**Министерство профессионального образования и занятости населения Приморского края**

**КРАЕВОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

Отделение экономики и сервиса

Специальность *09.02.07*

Информационное системы и программирование

Работа допущена к защите:

Завотделением \_\_\_\_\_\_\_ А.А.Савина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**КУРСОВОЙ ПРОЕКТ**

**по теме:**

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АДМИНИСТРАТОРА АКВАПАРКА**

**Дисциплина: Основы алгоритмизации и программирования**

|  |  |
| --- | --- |
| **Выполнил**  студент группы 341 | **Руководитель**  преподаватель |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Глушенко | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Пушкин |
| Оценка, полученная на защите КР, | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Владивосток 2023**

**Министерство профессионального образования и занятости населения Приморского края**

**краевое государственное автономное профессиональное**

**образовательное учреждение**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

**(КГА ПОУ «ППК»)**

**Отделение Экономики и сервиса**

**З А Д А Н И Е**

на курсовой проект

Заведующий отделением \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Савина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

|  |
| --- |
| *студенту Глушенко Александру Владимировичу 1125 группы* |

(фамилия, имя, отчество)

на тему «Разработка приложения для администратора аквапарка»

|  |
| --- |
|  |
|  |

Вопросы, подлежащие разработке (исследованию):

|  |
| --- |
| 1. Изучение предметной области. Выделение объектов и связей между ними. |
| 1. Разработка схемы базы данных |
| 1. Построение диаграммы объектов. |
| 1. Написание технического задания на разработку системы |
| 1. Разработка функциональной части приложения |
| 1. Разработка пользовательского интерфейса приложения |
|  |

Основные источники информации и прочее, используемые для разработки темы

|  |
| --- |
| 1. Гаурав Арораа, Джефри Чилберто – Паттерны проектирования для C# и платформы .Net Core, 2021 |
| 1. Эндрю Лок – Asp.Net Core в действии, 2021 |
| 1. Прайс Марк – C# 10 и .Net 6. Современная кросс-платформенная разработка, 2023 |
|  |

Срок предоставления работы «17» мая 2023 г.

Дата выдачи задания «01» апреля 2023 г.

Руководитель КП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Пушкин

(подпись) (и.о. фамилия)

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Глушенко

(подпись) (и.о. фамилия)

**Лист нормоконтроля**

Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Нормоконтроль

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (и.о. фамилия)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ](#_Toc796723) 3

[1 Проектирование информационной системы 4](#_Toc796723)

[1.1 Определение пользовательских требований 4](#_Toc796723)

[1.2 Диаграмма вариантов использования 5](#_Toc796723)

[1.3 Диаграмма деятельности системы 6](#_Toc796723)

[1.4 Схема БД 9](#_Toc796723)

[2 Реализация информационной системы 15](#_Toc796735)

[2.1 Описание интерфейса приложения 15](#_Toc796735)

[2.2 Пользовательские сценарии 24](#_Toc796735)

[ЗАКЛЮЧЕНИЕ 26](#_Toc796723)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc796723)

[ПРИЛОЖЕНИЕ А – Регистрация 28](#_Toc795068)

[ПРИЛОЖЕНИЕ Б – Редактирование и удаление блюда 29](#_Toc795069)

**ВВЕДЕНИЕ**

Курсовая работа на тему "Разработка приложения для администратора аквапарка" направлена на создание программного решения для упрощения работы администраторов аквапарка и повышения эффективности их деятельности. Основной целью данной работы является создание функционального и удобного приложения, которое позволит администраторам управлять и контролировать различные аспекты работы аквапарка. В рамках данной работы будут рассмотрены следующие задачи:

1. Изучение существующих программных решений для управления аквапарком и выявление их преимуществ и недостатков.
2. Разработка функциональных требований и интерфейса приложения на основе анализа потребностей администраторов аквапарка.
3. Создание программного кода и тестирование приложения на работоспособность и соответствие заданным требованиям. В результате выполнения курсовой работы ожидается получение полноценного приложения для управления аквапарком, способного значительно упростить и ускорить работу администраторов и улучшить качество обслуживания посетителей аквапарка.

**1. Проектирование информационной системы**

**1.1. Определение пользовательских требований**

Пользовательские требования для приложения для администратора аквапарка - это набор функциональных и нефункциональных требований, которые определяют, как приложение должно работать и каким образом должно взаимодействовать с пользователями - администраторами аквапарка.

Функциональные требования определяют конкретные функции и возможности, которые приложение должно предоставлять. Например, это может быть возможность управления расписанием работы аквапарка, отслеживание продажи билетов, создание отчетов о посещении аквапарка и т.д.

Нефункциональные требования определяют качественные характеристики приложения, такие как производительность, безопасность, удобство использования и т.д., например, приложение должно обеспечивать быструю загрузку данных, защиту данных администраторов аквапарка, простой и интуитивно понятный интерфейс.

Определение пользовательских требований является важным этапом разработки приложения для администратора аквапарка, поскольку позволяет определить требования и ожидания пользователей, на основе которых будет создано эффективное и удобное в использовании приложение.

Разработка приложения для администратора аквапарка должна учитывать потребности и ожидания пользователей. Ниже приведены некоторые пользовательские требования, которые можно учесть при создании такого приложения:

1. Управление продажей билетов - приложение должно предоставлять администраторам возможность покупки и продажи билетов, а также контроля за продажами.
2. Мониторинг посещаемости - приложение должно позволять администраторам отслеживать посещаемость аквапарка в режиме реального времени, а также предоставлять статистику о посещаемости за определенный период времени.
3. Создание отчетов - приложение должно предоставлять администраторам возможность создания отчетов о посещении аквапарка и продажах билетов за определенный период времени.
4. Управление персоналом - приложение должно позволять администраторам управлять персоналом аквапарка, включая назначение задач и контроль за выполнением.
5. Управление оборудованием аквапарка, включая деятельность всех его зон, таких как пищеблок, сауны, и зона с аттракционами.
6. Простой и интуитивно понятный интерфейс - приложение должно иметь простой и удобный интерфейс, понятный для пользователей с разным уровнем технической грамотности.

**1.2. Диаграмма вариантов использования**

Диаграмма вариантов использования приложения выглядит так (см. Рисунок 1):

Клиент может просматривать блюда, сауны и аттракционы аквапарка, бронировать билеты, создавать аккаунты и входить в них и просматривать список своих броней.

Администратор имеет те же права, что и клиент, но ко всему добавляется возможность редактировать блюда, сауны и аттракционы аквапарка, а также просматривать список всех билетов всех пользователей.

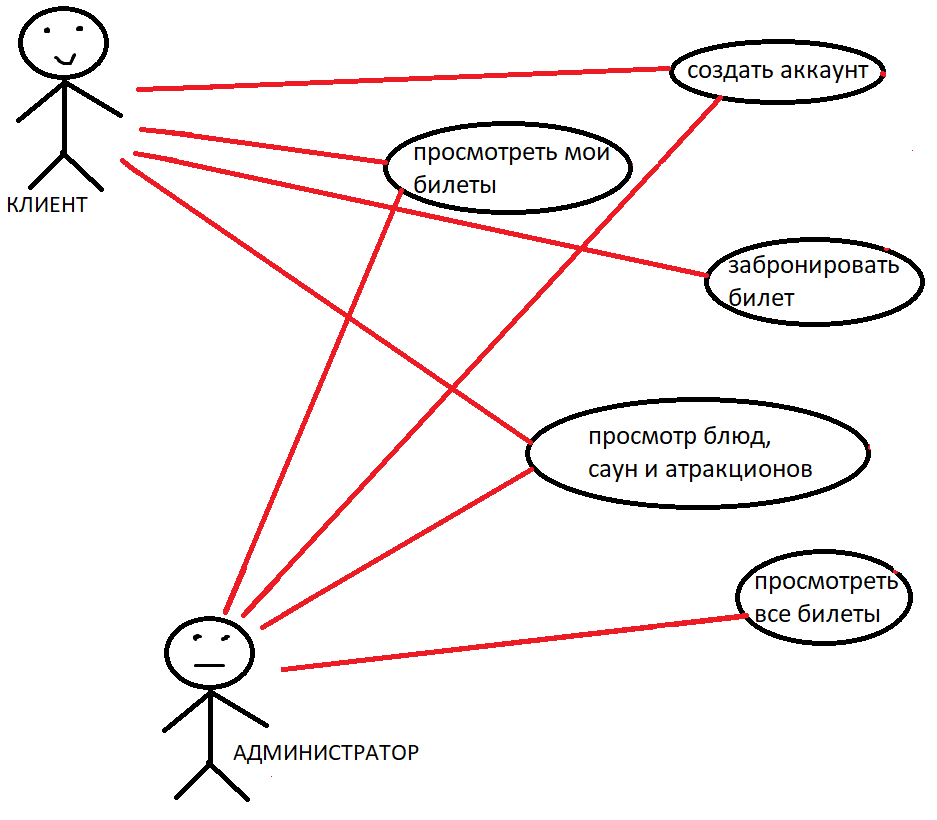


Рисунок 1. – Диаграмма вариантов использования приложения

**1.3. Диаграмма деятельности системы**

Процесс удаления из базы данных брони администратором (см. Рисунок 2):

Кликая ПКМ по брони пользователю высвечивается опция «Удалить», нажав на которую выскакивает окно подтверждения, если пользователь нажмёт «Нет», то ничего не произойдёт, если «Да», выбранная им ранее бронь будет удалена из базы данных.

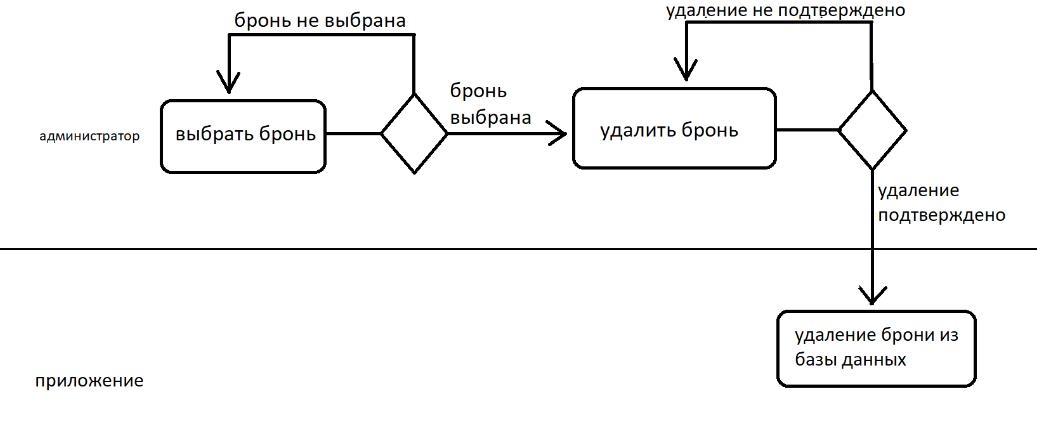


Рисунок 2 – Диаграмма деятельности системы удаления брони

Кликая ПКМ по блюду администратору высвечиваются три опции: добавить, редактировать и удалить. При нажатии на «Удалить» выскакивает окно подтверждения, соответственно, при нажатии «Нет» ничего не происходит, на «Да» - блюдо удаляется из базы данных. При нажатии на «Добавить» выскакивает окно добавления с формами для ввода данных о блюде, их можно оставить пустыми, можно заполнить, в любом случае на кнопку «Сохранить» в базу данных добавится новое блюдо. На кнопку «Редактировать» высветиться окно редактирования с формами, заполненными существующими данными, которые можно поменять. На кнопку «Сохранить» изменения сохраняются (См. рисунок 3).

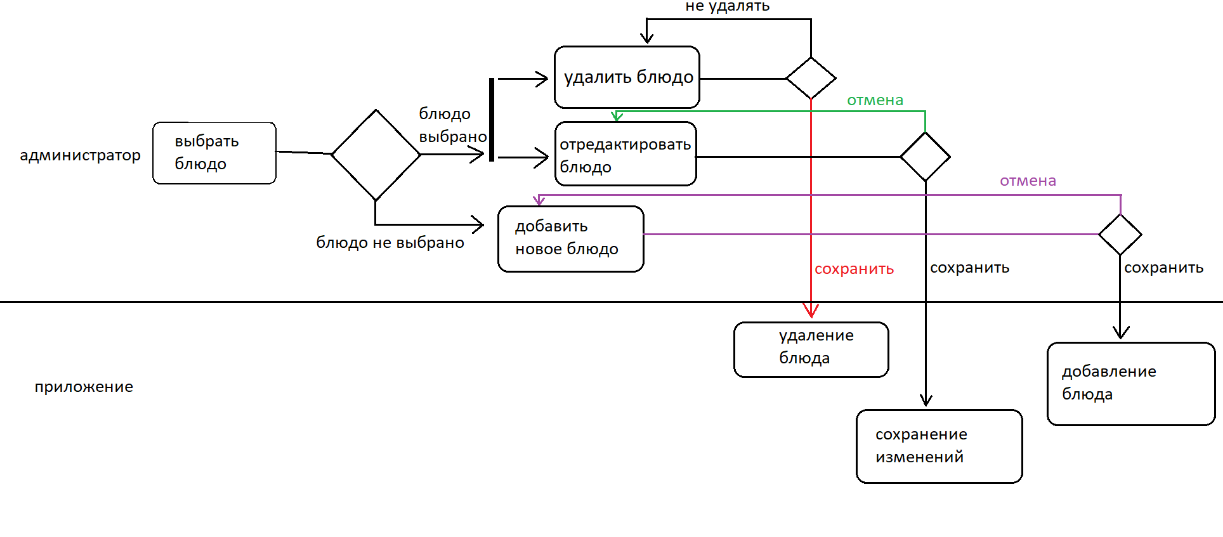


Рисунок 3 – Диаграмма деятельности системы записей с блюдами

При запуске приложения у пользователя есть выбор: войти в уже существующий аккаунт или создать новый. Если аккаунта у пользователя нет, то он перейдёт на окно регистрации. С него он может обратно перейти на окно авторизации, если вспомнил, что у него есть аккаунт или по другой причине. Чтобы создать новый аккаунт надо заполнить формы с именем, фамилией, отчеством, почтой и две формы для пароля. Если не все формы заполнены, то вылезет сообщение, информирующее об этом, и аккаунт не создаётся. Если все данные заполнены, то идёт ещё одна проверка: если пароли из обоих форм не совпадают, то вылезет сообщение, что пароли не совпадают, соответственно аккаунт тоже не создаётся. Если обе проверки пройдены, то создаётся новый аккаунт, а у пользователя всё также есть возможность вернуться на окно авторизации, чтобы войти с него в приложение (см. Рисунок 4).

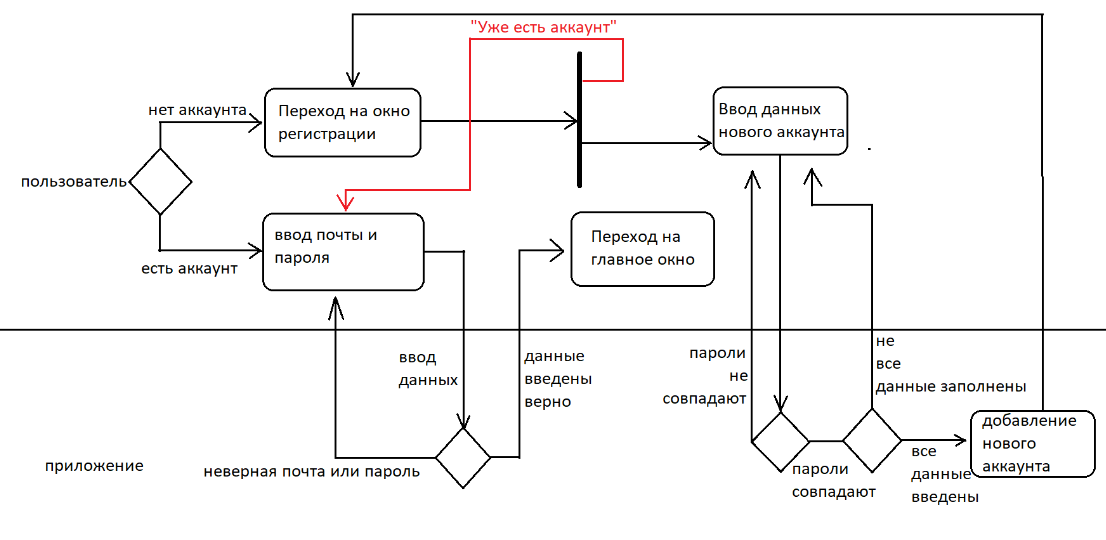


Рисунок 4 – Диаграмма деятельности системы аккаунтов в приложении

**1.4. Схема базы данных**

База данных содержит 7 таблиц (см. Рисунок 5).

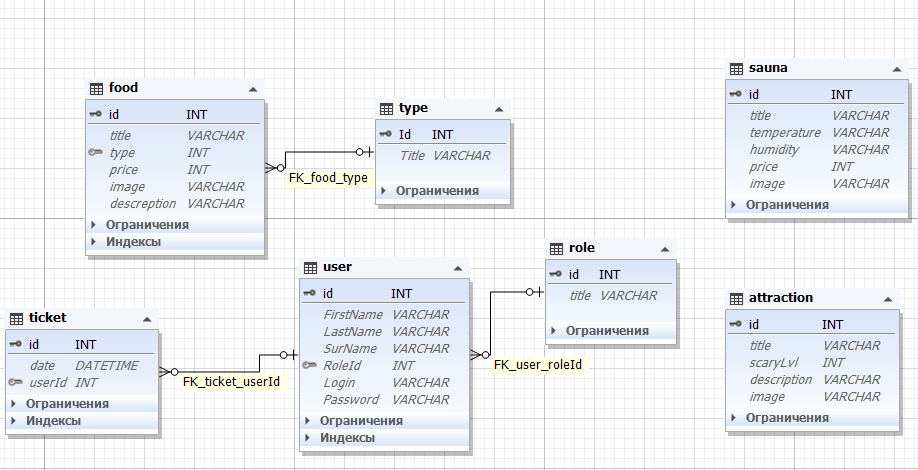


Рисунок 5 – Диаграмма базы данных

Таблица user хранит информацию о пользователе и содержит 7 свойств (см. Рисунок 6):

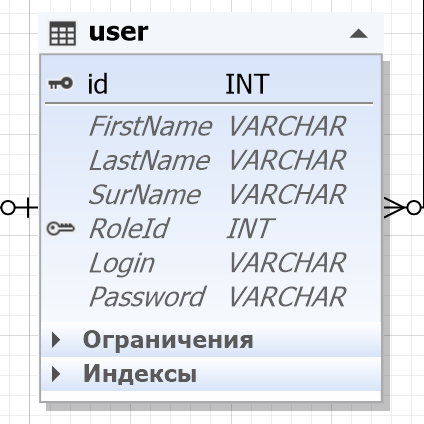


Рисунок 6 – Таблица user

1. Id – уникальный номер пользователя в базе данных.
2. FirstName – Имя пользователя.
3. LastName – Фамилия пользователя.
4. SurName – Отчество пользователя.
5. RoleId – айди роли пользователя (о ролях позже).
6. Login – Электронная почта пользователя.
7. Password – Пароль от аккаунта пользователя в приложении.

Таблица ticket хранит информацию о брони, которую может сделать пользователь и содержит 3 свойства (см. Рисунок 7):

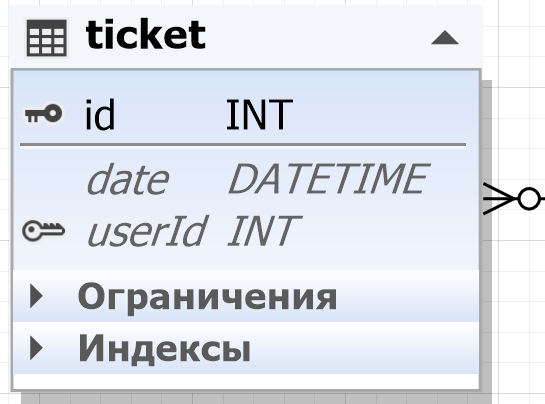


Рисунок 7 – Таблица ticket

1. id – уникальный номер билета в базе данных.
2. date – дата, на которую забронирован билет.
3. userId – айди пользователя, который забронировал билет.

Таблица role хранит информацию о ролях пользователей, у разных ролей разные права доступа. Содержит 2 свойства (см. Рисунок 8):

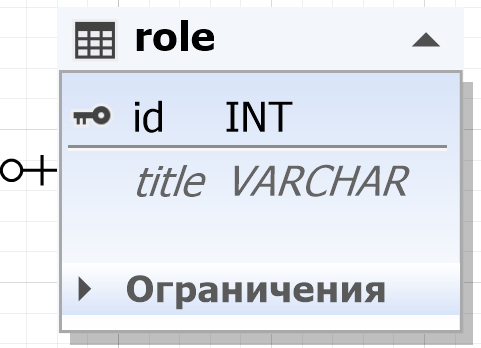


Рисунок 8 – Таблица role

1. id – уникальный номер роли в базе данных.
2. title – название роли.

Таблица attraction хранит информацию об аттракционах в аквапарке и содержит 5 свойств (см. Рисунок 9):

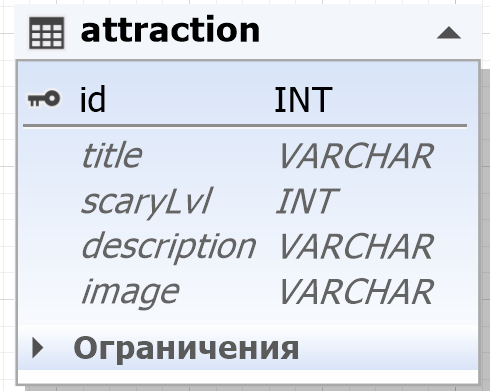


Рисунок 9 – Таблица attraction

1. id – уникальный номер аттракциона в базе данных.
2. title – название аттракциона.
3. scaryLvl – уровень страха аттракциона.
4. description – краткое описание аттракциона.
5. image – путь к картинке аттракциона.

Таблица sauna хранит информацию о банях и саунах в аквапарке и содержит 6 свойств (см. Рисунок 10):



Рисунок 10 – Таблица sauna

1. id – уникальный номер сауны в базе данных.
2. title – название сауны.
3. temperature – температура сауны.
4. humidity – влажность воздуха сауны.
5. price – цена посещения сауны.
6. image – путь к картинке сауны.

Таблица food хранит информацию о блюдах буфета аквапарка. Содержит 6 свойств (см. Рисунок 11):

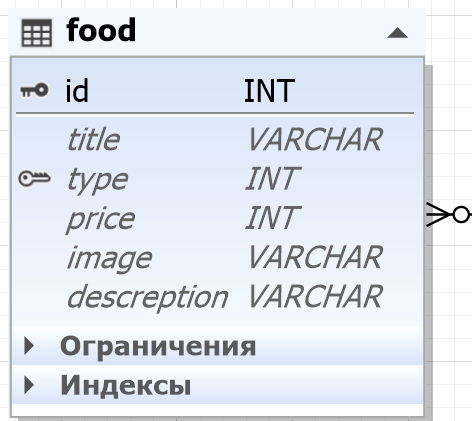


Рисунок 11 – Таблица food

1. id – уникальный номер блюда в базе данных.
2. title – название блюда.
3. type – айди типа блюда (о типах позже).
4. price – цена за 1 порцию блюда.
5. image – путь к картинке блюда. description – краткое описание блюда, его состав.

Таблица type хранит информацию о типах блюд, блюда делятся на разные типы (в данном случае на два). Содержит 2 свойства (см. Рисунок 12):

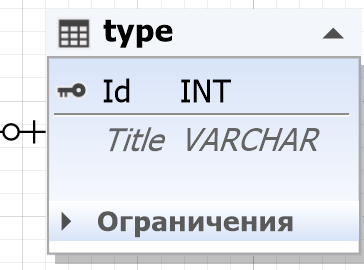


Рисунок 12 – Таблица type

1. id – уникальный номер типа в базе данных.
2. title – название типа

Таблицы user и ticket имеет связь многие к одному: у одного пользователя может быть много броней, но одна бронь может быть только у одного пользователя. Такую же связь имеют таблицы role и user: у пользователя может быть одна роль, а одна роль может быть у многих пользователей. Также связь многие к одному имеют таблицы type и food: у одного блюда может быть один тип, но один тип может включать в себя множество блюд.

**2. Реализация информационной системы**

**2.1. Описание интерфейса приложения**

Когда мы запускаем приложение, первым, что нам представляется нашему взору, является окно авторизации (см. Рисунок 13). Это окно содержит два поля ввода: одно для электронной почты и другое для пароля. Кроме того, мы можем увидеть логотип аквапарка, оформленный в фиолетовых цветах, который добавляет приятный эстетический элемент в интерфейс приложения. Чтобы обеспечить более удобный пользовательский опыт, под полем ввода пароля мы можем увидеть две кнопки. Первая кнопка с надписью «Забыл(а) пароль?» может пригодиться тем, кто не помнит свой пароль и хочет его сбросить. Вторая кнопка, помеченная надписью «Нет аккаунта?», будет использоваться пользователями, которые в первый раз используют приложение. Понятно, что, если пользователь никогда раньше не заходил в приложение, у него еще нет аккаунта, поэтому он кликнет на кнопку «Нет аккаунта?». После нажатия этой кнопки, приложение перенаправит пользователя на окно регистрации, где он сможет создать свой аккаунт и начать пользоваться всеми возможностями, которые предоставляет это приложение.

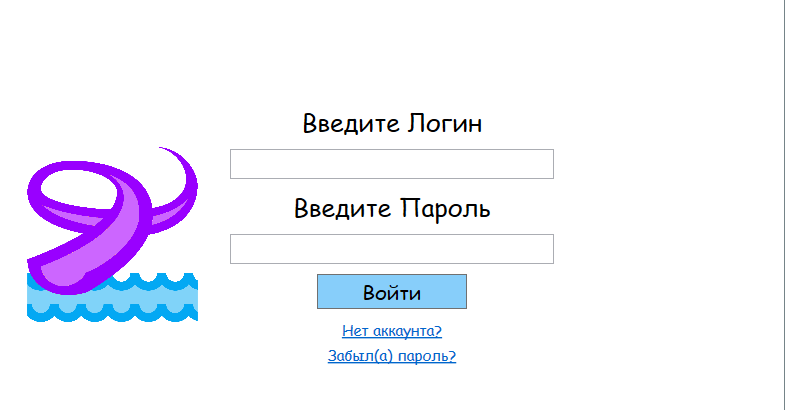


Рисунок 13 – Окно авторизации

При создании нового аккаунта в приложении, нам нужно будет заполнить несколько форм в окне регистрации (см. Рисунок 14). В этих формах мы должны ввести свое Имя, Фамилию, Отчество, адрес электронной почты и два пароля, для подтверждения своей личности и безопасности аккаунта. Важно убедиться, что все поля заполнены правильно и без ошибок, чтобы избежать проблем в будущем. Пароли должны быть одинаковы в обоих полях, чтобы мы могли быть уверены, что мы вводим правильный пароль. Кроме того, приложение проверяет, чтобы пароль был не менее трех символов, но не более пятнадцати символов, и что все формы были заполнены. Если все условия выполняются, то мы можем нажать на кнопку «Создать», и наш новый аккаунт будет добавлен в базу данных приложения. Кроме кнопки «Создать», на окне регистрации мы можем найти еще одну кнопку – «Уже есть аккаунт». Если мы случайно попали на окно регистрации, когда мы уже имеем аккаунт в приложении, мы можем нажать на эту кнопку, чтобы перейти обратно на окно авторизации. Это обеспечивает более удобный способ перемещения между двумя окнами и сохранение времени и усилий пользователя.

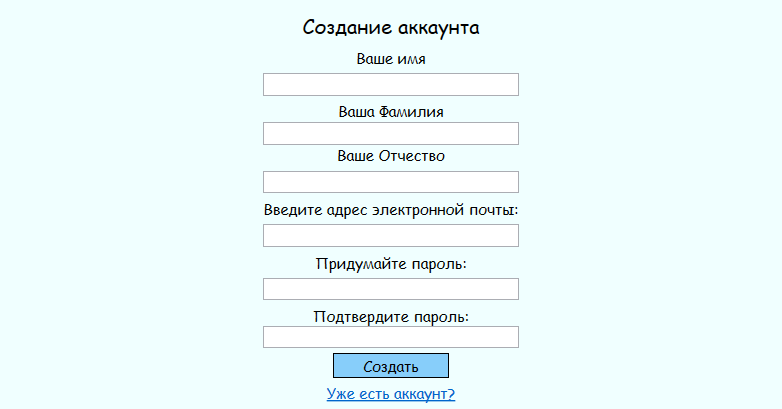


Рисунок 14 – Окно регистрации

См. Приложение А.

Когда пользователь столкнулся с проблемой восстановления пароля, он может использовать кнопку «Забыл(а) пароль» на окне авторизации, чтобы перейти на специальное окно восстановления пароля (см. Рисунок 15).

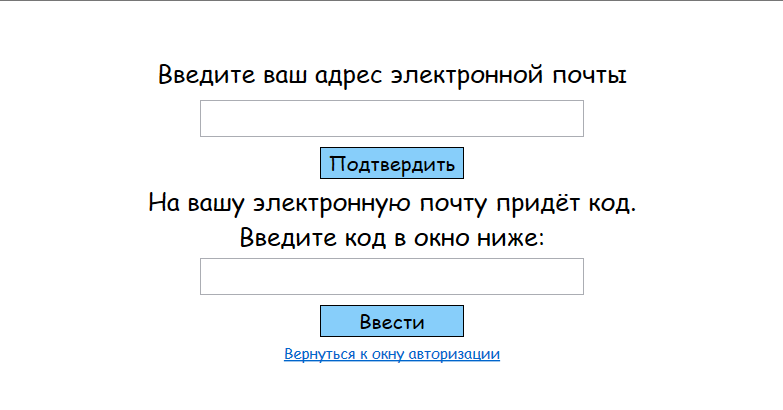


Рисунок 15 – Окно восстановления пароля

Здесь ему предоставляется окно для ввода электронной почты, которую он использовал при регистрации своего аккаунта. Если введённая электронная почта существует в базе данных приложения, то пользователю будет предоставлен специальный код подтверждения (см. Рисунок 16).



Рисунок 16 – Окно с кодом

Для ввода кода пользователю необходимо использовать соответствующее окно ввода кода и нажать кнопку «Ввести». После того, как пользователь ввёл код подтверждения, будет открыто небольшое диалоговое окно, в котором будет указан этот код подтверждения. Этот код можно скопировать и использовать для дальнейшего ввода в соответствующее окно на странице восстановления пароля. Если код подтверждения введён верно, то на экране появится диалоговое окно с паролем от данного аккаунта, который пользователь сможет скопировать и использовать для входа в приложение. Кроме того, под паролем на странице восстановления пароля есть кнопка «Вернуться к окну авторизации», которая позволит пользователю вернуться на окно авторизации и войти в приложение с использованием нового пароля. Таким образом, процесс восстановления пароля в приложении является простым и удобным для пользователей. Он включает в себя несколько шагов, начиная с ввода электронной почты и заканчивая получением нового пароля. Кроме того, на каждом шаге пользователю предоставляется возможность вернуться к предыдущему окну, если он решил изменить своё решение.

После успешной авторизации пользователя в системе и ввода правильных данных, происходит переход на главное окно приложения (см. Рисунок 17), которое представляет собой основное рабочее пространство пользователя.

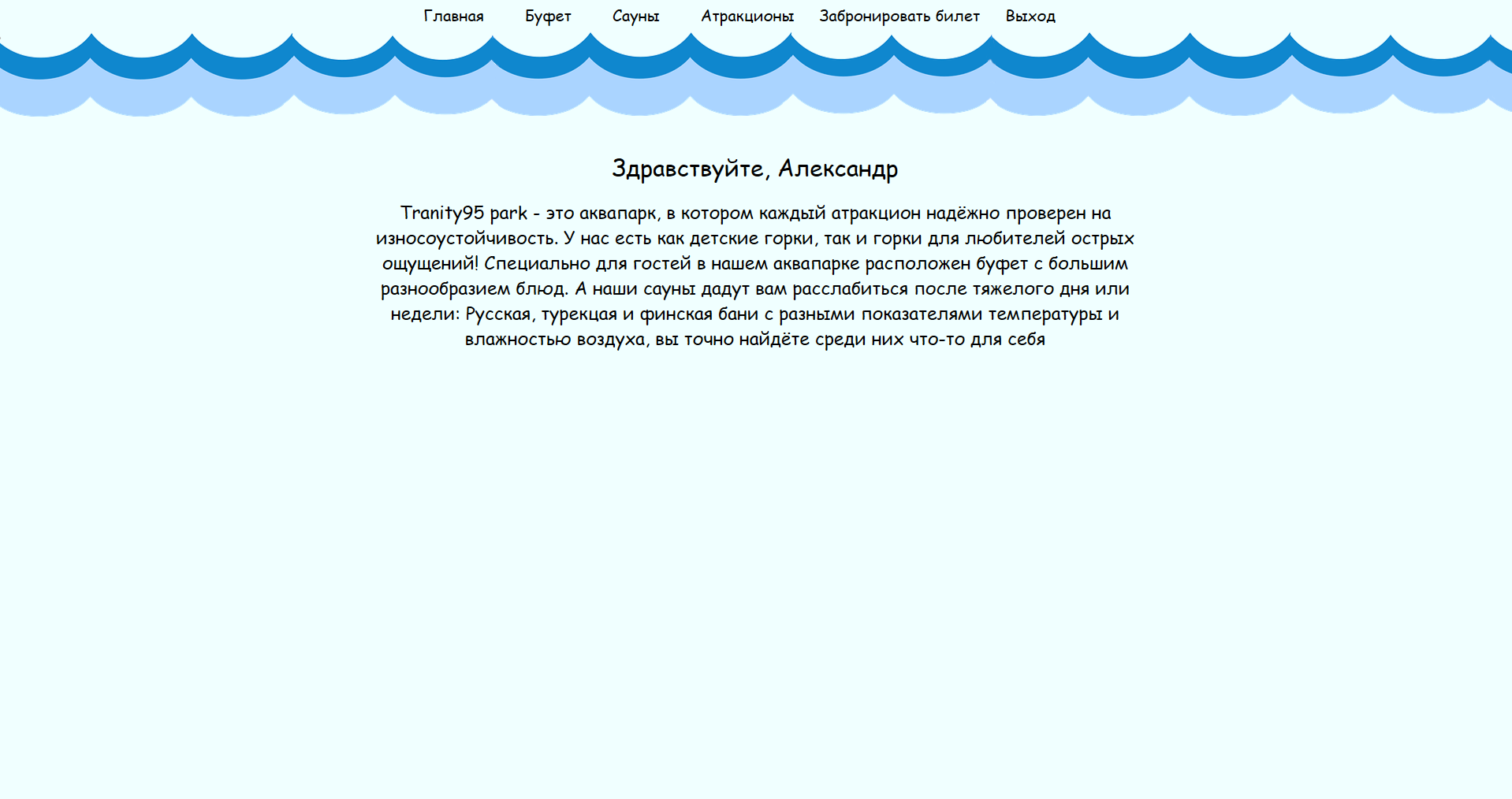


Рисунок 17 – Главное окно

Главное окно имеет удобный и интуитивно понятный интерфейс, который позволяет пользователям легко и быстро получить необходимую информацию о том, что представляет собой аквапарк. На главном окне расположена краткая информация об аквапарке, которая включает в себя описание его основных преимуществ и возможностей. Это позволяет пользователям ознакомиться с тем, что представляет собой аквапарк и какие услуги он предлагает. Сверху на главном окне расположены кнопки управления, которые позволяют пользователям переходить между различными разделами приложения. Например, кнопка «Буфет» позволяет пользователю перейти к списку блюд и напитков, которые можно заказать в кафе аквапарка, а кнопка «Сауны» переносит пользователя в список саун, где можно забронировать время. Кроме того, на главном окне имеется кнопка «Выход», которая позволяет пользователям выйти из приложения и закрыть сессию. Кнопка «Купить билет» позволяет пользователю перейти к разделу бронирования билетов, где можно выбрать наиболее подходящий вариант и произвести оплату. В целом, главное окно является центральным элементом приложения, который предоставляет пользователям быстрый и удобный доступ ко всем основным разделам и функциям приложения.

При нажатии на кнопку «Буфет», пользователь открывает для себя страницу (см. Рисунок 18), которая дает ему возможность ознакомиться со всем ассортиментом еды, которую можно заказать в аквапарке.

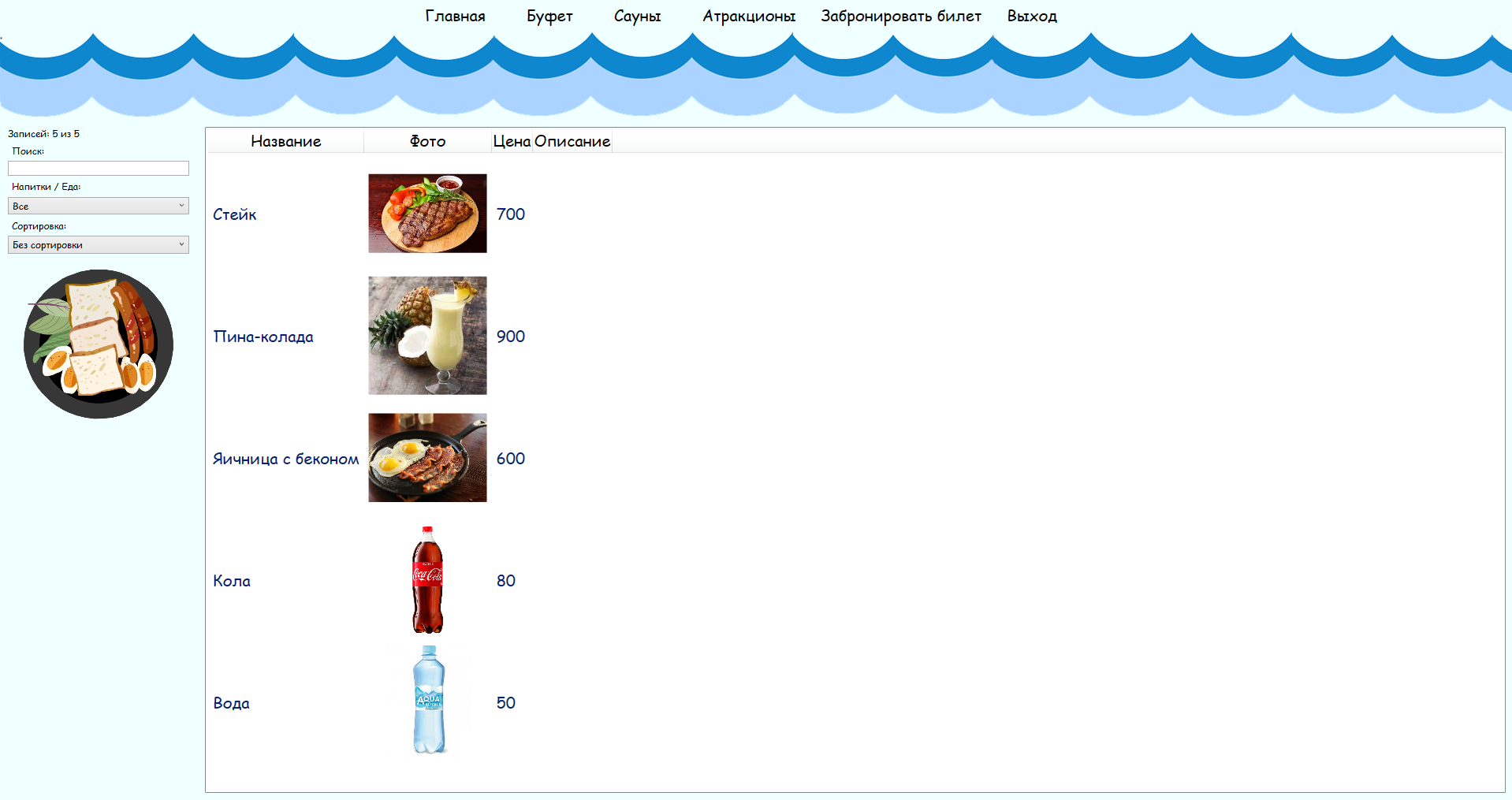


Рисунок 18 – Страница с блюдами буфета

Вся информация, которая относится к блюдам, находится на странице. Список блюд содержит название каждого блюда, цену, фотографию и состав. Важно отметить, что на странице расположены два выпадающих списка, которые позволяют пользователям сортировать блюда по цене и типу (например, еда или напитки). Кроме того, на странице есть окно поиска, которое помогает пользователям найти нужное блюдо, если они знают его название. Если пользователь является администратором, то он получает дополнительные привилегии и может удалить любое блюдо из базы данных, отредактировать уже существующее блюдо или добавить новое блюдо в специальном окне (см. Рисунок 19). В окне добавления нового блюда, которое доступно только администраторам, есть различные поля для ввода информации о блюде, такие как название, цена и тип блюда. Также администратор может импортировать картинку блюда, чтобы сделать его описание более привлекательным для пользователей. Важно отметить, что размер картинки не должен превышать 400 на 300 пикселей.

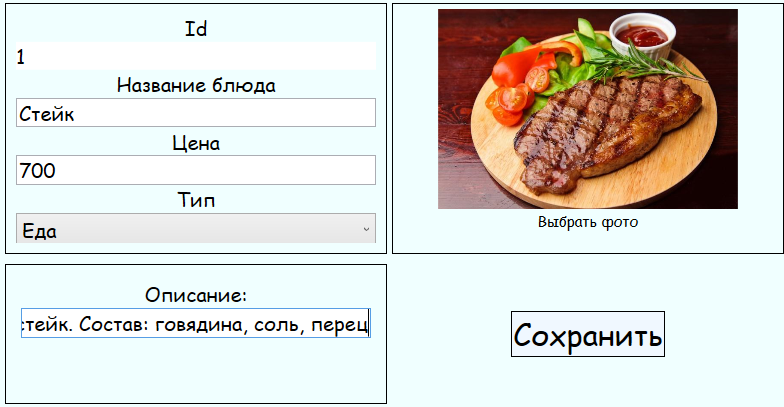


Рисунок 19 – Окно редактирования блюд

См. Приложение Б

При нажатии на кнопку «Аттракционы» пользователь попадает на страницу, где представлены все доступные в аквапарке аттракционы. Здесь пользователь может выбрать, какой аттракцион ему хочется попробовать, узнать его название, уровень страха, а также увидеть его фотографию и краткое описание. На странице с аттракционами также присутствуют функции поиска и сортировки, которые могут помочь пользователю найти нужный аттракцион по его параметрам. Функция поиска позволяет быстро и легко найти аттракцион, который интересует пользователя. А сортировка по цене помогает пользователям отсортировать все доступные аттракционы по цене в порядке возрастания или убывания. Если пользователь является администратором, то ему доступны дополнительные возможности. Он может удалять любой аттракцион, который больше не предоставляется в аквапарке, а также отредактировать информацию о существующем аттракционе. Кроме того, администратор может добавить новый аттракцион и внести необходимую информацию о нем в специальное окно, где можно указать название аттракциона, его уровень страха, цену и прикрепить к нему фотографию.

При нажатии на кнопку «Сауны» пользователь переходит на окно со списком всех саун аквапарка, аналогично с предыдущими двумя, на окне присутствует список саун с их названиями, температурой, влажностью, фотографией и ценой. Администратор всё также может редактировать и удалять существующие сауны и добавлять новые.

При нажатии на кнопку «Забронировать билет» пользователь перенаправляется на страницу бронирования билета (см. Рисунок 20), где он может забронировать место в аквапарке.

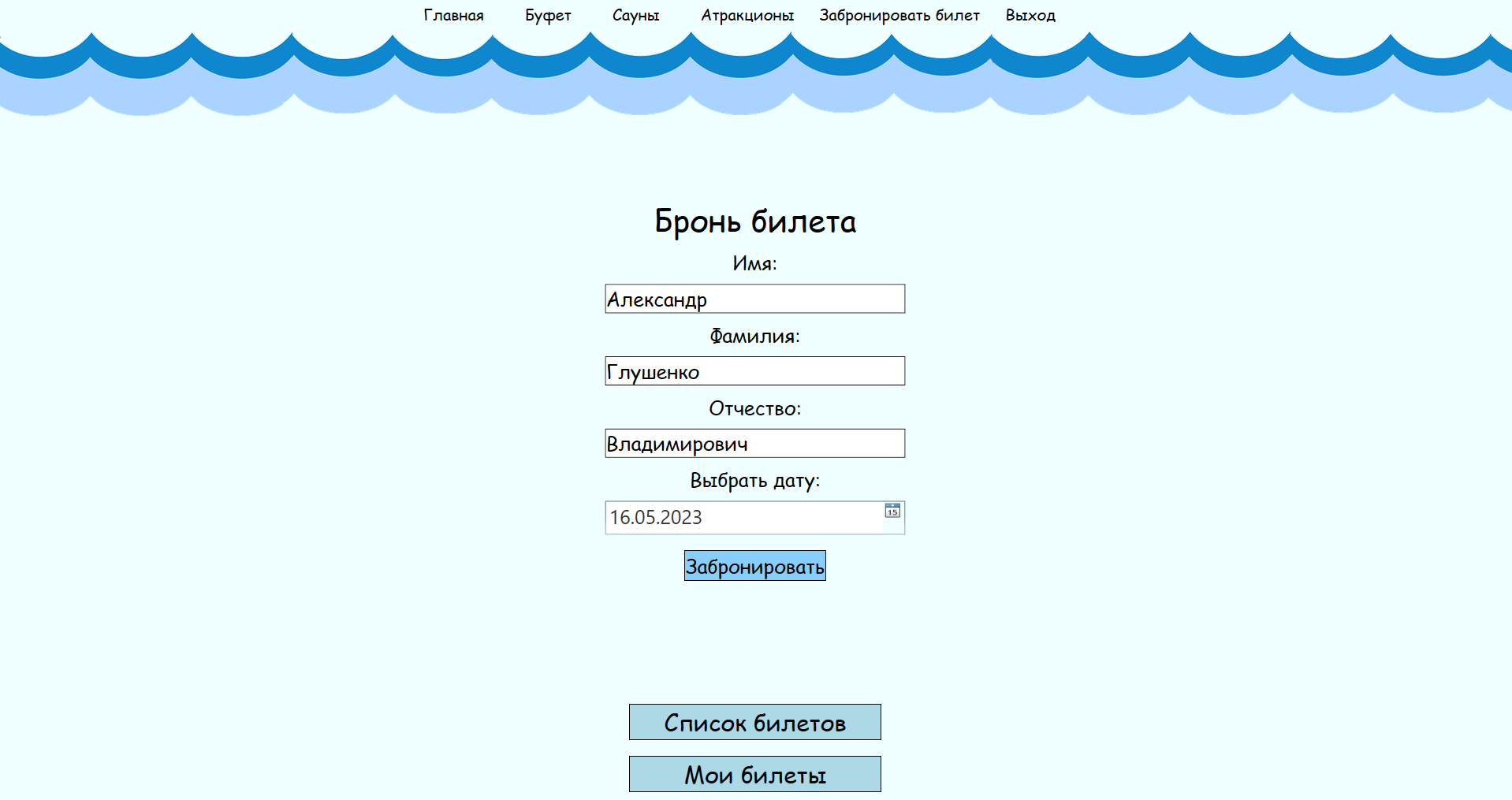


Рисунок 20 – Окно брони билета

Окно бронирования билета содержит три поля ввода, которые автоматически заполняются информацией о пользователе (имя, фамилия и отчество). Ниже находится окно ввода даты бронирования, где пользователь может указать желаемую дату посещения аквапарка. После ввода необходимой информации, пользователь может нажать кнопку «Забронировать», чтобы добавить новую бронь в базу данных. Кроме того, на странице бронирования билета также присутствуют две кнопки внизу страницы: «Мои билеты» и «Список билетов». Нажатие на первую кнопку позволяет пользователю просмотреть список всех его забронированных билетов, а вторая кнопка доступна только для администраторов и перенаправляет их на страницу со списком всех забронированных билетов всех пользователей. Таким образом, процесс бронирования билетов в аквапарке существенно упрощается благодаря данной функции, которая позволяет пользователям легко и удобно забронировать билеты на желаемую дату.

На окне со всеми билетами (см. Рисунок 21) есть поиск и сортировка по дате, а также возможность удалить любую бронь.

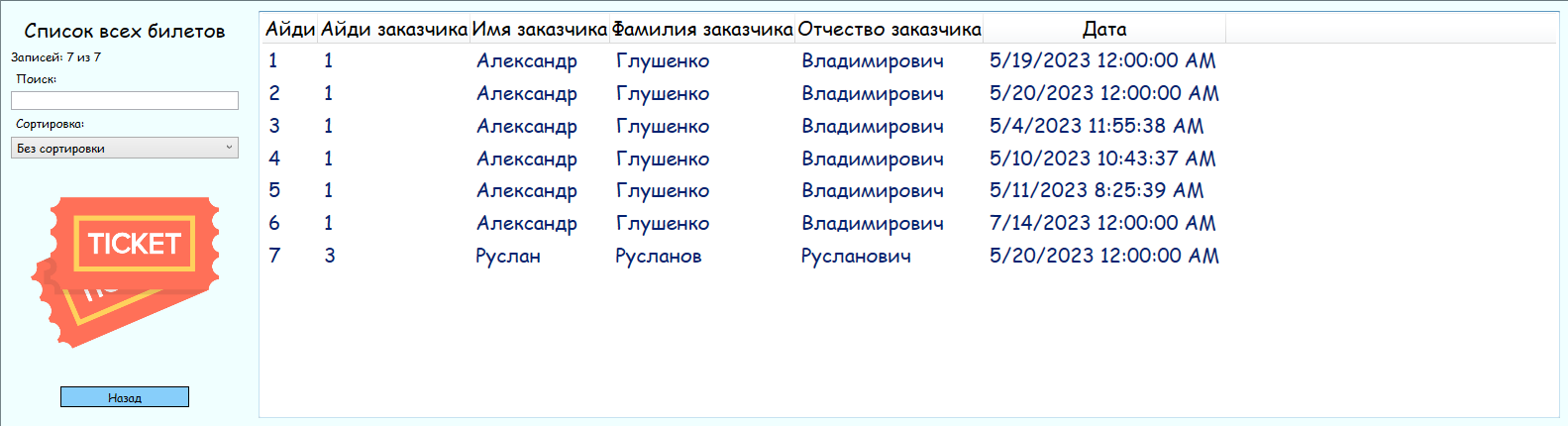


Рисунок 21 – Окно со списком билетов

При нажатии на кнопку "Мои билеты" пользователь переходит на страницу со списком всех его забронированных билетов (см. Рисунок 22). Эта страница содержит информацию о каждом билете, такую как его уникальный идентификатор (ID) и дата бронирования. Вверху страницы находятся окна поиска и сортировки по дате, которые позволяют пользователю быстро находить нужные билеты. Кроме того, на странице есть возможность удалить любую бронь, если пользователь передумал или по какой-то другой причине не может посетить аквапарк в запланированное время.

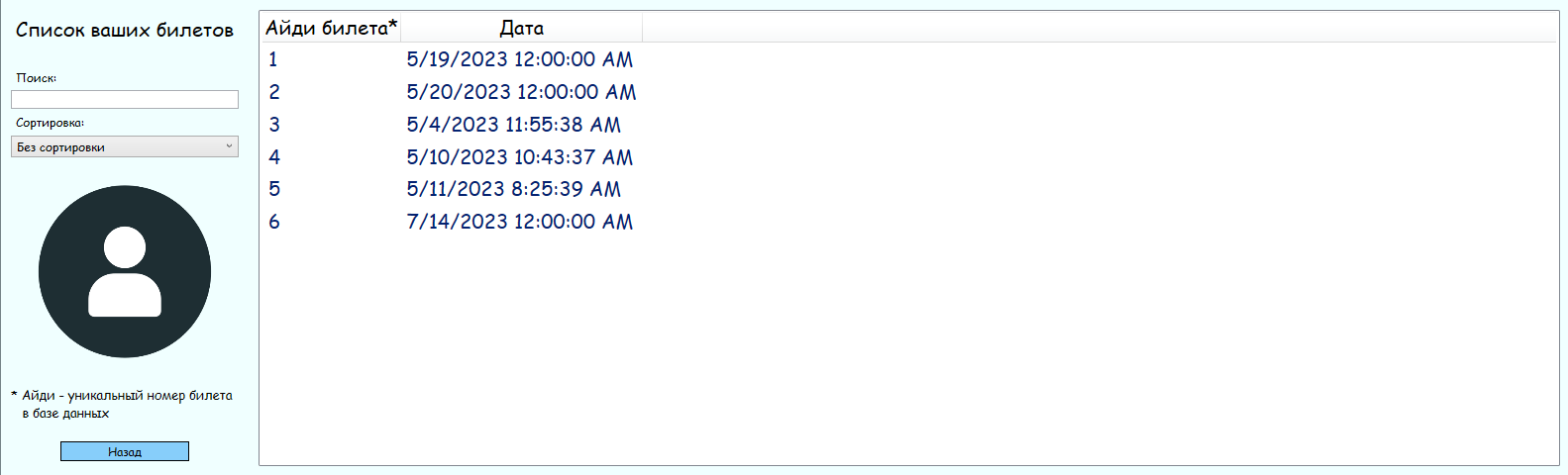


Рисунок 22 – Окно со списком билетов пользователя

Для этого необходимо нажать правой кнопкой мыши по билету, выбрать пункт «Удалить бронь» и подтвердить свой выбор. В целом, страница "Все билеты" предоставляет администраторам удобный и простой способ управления билетами на аквапарк. Она облегчает процесс поиска и удаления бронированных билетов, потому как даты броней с истёкшим сроком давности написаны красным цветом.

При использовании функции "Выход" на странице пользователь сможет завершить свою сессию в аккаунте и перейти на страницу авторизации. На странице авторизации ему придется ввести свой электронный адрес и пароль, чтобы снова войти в систему.

**2.2. Пользовательские сценарии**

Пользователь – неавторизованный клиент без аккаунта, конечной целью которого является забронировать билет на день после текущего (см. Рисунок 23).

1. Создать аккаунт.
   1. Перейти на окно регистрации.
   2. Ввести свои данные и нажать кнопку «Подтвердить».
2. Войти в аккаунт.
   1. Перейти на окно авторизации.
   2. Ввести почту и пароль и нажать кнопку «Войти».
3. Забронировать билет.
   1. На главной странице перейти в раздел «Забронировать билет».
   2. Выбрать завтрашнюю дату и нажать кнопку «Забронировать».

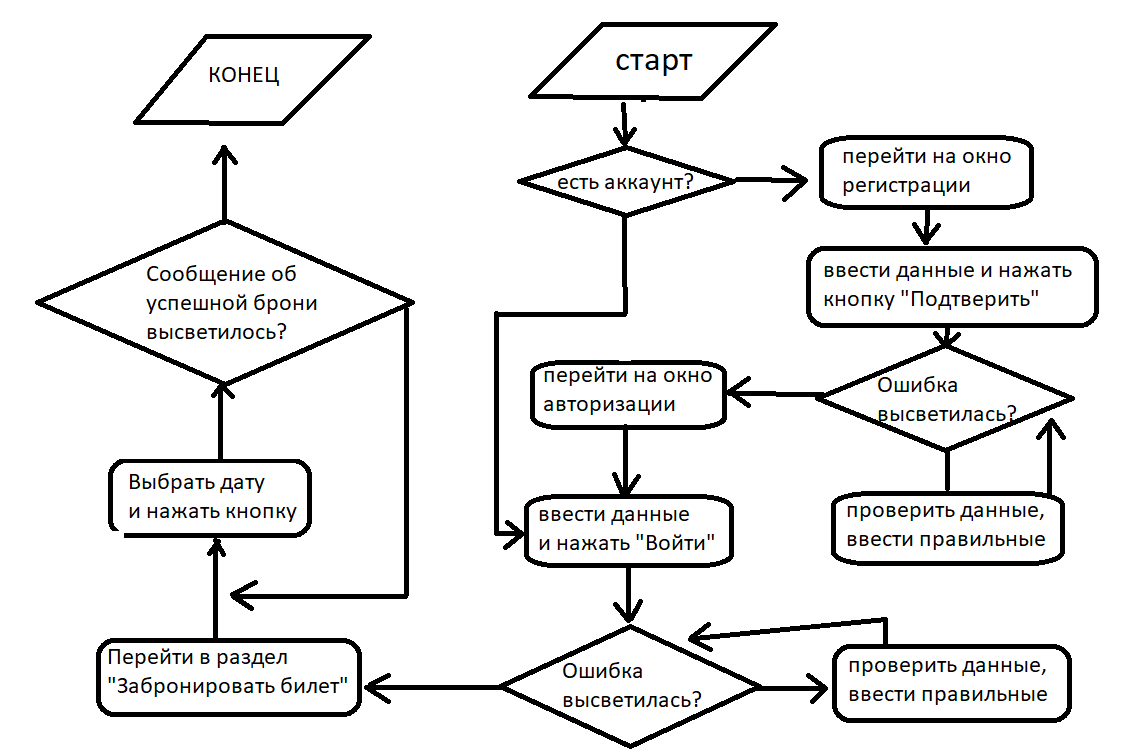


Рисунок 23 – Схема создания и входа в аккаунт

Пользователь – администратор с аккаунтом, конечной целью которого является удалить все брони с истёкшим сроком давности (см. Рисунок 24).

1) Войти в аккаунт

а) Ввести почту и пароль и нажать кнопку «Войти»

2) Удалить брони

а) Перейти в раздел «Забронировать билет»

б) Перейти в раздел «Все билеты»

в) Если текст даты красного цвета – она истекла, значит удалить.

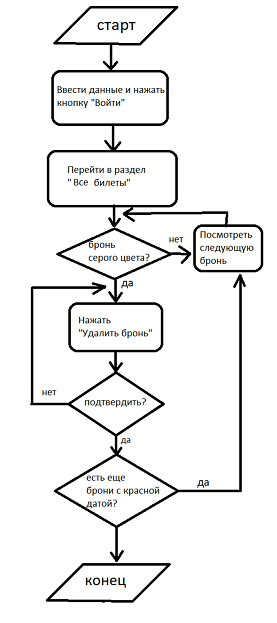


Рисунок 24 – Схема удаления истекших броней

**ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы было создано полноценное приложение для управления аквапарком, которое предоставляет администраторам удобный и эффективный инструмент для контроля и управления различными аспектами работы аквапарка. Приложение позволяет сократить время и усилия, затрачиваемые на рутинные операции, и повысить уровень обслуживания посетителей аквапарка. Таким образом, разработанное приложение является важным шагом в совершенствовании и автоматизации процессов управления аквапарком, способствуя более эффективной работе администраторов и повышению удовлетворенности посетителей. В дальнейшем, приложение может быть доработано и расширено с учетом новых потребностей и технологических возможностей, что позволит еще более оптимизировать и улучшить функционирование аквапарка.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

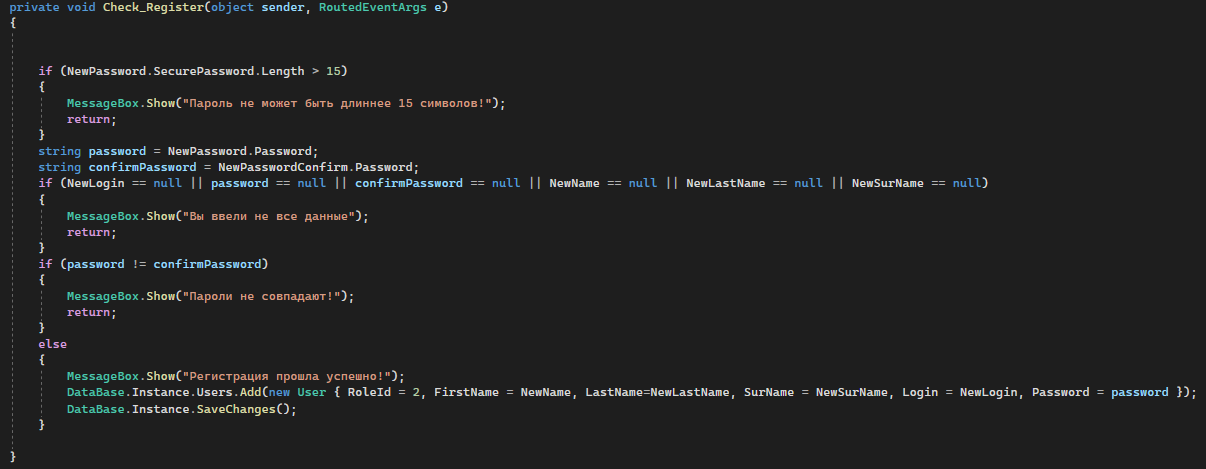
**Учебники и учебные пособия**

1. Гаурав Арораа, Джефри Чилберто – Паттерны проектирования для C# и платформы .Net Core, 2021.
2. Эндрю Лок – Asp.Net Core в действии, 2021.
3. Прайс Марк – C# 10 и .Net 6. Современная кросс-платформенная разработка, 2023.

**Ресурсы сети Интернет**

1. OpenAI GPT-3.5 (Generative Pre-trained Transformer 3.5), URL: https://openai.com/blog/chatgpt, 10.05.2023
2. Что такое Use case? Теория и примеры URL: https://testengineer.ru/chto-takoe-use-case, 11.05.2023
3. 100 лучших книг, URL: http://www.100bestbooks.ru, 11.05.2023
4. Сайт о программировании, URL: https://metanit.com/, 12.05.2023
5. Модели базы данных, URL: https://www.internet-technologies.ru, 14.05.2023
6. Базовые понятия ERD, URL: https://nsportal.ru/shkola/informatika-i-ikt/library/2014/12/22/bazovye-ponyatiya-erd, 14.05.2023
7. Чем отличаются базы данных, СУБД, URL: https://vc.ru/u/1321790-servermall/540774-chem-otlichayutsya-bazy-dannyh-subd-i-sistemy-baz-dannyh, 15.05.2023

**ПРИЛОЖЕНИЕ А  
Регистрация**

****

**ПРИЛОЖЕНИЕ Б  
Редактирование и удаление блюда**

****