МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ.ПРОФ. М.А. БОНЧ-БРУЕВИЧА»

Факультет инфокоммуникационных сетей и систем

Кафедра программной инженерии и вычислительной техники

Курсовая работа

по дисциплине «Разработка Java-приложений управления телекоммуникациями»

на тему «Строительство. Поставщики и подрядчики»

Выполнил:

студент 4 курса

Группа:

ИКПИ-03

Азарченко И.К.

Преподаватель:

Березин А.Ю.

Санкт-Петербург

2024

Оглавление

[1. Введение 4](#_Toc163904597)

[2. Постановка задачи 4](#_Toc163904598)

[3. Используемые технологии 5](#_Toc163904599)

[3.1 HTML 5](#_Toc163904600)

[3.2 CSS 5](#_Toc163904601)

[3.3 JavaScript 5](#_Toc163904602)

[3.4 JavaFX 5](#_Toc163904603)

[4. Планирование 5](#_Toc163904604)

[4.1 Создание идеи 6](#_Toc163904605)

[4.2 Разработка структуры проекта 6](#_Toc163904606)

[5. Диаграмма классов 7](#_Toc163904607)

[6. Описание классов 7](#_Toc163904608)

[7. Описание базы данных 9](#_Toc163904609)

[8. Страницы веб-приложения 10](#_Toc163904610)

[*Рисунок 9.1 - Главная страница* 10](#_Toc163904611)

[*Рисунок 9.2 – Состав напитков* 10](#_Toc163904612)

[*Рисунок 9.3 – Страница администрирования* 10](#_Toc163904614)

[*Рисунок 9.4 - Окно авторизации* 11](#_Toc163904615)

[*Рисунок 9.5 - Окно регистрации* 11](#_Toc163904616)

[*Рисунок 9.6 – Окно информации о заказе* 12](#_Toc163904618)

[9. Заключение 13](#_Toc163904619)

[10. Приложение 13](#_Toc163904620)

# 

# Введение

Целью работы является изучение материалов по заданной теме и формирование веб-приложения на основе собранного материала.

В процессе достижения цели необходимо решить следующие задачи:

* Определить программный инструментарий, применяемый для разработки веб-приложения;
* Выявить и учесть методы и способы представления на веб-страницах различных видов инфорамации, не препятствующее их доступности;
* Ознакомиться с основными правилами и рекомендациями по разработке и созданию веб-приложений и принципиально следовать им в своей работе;
* Определиться со структурой веб-страниц.

# Постановка задачи

Задача курсового проекта заключается в разработке веб-приложения, посвященного обработке заявок на барную стойку при помощи менеджера.

* Главная страница с разделами менеджера;
* Динамическая страница отправки заявок на заказ напитка;
* Страница авторизации пользователя;
* Страница регистрации пользователя;
* Динамическая страница обработки заявок (для администраторов);
* Динамическая страница администрирования пользователей (для администраторов).

А также на каждой странице будут присутствовать: навигационная панель (меню сайта) с переходами на разделы приложения, страница скалада, страницу обработки заявок(для администраторов), страницу администрирования пользователей(для администраторов), страницу создания продукта, страницу выхода пользователя из учетной записи.

# Используемые технологии

* 1. HTML

С помощью данного языка разметки было реализовано отрисовка страниц и взаимодействия с ними.

* 1. CSS

При помощи данного языка описания разметки, задавались параметры элементов HTML-документа.

* 1. JavaScript

По нажатию на кнопку реализовано всплывающее окно с подтверждением отправки заявки на обработку

* 1. JavaFX

Пользовательский интерфейс реализован с использованием JavaFX

# Планирование

Основная функция сайта - удовлетворять информационные запросы пользователя, поэтому потребуется удобное навигационное меню.

Этап планирование можно разделить на несколько пунктов:

* Создание идеи;
* Разработка структуры проекта;
* Проработка макета проекта.

4.1 Создание идеи

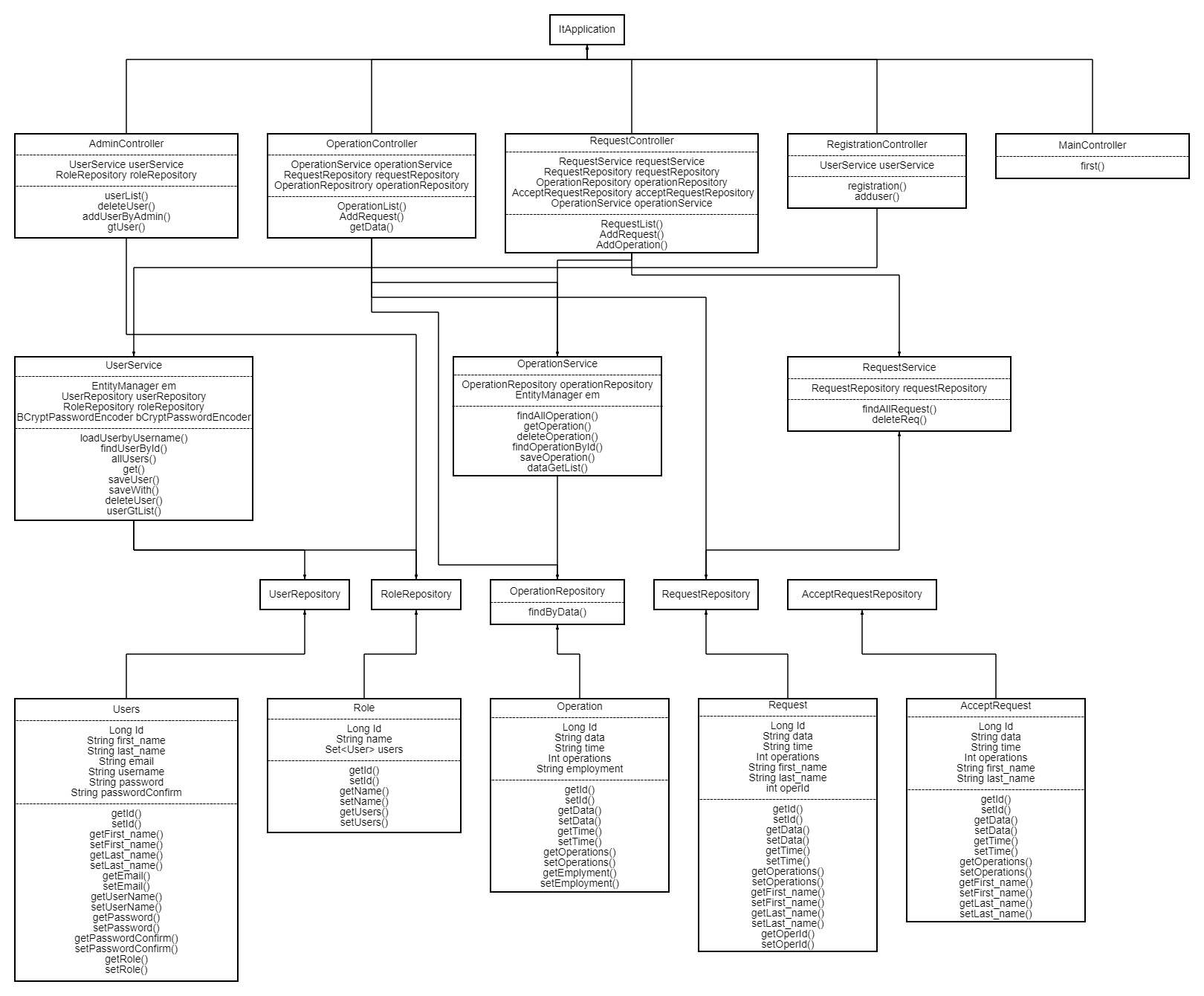
На данном этапе выбрана тема проекта «Ресторанный бизнес / общепит. Бар – торговля и склад». Далее, в соответствие с выбранной темой, собраны соответствующие материалы: текстовые и графические.

4.2 Разработка структуры проекта

Структура проекта подразумевает под собой разделы сайта, в соответствие с которыми будет формироваться навигационное меню и строиться дизайн проекта. Отдельные категории файлов помещены в свои папки: картинки в папку img, стили в папку css. В корне лежат основные страницы сайта:

* home.html - главная страница проекта;
* admin.html - страница администрирования пользователей и создание товара;
* login.html - страница авторизации пользователя;
* registration.html - страница создание учетной записи пользователя;
* storage.html - страница отображения блюд;
* ingredients.html - страница отображения продуктов на складе.

# Диаграмма классов



# Описание классов

Класс Users - создает таблицу с полями id, first\_name, last\_name, email, username, password.

Класс Roles - создает таблицу с полями id, name, user.

Класс Drink - создает таблицу с полями id, name, type.

Класс Request - создает таблицу с полями id, data, time, operations, first\_name, last\_name, operId.

Класс Ingredient - создает таблицу с полями id, name.

Класс Storage - создает таблицу с полями id, name, quantity.

Интерфейс UserRepository - класс реализует поиск, получение и сохранение данных таблицы Users.

Интерфейс DrinkRepository - класс реализует поиск, получение и сохранение данных таблицы Drink.

Интрефейс IngredientRepository - класс реализует поиск, получение и сохранение данных таблицы Ingredient.

Интерфейс RequestRepository - класс реализует поиск, получение и сохранение данных таблицы Request.

Интерфейс StorageRepository - класс реализует поиск, получение и сохранение данных таблицы Storage.

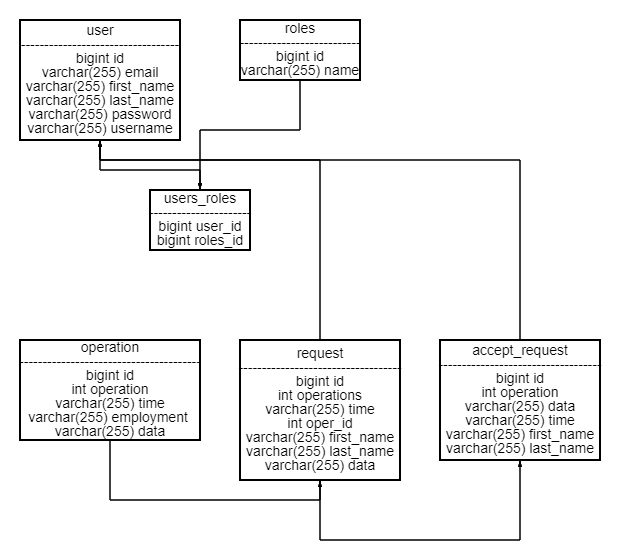
Класс UserService - реализует бизнес логику и классы, для взаимодействия в контроллере, такие как поиск поиск всех пользователей, удаление, сохранение, получение записи базы данных по Id. Для этого в классе инициализируется UserRepository.

Класс AdminController - реализует get и post запросы. В get запросе мы добавляем модель AllUsers, которая отрисовывает всех пользователей, находящиеся в базе данных. В post запросе мы работает с полученными данными со страницы, то есть при нажатии на кнопку delete, **give doctor, delete doctor,** на сервер поступает post запрос с id пользователя и далее мы работаем с ними (удаляем запись из бд, добавляем в поля значения).

Класс RegistrationCotroller - реализует get и post запросы. В get запросе создается пустой объект класса User, для дальнейшего добавления в базу данных. В post запросе имеется проверка данных: есть ли такой же пользователей уже в системе, совпадает ли пароль с полем подтверждение пароля, а также занесение пользователя в базу данных.

Класс MainController - реализует get запрос, который возвращает главную страницу.

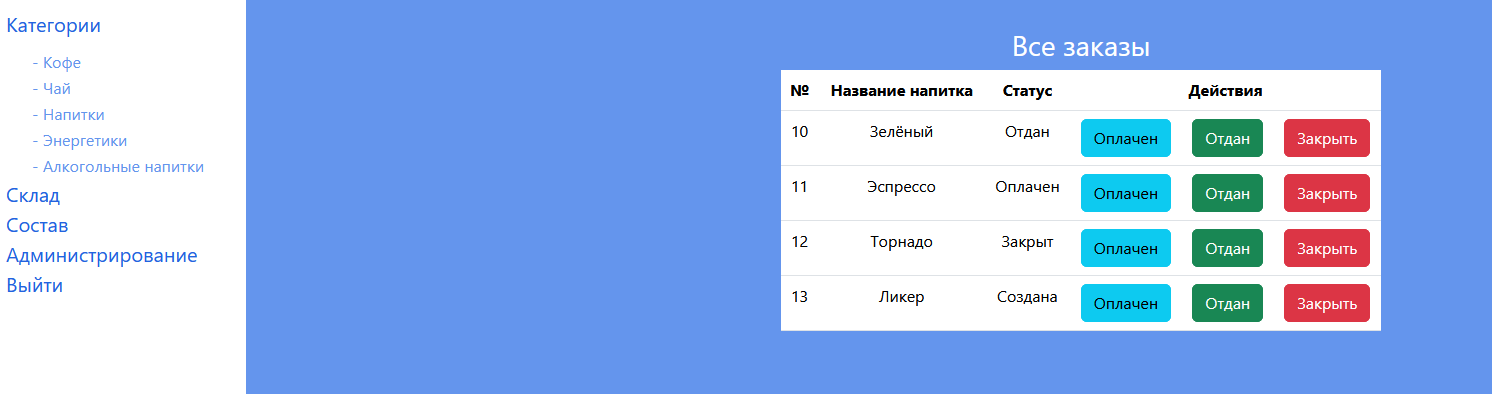
Схема базы данных



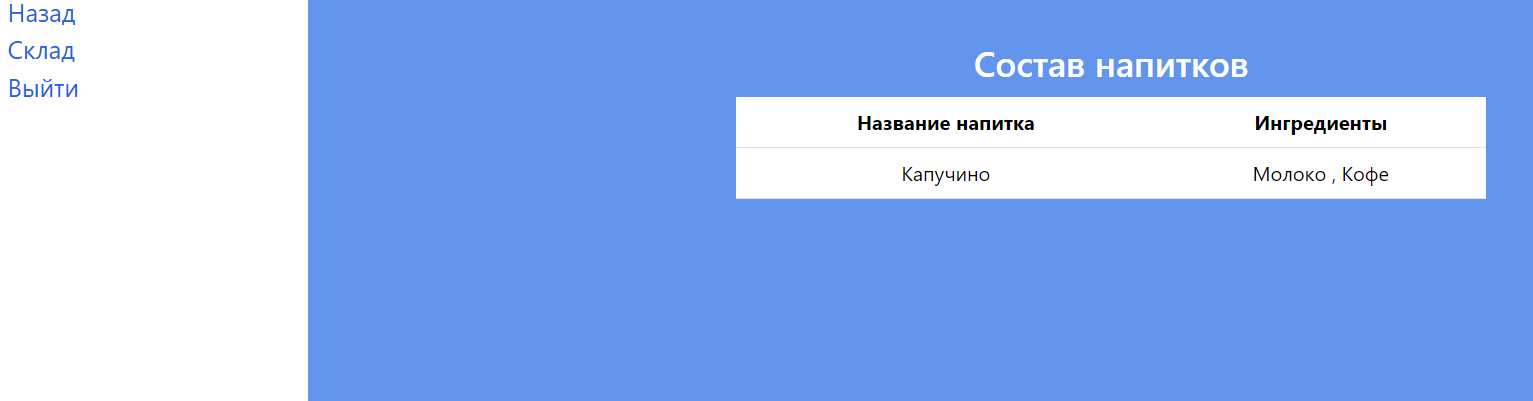
# Описание базы данных

В таблице user хранятся все зарегистрированные пользователи. В таблице roles хранятся роли для пользователей. В таблице users\_roles сопоставляются id пользователей и id ролей. В таблице operation хранятся все записи на бронирование операционных. В таблице request хранятся записи на бронирование, отправленные пользователями. В таблице AcceptRequest хранятся подтвержденные доктором или администратором заявки.

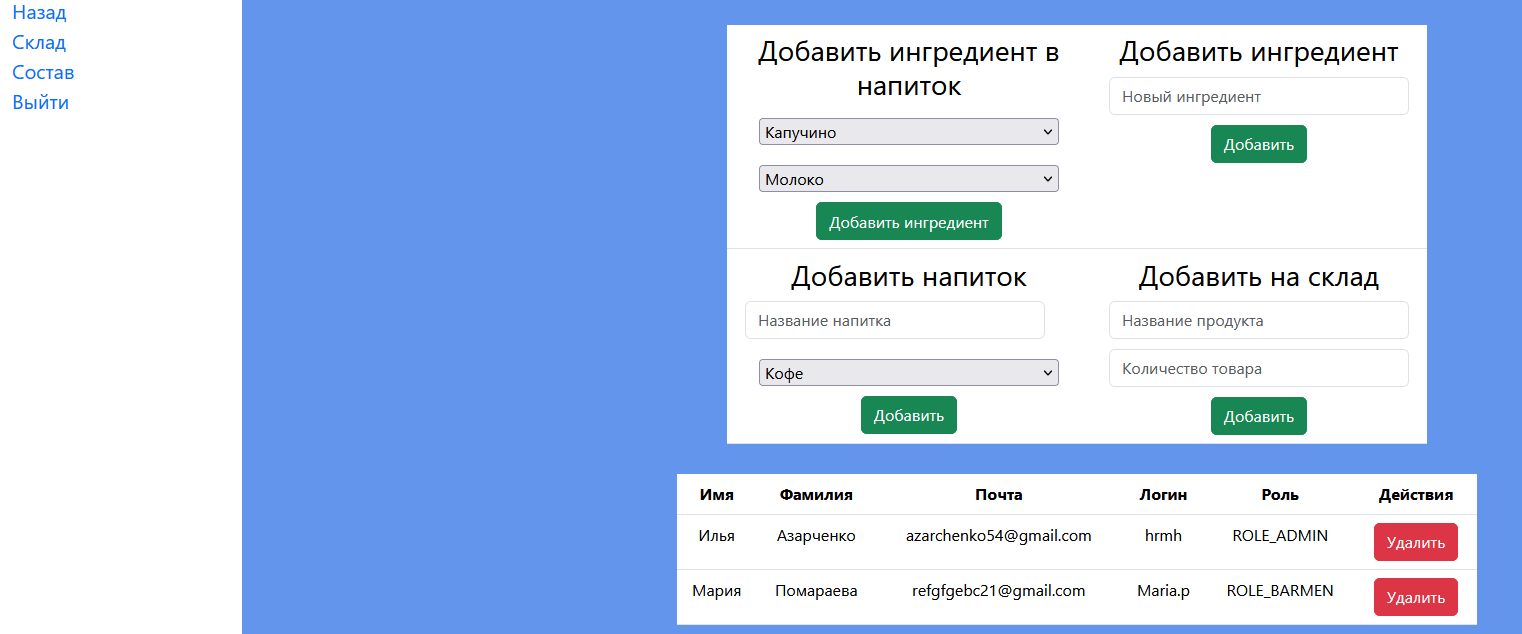
# Страницы веб-приложения



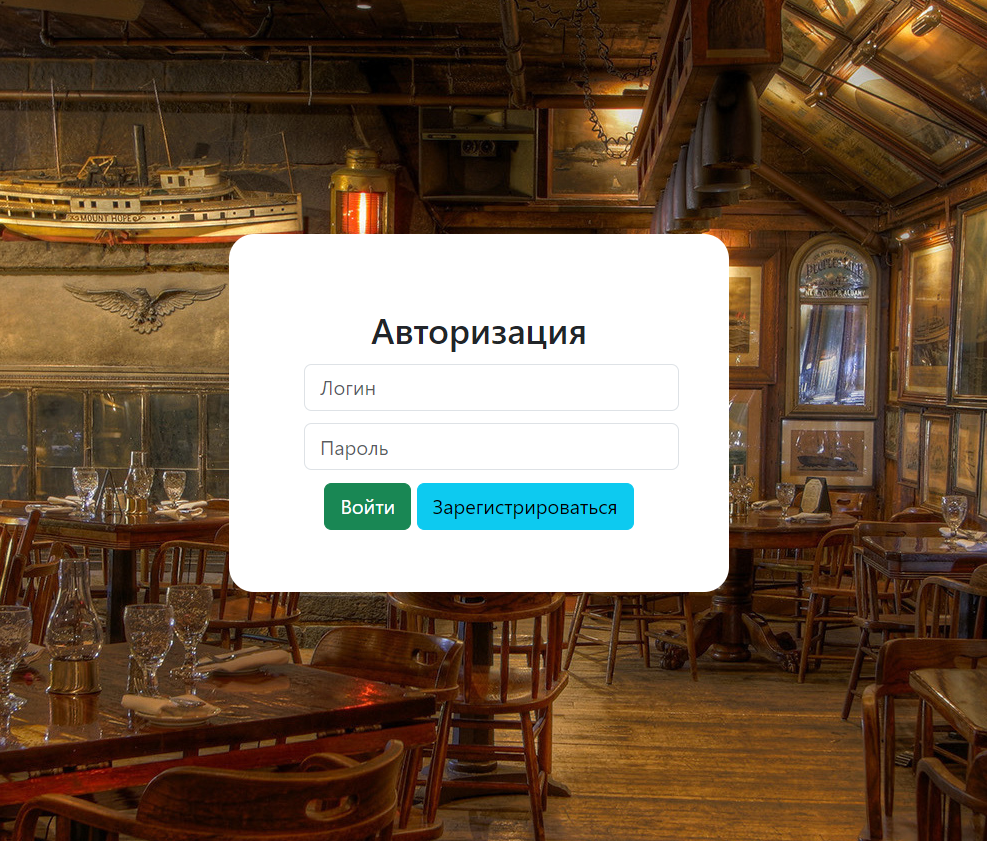
*Рисунок 9.1 - Главная страница*



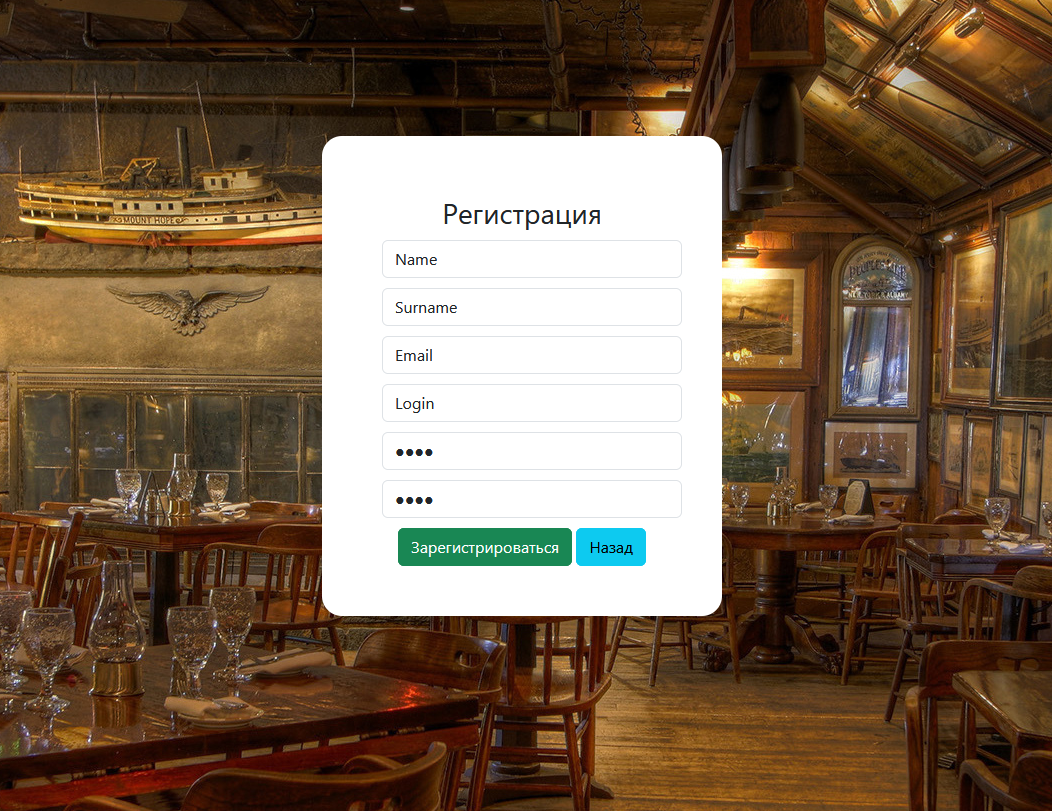
*Рисунок 9.2 – Состав напитков*



*Рисунок 9.3 – Страница администрирования*



*Рисунок 9.4 - Окно авторизации*



*Рисунок 9.5 - Окно регистрации*



*Рисунок 9.6 – Окно информации о заказе*

# Заключение

В результате выполнения курсовой работы было разработано веб-приложение, посвященное обработке заявок на бронирование операционных. Веб-приложение полностью удовлетворяет заданным требованиям.

# Приложение

В приложение приведена ссылка на GitHub основных модулей разработанного веб-приложения.