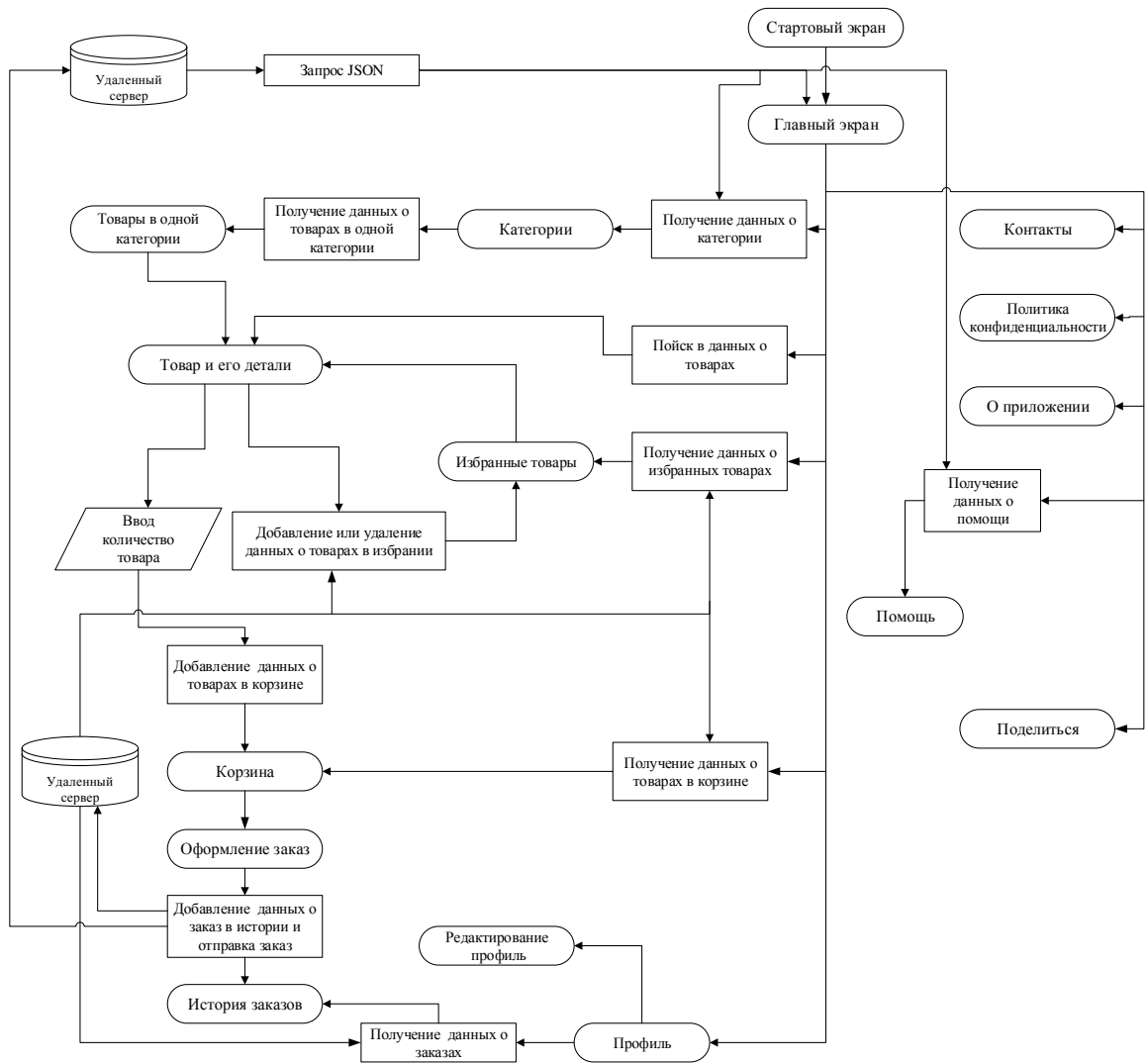


Рисунок 1.1 - Общая функциональная схема приложения

2 ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ИНТЕРНЕТ-МАГАЗИНА

В процессе разработки приложения интернет-магазина мы проходим несколько этапов. Сначала мы получим доменное имя, на котором размещены наши файлы, создадим необходимые ресурсы (базу данных, изображения, значок, файлы PHP и т. д.), загрузим наши файлы на сервер и, наконец, начнем разработку в Android Studio.

2.1 Доменное имя и хостинг

Доменное имя позволяет избежать цифрового отображения интернет-адреса веб-сайта, технической формой которого является IP-адрес, состоящий из последовательности цифр.

Чтобы выбрать доменное имя, нужно проверить три вещи:

- a. Длина: Чем длиннее имя, тем легче его запомнить. Выбранное имя должно состоять из 3 и 63 символов;
- b. Форма: Несколько слов могут быть связаны разделенными черточками. Заглавные буквы запрещены. Цифры от 0 до 9 допускаются;
- c. Доступность: Выбранное доменное имя (имя + расширение) должно быть бесплатным, оно не хочет использоваться другим лицом или организацией.

Для компании просто выбрать имя интернет-адреса. Чаще всего это название компании, бренда или конкретного продукта. В данном случае была связана фамилия с именем интернет-магазина, с расширением, которое зависит от местоположения (Россия): И, наконец, благодаря REG.RU было получено следующее доменное имя: arakazatechshop.ru

Веб-хостинг необходим, потому что мы хотим, чтобы определенные данные из нашего интернет-магазина были легко доступны в Интернете. Мы выбрали REG.RU в качестве веб-хоста, он дал нам место на сервере, подключенном к Интернету, для размещения наших файлов, после чего они будут доступны всем пользователям нашего приложения через Интернет (набрав название поле: arakazatechshop.ru)

Веб-хостинг (WebHosting) - это услуга, которая позволяет предприятиям и частным лицам размещать веб-сайты или веб-страницы в Интернете. Веб-хостинг - это компания, которая предоставляет технологии и услуги, необходимые для того, чтобы веб-сайт или веб-страница были доступны в Интернете; он предоставляет в распоряжение место для хранения на сервере, постоянно подключенном к Интернету.

Наш хостинг REG.RU позволяет нам, среди прочего, сделать ваше доменное имя, наши файлы и наши данные доступными для всех пользователей Интернета. REG.RU дает нам 7 ГБ памяти, и мы получаем имя пользователя и пароль для доступа к нему.

2.2 Создание базы данных

В нашем приложении нам нужно будет использовать базы данных. И эти данные можно разделить на две категории:

- Публичные данные, которые будут одинаковыми для всех пользователей приложения (товары, категории товаров и т. Д.). Эта база данных будет сохранена на удаленном сервере REG.RU
- Личные данные, которые различаются в зависимости от пользователя (товары в корзине, любимые товары и т. Д.). Эта база данных будет сохранена в самом приложении.

Базы данных гораздо эффективнее, когда дело доходит до поиска информации.

2.2.1 Удаленная база данных: MySQL

Как и электронные таблицы, базы данных организуют вашу информацию в списки. Однако, в отличие от электронных таблиц, вы можете отображать их в виде форм. Формы phpMyAdmin позволяют нам заполнять таблицы, которые мы создали в нашей базе данных.

Базы данных могут содержать столько данных, сколько вы хотите, и позволяют вам представлять эти данные по своему усмотрению.

Мы будем использовать mysql-5.7.23 нашей базы данных (u0717322_techshop_db), чтобы связать информацию, содержащуюся в таблицах: (tbl_category, tbl_currency, tbl_help, tbl_order, tbl_product, tbl_setting, tbl_shipping).

Таблица 2.1 – Описание удалённого базы данных

Название столбца	Тип	Описание
tbl_category – таблица категория		
Category_ID	int	Первичный ключ и идентификатор категория
Category_name	varchar	Название категория
Category_image	varchar	изображение категория
tbl_product – таблица продуктов		
product_id	int	Первичный ключ и идентификатор продукта
product_name	varchar	Название продукта
product_price	double	Цена продукта
product_status	varchar	доступность продукта
product_image	text	изображение продукта
product_description	text	Описание продукта
product_quantity	int	Количество продукта
category_id	int	идентификатор категория продукта

Продолжение таблицы 2.11

Tbl_order– таблица заказ продуктов		
id	int	Первичный ключ и идентификатор заказа
code	varchar	Код заказа
name	varchar	Цена заказчика
email	varchar	Электронная почта заказчик
phone	varchar	номер телефона заказчика
address	varchar	Адрес заказчик
shipping	varchar	тип доставки
Date_time	datetime	Время заказа
Order_list	text	Список продукта заказа
Order_total	varchar	Итоговая цена заказа
comment	text	комментарий
status	varchar	Статус заказа
Tbl_shipping– таблица		
shipping_id	int	Первичный ключ и идентификатор типа доставки
shipping_name	varchar	Название тип доставки
Tbl_help– таблица		
id	int	Первичный ключ и идентификатор помощь
title	varchar	название помощи
content	text	содержание помощи
Tbl_setting		
Variable	varchar	переменная
Value	varchar	значение

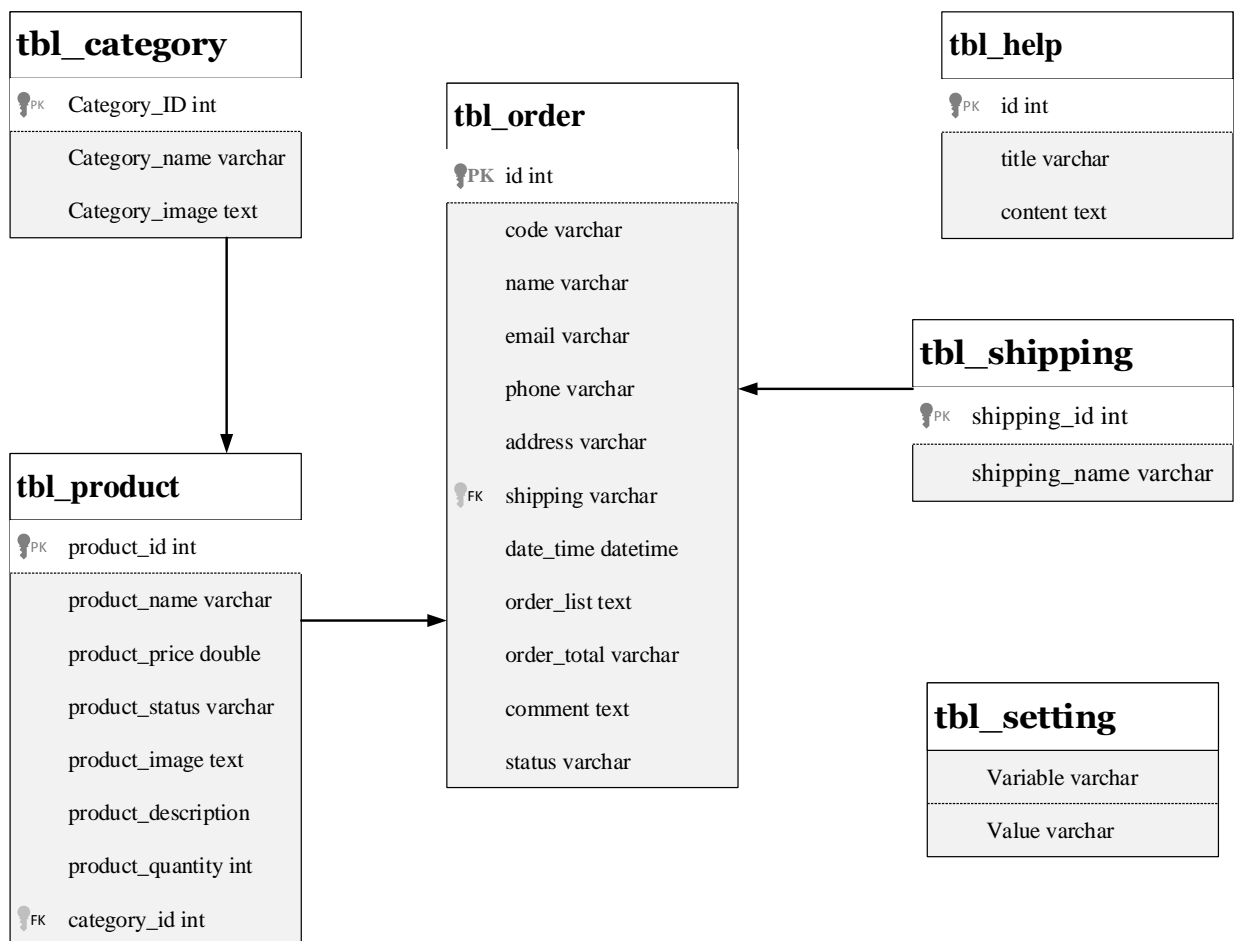


Рисунок 2.1 – Схема структура удалённого базы данных

Мы собираемся связать таблицу товаров с таблицей категорий, потому что каждый товар можно классифицировать в определенной категории. Таблица товаров, таблица типов доставки и таблица заказов также будут связаны.

Подключение к базе данных может быть сделано только при наличии 4 основных данных:

- имя хоста или имя сервера: адрес сервера;
- имя пользователя: логин;
- пароль: пароль;
- имя базы данных: название базы данных.

2.2.2 Локальная база данных: SQLite

Прежде всего, эта база данных SQLite очень легкая и требует всего 250 КБ памяти во время выполнения. Кроме того, она обрабатывает большинство преимуществ стандартной базы данных (подготовленные операторы, транзакции, синтаксис SQL).

В отличие от базы данных, расположенной на сервере, который использует MySQL, наша локальная база данных использует SQLite. Сначала мы создадим базу данных с помощью DB Browser (SQLite) и добавим ее в приложение в папке ресурсов.

Тем не менее, он поддерживает только 3 типа данных:

- TEXT: аналогично String в Java, означает строки;
- REAL: аналогичен double в Java, означает числа после десятичной точки;
- INTEGER: аналогично long в Java, означает целые числа.

Все остальные типы (например, даты) должны быть сохранены как текст, а затем преобразованы приложением.

В Android все классы, которые работают с базами данных, находятся в пакете android.database. Те, которые облегчают работу с SQLite, размещаются в android.database.sqlite. Эта база данных содержит три таблицы: tbl_favorite, tbl_cart, tbl_history

Таблица 2.2 – Описание базы данных внутри приложения

Название столбца	Тип	Описание
tbl_favorite – таблица избранного		
Id_favorite	int	Первичный ключ и идентификатор избранника
product_id	int	идентификатор продукта

Продолжение таблицы 2.2

tbl_cart – Таблица товаров в корзине		
id	int	Первичный ключ и идентификатор
product_name	varchar	Название продукта
quantity	int	Количество продукта
total_price	numeric	Цена
currency_code	varchar	код валюты
product_image	varchar	изображение продукта
tbl_history – таблица		
id	int	Первичный ключ и идентификатор заказа
code	varchar	Код заказа
Order_list	varchar	Список продукта заказа
Order_total	varchar	Итоговая цена заказа
Date_time	varchar	Время заказа

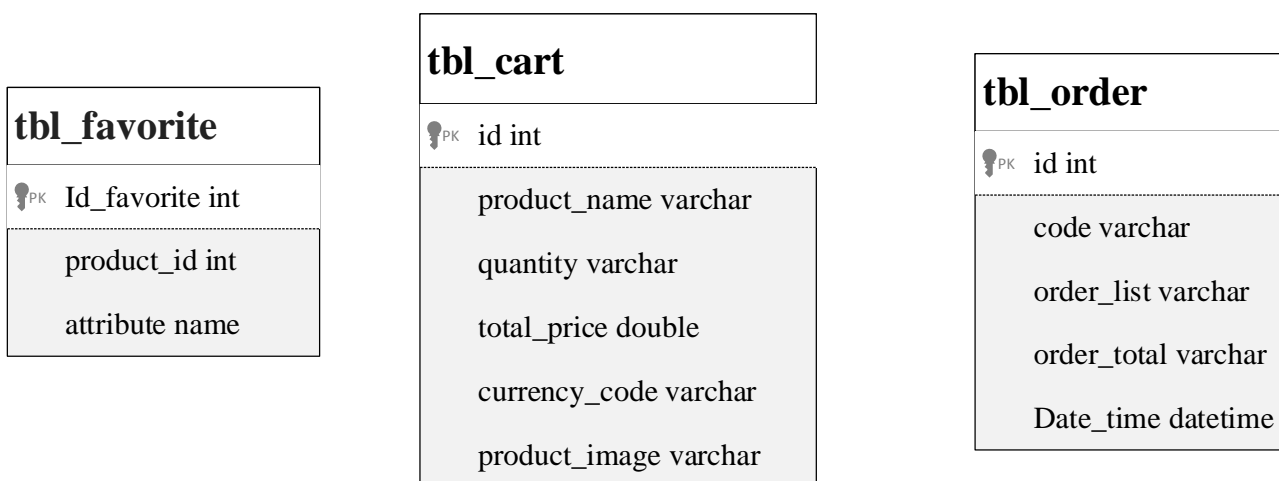


Рисунок 2.2 – Схема структура локального базы данных

2.3 Создание необходимых ресурсов

При разработке нашего приложения нам понадобятся некоторые другие вещи, такие как изображения, значки. И специально для нашего приложения мы будем использовать файл JSON для доступа к данным нашей удаленной базы данных.

2.3.1 Использование JSON с php

JSON - это способ хранения информации, очень похожий на базу данных. JSON - это формат данных, состоящий из пар имя / значение (или ключ / значение) в форме строк. Имя и значение разделяются двумя точками: каждая пара отделяется от следующей запятой.

Он стал очень популярным для хранения, чтения и обмена информацией в веб-приложениях и сервисах.

Поскольку JSON представлен в виде текста, то нельзя давать ему такие вещи, как функции или динамические значения даты, которые используют Date (). Поэтому в JSON нет методов или других функций, есть только текст. И это хорошо, потому что именно это делает его универсальным форматом для обмена данными. Чтобы использовать или манипулировать нашим JSON-файлом, необходимо установить веб-сервер и PHP.

JSON - это способ хранения информации, очень похожий на базу данных. Мы будем использовать JSON для доступа к информации нашей базы данных. Сначала вам нужно подключиться к базе данных, для этого вы будете использовать следующий код. Для большего удобства наш код JSON будет связан с файлом PHP.

```
<?php
```

```
$host = "localhost";  
$user = "u0717322_client";  
$pass = "clientacces322";
```

```

$database = "u0717322_techshop_db";
$connect = new mysqli($host, $user, $pass, $database);
if (!$connect) { die ("connection failed: " . mysqli_connect_error());}
else { $connect->set_charset('utf8');}

```

?>

Наша база данных содержит несколько таблиц, и для доступа к данным каждой таблицы необходимо было бы каждый раз подключаться, повторяя фрагмент кода выше. Но чтобы избежать этого повторения, все запросы каждой таблицы будут помещены в один файл PHP. Пользователь должен будет указать, какой запрос (запрос) он хочет использовать.

<?php

```

if (isset($_GET['get_recent']))
    { /* Запрос на получение информации о товаре*/ }
else if (isset($_GET['get_category']))
    { /* Запрос на получение информации о категории*/ }
else if (isset($_GET['get_tax_currency']))
    { /* Запрос на получение налоговую валюту*/ }
else if (isset($_GET['category_id']))
    { /* Запрос на получение информации о категории */ }
else if (isset($_GET['post_order']))
    { /* Запрос на отправление информации о товаре*/ }
else if (isset($_GET['get_shipping']))
    { /* Запрос на получение типа доставки*/ }
else if (isset($_GET['get_help']))
    { /* Запрос на получение помощь*/ }
elseif (isset($_GET['product_id']))
    { /* Запрос на получение информации об одной товар*/ }
else { header('Content-Type: application/json; charset=utf-8');
      echo "nomethodfound!"; }

```

?>

Каждый запрос отличается от других в зависимости от данных, которые он хочет получить или которые он хочет отправить. Но у них есть только одна общая черта: преобразование данных, полученных в текстовом формате JSON, и решение проблемы кодирования. Для этого в каждом запросе мы добавляем эти две строки кода:

```
header('Content-Type: application/json; charset=utf-8');  
echo $val = str_replace("\\'", "'", json_encode($set));
```

Файл php будет работать только после загрузки на сервер, на котором находится наша база данных.

2.3.2 Загрузка ресурсы

Для загрузки файлов требуется клиентская программа FTP или другое программное обеспечение, чтобы разместить наши файлы в Интернете. Но мы собираемся использовать Filezilla.

У нас уже есть информация для входа на FTP, которую мы получили при покупке доменного имени и хостинга. Мы вводим информацию о FTP-соединении в поля вверху программы и затем нажимаем QuickConnect.

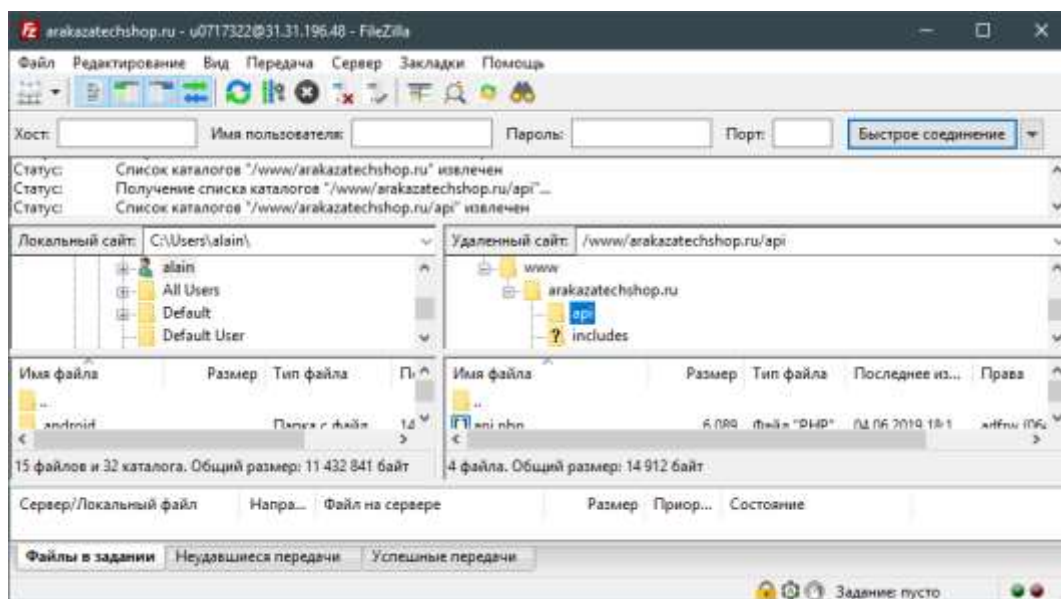


Рисунок 2.3 – Интерфейс программы FileZilla

После подключения к FTP-серверу браузер файлов на левой стороне программы позволяет находить файлы, созданные на вашем компьютере. После нахождения файлов мы копируем их в файловый браузер на ПРАВОЙ стороне в каталоге www.arakazatechshop.ru и в соответствии с нашим повторением (API) на хост-сервере.

Созданный файл PHP будет размещен на сервере. Как и любой другой php-файл, чтобы запустить наш php-файл, содержащий запрос JSON, необходимо выполнить в браузере ссылку: Доменное имя + направление + имя файла + расширение файла + имя запроса для выполнения

В нашем случае это дает:

www.arakazatechshop.ru/api/api.php?get_category

После запуска по этой ссылке мы получаем информацию о категориях:

```
[{"category_id":"4","category_name":"Техника для дома","category_image":"home.png","product_count":"0"}, {"category_id":"3","category_name":"Компьютерыноутбуки", "category_image":"notebook.png","product_count":"2"}, {"category_id":"2","category_name":"Смартфоныгаджеты","category_image":"smartphone.png","product_count":"7"}, {"category_id":"1","category_name":"Телевизоры, аудио, видео","category_image":"tv.png","product_count":"3"}]
```

2.3.3 Создание изображений и иконок

Каждое приложение имеет значок, который является символическим и специфичным для него. Это значок, который мы создадим сами. Используя PHOTOSHOP, мы создали изображение в формате JPG, которое будет отображаться в начале приложения, и две иконки в формате PNG.



Рисунок 2.4 – Изображение и значки нашего приложения

Необходимо знать, что это не единственный дизайн, который понадобится в нашем приложении. Другие рисунки могут быть получены через Android Studio, эти значки являются изменяемыми и имеют расширение XML

2.4 Структура разрабатываемого приложения

Android Studio - это IDE, использующая для разработки нашего Android-приложения собственную ОС Google для мобильной разработки. В Android Studio проект - это папка, которая разделяется на две основные части с определенными функциями: GradleScripts и App.

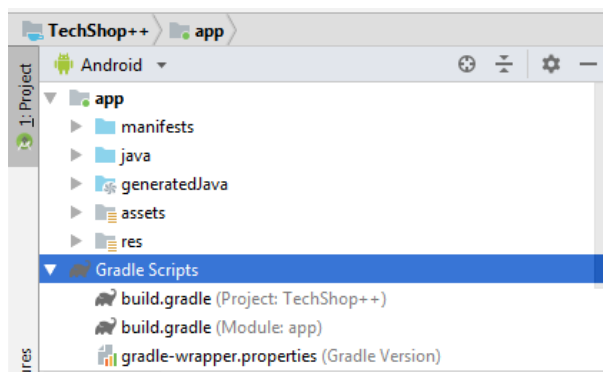


Рисунок 2.5 – Архитектура приложения

Часть GradleScripts содержит некоторые файлы, которые создаются при создании проекта, единственные файлы, которые нас интересуют, это файлы build.gradle (Модуль: app). В этом файле мы добавляем зависимости

(dependencies). Это набор файлов, которые мы добавляем в наш проект, чтобы было проще использовать элементы, методы, уже созданные.

Существуют разные способы добавления библиотек в проект Android Studio. Мы добавим эти библиотеки, добавив одну строку на библиотеку в файле build.gradle. В наш проект мы добавим 10 библиотек:

```
'com.android.support:appcompat-v7:28.0.0'  
'com.android.support:design:28.0.0'  
'com.android.support:cardview-v7:28.0.0'  
'com.android.support:recyclerview-v7:28.0.0'  
'com.android.support.constraint:constraint-layout:1.1.3'  
'com.balysv:material-ripple:1.0.2'  
'com.github.clans:fab:1.6.4'  
'com.android.volley:volley:1.1.0'  
'com.google.code.gson:gson:2.8.4'  
'com.makeramen:roundedimageview:2.3.0'
```

Часть приложения содержит 5 директорий:

1. Manifests (Манифесты) содержат файл AndroidManifest.xml;
2. Java содержит классы Java;
3. generatedJava содержит автоматически созданные файлы, которые никогда не следует изменять;
4. Assets (активы) содержат нашу базу данных;
5. resources (ресурсы) в основном содержат визуальную часть нашего приложения.

2.4.1 Манифесты

Приложение должно иметь файл AndroidManifest.xml (именно с этим именем) в своем корневом каталоге. В манифесте представлена важная информация о приложении для системы Android, система нуждается в этой

информации, чтобы иметь возможность выполнять код приложения. Среди прочего, манифест делает следующее:

- дает название пакета Java приложения, имя пакета служит уникальным идентификатором для приложения;
- в нем описываются компоненты приложения, которые представляют собой действия, услуги, широковещательные приемники и поставщики контента, которые составляют приложение;
- указывает разрешения, которые приложение должно иметь для доступа к защищенным частям API и взаимодействия с другими приложениями, он также объявляет разрешения, которые должны иметь другие пользователи для взаимодействия с компонентами приложения;
- в нем перечислены библиотеки, с которыми должно быть связано приложение.

Иконки: значок и текстовая метка, определенные в файле `AndroidManifest.xml`, являются отображаемыми в модуле запуска.

Разрешение: разрешение - это ограничение, ограничивающее доступ к части кода или данным устройства, ограничение накладывается для защиты критически важных данных и кода, которые могут быть использованы не по назначению, чтобы исказить или повредить пользовательский опыт.

Каждое разрешение идентифицируется уникальным ярлыком. Часто на этикетке указывается ограниченное действие. Например, вот некоторые разрешения, требуемые нашим приложением для действий:

- `"android.permission.INTERNET"`: доступ к интернет
- `"android.permission.ACCESS_NETWORK_STATE"`: доступ к сети
- `"android.permission.ACCESS_WIFI_STATE"` : доступ к Wi-Fi
- `"android.permission.WRITE_EXTERNAL_STORAGE"` : запись во внешнее хранилище
- `"android.permission.READ_EXTERNAL_STORAGE"`: чтение во внешнее хранилище

Библиотеки: Каждое приложение связано с библиотекой Android по умолчанию, которая включает в себя базовые пакеты для сборки приложений. Однако некоторые пакеты находятся в своих собственных библиотеках. Если приложение использует код из одного из этих пакетов, оно должно явно запросить привязку к ним.

2.4.2 Визуальные ресурсы

Ресурсы, необходимые элементы для разработки визуальной части проекта Android, включают макеты и меню, переменные, изображения (значки и т. Д.), Цвета, стили, темы, которые хранятся в каталог `res/`. Различные каталоги создаются базовыми, такими как:

- `res/drawable/`: для всех наших изображений;
- `res/layout/`: будет включать все наши файлы XML, соответствующие нашим макетам;
- `res/menu/`: чтобы сохранить наши меню в формате XML;
- `res/mipmap/`: для хранения наших иконок;
- `res/values/attrs`: для всех атрибутов наших элементов;
- `res/values/colors`: для всех цветов, которые мы используем в вашем проекте;
- `res/values/dimens`: для всех размеров наших элементов;
- `res/values/integer`: для всех значений наших элементов;
- `res/values/strings`: для всех переменных и слов нашего приложения;
- `res/values/styles`: для разных стилей нашего приложения;
- `res/xml/`: для хранения файлов XML.

Макеты и меню являются важными частями нашего проекта. В разделе меню имеется 7 меню:

- `Drawer_menu.xml` - соответствует горизонтальному меню, которое отображается только при нажатии кнопки меню;
- `Main.xml` - пустое меню;

- Menu_cart - меню окна корзины с кнопкой удаления;
- Menu_Image - меню с опцией сохранения и закрытия;
- Menu_list - меню с опциями «Поделиться» и «Корзина»;
- Поиск - меню поиска;
- Навигация - меню снизу.

Каталог макетов содержит действия, фрагменты и другие визуальные файлы XML. Во многих случаях активность будет соответствовать виду вашей заявки. Мы постараемся использовать минимально возможные мероприятия. Попытаемся создать вложенные представления, в которых будем использовать только одно действие с несколькими методами вместо нескольких связанных действий. Фрагменты соответствуют «концам зрения», они менее требовательны, что деятельность и является одним из решений для облегчения нашей деятельности приложения.

У нас всего:

- 11 activity (активность): activity_about_app.xml, activity_cart.xml, activity_checkout.xml, activity_contact.xml, activity_help.xml, activity_history.xml, activity_image_detail.xml, activity_main.xml, activity_product.xml, activity_product_detail.xml, activity_splash.xml;
- 2 Fragment (фрагмент): fragment_profile.xml , fragment_recent.xml;
- 5 include_cart_total.xml, include_empty_cart.xml, include_empty_history.xml, include_row_separator.xml, input_dialog.xml;
- 5 item_cart.xml, item_category.xml, item_help.xml, item_history.xml, item_product.xml;
- 4 nav_header.xml, spinner_dropdown_item.xml, spinner_item.xml, toolbar.xml.

2.4.3 Классы Java

Эффективная структура нашего проекта позволит легко найти себя и эффективно управлять различными компонентами нашего проекта.

При создании нового проекта Android Studio у нас автоматически появляется подпапка Java, которая будет содержать весь код Java. Подпапка Java находится в пакете нашего проекта "com.TechShop.alainbano".

Наши различные файлы Java будут помещены в корневой каталог com.TechShop.alainbano в разных подкатегориях:

- подпапка деятельность «Activities» будет содержать все действия нашего проекта;
- подпапка адаптеры «Adapters» будет содержать различные адаптеры, которые мы создали для отображения наших данных в ListView, например, GridView;
- подпапка фрагменты «Fragments» будет содержать различные фрагменты, которые мы сможем создать для нашего проекта;
- подпапка модели «Models» будет содержать все модели, представляющие объекты нашего приложения;
- подпапка утилиты «Utilities» будет содержать наши файлы Java, которые будут содержать различные коды.

Activities

Activity - это уникальная и целенаправленная задача, которую может выполнить пользователь. Практически во всех действиях, взаимодействующих с пользователем, класс действий отвечает за создание окна, в котором можно разместить свой пользовательский интерфейс с помощью setContentView (View). Хотя действия часто представляются пользователю в виде полноэкранных окон, их также можно использовать другими способами. Существует метод, используемый почти всеми подклассами Activity:

onCreate (Bundle) - это то место, где можно начинать свой бизнес. Что еще более важно, здесь можно обычно вызывать setContentView (int) с ресурсом презентации, определяющим пользовательский интерфейс, и использовать

findViewById (int) для извлечения виджетов из этого пользовательского интерфейса, с которым можно взаимодействовать программно.

1. public class ActivitySplash

- public void onCreate(Bundle savedInstanceState)

2. public class MainActivity

- protected void onCreate(Bundle savedInstanceState)
- private BottomNavigationView.OnNavigationItemSelectedListener navListener
- public boolean onOptionsItemSelected(@NonNull MenuItem item)
- public void showAlertDialog()
- public boolean onCreateOptionsMenu(Menu menu)
- public boolean onOptionsItemSelected(MenuItem item)
- private void makeJsonObjectRequest()

3. public class MyApplication

- public void onCreate()
- public static synchronized MyApplication getInstance()
- public RequestQueue getRequestQueue()
- public <T> void addToRequestQueue(Request<T> req, String tag)
- public <T> void addToRequestQueue(Request<T> req)
- public void cancelPendingRequests(Object tag)

4. public class ActivitySettings

- protected void onCreate(Bundle savedInstanceState)
- protected void onCreate(Bundle savedInstanceState)
- private void initToolbar()
- public boolean onOptionsItemSelected(MenuItem item)
- public ActionBar getSupportActionBar()
- public void setSupportActionBar(@Nullable Toolbar toolbar)
- public MenuInflater getMenuInflater()
- public void setContentView(@LayoutRes int layoutResID)
- public void setContentView(View view)

- public void setContentView(View view, ViewGroup.LayoutParams params)
 - public void addContentView(View view, ViewGroup.LayoutParams params)
 - protected void onTitleChanged(CharSequence title, int color)
 - public void onConfigurationChanged(Configuration newConfig)
 - public void invalidateOptionsMenu()
 - private AppCompatActivityDelegate getDelegate()
5. public class ActivityProductDetail
- protected void onCreate(Bundle savedInstanceState)
 - getData();
 - initComponents();
 - displayData();
 - setupToolbar();
 - makeJsonObjectRequest();
 - public boolean onCreateOptionsMenu(Menu menu)
 - public boolean onOptionsItemSelected(MenuItem menuItem)
 - private void requestStoragePermission()
6. public class ActivityProduct
- protected void onCreate(Bundle savedInstanceState)
 - public boolean onCreateOptionsMenu(Menu menu)
 - public boolean onOptionsItemSelected(MenuItem menuItem)
 - public void onContactSelected(Product product)
7. public class ActivityImageDetail
- protected void onCreate(Bundle savedInstanceState)
 - private void initToolbar()
 - public boolean onCreateOptionsMenu(Menu menu)
 - public boolean onOptionsItemSelected(MenuItem menuItem)
 - private void requestStoragePermission()
 - private void showSettingsDialog()
 - private void openSettings()

- public class saveImage
8. public class ActivityHistory
 - public void onCreate(Bundle savedInstanceState)
 - public boolean onCreateOptionsMenu(Menu menu)
 - public boolean onOptionsItemSelected(MenuItem item)
 - public void showClearDialog(int flag, int id)
 - public void clearData()
 - public class getDataTask
 - public void getDataFromDatabase()
 9. public class ActivityHelp
 - protected void onCreate(Bundle savedInstanceState)
 - public boolean onOptionsItemSelected(MenuItem item)
 10. public class ActivityContact
 - protected void onCreate(Bundle savedInstanceState)
 11. public class ActivityCheckout
 - protected void onCreate(Bundle savedInstanceState)
 - public void setupToolbar()
 - private void getSpinnerData()
 - private void getShipping(JSONArray jsonArray)
 - private String setShipping(int position)
 - public void submitOrder()
 - public void getValueFromEditText()
 - public void requestAction()
 - public void getTaxCurrency()
 - public void getDataFromDatabase()
 - public void dialogSuccessOrder()
 - private static String getRandomString(final int sizeOfRandomString)
 - public boolean onOptionsItemSelected(MenuItem item)
 - public void onResume()

12. public class ActivityCart

- public void onCreate(Bundle savedInstanceState)
- public boolean onCreateOptionsMenu(Menu menu)
- public boolean onOptionsItemSelected(MenuItem item)
- public void showClearDialog(int flag, int id)
- public void clearData()
- public class GetDataTask
- public void getDataFromDatabase()
- public void onBackPressed()
- public void onConfigurationChanged(final Configuration newConfig)

13. public class ActivityAboutApp

- protected void onCreate(Bundle savedInstanceState)

Фрагменты

Каталог «Fragments» содержит Java-классы, которые служат для завершения соответствующих фрагментов. Классы фрагментов работают с адаптерами, чтобы разрешить отображение.

1. public class FragmentRecent, public class FragmentCategory, public class FragmentFavorite и public class FragmentHelp используют одни и те же типы методов. Разница лишь в том, что эти методы используют разные объекты.

- public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- private void onRefresh()
- private void fetchData()
- public void onCreateOptionsMenu(Menu menu, MenuInflater inflater)
- public boolean onOptionsItemSelected(MenuItem item)
- public void onContactSelected(Product product)

- //public void onContactSelected(Help help)
 - //public void onContactSelected(Category category)
2. public class FragmentProfile
 - public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
 - public void onResume()

Adapters

Адаптер для объекта действует как мост между AdapterRecyclerView и базовыми данными для этого представления. Адаптер обеспечивает доступ к элементам данных. Адаптер также отвечает за создание View для каждого образца объекта.

1. public class AdapterCart
 - public class ViewHolder extends RecyclerView.ViewHolder
 - public AdapterCart(Context context, List<Cart> arrayCart)
 - public ViewHolderonCreateViewHolder(ViewGroup parent, int viewType)
 - public void onBindViewHolder(final ViewHolder holder, final int position)
 - public int getItemCount()
2. public class RecyclerViewAdapterCategory
 - public class MyViewHolder extends RecyclerView.ViewHolder
 - public RecyclerViewAdapterCategory(Context context, List<Category> categoryList, ContactsAdapterListener listener)
 - public MyViewHolderonCreateViewHolder(ViewGroup parent, int viewType)
 - public void onBindViewHolder(MyViewHolder holder, final int position)
 - public int getItemCount()
 - public Filter getFilter()
 - public interface ContactsAdapterListener
3. public class RecyclerViewAdapterFavoritesProduct

- public class MyViewHolder extends RecyclerView.ViewHolder
 - public RecyclerViewAdapterFavoritesProduct(Context context, List<Product>productList, ContactsAdapterListener listener)
 - public MyViewHolderonCreateViewHolder(ViewGroup parent, int viewType)
 - public void onBindViewHolder(MyViewHolder holder, final int position)
 - public int getItemCount()
 - public Filter getFilter()
 - public interface ContactsAdapterListener
4. public class RecyclerViewAdapterHelp
- public class MyViewHolder extends RecyclerView.ViewHolder
 - public RecyclerViewAdapterHelp(Context context, List<Product>productList, ContactsAdapterListener listener)
 - public MyViewHolderonCreateViewHolder(ViewGroup parent, int viewType)
 - public void onBindViewHolder(MyViewHolder holder, final int position)
 - public int getItemCount()
 - public Filter getFilter()
 - public interface ContactsAdapterListener
5. public class RecyclerViewAdapterHistory
- public class MyViewHolder extends RecyclerView.ViewHolder
 - public RecyclerViewAdapterHistory(Context context, List<Product>productList, ContactsAdapterListener listener)
 - public MyViewHolderonCreateViewHolder(ViewGroup parent, int viewType)
 - public void onBindViewHolder(MyViewHolder holder, final int position)
 - public int getItemCount()
 - public Filter getFilter()
6. public class RecyclerViewAdapterProduct
- public class MyViewHolder extends RecyclerView.ViewHolder
 - public RecyclerViewAdapterProduct(Context context, List<Product> productList, ContactsAdapterListener listener)

- `public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType)`
- `public void onBindViewHolder(MyViewHolder holder, final int position)`
- `public int getItemCount()`
- `public Filter getFilter()`
- `public interface ContactsAdapterListener`

Модели

Классы в «Models» представляют объекты, используемые в нашем приложении, каждый объект имеет атрибуты, представленные полем. Каждый класс содержит метод и конструктор для получения значений атрибутов. Эти методы и конструктор общедоступны для использования в других классах нашего приложения.

1. `public class Cart`

- `public Cart(String menuId, String menuName, String menuQuantity, String menuPrice)`
- `public String getMenuId()`
- `public void setMenuId(String menuId)`
- `public String getMenuName()`
- `public void setMenuName(String menuName)`
- `public String getMenuQuantity()`
- `public void setMenuQuantity(String menuQuantity)`
- `public String getMenuPrice()`
- `public void setMenuPrice(String menuPrice)`

2. `public class Category`

- `public String getCategory_id()`
- `public String getCategory_name()`
- `public String getCategory_image()`

- public String getProduct_count()
3. public class Help
 - public String getId()
 - public String getTitle()
 - public String getContent()
 4. public class History
 - public History(String id, String code, String order_list, String order_total, String date_time)
 5. public class Product
 - public long getProduct_id()
 - public String getProduct_name()
 - public String getCategory_id()
 - public String getCategory_name()
 - public double getProduct_price()
 - public String getProduct_status()
 - public String getProduct_image()
 - public String getProduct_description()
 - public String getCurrency_code()
 - public double getTax()
 - public int getProduct_quantity()

Utilities

Каталог «Utilities» содержит классы Java, которые служат для хранения определенных необходимых функций. Содержит во всех 9 классах:

1. public class AppBarLayoutBehavior extends AppBarLayout.Behavior: Он
 содержит различные действия, касающиеся панели приложения;
 - public AppBarLayoutBehavior();

- `public AppBarLayoutBehavior(Context context, AttributeSet attrs);`
 - `public void onNestedScroll(CoordinatorLayout coordinatorLayout, AppBarLayout child, View target, int dxConsumed, int dyConsumed, int dxUnconsumed, int dyUnconsumed, int type);`
 - `public void onNestedPreScroll(CoordinatorLayout coordinatorLayout, AppBarLayout child, View target, int dx, int dy, int[] consumed, int type);`
 - `private void stopNestedScrollIfNeeded(int dy, AppBarLayout child, View target, int type);`
2. `public class Constant`: он содержит различные значения, используемые в нашем приложении, такие как ссылка на базы данных и т. д.;
3. `public class DBHelper extends SQLiteOpenHelper`: все методы и команды, используемые для манипулирования данными в нашей базе данных, хранятся в этом классе. Эти методы:
- `public DBHelper(Context context);`
 - `public void createDataBase();`
 - `private boolean checkDataBase();`
 - `private void copyDataBase();`
 - `public void openDataBase();`
 - `public ArrayList<ArrayList<Object>>getAllData();`
 - `public ArrayList<ArrayList<Object>>getAllDataHistory();`
 - `public boolean isDataExist(long id);`
 - `public boolean isProductFavorites(long id);`
 - `public void remove_from_favorites(long id);`
 - `public void add_to_favorites(long product_id);`
 - `public void addData(long id, String product_name, int quantity, double total_price, String currency_code, String product_image);`
 - `public void addDataHistory(String code, String order_list, String order_total, String date_time);`
 - `public void deleteData(long id);`

- `public void deleteDataHistory(long id);`
 - `public void deleteAllData();`
 - `public void deleteAllDataHistory();`
 - `public void updateData(long id, int quantity, double total_price);`
 - `public boolean isDataHistoryExist(long id);`
 - `public boolean isPreviousDataExist();`
 - `public void close();`
4. `public class ItemOffsetDecoration`: Содержит следующие методы:
- `public ItemOffsetDecoration(int itemOffset);`
 - `public ItemOffsetDecoration(@NonNull Context context, @DimenRes int itemOffsetId);`
 - `public void getItemOffsets(Rect outRect, View view, RecyclerView parent, RecyclerView.State state);`
5. `public class MyDividerItemDecoration`: Содержит методы, которые нужно выполнить, чтобы разделить каждый элемент (объект отображается в приложении):
- `public MyDividerItemDecoration(Context context, int orientation, int margin);`
 - `public void setOrientation(int orientation);`
 - `public void onDrawOver(Canvas c, RecyclerView parent, RecyclerView.State state);`
 - `public void drawVertical(Canvas c, RecyclerView parent);`
 - `public void drawHorizontal(Canvas c, RecyclerView parent);`
 - `public void getItemOffsets(Rect outRect, View view, RecyclerView parent, RecyclerView.State state);`
6. `public class RoundedDrawable`: содержит методы, которые округляют рисунок:
- `public RoundedDrawable(Bitmap bitmap);`
 - `public static RoundedDrawable fromBitmap(Bitmap bitmap);`
 - `public static Drawable fromDrawable(Drawable drawable);`
 - `public static Bitmap drawableToBitmap(Drawable drawable);`

7. `publicclassRoundedImageView`: содержит методы, которые отображают изображения в виде круга.
- `public RoundedImageView(Context context);`
 - `public RoundedImageView(Context context, AttributeSet attrs);`
 - `public RoundedImageView(Context context, AttributeSet attrs, int defStyle);`
 - `public void setImageBitmap(Bitmap bm);`
 - `public void setImageResource(int resId);`
 - `public void setImageURI(Uri uri).`
8. `publicclassSharedPreferences`: класс, содержащий методы для получения значений профиля по умолчанию
- `public SharedPreferences(Context context);`
 - `private String getString(int string_id);`
 - `public String getYourName();`
 - `public String getYourEmail();`
 - `public String getYourPhone();`
 - `public String getYourAddress();`
9. `publicclassUtils`: методы для проверки интернет-соединения, получения текущего времени и проверки достоверности электронной почты находятся в этом классе.
- `public Utils(Context context)`
 - `public static boolean isNetworkAvailable(Activity activity)`
 - `public static String getFormattedDateSimple(String date_str)`
 - `public static String getFormattedDate(String date_str)`
 - `public static boolean isValidEmail(String email)`

3 ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ

После разработки нашего приложения, приложение, наконец, готово к использованию. Сначала нужно проверить, хорошо ли он работает и нужно ли вносить изменения. Для этой задачи нам понадобится эмулятор. Мы будем использовать genymotion в качестве эмулятора.

При первой загрузке приложения появится экран загрузки на некоторое время. И тогда приложение отображает главную страницу.

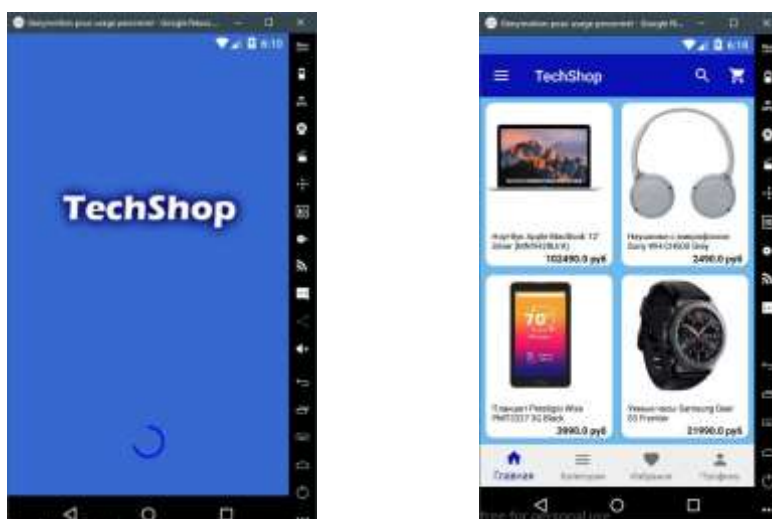


Рисунок 3.1 – Экран загрузки приложения и Начальный экран приложения.

Наше приложение имеет меню внизу, и для доступа к дополнительным функциям мы добавили ящик.

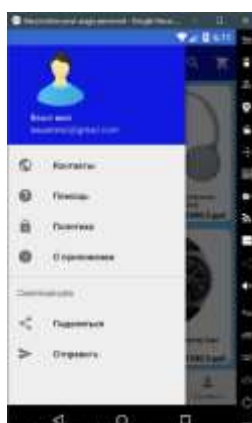


Рисунок 3.2 – Ящик дополнительного меню.

В ящике находятся такие функции, как: контакты, помощь, политика конфиденциальности, информация о приложении, поделиться и отправить.

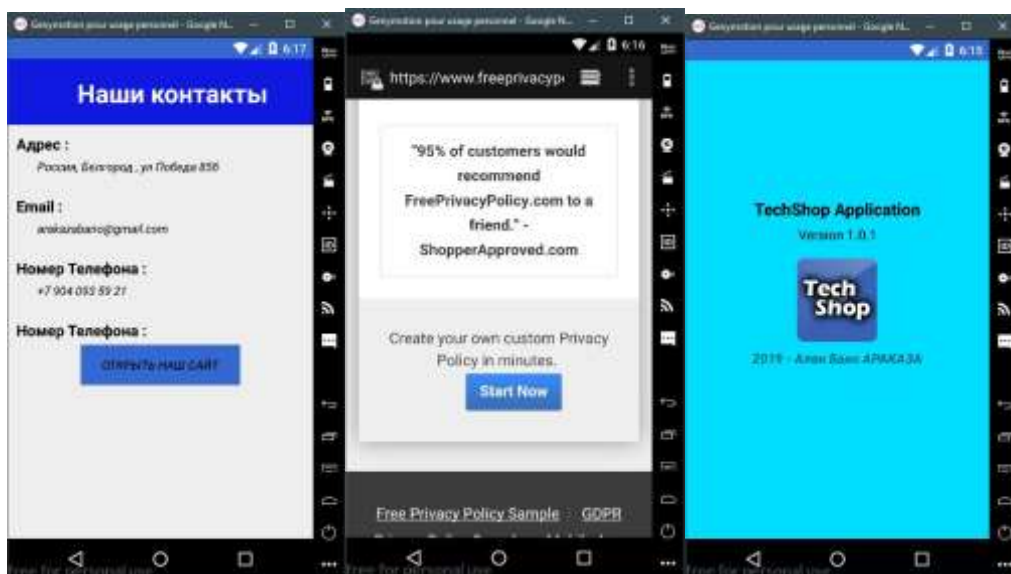


Рисунок 3.3 – Экран контакта, экран политики конфиденциальности и экран информации о приложении.

Меню Помощь предлагает несколько вариантов помощи, и каждый открывается в новом окне.

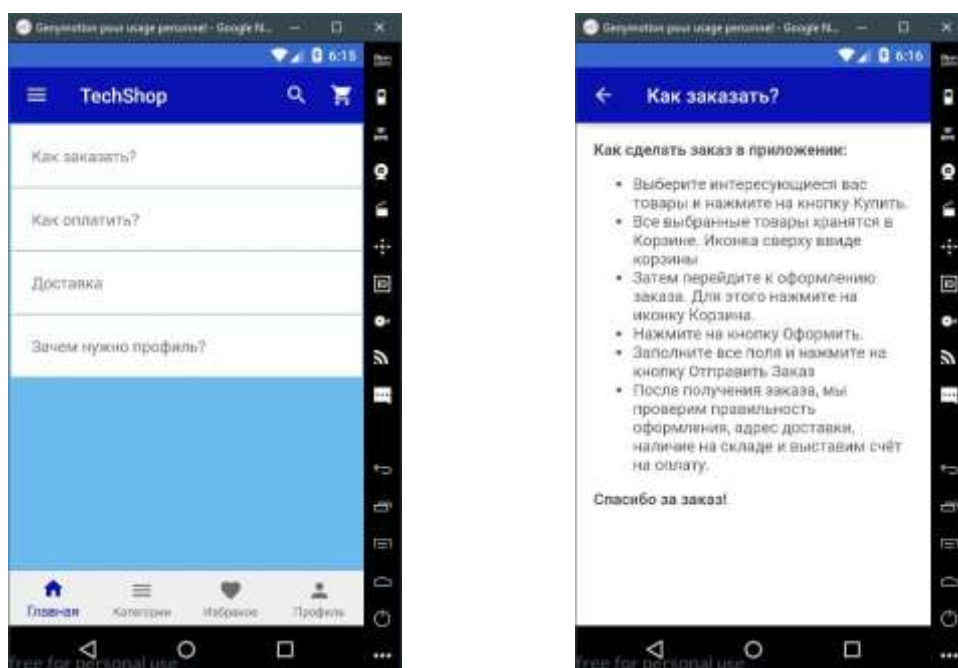


Рисунок 3.4 – Экран Помощь и экран “Как заказать”.

В меню «Поделиться» вы увидите различные варианты «поделиться» приложением, а в меню «Отправить» отображается небольшое сообщение.

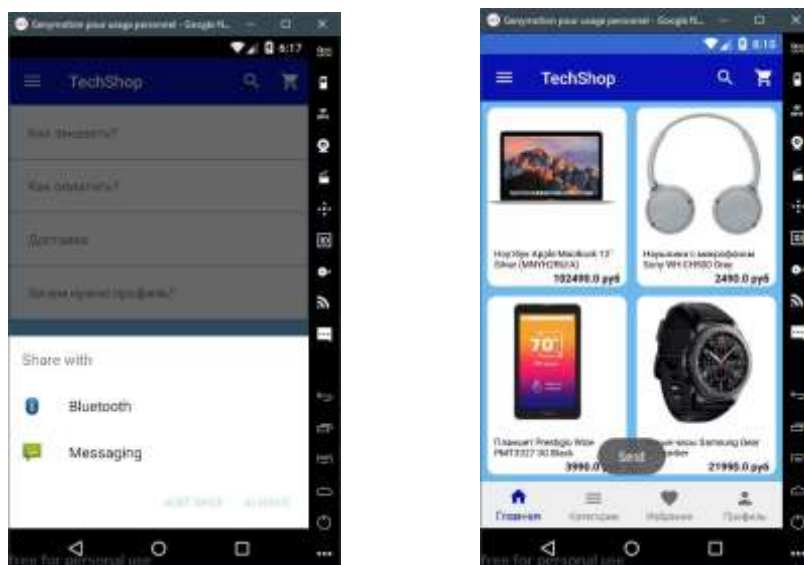


Рисунок 3.5 – Экран Помощь и экран “Как заказать”.

На экране главной страницы есть панель с функцией поиска и «Корзина»

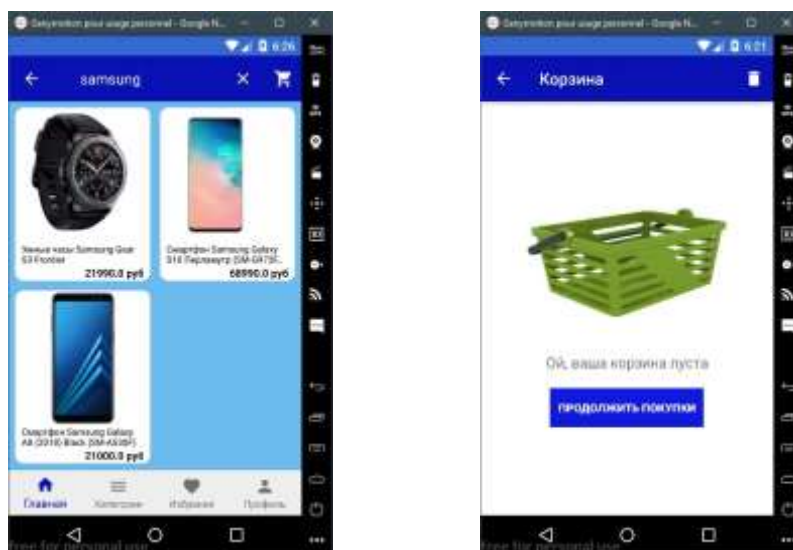


Рисунок 3.6 – Экран главной страницы с результатом поиска по слову «Samsung», экран «корзина» пустой и экран «корзина» с товарами.

Главное меню внизу дает нам 4 варианта: Главной, Категория, Избранное и Профиль. Первые 3 варианта позволяют нам сделать выбор товара.

- Выбор «Главная»: показы всех товаров
- Выбор «категория»: отображает товар по категории

- Выбор «Избранное»: отображает товары, которые мы уже выбрали в качестве избранных

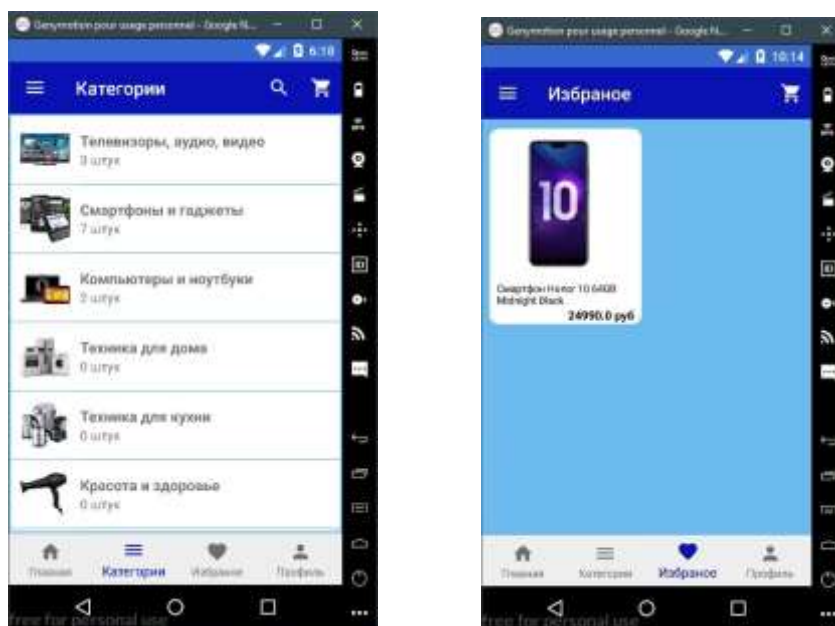


Рисунок 3.7 – Экран«Категория» и экран «Избранное».

- Выбор «Профиль»: отображает информацию о пользователе, а также историю выполненных заказ. Информация о пользователе может быть отредактирована

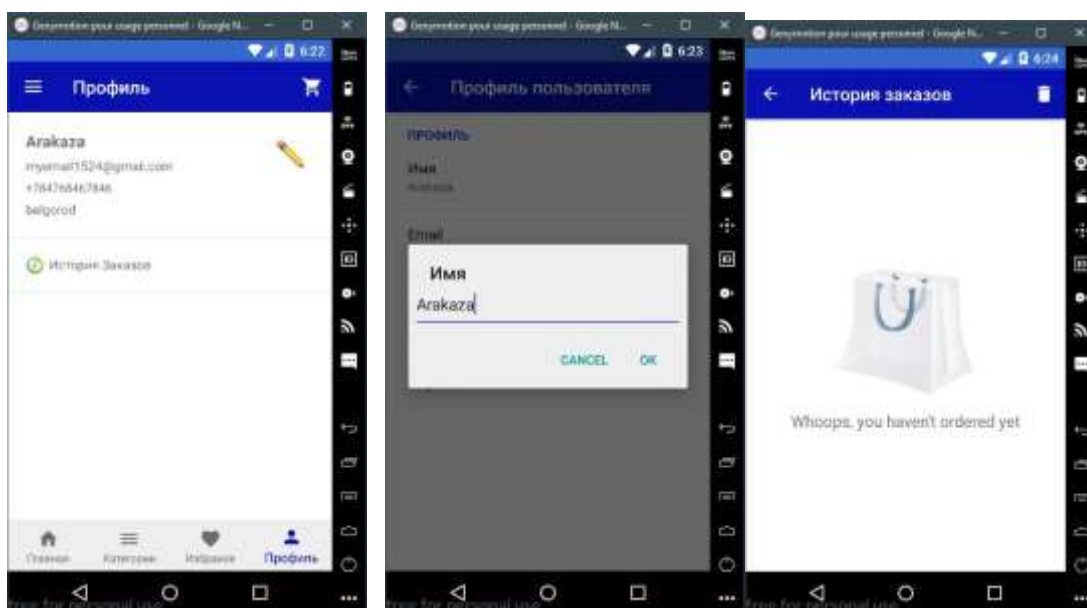


Рисунок 3.8 – Экран«Профиль», экранредактирован«Профиль» и экран «История заказа» пустой.

При выборе товара (детальный экран товара) нам предлагается два варианта: Добавить товар в «Корзину». Мы должны нажать на кнопку справа и указать количество товаров. Добавить товар в избранное, нажав на кнопку слева.

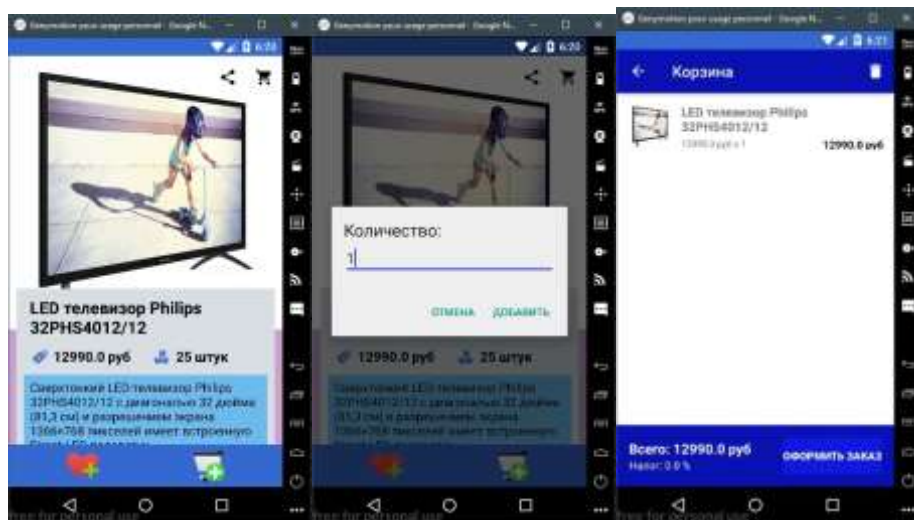


Рисунок 3.9 – Экран детали товара, экран указания количество товаров и экран «корзина» с товаром.

Выполнение заказа производится из «Корзины» нажатием кнопки «Оформить заказ». Откроется окно для дополнения информации о пользователе. После этого мы должны подтвердить заказ.

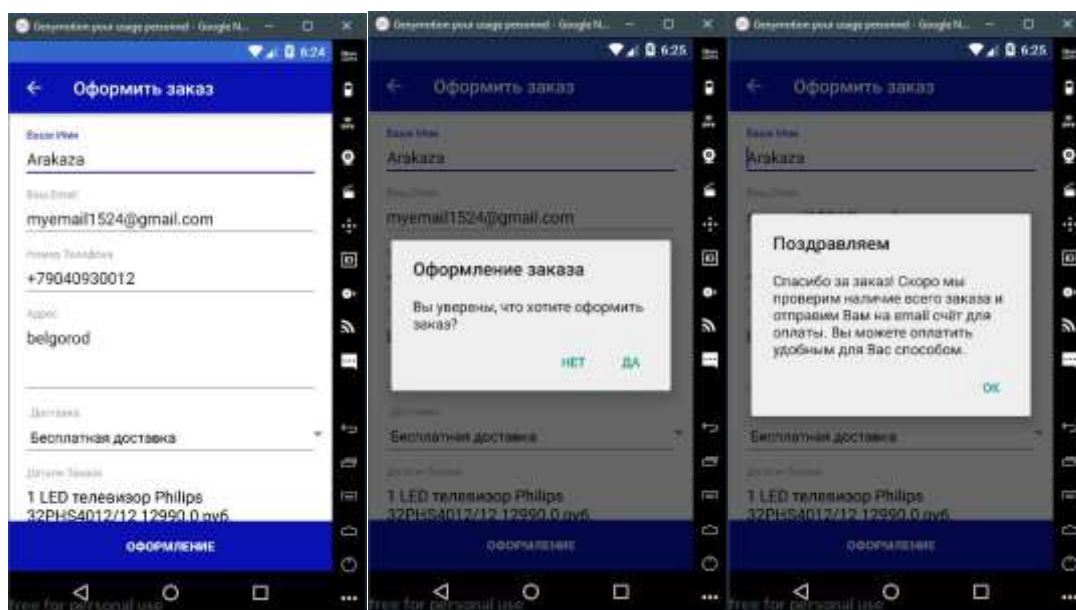


Рисунок 3.10 – Экран оформление заказ, экран подтверждение заказ и экран поздравления.

Как только заказ выполнен, «корзина» становится пустой и в истории заказов появляется выполненный заказ.

При нажатии на уже выполненной команды можно увидеть подробности этого заказа.

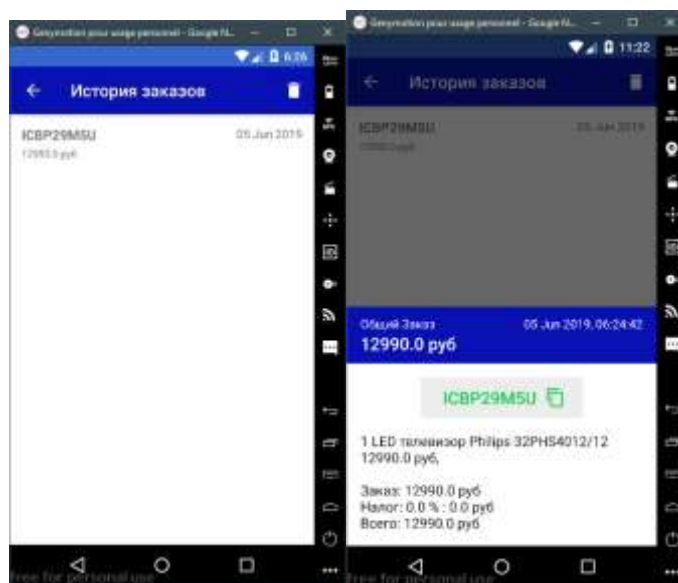


Рисунок 3.11 – Экраны «История заказов» и экранподробности заказа.

ЗАКЛЮЧЕНИЕ

В представленной работе видно, как может планироваться разработка интернет-приложения журнала TechShop. Было создано приложение, способное отображать доступные товары. Приложение имеет несколько функций, таких как отображение товаров в соответствии с их категориями, создание профиля клиента, возможность размещения заказа на товар, поиск товара и т. д. И все это было сделано с использованием баз данных.

Эта работа позволила:

- ✓ расширить знания объектно-ориентированного программирования;
- ✓ улучшить наши знания о создании приложений с помощью Android Studio;
- ✓ создать две базы данных для хранения данных из магазина и научиться обмениваться данными между сервером и приложением;.
- ✓ разработать программу приложения для интернет-магазина благодаря языку программирования Java.

Цели и задачи представленной выпускной квалификационной работы достигнуты. Разработанное приложение обладает всеми необходимыми функциями для покупок и вполне может стать основой для операции покупки товаров в интернет-магазине.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов / Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано, 2-е издание Издательство: Питер, 2016. — 640 с.
2. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. / П. Дейтел, Х. Дейтел, А. Уолд 3-е изд. — СПб.: Питер, 2016. — 512 с.
3. Баканов М.В., Базы данных. Системы управления базами данных, [текст] / Стружкин Н.П., Годин В.В. – Мн.:РИПО, 2016.. – 268 с. – ISBN 978-985-503-558-0
4. Радченко Г.И., Объектно-ориентированное программирование / Г.И. Радченко, Е.А. Захаров. – Ч.: Издательский центр ЮУрГУ, 2013. – 167 с.
5. Васильев А. Н., Java объектно-ориентированное программирование / Васильев А.Н. – СПб: Питер, 2013. – 400 с.
6. Кирютенко Ю.А., Объектно-ориентированное программирование/ Кирютенко Ю.А.– М.: Вузовская Книга, 2013. – 400 с.
7. Бурнет Эд. Привет, Android! Разработка мобильных приложений / Эд Бурнет СПб: Питер. 2012. — 256 с.
8. Березовская Ю.В., Юфрякова О.А., Вологодина В.Г. и др. Введение в разработку приложений для ОС Android / Ю.В. Березовская, О.А. Юфрякова, В.Г. Вологодина, О.В. Озерова, Э.Е. Куликов, Э.Е. Латухина , М.Н. Пархимович - 2-е издание М.: НОУ "ИНТУИТ", 2016. - 434 с.
9. Колисниченко Д.Н. Программирование для Android 5. Самоучитель/ Д.Н. Колисниченко СПб.: БХВ-Петербург, 2015г.-303с.
10. Латухина Е.А., Юфрякова О.А., Березовская Ю.В., Носов К.А. Разработка приложений для смартфонов на ОС Android / Е.А. Латухина, О.А. Юфрякова, Ю.В. Березовская, К.А. Носов – М.: Национальный Открытый Университет "ИНТУИТ", 2016. – 252 с.
11. Гриффитс Дон, Гриффитс Дэвид. HeadFirst. Программирование для Android. / Дон Гриффитс, Дэвид Гриффитс СПб.: Питер, 2016. — 704 с.

12. Федотенко М. А. Разработка мобильных приложений. Первые шаги / М. А. Федотенко; под ред. В. В. Тарапаты. — М.: Лаборатория знаний, 2019. — 335 с.: ил. — (Школа юного программиста).
13. Рето Майер Android 4. Программирование приложений для планшетных компьютеров и смартфонов / Рето Майер. - М.: Эксмо, 2013. - 816 с.
14. Кей С. Хорстманн. Java. Библиотека профессионала. Том 1./ Кей С. Хорстманн. Издательство: Вильямс, Основы. 10-е издание, 2016.
15. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений / Буч Гради Издательство Вильямс, 3-е издание, 2008
16. Вайсфельд М. Объектно-ориентированное мышление. / М. Вайсфельд — СПб.: Питер, 2014. — 304 с.
17. Kreibich J. Using SQLite / Jay Kreibich. Издательство: O'ReillyMedia, 2010, с 530
18. Атре, Ш. Структурный подход к организации баз данных;/ Ш. Атре, М.: Финансы и статистика -, 2009. - 317 с.
19. Ахаян, Рубен; Горев, Андрей; Макашарипов, Сергей Эффективная работа с СУБД; СПб: Питер - , 2014. - 704 с.
20. Веллинг, Люк Разработка веб-приложений с помощью PHP и MySQL / Люк Веллинг , Лора Томсон. - М.: Вильямс, 2013. - 848 с.
21. Кузнецов, М. Объектно-ориентированное программирование на PHP / М. Кузнецов, И. Симдянов. - М.: БХВ-Петербург, 2012. - 608 с.
22. Хабибуллин, Ильдар Самоучитель Java / Ильдар Хабибуллин. - М.: БХВ-Петербург, 2014. - 768 с.
23. Савитч, Уолтер Язык Java. Курс программирования / Уолтер Савитч. - М.: Вильямс, 2010. - 928 с.
24. Сеттер, Р. В. Изучаем Java на примерах и задачах / Р.В. Сеттер. - М.: Наука и техника, 2016. - 240 с.
25. Берд, Барри Java для чайников / Барри Берд. - М.: Диалектика / Вильямс, 2013. - 521 с.

26. Голощапов, Алексей Google Android. Программирование для мобильных устройств / Алексей Голощапов. - М.: БХВ-Петербург, 2012. - 448 с.
27. Роджерс, Рик Android. Разработка приложений / Рик Роджерс и др. - М.: ЭКОМ Паблишерз, 2010. - 400 с.