Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет им. Г.В. Плеханова»

**Московский приборостроительный техникум**

**Курсовой проект**

ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем»

МДК 01.01 «Разработка программных модулей»

Специальность 09.02.07 «Информационные системы и программирование»

Тема: «Разработка мобильного приложения по продаже товаров и услуг DP Stuff Provider»

**Пояснительная записка**

Листов: 29

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Шимбирёв

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.М. Суслин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

2021

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

Утверждаю

Заместитель директора по учебной работе

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Клопов

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Задание

на выполнение курсового проекта (курсовой работы)

Суслина Александра Михайловича

(фамилия, имя, отчество студента — полностью)

студенту группы П50-2-18 специальности 09.02.07 «Информационные системы и программирование» по МДК 01.01 «Разработка программных модулей»

1. Исходные данные к проекту (работе):
   1. Тема: «Разработка мобильного приложения подбора и информирования мероприятий в городе».
   2. Состав курсового проекта:
      1. Задание на выполнение курсового проекта
      2. Пояснительная записка
      3. Программный продукт (Инсталляционный пакет) на электронном носителе
      4. Программный продукт (Исходный проект) на электронном носителе
      5. Презентация на электронном носителе
2. Содержание задания по проекту (работе) ⁠— перечень вопросов, подлежащих разработке

|  | Разрабатываемый вопрос | Объем от всего задания, % | Срок выполнения |
| --- | --- | --- | --- |
| А | Описательная часть проекта (введение, общее описание и т. д.) | 5% |  |
| 1. | Введение | 3 | 29.01.21 |
| 2. | Цель разработки | 1 | 29.01.21 |
| 3. | Средства разработки | 1 | 29.01.21 |
| Б | Анализ задачи и её постановка | 10% |  |
| 1. | Определение требований к программе | 2 | 05.02.21 |
| 2. | Спецификация программы (описание задачи, описание входных и выходных данных, метод) | 3 | 05.02.21 |
| 3. | Тесты, контроль целостности данных | 5 | 05.02.21 |
| В | Проектирование и реализация | 50% |  |
| 1. | Схемы проекта (диаграмма классов, функциональная схема, структурная схема, схема пользовательского интерфейса) | 15 | 12.03.21 |
| 2. | Реализация в инструментальной среде | 35 | 12.03.21 |
| Г | Технологическая часть проекта | 10% |  |
| 1. | Инструментальные средства разработки | 3 | 23.04.21 |
| 2. | Отладка программа | 2 | 23.04.21 |
| 3. | Защитное программирование | 3 | 23.04.21 |
| 4. | Характеристика программы | 2 | 23.04.21 |
| Д | Программная документация | 20% |  |
| 1. | Приложение А. Эскизный проект | 5 | 30.04.21 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Разрабатываемый вопрос | Объем от всего задания, % | Срок выполнения |
| 2. | Приложение Б. Скрипт базы данных | 5 | 07.05.21 |
| 3. | Приложение В. Текст программы | 5 | 14.05.21 |
| 4. | Приложение Г. Руководство пользователя | 5 | 21.05.21 |
| Е | Экспериментальная часть проекта | 5% |  |
| 1. | Электронный носитель: исходный проект, эксплуатационный пакет, презентация, документация. | 5 | 28.05.21 |

Руководитель курсового проекта (работы) Шимбирёв Андрей Андреевич, преподаватель

«12» января 2021 года \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Шимбирёв /

Дата выдачи курсового задания «12» января 2021 года

Срок сдачи законченного проекта (работы) «21» июня 2021 года

Задание принял к исполнению

«12» января 2021 года \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.М. Суслин/

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc73180509)

[1. ОБЩАЯ ЧАСТЬ 7](#_Toc73180510)

[1.1. Цель разработки 7](#_Toc73180511)

[1.2. Средства разработки 7](#_Toc73180512)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 9](#_Toc73180513)

[2.1. Постановка задачи. 9](#_Toc73180514)

[2.1.1. Входные и выходные данные. 9](#_Toc73180515)

[2.1.2. Подробные требования к проекту. 9](#_Toc73180516)

[2.2. Внешняя спецификация. 11](#_Toc73180517)

[2.2.1. Описание задачи. 11](#_Toc73180518)

[2.2.2. Входные и выходные данные. 11](#_Toc73180519)

[2.2.3. Методы. 12](#_Toc73180520)

[2.2.4. Тесты. 12](#_Toc73180521)

[2.2.5. Контроль целостности данных. 12](#_Toc73180522)

[2.3. Проектирование 14](#_Toc73180523)

[2.3.1. Схема архитектуры программы 14](#_Toc73180524)

[2.3.2. Структурная схема программы 14](#_Toc73180525)

[2.3.3. Функциональная схема 14](#_Toc73180526)

[2.3.4. Диаграмма классов 15](#_Toc73180527)

[2.3.5. Модель базы данных 16](#_Toc73180528)

[2.4. Результаты работы программы 17](#_Toc73180529)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 26](#_Toc73180530)

[3.1. Инструментальные средства разработки 26](#_Toc73180531)

[3.2. Отладка программы 26](#_Toc73180532)

[3.3. Защитное программирование 27](#_Toc73180533)

[3.4. Характеристики программы 27](#_Toc73180534)

[ЗАКЛЮЧЕНИЕ 28](#_Toc73180535)

[СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ 29](#_Toc73180536)

ПРИЛОЖЕНИЕ А Эскизный проект.

ПРИЛОЖЕНИЕ Б Скрипт базы данных.

ПРИЛОЖЕНИЕ В Текст программы.

ПРИЛОЖЕНИЕ Г Руководство Пользователя.

# ВВЕДЕНИЕ

Мобильное приложение по продаже товаров и услуг DP Stuff Provider предназначено для упрощения поиска нужного пользователю товара или услуги и оформления заказа, также благодаря мобильному приложению будет ускорен процесс оформление и доставки заказа до клиента.

Актуальностью разработки мобильного приложения по продаже товаров и услуг является постоянно растущий спрос на качественные товары и услуги, на скорость доставки товаров до рук клиента, а также на удобство оформления заказа, не выходя из дома.

Практичностью написания мобильного приложения по продаже товаров и услуг обуславливается актуальностью задачи в написании программ, связанных с облачными хранилищами, процессами получения и работы с данными из них, так и развитии в программировании на мобильных устройствах, которое будет востребовано и актуально в любое время.

Для написания курсовой работы и достижения выше обозначенной цели были изучены и использованы материалы следующих тематик:

* Создание Api на платформе ASP.NET для взаимодействия с базой данных на сервере;
* Создание и администрирование базы данных, с помощью СУБД MS SQL Server;
* Размещение базы данных и веб-приложения на облачном сервисе Azure;
* Разработка мобильного приложения c помощью IDE Android Studio на языке Kotlin;
* Использование библиотеки GoogleMaterials;
* Обращение к Api с помощью библиотеки Retrofit 2 для android.

1. ОБЩАЯ ЧАСТЬ
   1. Цель разработки

Автоматизировать бизнес-процесс интернет-магазина по продаже товаров и услуг, а также ускорить процесс доставки заказов до клиентов и обеспечить их удобным интерфейсом для поиска и заказа нужных товаров на мобильной платформе Android.

* 1. Средства разработки

В качестве инструментальных средств разработки использовалось ПО, предоставленное в таблице 1.

Таблица 1 - Программные средства

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средства | Название средства | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Система управления базами данных | MS SQL Server 2017 | Создание и администрирование базы данных |
| 2 | Текстовый редактор | Microsoft Word 2016 | Разработка документации, формирование отчетных документов по шаблонам |
| 3 | Среда разработки мобильного приложения | Android Studio 4.1.3 | Разработка мобильного приложения |
| 4 | Среда разработки Api | Visual Studio 2019 | Разработка и управление Api |

В качестве вычислительной техники использовался стационарный компьютер. Его характеристики представлены в таблице 2.

Таблица 2 - Технические средства разработки

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Центральный процессор: | Intel Core i5-3450 |
| 2 | Количество ядер | 4 |
| 3 | Видеоадаптер: | GeForce GTX 660 (2 ГБ) |
| 4 | Системная память | 8 ГБ (DDR3) |
| 5 | Твердотельный накопитель: | 476 ГБ SSD |
| 6 | Разрешение экрана | 1920х1080 |
| 7 | Операционная система | Windows 10 Pro |

В качестве устройства, на котором тестировалось мобильное приложение, использовался смартфон Vivo, его характеристики представлены в таблице 3.

Таблица 3 - Технические средства разработки

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Центральный процессор: | MTK6762R |
| 2 | Количество ядер | 8 |
| 3 | Видеоадаптер: | PowerVR GE8320 (650 МГц) |
| 4 | Оперативная память | 4 ГБ (DDR3) |
| 5 | Операционная система | Android 8.1.0 |

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Постановка задачи.

Необходимо разработать мобильное приложение на платформе Android.

Приложение должно предоставлять пользователю возможность поиска нужного товара (по наименованию или категории товара), возможность добавления нужных товаров в корзину (где можно будет менять количество заказываемого товара), возможность авторизации и регистрации, возможность оформление заказа (с указанием адреса, даты, времени и комментарием к заказу), а также возможность просмотра личной информации и отслеживание статуса заказа в личном кабинете. Для взаимодействия мобильного приложения с базой данных, необходимо разработать Api, в качестве ответа Api должна отправлять json строку.

* + 1. Входные и выходные данные.

Входными данными программы являются логин и пароль пользователя, адрес доставки товаров и наименование товара для поиска. Выходными данными являются получаемые данные с API.

* + 1. Подробные требования к проекту.
* Сервер базы данных и Api должны располагаться на облачном сервисе Azure;
* Обращение к данным из облачного хранилища должны осуществляться через программный интерфейс (Api);
* Все пароли должны хешироваться алгоритмом SHA-256 и храниться в БД исключительно как хеш-строка;
* Залогиненный пользователь, также, как и корзина, должны сохраняться между сессиями работы приложения;
* Необходимо разработать меню навигации с 4-мя пунктами:
  + Главная – здесь будет отображаться список самых популярных (часто заказываемых) товаров;
  + Каталог – здесь будет отображаться список с категориями, по которому можно пройтись, соблюдая всю иерархию категорий, а также поле поиска, воспользовавшись которым можно будет найти товар по наименованию. Поиск по наименованию необходимо разработать так, чтобы при поиске возвращался список товаров, включающих в своё наименование искомую строку. Поиск по категории необходимо разработать так, чтобы при поиске возвращался список товаров, входящих в искомую категорию или если искомая категория является родительской;
  + Корзина – здесь будут отображаться товары, которые перед этим в неё добавили. Здесь также можно менять количество заказываемого товара, а также перейти к оформлению заказа, где клиенту нужно указать адрес, дату, удобное время для принятия доставки и комментарий, после чего заказ занесётся в БД, и пользователь сможет посмотреть статус и код заказа в своём профиле;
  + Профиль – здесь будет отображаться личная информация клиента (если клиент авторизован), в случае если клиент не авторизован ему нужно предоставить возможность авторизации или регистрации. Также здесь должна быть возможность просмотра списка оформленных заказов, где можно будет посмотреть общую информацию по заказу и код для получения заказа;
* Все пользовательские формы должны проверяться на корректность, и следует показывать всплывающие подсказки, в случаях некорректного ввода;
* В случае ошибок (не отвечает сервер) – пользователь должен получить соответствующее предупреждение.
  1. Внешняя спецификация.
     1. Описание задачи.

При запуске программы пользователя встречает главное окно со списком самых популярных заказов, где пользователь может добавить нужные товары в корзину, или воспользоваться меню и перейти в нужный ему пункт. После того как пользователь будет авторизирован, он может оформить новый заказ, после того как наберёт корзину с нужными ему товарами.

* + 1. Входные и выходные данные.

Таблица 3 - входные данные

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Поле | Тип данных | Ограничение | Формат ввода | Описание |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | Почта | Строка | Не меньше 6 символов. Строка должна заканчиваться @mail.ru и ей подобными. Не должна быть пустой | Поле для ввода текста | Пользователь вводит данные своей почты |
| 2 | Пароль | Строка | Должен содержать буквы, и цифры. Не должен быть пустым | Поле для ввода текста | Пользователь вводит свой пароль |

Таблица 4 - выходные данные

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Поле | Тип данных | Ограничение | Формат вывода | Описание |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | Наименование товара | Строка | Не должно быть пустым | Поле для вывода текста | Пользователь виде данные о наименование почты |
| 2 | Описание товара | Строка | Не должно быть пустым. Должно сдержать информацию о конкретном месте | Поле для вывода текста | Пользователь видит данные о местоположение места |
| 3 | Адрес главного изображения товара | Строка | Не должно быть пустым | Поле для вывода текста | Пользователь видит описание местоположения |

* + 1. Методы.

Для разработки использовалась методология ООП.

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

[[https://ru.wikipedia.org/wiki/Объектно-ориентированное\_программирование](https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование%23:~:text=Объе́ктно-ориенти́рованное%20программи́рование%20(ооп)%20—%20методология,которых%20является%20экземпляром%20определенного%20класса)]

* + 1. Тесты.

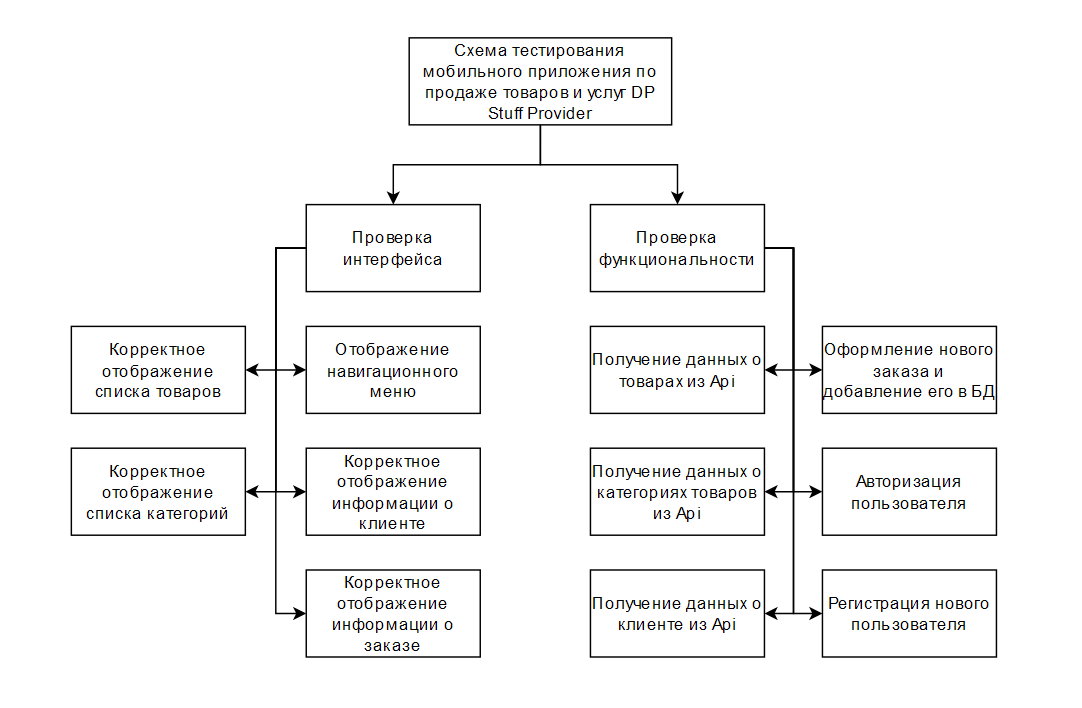


Рисунок 1 – Тесты

* + 1. Контроль целостности данных.

База данных приведена к третьей нормальной форме.

Таблица 5 – контроль целостности данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ситуация | Аномалия | Реакция программы | Примечание |
|  | 1 | 2 | 3 | 4 |
| 1 | Пользователь нажимает на кнопку для регистрации | Переход не осуществляется | Открывается окно регистрацией |  |
| 2 | Пользователь нажимает на кнопку «Добавить в корзину» | Товар не добавился в корзину | Товар добавился в корзину |  |
| 3 | Нажатие на кнопку «Оформить» | Заказ не оформился и не добавился в БД | Вывод всплывающего сообщения об успешном оформлении заказа |  |

* 1. Проектирование
     1. Схема архитектуры программы

Схема архитектуры программы: локальная

* + 1. Структурная схема программы

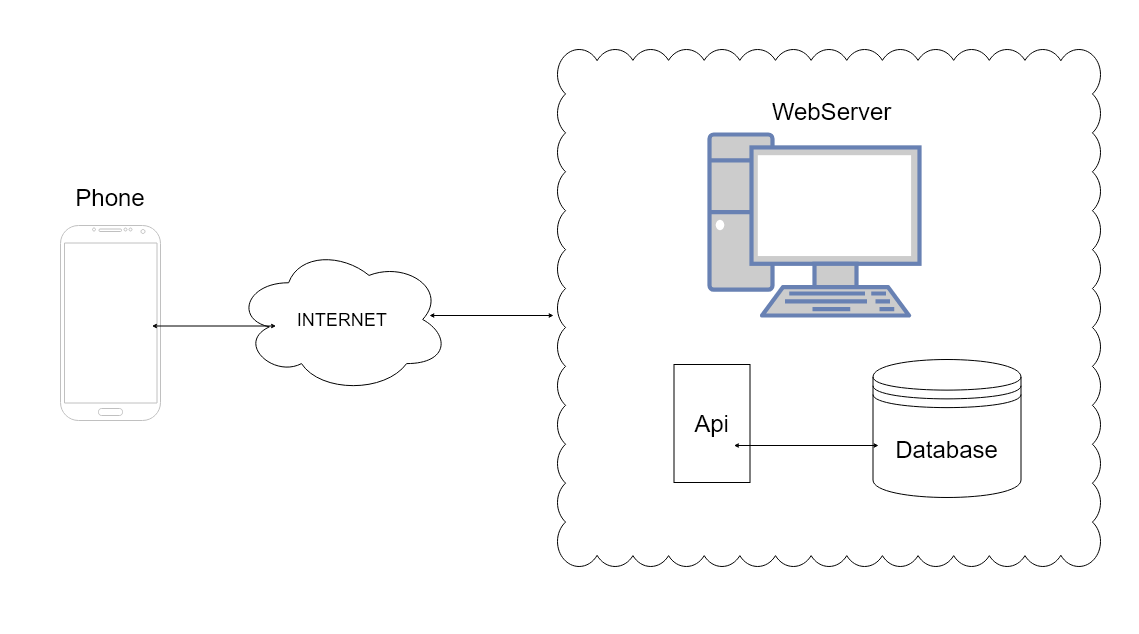


Рисунок 2 - Структурная схема

* + 1. Функциональная схема

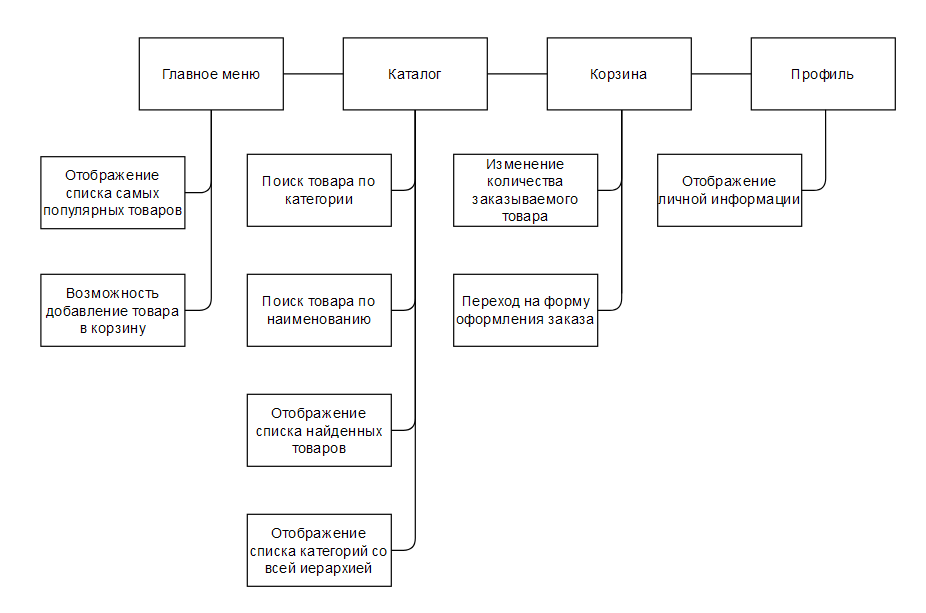


Рисунок 3 - Функциональная схема

* + 1. Диаграмма классов

Таблица 6 - Диаграмма классов

| № | Название | Назначение |
| --- | --- | --- |
| 1 | CartAdapter.kt | Адаптер для заполнения корзины |
| 2 | CategoryAdapter.kt | Адаптер для заполнения категорий каталога |
| 3 | OrderAdapter.kt | Адаптер для заполнения заказов клиента |
| 4 | ProductAdapter.kt | Адаптер для заполнения информации о товарах |
| 5 | AccountFragment.kt | Фрагмент аккаунта залогиненного пользователя |
| 6 | AccountLoginFragment.kt | Фрагмент формы авторизации пользователя |
| 7 | AccountNotLoginFragment.kt | Фрагмент не залогиненного пользователя |
| 8 | CartEmptyFragment.kt | Фрагмент пустой корзины |
| 9 | CartFragment.kt | Фрагмент корзины, корзина заполняется по списку товаров cartList из MainActivity |
| 10 | CatalogFragment.kt | Фрагмент каталога с поиском товаров по наименованию и категориям |
| 11 | CheckoutFragment.kt | Фрагмент оформления нового заказа |
| 12 | ErrorFragment.kt | Фрагмент заглушки на случай ошибок (не отвечает api) |
| 13 | HomeFragment.kt | Фрагмент домашней страницы со списком популярных товаров |
| 14 | OrdersFragment.kt | Фрагмент со списком заказов пользователя |
| 15 | ProductsFragment.kt | Фрагмент со списков товаров (по наименованию или категории) |
| 16 | RegistrationFragment.kt | Фрагмент регистрации нового пользователя |
| 17 | ClientApiService.kt | Класс, реализующий все методы Api-интерфейсов |
| 18 | MainActivity.kt | Основное активити, включающая в себя меню навигации и все фрагменты |

* + 1. Модель базы данных

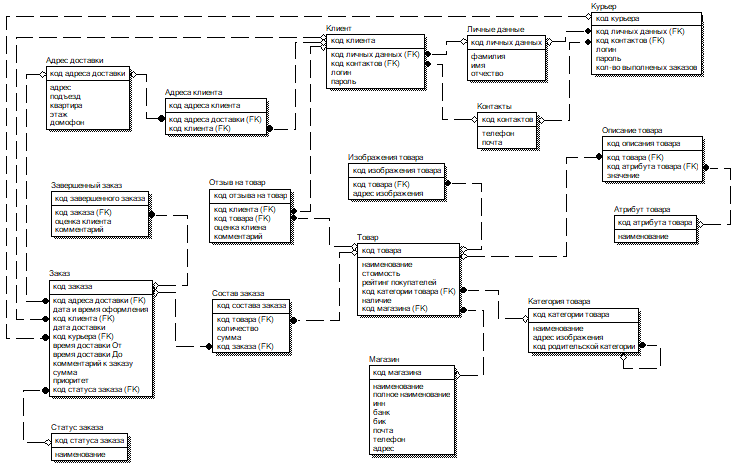


Рисунок 4: Даталогическая модель базы данных

* 1. Результаты работы программы

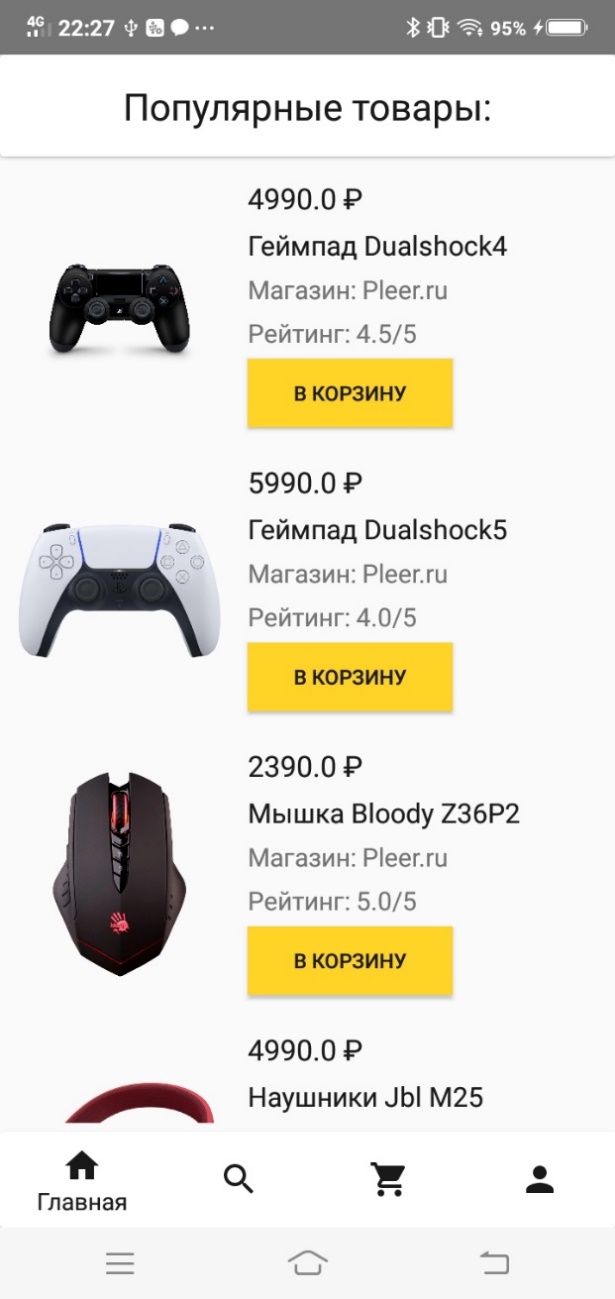


Рисунок 5 – Главная

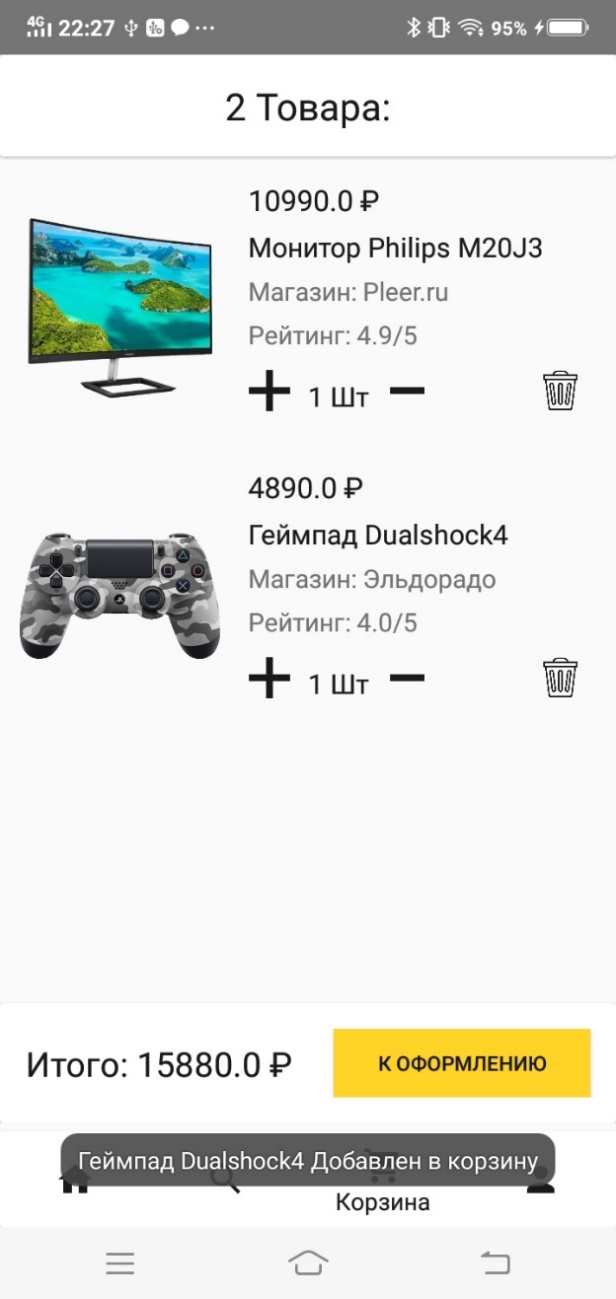


Рисунок 6 – Корзина

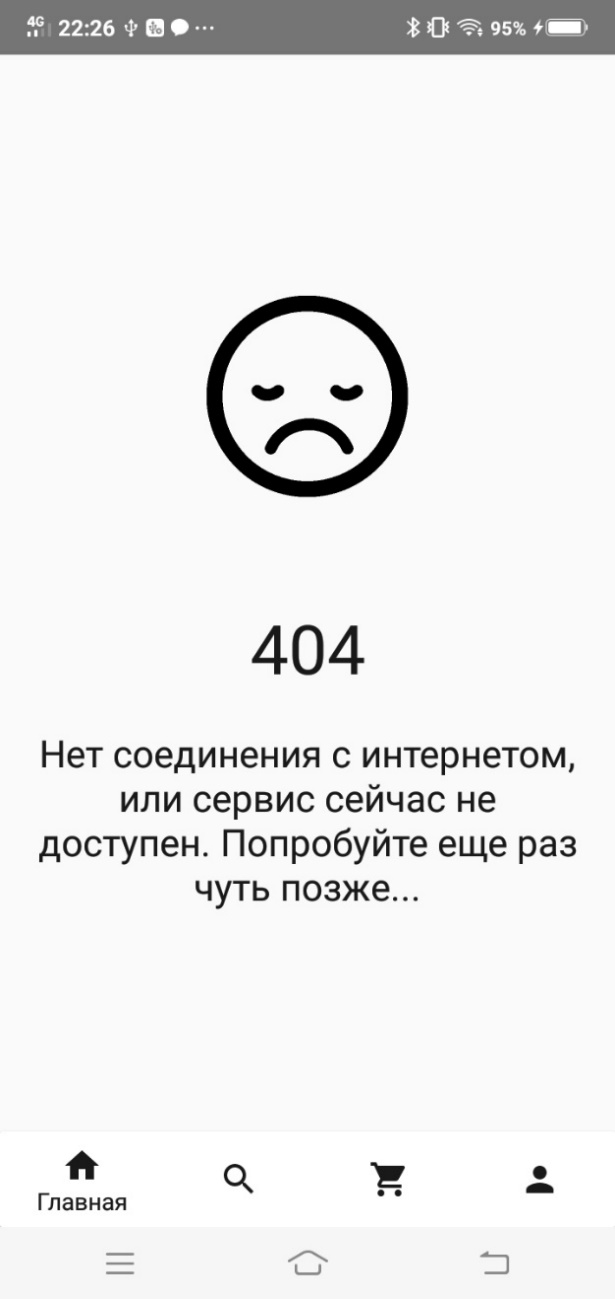


Рисунок 7 – Заглушка на случай ошибок

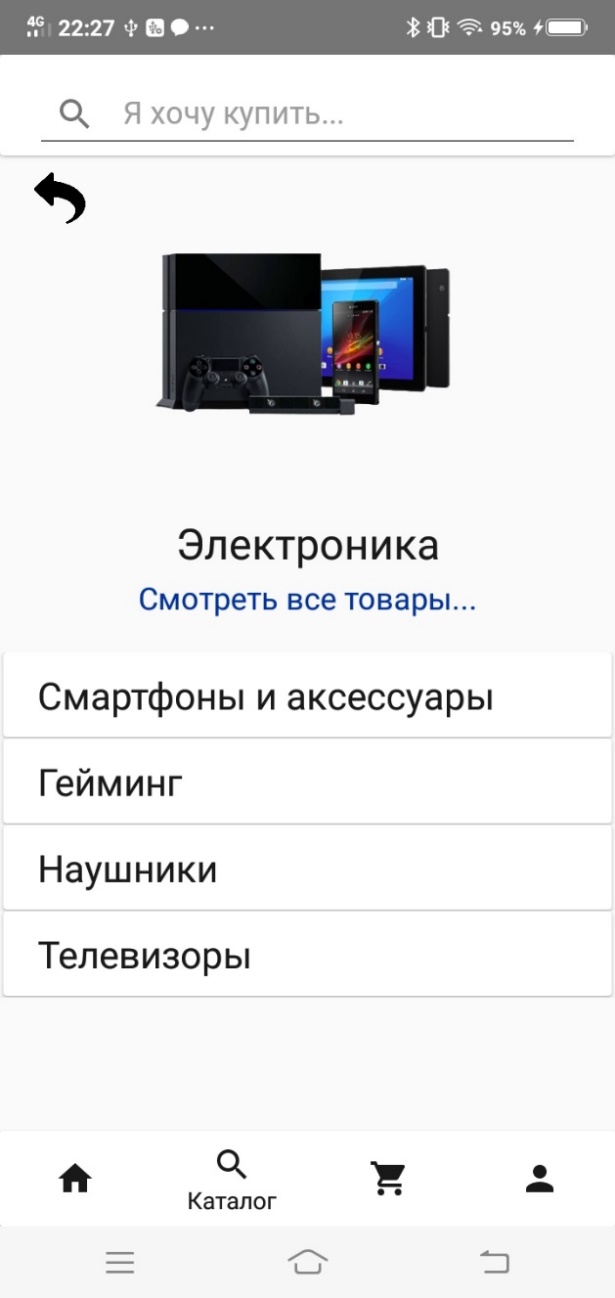


Рисунок 8 - Каталог

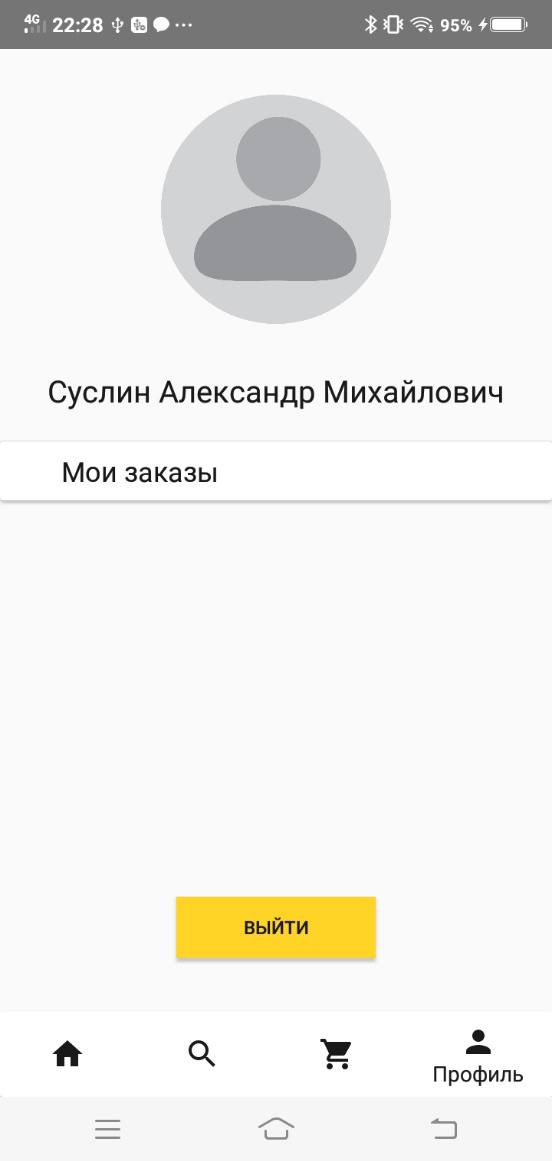


Рисунок 9 – Профиль пользователя

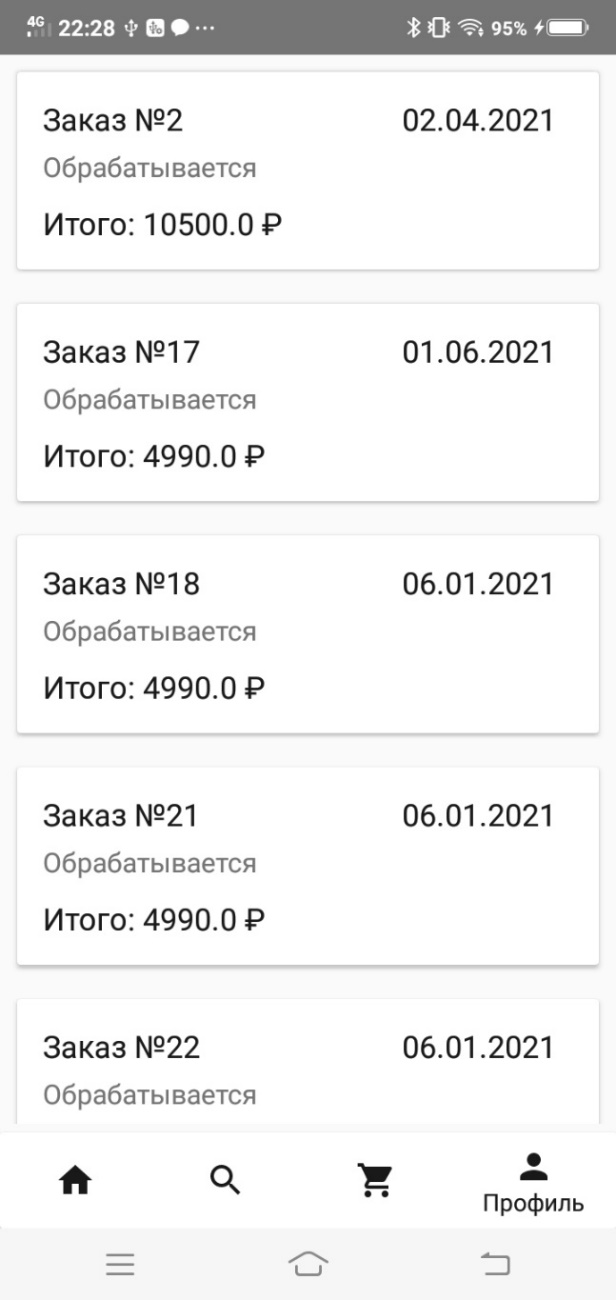


Рисунок 10 – Список заказов пользователя

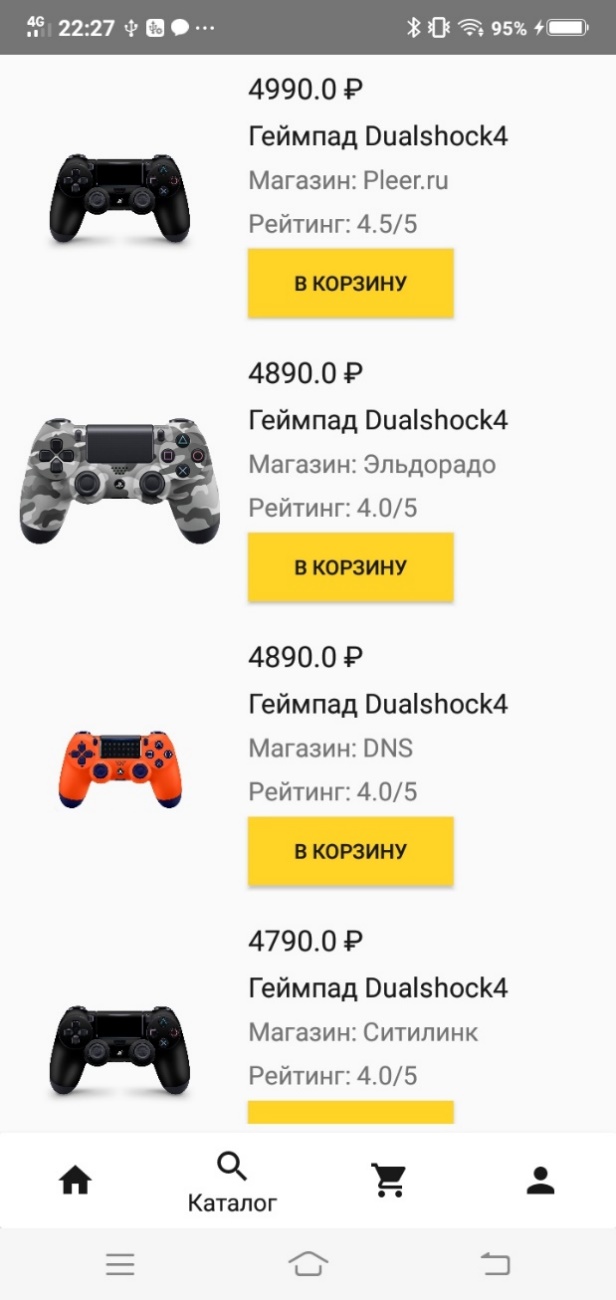


Рисунок 11 – Окно со списком товаров

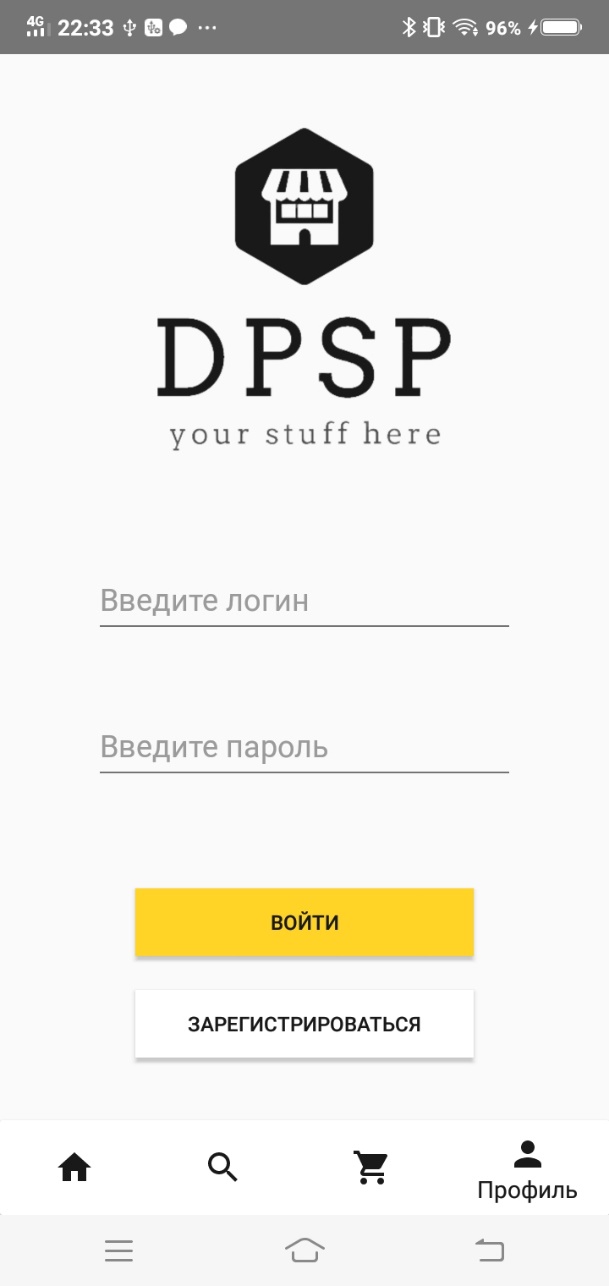


Рисунок 12 - Авторизация

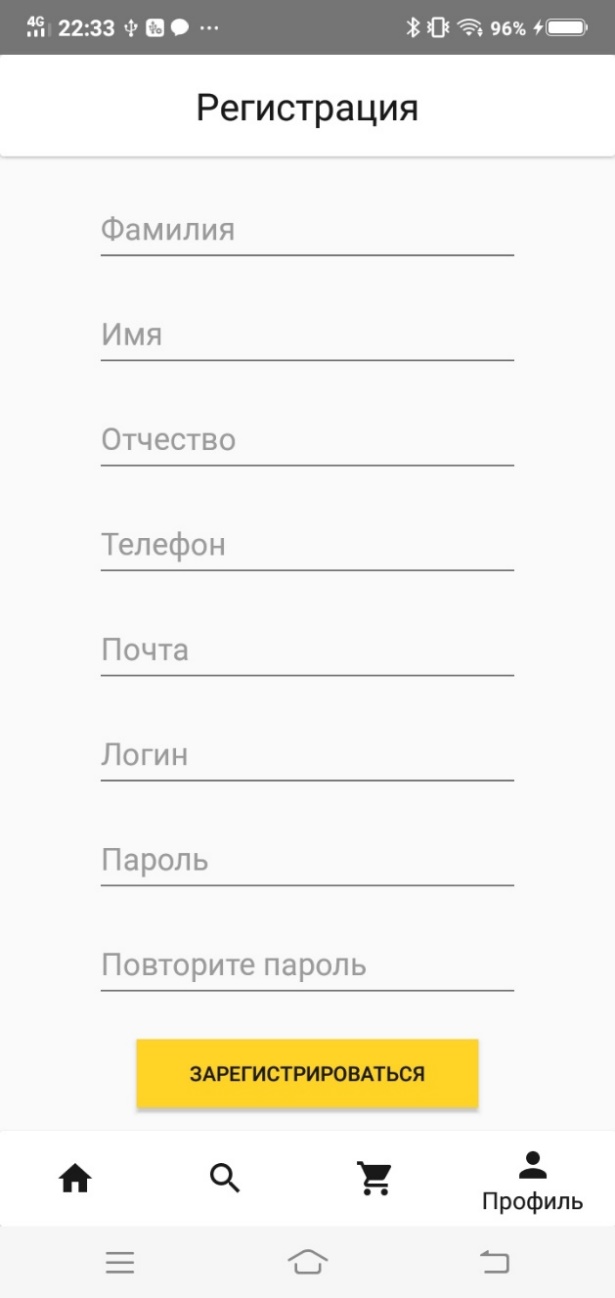


Рисунок 13 - Регистрация

1. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ
   1. Инструментальные средства разработки

В процессе разработке было решено использовать среду Android Studio, по причине удобства в создании мобильных приложений. Для разработки мобильного приложения был выбран язык Kotlin так, как он удобен и легок в освоении.

Плюсы:

* Универсальность
* Instant Run. Это функция, которой на протяжении всего времени развития Android Studio было уделено довольно много внимания, благодаря чему к выходу версии 3.0 она уже работала в полноценном режиме. Instant Run включена для того, чтобы разработчик приложений для Android после изменения кода мог сразу оценить, как это изменение повлияет на результат — и без дополнительных временных затрат на перекомпиляцию.
* Процесс разработки, который подстраивается под разработчика.
* редактор кода, с которым удобно работать.
* позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана.
* тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе.

Минусы:

* скупые возможности персонализации проявляются в редакторе кода и общих настройках. Для написания программы, которая будет формировать API запросы и связывать конечного пользователя с базой данных была выбрана среда разработки VisualStudio, язык C#, платформа разработки ASP.NET.
  1. Отладка программы

В процессе отладки была обнаружена ошибка, из-за которой время на путь не отображалось, это было связано с тем, что заполнение полей было быстрее, чем получение данных с API.

* 1. Защитное программирование

Решением проблемы чтения данных с API было использование асинхронного программирования (Coroutine)

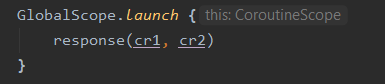


Рисунок 14 - решение проблемы

* 1. Характеристики программы

Таблица 7 - Характеристики программы

| № | Название | Кол-во строк | Размер в КБ |
| --- | --- | --- | --- |
| 1 | CartAdapter.kt | 156 | 6.61 |
| 2 | CategoryAdapter.kt | 131 | 6.02 |
| 3 | OrderAdapter.kt | 49 | 2.18 |
| 4 | ProductAdapter.kt | 101 | 4.36 |
| 5 | AccountFragment.kt | 44 | 1.51 |
| 6 | AccountLoginFragment.kt | 53 | 2.08 |
| 7 | AccountNotLoginFragment.kt | 32 | 0.953 |
| 8 | CartEmptyFragment.kt | 34 | 0.993 |
| 9 | CartFragment.kt | 56 | 2.09 |
| 10 | CatalogFragment.kt | 75 | 2.65 |
| 11 | CheckoutFragment.kt | 118 | 4.86 |
| 12 | ErrorFragment.kt | 27 | 0.786 |
| 13 | HomeFragment.kt | 56 | 1.91 |
| 14 | OrdersFragment.kt | 44 | 1.7 |
| 15 | ProductsFragment.kt | 74 | 2.7 |
| 16 | RegistrationFragment.kt | 87 | 3.43 |
| 17 | ClientApiService.kt | 280 | 10.7 |
| 18 | MainActivity.kt | 83 | 3.04 |

# ЗАКЛЮЧЕНИЕ

Исходя из приведенных исследований, можно сделать вывод, что в итоге разработки мобильного приложения по продаже товаров и услуг, проблематика системы, в создании базы данных, была упрощена, что позволило сделать ее адаптивной. Конечный программный, продукт является реализуемым и выполняет свои запланированные функции, как приложение по продаже товаров и услуг.

На освоении новой полученной информации по разрабатываемой теме, был получен новый уровень знания в программировании и решении нестандартных проблем, которые требовали гибкого подхода. Тема актуальности в разработке мобильных приложений работы с облачными хранилищами подтвердила себя и закрепилась как основная для изучения в моем арсенале. Материалы, предназначенные для использования в написании мобильного приложения, веб приложения и облачного хранилища были изучены и раскрыты по мере их необходимости.

Основываясь на выше приведенных фактах о тематики разработки приложения и изучении проблемной области, можно сделать вывод, что полученные практический, теоретический опыт и знания, внесут свой вклад в создании будущих проектов.

# СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. Head First. Программирование для Android. Гриффитс Д. Питер, 2016

2. Android NDK. Руководство для начинающих. Сильвен Ретабоуил. ДМК Пресс, 2016

3. Mastering Android NDK. Sergey Kosarevsky, Viktor Latypov. Packt Publishing Ltd., 2015

4. Android. Программирование для профессионалов. Билл Филлипс, К. Стюарт, Кристин Марсикано. Питер, 2017

5. Android. Технологии асинхронной обработки данных. Андерс Ёранссон. ДМК Пресс, 2015

6. Asynchronous Android Programming. Packt Publishing Ltd., 2016

7. Android Concurrency. Addison-Wesley Professional, 2016

8. Android Security Internals: An In-Depth Guide to Android's Security Architecture. Nikolay Elenkov. No Starch Press, 2014

9. Android Security Cookbook. Keith Makan, Packt Publishing Ltd., 2013

10. Android Hacker's Handbook. Joshua J. Drake, Zach Lanier, Collin Mulliner, Pau Oliva Fora, Stephen A. Ridley, Georg Wicherski. Wiley, 2014

11. Bulletproof Android: Practical Advice for Building Secure Apps (Developer's Library). Godfrey Nolan. Addison-Wesley Professional, 2014

12. Android User Interface Design: Implementing Material Design for Developers (2nd Edition) (Usability). Ian G. Clifton. Addison-Wesley Professional, 2015

13. Android Design Patterns and Best Practice (1st Edition). Kyle Mew. Packt Publishing Ltd., 2016

14. Embedded Programming with Android: Bringing Up an Android System from Scratch (Android Deep Dive, 1st Edition). Addison-Wesley Professional, 2015

15. Android Application Testing Guide Diego Torres Milano, Packt Publishing Ltd., 2011

16. Inside the Android OS: Building, Customizing, Managing and Operating Android System Services (Android Deep Dive, 1st Edition)*.* G. Blake Meike, Addison-Wesley Professional, 2018

17. Reactive Programming with RxJava. Creating Asynchronous, Event-Based Applications*.* Ben Christensen, Tomasz Nurkiewicz. -O'Reilly Media, 2016

**18. Jonathon Manning, Paris Buttfield-Addison Mobile Game Development with Unity: Build Once, Deploy Anywhere, 2014**

**19. Mark L. Murphy Busy Coder’s Guide to Android Development, 2008**

20. Mastering Android NDK. Sergey Kosarevsky, Viktor Latypov. Packt Publishing Ltd., 2015