МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 Программное обеспечение информационной безопасности мобильных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к дипломному проекту на тему:**

Мобильное приложение для магазина одежды

Дипломник Ероховец Иван Александрович

(Ф.И.О.)

Руководитель проекта преп. стаж. Н. И. Уласевич

(учен. степень, звание, должность, подпись, Ф.И.О.)

И. о. заведующего кафедрой ст. преп. Е. А. Блинова

(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультант ------------------------------------

(учен. степень, звание, должность, подпись, Ф.И.О.)

Нормоконтролер --------------------------------

(учен. степень, звание, должность, подпись, Ф.И.О.)

Дипломный проект защищен с оценкой

Председатель ГЭК --------------------------------

(учен. степень, звание, должность, подпись, Ф.И.О.)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 Программное обеспечение информационной безопасности мобильных систем

УТВЕРЖДАЮ

И. о. заведующего кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Блинова

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

**ЗАДАНИЕ**

**на дипломный проект студенту**

Ероховцу Ивану Александровичу

(фамилия, имя, отчество)

1. Тема проекта: Мобильное приложение для магазина одежды

2. Тема утверждена приказом по университету от 08.05.25 № 143-с

3. Срок сдачи студентом законченного проекта: 05.06.2025 г.

4. Исходные данные к проекту (требования к системе):

4.1 Назначение: *мобильное приложение, предоставляющее возможность выбора и приобретения одежды.*

4.2 Основные функциональные возможности: *регистрация и авторизация пользователей, просмотр одежды, добавление выдранной одежды в корзину, оформление заказа, просмотр статуса заказов, сортировка, редактирование, создание и удаление одежд, удаление товаров, модерирование комментариев, выдача доступа к созданию товаров, изменение статуса.*

4.3 Пользовательские роли: *Администратор (удаление товаров, модерирование комментариев, выдача доступа к созданию товаров); Пользователь (просмотр одежды, добавление выдранной одежды в корзину, оформление заказа, просмотр статуса заказов, сортировка), Поставщик (создание, редактирование и удаление своего товара); Продавец (выдача товара со склада, изменения статуса заказа).*

4.4 Целевая аудитория: *люди, владеющие базовыми навыками использования смартфона.*

4.5 Программная платформа: *Android Studio 2024.3.1, Firebase. Flutter 3.29, Dart 3.7.2.*

5. Содержание расчетно-пояснительной записки:

1. *Реферат*
2. *Содержание*
3. *Введение*
4. *Раздел 1: обзор аналогов и постановка задачи*
5. *Раздел 2: проектирование мобильного приложения*
6. *Раздел 3: программная реализация приложения*
7. *Раздел 4: тестирование мобильного приложения*
8. *Раздел 5: руководство пользователя*
9. *Раздел 6: анализ информационной безопасности приложения*
10. *Раздел 7: технико-экономическое обоснование проекта*
11. *Заключение*
12. *Список использованных источников*
13. *Приложения и графическая часть*

6. Перечень графического материала (с точным указанием обязательных чертежей):

1. *Диаграмма развертывания*
2. *Логическая схема базы данных*
3. *Диаграмма вариантов использования*
4. *Блок-схема алгоритма оформления заказа*
5. *Блок-схема алгоритма смены пароля*
6. *Таблица расчета экономических показателей*
7. *Скриншот главной страницы*

7. Консультанты по проекту с указанием относящихся к ним разделов проекта

|  |  |
| --- | --- |
| Раздел | Консультант |
| *Технико-экономическое обоснование проекта* | *А. С. Соболевский* |

8. Дата выдачи задания:

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.И. Уласевич

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.А. Ероховец

(подпись)

9. Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов дипломного проекта | Срок выполнения этапов проекта | Примечание |
| *1* | *Обзор аналогов и постановка задачи* | 25.03–31.03.24 |  |
| *2* | *Проектирование мобильного приложения* | 01.04–07.04.24 |  |
| *3* | *Программная реализация приложения* | 08.04–28.04.24 |  |
| *4* | *Анализ информационной безопасности приложения* | 29.04–02.05.24 |  |
| *5* | *Тестирование мобильного приложения* | 03.05–06.05.24 |  |
| *6* | *Руководство пользователя* | 07.05–12.05.24 |  |
| *7* | *Технико-экономическое обоснование проекта* | 13.05–19.05.24 |  |
| *8* | *Оформление пояснительной записки* | 20.05–28.05.24 |  |
| *9* | *Рецензирование дипломного проекта* | 17.06–20.06.24 |  |
| *10* | *Допуск рабочей комиссии к защите в ГЭК* | 17.06–21.06.24 |  |
| *11* | *Защита дипломного проекта* | 24.06–26.06.24 |  |

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руководитель проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

**Реферат**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Ероховец И.А.*

Пров.

Уласевич Н.И.

Н. контр.

.

Утв.

Реферат

Лит.

Листов

1

*-, 2025*

У

**Abstract**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Ероховец И.А.*

Пров.

Уласевич Н.И.

Н. контр.

.

Утв.

Abstract

Лит.

Листов

1

*-, 2025*

У

Содержание

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Ероховец И.А*

Пров.

*Уласевич Н.И*

Н. контр.

.

.

Утв.

Содержание

Лит.

Листов

2

*-, 2025*

У

[Введение 6](#_Toc194299233)

[1 Обзор аналогов и постановка задачи 7](#_Toc194299234)

[1.1 Обзор аналогов проектируемого мобильного приложения 7](#_Toc194299235)

[1.1.1 Приложение «Lamoda» 7](#_Toc194299236)

[1.1.2 Мобильное приложение «Mark Formelle» 9](#_Toc194299237)

[1.1.3 Мобильное приложение «Ostin» 10](#_Toc194299238)

[1.2 Постановка задачи 11](#_Toc194299239)

[1.3 Выводы по разделу 11](#_Toc194299240)

[2 Проектирование мобильного приложения 13](#_Toc194299241)

[2.1 Выбор технологий и средств 13](#_Toc194299242)

[2.1.1 Среда разработки Android Studio 13](#_Toc194299243)

[2.1.2 Язык программирования Dart 14](#_Toc194299244)

[2.1.3 Облачная база данных Firebase 15](#_Toc194299245)

[2.1.4 Фреймворк Flutter 15](#_Toc194299246)

[2.2 Проектирование базы данных 16](#_Toc194299247)

[2.3 Проектирование мобильного приложения 20](#_Toc194299248)

[2.4 Диаграммы и блок-схемы приложения 22](#_Toc194299249)

[2.4.1 Диаграмма вариантов использования 22](#_Toc194299250)

[2.5.1 Диаграмма развертывания 24](#_Toc194299251)

[2.5.2 Блок-схема оформления заказа из заведения 24](#_Toc194299252)

[2.7 Выводы по разделу 25](#_Toc194299253)

# Введение

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Ероховец И.А.*

Пров.

Уласевич Н.И.

Н. контр.

.

.

.

.

.

Утв.

Введение

Лит.

Листов

1

*---, 2025*

У

В современном мире, где время является одним из самых ценных ресурсов, люди ежедневно сталкиваются с необходимостью решать повседневные задачи, требующие значительных временных затрат. Одной из таких задач является выбор и покупка одежды. С каждым годом все больше людей стремятся ускорить этот процесс, минимизировать возникающие проблемы и максимально упростить удовлетворение своих потребностей.

В ответ на эти вызовы возникла необходимость в создании мобильных приложений, которые не только упрощают процесс выбора и покупки одежды, но и предоставляют доступ к товарам из разных уголков мира. Такие приложения способны значительно экономить время пользователей, делая процесс более удобным и эффективным.

Целью данного дипломного проекта является разработка мобильного приложения для онлайн-покупки одежды. Приложение будет не только выполнять базовые функции интернет-магазина, но и предлагать дополнительные возможности: помощь в выборе одежды на основе предпочтений пользователя, доступ к отзывам, а также рекомендации, учитывающие стиль и вкусы клиент.

Для достижения цели дипломного проекта, необходимо решить ряд задач:

* изучить уже существующие аналоги и выявление их преимуществ и недостатков;
* подбор инструментов и платформ для реализации приложения;
* создание базы, содержащей всю необходимую информацию о товарах, пользователях и заказах;
* создание архитектуры, обеспечивающей удобство использования и стабильность работы;
* проектирование интуитивно понятного и визуально привлекательного интерфейса с возможностью фильтрации товаров;
* разделить пользователей в системе на три роли: администраторы, продавцы и клиенты;
* реализация функционала для трех типов пользователей — администраторов, продавцов и клиентов.

Приложение будет интегрировано с базой данных, что позволит хранить и обрабатывать все необходимые данные о товарах и пользователях. Особое внимание будет уделено созданию удобного интерфейса, который обеспечит комфортное взаимодействие с приложением.

Реализация данного проекта призвана упростить процесс выбора и покупки одежды, сделав его быстрым, удобным и доступным для пользователей. Актуальность проекта обусловлена ростом популярности онлайн-покупок, а также постоянным спросом на одежду и делает разработку такого приложения особенно важной.

# 1 Обзор аналогов и постановка задачи

ФИО

Подпись

Дата

Лист

1

БГТУ 01.00.ПЗ

Разраб.

*Ероховец И.А.*

Пров.

Уласевич Н.И.

Н. контр.

.

.

.

Утв.

1 Обзор аналогов и постановка задачи

Лит.

Листов

5

*-, 2025*

У

Для составления технических требований к проекту был проведен полный анализ схожих мобильных приложений. На основе анализа были выделены ключевые факторы, которые стоит учесть при разработке данного мобильного приложения.

## 1.1 Обзор аналогов проектируемого мобильного приложения

В настоящее время представлено достаточно количество мобильных приложений для приобретения одежды онлайн. Основным назначением такого рода приложения – предоставление возможности приобретения какого-либо товара любого предоставленного размера и других критериев. Однако не все приложение следят за актуальностью некоторых товаров, а также их наличия у поставщиков. Важными аспектами при выполнении оценки таких приложений является их функциональность и удобство использования. Также важно, чтобы приложение работало стабильно при высокой нагрузке. Это ещё больше обеспечит комфортное использование для большого количества пользователей одновременно и может кратно увеличить доходы.

Наиболее популярными на данных момент мобильные приложения с похожим функционалом являются:

* Lamoda;
* Mark Formelle;
* Ostin.

### 1.1.1 Приложение «Lamoda»

«Lamoda» – это сервис для онлайн покупок, которые предлагает большой выбор одежды и обуви.

Пользователям предоставляется выбор одежды различных брендов и категорий, таких как женская, мужская и детская одежда, а также спортивная. Lamoda предлагает покупателям разные способы доставки, можно как забрать самостоятельно, так и заказать курьера на дом. Присутствуют и различные способы оплаты, включая как оплату наличные так и онлайн-платежи.

Lamoda активно развивает своё приложение, регулярно следит за новыми трендами и пополняется новыми брендами. За счёт удобного интерфейса и функционала, приложение стало популярным среди пользователей.

На главной странице приложения пользователю предлагается выбор своей страны и дальнейший вход в систему. После входа в систему пользователь может ознакомиться с популярными предложениями, которые разбиты на определённые категории, а именно «Одежда» «Обувь» «Аксессуары» и другие, что делает поиск нужных товаров более простым и быстрым.

Приложение стало популярным, благодаря своему удобству использования. На рисунке 1.1 представлена страница со списком товаров.

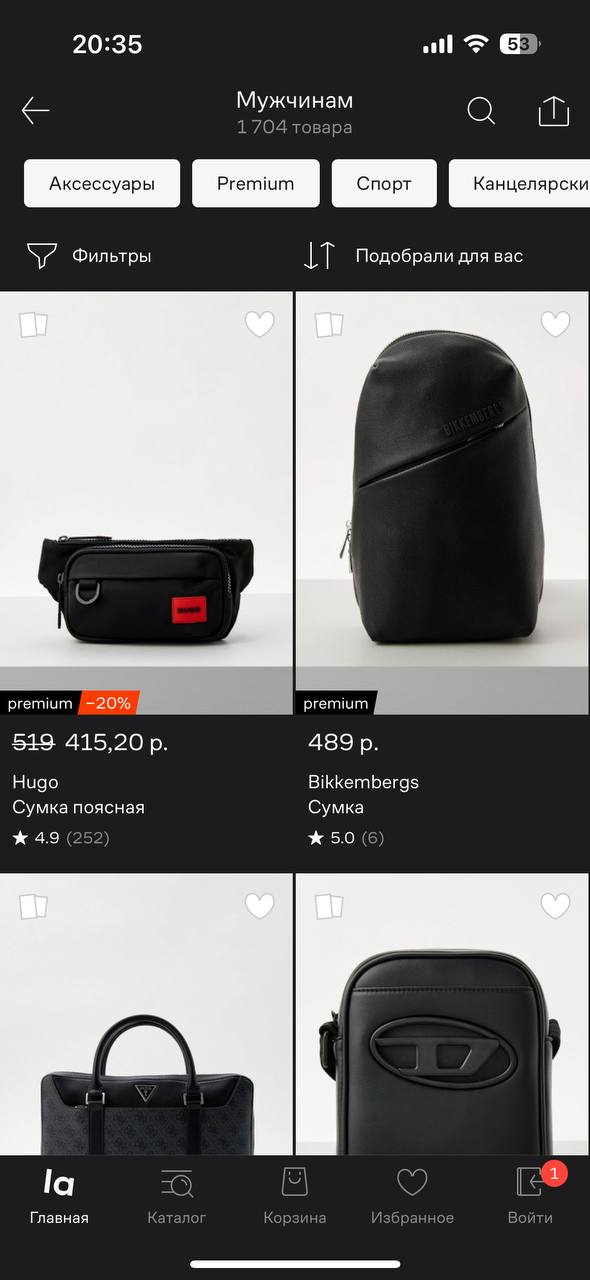


Рисунок 1.1 – Каталог товаров в приложении «Lamoda»

Каждый товар в приложении имеет свою уникальную страницу, на которой представлена вся необходимая информация для потенциального покупателя. Страница товара включает подробное описание товара, доступные размеры, актуальные цены, а также отзывы от других пользователей. Отзывы играют важную роль, так как они позволяют покупателю ознакомиться с мнением людей, которые уже приобрели данный товар. Это помогает сформировать более полное представление о качестве продукции и сделать осознанный и взвешенный выбор при покупке.

Процесс оформления заказа также хорошо продуман и максимально удобен для пользователя. После добавления товаров в корзину покупателю предлагается выбрать наиболее подходящий способ доставки, а также вариант оплаты. Для удобства пользователей предусмотрено несколько способов оплаты: это может быть оплата онлайн, с использованием банковской карты или электронного кошелька, либо оплата наличными при получении товара. Такой подход позволяет каждому покупателю выбрать наиболее комфортный для него вариант.

На рисунке 1.2 представлен интерфейс страницы с товаром.

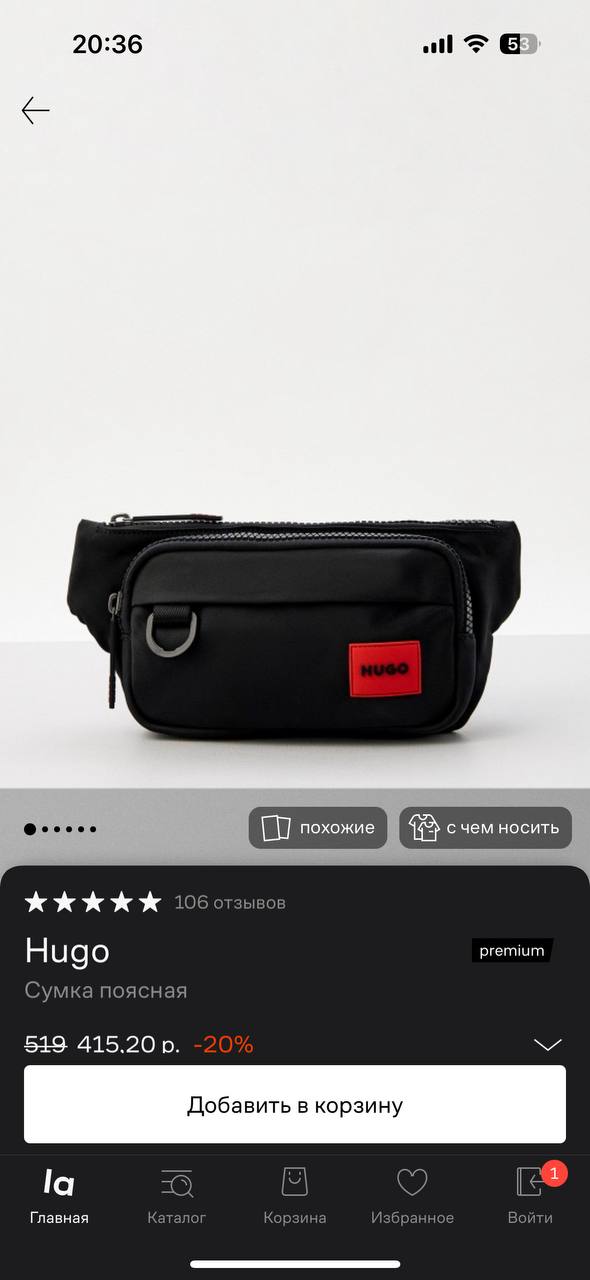


Рисунок 1.2 – Страница товара в приложении «Lamoda»

Ключевые особенности приложения Lamoda:

* широкий ассортимент одежды;
* удобный поиск и навигация по категориям;
* рейтинг и отзывы;
* возможность оформления заказа и оплаты онлайн;
* интеграция с картами для отслеживания доставки;
* гармонично выбранные цвета и дизайн.

К основным минусам можно отнести большое количества товара, что иногда усложняет навигацию..

В целом, мобильное приложение Lamoda является удобным и функциональным, особенно для тех, кто предпочитает шопинг не выходя из дома.

### 1.1.2 Мобильное приложение «Mark Formelle»

Мобильное приложение «Mark Formelle» предлагает пользователям возможности для шопинга и создания любого образа. Основная цель приложения – облегчить процесс выбора и привлечение новых клиентов, предлагая актуальные модные решения.

Приложение отличается рядом полезных возможностей, которые делают шопинг комфортным и увлекательным. Среди ключевых функций можно выделить:

* широкий ассортимент одежды;
* добавление товаров в избранное;
* получение пользовательских рекомендаций на основе предпочтений;
* доступ к эксклюзивным скидкам и акциям.

Несмотря на очевидные преимущества, приложение имеет и несколько недостатков, которые могут повлиять на пользовательский опыт. Например, для полноценного использования всех функций, включая оформление заказов и сохранение товаров в избранное, требуется обязательная регистрация. Это может стать барьером для некоторых пользователей, которые предпочитают совершать покупки без создания учетной записи.

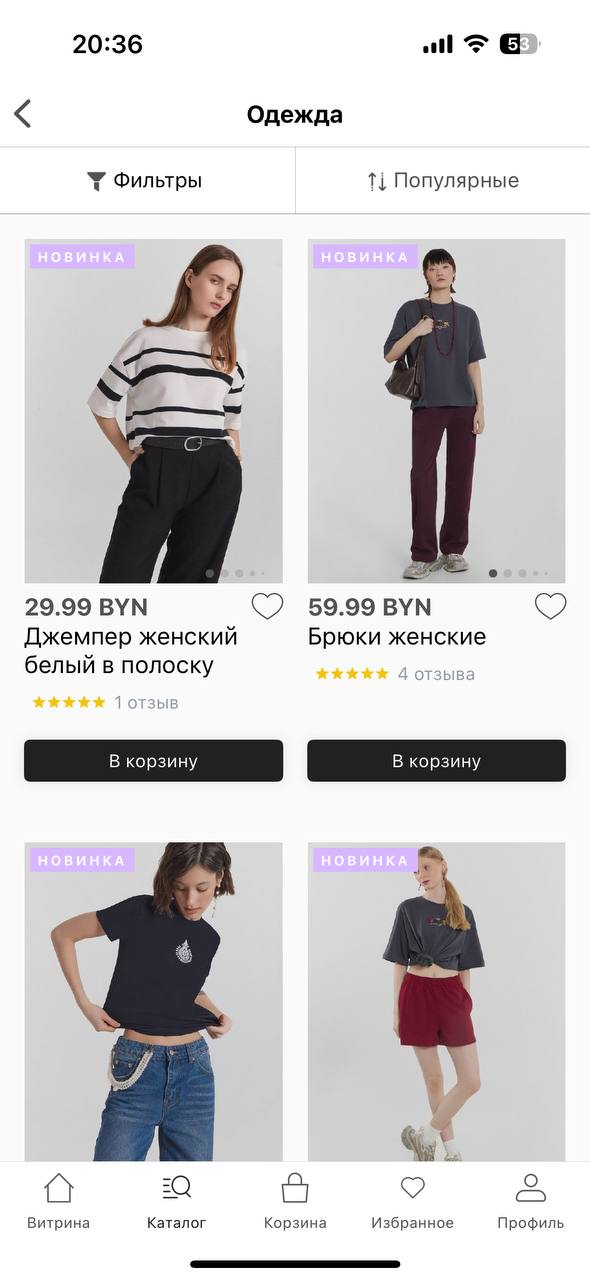


Рисунок 1.3 – Страница с каталогом в приложении «Mark Formelle»

Еще одним недостатком является отсутствие функции просмотра истории покупок, которая могла бы значительно упростить процесс повторного заказа уже приобретенных товаров. Кроме того, ассортимент приложения ограничивается исключительно одеждой бренда «Mark Formelle», что не позволяет пользователям приобретать товары других брендов. Это ограничение может снизить интерес у тех, кто ищет более разнообразный выбор.



Рисунок 1.4 – Страница с товаром в приложении «Mark Formelle»

В целом, приложение «Mark Formelle» является современным и удобным инструментом для покупки одежды, который сочетает в себе широкий функционал и приятный дизайн. Однако некоторые недостатки, такие как обязательная регистрация и ограниченность ассортимента, могут быть доработаны в будущем, что сделает приложение еще более удобным и привлекательным для пользователей.

### 1.1.3 Мобильное приложение «Ostin»

Приложение для заказа одежды и аксессуаров бренда «OSTIN» представляет собой удобный инструмент для совершения онлайн-покупок, предлагая пользователям множество полезных функций. Основное предназначение приложения заключается в том, чтобы обеспечить комфортный и быстрый доступ к товарам бренда, а также упростить процесс выбора и оформления заказа.

Пользовательский интерфейс соответствует современным стандартам мобильной торговли, что позволяет пользователям легко ориентироваться и поддерживать интерес к приложению. В нашем дипломном проекте мы также будем следовать этим принципам.

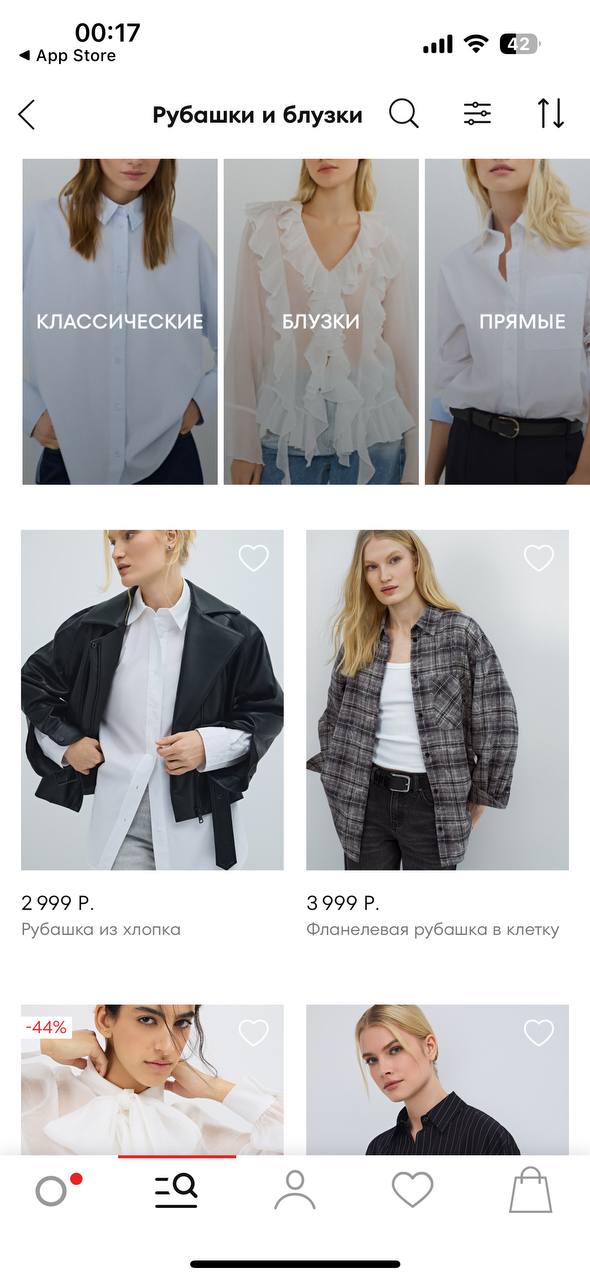


Рисунок 1.5 – Страница с каталогом в приложении «Ostin»

Меню приложения также заслуживает внимания. Оно позволяет выбирать товары по категориям, что упрощает навигацию по каталогу. Пользователи могут искать нужные вещи по названию, добавлять их в корзину с помощью соответствующей кнопки, а затем переходить в раздел корзины для управления выбранными товарами. В корзине можно изменять количество товаров, видеть общую сумму заказа, а также оформить покупку, нажав кнопку «Оформить заказ».

К главным особенностям данного приложения можно отнести:

* широкий выбор одежды;
* возможность отслеживания истории покупок;
* простой процесс оформления и изменения заказов.

На рисунке 1.4 можно увидеть страницу с товаром.



Рисунок 1.3 –Страница товара в приложении «Ostin»

Несмотря на свои достоинства, приложение имеет и некоторые недостатки. Например, для его полноценной работы требуется стабильное подключение к интернету. Кроме того, информация о наличии товаров и текущих акциях не всегда обновляется вовремя, что может вызывать неудобства у пользователей. Однако приложение предоставляет возможность просмотреть и изменить детали заказа перед его подтверждением, что обеспечивает дополнительный контроль над процессом покупки и делает взаимодействие с ним максимально комфортным.

В целом, приложение OSTIN обладает широким функционалом, который охватывает основные потребности пользователей. Оно практически не содержит лишних элементов, что делает его удобным инструментом для шопинга. Тем не менее, улучшение точности обновления данных о наличии товаров и акциях могло бы сделать это приложение еще более привлекательным для пользователей.

## 1.2 Постановка задачи

Разрабатываемый проект, является мобильным приложением для выбора и приобретения какой либо части гардероба человека, оформление, заказ, а так-же отслеживания товара и у приложения должен быть следующий функционал:

* регистрация и авторизация пользователей;
* возможность просмотра товара;
* возможность фильтрации товара;
* возможность выбора товара по размеру и количества;
* возможность добавления товара в корзину;
* возможность оформления заказа и его доставку;
* возможность пользователю просматривать список всех его заказов.

## 1.3 Выводы по разделу

В данной главе были разобраны три самых популярных приложения, которые являются аналогами приложения «Мобильное приложение для магазина одежды**»**»: «Lamoda», «Mark Formelle» и «Ostin». Мы остановились на трёх приложениях, так как остальные практически не отличаются по функционалу и пользовательскому интерфейсу.

Из приложений «Lamoda» я выделю то, что нужно работать во всех направлениях, а именно сотрудничать с разными брендами и сделать личный кабинет со всей информацией о заказ и пользователе, а из «Ostin» и «Mark Formelle» –взять за идею карточку товара, которая сделана достаточно просто и красиво, без лишних деталей.

Проанализировав все вышесказанное можно выделить основной функционал разрабатываемого в курсовом проекте мобильного приложения для магазина одежды:

* регистрация и авторизация пользователей;
* наличие нижней навигационной панели;
* возможность просмотра информации в личном кабинете;
* поиск и сортировка товара по разным параметрам;
* просмотр заказов и их статусов.

Стоит отметить, что анализ аналогичных приложений позволяет выявить лучшие практики и технологии, применяемые другими разработчиками. Это может помочь оптимизировать функциональность и удобство использования нашего приложения, что подчеркивает важность такого обзора.

# 2 Проектирование мобильного приложения

ФИО

Подпись

Дата

Лист

1

БГТУ 02.00.ПЗ

Разраб.

*Ероховец И.А.*

Пров.

Уласевич Н.И.

Н. контр.

.

.

.

.

.

Утв.

2 Проектирование мобильного приложения

Лит.

Листов

15

*-, 2025*

У

## 2.1 Выбор технологий и средств

При реализации дипломного проекта использовались язык программирования Dart и фреймворк Flutter.

В качестве средств разработки, использовались программные средства Android Studio, Visual Studio Code.

### **2.1.1 Среда разработки Android Studio**

Android Studio — это мощная и бесплатная среда разработки, разработанная компанией Google специально для создания приложений на платформе Android. Она предоставляет обширный набор инструментов, которые значительно упрощают процесс разработки и развертывания мобильных приложений.

Android Studio предлагает широкий набор функций, направленных на повышение производительности разработчика. Среди них поддержка подсветки синтаксиса, автодополнения кода, мощные средства отладки и профилирования, а также интеграция с системами контроля версий, такими как Git. Эти возможности помогают разработчикам сосредоточиться на написании кода, снижая затраты времени на рутинные задачи.

Одним из главных преимуществ Android Studio является встроенная поддержка Gradle-системы автоматизации сборки. Gradle позволяет эффективно управлять зависимостями, конфигурациями сборки и процессом развертывания, что особенно полезно при работе с крупными проектами.

Для упрощения тестирования приложений, Android Studio предоставляет мощные инструменты, такие как эмуляторы устройств, позволяющие запускать и тестировать приложения на различных конфигурациях Android без необходимости использования физического устройства. Это значительно ускоряет процесс отладки и улучшения качества приложения.

Одной из ключевых особенностей Android Studio является его тесная интеграция с Android SDK, что обеспечивает разработчикам доступ ко всем новейшим API и библиотекам. Это позволяет быстро адаптироваться к изменениям в платформе Android и использовать новейшие возможности операционной системы.

Android Studio разработана с учетом производительности и ресурсоэффективности, что позволяет комфортно работать с ней даже на компьютерах со скромными характеристиками. Кроме того, среда разработки поддерживает настройку пользовательского интерфейса, что позволяет разработчикам адаптировать ее под свои индивидуальные предпочтения.

На протяжении многих лет среда разработки Android Studio зарекомендовала себя как стандарт для разработки приложений под Android, благодаря своей стабильности, функциональности и поддержке сообщества. Она предоставляет все необходимые инструменты для создания высококачественных мобильных приложений, что делает ее незаменимым инструментом для Android-разработчиков. Помимо этого, регулярные обновления и улучшения Android Studio помогают разработчикам оставаться в курсе новейших технологий и методик, что способствует созданию инновационных и современных приложений.

### 2.1.2 Язык программирования Dart

Dart — это язык программирования общего назначения, разработанный компанией Google, предназначенный прежде всего для создания прикладных приложений. Его основной сферой применения на сегодняшний день является разработка графических приложений с использованием кроссплатформенного фреймворка Flutter. Flutter позволяет создавать клиентские веб-приложения, мобильные приложения под Android и iOS, а также десктопные приложения для Windows, macOS и Linux. Это означает, что мы можем написать программу на Dart с общим кодом и скомпилировать её сразу под все поддерживаемые платформы, включая Android, iOS, Windows, Linux, macOS и веб-браузеры. Однако применение языка Dart не ограничивается только фреймворком Flutter.

В своем развитии Dart испытал влияние более ранних языков, таких как Smalltak, Java, JavaScript. Его синтаксис похож на синтаксис других си-подобных языков. При этом Dart - объектно-ориентированный язык. Все значения, которые используются в программе на Dart, представляют объекты.

Dart предоставляет мощные инструменты и функции, которые существенно упрощают создание мобильных и веб-приложений. Он гарантирует отличную производительность и удобство использования благодаря статической и динамической типизации, что дает возможность выявлять ошибки на этапе компиляции и одновременно обеспечивает гибкость в написании кода. Эти особенности способствуют созданию более надежного и безопасного кода программистами, позволяя сократить время на его отладку.

Одним из ключевых преимуществ Dart является его асинхронная модель программирования, которая позволяет эффективно обрабатывать параллельные задачи, что особенно полезно в приложениях с высокой нагрузкой. Это делает Dart идеальным выбором для разработки отзывчивых пользовательских интерфейсов и работы с сетевыми запросами.

Dart также предлагает обширную стандартную библиотеку, содержащую множество полезных функций и классов для работы с коллекциями, файлами, сетевыми запросами и другими задачами. Данная библиотека существенно уменьшает время разработки, позволяя сконцентрироваться на логике приложения.

### **2.1.3 Облачная база данных Firebase**

Firebase — платформа для разработки мобильных и веб-приложений, разработанная компанией Google. Она представляет собой обширный инструментарий и сервисы, позволяющие разработчикам оперативно создавать, тестировать и внедрять приложения, а также управлять их функциональностью и производительностью.

Основным элементом Firebase является облачное хранилище данных Firestore, предоставляющее возможности хранения и синхронизации данных в режиме реального времени. Firestore упрощает создание и управление базами данных, обеспечивая высокую доступность и масштабируемость приложений. Поддержка оффлайн-режима гарантирует возможность работы с приложением даже при временном отсутствии интернет-соединения.

Firebase также предлагает мощные средства аутентификации пользователей, поддерживающие различные методы, включая электронную почту, социальные сети и анонимный вход. Это упрощает управление пользователями и повышает безопасность приложений.

В дополнение, Firebase содержит инструменты аналитики, которые помогают разработчикам отслеживать поведение пользователей и оценивать производительность приложений. Инструменты аналитики предоставляют ценную информацию о взаимодействии пользователей с приложением, что позволяет принимать обоснованные решения по его улучшению.

Firebase предоставляет возможность отправки уведомлений в реальном времени посредством Firebase Cloud Messaging (FCM). Этот сервис позволяет разработчикам отправлять push-уведомления пользователям на мобильных устройствах и в веб-приложениях, что способствует повышению вовлеченности и удержанию пользователей.

Другой значимой функцией Firebase является хостинг, позволяющий разработчикам быстро разворачивать статические и динамические веб-приложения с поддержкой HTTPS. Хостинг Firebase обеспечивает высокую скорость загрузки и легкость управления, что делает его отличным выбором для веб-разработки.

Firebase также интегрируется с другими сервисами Google, такими как Google Analytics и Google Cloud, что позволяет разработчикам использовать мощные инструменты для анализа и масштабирования приложений.

За последние годы Firebase подтвердил свою репутацию надежной платформы для разработки приложений, благодаря своей функциональности, удобству использования и поддержке сообщества. Регулярные обновления и новые функции делают Firebase современным инструментом, который помогает разработчикам создавать качественные и высокопроизводительные приложения.

### **2.1.4 Фреймворк Flutter**

Flutter — это бесплатный и невероятно удобный инструмент разработки приложений, созданный компанией Google. С его помощью можно создавать приложения не только для мобильных платформ Android и iOS, но и для веба и настольных систем, что делает его идеальным вариантом для разработчиков,стремящихсякэкономиивремениисил,разрабатываякроссплатформенные решения.

Главное преимущество Flutter – его обширный функционал. Он включает в себя богатую библиотеку виджетов, а также функцию "горячей перезагрузки", позволяющую мгновенно видеть изменения в коде. Это значительно ускоряет процесс разработки. К тому же, интеграция с различными инструментами тестирования и отладки облегчает поиск и исправление ошибок.

Flutter также обеспечивает отличную производительность благодаря своей системе рендеринга. Он использует собственный движок для графики и анимации, поэтому приложения выглядят одинаково хорошо на всех платформах и работают плавно и быстро.

Для тестирования Flutter предоставляет мощные инструменты, включая фреймворки для модульного и интеграционного тестирования. Это позволяет проверять каждую часть приложения, обеспечивая его качество на всех этапах разработки.

Ещё одно значительное преимущество Flutter — возможность использования единой кодовой базы для всех платформ. Разработчики пишут код один раз, и он работает на Android, iOS, вебе и десктопах. Это значительно экономит время и усилия, избавляя от необходимости адаптировать приложение под каждую платформу отдельно.

Не стоит забывать и о поддержке сообщества. Flutter постоянно обновляется, а разработчики могут использовать множество пакетов и библиотек, упрощающих работу. Все эти обновления и улучшения способствуют созданию современных, удобных и функциональных приложений.

В целом, Flutter стал важным инструментом для многих разработчиков благодаря своей гибкости, производительности и активному сообществу. Этот фреймворк позволяет создавать отличные кроссплатформенные приложения, и он подходит для разработчиков, желающих работать быстро и эффективно.

## 2.2 Проектирование базы данных

Первым шагом в разработке приложения является определение предметной области, которой оно будет охватывать, а также проектирование общей архитектуры базы данных. Необходимо четко понимать, какие именно данные будут обрабатываться, где они будут размещены и в каком формате будут представлены.

Логическая схема базы данных со всеми связями представлена на рисунке 2.2.

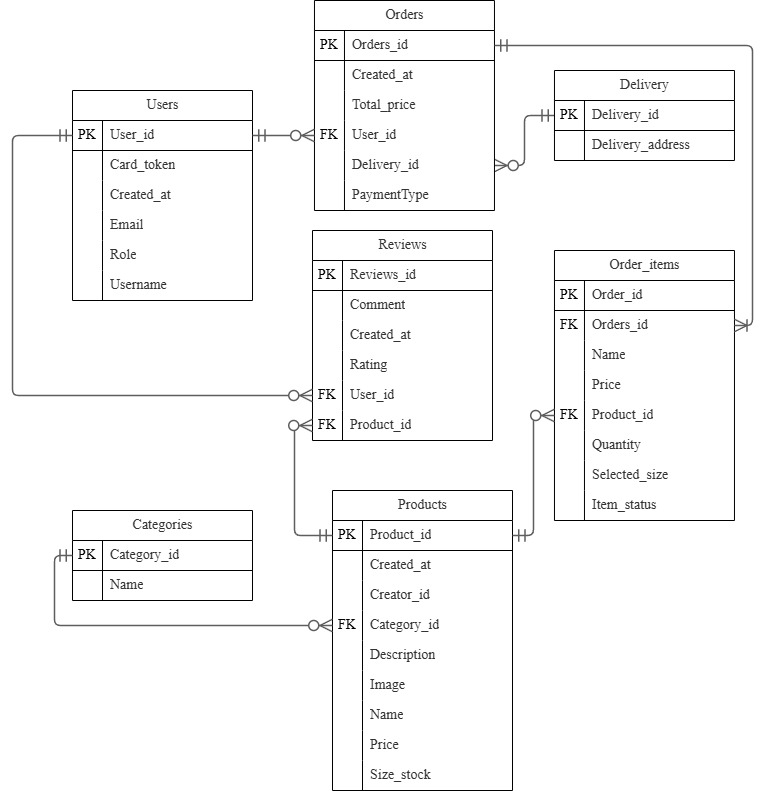


Рисунок 2.2 – Схема базы данных

Всего в базе данных 7 таблиц. Название и назначение каждой спроектированной коллекции приведено в таблице 2.1.

Таблица 2.1 – Таблицы базы данных

|  |  |
| --- | --- |
| Название таблицы | Назначение таблицы |
| Users | Хранит данные о пользователях |
| Orders | Данные о списке всех заказов |
| Products | Хранит информацию о товаре |
| Reviews | Хранит отзывы о товаре |
| Delivery | Хранит возможные места доставки товара |
| OderItems | Данные о каждом отдельном заказе |
| Categories | Хранит доступные категории товаров |

Подробное описание каждой коллекции приведено в таблицах 2.2 – 2.8. А именно описаны типы и названия полей, а также приведено их краткое описание.

Таблица 2.2 – Структура таблицы Delivery

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| DeliveryId | string | Уникальный идентификатор |
| DeliveryAddress | string | Точный адрес доставки |

В таблице Delivery ( таблица 2.2 ) содержится информация о возможных местах доставки товара. Благодаря данной таблице можно ознакомиться с пунктами доставки, а также содержит в себе внешний ключ для связи с таблицей Orders (таблица 2.5 ).

Вся информация о товаре, хранится в таблице Products ( таблица 2.3 ). Благодаря данной таблице, продавцы могу добавлять, изменять или удалять информацию о товарах, тем самым всегда поддерживать актуальность в наличии или точной стоимости каждой позиции.

При создании товара, когда продавец вводит данные товара, они сохраняются в базе данных в таблице Products (таблица 2.3). Каждая строка в таблице Products представляет собой отдельный товар и содержит следующую информацию: идентификатор товара, дату создания товара, идентификатор категории товара, идентификатор создавшего товар, описание товара, изображение с товаром, название товара, цена, доступные размеры и количество.

Таблица 2.3 – Структура таблицы Products

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| ProductId | number | Уникальный идентификатор товара |
| CreatedAt | timestamp | Дата регистрации товара |
| CategoryId | number | Идентификатор категории |
| CreatorId | string | Идентификатор пользователя |
| Description | string | Дополнительная информация о товаре |
| Image | string | Изображение товара |
| Name | string | Полное название товара |
| Price | number | Цена товара |
| SizeStock | map | Размер и количество на складе |

Хранение данных в таблице Users обеспечивает возможность сохранения и извлечения информации о пользователях, что позволяет администратору просматривать общее количество зарегистрированных клиентов.

Таблица 2.4 – Структура таблицы Users

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| 1 | 2 | 3 |
| UserId | string | Уникальный идентификатор пользователя |
| CardToken | string | Токен карты клиента |
| CreatedAt | timestamp | Дата создания аккаунта |
| Email | string | Электронная почта пользователя |
| Role | string | Роль пользователя |
| Username | string | Никнейм пользователя |

После оформления заказа, информация об идентификаторе пользователя, датой добавления товара, статусом заказа, общей суммой, идентификатор покупателя, адресом доставки и типом оплаты попадает в таблицу Orders (таблица 2.5). После чего, пользователь, благодаря полю с информацией о статусе заказа, сможет отслеживать его готовность в своем личном кабинете.

Таблица 2.5 – Структура таблицы Orders

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| OrdersId | string | Уникальный идентификатор заказ(а/ов) |
| CreatedAt | timestamp | Дата и время заказа |
| TotalPrice | number | Общая сумма заказа |
| UserId | string | Статус заказа |
| DeliveryId | number | Адрес доставки заказа |
| PaymentType | string | Тип оплаты |

Также в таблице Reviews (таблица 2.6). хранится о отзывах, которые клиент загружает о приобретенном товаре, а именно идентификатор отзыва, текст отзыва, дата создания, оценку товара, идентификатор товара, идентификатор пользователя оставившего отзыв.

Таблица 2.6 – Структура таблицы Reviews

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| ReviewsId | string | Уникальный идентификатор отзыва |
| Comment | string | Текст отзыва |
| CreatedAt | timestamp | Дата создания |
| Rating | number | Оценка товара от 0 до 5 |
| ProductId | number | Идентификатор товара |
| UserId | string | Идентификатор оставившего отзыв |

В таблице Categories ( таблица 2.7 ) представлена информация о категории товара, а именно его Id и название.

Таблица 2.7 – Структура таблицы Categories

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| CategoryId | number | Уникальный идентификатор категории товара |
| Name | string | Название категории |

В таблице OrderItems ( таблица 2.8 ) хранится информация о конкретном заказе, а именно идентификатор заказа, идентификатор заказов, название товара, цена, идентификатор товара, количество и его размер.

Таблица 2.8 – Структура таблицы OrderItems

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Назначение |
| OrderId | string | Уникальный идентификатор заказа |
| OrdersId | string | Идентификатор заказов |
| Name | string | Название товара |
| Price | number | Цена товара |
| ProductId | number | Идентификатор товара |
| Quantity | number | Количество товара |
| SelectedSize | string | Размер товара |
| Status | string | Статус товара |

По итогу мы получаем базу данных, которая соответствует как минимум первым трем формам нормализации.

## 2.3 Проектирование мобильного приложения

Для создания мобильного приложения был выбран язык программирования *Dart* и фреймворк *Flutter* [7]. Использование данного языка и фреймворка предоставляет ряд значимых преимуществ:

* высокая производительность благодаря компиляции в нативный код, что обеспечивает высокую скорость работы и быструю загрузку;
* встроенная поддержка асинхронного программирования, что позволяет управлять асинхронными задачами и повышает отзывчивость приложения;
* поддержка *JIT* компиляции для быстрой разработки и тестирования.

Фреймворк *Flutter* дополняет преимущества Dart следующими функциями:

* *Hot Reload*, который дает мгновенное применение изменений в коде и немедленный просмотр результата в режиме реального времени, что ускоряет процесс разработки и тестирования;
* использование готовых настраиваемых виджетов, которые следуют рекомендациям по дизайну *Material Design* (для *Android*) и *Cupertino* (для *iOS*), что позволяет быстро создавать качественные и стильные интерфейсы.

Кроме того, использование Dart и Flutter обеспечивает единообразие в разработке, поскольку Dart является основным языком программирования для Flutter, что упрощает командную работу и поддержку проекта. Подход Flutter к созданию интерфейса также позволяет создавать адаптивные приложения, которые могут автоматически адаптироваться под разные устройства и экраны. Благодаря активной и развивающейся сообществу разработчиков, Dart и Flutter постоянно обновляются и улучшаются, что обеспечивает доступ к новейшим технологиям и инструментам для создания высококачественных мобильных приложений.

Для создания мобильного приложения необходимо в терминале ввести команду «*flutter create shop\_project*», которая автоматически создает проект с разделением на папки и структурированным кодом.

Папка *lib* содержит файл *main.dart*, который является главным файлом приложения. Именно там начинается выполнение приложения. Остальные файлы в этой папке содержат коды страниц приложения. В каждом файле отдельная страница представлена в виде класса, в котором описывается разметка страницы, а также обработка всех функций и создание маршрутов для связи с сервером.

Папка test содержит тесты для приложения, где можно создавать файлы тестов для различных частей приложения, таких как виджеты или модели.

Папка *build* создается автоматически при сборке приложения и содержит артефакты сборки. Папка *android* содержит конфигурационные файлы и код для сборки приложения под *Android*. Папка *ios* содержит конфигурационные файлы и код для сборки приложения под *iOS*. Папка *web* содержит файлы для сборки приложения под веб. Здесь находятся файлы *index.html*, *main.dart.js* и другие ресурсы для веб-сборки. Папка *macos* содержит конфигурационные файлы и код для сборки приложения под *macOS*. Папка *windows* содержит конфигурационные файлы и код для сборки приложения под *Windows*. Папка *linux* содержит конфигурационные файлы и код для сборки приложения под *Linux*.

Файл *pubspec.yaml* используется для управления зависимостями проекта. Здесь указываются пакеты, которые используются в приложении, а также такие настройки, как название приложения, версия и ресурсы.

Файл *README.md* содержит описание проекта, инструкции по установке и использованию. В папке *assets* хранятся статические ресурсы приложения, такие как изображения, шрифты. Эти ресурсы указываются в файле *pubspec.yaml*.

Пример описанной архитектуры приведен на рисунке 2.3.

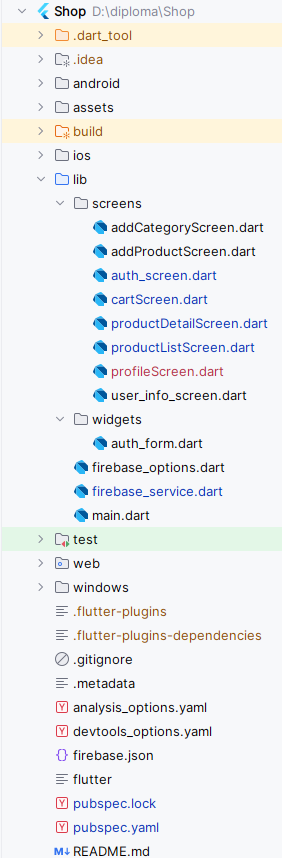


Рисунок 2.3 – Пример архитектуры файлов мобильного приложения

Страницы являются основополагающими компонентами приложения. На них ведет маршрутизация со стороны клиента. Каждая страница отвечает за отдельные функции, предназначенные именно ей.

В *Flutter* приложении страницы представляют собой виджеты, которые описывают пользовательский интерфейс для конкретного экрана. Эти виджеты могут включать в себя различные элементы: текст, изображения, кнопки и формы.

Маршрутизация между страницами осуществляется с помощью навигационных методов, таких как *Navigator.push* и *Navigator.pop*, которые позволяют перемещаться вперед и назад между страницами.

Каждая страница в приложении может иметь свою собственную логику и состояние, управляемое с помощью различных методов управления состоянием, таких как *StatefulWidget*, *Provider* или *Bloc*.

Страницы также могут взаимодействовать друг с другом, передавая данные через параметры маршрутизации или используя глобальные состояния.

Страницы были разделены на следующие логические сущности:

* *addCategoryScreen* – страница добавления категории;
* *addProductScreen* – страницы добавления товара;
* authScreen – страница авторизации;
* *cartScreen* – страница корзины;
* *productDetailScreen* –страница с товаром;
* *productListScreen* – страница со списком товаров;
* *profileScreen* – страница профиля;

Такое большое количество страниц было необходимо для обеспечения максимального комфорта и интуитивного взаимодействия пользователя с мобильным приложением. А также для разбиения классов, что удобно при написании кода, когда нужно обратиться к определенному классу. Всегда удобно, когда на определенную функцию приложения есть своя страница.

## **2.4 Диаграммы и блок-схемы приложения**

В данном подразделе будут представлены такие блок-схемы и диаграммы как:

* диаграмма вариантов использования;
* диаграмма компонентов;
* блок-схема оформления заказа.

### **2.4.1 Диаграмма вариантов использования**

В диаграмме вариантов использования, отображено разделение возможностей администратора, пользователя, поставщика и продавца.

Диаграмма вариантов использования администратора обозначена на рисунке 2.5.

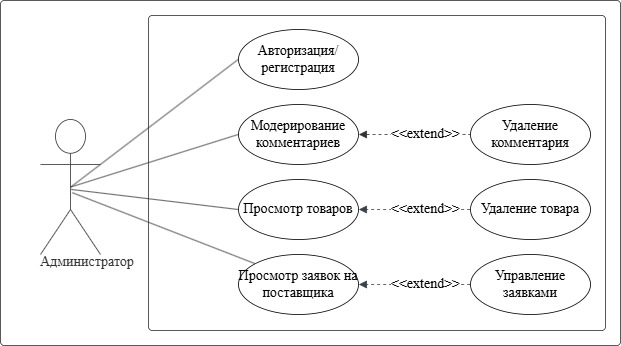


Рисунок 2.4 – Диаграмма вариантов использования администратора

Администратор может управлять заказами и их статусами, а также отзывами. Диаграмма вариантов использования пользователя представлена на рисунке 2.6.

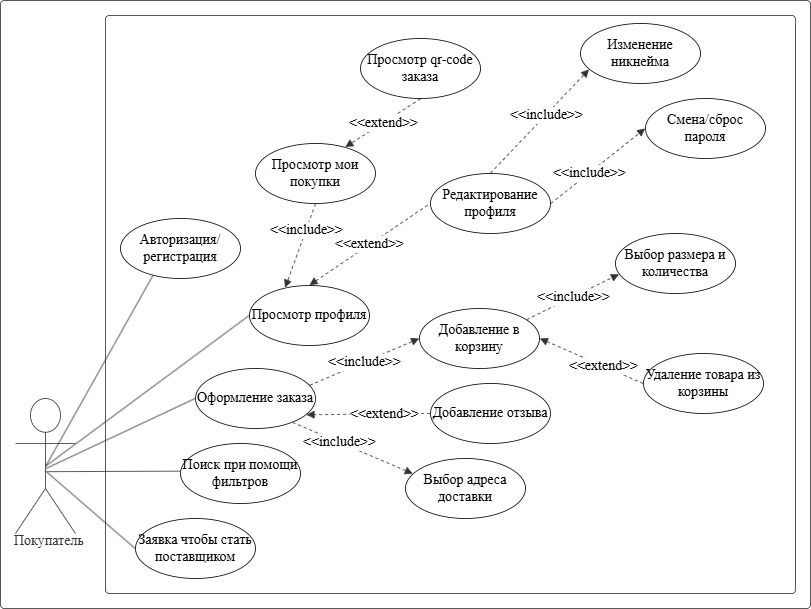


Рисунок 2.5 – Диаграмма вариантов использования пользователя

Пользователь может осуществлять поиск и добавлять необходимые ему товары в корзину, после чего оформлять заказ добавленных в корзину товаров, просматривать заказы.

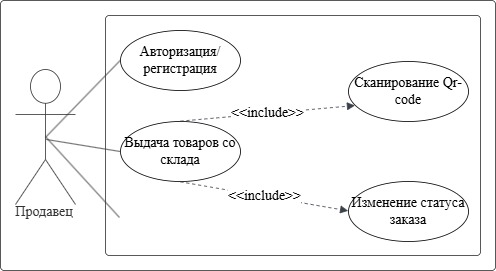


Рисунок 2.5 – Диаграмма вариантов использования продавца

Продавец может осуществлять редактирование и создания собственного товара.



Рисунок 2.6 – Диаграмма вариантов использования поставщика

Поставщик может осуществлять создание и редактирование собственных товаров, а также их удаление.

### **2.5.1 Диаграмма развертывания**

Диаграмма развертывания помогает визуализировать архитектуру системы и понять, как различные компоненты взаимодействуют друг с другом.

Для разработки мобильного приложения на основе операционной системы Android был задействован язык Dart. В качестве БД использовался Firebase Database.

Диаграмма развертывания представлена на рисунке 2.7.

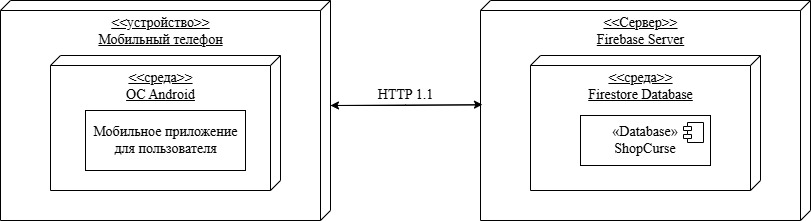


Рисунок 2.6 – Диаграмма развертывания

Цель диаграммы развертывания – предоставить обзорную картину архитектуры системы, выделить основные компоненты и их взаимодействие, а также помочь разработчикам и архитекторам понять и изменять систему.

### **2.5.2 Блок-схема оформления заказа из заведения**

Блок-схема является мощным инструментом, который позволяет визуализировать и упорядочить последовательность действий, потоки данных, принятие решений, ветвления и циклы в компактной и понятной форме. Она представляет собой графическое изображение, которое помогает лучше понять сложные процессы.

В блок-схеме каждое действие или компонент представлен прямоугольным блоком, в котором четко указано его название или описание. Эти блоки соединены линиями, которые обозначают потоки данных, управления или связи между компонентами системы. Линии могут иметь различные формы и типы стрелок, что позволяет указать на разные виды связей и направлений потока информации.

Использование блок-схемы позволяет не только визуализировать процесс, но и выявить потенциальные ошибки и исключения, что, в свою очередь, способствует повышению надежности системы. Благодаря этому разработчики могут заранее обнаружить слабые места в алгоритме и предусмотреть возможные проблемы.

Кроме того, блок-схема значительно облегчает коммуникацию между членами команды. Все участники проекта могут быстро и четко понять сложные процессы, что способствует более эффективному взаимодействию и совместной работе. Это особенно важно в командах с разными уровнями опыта и понимания.

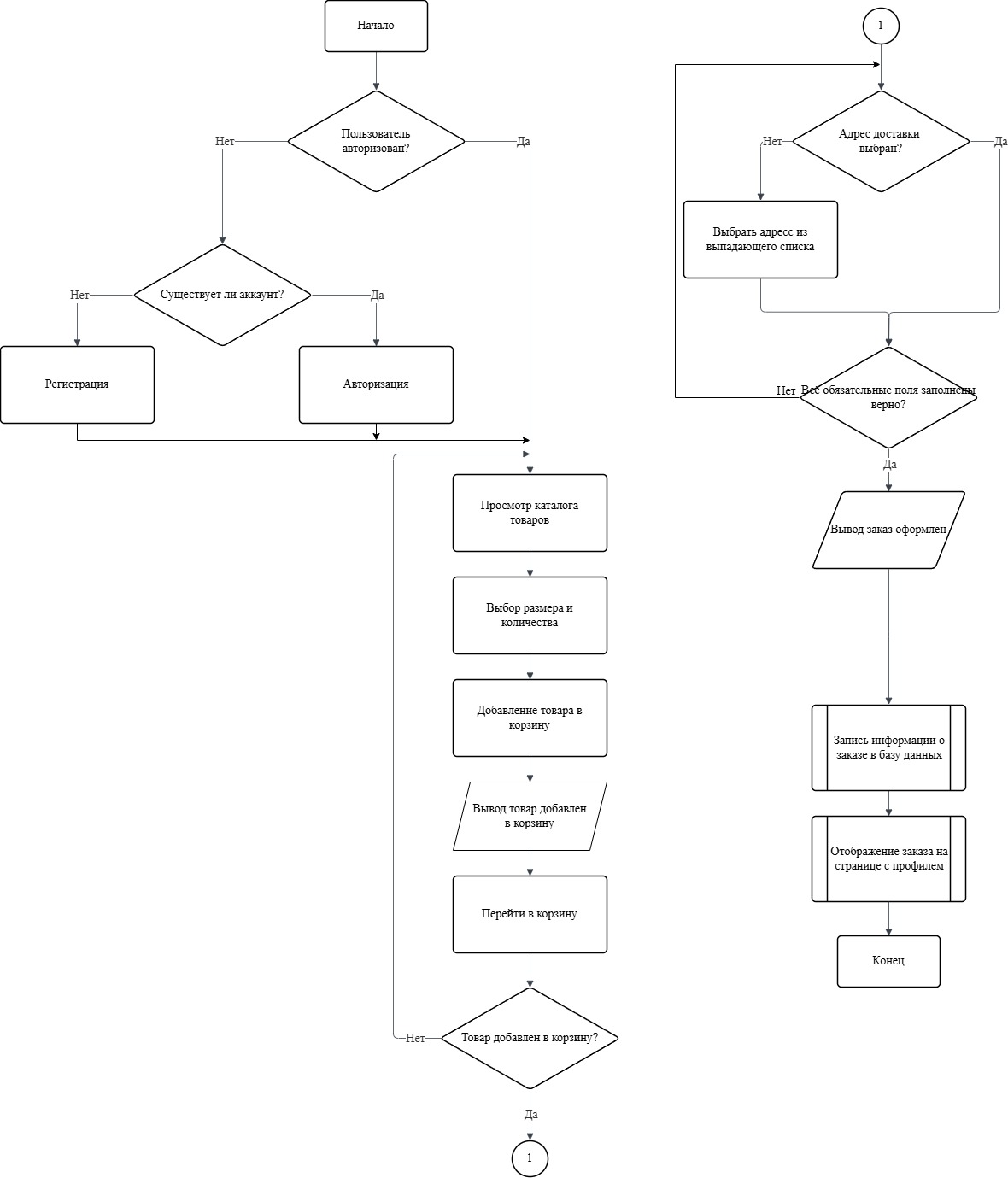


Рисунок 2.8 – Блок-схема оформления заказа

Блок-схема является удобным инструментом для анализа и проектирования алгоритмов программ, описания бизнес-процессов, планирования задач и разработки архитектуры систем.

### **2.5.3 Блок-схема смены пароля**

Также на рисунке 2.8 представлена блок-схема смены пароля пользователям. Которая отражает все возможные способы смены пароля с различными статусами пользователя.

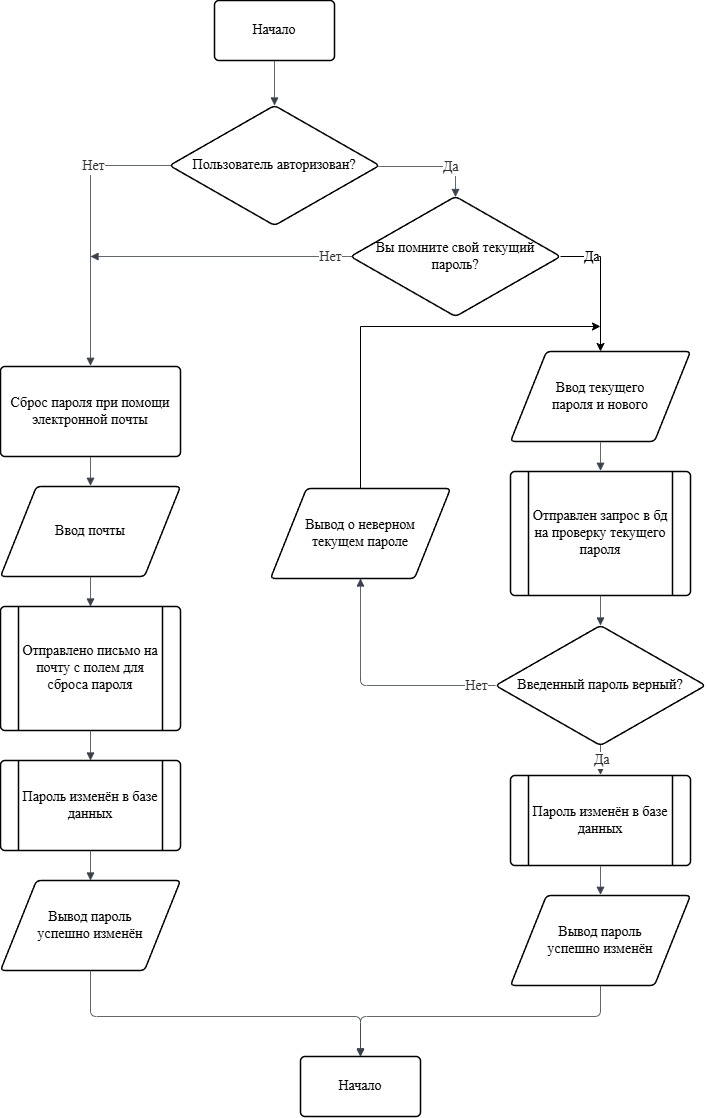


Рисунок 2.9 – Блок-схема смены пароля

Смена пароля одна из важных составляющих для сохранения конфиденциальности и безопасности пользователя приложения.

## 2.7 Выводы по разделу

В данной главе было рассмотрено проектирование каждого отдельного компонента приложения. По части мобильного приложения был определен основной функционал приложения, также были разработаны диаграмма вариантов использования приложения, была также спроектирована база данных.

Список использованных источников

1. Главная страница сайта Lamoda [Электронный ресурс] / lamoda.ru– Режим доступа https://www.lamoda.ru/men-home/?sitelink=topmenuM – Дата доступа: 30.03.2025.
2. Главная страница сайта Mark Formelle [Электронный ресурс] / markformelle.by – Режим доступа https://markformelle.by/magaziny/– Дата доступа: 31.03.2025.
3. Главная страница сайта Ostin [Электронный ресурс] / ostin.com – Режим доступа: https://ostin.com/?utm\_referrer=https%3A%2F%2Fwww.google.com%2F/– Дата доступа: 25.03.2025.
4. Главная страница сайта Android Studio [Электронный ресурс] / developer.android.com – Режим доступа https://developer.android.com/studio/intro – Дата доступа: 15.03.2025.
5. Android documentation for app developers. The activity lifecycle [Электронный ресурс] / developer.android.com – Режим доступа: <https://developer.android.com/guide/components/activities/activity-lifecycle> – Дата доступа: 11.03.2025.
6. Введение в язык Dart [Электронный ресурс] / Metanit – 2025. – Режим доступа: https://metanit.com/dart/tutorial/1.1.php – Дата доступа: 20.03.2025.