МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1–40 01 01 Программное обеспечение информационных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ НА ТЕМУ:**

«Реализация web-приложения магазина одежды»

Выполнил студент Яшный Никита Сергеевич

(Ф.И.О.)

Руководитель работы ст. преп. Дубовик Марина Владимировна

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

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

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

Курсовая работа защищена с оценкой

Минск 2024

**Содержание**

[Введение 4](#_Toc161915730)

[1 Постановка задачи 5](#_Toc161915731)

[1.1 Аналитический обзор аналогов 5](#_Toc161915732)

[1.1.1 Zara 5](#_Toc161915733)

[1.1.2 Bershka 6](#_Toc161915734)

[1.1.3 Pull & Bear 6](#_Toc161915735)

[1.1.4 Lamoda 7](#_Toc161915736)

[1.2 Разработка функциональных требований определение вариантов использования 8](#_Toc161915737)

[1.3 Вывод 9](#_Toc161915738)

[2 Проектирование web-приложения 10](#_Toc161915739)

[2.1 Архитектура приложения 10](#_Toc161915740)

[2.2 Общая структура приложения 10](#_Toc161915741)

[2.3 Структура базы данных 10](#_Toc161915742)

[2.4 Вывод 12](#_Toc161915743)

# Введение

В современных реалиях самым ценным ресурсом человека является время. Новый темп жизни не позволяет тратить этот ограниченный ресурс на посещение физических магазинов для ознакомления с ассортиментом или же для поиска необходимого предмета гардероба. Постоянное развитие и увеличение количества молодых брендов является доступной роскошью для людей, проживающих в больших городах, однако население, живущее в населенных пунктах малого размера, не имеет данных привилегий.

Данное программное средство позволяет решать многочисленные проблемы пользователей, включая ознакомление с ассортиментом бренда и подробной информацией по каждой позиции, актуальными предложениями, наличием определенных товаров в магазинах и дает возможность сравнивать цены.

Возможность создания личного кабинета позволяет пользователю добавлять понравившиеся модели в избранное, сравнивать их, оформлять заказ, указывая адрес доставки, получать актуальную информацию по электронной почте и иметь доступ к истории заказов.

Данное программное средство незаменимо для администратора, который имеет удаленный доступ к базе данных, может проверять и изменять ее наполнение, выполнять поиск и сортировку товаров, отслеживать заказы и осуществлять помощь покупателям, имея доступ к их личному кабинету. Основные задачи к приложению:

* Обеспечение возможности регистрации и авторизации;
* Поддержка ролей администратора и пользователя;
* Наличие оповещения клиента о выполнении заказа по email;
* Обеспечение возможности выполнять поисковые запросы;
* Обеспечение предоставления каталога товаров;
* Обеспечение возможности добавления товаров администратору;
* Наличие корзины;
* Наличие фильтра по категориям;
* Обеспечение возможности оставлять отзывы;
* Обеспечение возможности изменять статус заказа администратору;
* Обеспечение возможности редактирования товаров администратору.

Главная цель данного курсового проектирования – это разработка программного средства, которое реализует все вышеперечисленные функции и решает поставленные задачи.

В пояснительной записке вы сможете найти краткую информацию о похожих продуктах, архитектуре, реализации проекта, руководстве пользователя.

Изменить задачи их в первый раздел

# 1 Постановка задачи

## 1.1 Аналитический обзор аналогов

Были проанализированы цель и задачи, поставленные в данном курсовом проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

# Zara

Одним из самых популярных альтернативных решений является интернет-ресурс «Zara», выступающая известным брендом молодёжной одежды, собравшим огромную клиентскую базу во всём мире. Главным преимуществом данного ресурса является дизайн, передающий эстетику бренда. Также у сайта широкий функционал, доступный любому пользователю: удобный поиск товара, наличие сортировки по категории, полу, размеру. Данное программное средство предоставляет пользователю возможность легко найти нужный товар, добавить его в корзину и совершить покупку.

Из выявленных недостатков данного ресурса необходимо выделить отсутствие возможности доставки по Беларуси. А также отсутствие адаптивного дизайна под все виды устройств.

Интерфейс интернет-ресурса представлен на рисунке 1.1.

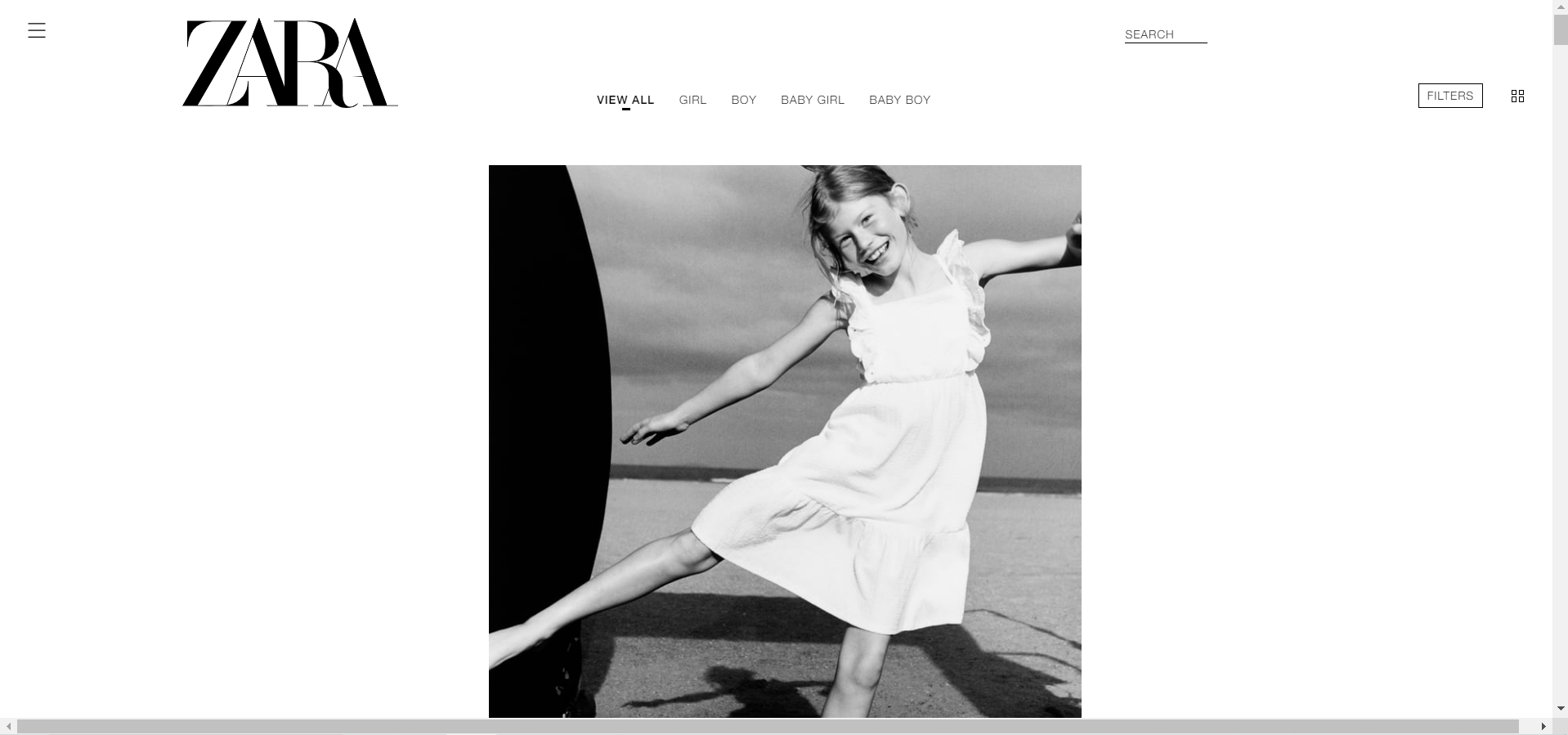


Рисунок 1.1 – Интернет-ресурс «Zara»

Из выявленных недостатков данного ресурса можно выделить отсутствие возможности зарезервировать товар на какое-то время, что гарантирует пользователю, что товар не исчезнет после попадания в корзину. Также на сайте нет возможности просмотреть популярные товары и количество продаж за определённые периоды.

# Bershka

Ещё одним альтернативным решением задач, поставленных в данном курсовом проекте, является интернет-ресурс «Bershka». С помощью данного программного средства пользователь может ознакомиться с каталогом данного бренда одежды, выбрать для себя подходящий по вкусу товар, а также оформить доставку, введя данные своей банковской карты.

Интерфейс интернет-ресурса представлен на рисунке 1.2.

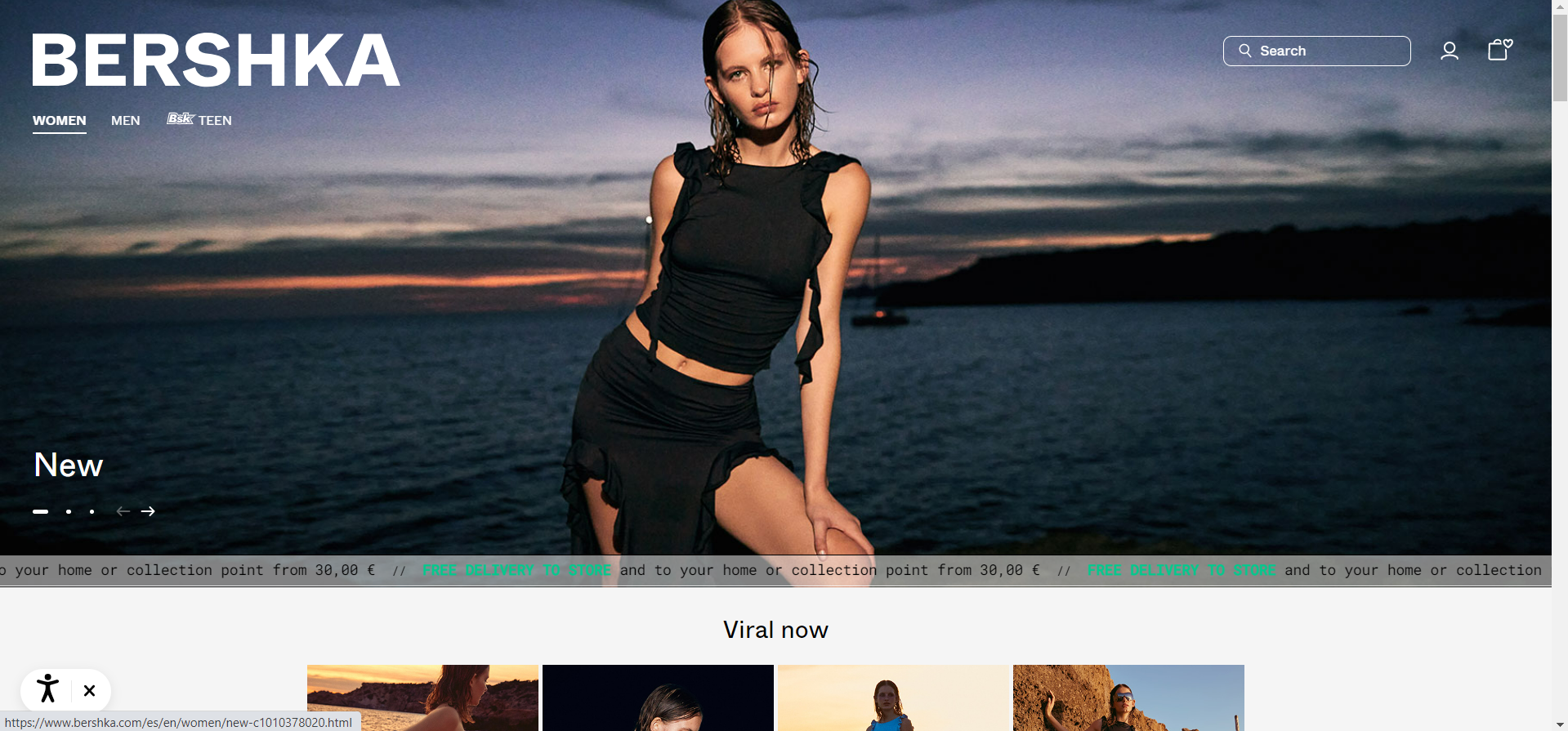


Рисунок 1.2 – Интернет-ресурс «Bershka»

Одним из главных недостатков такой альтернативы является отсутствие возможности посмотреть количество оставшихся на складе товаров, а также невозможность получить отзыв от других пользователей. В данном интернет-ресурсе тоже отсутствует возможность зарезервировать понравившийся товар на какое-то время.

# Pull & Bear

Ещё одной альтернативой решения поставленных задач является интернет-ресурс «Pull&Bear». С помощью данного программного средства пользователь может ознакомиться со списком доступных для покупки товаров, выбрать доступную для любого бюджета одежду. Немаловажным преимуществом является то, что пользователь может оценить образ на многочисленных фотографиях моделей. Также немаловажна возможность быть уведомлённым о заказе путём получения электронного письма. На сайте крайне удобное разделение на мужскую, женскую и детскую категории.

Интерфейс интернет-ресурса представлен на рисунке 1.3.

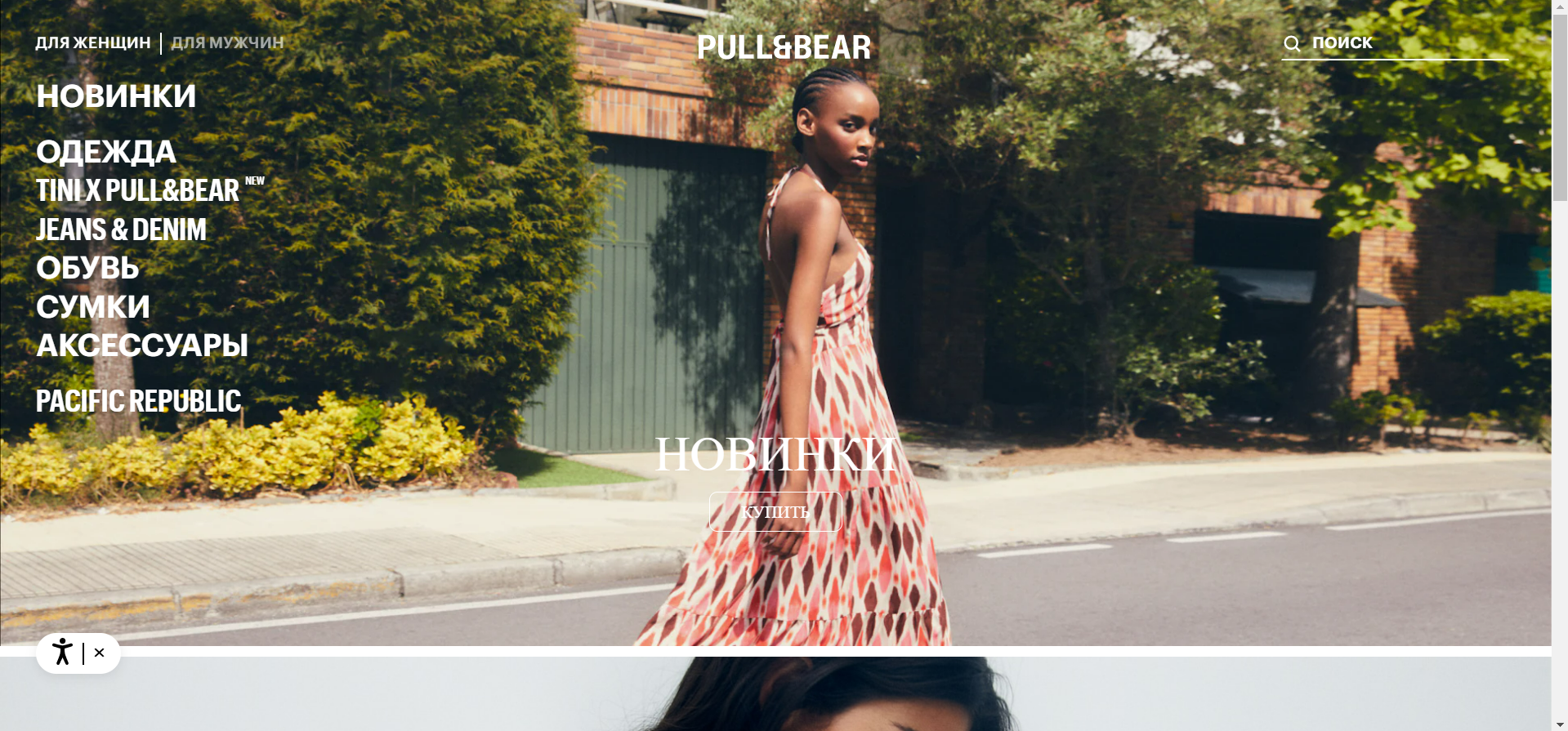


Рисунок 1.3 – Интернет-ресурс «Pull&Bear»

Однако, недостатком данной альтернативы являются отсутствие качественной оптимизации, из-за чего даже при наличии высокоскоростного интернета многие разделы долго загружаются. Также, как и у приведённых ранее примеров альтернатив, на сайте отсутствует возможность резервирования и просмотра популярных позиций.

# Lamoda

В список альтернатив, рассматриваемых для решения поставленных задач, также можно включить интернет-ресурс «Lamoda». Данное программное средство позволяет ознакомиться с предложениями от различных брендов в рамках одного маркетплейса. Приложение позволяет выбирать товар, сортируя не только по классическим категориям вроде пола и размера, но и по брендам и популярности, что является крайне удобным при поиске необходимого товара.

Интерфейс интернет-ресурса представлен на рисунке 1.4.

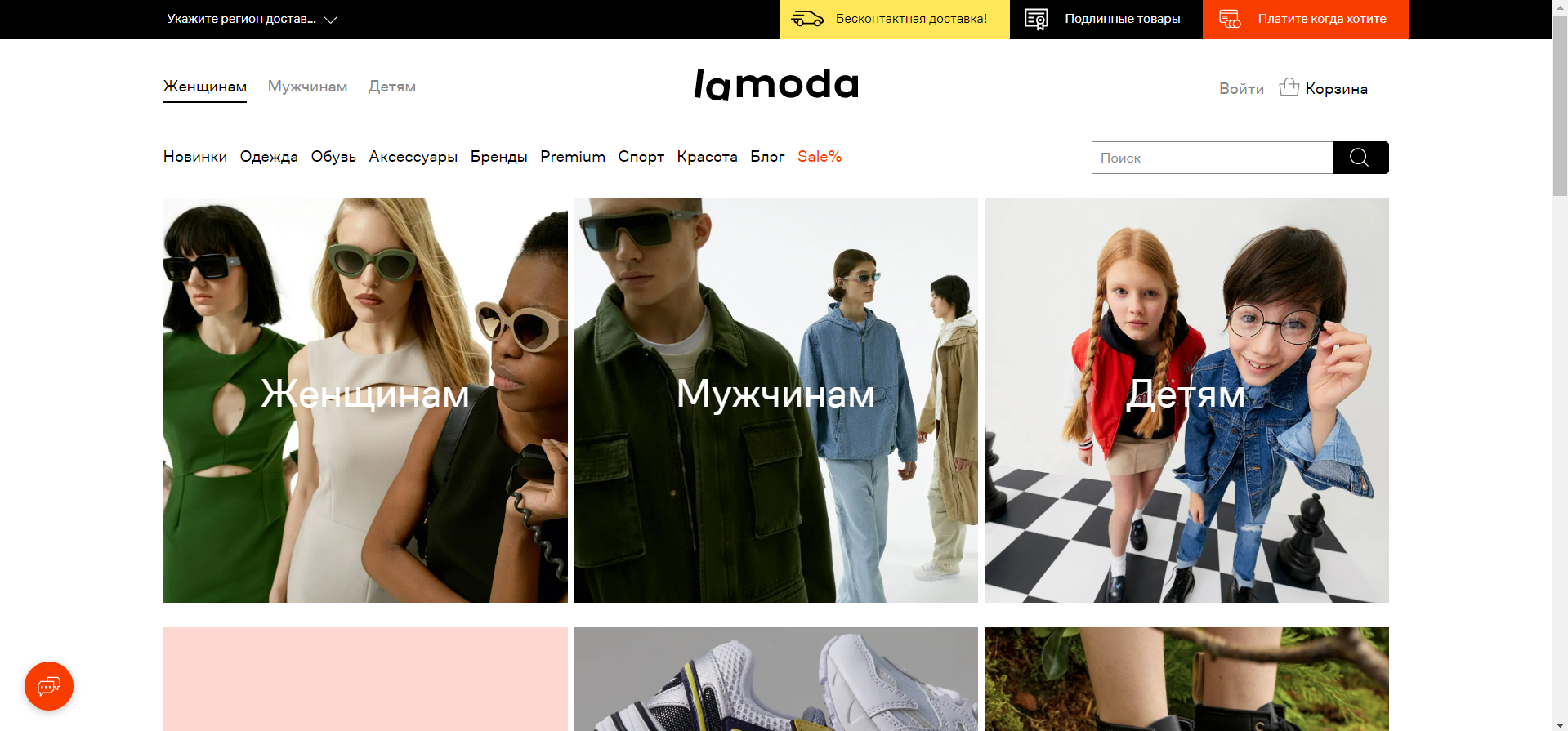


Рисунок 1.4 – Интернет-ресурс «Lamoda»

Недостатком данного программного средства является отсутствие возможности легко отменить свой заказ прямо с сайта без связи с продавцом и так же, как у конкурентов отсутствует возможность резервирования.

## 1.2 Разработка функциональных требований определение вариантов использования

Сюда можно из введения задачи

Функциональные требования приложения определяют, как оно должно обрабатывать данные и запросы пользователя и предоставлять ему необходимый пользовательский интерфейс. Это может включать в себя описание того, как данные должны храниться и организовываться, как происходит поиск и выборка данных, каким образом обновляются данные.

Для хранения данных приложение должно использовать базу данных. Страницы должны обновляться без перезагрузки. Вход в приложение должен осуществляться только после авторизации.

Помимо функциональных требований, важно также определить роли пользователей и их варианты использования системы. Это помогает определить, какие функции должны быть доступны для каждой роли, какие данные должны быть доступны для каждой роли.

Роли пользователя – это набор прав, которые пользователь может получить в системе. В зависимости от роли пользователя, он может иметь доступ к различным функциям системы. В данном проекте роли пользователей будут следующими:

* GUEST;
* USER;
* ADMIN;

На основе предоставленного списка ролей была построена uml-диаграмма вариантов использования. Диаграмма использования изображена на рисунке 1.5.

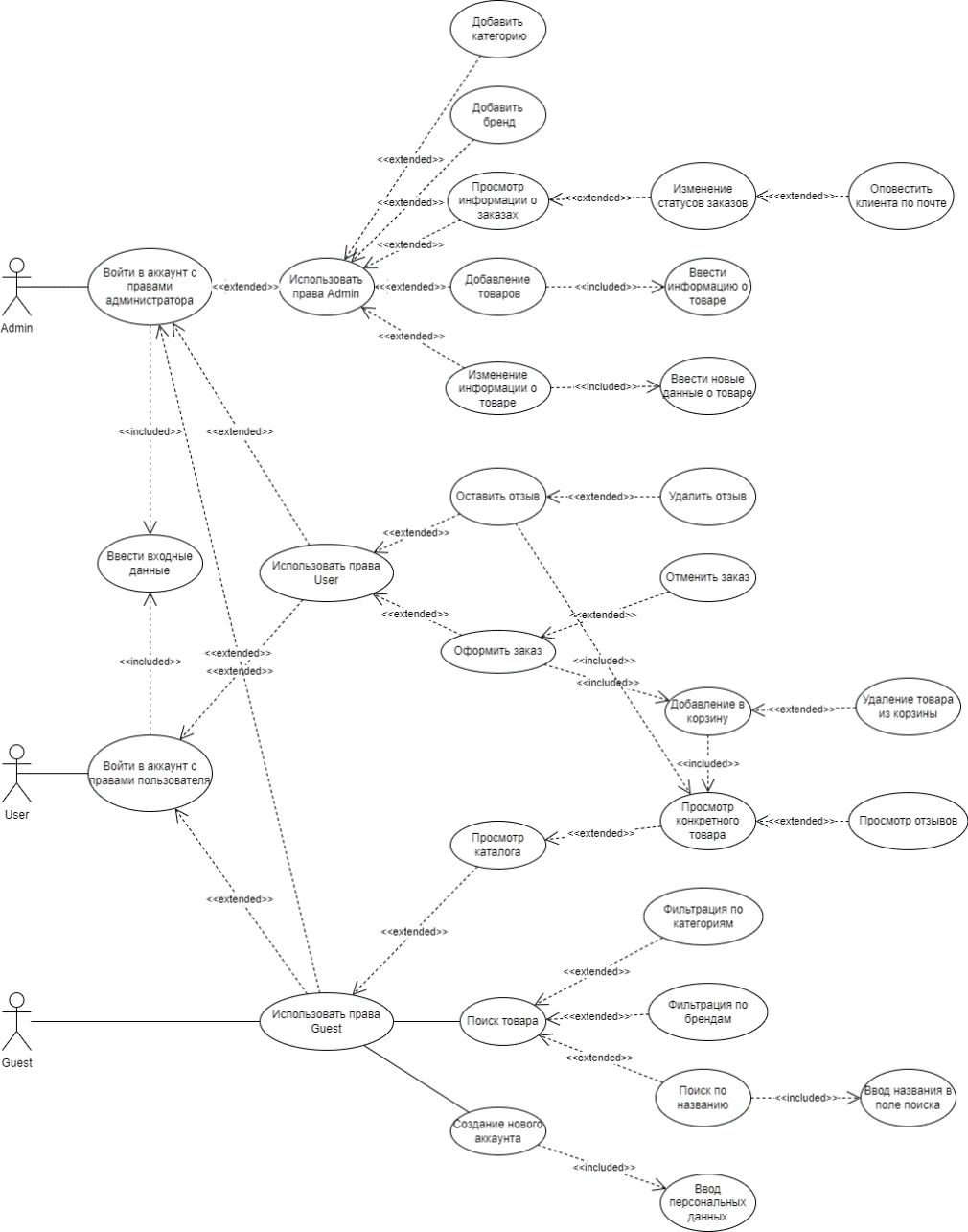


Рисунок 1.5 – Диаграмма вариантов использования попробовать изменить чтобы человечки сбоку были

Роль GUEST позволяет только просмотреть отзывы и каталог, а также создать новый аккаунт. После создания аккаунта пользователь получает роль USER.

Роль USER дает возможность просматривать каталог, оформлять заказы, добавлять товары в корзину.

Роль ADMIN также может просматривать информацию о товарах, помимо этого у него есть доступ к статистике продаж. Также он может добавлять новые позиции товара, изменять имеющиеся и изменять статусы заказов.

Приложение обладает тремя ролями, на которые разложены все основные функции для корректного использования.

Схему во второй раздел

## 1.3 Выводы по разделу

Итого, был проведен аналитический обзор аналогов, которые уже существуют на рынке. Этот обзор позволил определить основные характеристики и функциональные возможности, которые необходимо предусмотреть в разрабатываемой системе. Также были определены функциональные требования приложения, а также роли пользователей и варианты использования системы в зависимости от этих ролей.

# 2 Проектирование web-приложения

## 2.1 Архитектура приложения

Приложение построено на основе клиент-серверной архитектуры. Клиентская часть приложения разработана с использованием React, а серверная часть основана на Node.js и Express. Для хранения и управления данными используется база данных PostgreSQL.

Клиентскую часть называют frontend. Для реализации пользовательского интерфейса был выбран React. Он предоставляет эффективное модульное решение для создания пользовательского интерфейса. React-компоненты отвечают за отображение данных и взаимодействие с пользователем. Клиентская часть отправляет HTTP-запросы на сервер для получения данных и обновления информации.

Серверную часть называют backend. Для её реализации был выбран Node.js, он является средой выполнения JavaScript на стороне сервера, обеспечивает высокую производительность и масштабируемость. Express – это минималистичный и гибкий фреймворк для создания web-приложений на основе Node.js. Он облегчает разработку маршрутов, запросов и ответов, а также управление сеансами и промежуточным программным обеспечением. Сервер обрабатывает запросы от клиента, включая маршрутизацию запросов к соответствующим обработчикам.

Выбор СУБД остановился на PostgreSQL, она является реляционной базой данных с открытым исходным кодом, предоставляет надежное хранение и управление структурированными данными. Node.js сервер использует ORM Sequelize для CRUD операций.

Взаимодействие между клиентской и серверной частью приложения осуществляется следующим образом: клиент отправляет HTTP-запросы на сервер, используя AJAX. Сервер обрабатывает запросы, выполняет требуемые операции в базе данных и формирует HTTP-ответы. Ответы сервера в формате JSON. Клиентская часть обрабатывает ответы от сервера и обновляет пользовательский интерфейс соответствующим образом.

## 2.2 Общая структура приложения

В разработке.

## 2.3 Структура базы данных

Диаграмма базы данных таблиц (Database Table Diagram) – это визуальное представление структуры базы данных и отношений между таблицами, которые

хранятся в этой базе данных. Добавить все столбцы

Диаграмма показывает связи между таблицами и полями, а также отношения между ними, такие как связи "один-ко-одному", "один-ко-многим", "многие-ко-многим".

База данных состоит из 12 таблиц, связанных между собой внешними ключами.

Диаграмма базы данных представлена на рисунке 2.1.

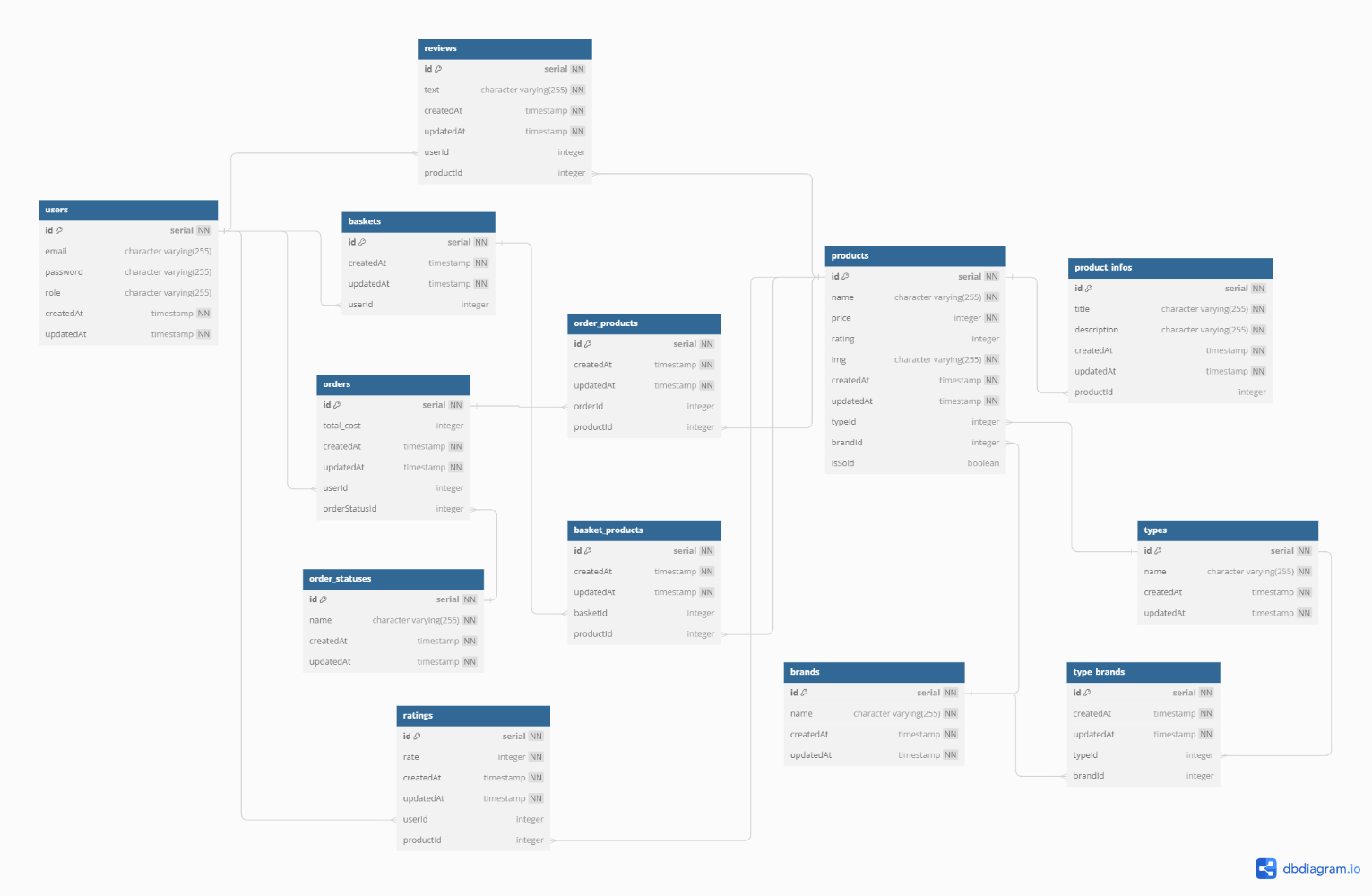


Рисунок 2.1 – Диаграмма базы данных

Таблица «USER» хранит информацию о пользователях, в ней хранится идентификатор пользователя, его email, пароль и роль.

Таблица «PRODUCT» хранит информацию о товарах, состоит из идентификатора, названия, цены, рейтинга, изображения, идентификатора типа и идентификатора бренда. Связана с таблицами «TYPE» и «BRAND» по столбцам typeId и brandId соответственно.

Таблица «PRODUCT\_INFO» хранит информацию о характеристиках товаров, состоит из идентификатора, идентификатора товара, названия и описания. Связана внешним ключом с таблицей «PRODUCT» по столбцу product\_id.

Таблица «REVIEW» хранит информацию об отзывах, состоит из идентификатора, идентификатора пользователя, идентификатора товара и текста. Связана с таблицами «USER» и «PRODUCT» по столбцам user\_id и product\_id соответственно.

Таблица «BASKET» хранит информацию о корзине, состоит из идентификатора и идентификатора пользователя. Связана с таблицей «USER» по столбцу user\_id.

Таблица «BASKET\_PRODUCT» хранит информацию о наполнении корзины, состоит из идентификатора, идентификатора корзины и идентификатора товара. Связана с таблицами «BASKET» и «PRODUCT» по столбцам basket\_id и product\_id соответственно.

Таблица «ORDER» хранит информацию о заказах, состоит из идентификатора, идентификатора пользователя и идентификатора статуса заказа. Связана с таблицей «USER» и «ORDER\_STATUS» по столбцам user\_id и status\_id соответственно.

Таблица «ORDER\_PRODUCT» хранит информацию о составе заказа, состоит из идентификатора, идентификатора заказа, идентификатора товара и количестве. Связана с таблицами «ORDER» и «PRODUCT» по столбцам order\_id и product\_id соответственно.

Таблица «ORDER\_STATUS» хранит информацию о статусах заказа, состоит из идентификатора и названия статуса.

Таблица «RATING» хранит информацию о рейтинге товара, состоит из идентификатора, идентификатора пользователя, идентификатора товара и оценки. Связана с таблицами «USER» и «PRODUCT» по столбцам user\_id и product\_id соответственно.

Таблица «TYPE» хранит информацию о категориях товара, состоит из идентификатора и названия типа.

Таблица «BRAND» хранит информацию о брендах, состоит из идентификатора и названия бренда.

## 2.4 Вывод

В этой главе была рассмотрена архитектура приложения, основанная на клиент-серверной модели. Описаны основные компоненты: клиентская часть, реализованная с использованием React, и серверная часть, основанная на Node.js и Express. Также было упомянуто, что для хранения и управления данными применяется база данных PostgreSQL. Описано взаимодействие между клиентской и серверной частью через HTTP-запросы. Была представлена общая структура приложения, а также детально описана структура базы данных.