**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**Рязанский государственный радиотехнический университет им. В.Ф. Уткина**

**Кафедра электронных вычислительных машин**

|  |  |
| --- | --- |
|  | **К ЗАЩИТЕ**  руководитель КП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.С. Дудко  «\_\_»\_\_\_\_\_\_\_\_\_ 2023 г. |

**КУРСОВОЙ ПРОЕКТ**

по дисциплине

**«Программирование клиентских приложений»**

Выполнил студент группы 940

Большаков И.А \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата сдачи на проверку, подпись

Руководитель проекта

Дудко И. С. \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка дата защиты, подпись

Рязань 2023

ЗАДАНИЕ

на курсовой проект по дисциплине   
«Программирование клиентских приложений»

Исполнитель: Большаков И. А.

Тема: Разработка клиентской части информационной системы. Предметная область: «Экопарк»

Краткое описание предметной области:

Экопарк – туристический объект, предоставляющий услуги отдыха. Система предназначена для обработки информации о гостях, их взаимодействиях с системой, оказанных услугах. Есть возможность забронировать частный дом. Дом обладает площадью, количеством кроватей, вместимостью. Каждый дом убирается уборщиком. Уборщик определяется во время определения смены. Гости могут забронировать место. Одновременное количество броней может быть больше одной. Данные фиксируются в виде ФИО того, кто забронировал, время бронирования, время заезда и выезда, а так - же сами объекты брони. Каждая бронь имеет уникальный номер. Еще на объекте есть так называемые термы. Термы представляют из себя комплекс услуг с бассейнами, саунами и парными. Все данные о посещении записываются, и при покидании территории производится итоговая стоимость посещения экопарка.

**Требования к проекту**

**Основная цель**: разработать клиентское приложение для информационной системы.

**Основные задачи**:

1. провести анализ требований к клиентской части ИС;
2. разработать спецификацию требований;
3. подготовить комплект документов необходимый для реализации информационной системы;
4. разработать прототип интерфейса;
5. разработать дерево форм;
6. реализовать интерфейс пользователя на языке программирования высокого уровня;
7. провести тестирование клиентской части ИС;
8. сформировать программную документацию к ИС;
9. разработать установочный пакет ИС.

**Требования к реализации клиентской части**:

* техническое задание не должно противоречить построенным на основе предметной области моделям;
* спецификация требований должна включать все выявленные ограничения;
* интерфейс должен включать формы для всех действующих ролей;
* клиентская часть должна быть реализована на языках C# или Java с помощью IDE открытого доступа;
* клиентская часть должна содержать формы, работающие в отсоединенном режиме;
* программная документация должна включать руководства пользователя и программиста;
* установочный пакет должен содержать все необходимые ресурсы для запуска ИС;
* разработанная ИС должна быть заполнена информацией достаточной для проверки работоспособности всех функций системы.

**К защите представить**:

- работоспособное ПО, установленное на компьютере в аудитории 210, 209 или 122;

- пояснительную записку в распечатанном виде;

- дистрибутив ИС (сценарий создания БД, инсталлятор клиентской части и т.д.); дополнительное ПО, использованное при разработке; исходные тексты программ.

Требования к содержанию пояснительной записки:

Титульный лист

Задание

Содержание

Введение

1. Разработка технического задания
   1. Анализ требований к клиентской части ИС
   2. Цели создания ИС
   3. Характеристики ИС
   4. Спецификация требований к ИС
2. Проектирование интерфейса пользователя
   1. Разработка дерева форм
   2. Разработка прототипа интерфейса пользователя
3. Реализация клиентской части информационной системы
   1. Выбор средств реализации
   2. Организация взаимодействия клиентской программы с БД
   3. Разработка интерфейса пользователя
4. Разработка установочного пакета
5. Разработка программной документации

Заключение

Список использованных источников

Приложение

Дата выдачи задания: «9» февраля 2023 г.

Срок сдачи проекта: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Подпись исполнителя: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись руководителя: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

[**Введение** 5](#_Toc131393346)

[**1 Разработка технического задания** 7](#_Toc131393347)

[**1.1 Анализ требований к клиентской части ИС** 7](#_Toc131393348)

[**1.2 Цели создания ИС** 8](#_Toc131393349)

[**1.3 Характеристики ИС** 8](#_Toc131393350)

[**1.4 Спецификация требований к ИС** 8](#_Toc131393351)

[**2 Проектирование интерфейса пользователя** 10](#_Toc131393352)

[**2.1 Разработка дерева форм** 10](#_Toc131393353)

[**2.2 Разработка прототипа интерфейса пользователя** 12](#_Toc131393354)

[**2 Реализация клиентской части информационной системы** 19](#_Toc131393355)

[**2.1 Выбор средств реализации** 19](#_Toc131393356)

[**2.2 Организация взаимодействия клиентской программы с БД** 19](#_Toc131393357)

[**2.3 Разработка интерфейса пользователя** 19](#_Toc131393358)

[**3 Разработка установочного пакета** 29](#_Toc131393359)

[**4 Разработка программной документации** 31](#_Toc131393360)

[**Заключение** 36](#_Toc131393361)

[**Список использованных источников** 36](#_Toc131393362)

[**Приложение** 36](#_Toc131393363)

**Введение**

В наши дни активно развивается сфера информационных технологий, благодаря чему область её применения расширяется на все области человеческой деятельности.

Одной из наиболее активно развивающихся областей компьютерных технологий, является автоматизация обработки информации на основе программно-вычислительных комплексов информационных систем, которые пользуются большими объемами данных, хранящимися во внешней памяти. Такие хранилища имеют сложную структуру, в основании которой лежит такой объект, как таблица. Автоматизация практически всех видов деятельности строится по принципу реляционной базы данных, что позволяет сократить финансовые и ресурсоемкие затраты, что, безусловно, является актуальной задачей в настоящее время.

Клиент-сервер – это архитектура информационных систем, в которой часть функций предоставлены клиенту, а обработка, хранение, выдача информации происходит на сервере, что позволяет строго разделить работу информационной системы на так называемые слои.

 Клиент-серверная информационная система состоит из трех основных слове:

* cервер баз данных с предустановленной на нем системой управления базами данных, или СУБД, управляющей хранением данных, доступом и защитой, резервным копированием, отслеживающей целостность данных и выполняющая запросы клиента;
* клиент, выполняющий логику приложения, посылающий запросы к серверу и получающий ответы от него;
* Сеть и коммуникационное программное обеспечение, которое осуществляет взаимодействие между клиентом и сервером посредством сетевых протоколов;

Рассматривая вопрос о том, какие преимущества имеет архитектура клиент-сервер, стоит сказать, что создание таких приложений позволяет добиться ряда важнейших факторов:

Сохранность информации;

Устойчивость к сбоям;

Масштабируемость;

Большая защищенность информации от несанкционированного доступа;

Цель курсовой работы – разработка клиент-серверного приложения на базе платформы .NET с использованием языка C#, технологий доступа к данным и использованием СУБД MySql.

Вначале идет анализ и определение требований к ИС, в частности к клиентской части.

Во второй части идет определение внешнего вида клиентской части, в том числе разработка дерева форм и прототипирование пользовательских интерфейсов.

Третья часть представляет из себя разработку полного клиентского приложения.

Четвертая и пятая части предполагают разработку дистрибутива и документации.

В конце подведены основные итоги проекта.

**1 Разработка технического задания**

* 1. **Анализ требований к клиентской части ИС**

**Описание предметной области**

Экопарк – туристический объект, предоставляющий услуги отдыха. Система предназначена для обработки информации о гостях, их взаимодействиях с системой, оказанных услугах. Есть возможность забронировать частный дом. Дом обладает площадью, количеством кроватей, вместимостью. Каждый дом убирается уборщиком. Уборщик определяется во время определения смены. Гости могут забронировать место. Одновременное количество броней может быть больше одной. Данные фиксируются в виде ФИО того, кто забронировал, время бронирования, время заезда и выезда, а так - же сами объекты брони. Каждая бронь имеет уникальный номер. Еще на объекте есть так называемые термы. Термы представляют из себя комплекс услуг с бассейнами, саунами и парными. Все данные о посещении записываются, и при покидании территории производится итоговая стоимость посещения экопарка.

Исходя из описания предметной области, необходимо разработать клиентское приложение, которое будет работать с информационной системой.

Приложение будет представлять из себя классическое настольное приложение, поставляемое через дистрибутив на настольные ПК будущих пользователей. Так как ИС имеет разные входные данные, различные функции и т. п, необходимо разработать функционал разделения ответственности, иными словами регистрация и аунтефикация. Таким образом, можно составить список требований:

* приложение должно быть настольным;
* приложение должно распостраняться через дистрибутив;
* предусмотреть работу для пользователей различных ролей;
* приложение должно иметь графический интерфейс, иметь формы для ввода входных данных и элементы для отображения выходных данных;
* приложение должно уметь обращаться к удаленным хранилищам данных, получать, обрабатывать, сохранять и изменять данные необходимым образом;
  1. **Цели создания ИС**

Основными целями создания ИС являются автоматизация рутинных процессов при работе с данной предметной областью, хранение, обработка и выдача информации пользователям, выполнение их требований на высшем, безошибочном уровне. За счет автоматизации процесса обработки данных сводится на «нет» вероятность человеческой ошибки в процессе работы. Появится возможность быстрого выполнения определенных действий. Основными процессами, подвергнувшимся автоматизации, являются:

* Возможность бронирования мест пользователями;
* Возможность вносить данные об услугах, которыми воспользовались пользователями;
* Расчет стоимости посещения экопарка;
* Создание смен рабочих;
  1. **Характеристики ИС**

Основным предназначением информационной системы является обработка пользовательских данных и структуризация информации о деятельности экопарка.

* 1. **Спецификация требований к ИС**

**Общее описание**

Приложение будет представлять из себя классическое настольное приложение, поставляемое через дистрибутив на настольные ПК будущих пользователей. Приложение будет работать под управлением ОС Windows. Минимальная версия ОС – 7.

**Требования к пользователям системы**

Основные пользователи ИС:

* Обычный пользователь;
* Сотрудник рецепшена;
* Администратор;

**Требования к функциям системы**

* Возможность бронирования мест пользователями;
* Просмотр своих броней;
* Возможность вносить данные об услугах, которыми воспользовались пользователями;
* Создание смен рабочих;
* добавление сотрудников;
* Бронирование мест сотрудниками;
* Добавление информации об услугах;

**Требования к качеству системы**

* **Удобство пользования** должно быть на высоком уровне. Интуитивное расположение элементов управления, удобные и понятные формы.
* **Высокая производительность**. Не использовать «тяжелых» технологий, максимально часто использовать асинхронную модель поведения, максимально отзывчивое приложение.
* **Надежность**. Приложение должно обрабатывать ошибки, оповещать об этом пользователя и сообщать, что ему делать. Не должно экстренно закрываться, подвергая данные риску.
* **Безопасность**. Приложение должно быть безопасным как с точки зрения пользовательских данных, так и данных ОС. Не допускать их повреждения. Не допускать утечек данных, не должно быть уязвимостей в безопасности.

1. **Проектирование интерфейса пользователя**
   1. **Разработка дерева форм**

На рисунке 1 представлено дерево форм, разработанное в ходе анализа требований.



Рисунок 1 – Дерево форм

Заставка приложения – форма с изображением, появляется при запуске приложения.

Окно авторизации – форма, в которой пользователь осуществляет вход в систему с помощью логина и пароля. В зависимости от типа пользователя, открывается одна из трех основных форм.

Окно регистрации – позволяет зарегистрироваться пользователю.

Форма пользователя – форма, содержащая информацию и функционал, доступный для пользователя, а именно: отмена, добавление, редактирование броней; просмотр своих броней; просмотр истории посещений.

Форма добавления/редактирования броней – форма, позволяющая взаимодействовать с бронями.

Форма просмотра броней – просмотр своих броней.

Форма просмотра посещений – форма, позволяющая посмотреть историю посещений.

Форма сотрудника рецепшена – форма, предназначенная для персонала. Позволяет просмотреть клиентов, просматривать, добавлять/удалять брони, фиксировать оказанные услуги, производить итоговый расчет.

Форма просмотра клиентов – форма, позволяющая просмотреть информацию о клиентах.

Форма добавления услуги в посещение – форма, позволяющая добавить услугу в выбранное посещение.

Форма администратора – форма, содержащая информацию и функционал, доступный для администратора, а именно: просмотр, добавление, редактирование сотрудников; просмотр, добавление, редактирование смен; возможность добавить новую услугу в общий список.

Форма добавления/редактирования сотрудника – форма, позволяющая изменять информацию о сотрудниках.

Форма добавления/редактирования смены – форма, позволяющая изменять информацию о сменах.

Форма добавления услуги – форма, позволяющая добавить новую услугу в общий список.

* 1. **Разработка прототипа интерфейса пользователя**

Форма входа (рисунок 2).

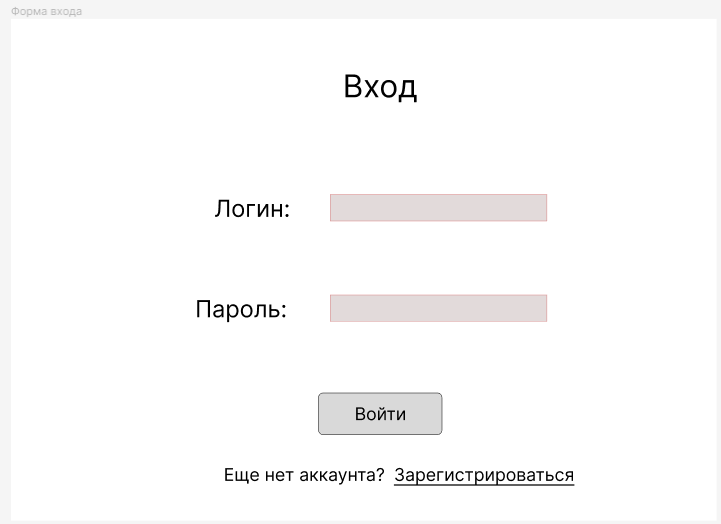


Рисунок 2 – Форма входа

Форма регистрации (рисунок 3).

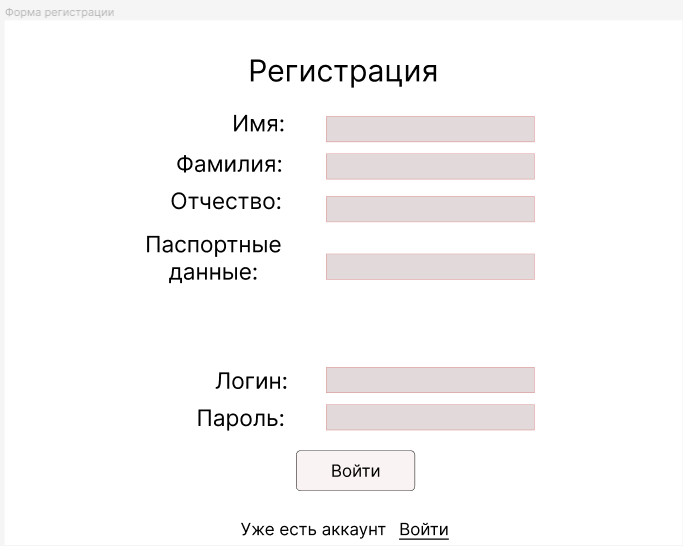


Рисунок 3 – Форма регистрации

Форма пользователя просмотра броней (рисунок 4).

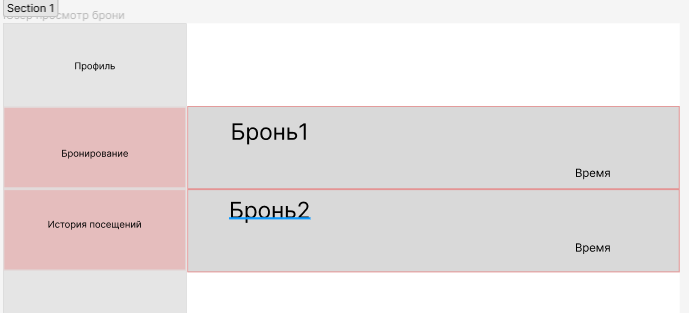


Рисунок 4 – Форма пользователя просмотра броней

Форма пользователя добавления/редактирования бронирования (рисунок 5).

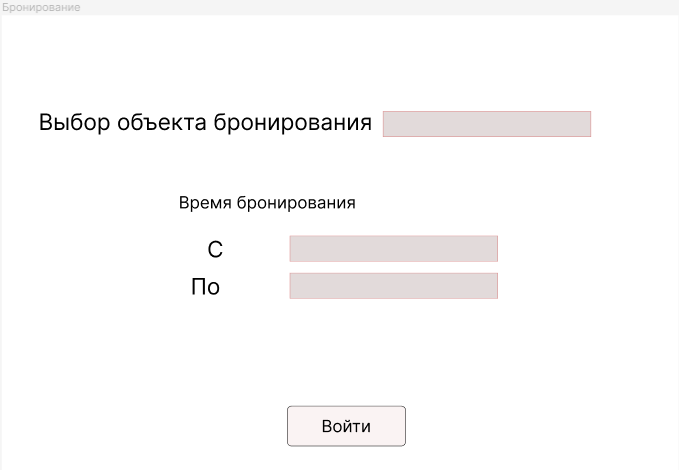


Рисунок 5 – Форма пользователя добавления/редактирования бронирования

Форма просмотра посещений пользователя (рисунок 6).

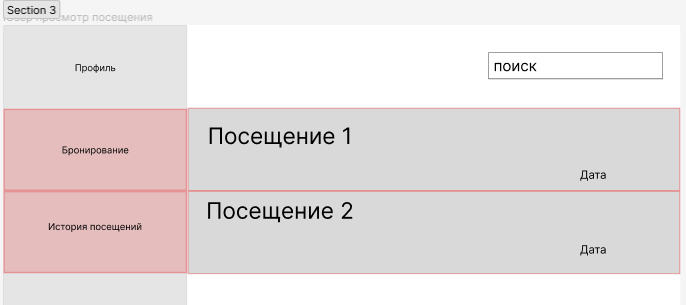


Рисунок 6 – Форма просмотра посещений пользователя

Форма работника просмотра посещений (рисунок 7).

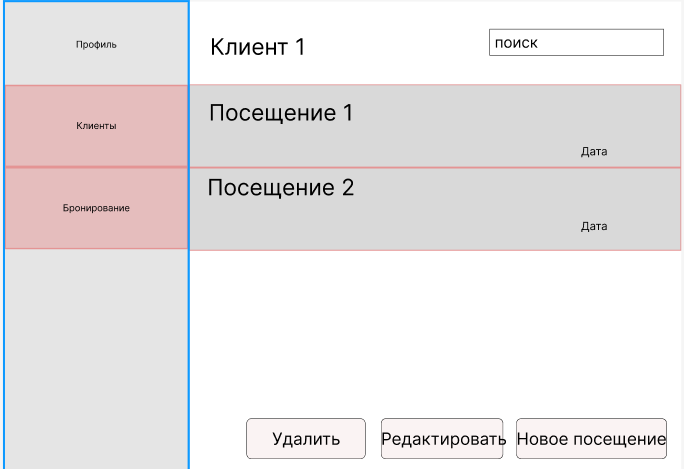


Рисунок 7 – Форма работника просмотра посещений

Форма просмотра посещений (рисунок 8).

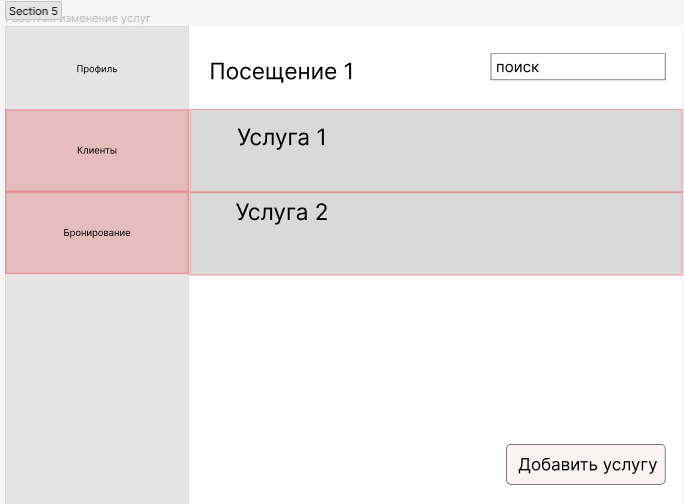


Рисунок 8 – Форма просмотра посещений

Форма добавления услуг в посещение (рисунок 9).

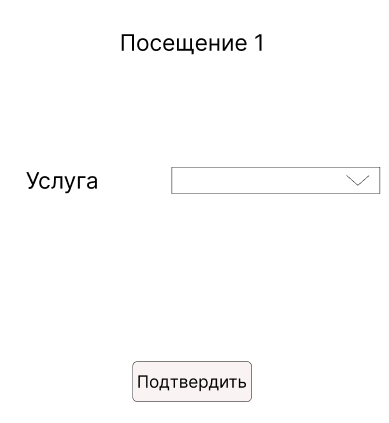


Рисунок 9 – Форма добавления услуг в посещение

Форма просмотра клиентов (рисунок 10).

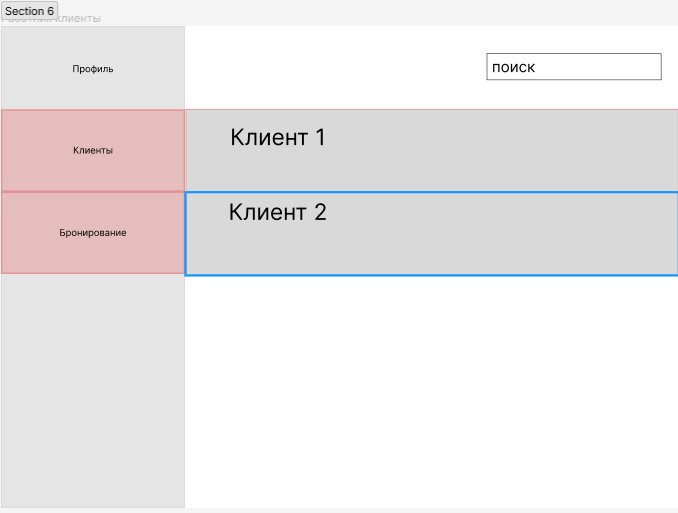


Рисунок 10 – Форма просмотра клиентов

Форма просмотра броней клиента (рисунок 11).

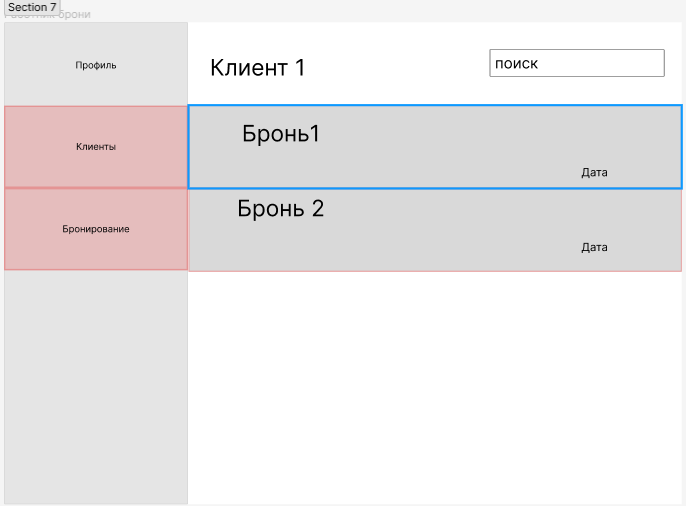


Рисунок 11 – Форма просмотра броней клиента

# **Реализация клиентской части информационной системы**

## **Выбор средств реализации**

В качестве основной платформы для обеспечения работы программы была выбрана платформа .NET, а именно .NET Framework версии 4.5 для совместимости со старыми ПК в режиме совместимости с Windows 7. Язык программирования, соответственно, был выбран из семейства .NET, а именно ЯП C# версии 7.3 (ограничения .NET Framework 4.5). В качестве IDE была использована Visual Studio. Технология, используемая для разработки пользовательского интерфейса – WindowsForms. Средством взаимодействия клиентской части ИС с базой данных выступила технология Entity Framework. Эта технология позволяет работать с СУБД используя объекта языка программирования. Иными словами, это ORM. В качестве СУБД выступила MS SQL Server. И, наконец, в качестве менеджера для взаимодействия данных выступили DBeaver и management studio.

## **Организация взаимодействия клиентской программы с БД**

Средством взаимодействия клиентской части ИС с базой данных выступила технология Entity Framework. Эта технология позволяет работать с СУБД используя объекта языка программирования. Иными словами, это ORM. Сначала была создана база данных на сервере. Затем, при помощи code first reverse engineering подхода, были созданы классы сущностей базы данных в коде языка программирования, а так же контекст взаимодействия с базой данных из кода для выполнения основных операций (добавление, просмотр и т. д). Путем предоставления графического интерфейса пользователю он вводит исходные данные для программы, которые валидируются, затем, при помощи контекста, взаимодействуют тем или иным способом с БД.

## **Разработка интерфейса пользователя**

Используя прототипы графического пользовательского интерфейса, описанные в п. 2.2, были разработаны формы, выполняющие те или иные функции. На рисунке 12 изображена форма входа.

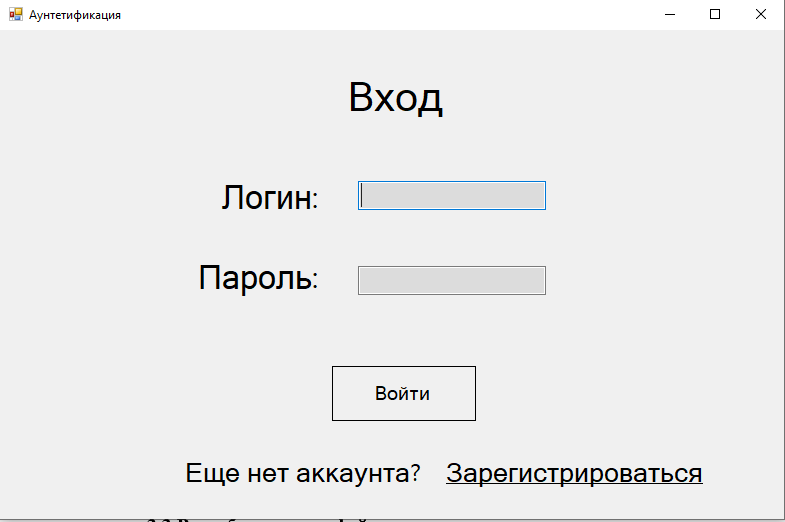


Рисунок 12 – Форма входа

Форма регистрации (Рисунок 13).

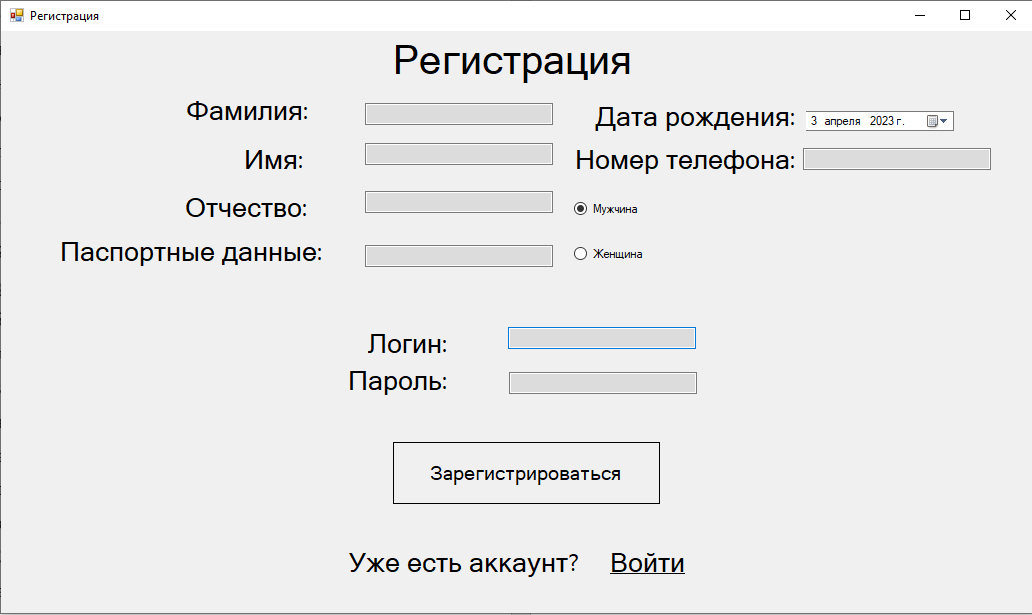


Рисунок 13 – Форма регистрации

Главная форма пользователя (рисунок 14).

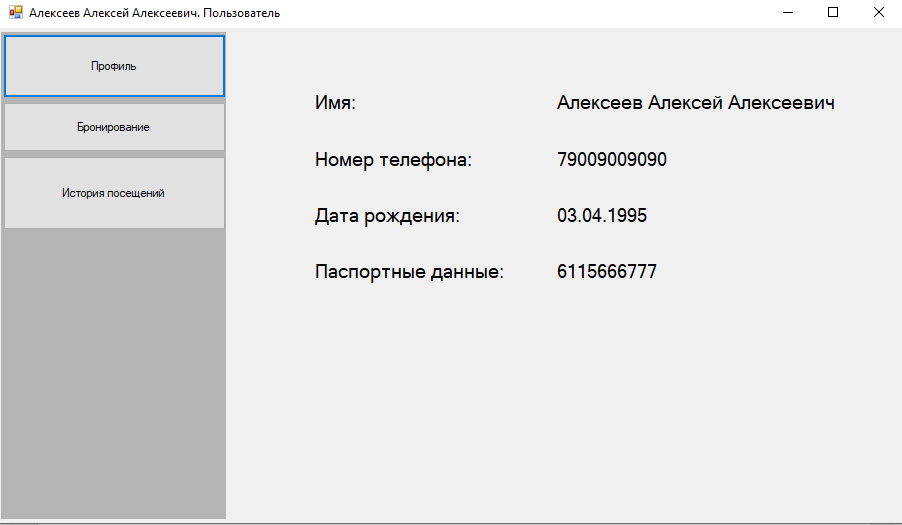


Рисунок 14 – Главная форма пользователя

Форма просмотра броней (рисунок 15).

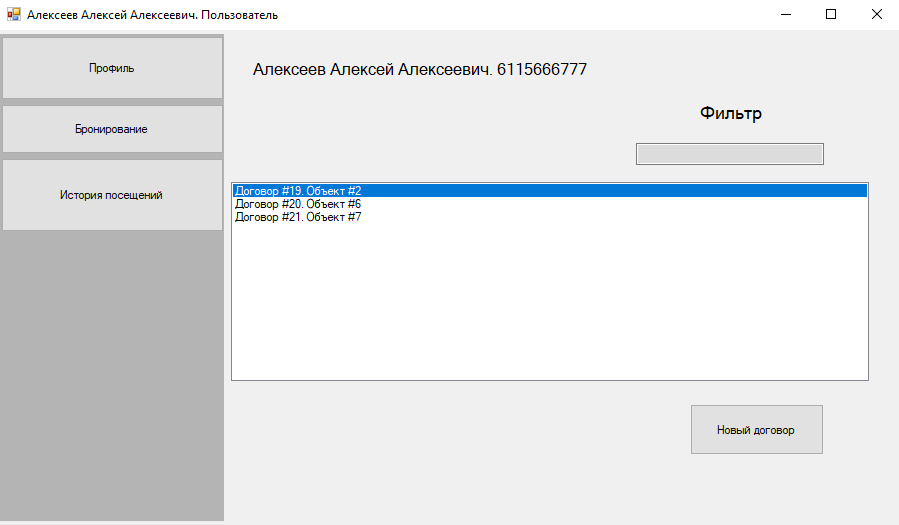


Рисунок 15 – Форма просмотра броней

Форма бронирования (рисунок 16).

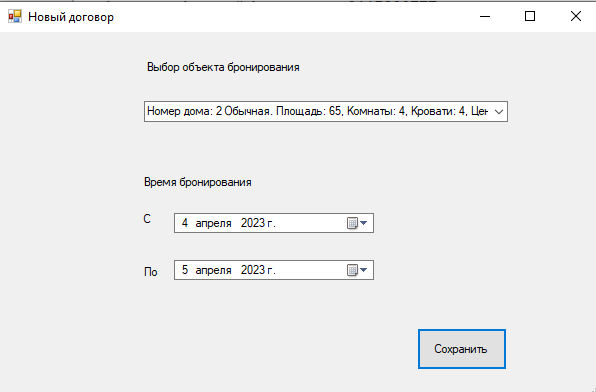


Рисунок 16 – Форма бронирования

Форма просмотра посещений (рисунок 17).

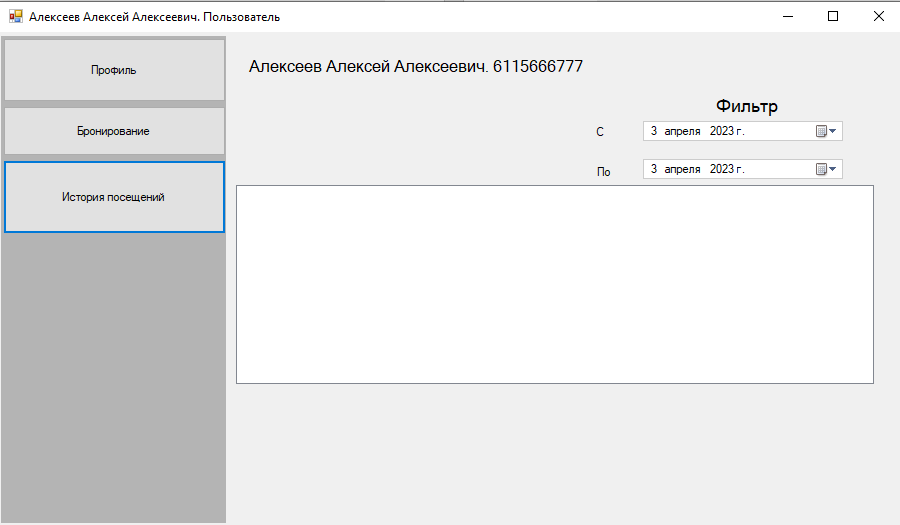


Рисунок 17 – Форма просмотра посещений

Основная форма сотрудника (рисунок 18).

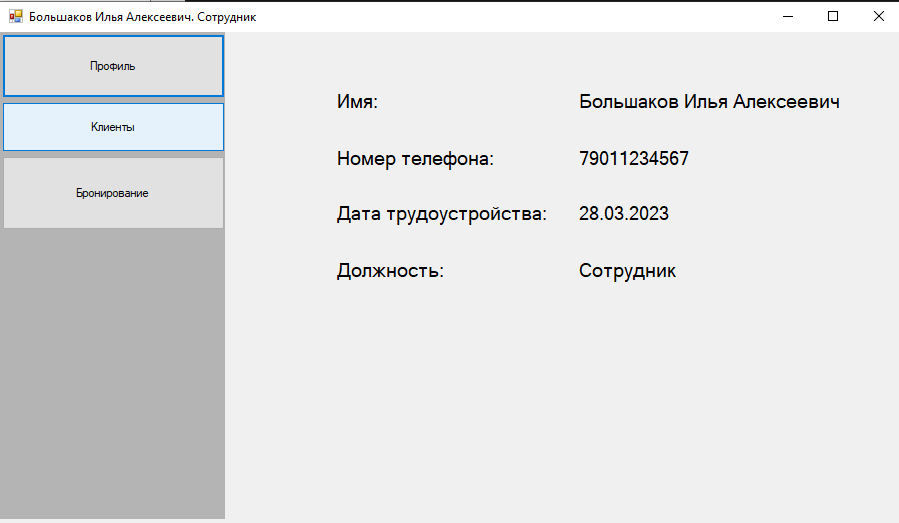


Рисунок 18 – Основная форма сотрудника

Форма просмотра клиентов (рисунок 19).

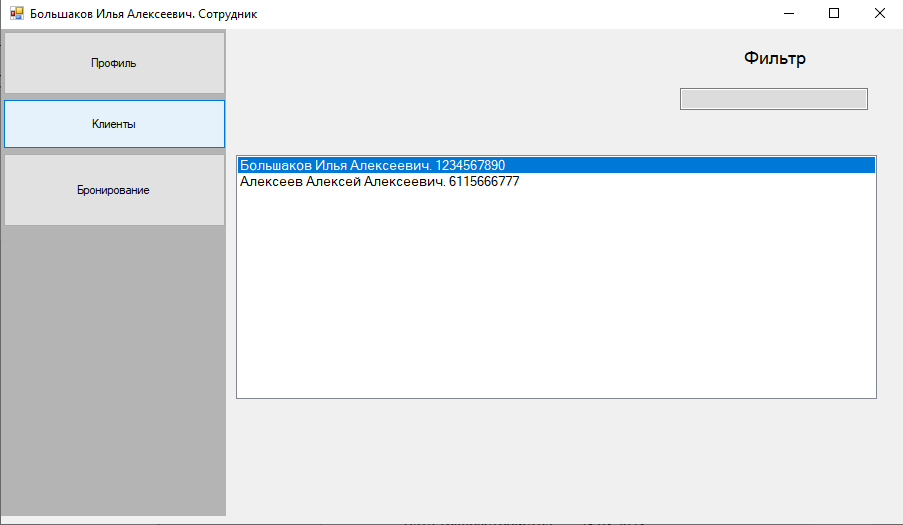


Рисунок 19 – Форма просмотра клиентов

Форма просмотра посещений (рисунок 20).

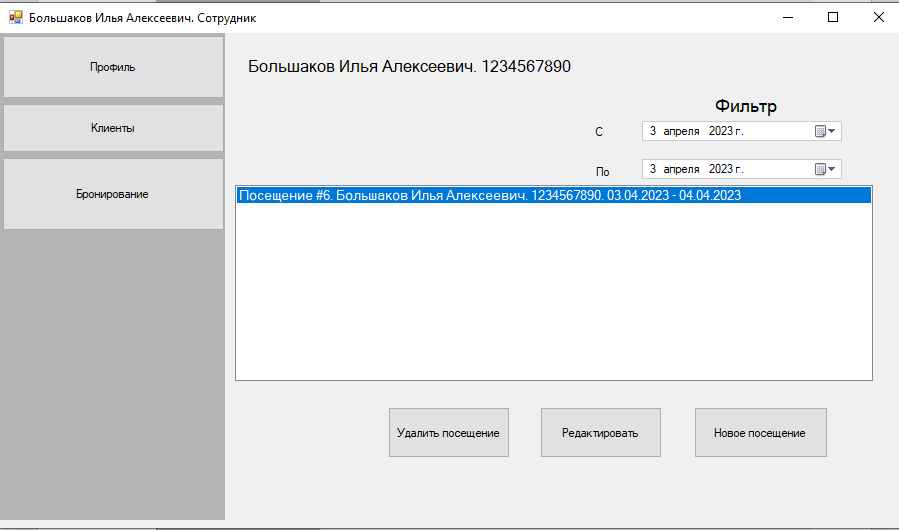


Рисунок 20 – Форма просмотра посещений

Форма добавления нового посещения (рисунок 21).

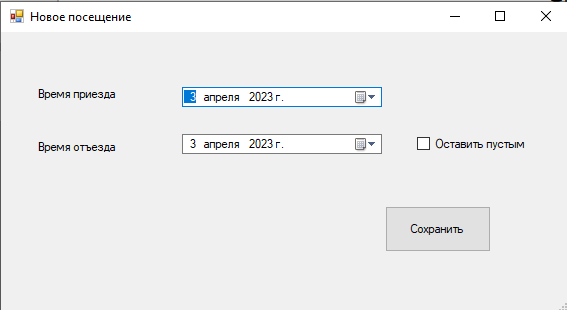


Рисунок 21 – Форма добавления нового посещения

Форма просмотра услуг на посещении (рисунок 22).

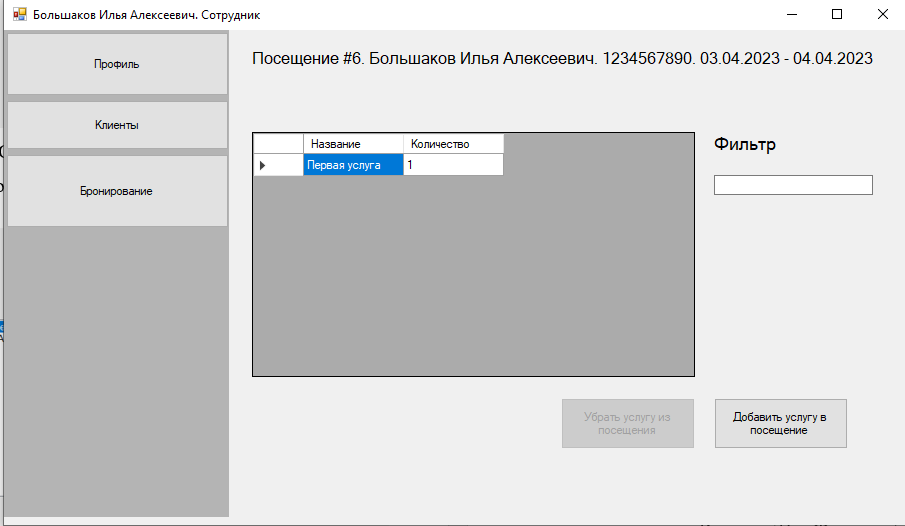


Рисунок 22 – Форма просмотра услуг на посещении

Форма просмотра сотрудников (рисунок 23).

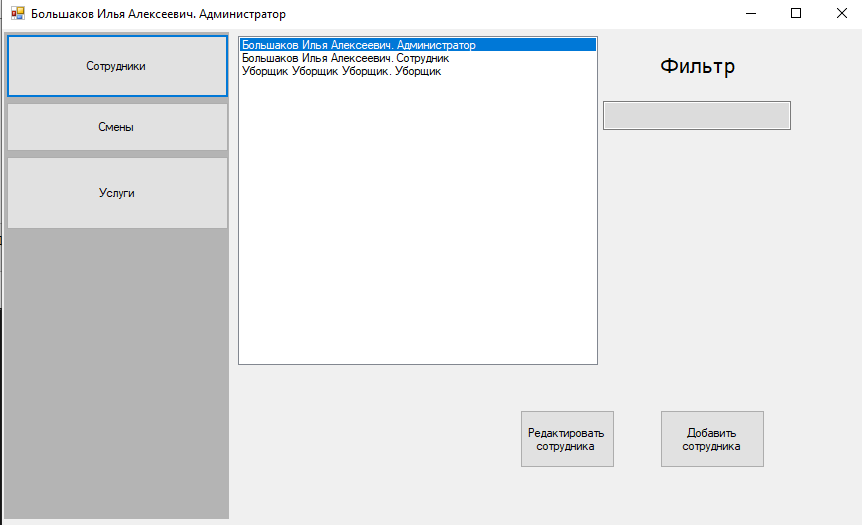


Рисунок 23 – Форма просмотра сотрудников

Форма добавления сотрудников (рисунок 24).

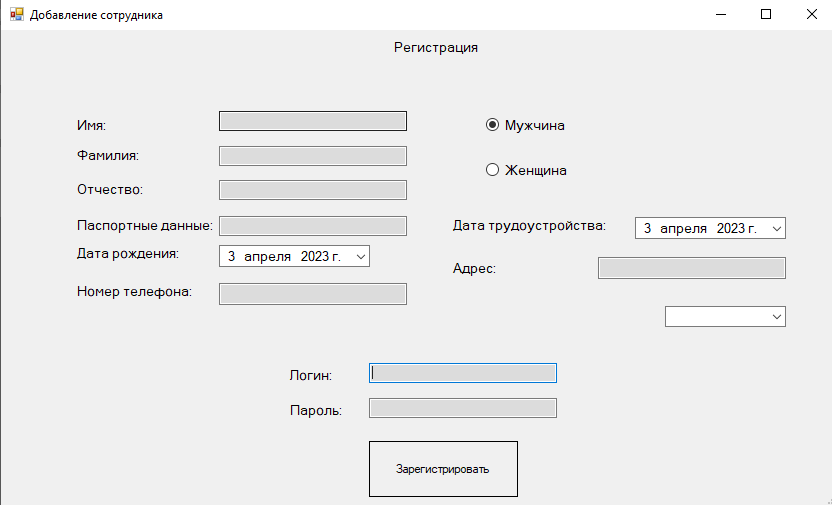


Рисунок 24 – Форма добавления сотрудников

Форма просмотра смен (рисунок 25).

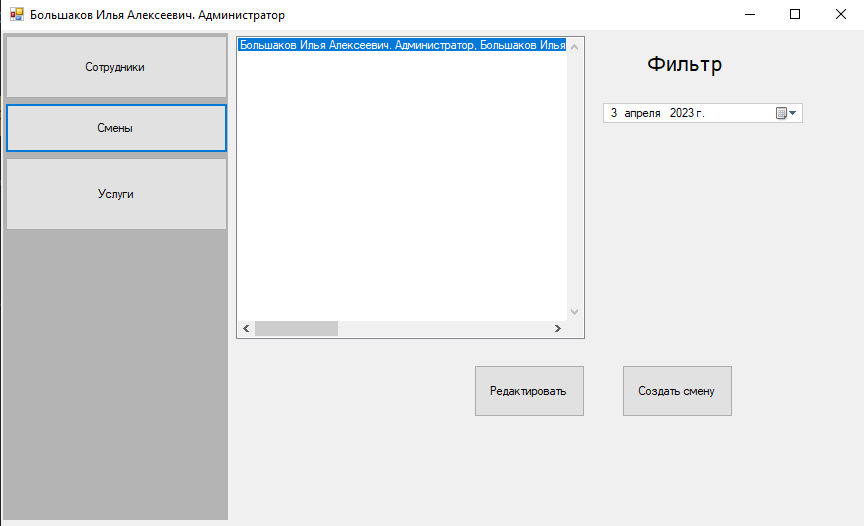


Рисунок 25 – Форма просмотра смен

Форма добавления смен (рисунок 26).

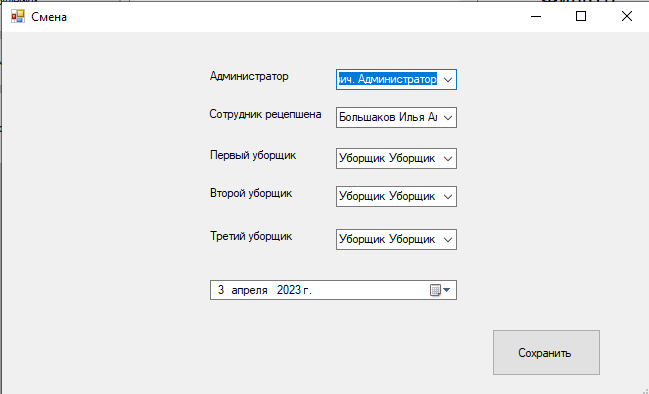


Рисунок 26 – Форма добавления смен

Форма просмотра услуг (рисунок 27).

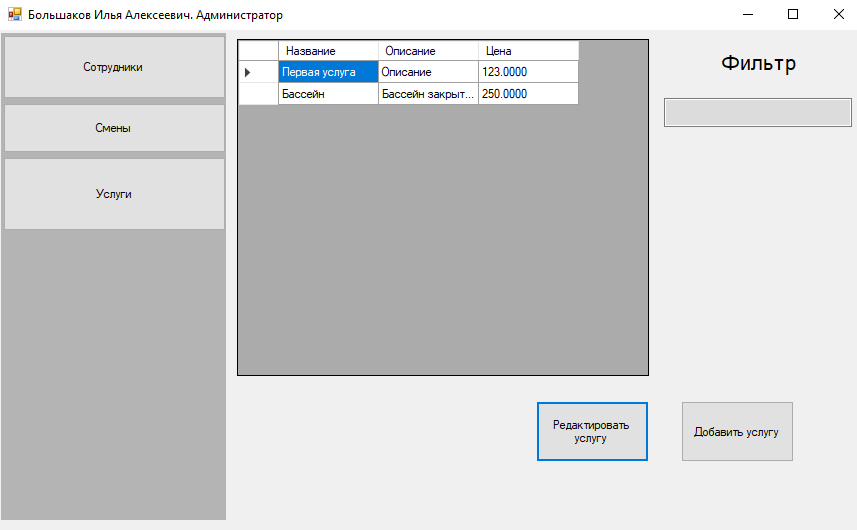


Рисунок 27 – Форма просмотра услуг

Форма добавления услуг (рисунок 28).

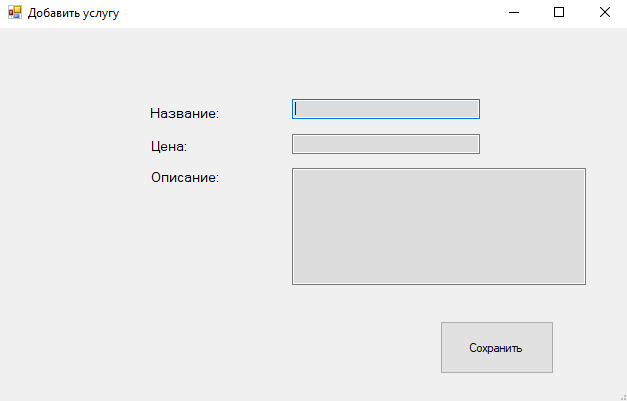


Рисунок 28 – Форма добавления услуг

# **Разработка установочного пакета**

Для создания установочного пакета воспользуемся расширением Microsoft Visual Studio Installer Project. Данное ПО является расширением для IDE Visual Studio.

Далее создадим проект установщика (рисунок 29).

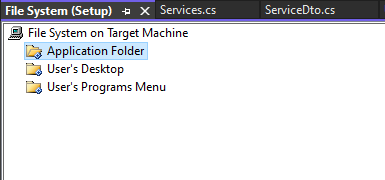


Рисунок 29 – Создание проекта установщика

Добавим выходные файлы (рисунок 30).

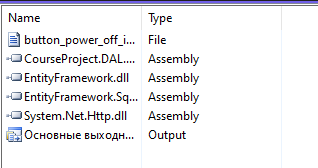


Рисунок 30 – Выходные файлы проекта

После остальной настройки главных параметров соберем проект, и запустим установщик (рисунок 31 - 33).

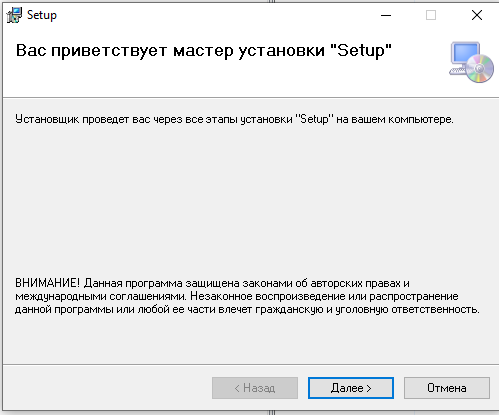


Рисунок 31 – Инсталлятор

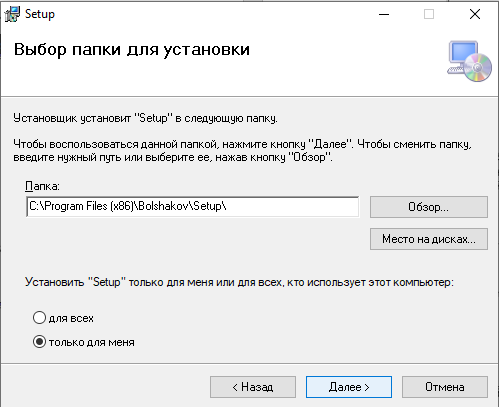


Рисунок 32 – Выбор каталога для установки

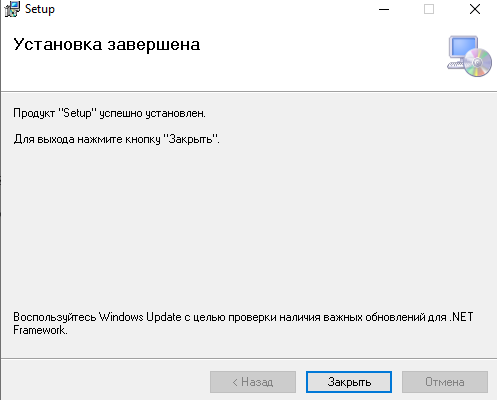


Рисунок 33 – Успешная установка

# **Разработка программной документации**

**Руководство программиста**

**Назначение и условия применения программы**

Программа CourseProject является клиентским приложением для взаимодействия с БД и обеспечением работы с информационной системой.

Условия применения программы:

* Объем ОЗУ: 4 GB;
* процессор Intel Pentium 4;
* любое видеоядро;
* экран с разрешением не менее 800x600, 256 цветов;
* операционная система: Windows 7;
* клавиатура;
* мышь.

**Характеристики программы**

Разработанная прикладная программа имеет следующие характеристики:

* Название исполняемого файла – CourseProject.exe;
* размер исполняемого файла – 200 КБ;
* версия файла – 1.0.0;
* название продукта – CourseProject;
* язык – русский.

**Обращение к программе**

Установка программы осуществляется при помощи инсталлятора Setup.exe (см. п. 4). После успешной инсталляции нужно перейти в каталог установленного приложения и открыть исполняемый файл CourseProject.exe или же с рабочего стола запустить исполняемый файл CourseProject.exe.

**Входные и выходные данные**

Входные данные вводятся при помощи форм, которые представлены в виде графического интерфейса. Выходные данные выводятся в виде списков и таблиц.

**Сообщения**

Сообщение об неизвестной ошибке (рисунок 34).

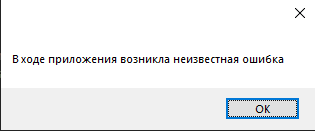


Рисунок 34 - Сообщение об неизвестной ошибке

Ошибка при бронировании (рисунок 35).

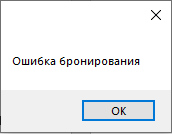


Рисунок 35 – Ошибка при бронировании

Сообщение об некорректном логине или пароле (рисунок 36).

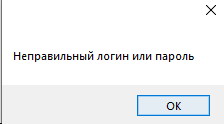


Рисунок 36 – Сообщение об некорректном логине или пароле

Сообщение при попытке отредактировать услугу не выбрав ее (рисунок 37).

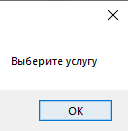


Рисунок 37 Сообщение при попытке отредактировать услугу не выбрав ее

Попытка удалить посещение, в котором имеются услуги (рисунок 38).

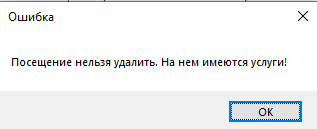


Рисунок 38 – Попытка удалить посещение, в котором имеются услуги

Подтверждение удаления посещения (рисунок 39).

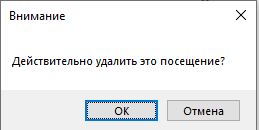


Рисунок 39 - Подтверждение удаления посещения

**Руководство оператора**

**Назначение и условия применения программы**

Программа CourseProject является клиентским приложением для взаимодействия с БД и обеспечением работы с информационной системой.

Условия применения программы:

* Объем ОЗУ: 4 GB;
* процессор Intel Pentium 4;
* любое видеоядро;
* экран с разрешением не менее 800x600, 256 цветов;
* операционная система: Windows 7;
* клавиатура;
* мышь.

**Характеристики программы**

Разработанная прикладная программа имеет следующие характеристики:

* Название исполняемого файла – CourseProject.exe;
* размер исполняемого файла – 200 КБ;
* версия файла – 1.0.0;
* название продукта – CourseProject;
* язык – русский.

**Обращение к программе**

Установка программы осуществляется при помощи инсталлятора Setup.exe (см. п. 4). После успешной инсталляции нужно перейти в каталог установленного приложения и открыть исполняемый файл CourseProject.exe или же с рабочего стола запустить исполняемый файл CourseProject.exe.

**Входные и выходные данные**

Входные данные вводятся при помощи форм, которые представлены в виде графического интерфейса. Выходные данные выводятся в виде списков и таблиц.

**Сообщения**

Сообщения описаны в пункте «Руководство программиста», п.5.

# **Заключение**

В данном курсовом проекте был проведен анализ, выявлена спецификация требований к клиентскому приложению. Было разработано дерево форм, прототипы форм и интерфейсы форм. Было разработано клиентское приложение, взаимодействующее с базой данных и обеспечивающее необходимый функционал, соответствующий требованиям. Была написана документация и разработан установочный пакет.

# **Список использованных источников**

1. Документация по C# [Электронный ресурс] – режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/, свободный. Дата обращения: 03.04.2023.

# **Приложение**

**ПРИЛОЖЕНИЕ А. ССЫЛКА НА РЕПОЗИТОРИЙ С ИСХОДНЫМИ ТЕКСТАМИ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ**

https://github.com/Ilya-Bolshakov/LastCourseProject