*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка программной системы   
информационной системы «Интернет-магазин»*

*Выполнил: студент гр. ПРИ-115*

*Обрубов М.О.*

*Принял: доц. кафедры ИСПИ*

*Вершинин В.В.*

*Владимир, 2017*

**Аннотация**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc497417741)

[1 ПОСТАНОВКА ЗАДАЧИ 5](#_Toc497417742)

[2 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc497417743)

[2.1. Описание предметной области 6](#_Toc497417744)

[2.2. Словарь предметной области 7](#_Toc497417745)

[3 МОДЕЛИРОВАНИЕ СТРУКТУРЫ ОБЪЕКТОВ ПРЕДМЕТНОЙ ОБЛАСТИ И ИХ ВЗАИМОДЕЙСТВИЯ НА КОНЦЕПТУАЛЬНОМ УРОВНЕ 10](#_Toc497417746)

[3.1. Перечень ролей программной системы 10](#_Toc497417748)

[3.2. Диаграмма прецедентов 10](#_Toc497417749)

[3.3. Спецификация прецедентов 11](#_Toc497417750)

[4 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНОЙ СИСТЕМЫ 15](#_Toc497417751)

[4.1. Диаграмма классов 15](#_Toc497417753)

[4.2. Диаграмма состояний 20](#_Toc497417754)

[4.3. Диаграмма последовательностей 21](#_Toc497417755)

[5 ТЕСТИРОВАНИЕ СИСТЕМЫ 23](#_Toc497417756)

[ЗАКЛЮЧЕНИЕ 24](#_Toc497417757)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc497417758)

[ПРИЛОЖЕНИЕ А. 26](#_Toc497417759)

# ВВЕДЕНИЕ

В 21 веке любой бизнес-процесс нуждается в информатизации. Без внедрения информационных технологий практически каждый процесс будет отставать от своих конкурентов.

В последние года значительные обороты набирают веб-ресурсы. Они позволяют вести бизнес по всему земному шару, так как доступ в интернет есть практически везде. Веб-сайты могут быть созданы либо для удовлетворения потребностей людей, путем предложения им товаров или услуг, либо для автоматизации деятельности каких-либо компаний.

Темой данного курсового проектирования является разработка программной системы для осуществления своей деятельности интернет-магазином. В ходе выполнения курсового проектирования будет спроектировано и реализовано веб-приложение, позволяющее удобно и с меньшими затратами управлять данным бизнесом.

Работа интернет-магазина заключается в взаимодействии с клиентами и предоставлении им товаров или услуг. Помимо этого, в работу интернет-магазина входит большой объем работы в сфере рекламы, маркетинга, удержания и повторного привлечения клиента, но, так как курсовое проектирование рассчитано только на разработку программной системы, то эти темы не могут быть затронуты.

Так как интернет-магазины давно существуют в практически любой сфере продаж, необходимо выбрать одну конкретную сферу. В данном курсовой проектировании я решил выбрать сферу продажи детской одежды. Таким образом, далее сущность интернет-магазина будет рассмотрена именно в этой специфике.

# ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программную систему для информационной системы «Интернет-магазин», для этого:

* выполнить исследование и анализ предметной области;
* разработать прототип программной системы;
* выполнить моделирование работы программной системы;
* разработать схему;
* реализовать программную систему с использованием выбранных средств и технологий.

Для выполнения работы необходимо использовать:

* методология проектирования и разработки – RUP;
* язык моделирования – UML;
* платформа разработки – ASP.NET.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Описание предметной области

Интернет-магазин – это предприятие по продаже различного рода товаров клиентам с помощью Интернета. Взаимодействие клиента и продавца осуществляется удаленно. Клиент сам выбирает себе необходимые товары и оплачивает их. Продавец же принимает заказ, формирует его и отправляет покупателю любым доступным способом. Далее можно рассмотреть подробнее алгоритм работы интернет магазина и взаимодействия его с клиентом.

Когда клиент попадает на страницу с перечнем товаров, например, «Каталог товаров», он видит все доступные для покупки товары. Они представлены в виде миниатюр (либо списком, либо плитками по несколько в строке). Клиент может выбрать любой понравившийся ему товар и попасть на страницу с его подробным описанием.

На странице с описанием товара может быть размещена любая информация о нём, например, фото товара, его полное описание, размер, цвет, наличие на складе магазина и многое другое. Если товар удовлетворяет клиента, то он выбирает необходимые ему характеристики и нажимает на кнопку «Добавить товар в корзину» и товар добавляется в корзину клиента. Далее таким же образом клиент добавляет другие товары.

Когда клиент «собрал» в корзину все необходимые товары, он приступает к оформлению заказа. Сделать он это может перейдя на страницу с перечнем товаров, добавленных в корзину и нажав кнопку «Оформить заказ». Следует учесть, что при оформлении заказа требуется, чтобы клиент был зарегистрирован и авторизован в системе. После нажатия на кнопку «Оформить заказ», клиент попадает на страницу с данными заказа, там он проверяет существующие данные и добавляет отсутствующие, необходимые для успешного оформления заказа.

После того, как клиент подтверждает введенные данные, заказ отправляется на проверку и подтверждение менеджером интернет-магазина. Менеджер проверяет данные заказа и любым доступным способом уточняет заказ у клиента. Как только менеджер подтверждает заказ, клиент либо оплачивает заказ онлайн, либо ожидает его и оплачивает сотруднику службы доставки. Сам заказ переходит в состояние доставки.

Помимо работы с клиентом, интернет-магазин нужно регулировать изнутри. Этим занимается руководитель или администратор интернет-магазина. Он может:

* управлять товарами, а именно добавлять, изменять или удалять их;
* управлять пользователями, а именно регистрировать новых пользователей, редактировать данные уже существующих или удалять их, а также изменять их роли в системе;
* управлять заказами, а именно изменять или удалять их, если вдруг возникнут ошибки в работе системы.

В рамках данного курсового проектирования будет рассмотрен только тот функциональный минимум, который необходим для работы интернет-магазина.

* 1. Словарь предметной области

Интернет-магазин – предприятие, осуществляющее продажу товаров покупателям в онлайн режиме.

Гость – пользователь, не зарегистрированный или не авторизованный в системе. Гость может авторизоваться или зарегистрироваться и получить дополнительные функции, доступные зарегистрированным пользователям. Гость имеет доступ к списку товаров и может просматривать подробную информацию о них, без возможности дальнейшего взаимодействия (добавления в корзину).

Зарегистрированный пользователь – пользователь, зарегистрированный в системе. Имеет следующие характеристики: логин и пароль для аутентификации и авторизации, роль в системе, адрес электронной почты, номер телефона, фамилию, имя, отчество, а также ряд характеристик, необходимых для функционирования системы. Предполагается, что зарегистрированный пользователь уже авторизован в системе, и функция авторизации ему недоступна.

Покупатель – зарегистрированный пользователь, который может совершать покупки в интернет-магазине. Покупатель имеет адрес доставки и ряд характеристик, необходимых для функционирования системы. Покупателю доступны функции гостя (за исключением функции регистрации и авторизации). Кроме этого покупатель может редактировать личную информацию, оформлять, оплачивать, а также просматривать все свои заказы.

Сотрудник – зарегистрированный пользователь, который обеспечивает функционирование интернет-магазина. Помимо характеристик зарегистрированного пользователя сотрудник имеет: порядковый номер сотрудника и информацию о заработной плате.

Менеджер – сотрудник интернет-магазина, взаимодействующий с покупателями (подтверждает и отправляет заказы). Менеджер имеет ряд функций: просматривать список доступных ему заказов, изменять статус заказа (например, переводить заказ из статуса «Созданный» в статус «Подтвержденный», т.е. подтверждать заказ).

Администратор – сотрудник интернет-магазина, осуществляющий управление основными сущностями интернет-магазина: пользователями, заказами, товарами и т.д.

Корзина – множество товаров, которые покупатель добавил для последующего оформления заказа.

Заказ – сущность, содержащая в себе список товаров, которые заказал (и, возможно, оплатил) покупатель. Помимо списка товаров заказ включает в себя статус (состояние заказа), информацию о покупателе, который сделал заказ и о менеджере, который сопровождает этот заказ.

Товар – вещь, которая покупатель может купить в интернет-магазине. Товар имеет следующие характеристики: наименование, цена, и др.

Оплата – процесс передачи денежных средств за приобретенный товар интернет-магазину.

Наличная оплата – оплата, которая производится при получении заказа.

Безналичная оплата – оплата, которая производится в онлайн режиме одним из доступных способов безналичного расчета при участии внешних платежных сервисов.

# МОДЕЛИРОВАНИЕ СТРУКТУРЫ ОБЪЕКТОВ ПРЕДМЕТНОЙ ОБЛАСТИ И ИХ ВЗАИМОДЕЙСТВИЯ НА КОНЦЕПТУАЛЬНОМ УРОВНЕ

1. 1. Перечень ролей программной системы

Роли проектируемой программной системы перечислены в следующем списке:

1. Гость.
2. Покупатель.
3. Менеджер.
4. Администратор.

Подробное описание каждой роли можно найти в словаре предметной области, а их функции на диаграмме прецедентов.

* 1. Диаграмма прецедентов

В ходе анализа предметной области, изучения структуры и функций проектируемой программной системы была разработана схема функций программной системы, т.н. диаграмма прецедентов. На ней изображены действующие лица (актеры), взаимодействующие с системой, а также функции системы (прецеденты), которые они могут выполнять. Диаграмма прецедентов представлена на рис. 1.

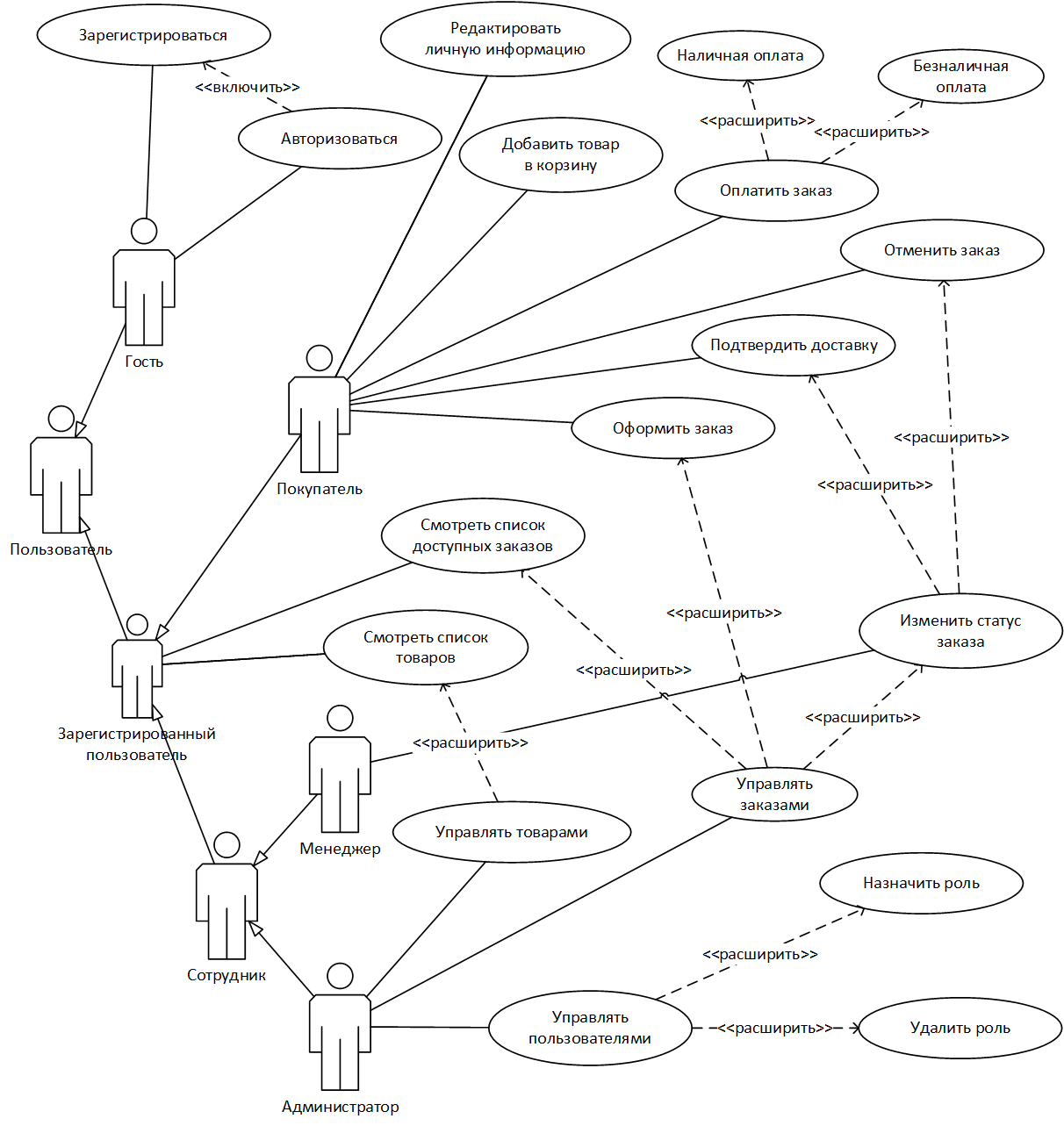


Рисунок 1. Диаграмма прецедентов.

* 1. Спецификация прецедентов

Ниже представлена словесная спецификация нескольких прецедентов программной системы: «Добавить товар в корзину», «Оформить заказ», «Подтвердить заказ» и «Назначить роль».

* + 1. ****Спецификация прецедента «Добавить товар в корзину»****

**Название:** Добавить товар в корзину.

**Краткое описание:** Действующее лицо добавляет выбранный товар в корзину.

**Действующее лицо:** Покупатель.

**Предусловие:** Действующее лицо находится на странице товара, который он хочет добавить в корзину.

**Основной поток:**

1. Действующее лицо нажимает на кнопку «Добавить в корзину».
2. Система сохраняет товар во внутреннем представлении объекта «Корзина».

**Постусловие:** Товар добавлен в корзину.

**Альтернативный поток:** Нет.

* + 1. ****Спецификация прецедента «Оформить заказ»****

**Название:** Оформить заказ.

**Краткое описание:** Действующее лицо оформляет заказ из товаров, находящихся в корзине.

**Действующее лицо:** Покупатель.

**Предусловие:** Действующее лицо находится на странице с корзиной товаров.

**Основной поток:**

1. Действующее лицо нажимает на кнопку «Оформить заказ».
2. Система перенаправляет действующее лицо на страницу оформления заказа.
3. Действующее лицо заполняет все необходимые для оформления заказа поля.
4. Действующее лицо нажимает кнопку «Готово».
5. Система сохраняет заказ в БД со статусом «Создан».

**Постусловие:** Заказ создан и сохранен в системе.

**Альтернативный поток 1:** *не заполнены обязательные поля*

1. На шаге 4 не заполнено хотя бы одно обязательное поле.
2. Система отображает сообщение действующему лицу о том, что не заполнены обязательные поля.
   * 1. Спецификация прецедента «Подтвердить заказ».

**Название:** Подтвердить заказ.

**Краткое описание:** Действующее лицо подтверждает созданный покупателем заказ, и он становится доступным для исполнения.

**Действующее лицо:** Менеджер.

**Предусловие:** Действующее лицо находится на странице с заказами, которые создали покупатели.

**Основной поток:**

1. Действующее лицо нажимает кнопку «Подтвердить заказ», находящуюся в строке необходимого заказа.
2. Система перенаправляет действующее лицо на страницу подтверждения товара.
3. Действующее лицо проверяет детали заказа.
4. Действующее лицо нажимает кнопку «Подтвердить».
5. Система переводит заказ в статус «Подтвержден» и обновляет его в БД. **Постусловие:** Заказ подтвержден.

**Альтернативный поток:** *заказ не верный*

1. На шаге 3 действующее лицо нажимает кнопку «Отклонить».
2. Система переводит заказ в статус «Отклонен».
   * 1. Спецификация прецедента «Назначить роль».

**Название:** Назначить роль.

**Краткое описание:** Действующее лицо назначает роль пользователю, зарегистрированному в системе.

**Действующее лицо:** Администратор.

**Предусловие:** Действующее лицо находится на странице с пользователями системы.

**Основной поток:**

1. Действующее лицо нажимает на кнопку «Изменить роль» в строке выбранного пользователя.
2. Система перенаправляет действующее лицо на страницу с ролями пользователя.
3. Действующее лицо нажимает на кнопку «Добавить роль».
4. Действующее лицо выбирает роль, которую необходимо назначить пользователю.
5. Действующее лицо нажимает кнопку «Добавить».
6. Система сохраняет изменения в БД.

**Постусловие:** Пользователю назначена роль.

**Альтернативный поток:** *нет.*

# ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНОЙ СИСТЕМЫ

1. 1. Диаграмма классов

В результате проведения анализа предметной области на основе прецедентной модели были выделены основные сущности предметной области, их атрибуты и поведение, определены взаимосвязи между сущностями и разработана диаграмма классов предметной области (рис. 2).

Ниже приведено словесное описание сущностей системы, их атрибутов и методов.

1. RegisteredUser (Зарегистрированный пользователь)

*Атрибуты:*

* Login – логин пользователя для авторизации в системе;
* Password – пароль для авторизации в системе;
* Name – имя пользователя;
* Surname – фамилия пользователя;
* Birthdate – дата рождения;
* Phone – контактный телефон пользователя;
* Email – адрес электронной почты.

1. Buyer (Покупатель)

*Атрибуты:*

* Goods – корзина;
* Address – адрес доставки.

*Методы:*

* AddToCart – добавить товар в корзину;
* Checkout – оформить заказ;
* PayOnline – оплатить онлайн;
* CancelOrder – отменить заказ;
* ConfirmDelivering – подтвердить доставку заказа;
* EditPersonalInfo – изменить личную информацию.

1. Employee (Сотрудник)

*Атрибуты:*

* MonthSalary – зарплата за месяц.

1. Manager (Менеджер)

*Методы:*

* ConfirmOrder – подтвердить заказ;
* CancelOrder – отменить заказ.

1. Admin (Администратор)

*Методы:*

* AddUser – добавить пользователя;
* MofifyUser – изменить данные пользователя;
* DeleteUser – удалить пользователя;
* ChangeRole – изменить роль пользователя;
* AddGoods – добавить товар;
* ModifyGoods – изменить данные товара;
* DeleteGoods – удалить товар;
* DeleteOrder – удалить заказ.

1. Address (Адрес)

*Атрибуты:*

* Postcode – почтовый индекс;
* Country – страна;
* Region – область;
* City – город;
* Street – улица;
* House – дом;
* Flat – квартира.

1. Order (Заказ)

*Атрибуты:*

* Status – статус заказа;
* Buyer – покупатель;
* Manager – менеджер, ведущий заказ;
* Goods – корзина с товарами;
* DeliveryAddress – адрес доставки;
* PaymentType – тип оплаты;
* IsPayed – оплачен ли заказ;
* Cost – сумма заказа.

1. ShoppingCart (Корзина)

*Атрибуты:*

* Goods – список товаров в корзине.

1. Goods (Товар)

*Атрибуты:*

* Name – наименование;
* Price – цена.

1. OrderStatus (Статус заказа – перечисление)

* Created – создан;
* Confirmed – подтвержден;
* Payed – оплачен;
* Delivering – доставляется;
* Delivered – доставлен;
* Cancelled – отменен.

1. PaymentType (Тип оплаты – перечисление)

* Cash – наличная оплата;
* Cashless – безналичная оплата.

Так как на данном этапе проектировалась диаграмма классов концептуального уровня, то классы, необходимые для функционирования системы, например, классы доступа к данным, на диаграмме классов представлены не будут.

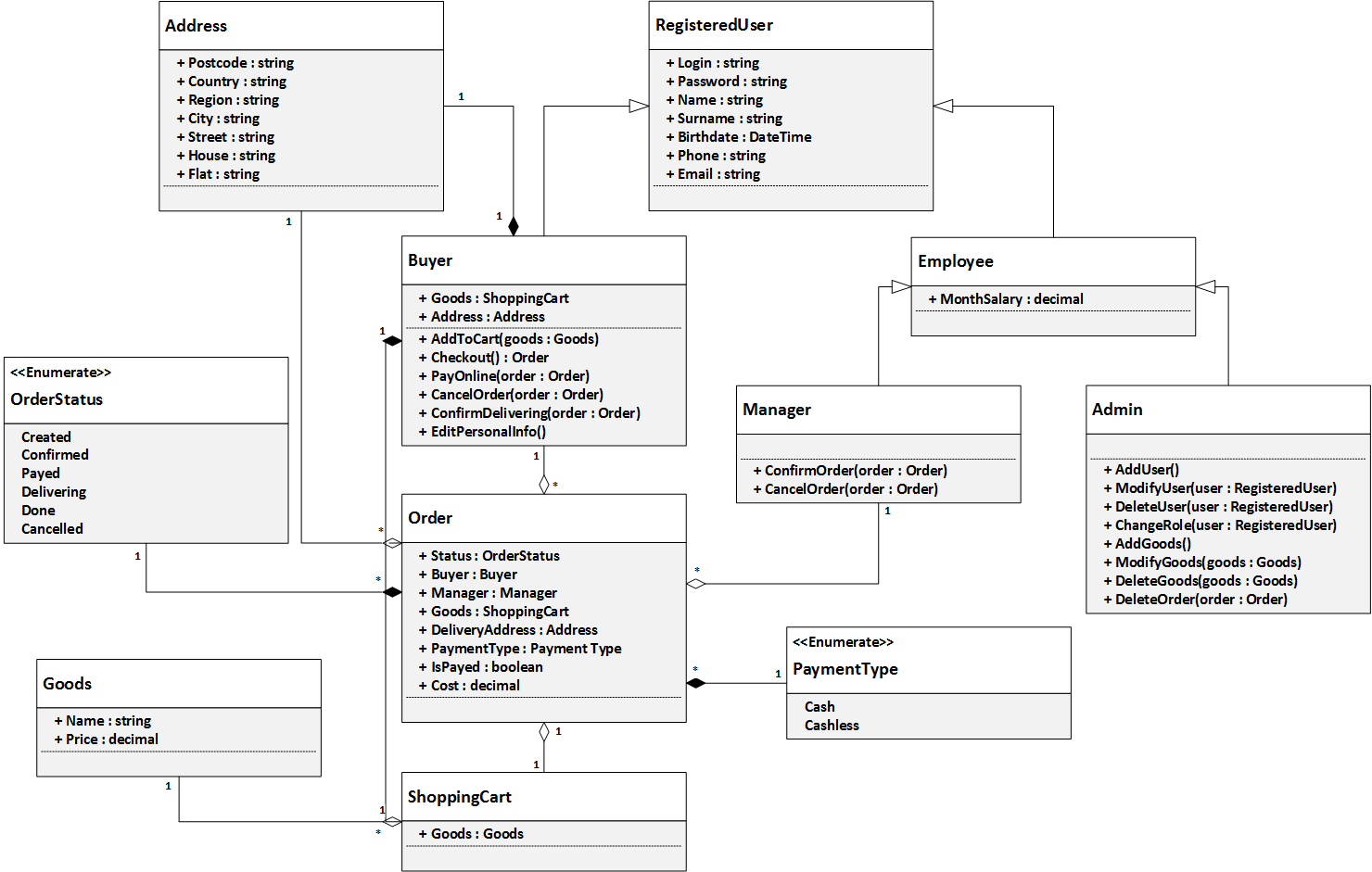


Рисунок 2. Диаграмма классов.

* 1. Диаграмма состояний

Для составления диаграммы состояний был выбран объект «Заказ», так как он является основным объектом бизнес-процессов интернет-магазина и может находиться во многих состояниях.

Ниже представлено словесное описание изменения состояния объекта «Заказ».

1. Покупатель создает заказ, указав в нём все необходимые параметры. Объект «Заказ» создан и сохранен во внутреннем представлении в системе со статусом «Создан».
2. Менеджер интернет-магазина проверяет параметры заказа на корректность (возможность доставки по указанному адресу, наличие товаров на складе и т.д.) и переводит заказ в состояние «Подтвержден», если заказ корректен.
3. В заказе имеются ошибки. Переход к п. 7.
4. Покупатель оплачивает заказ, и система переводит заказ в состояние «Оплачен».
5. Менеджер видит, что заказ оплачен, отправляет покупателю заказ и переводит заказ в статус «Доставляется».
6. Покупатель получает заказ и переводит заказ в статус «Завершен».
7. Заказ переходит в статус «Отменен».

Разработанная диаграмма состояний представлена на рис. 3.

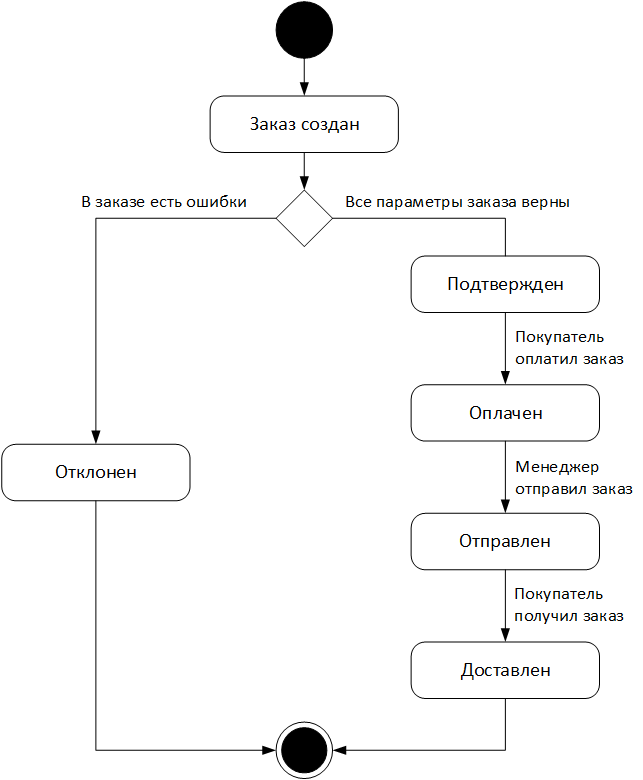


Рисунок 3. Диаграмма состояний.

* 1. Диаграмма последовательностей

Диаграммой последовательностей был описан вариант использования «Оплатить заказ».

На диаграмме: Покупатель – действующее лицо, OrderController – класс, реализующий работу с заказами, OrderDAO – класс, реализующий взаимодействие с базой данных, в частности с сущностью «Заказ», PayController – класс, реализующий работу с внешней платежной системой, Платежная система – внешняя платежная система, с помощью которой осуществляется оплата заказов в интернет-магазине.

Диаграмма последовательностей представлена на рис. 4.

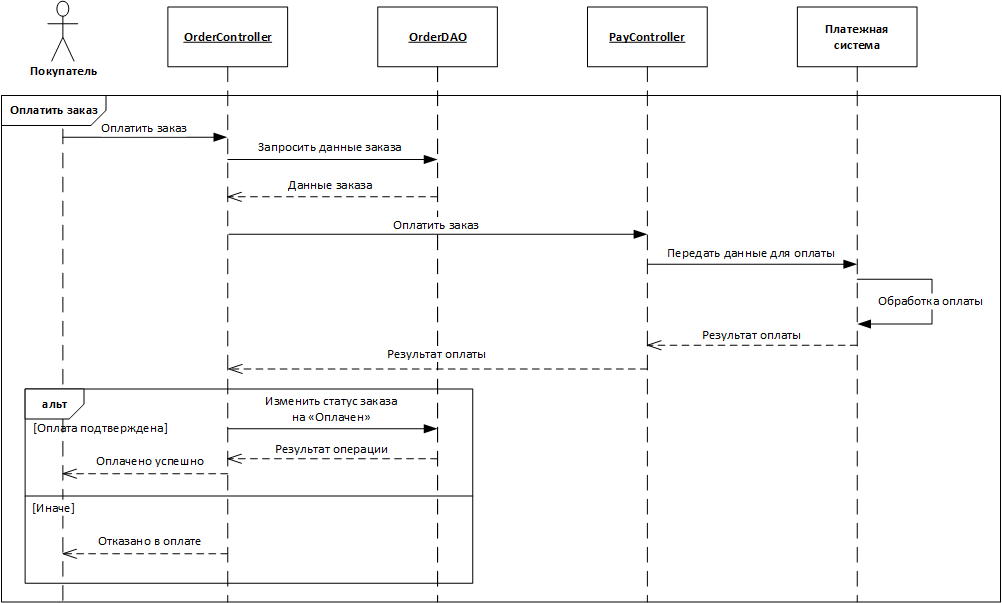


Рисунок 4. Диаграмма последовательностей.

# РЕАЛИЗАЦИЯ ПРОТОТИПА ПРОГРАММНОЙ СИСТЕМЫ

* 1. Структура проекта

Для реализации программной системы был выбран шаблон MVC, так как он является наиболее удобным для реализации объектно-ориентированных программных систем.

Шаблон проектирования MVC предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: Модель, Представление и Контроллер – таким образом, что модификация каждого компонента может осуществляться независимо.

Модель – предоставляет данные и реагирует на команды контроллера, изменяя свое состояние.

Представление – отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер – интерпретирует действия пользователя, оповещая модель о необходимости изменений. []

Для взаимодействия с подсистемой хранения данных реализован слой доступа к данным – классы DAO (Data Access Object) для основных сущностей. Работа с данными осуществляется с помощью фреймворка ADO.NET Entity Framework, а запросы к БД строятся с помощью технологии LINQ to Entities.

Исходный код проекта веб-приложения представлен в приложении А

* 1. Подсистема хранения данных

При проектировании модели данных было принято решение использовать технологию ADO.NET Entity Framework 6, а в частности подход Database First. Суть этой технологии заключается в том, что сначала проектируется и реализуется структура базы данных, а потом по ней автоматически генерируются классы сущностей, т.н. value-objects.

Для хранения аутентификационных данных пользователей использовалась стандартная база данных и библиотека классов – ASP.NET Identity. К уже существующим сущностям были добавлены сущность предметной области. Диаграмма сущностей базы данных представлена на рис. 5.

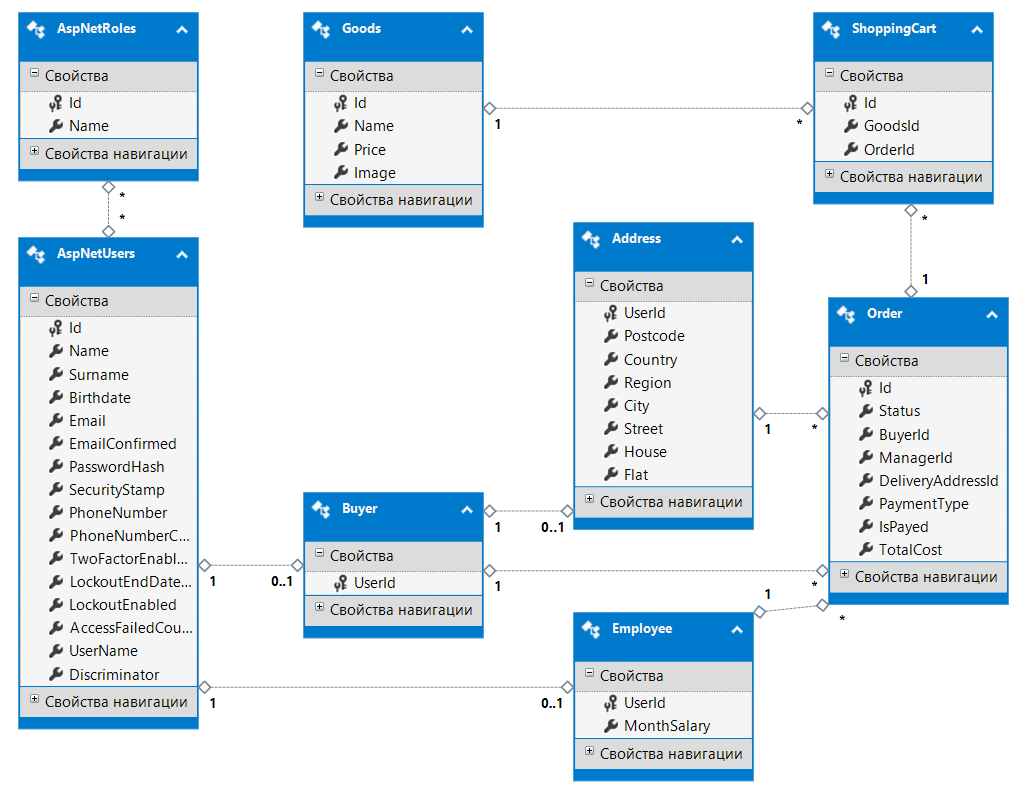


Рисунок 5. Диаграмма сущностей БД.

* 1. Ролевая модель

Контроль за доступом к ресурсам приложения на основании роли авторизованного пользователя был выполнен на нескольких уровнях:

* на уровне карты сайта (доступ к основным разделам сайта ограничен в файле конфигурации карты сайта Mvc.sitemap);
* на уровне контроллеров (с помощью аннотаций [Authorized]);
* на уровне представлений (на основании возвращаемого значения метода User.IsInRole() отображаются те или иные элементы пользовательского интерфейса).
  1. Внешний сервис

Так как интернет-магазину необходим инструмент онлайн-оплаты заказов, то веб-приложению необходима внешняя платежная система. Ввиду того, что в рамках курсового проектирования разрабатывается прототип программной системы, то для имитации работы внешней платежной системы было решено реализовать собственный веб-сервис, работающий по протоколу SOAP (Simple Object Access Protocol).

В подсистеме хранения данных веб-сервиса хранятся данные о счетах клиентов (данные банковских карт и сумма денег на счету). Запросы к базе данных реализованы в классе DAO с помощью ADO.NET Entity Framework и LINQ to Entities.

Веб-сервис предоставляет единственный метод, который имитирует онлайн-оплату по данным банковской карты и сумме, которую необходимо списать. В данном методе проверяются данные банковской карты и в случае успешной проверки с баланса пользователя списывается заданная сумма.

Исходный код веб-сервиса полностью представлен в приложении В.

* 1. Логгирование

В процессе работы любого приложения возникают ошибки. Для того, чтобы своевременно находить причины ошибок и устранять их, нужно иметь информацию о том, как ведет себя приложение в процессе своей работы. Для этого используется логгирование.

Логгирование – это процесс записи информации о происходящих с каким-то объектом (или в рамках какого-то процесса) событиях в журнал (лог-файл). []

В ходе разработки веб-приложения в рамках курсового проектирования применялась сторонняя библиотека NLog.

* 1. Контроль версий

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. СКВ даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое. []

В ходе разработки программной системы использовалась система контроля версий Git и в частности сервис GitHub.

В ходе работы в общей сложности было сделано 60 коммитов.

* 1. Пользовательский интерфейс

Скриншоты пользовательского интерфейса представлены на рис. 6 – 16.

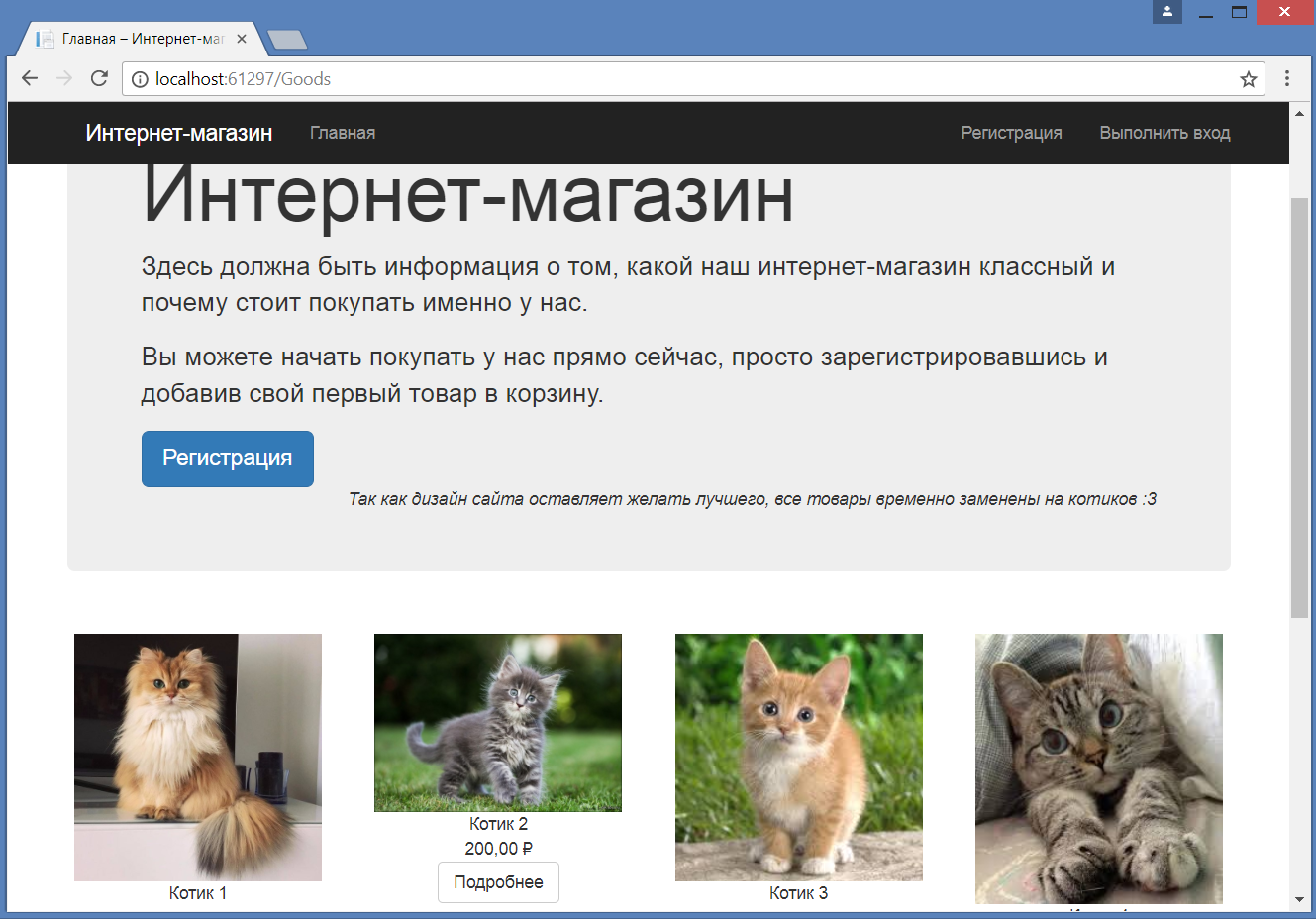


Рисунок 6. Главная страница.

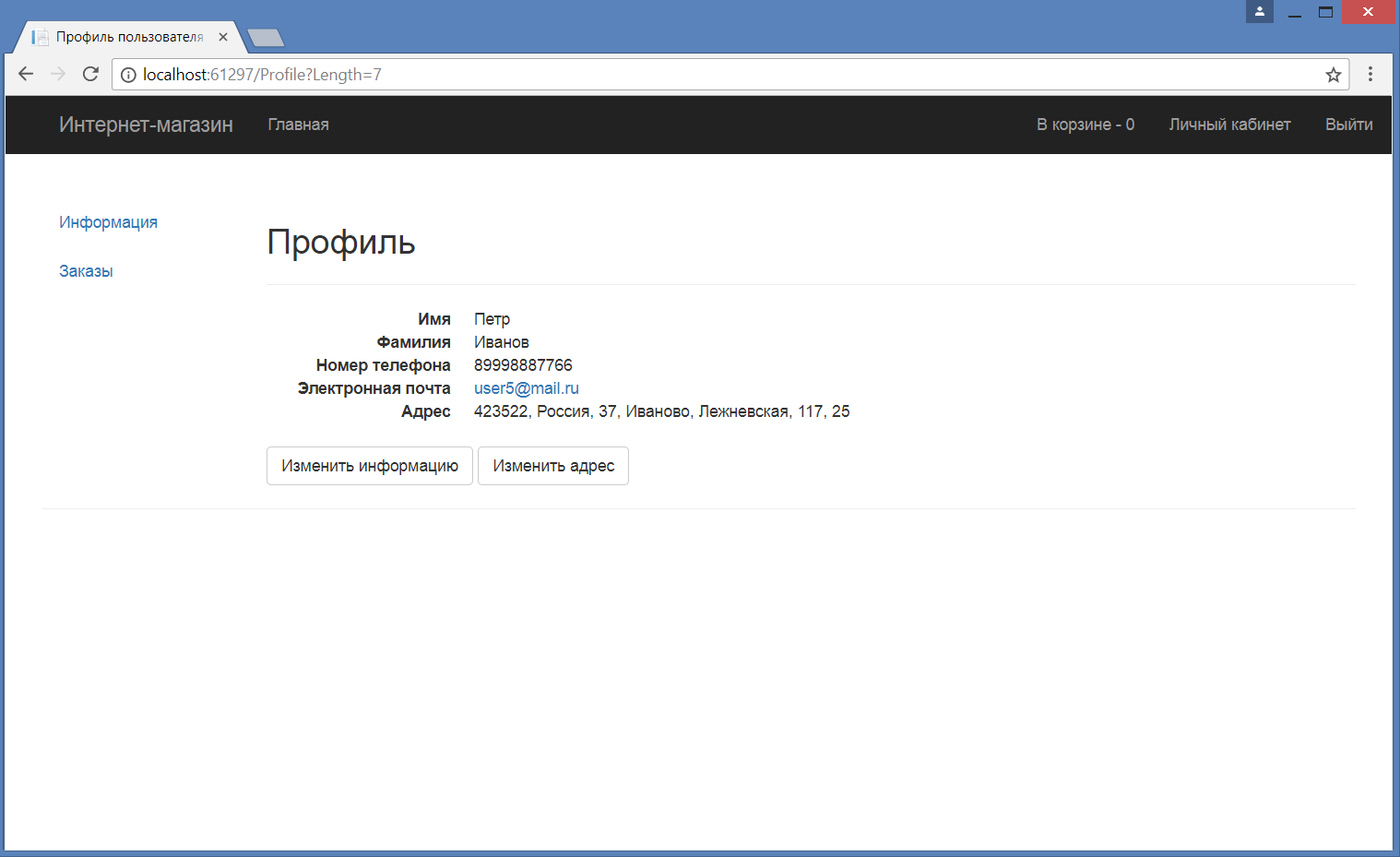


Рисунок 7. Страница профиля пользователя.

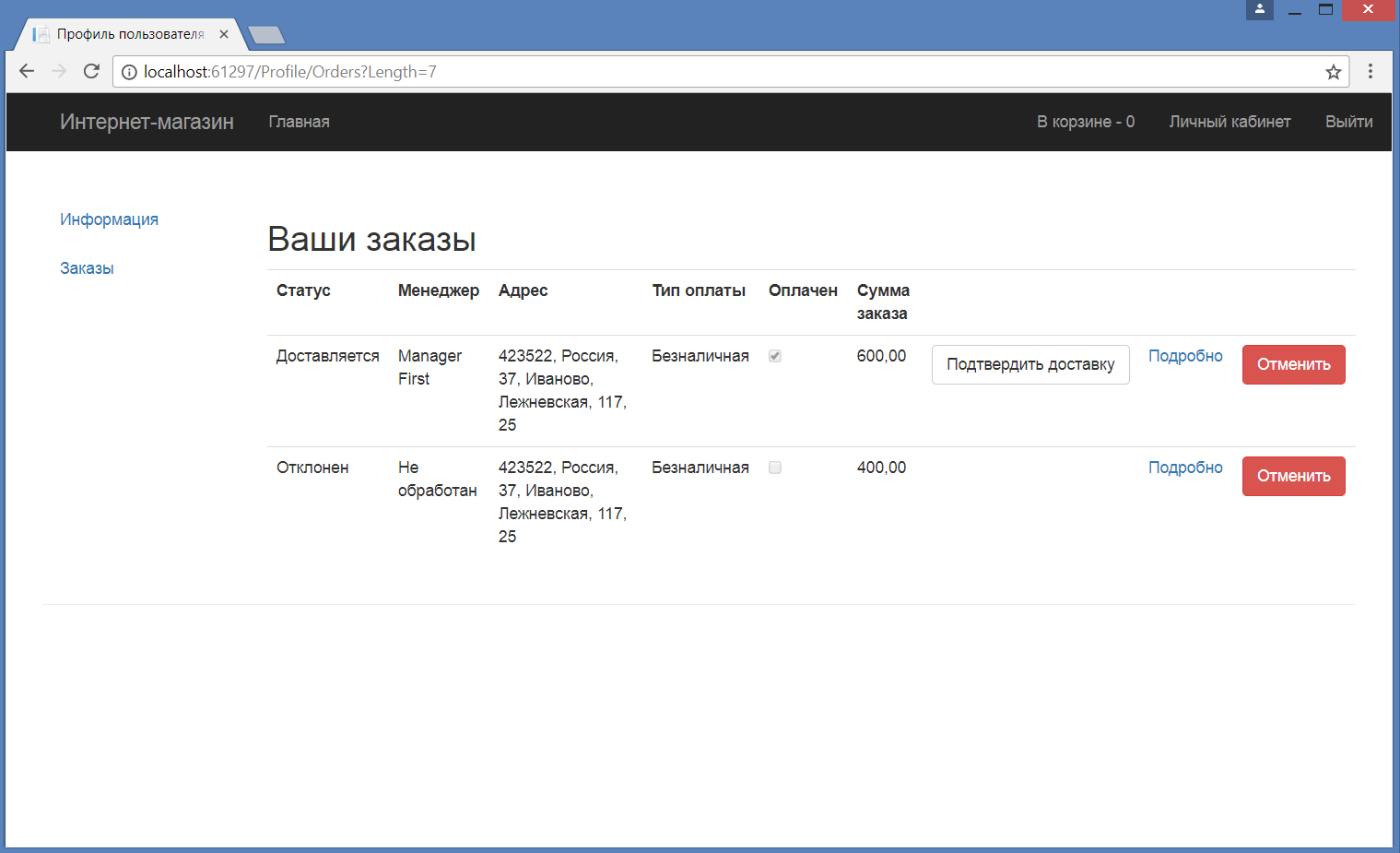


Рисунок 8. Страница с заказами пользователя.

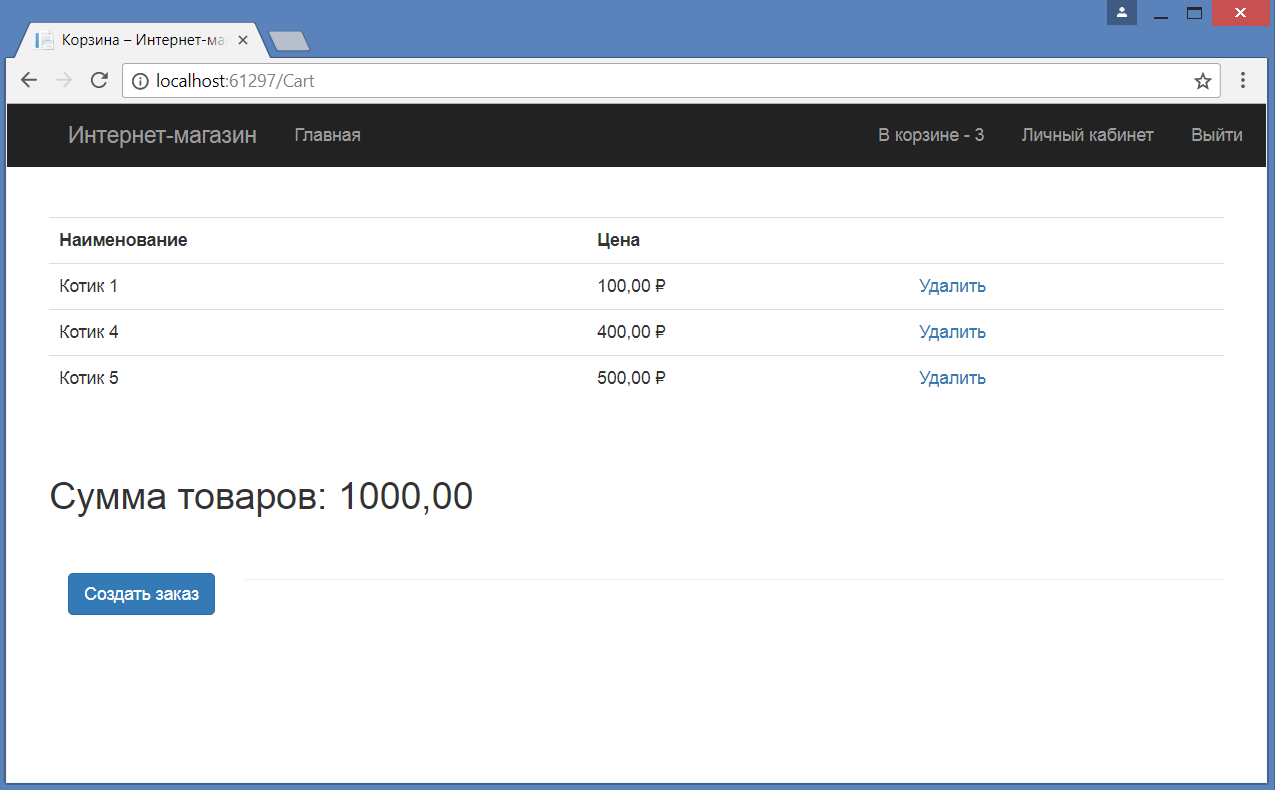


Рисунок 9. Страница корзины.

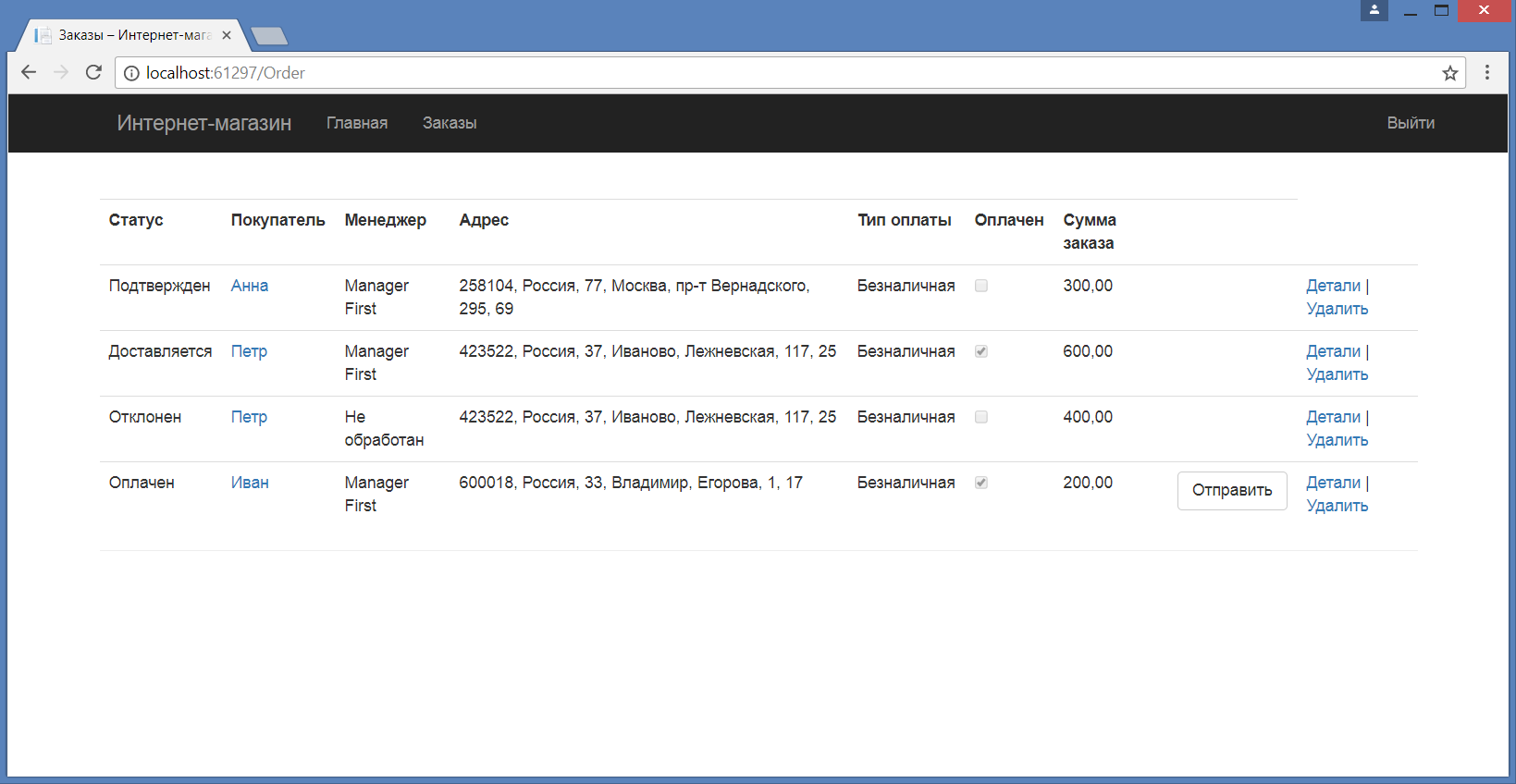


Рисунок 10. Страница с заказами для менеджера.

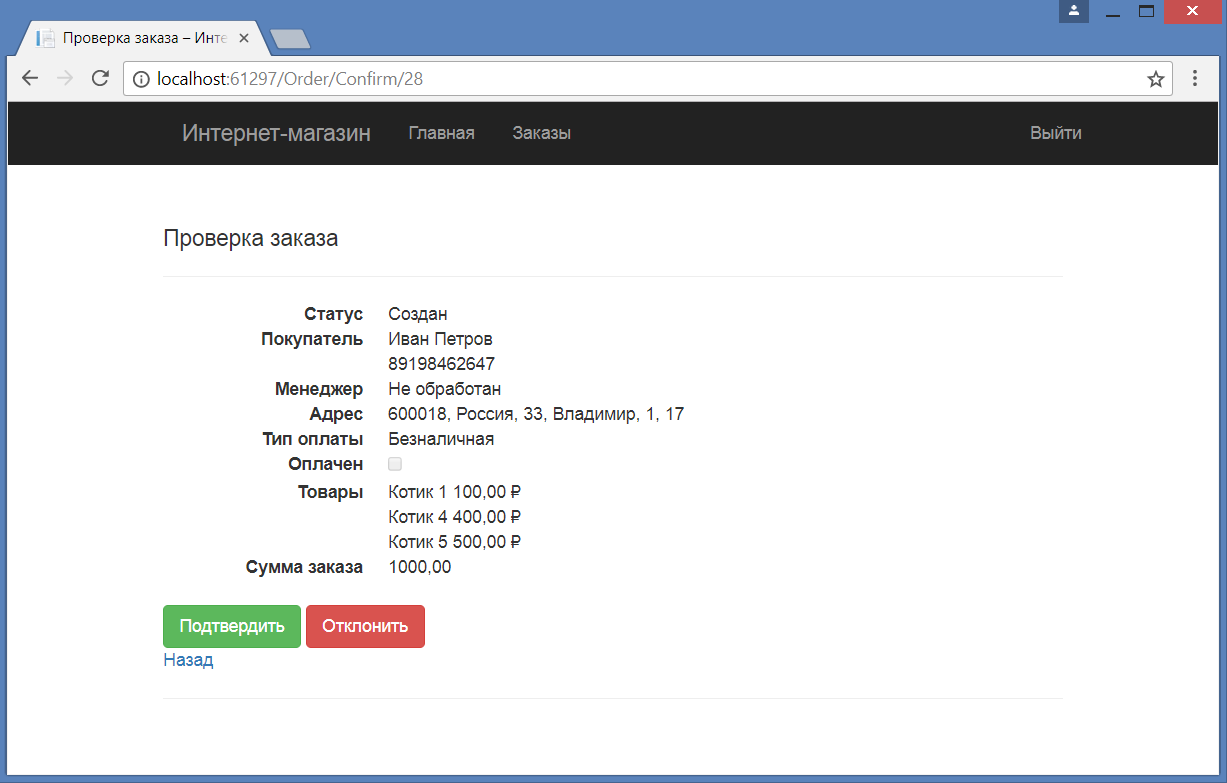


Рисунок 11. Страница подтверждения заказа.

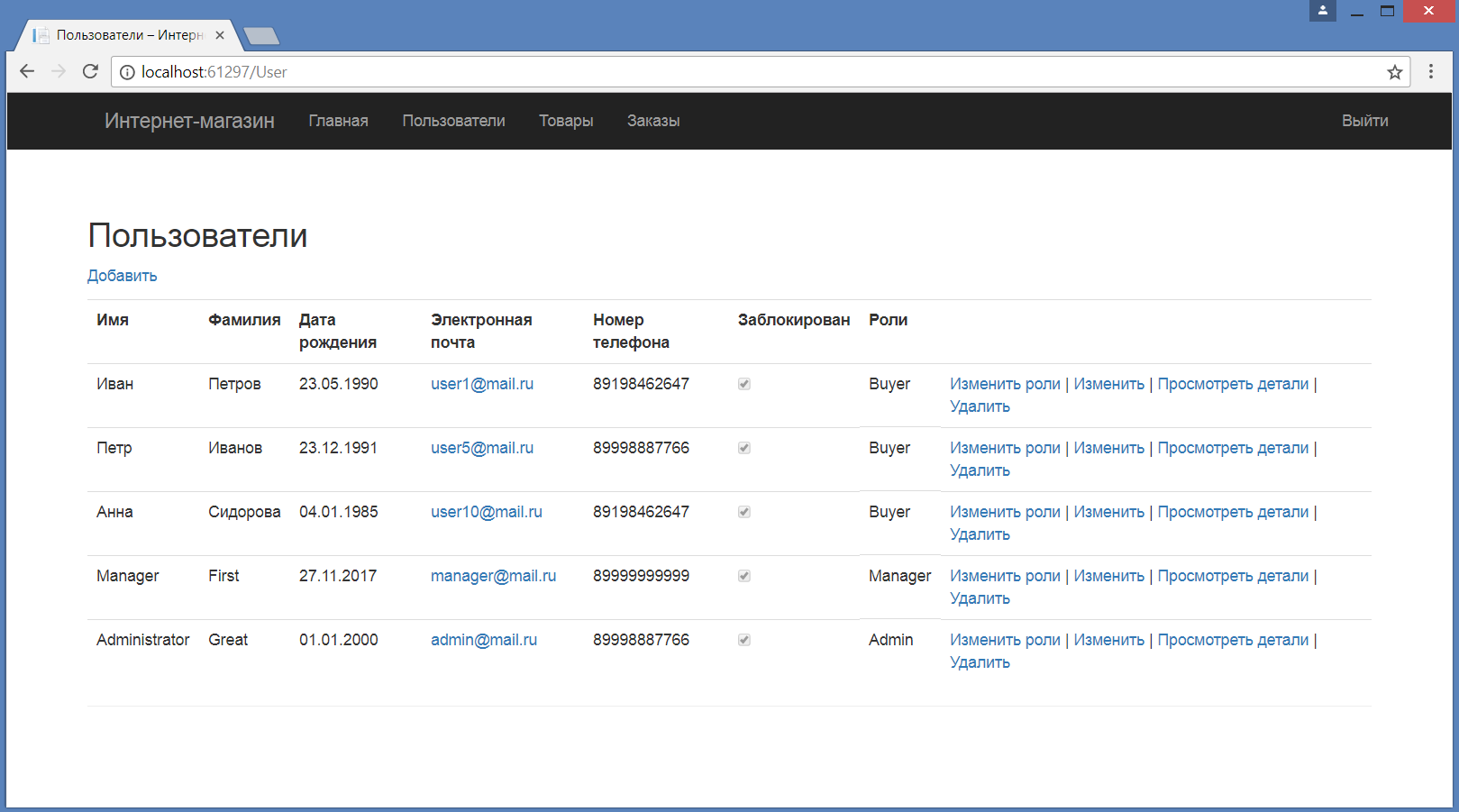


Рисунок 12. Страница с пользователями для администратора.

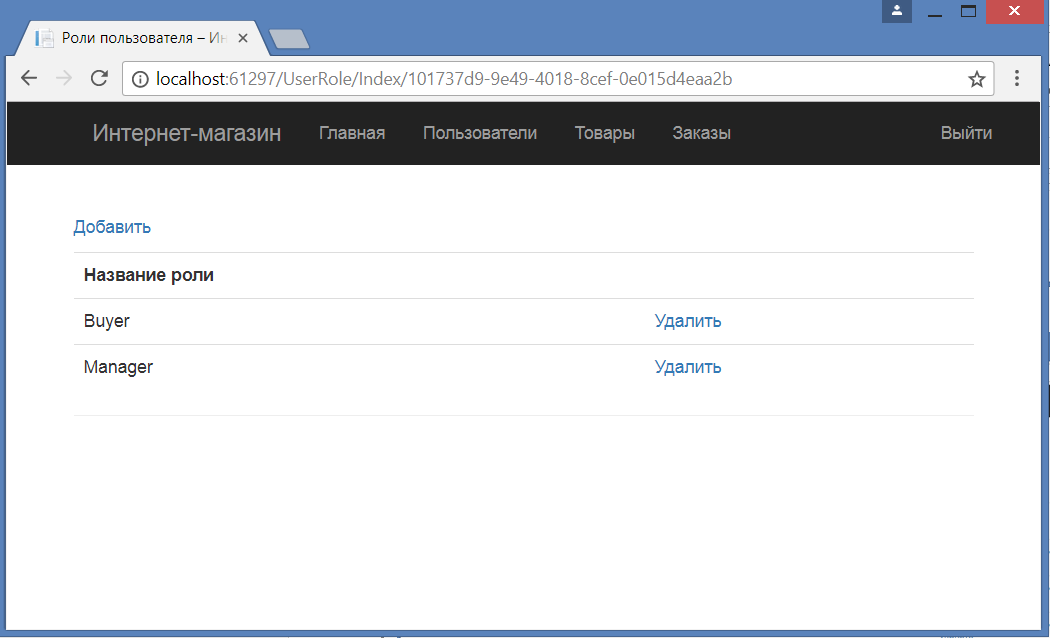


Рисунок 13. Страница изменения ролей пользователя.

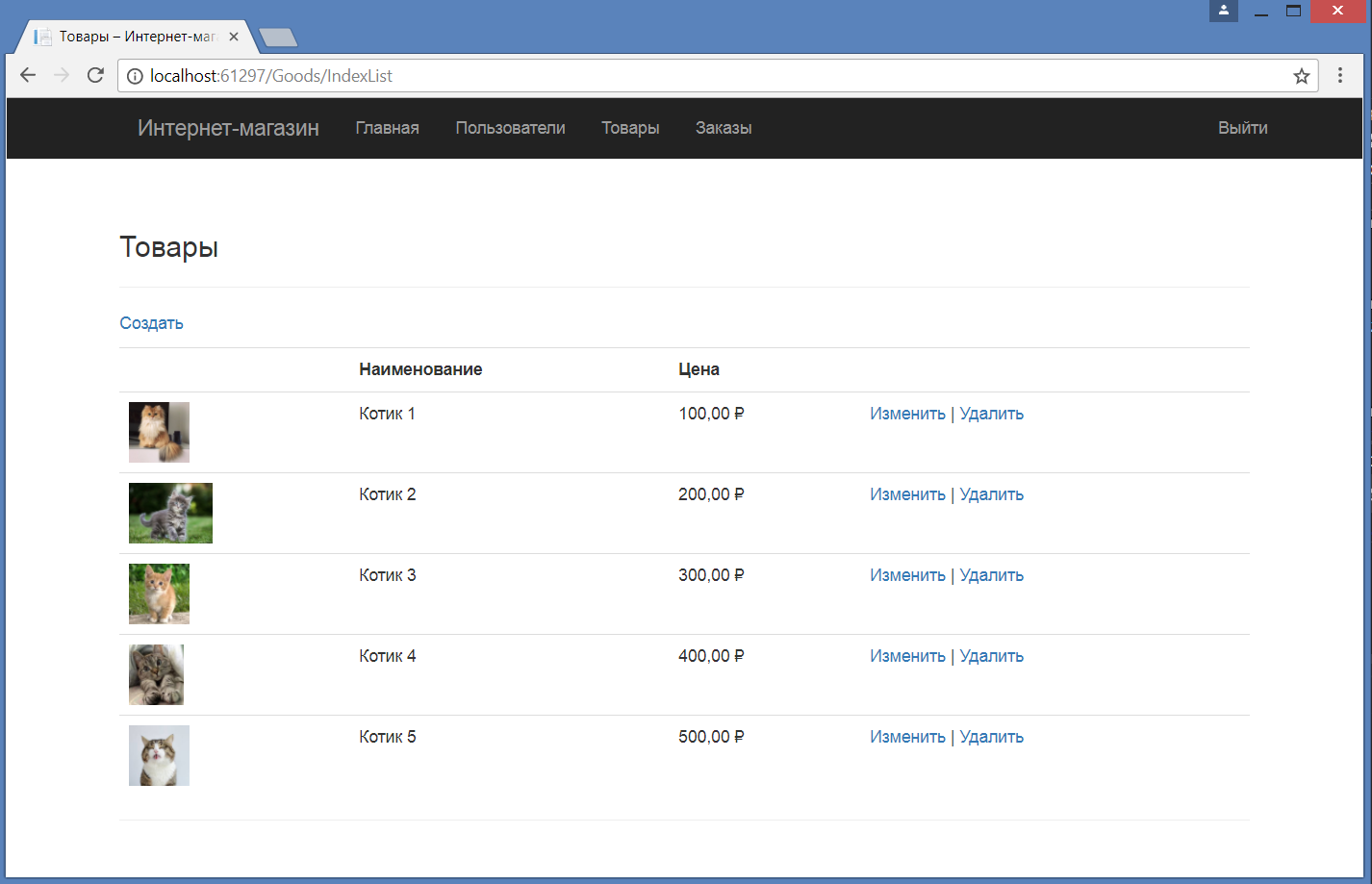


Рисунок 14. Страница с товарами для администратора.

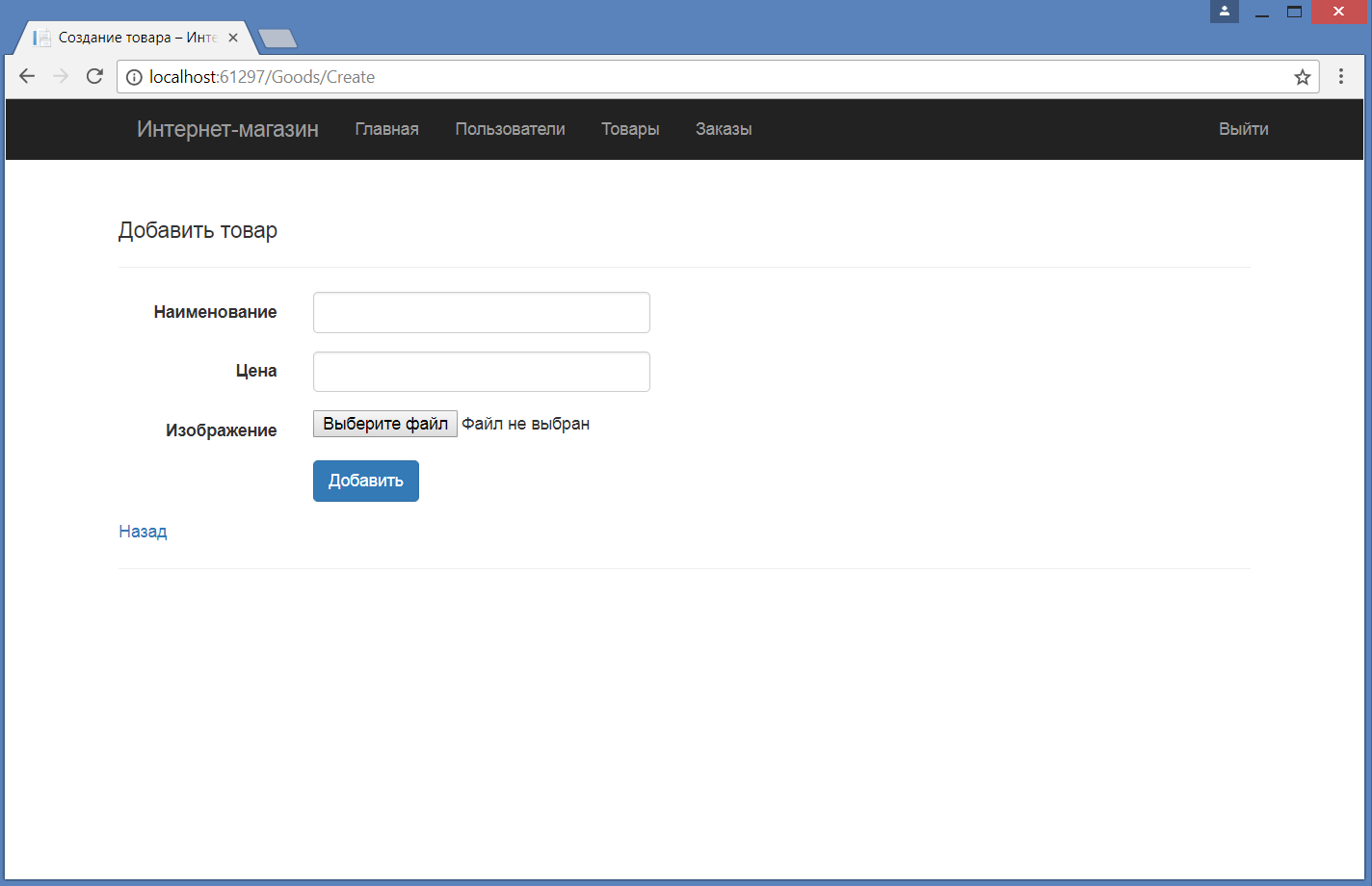


Рисунок 15. Страница добавления товара.

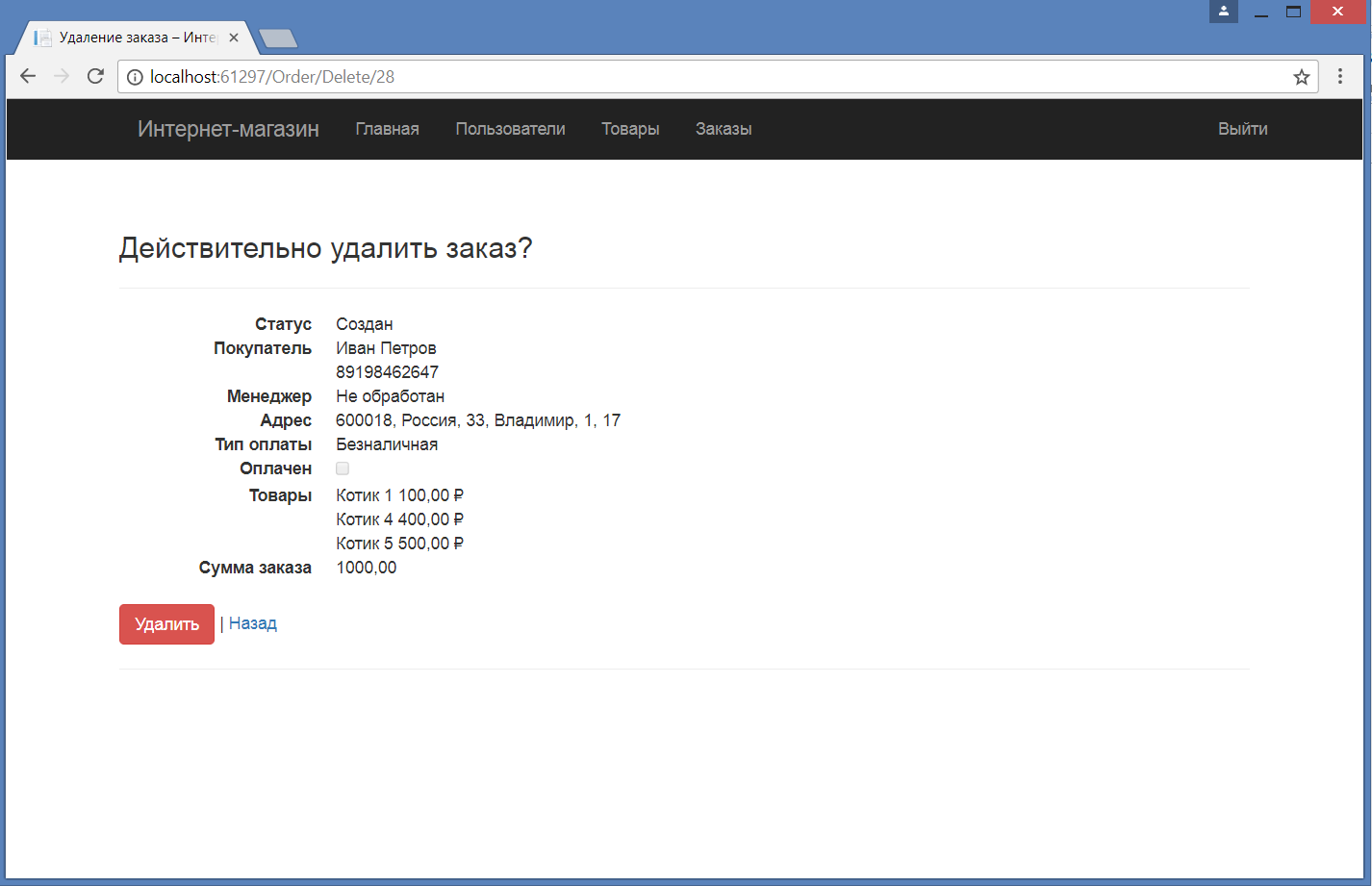


Рисунок 16. Страница удаления заказа.

# ТЕСТИРОВАНИЕ СИСТЕМЫ

# ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ А.