

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
_____ О.В. Непомнящий
подпись инициалы, фамилия
« _____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Интернет магазин. Мобильное приложение клиента.
тема

Руководитель	_____	<u>доцент, канд. физ.-мат. наук</u>	<u>К. В. Коршун</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>В. А. Алексеев</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	<u>доцент, канд. техн. наук</u>	<u>В. И. Иванов</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2019

СОДЕРЖАНИЕ

Введение.....	3
1 Анализ технического задания.....	5
1.1 Обзор существующих аналогов.....	5
1.1.1 100sp	6
1.1.2 24-OK.RU	7
1.1.3 СП 2.0.....	9
1.1.4 Optwear Mobile	10
1.1.5 Итоги обзора аналогов	12
1.2 Проектирование интерфейса.....	12
1.3 Выбор средств разработки	14
1.3.1 Язык программирования Java.....	14
1.3.2 Реактивное программирование с RX Java 2.....	15
1.3.3 Внедрение зависимостей с Dagger 2.....	16
1.3.4 Apollo-Android.....	17
2 Реализация и документация.....	18
2.1 Реализация структуры приложения	18
2.2 Реализация графического интерфейса	22
2.3 Примеры интерфейса приложения.....	24
2.4 Инструкция пользователя	26
2.5 Инструкция разработчика	28
Заключение	29
Список использованных источников	30

ВВЕДЕНИЕ

В течении последних лет активное развитие получили различные сервисы совместных покупок товаров через интернет. Подобные сервисы работают по довольно простому принципу – участники собираются вместе и покупают оптовую партию товара у поставщика, таким образом каждый из участников приобретает для себя данный товар по выгодной стоимости. В последние годы подобные сервисы активно развиваются и постоянно увеличивают количество клиентов, что в свою очередь подталкивает развитие информационных систем управления интернет-магазином.

Одним из таких сервисов совместных покупок является «optwear.ru», который был создан в Красноярске и функционирует с 2010 года. В данный момент этот сервис имеет значительно расширенный функционал по сравнению с моментом его создания: совершение совместных покупок, форум для участников и организаторов, личные сообщения, рейтинги организаторов, избранные товары, поиск и фильтрация товаров по различным критериям и многое другое. Данный сервис помимо сайта представлен также в виде мобильных приложений для операционных систем iOS и Android.

Целью выпускной квалификационной работы является разработка нового мобильного приложения «optwear.ru» для устройств под управлением операционной системы Android. Разработанная программа обладает следующими особенностями:

- использование системы авторизации и регистрации пользователей, корректная обработка взаимодействия неавторизованного пользователя с функционалом приложения, требующим авторизации (личные сообщения, корзина покупок, избранные товары и т.д.), подтверждение номера телефона при регистрации;
- корректная обработка отсутствия соединения с интернетом и проблем на стороне сервера;

- экономия интернет-трафика пользователя (использование GraphQL, кэширование запросов, версионирование редко изменяющихся данных и т.д.), оптимальное использование оперативной и постоянной памяти на устройстве пользователя;

- выполнение всех операций с загрузкой и сохранением данных в отдельных потоках приложения, корректное взаимодействие между основным (User Interface) потоком и дополнительными;

- корректная работа с жизненным циклом View-элементов при изменении ориентации экрана устройства или сворачивании приложения – сохранение необходимых данных, корректное отображение интерфейса;

- взаимодействие с другими приложениями, установленными на устройстве пользователя – возможности открыть ссылку на социальные сети сервиса в соответствующих приложениях, поделиться ссылкой на какой-либо товар или закупку, используя сторонние приложения, и возможность открыть ссылку на сайт «optwear.ru» в мобильном приложении.

Для достижения поставленной цели в работе решаются следующие задачи:

- 1) Провести анализ существующих аналогов и старого мобильного приложения сервиса «optwear»;

- 2) Проанализировать задание на выпускную квалификационную работу и выполнить проектирование архитектуры приложения, выбрать необходимые инструменты разработки;

- 3) Реализовать приложение с учётом всех технических требований, придерживаясь разработанных ранее архитектурных решений;

- 4) Составить инструкции по сборке, запуску и использованию написанного мобильного приложения.

1 Анализ технического задания

В соответствии с техническим заданием необходимо разработать мобильное приложение для сервиса совместных покупок «optwear.ru». Для того, чтобы выполнить требования технического задания, в первую очередь, был произведен обзор существующих сервисов, имеющих мобильные приложения для операционной системы Android, для изучения типичных для таких программ функций и возможностей. Помимо прочего, старое мобильное приложение сервиса «optwear.ru» было также внимательно изучено и проанализировано.

1.1 Обзор существующих аналогов

На сегодняшний день существует целый ряд сервисов для совершения коллективных покупок, имеющих собственное мобильное приложение для устройств под управление операционной системы Android. Рассмотрим самые популярные и близкие по функционалу к разрабатываемому мобильному приложению варианты. В таблице 1 приведены основные сведения из магазина мобильных приложений «Play Market» о наиболее подходящих для сравнения сервисах совместных покупок, а также о старом мобильном приложении сервиса «optwear.ru».

Таблица 1 – Основные сведения об аналогичных мобильных приложениях

Название	Дата публикации	Последнее обновление	Количество скачиваний	Количество Оценок	Средняя оценка
100sp	05.08.2015	07.09.2019	более 100 тыс.	3782	4.4
24-OK.RU	03.08.2017	26.12.2018	более 10 тыс.	96	3.1
СП 2.0	20.08.2014	30.04.2019	более 100 тыс.	770	4.2
Optwear Mobile	05.08.2015	23.10.2018	более 10 тыс.	294	3.6

1.1.1 100sp

100sp.ru – один из первых и самых крупных на данный момент сервисов совместных покупок в России, который начал свою работу в 2007 году. В данном сервисе представлено более 1.5 млн. товаров и более 6 тыс. брендов [1]. С 2015 года данный сервис также представлен в виде мобильного приложения для устройств, работающих под управлением операционной системы Android. Интерфейс данного приложения представлен на рисунке 1.

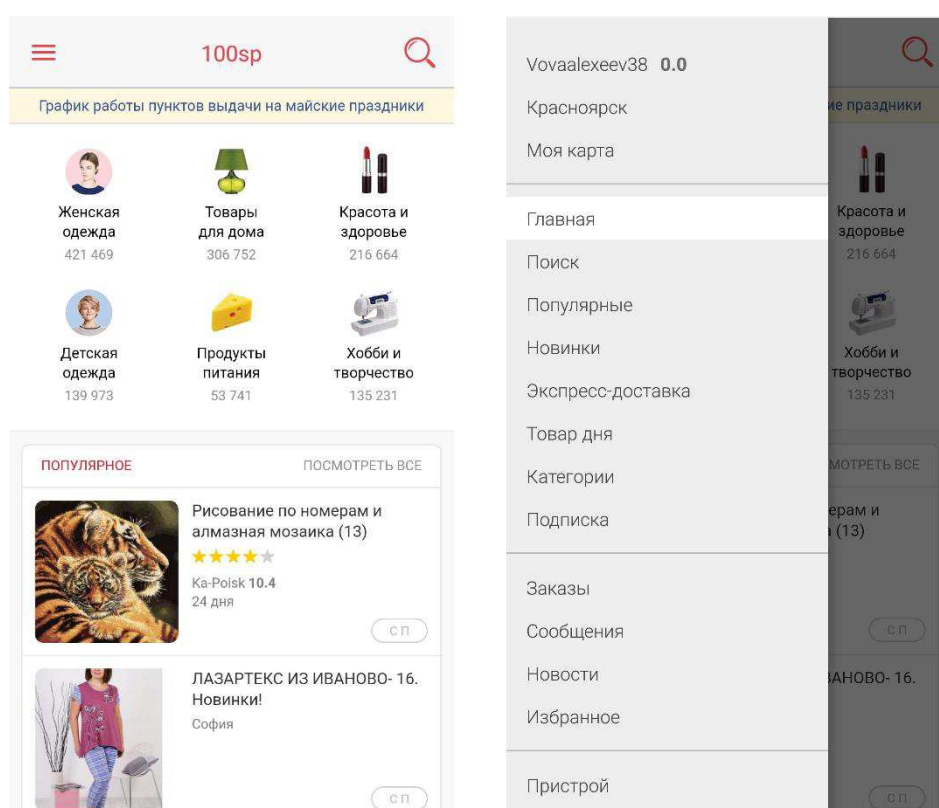


Рисунок 1 – Интерфейс главной страницы и меню навигации мобильного приложения 100sp

Данное мобильное приложение имеет широкий набор функций, в число которых входят поиск, личные сообщения, карты с пунктами выдачи товара, калькулятор расчёта стоимости доставки и многое другое. Но помимо прочего данное приложение не лишено некоторых недостатков:

- многие экраны приложения выполнены в виде окна браузера, в котором открываются страницы с сайта, а не в виде элементов из библиотеки Android SDK – это делает интерфейс приложения не таким гибким и удобным и в отдельных случаях приводит к некорректному отображению данных;
- малое количество параметров для фильтрации товаров – например, нет возможности выбрать нужный размер, цвет, бренд, конкретного организатора и т.д.;
- нет возможности поделиться ссылкой на товар из приложения и открыть товар в приложении по ссылке на сайте.

1.1.2 24-OK.RU

24-OK.RU – это сервис совместных покупок, основанный в Красноярске в 2012 году. В данный момент сервис насчитывает более 215 тыс. зарегистрированных пользователей и за время своей работы обработал более 20 млн. заказов [2]. Сразу после запуска приложения открывается страница авторизации, показанная на рисунке 2, не пройдя которую нет возможности получить доступ к функционалу сервиса.

Из особенностей и положительных характеристик хотелось бы отметить простой и интуитивный пользовательский интерфейс, яркий и запоминающийся корпоративный дизайн, удобная навигация, что и показано на рисунке 3.

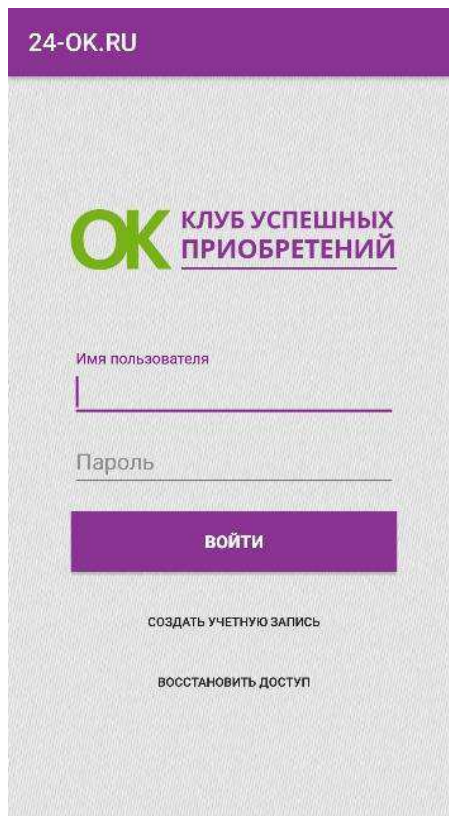


Рисунок 2 – Начальная страница приложения 24-OK.RU

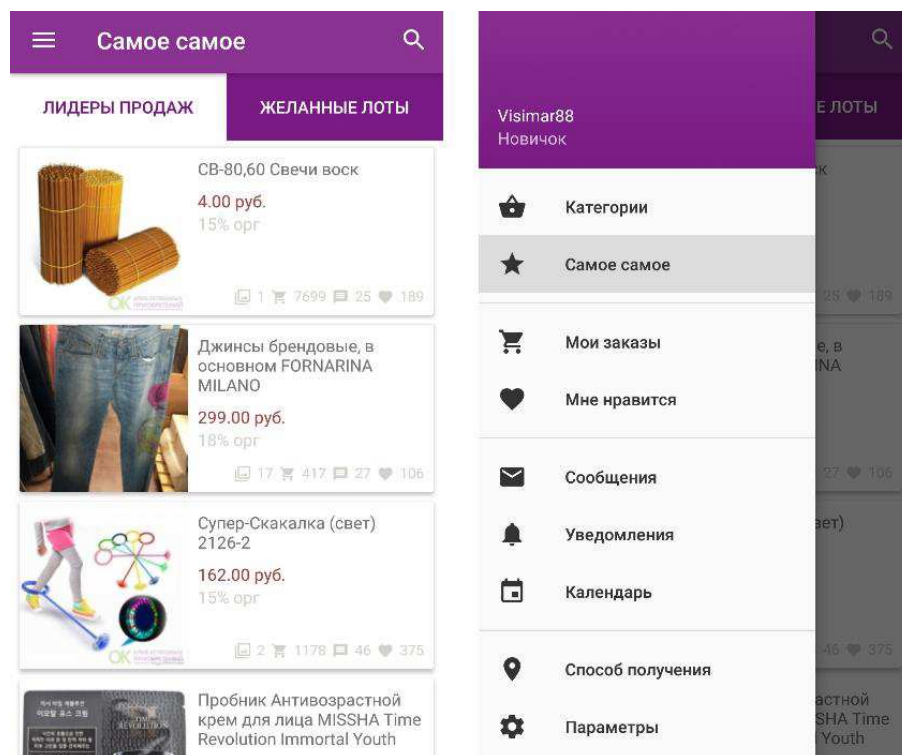


Рисунок 3 - Интерфейс главной страницы и меню навигации мобильного приложения 24-OK.RU

Из основных недостатков данного приложения можно выделить следующее:

- отсутствие каких-либо параметров для фильтрации на всех страницах, где представлено множество товаров – товары в выбранной закупке, переход к товарам через иерархию категорий, поиск;
- некорректная работа большого количества страниц – бесконечная загрузка, после которой приложение самостоятельно закрывается, появление пустых экранов.

1.1.3 СП 2.0

Sp2all.ru – проект, официально запущенный 20 ноября 2010 года, насчитывающий более 128 тыс. зарегистрированных пользователей [3]. Мобильное приложение данного сервиса для ОС Android называется СП 2.0, его главная страница представлена на рисунке 4.

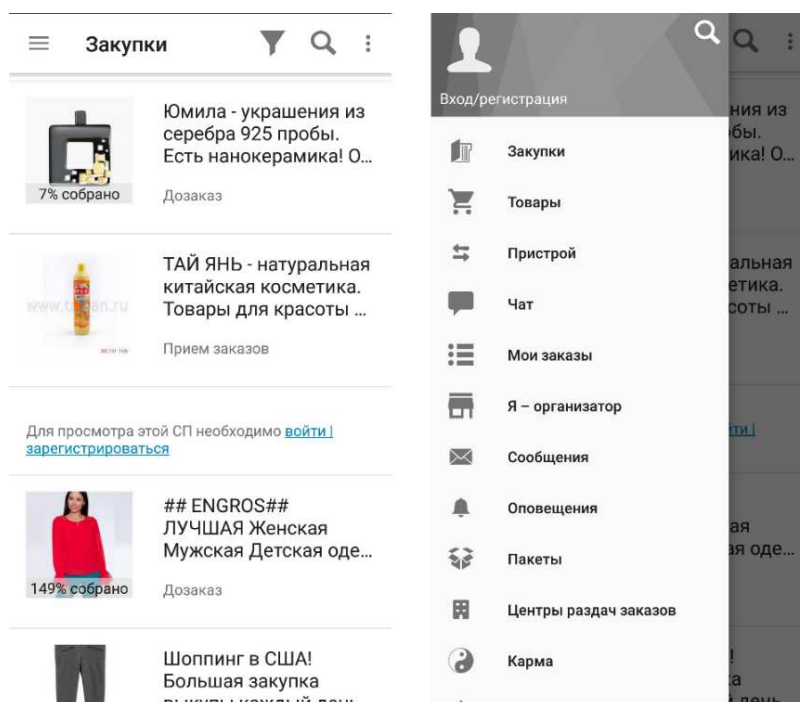


Рисунок 4 - Интерфейс главной страницы и меню навигации мобильного приложения СП 2.0

После подробного ознакомления с его функционалом можно выделить следующее:

- пользовательский интерфейс малофункционален и не является интуитивным, не используются принципы Material Design [11], что негативно выделяет его на фоне других приложений;
- небольшое количество параметров фильтрации товаров, недостаточно гибкий поиск товаров;
- некоторые действия в приложении приводят к его некорректной работе или закрытию и перезапуску.

1.1.4 Optwear Mobile

Мобильное приложение сервиса «optwear.ru» для ОС Android, именуемое Optwear Mobile, было выпущено в 2015 году и с того момента множество раз обновлялось и дополнялось новым функционалом [4]. Главная страница приложения представлена на рисунке 5.

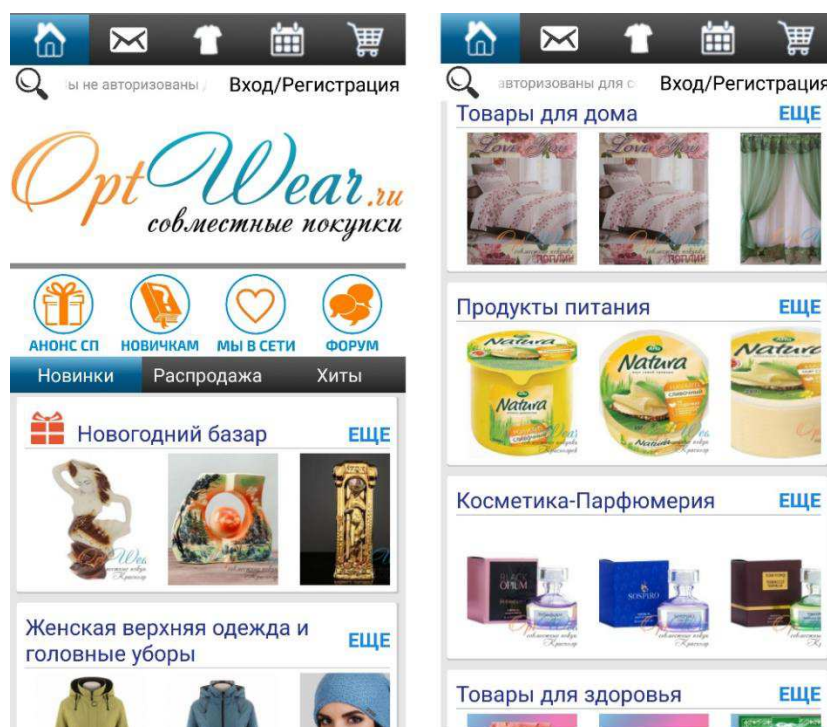


Рисунок 5 – Главная страница мобильного приложения Optwear Mobile

Рассмотрим основные проблемы, по причине которых было принято решение создать новое мобильное приложение:

- устаревший пользовательский интерфейс, некорректная и неочевидная работа навигации, отсутствие фирменного стиля компании;
- огромное количество неверных архитектурных решений при проектировании приложения сделали данный проект крайне неэффективным с точки зрения доработки и внесения новых функций;
- некорректная работа большинства функций приложения – личные сообщения, корзина, календарь, регистрация, уведомления и многое другое;
- неоптимальность с точки зрения использования ресурсов мобильного устройства, вследствие чего медленная работа и большое количество занимаемой постоянной и используемой оперативной памяти, частые зависания и принудительные перезапуски;
- отсутствует функционал работы со ссылками на товар – нельзя поделиться ссылкой на товар из приложения или открыть в нём действующую ссылку;
- многие страницы реализованы в виде окна браузера с открывающимися в них ссылками на сайт, что делает их неудобными в использовании и приводит к некорректному отображению элементов интерфейса;
- нет многих необходимых функций – настроек приложения, работы с «избранными» товарами, уведомлений, возможности обратной связи с разработчиками;
- низкая оценка в «Play Market» и малое количество скачиваний более чем за 3 года после выпуска мобильного приложения.

1.1.5 Итоги обзора аналогов

В ходе обзора нескольких наиболее актуальных мобильных приложений для сервисов совместных покупок были сделаны следующие выводы:

1) Функционально наиболее соответствующим поставленным задачам примером является приложение «100sp», но полным набором необходимых параметров не обладает ни один из аналогов;

2) С точки зрения пользовательского интерфейса наиболее подходящие решения используются в приложениях «100sp» и «24-OK.RU», но в них также есть множество элементов, требующих изменений или доработки;

3) Мобильное приложение, которое в данный момент используется сервисом «optwear.ru», крайне сложно и не оптимально поддерживать и развивать, оно имеет множество критичных ошибок и недостающих функций.

С учётом всего вышеперечисленного было принято решение создать полностью новое приложение «optwear.ru» для ОС Android, в котором будут присутствовать все необходимые функции и спроектирован современный пользовательский интерфейс с применением концепций Material Design [11].

1.2 Проектирование интерфейса

Основные требования, предъявляемые к визуальной части создаваемого мобильного приложения, заключались в современном дизайне, интуитивности и простоте использования. Также было необходимо создать единый визуальный стиль приложения с использованием концепции Material Design [11] и фирменной цветовой схемой.

На основе технического задания были созданы макеты интерфейса для базовых экранов мобильного приложения. На рисунках 6 и 7 изображены некоторые из составленных макетов.

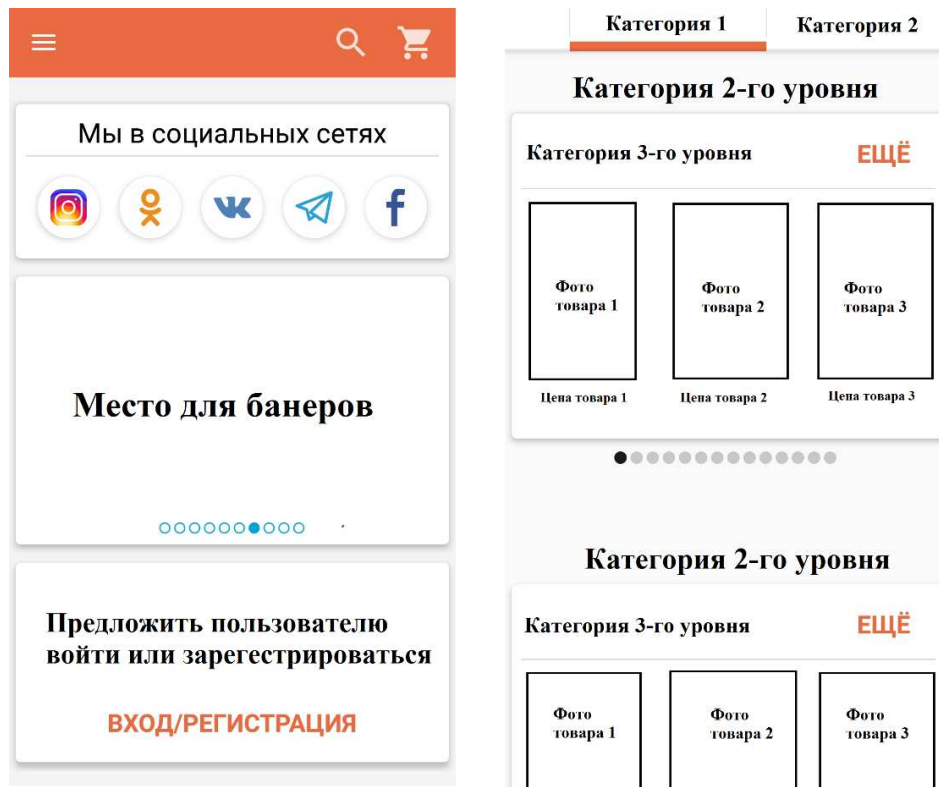


Рисунок 6 – Макет интерфейса главной страницы приложения

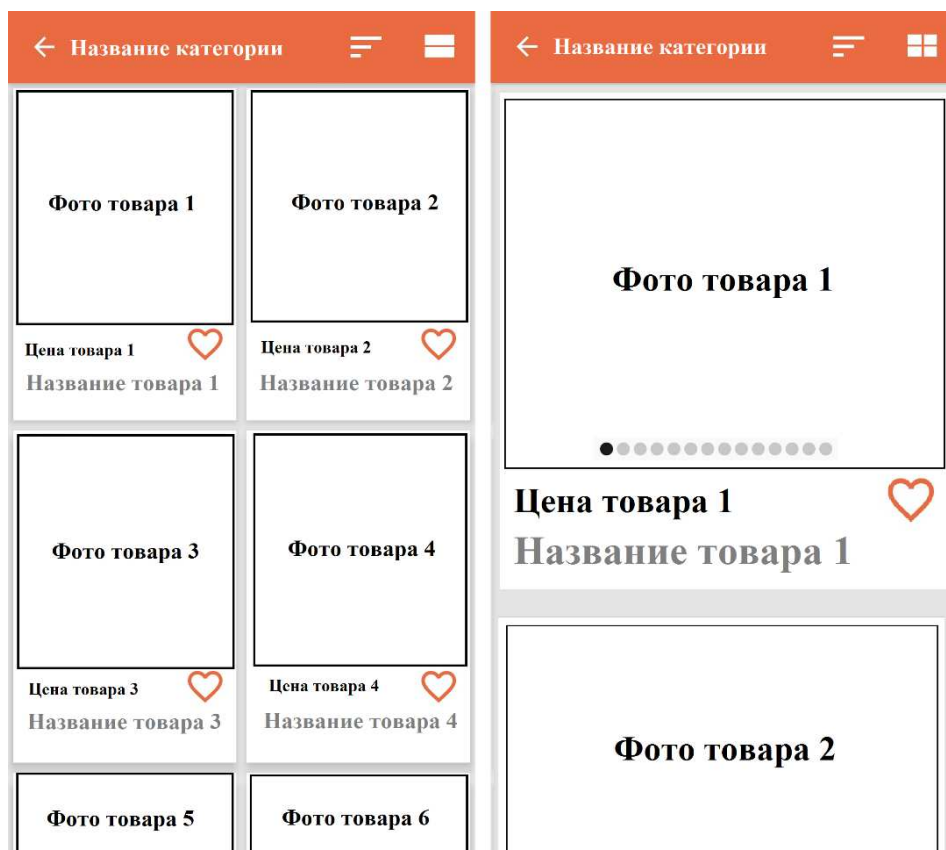


Рисунок 7 – Макет интерфейса страницы с товарами

1.3 Выбор средств разработки

Приложение должно работать на мобильных устройствах под управлением операционной системы Android и обеспечивать графический интерфейс пользователя. Помимо прочего, используемые в разработке инструменты должны быть достаточно гибкими, качественно документированными и удобными в применении. Также немаловажным критерием является активное развитие того или иного инструмента со стороны его разработчиков, так как ОС Android быстро развивается и обновляется и используемые программные модули должны поддерживать эти изменения и продолжать корректно работать, приобретая при этом новый функционал с последующими обновлениями.

1.3.1 Язык программирования Java

Есть множество различных языков программирования, при помощи которых можно написать мобильное приложение, которое будет запускаться на ОС Android, вот самые популярные из них:

- Java;
- Kotlin;
- C# (Xamarin);
- Python (Kivy);
- Javascript (PhoneGap Build, Adobe Cordova, React Native).

Язык программирования Java является мультипарадигменным (преимущественно объектно-ориентированным) сильно типизированным кроссплатформенным языком программирования общего назначения, первая версия которого была выпущена в 1995 году компанией Sun Microsystems. Программы на Java транслируются в байт-код, выполняемый виртуальной

машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор [5].

Так как Java является одним из наиболее популярных языков программирования, а также потому что большинство учебных пособий и примеров написаны на этом языке, выбор был сделан в пользу этого варианта.

Язык Java активно используется для создания мобильных приложений для операционной системы Android. При этом программы компилируются в нестандартный байт-код для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google [6].

1.3.2 Реактивное программирование с RX Java 2

Реактивное программирование — парадигма программирования, ориентированная на потоки данных и распространение изменений. Применяя «реактивные» инструменты в разработке можно создавать потоки данных или событий, комбинировать и трансформировать данные при помощи различных операторов, а также «подписываться» на события для совершения различных действий с данными в момент их получения.

RxJava 2 - это Java-реализация Reactive Extensions: библиотеки для составления асинхронных и событийных программ с использованием наблюдаемых последовательностей. Данная библиотека расширяет шаблон проектирования «наблюдатель» (объект автоматически оповещает «подписанные» на него объекты при изменении своего состояния при помощи вызова одного из их методов) для поддержки последовательностей данных или событий и добавляет операторы, которые позволяют комбинировать последовательности вместе, упрощая работу с низкоуровневой

многопоточностью, синхронизацией, безопасностью потоков и декларативностью структуры данных [7].

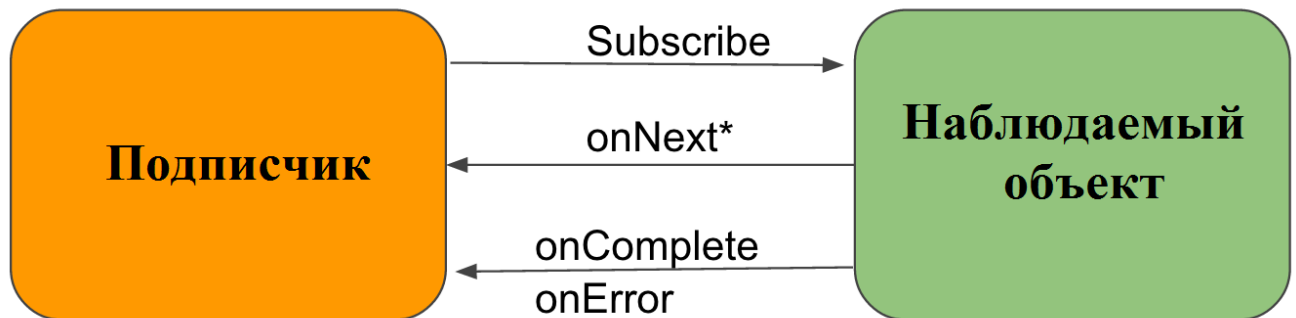


Рисунок 8 – Базовая схема взаимодействия объектов в RxJava

1.3.3 Внедрение зависимостей с Dagger 2

Внедрение зависимости (англ. Dependency injection, DI) – это процесс предоставления внешней зависимости программному компоненту. Является специфичной формой «инверсии управления» (англ. Inversion of control, IoC), когда она применяется к управлению зависимостями. В полном соответствии с принципом единственной ответственности объект отдаёт заботу о построении требуемых ему зависимостей внешнему, специально предназначенному для этого общему механизму, самостоятельно не предпринимая никаких шагов для получения или создания зависимостей, а лишь предоставляя для их установки специальные методы или принимая их в качестве аргументов в своём конструкторе. Внедрение зависимостей делает код более гибким, так как становится легче создавать альтернативные реализации какого-либо сервиса внутри приложения, что особенно помогает при тестировании.

Dagger 2 – это полностью статическая библиотека для внедрения зависимостей в Java и Android, работающая на этапе компиляции. Данный инструмент активно использует аннотации Java для генерации кода по

внедрению зависимостей, что делает его достаточно гибким и удобным в использовании. Для использования данной библиотеки используются два основных типа компонентов:

- модули – это классы, куда помещается код создания тех или иных внедряемых объектов. Каждый модуль должен включать в себя объекты, близкие по смыслу – например, отдельный модуль для работы с локальными данными пользователя;

- компоненты – это классы-посредники между тем, кому необходимо внедрение зависимостей и модулями. Компоненты знают, какой модуль или компонент отвечает за создание какого-либо объекта.

1.3.4 Apollo-Android

GraphQL – это язык запросов для API на получение или изменение данных, являющийся альтернативой REST API. Он не специфичен для одной платформы и работает для всех типов клиентов, включая и мобильные приложения для ОС Android. GraphQL располагается между сервером и клиентом и помогает запрашивать данные более оптимизированным способом.

Apollo-Android является совместимым с GraphQL клиентом, который генерирует модели Java из стандартных запросов. Эти модели предоставляют безопасный API для работы с GraphQL сервером. Apollo помогает сохранять запросы вместе, упорядоченными и легкодоступными из кода Java.

Помимо вышеперечисленного Apollo имеет встроенные инструменты для совместной работы с RxJava 2, а также несколько возможных вариантов кэширования полученных из сети данных. Всё это делает его незаменимым инструментом для взаимодействия с GraphQL сервером из Android приложения [8].

2 Реализация и документация

2.1 Реализация структуры приложения

При разработке сложного и объёмного проекта часто можно столкнуться со структурными проблемами в организации взаимодействия элементов программы и зависимостях между ними. Как следствие – плохо структурированный код довольно трудно поддерживать и обновлять, он плохо читается и часто является непригодным для тестирования. Решением таких проблем в значительной мере является использование архитектурных шаблонов проектирования.

При разработке было принято решение использовать архитектурный шаблон MVP (Model View Presenter), схема реализации которого представлена на рисунке 9, по следующим причинам:

- чёткое разделение бизнес-логики приложения и его визуальной части – классы, отвечающие за графическую составляющую, не имеют доступа ни к каким компонентам, кроме соответствующего им Presenter'а. Это также сильно упрощает процесс тестирования отдельных функций и классов программы;
- простота и удобство реализации – большое количество разработчиков знакомы с принципами работы данного архитектурного шаблона, что упрощает возможную в будущем совместную работу над проектом нескольких программистов;
- значительное увеличение скорости разработки по причине лучшей читаемости кода и удобства добавления нового функционала в программу.

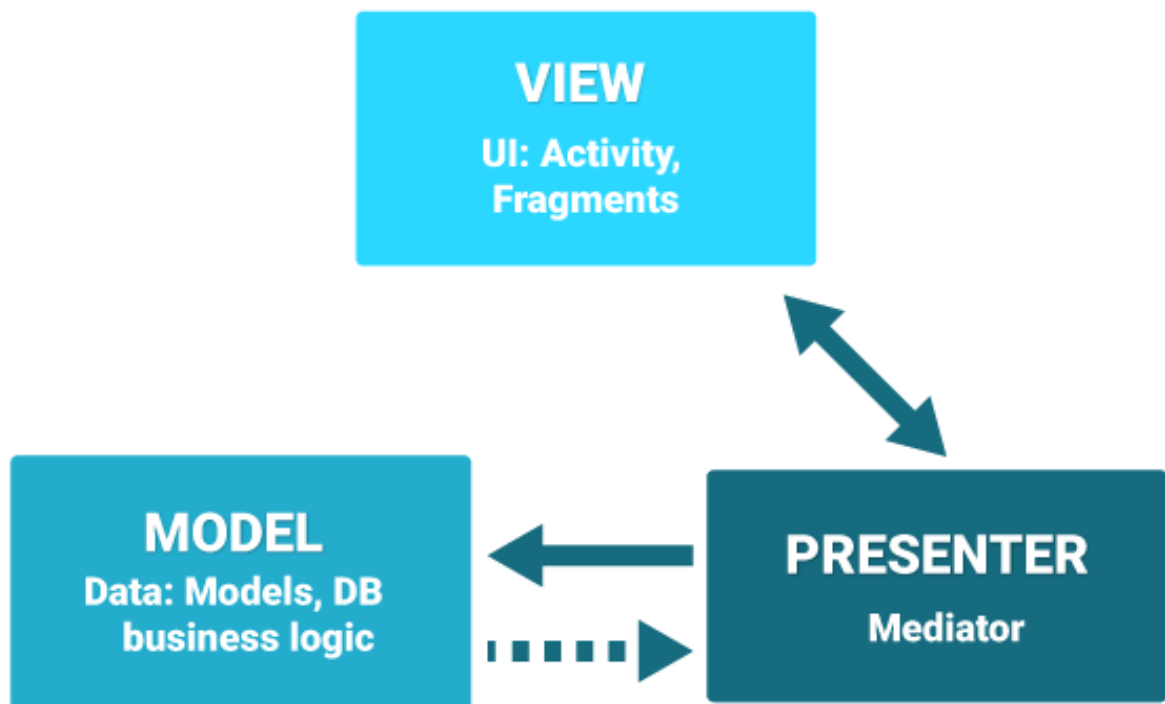


Рисунок 9 – Взаимодействие компонентов приложения при использовании шаблона MVP

Как можно заметить на рисунке 9, классы в данном шаблоне разделяются на три основных типа [9]:

- view – прорисовка интерфейса и взаимодействие с пользователем, анимации, переходы между экранами приложения, обработка событий жизненного цикла отдельного экрана;
- presenter – форматирование данных, реакция на события, логика представление и управления View;
- model – работа с загрузкой и отправкой данных по API, с локальным хранилищем данных (SQLite, Shared Preferences), комбинирование запросов и обработка результатов для последующей передачи их в Presenter.

На основе данного шаблона была создана базовая диаграмма классов для разрабатываемого мобильного приложения, изображённая на рисунке 10.

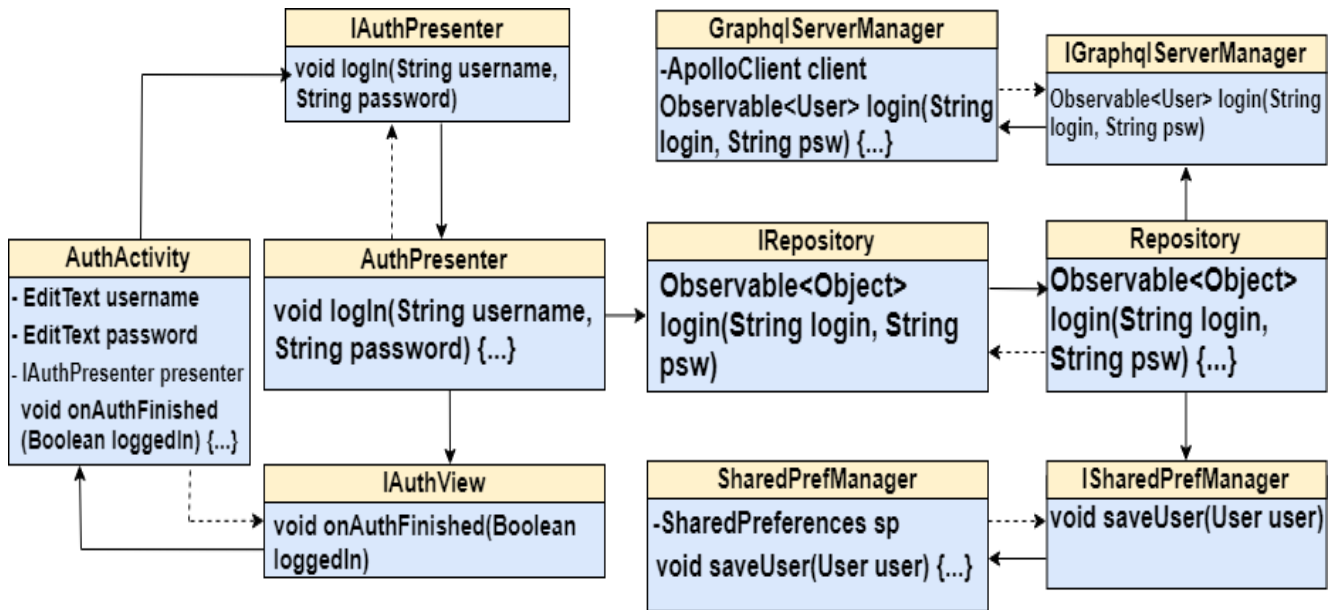


Рисунок 10 – Базовая диаграмма классов

Связь между элементами View и Presenter осуществляется по средствам вызова методов у интерфейсов – на рисунке 10 класс AuthActivity реализует интерфейс IAuthView, а AuthPresenter реализует интерфейс IAuthPresenter. Такая реализация позволяет проще тестировать отдельные компоненты, так как для тестирования View в данном случае нам не обязательно иметь полностью работающий Presenter – достаточно подставить «заглушку» в виде класса, реализующего интерфейс IAuthPresenter. В свою очередь View получает нужный экземпляр Presenter при помощи ранее описанного механизма внедрения зависимостей.

У каждого Presenter’а есть следующие поля:

- repository – экземпляр класса, реализующего интерфейс IRepository, который используется для различных операций с источниками данных;
- schedulerProvider – используется для определения потоков приложения, в которых будут выполняться запросы данных и в которые будут приходить события, являющиеся результатами выполнения этих запросов;
- compositeDisposable – используется для управления жизненным циклом подписок на различные события – например, если был запрошен

список товаров и пользователь покинул экран до того, как был получен ответ от сервера, presenter отписывается от получения результата данного запроса.

При вызове какого-либо из методов класса, реализующего интерфейс IRepository, происходит вызов одного из методов GraphQLServerManager или SharedPreferencesManager. Взаимодействие между данными классами также реализуется через их интерфейсы. В разработанной структуре приложения класс Repository является одним из ключевых – в нём происходит вся работа с различными источниками данных, а также с бизнес-логикой приложения.

Класс GraphQLServerManager является реализацией интерфейса GraphQLServerManager, он отвечает за отправку запросов на сервер и возвращает данные в виде нужной модели Java-объекта или коллекции таких объектов. Главным полем данного класса является ApolloClient – клиент для взаимодействия с сервером. Данный клиент конфигурируется при старте приложения, в нём устанавливается URL сервера, параметры кэширования, максимальное время выполнения запроса, политика повторных запросов при возникших ошибках и многое другое.

Класс SharedPreferencesManager отвечает за сохранение данных в постоянную память устройства и последующее считывание этих данных – например, сохранение данных пользователя после успешной регистрации или авторизации и считывание этих данных при следующих запусках приложения.

Структура проект в IDE Android Studio представлена на рисунке 11.

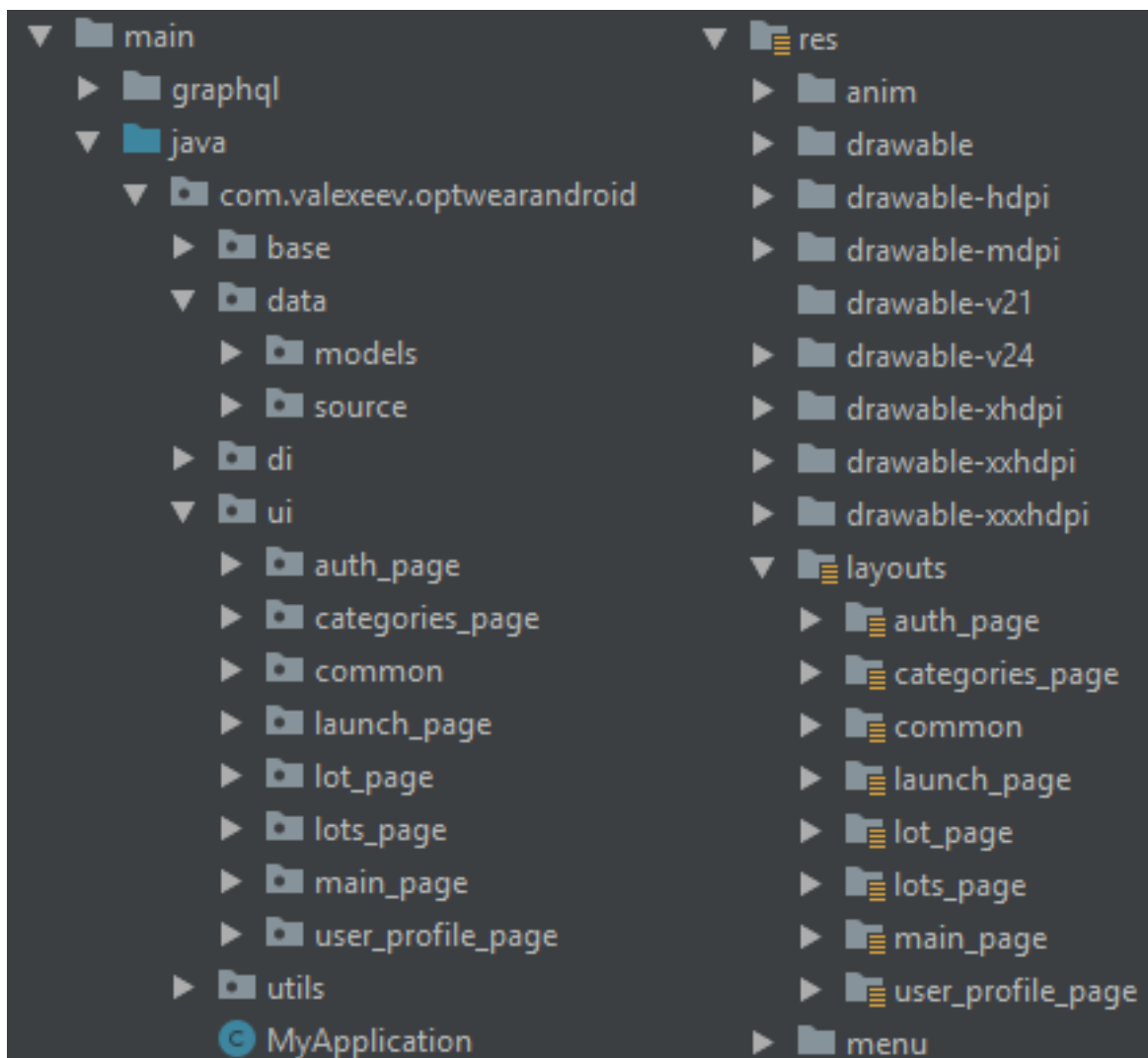


Рисунок 11 – Структура проекта в IDE Android Studio

2.2 Реализация графического интерфейса

Графический интерфейс пользователя в мобильных приложениях для ОС Android представляет собой иерархию объектов `android.view.View` и `android.view.ViewGroup` [10]. Каждый объект `ViewGroup` представляет контейнер, который содержит и упорядочивает дочерние объекты `View`. Простые объекты `View` представляют собой элементы управления и прочие виджеты, например кнопки, текстовые поля, индикаторы загрузки и т.д. Через эти элементы пользователь взаимодействует с программой. Файлы, определяющие графический интерфейс приложения, имеют расширение `.xml`, пример такого файла показан на рисунке 12, но также есть возможность

создавать элементы программно в коде Java. В данном проекте все интерфейсы описаны исключительно в XML файлах с целью лучшей читаемости кода разметки.

```
<?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="match_parent"
  android:orientation="vertical"
  android:layout_gravity="center"
  android:gravity="center">

  <ImageView
    android:id="@+id/logo_image_view"
    android:layout_width="match_parent"
    android:layout_height="140dp"
    android:src="@drawable/ic_optwear_logo"
    android:scaleType="fitCenter" />

  <include
    layout="@layout/loading_indicator" />

</LinearLayout>
```

Рисунок 12 – Пример XML файла интерфейса для загрузочного экрана

Также в разработке графического интерфейса активно применялись принципы Material Design – они позволяют совершенствовать визуальную составляющую приложения и служат для беспрепятственного взаимодействия различных функций на одной платформе. Поверхности и края элементов в данном направлении дизайна сочетают визуальные образы и сигналы, создавая подсказки, помогающие пользователю интуитивно ориентироваться в мобильном приложении. Кроме того, Material Design использует принципы полиграфического дизайна для эффектной фокусировки внимания на нужном элементе, упрощения навигации в интерфейсе, интуитивной передачи смысла его элементов.

Для Material Design характерны насыщенные, ровные цвета, резкие, очерченные края, крупная типографика и немалые отступы между элементами. Также данная концепция подразумевает активное использование анимации, что и было реализовано в данном проекте [11].

2.3 Примеры интерфейса приложения

На рисунке 11 изображен интерфейс главного экрана приложения, на котором расположены блоки со ссылками на социальные сети компании, блок баннеров и предложение войти в свой аккаунт или зарегистрироваться, которое отображает только для неавторизованных пользователей. В левом верхнем углу находится кнопка для открытия меню навигации. В правом верхнем углу экрана расположены кнопки для перехода к поиску и к корзине товаров пользователя.

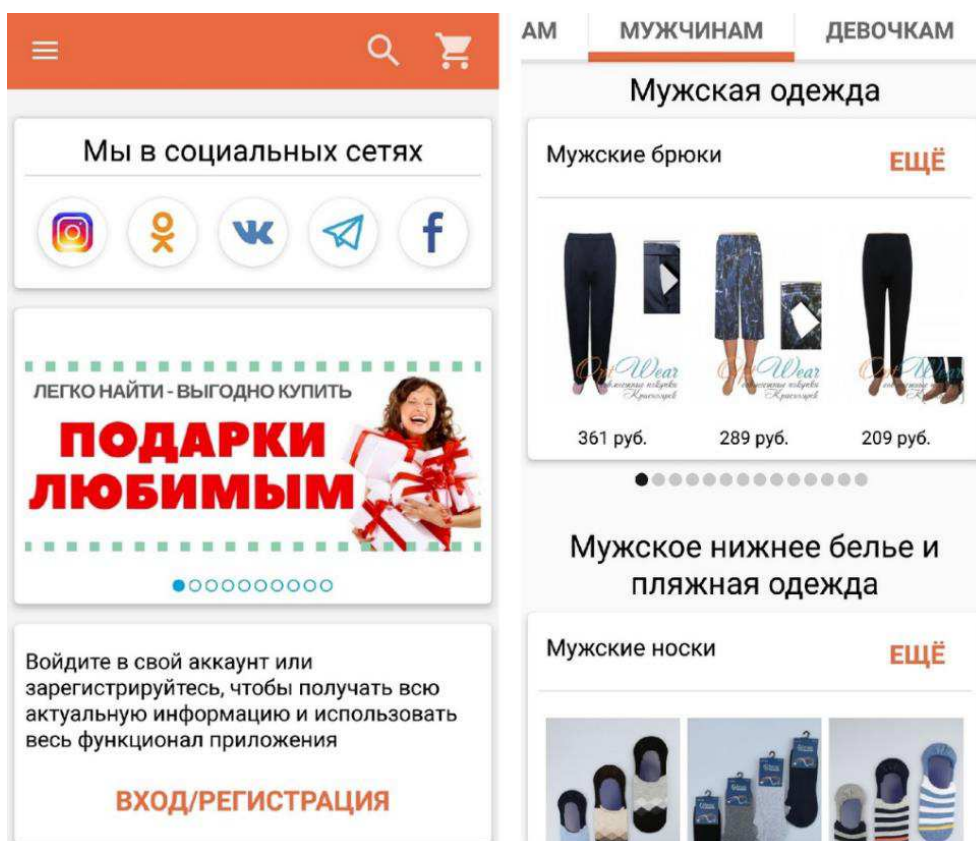


Рисунок 11 – Главный экран приложения

На рисунке 12 изображена страница со списком товаров. Для данных товаров можно менять вид их расположения на экране (крупные элементы с пролистыванием между всеми фотографиями товара или более компактное расположение элементов в виде сетки) и порядок их сортировки (сортировка по цене, названию или актуальности товара).

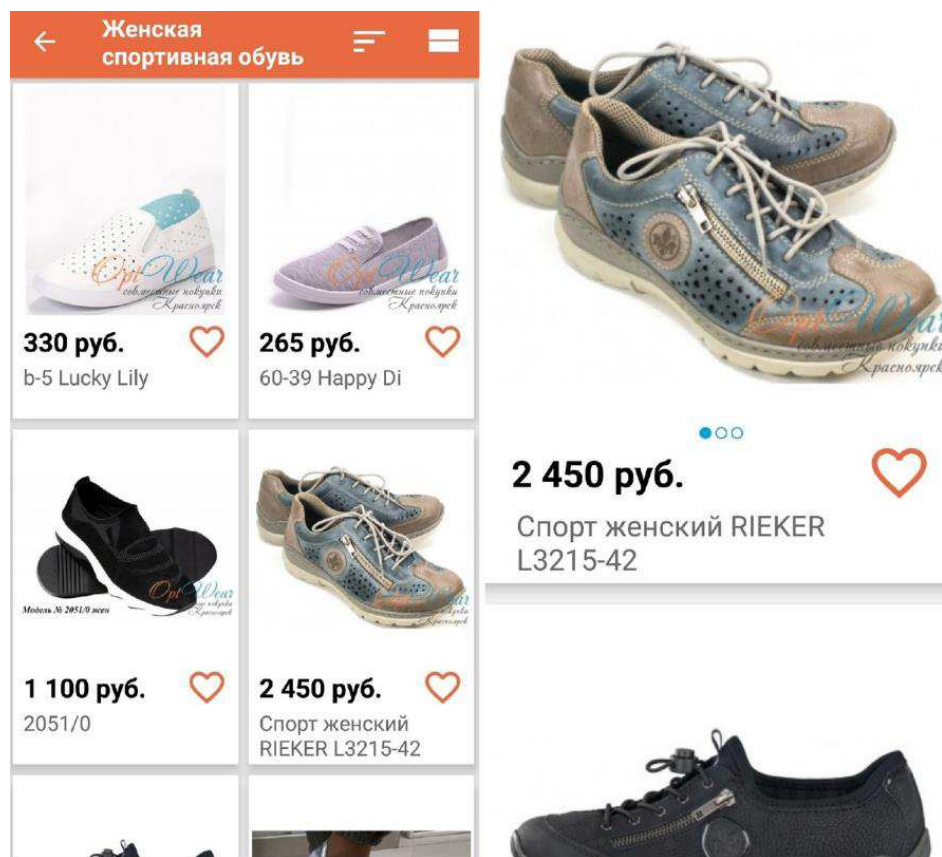


Рисунок 12 – Интерфейс страницы с товарами

На рисунке 13 изображён интерфейс страницы отдельно взятого товара, позволяющий просматривать все его фотографии, подробно ознакомиться с его характеристиками, выбрать нужный центр выдачи, свойства (размер, цвет и тд.) и количество приобретаемого товара. Также есть возможность поделиться ссылкой на данный товар через установленные на устройстве приложения, которые поддерживают данный функционал.

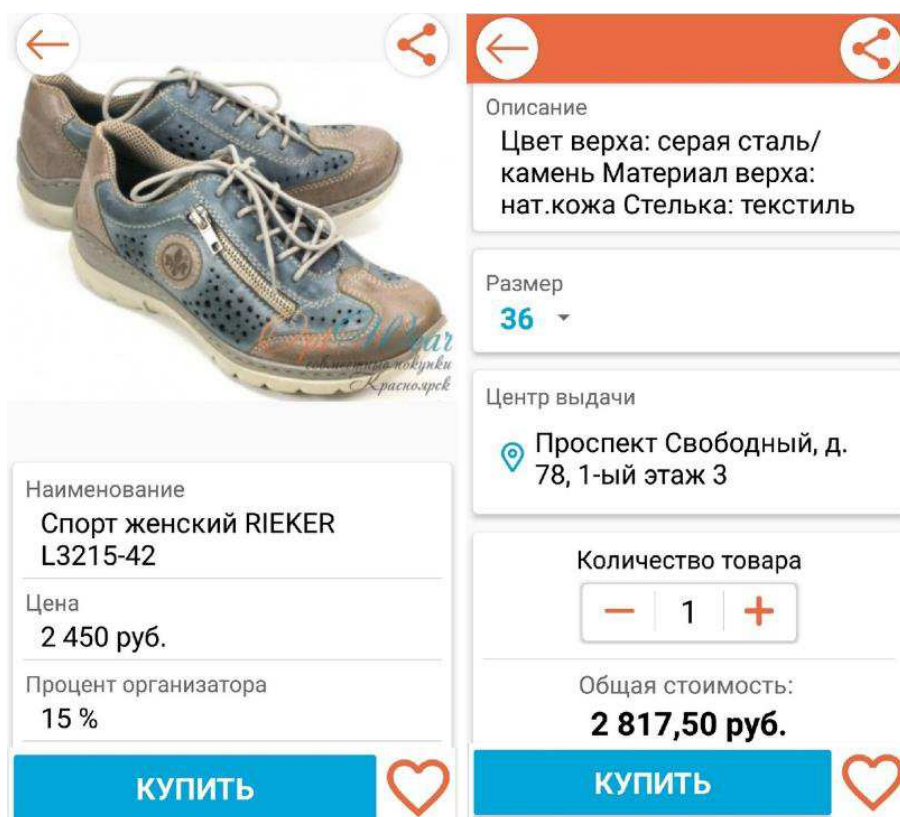


Рисунок 13 – Интерфейс страницы товара

2.4 Инструкция пользователя

При открытии мобильного приложения в первую очередь запускается начальный экран с логотипом «optwear.ru» и индикатором загрузки, после чего будет открыта главная страница. При нажатии на логотипе одной из социальных сетей в блоке «Мы в социальных сетях» будет открыта страница компании в соответствующем приложении. Если подходящего приложения нет на устройстве – ссылка откроется в окне браузера. При нажатии на один из баннеров произойдёт переход к соответствующей баннеру странице (категория товаров, конкретный товар, закупка и т.д.). Если пользователь ещё не авторизован – будет показан блок с предложением войти в существующий аккаунт или создать новый. При нажатии на кнопку «вход/регистрация» произойдёт открытие страницы авторизации. Ниже

расположены вкладки с различными категориями и показано по 3 товара из каждой подкатегории.

При переходе во вкладку «Мой профиль» откроется страница с информацией о текущем пользователе либо предложение об авторизации. На данной вкладке можно дополнить или изменить данные текущего профиля (пароль для входа в аккаунт, номер телефона, адрес электронной почты, центр выдачи по умолчанию) или выйти из него.

При переходе во вкладку «Товары» откроется страница с выбором категории. После выбора категории первого уровня будут отображены категории второго и третьего уровня. При выборе конечной категории открывается страница со списком товаров, соответствующих данной категории.

При переходе во вкладку «Сообщения» будет отображена страница с сообщениями для данного пользователя или предложение авторизации, после чего можно открывать сообщения выбирая их из списка.

При выборе вкладки «Отложено» откроется список избранных товаров, которые были помечены текущим пользователем, или предложение об авторизации. При нажатии на какой-либо элемент списка произойдёт переход на страницу конкретного товара.

При выборе вкладки «Оценить приложение» будет открыта страница приложения в Play Market, а при выборе пункта меню «Сообщить об ошибке» откроется диалоговое окно с полем для ввода текста и кнопкой для его отправки разработчикам.

При нажатии на иконку корзины произойдёт переход на страницу со списком товаров, находящихся в корзине пользователя, или предложение об авторизации. При нажатии на одном из элементов списка откроется страница с этим товаром.

При нажатии на иконку поиска откроется диалоговое окно для ввода поискового запроса и установки дополнительных параметров, после чего будет открыта страница с найденными товарами.

При переходе на страницу с конкретным товаром будет выведена вся информация по данной позиции – фотографии, название, цена, процент организатора, характеристики, выбранные параметры, центр выдачи и количество приобретаемого товара. При нажатии на кнопку «Купить» товар будет помещён в корзину текущего пользователя или будет выведено сообщение с предложением об авторизации.

2.5 Инструкция разработчика

Для компиляции исходного кода необходимо установить комплект разработчика приложений на языке Java (JDK – Java Development Kit). После этого необходимо установить IDE, способную открывать Android-проекты (Android Studio, Eclipse, NetBeans), а также Android SDK (Software Development Kit). Чаще всего Android SDK устанавливается автоматически при установке IDE. После выполнения всех вышеперечисленных действий можно открыть проект в IDE и запустить его кнопкой «запуск».

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был проведён анализ аналогов, спроектировано и реализовано мобильное приложение для сервиса совместных покупок «optwear.ru» для устройств под управлением операционной системы Android.

Был разработан графический интерфейс и структура приложения, реализован различный функционал, в том числе авторизация, навигация по категориям, сортировка и фильтрация товаров, поиск и многое другое.

В разработанном приложении присутствуют недостатки, исправить их можно благодаря будущей доработке, путем добавления следующих функций:

- Авторизации через социальные сети;
- Открытие страницы организатора закупки;
- Добавление блока «Похожие товары» на страницу выбранного товара.

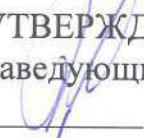
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 100sp – Сервис совместных покупок [Электронный ресурс]. – Режим доступа: <http://www.100sp.ru> (дата обращения: 02.05.2019).
2. 24ok – Сервис совместных покупок [Электронный ресурс]. – Режим доступа: <https://24-ok.ru> (дата обращения: 02.05.2019).
3. sp2all – Сервис совместных покупок [Электронный ресурс]. – Режим доступа: <https://sp2all.ru> (дата обращения: 02.05.2019).
4. optwear – Сервис совместных покупок [Электронный ресурс]. – Режим доступа: <https://optwear.ru> (дата обращения: 02.05.2019).
5. Эккель, Б. Философия Java, 4-е издание / Б. Эккель – изд. Питер, 2019. – 237 с.
6. Гриффитс, Д. Head First. Программирование для Android / Дэвид Гриффитс, Дон Гриффитс – изд. Питер, 2018. – 462 с.
7. GitHub – ReactiveX/RxJava [Электронный ресурс]. – Режим доступа: <https://github.com/ReactiveX/RxJava> (дата обращения: 02.05.2019).
8. GitHub – apollographql/apollo-android [Электронный ресурс]. – Режим доступа: <https://github.com/apollographql/apollo-android> (дата обращения: 02.05.2019).
9. Medium – MVP в Android приложении. [Электронный ресурс]. – Режим доступа: <https://medium.com/@skidanolegs/mvp-в-android-приложении-часть-1-9b08c46559fc> (дата обращения: 02.05.2019).
10. Android Developers Documentation – Обзор пользовательского интерфейса. [Электронный ресурс]. – Режим доступа: <https://developer.android.com/guide/topics/ui/overview.html?hl=ru> (дата обращения: 02.05.2019).
11. Material Design – Material Foundation [Электронный ресурс]. – Режим доступа: <https://material.io/design/foundation-overview/> (дата обращения: 02.05.2019).

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

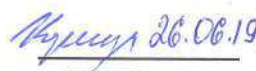


Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ВТ

О.В. Непомнящий
подпись инициалы, фамилия
« 25 » 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Интернет магазин. Мобильное приложение клиента.
тема

Руководитель	 <u>26.06.19.</u> подпись, дата	<u>доцент, канд. физ.-мат. наук</u> должность, ученая степень	<u>К. В. Коршун</u> инициалы, фамилия
Выпускник	 <u>26.06.19.</u> подпись, дата		<u>В. А. Алексеев</u> инициалы, фамилия
Нормоконтролер	 <u>27.06.19</u> подпись, дата	<u>доцент, канд. техн. наук</u> должность, ученая степень	<u>В. И. Иванов</u> инициалы, фамилия

Красноярск 2019