

## Содержание

Введение.....	4
Глава 1 Теоретическая часть.....	7
1.1 Предпроектной исследование предметной области .....	7
1.2 Характеристика инструментальных средств разработки.....	10
1.3 Анализ некоторых технологий, используемых при реализации программного комплекса.....	15
1.4 Анализ требований и определение спецификаций программного обеспечения.....	20
Глава 2 Практическая часть .....	24
2.1 Постановка задачи.....	24
2.2 Проектирование программного обеспечения.....	27
2.3 Разработка пользовательских интерфейсов программного обеспечения.....	33
2.4 Тестирование и отладка программного обеспечения.....	39
2.5 Руководство по использованию программного обеспечения.....	41
Заключение .....	60
Список использованных источников .....	62
Приложение А .....	64
Приложение Б.....	65

## Введение

Цель данной выпускной квалификационной работы заключается в разработке автоматизированной информационной системы фирмы по ремонту квартир. В ход выполнения данной работы были приобретены и подтверждены навыки, приобретённые в ходе обучения по специальности «Программирование в компьютерных системах» в Колледже информатики и программирования.

Данная автоматизированная информационная система необходима для автоматизации некоторых процессов в работе небольшой фирмы, выполняющей ремонты квартир

Система будет состоять из приложения под операционную систему Windows, Rest API сервиса, который будет работать на удаленном сервере, что позволит работать нескольким работникам одновременно, а также база данных, с которой будет держать связь Rest API сервис.

Основной задачей будет автоматизация и повышения эффективности работы персонала в рамках определенных задач.

Стоит отметить, данная автоматизация решит вопросы, связанные с количеством времени, которое тратиться на создание заказа на ремонт, а также на некоторые другие технологические операции. После решения данных вопросах сократиться время обслуживания клиентов и повысится качество данного обслуживания, а также повысится обратная отдача между клиентом и теми работниками, которые будут выполнять данный ремонт.

Процесс автоматизации всех процессов в настоящее время с развитием технологий приобрело большой интерес у некоторых директоров и в целом у всего руководящего персонала, который хочет повысить качество и скорость обслуживания клиентов.

Как уже говорилось ранее, цели данного проекта заключается в разработке автоматизации информационной системы фирмы по ремонту

квартир, в котором будет реализован набор всех требуемых для данной программы функций.

Для достижения данной цели передо мной стоят следующие задачи:

- проектирование базы данных как для сервера, так и для локального компьютера;
- созданием универсальных методов для обработки ошибок и работы в рамках работы сервера;
- описание необходимых моделей для работы с клиентским приложением;
- создание контроллеров обработки данных поступающих от клиентского приложения;
- разработка интерфейсов для работы клиентского приложения;
- разработка модуля для загрузки справочных данных на сервер;
- создание необходимых обработок для работы со сторонним сервером;
- создание необходимых шаблонов для вывода документов о смете и договоре по каждому заказу;
- написание пояснительной о программе.

Для решения перечисленных выше задач в процессе работы над созданием мобильной игры необходимо пройти следующие стадии и этапы разработки:

- предпроектное исследование предметной области;
- выбор инструментальных средств разработки программы;
- проектирование игрового процесса;
- разработка пользовательского интерфейса;
- тестирование и отладка программного обеспечения;
- разработка справочной системы;
- разработка руководств по использованию программы.

На каждом этапе либо стадии разработки данной работы будут применяться различные инструментальные средства разработки программного обеспечения:

- Visual Studio 2019 Enterprise;
- draw.io;
- система управления базами данных DB Browser for SQLite;
- сервер базы данных SQL Server Management Studio;
- система управления базами данных SQL Server Management Studio.

## Глава 1 Теоретическая часть

### 1.1 Предпроектное исследование предметной области

#### 1.1.1 Анализ предметной области объекта автоматизации

Данный проект разработан в целях увеличения скорости работы компании и представляет из себя набор клиент серверного программного обеспечения:

- Базы данных для хранения все информации;
- Сервера, который выполняет прослойку между базой данных и прочими программа в данном комплексе;
- Приложения для операционной системы Windows.

Для актуализации понимания о предметной области были изучены программы аналоги и были изучены необходимые и важные функции:

- подсчет стоимости заказа на ремонт;
- создание договора на основе шаблона, созданного заранее;
- импорт информации о материалах и услугах, предоставляемых данной организацией из файла программы Excel;
- возможность добавления, редактирования информации об услугах;
- возможность работы с данными о сотрудниках (прием на работу, увольнение и кадровые перемещения) с документированием в формате Word;
- возможность отдельного доступа к программе путем назначения прав доступа для сотрудников, а также возможность их редактирования;
- учет данных о клиентах и сотрудниках, а также редактирование данных об этом факте;
- учет оказанных услуг и учет платежей, а также система отчетов о проделанных работах и проданных услугах;
- фиксирование замеров помещений и вычисление площади некоторых поверхностей.

### 1.1.2 Актуальность разрабатываемого программного обеспечения

Актуальность данного программного обеспечения заключается в том, что в настоящее время очень маленькое количество компаний занимается разработкой данного специализированного программного обеспечения для автоматизации работы фирм по ремонту квартир, но большое количество компаний требуются в данном программном обеспечении.

Основной причиной актуальности данного программного обеспечения можно считать рост рынка ремонтно-строительных работ, который стабилизировался в условиях нынешних экономических потрясений и состояния экономики России, связанных с ростом количества жилья сданных без отделки квартир относительно квартир сданных с минимальным ремонтом, а также в следствии роста количества вторичного жилья требуемого ремонта в том числе и капитального.

Стоит отметить, что в связи с сложными кризисными временами отделочники-непрофессионалы отсеиваются с рынка быстрее и уступают место специалистам, которые делают качественный ремонт за адекватную цену, что подталкивает людей платить нежелая делать ремонт самостоятельно. [7]

### 1.1.3 Обзор существующих аналогов

В качестве существующих аналогов можно рассмотреть похожую по концепцию программу распространяемую платно по средствам лицензии и разработанную компанией «ПростойСофт»[8], конфигурацию «Ремонт квартир»[9] которая устанавливается на базовую версию их программного обеспечения под название «Учет клиентов».

Данное программно обеспечение имеет следующие основные функции:

- учет клиентов;
- учет событий;
- формирование отчетов;
- хранение справочников;

- формирование документов по шаблонам;
- учет платежей.

Стоит отметить, что данная система является слишком громоздкой и не может работать на нескольких компьютерах одновременно так как база располагается локально на компьютере, таким образом, представляется нецелесообразным использование данной программы невозможно при работе нескольких людей одновременно.

Также стоит отметить, что данное программное обеспечение поставляется за 8000 рублей за одно рабочее место, что не соизмеримо с функционалом при покупке более чем одной копии в компанию.

Также приведу фотографию пользовательского интерфейса которые представлены на рисунке 1

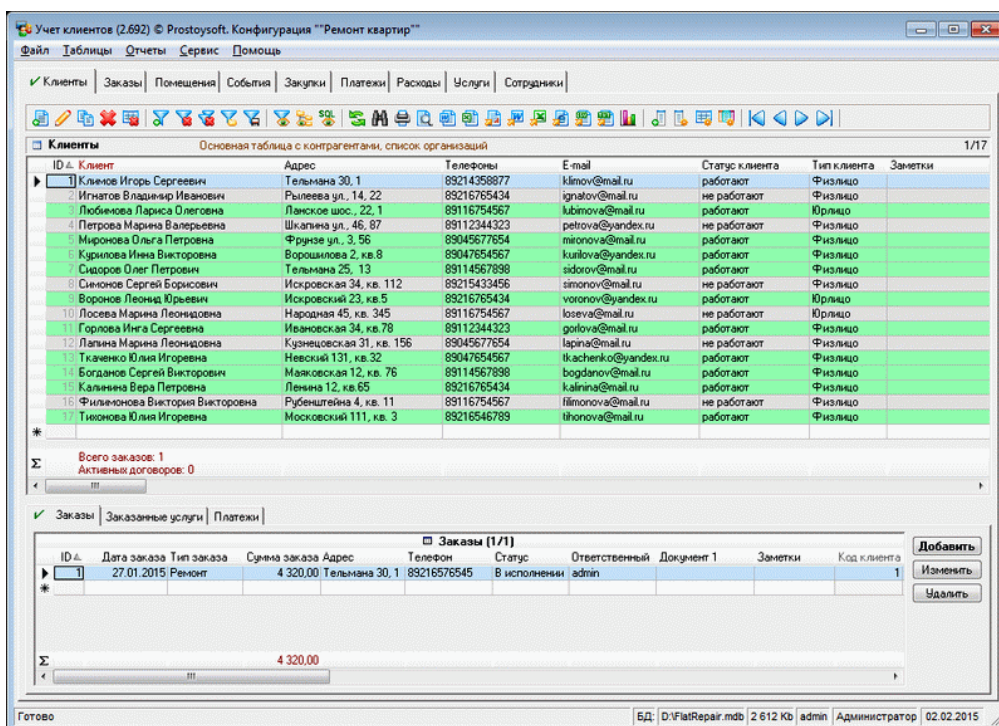


Рисунок 1– Пример пользовательского интерфейса аналога

Также аналогами данного программного комплекса можно представить некоторые конфигурации 1С Предприятия, но они помимо того, что имеют изначально не маленькую стоимость, так еще и потребует некоторых доработок за отдельную сумму денег.





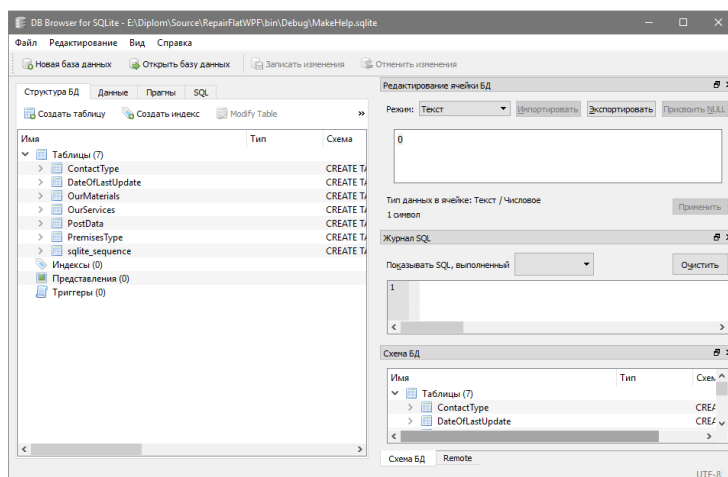


Рисунок 3 – Интерфейс DB Browser for SQLite

Кроме локальной базы данных, также встала необходимость использования серверной системы управления базами данных. В качестве данной системы управления базами данных был выбран «Microsoft SQL Server»[12]. Данная систему управления базами данных была выбран по причине простоты развертки данных баз данных, как на Linux, так и на Windows серверах, что повышает совместимость и уменьшает стоимость развертки данной системы управления базами данных на сервере.

В качестве инструмента взаимодействия с серверной системой управления базами данных был выбран «SQL Server Management Studio»[13]. Данное приложение представляет полный доступ к возможностям Microsoft SQL Server» через удобный пользовательский интерфейс пример которого представлен на рисунке 4.

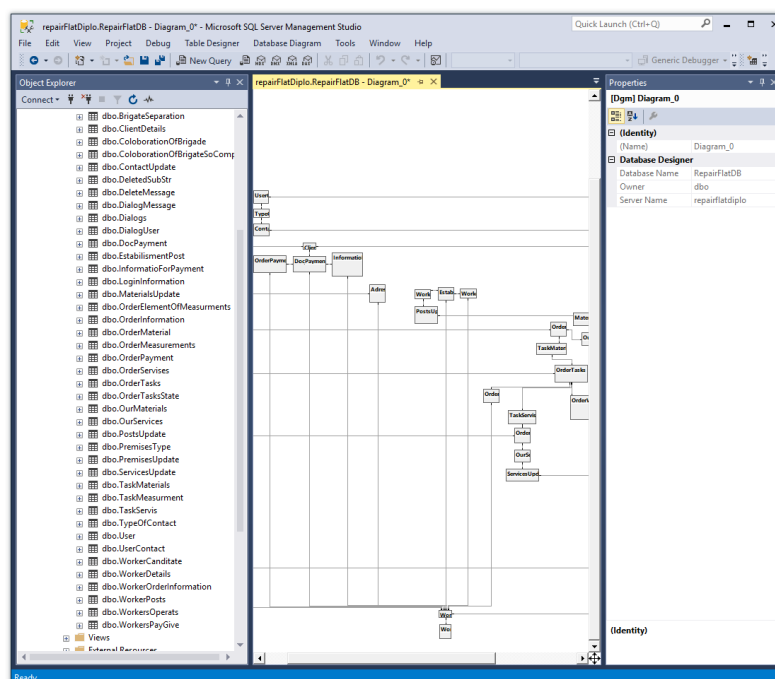


Рисунок 4 – Интерфейс SQL Server Management Studio

В качестве среды разработки была выбрана Visual Studio 19 с, удобным для пользователя интерфейсом и наличием отладчика кода.

Программирование выполнялось на языке программирования C#, тип проекта:

- для Windows приложения — проект графического окна, основанного на технологии WPF;
- для серверной части данного комплекса — ASP.NET Web Application (.NET Framework);
- для создания базы данных — проект базы данных.

Visual Studio 2019 (VS) представляет собой интегрированную среду разработки (Integrated Development Environment, IDE).

Пример пользовательского интерфейса VS представлен на рисунках 5-6. Стоит отметить, что в новой версии VS был переработан дизайн некоторых объектов, пример интерфейса которых и приведен ниже.

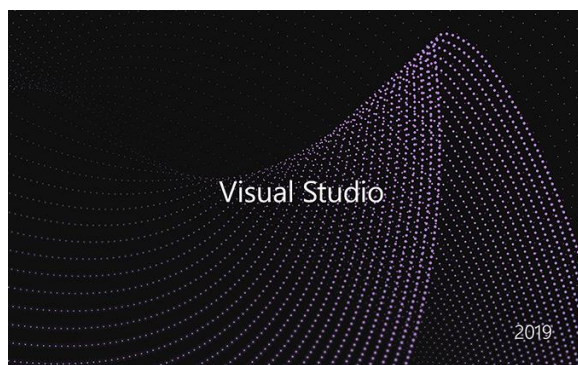


Рисунок 5 – Обновленный интерфейс загрузки Visual Studio 2019

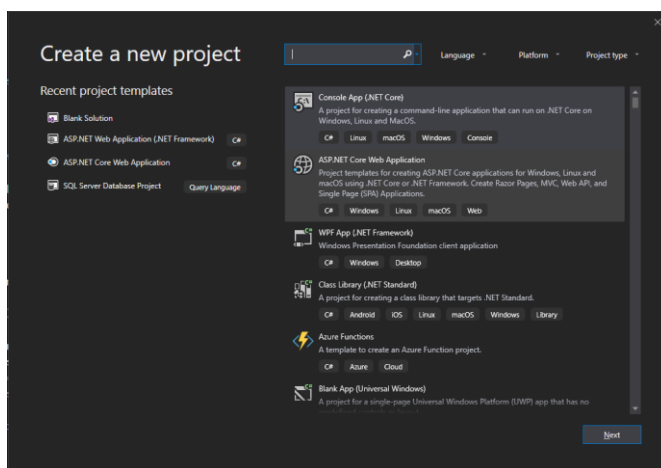


Рисунок 6 – Обновленный интерфейс выбора типа приложения Visual Studio 2019

С помощью VS многие из этих рутинных и трудоемких задач, связанных с повседневной работой программиста, можно упростить и облегчить.

В состав VS входит целый набор типовых проектов, из которых каждый разработчик может подобрать именно то, что ему в данный момент требуется. Каждый раз, когда вы создаете новый проект, VS автоматически создаст "скелет" будущего приложения, причем этот код можно немедленно скомпилировать и запустить на исполнение. В составе каждого типового проекта имеются элементы, которые по желанию добавлять в ваш проект.

Любой проект, в любом случае, содержит автоматически сгенерированный код, который представляет собой основу будущей программы. VS предлагает множество готовых к использованию элементов управления, включая и код, необходимый для их создания. Это экономит время разработчиков, избавляя их от необходимости каждый раз заново

создавать типовой программный код для решения часто встречающихся задач. [14]

Для отладки серверной части программной части разрабатываемого программного обеспечения был выбран соответствующий инструмент «Postman»[15]. Интерфейс данной программы представлен на рисунке 7.

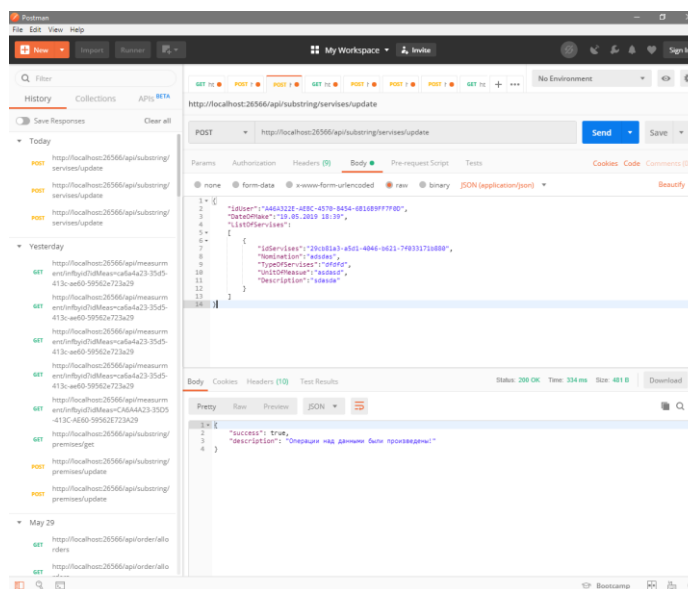


Рисунок 7 – Интерфейс программы Postman

### 1.2.2 Характеристика языка программирования

Дипломная работа выполнена с использованием объектно-ориентированного языка программирования С# и некоторых сторонних библиотеках, предназначенных для данного языка программирования.

С# — язык программирования, сочетающий объектно—ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с С# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

С# относится к семье языков с С-подобным синтаксисом, из них его синтаксис наиболее близок к С++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на

функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML.[16]

### 1.3 Анализ некоторых технологий, используемых при реализации программного комплекса

#### 1.3.1 Технология Entity Framework

Разработка автоматизированной информационной системы подразумевает использование базы данных как средство хранения обработанной информации.

Для работы с базами данных можно использовать язык запросов SQL. Данный язык программирования разработан для управления реляционными базами данных. Язык был создан в 1970х годах и назывался «SEQUEL» и был предназначен для управления системой баз данных System R. Через некоторое время данный язык был переименован «SQL» и в 1979 году был опубликован в качестве коммерческого продукта Oracle V2.

В 1986 году был создан первый официальный стандарт языка, после чего были созданы еще несколько стандартов языка. Несмотря на существование стандартов, большинство средств управления СУБД используют встроенный язык запросов, который иногда очень кардинально отличается от стандартов это объясняется большим объемом и сложностью описанного стандарта, а также нехваткой в нем некоторых спецификаций в важных областях.[17]

При реализации программного комплекса появится необходимость работы с запросами. Данную работу можно проводить многими способами, но помимо использования языка T-SQL, который необходимо было бы использовать по причине использования в качестве системы управления базами данных «Microsoft SQL Server». Использование данного языка запросов может привести к ошибкам после переименования некоторых полей, что возможно при разработке системы при одновременном проектировании данной.

В целях обеспечения безопасности была выбрана соответствующая технология, а именно «ADO.NET Entity Framework».

Данная технология представляет возможность работы с базами данных как работу с некоторыми сущностями. Сущностями в данном случае называют некий набор данных, которые ассоциативны с определенным объектом, но считать таблицы базы данных набором сущностей не совсем корректно, так как работа выполняется с объектами и их наборами.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из базы данных, но и получать объекты, связанные различными ассоциативными связями.

Еще одним ключевым понятием Entity Data Model. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней:

- концептуальном на котором происходит определение классов сущностей, используемых в приложении;
- уровень хранилища, который определяет таблицы, столбцы, отношения между таблицами и типами данных, которыми сопоставляется используемая база данных;
- уровень сопоставления (маппинга) который служит посредником между свойствами класса и сущности и столбцами таблицы.

Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.[18]

Для работы с Entity Framework необходимо изначально скачать NuGet пакет Entity Framework. Для скачивания NuGet пакетов необходимо правой кнопкой нажать на название своего проекта на платформе .Net, после чего

выбрать пункт, выделенный на рисунке 8. После чего перейти на первую вкладку «Browse» и в строку поиска ввести Entity Framework и в результатах выбрать пункт «EntityFramework» и нажать кнопку «Install» пример выбора данного пакета представлен на рисунке 9.

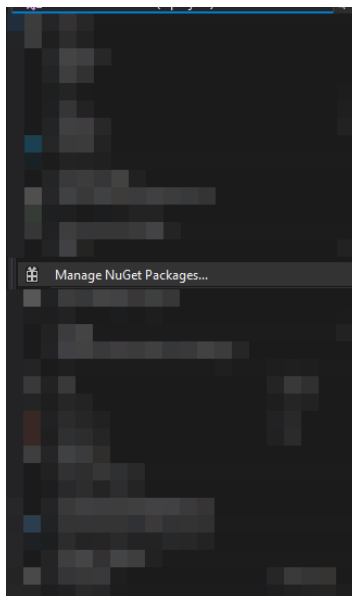


Рисунок 8 – Открытие менеджера пакетов NuGet

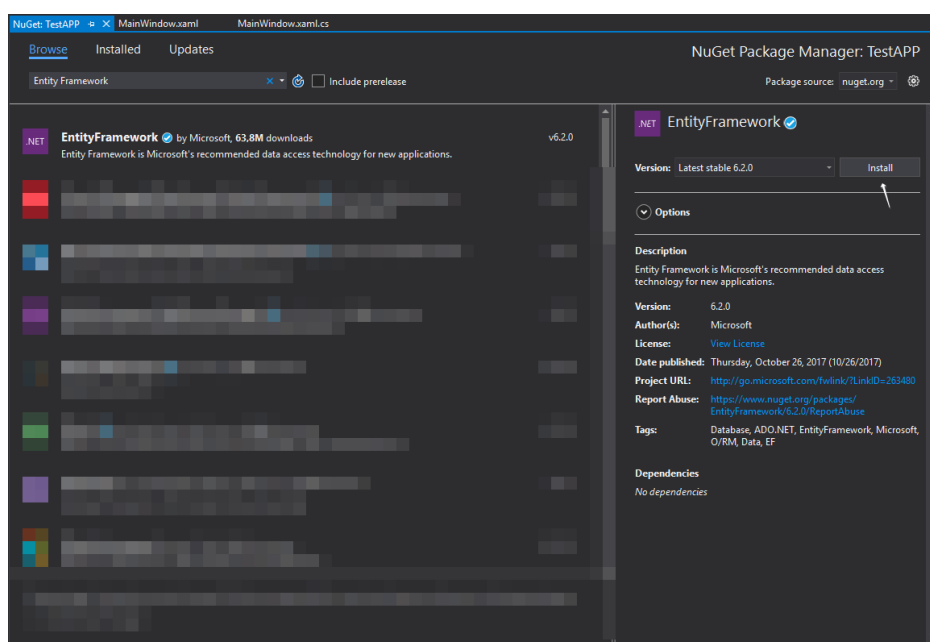


Рисунок 9 – Выбор необходимого пакета

После установки пакета необходимо в проект добавить объект «ADO.NET Entity Data Model» который добавляется путем нажатия правой кнопки на название проекта на платформе .Net после чего выбрать пункт

«Добавить» затем «Новый элемент» после чего перейти в раздел «Данные» и выбрать пункт «ADO.NET Entity Data Model» после чего ввести название и нажать кнопку добавить пример добавления представлен на рисунке 10.

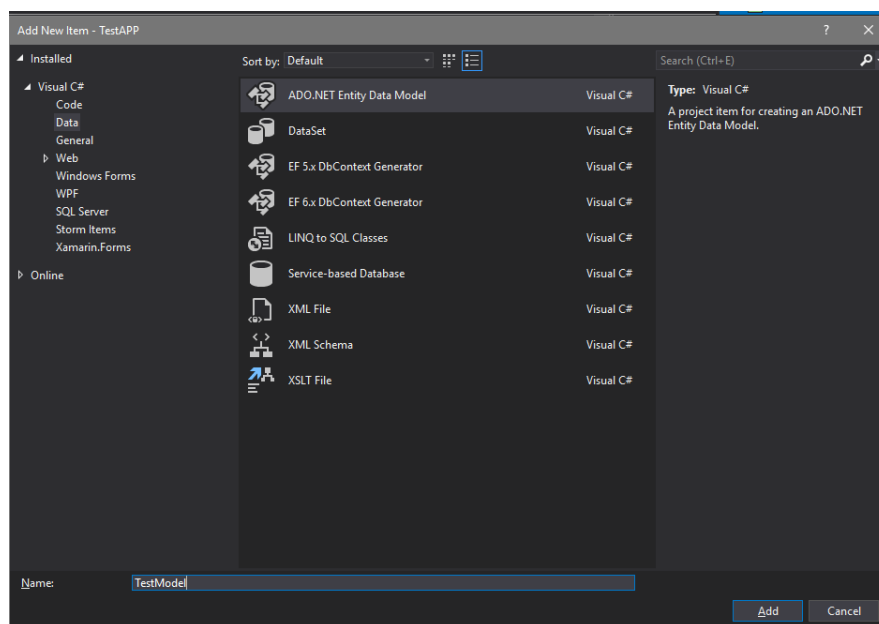


Рисунок 10 – Добавление «ADO.NET Entity Data Model» в проект

После нажатия кнопки добавить необходимо нажать далее, после чего указать строку подключения к базе данных путем нажатия кнопки «New Connection» на форме, представленной на рисунке 11. После указания строки подключения необходимо выбрать пункт «Yes, include sensitive data in the connection string», после чего нажать «Далее» и в открывшемся окне выбрать необходимые объекты для копирования в базу данных пример выбора данных представлен на рисунке 12, после чего нажать «Выполнить».



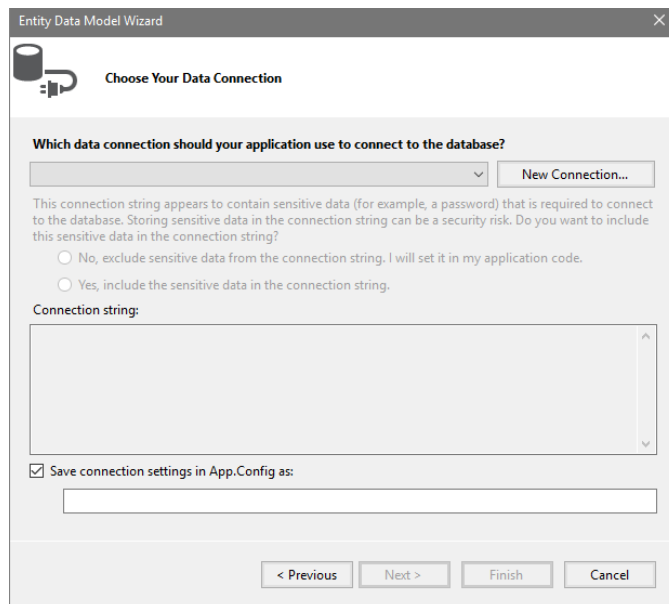


Рисунок 11 – Окно для указания строки подключения

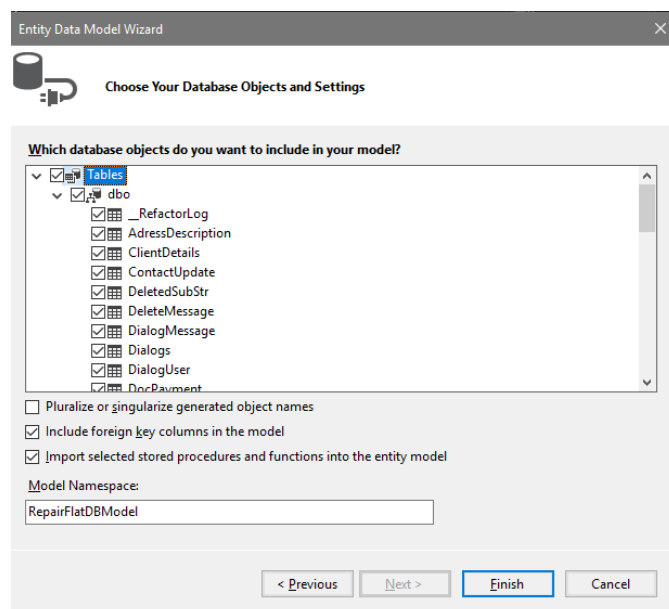


Рисунок 12 – Окно выбора необходимых компонентов базы данных

Через некоторое время данные будут скопированы и представлены в решении в соответствующем пункте в качестве примера, представленного на рисунке 13 указана модель, которая была скопированы из базы данных.

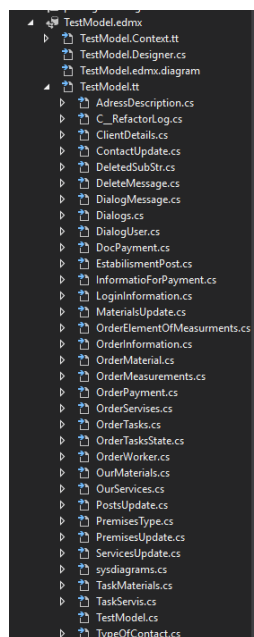


Рисунок 13 – Пример отображения классов, отображающих «сущности»

#### 1.4 Анализ требований и определение спецификаций программного обеспечения

Диаграмма вариантов использования клиентского приложения представлена на рисунке 14. Данная диаграмма описывает работу приложения по соответствующим сценариям, которые прописаны в зависимости от роли, которая выдана определенному логину и паролю.

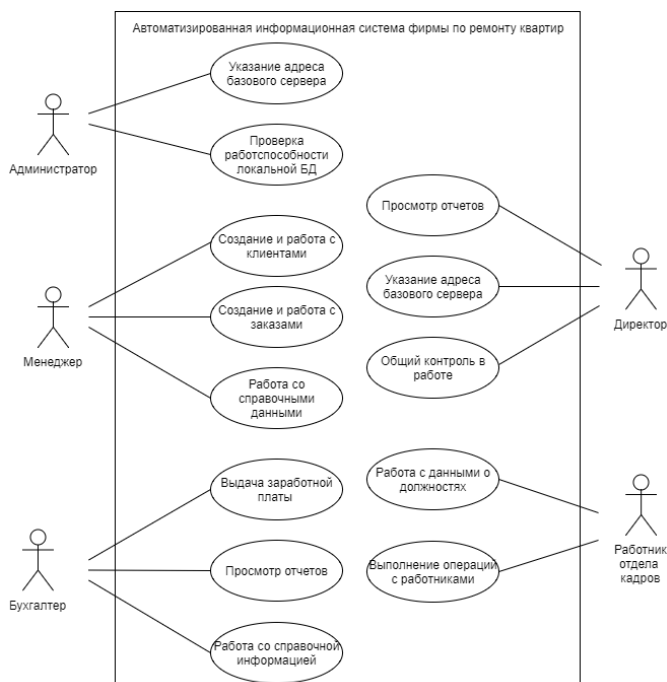


Рисунок 14 - Диаграмма вариантов использования системы

Контекстная диаграмма потоков данных, которая представлена на рисунке 15. Описывает взаимодействие между пользователями и сервером обработки информации посредством клиентского приложения.



Рисунок 15 – Контекстная диаграмма потоков данных

На данной диаграмме отображены данные, которые предоставляются пользователям:

- о работниках;
- о клиентах;
- о деталях заказа;
- о справочной информации.

Программа в качестве ответа демонстрирует информацию, о том, что данные были добавлены, а также демонстрирует общий список данных, которые необходимы в данном контексте.

Также ниже представлена детализированная диаграмма потоков данных в программном комплексе, представленная на рисунке 16. Данная диаграмма детализирует контекстную диаграмму, приведенную выше.

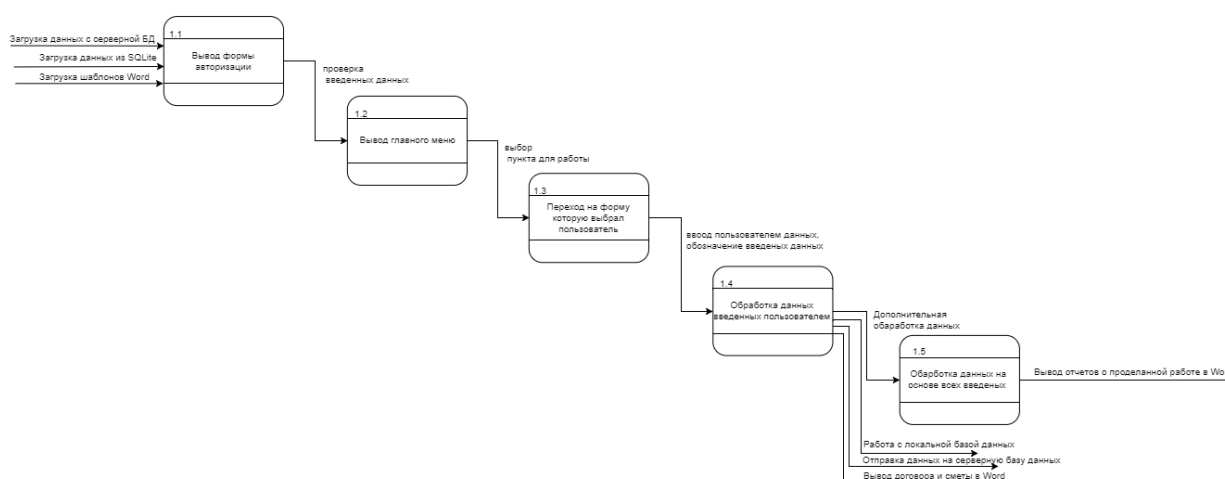


Рисунок 16 - Детализированная диаграмма потоков данных

Для разрабатываемого программного комплекса была составлена контекстная функциональная диаграмма, описывающая схему взаимодействия серверного модуля с прочими модулями и представлена она на рисунке 17. На данной диаграмме показано, что в качестве основного модуля можно считать модуль сервера, который обрабатывает все необходимое для работы клиента. Сервер в свою очередь выполняет полную связь с базой данных.



Рисунок 17 – Контекстная функциональная диаграмма

Далее была составлена детализированная функциональная диаграмма, представленная на рисунке 18. Данная диаграмма описывает основные функции со стороны серверной части при стандартной работе пользователя с системой, аналогичной детализированной диаграммы потоков данных.

На вход при стандартной работе подается строка подключения, которая берется из настроек и может быть изменена, а также данные от пользователя, либо вызов соответствующей операции по ссылке.

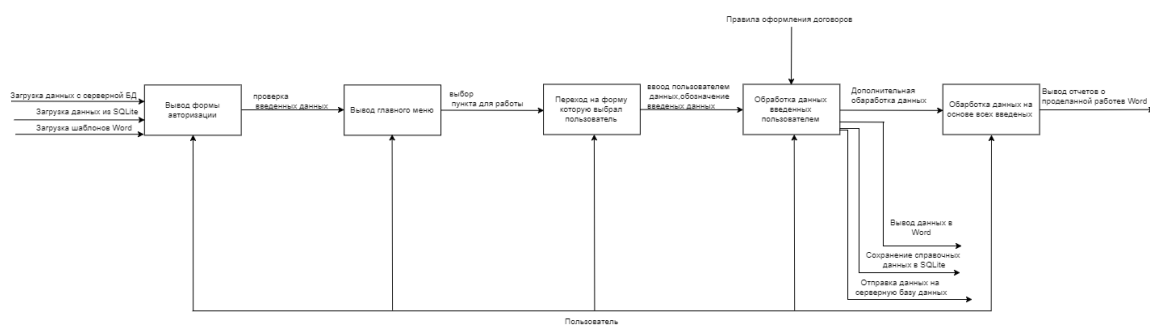


Рисунок 18 – Детализированная функциональная диаграмма

## Глава 2 Практическая часть

### 2.1 Постановка задачи

#### 2.1.1 Функции программного продукта

Среди основных функций разрабатываемого программного обеспечения можно отметить следующие:

- подсчет стоимости заказа на ремонт;
- создание договора на основе шаблона, созданного заранее;
- импорт информации о материалах и услугах, предоставляемых данной организацией из файла программы Excel;
- возможность добавления, редактирования информации об услугах;
- возможность работы с данными о сотрудниках (прием на работу, увольнение и кадровые перемещения) с документированием в формате Word;
- возможность раздельного доступа к программе путем назначения прав доступа для сотрудников, а также возможность их редактирования;
- учет данных о клиентах и сотрудниках, а также редактирование данных об этом факте;
- учет оказанных услуг и учет платежей, а также система отчетов о проделанных работах и проданных услугах;
- фиксирование замеров помещений и вычисление площади некоторых поверхностей.

#### 2.1.2 Описание входной информации

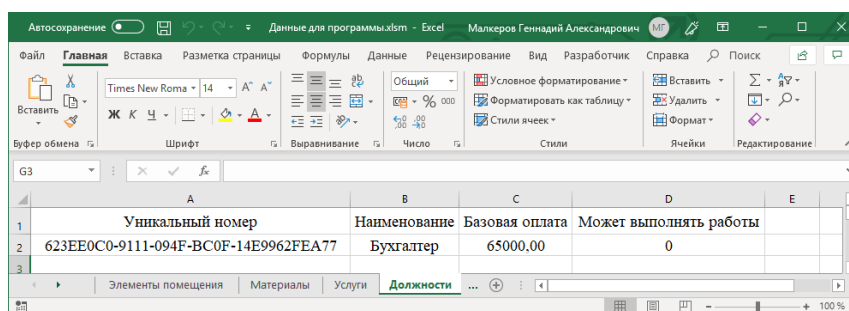
В качестве входной информации можно отметить следующие основные данные необходимые для работы приложения:

- данные о сотрудниках и клиентах;
- данные о заказе в том числе размеры помещений;
- данные о материалах и услугах, предоставляемых данной компанией;

- данные о должностях в данной компании;
- данные о выполненных этапах и услугах;
- данные о платежах и выплатах.

Справочную информацию, которую необходимо загрузить на сервер предполагается загружать с помощью специально подготовленного файла. Данный Excel файл содержит информацию обо всех типах справочной информации, которую предполагается использовать в качестве данной.

Пример данного файла представлен на рисунке 19. Также стоит отметить, что данный файл содержит в себе скрипт, который создает уникальный идентификатор определенного элемента, что позволяет помимо добавления производить и обновление данных.



	А	В	С	Д	Е
1	Уникальный номер	Наименование	Базовая оплата	Может выполнять работы	
2	623EE0C0-9111-094F-BC0F-14E9962FEA77	Бухгалтер	65000,00	0	

Рисунок 19 – Пример отображения файла Excel

### 2.1.3 Описание выходной информации

В качестве выходной информации в данной программе служат отчеты об операциях, выполняемых в данной АИС, а также смета и договор по каждому заказу.

### 2.1.4 Требование к программному обеспечению

Системные требования для клиентского приложения.

Необходимыми программными средствами для работы будут являться:

- операционная система Windows 10;
- приложение Microsoft Word и Microsoft Excel;
- наличие .Net Framework 4.7 версии и выше.

Необходимыми аппаратными средствами является:

- процессор с тактовой частотой 2,7 ГГц или выше. Рекомендуется использовать двухъядерный процессор;
- 1 ГБ ОЗУ (3 ГБ при выполнении в виртуальной машине);
- 10 ГБ доступного пространства на жестком диске;
- видеоадаптер с поддержкой DirectX 9 (разрешение 1600 x 900 или больше);
- монитор с разрешением не меньше, чем (1600 x 900);
- клавиатура и мышь.

Функциональные и нефункциональные требования для работы.

В качестве функциональных требований можно считать следующие:

- программа должна осуществлять добавление, редактирование, удаление, просмотр информации, а также ограничить доступ к данным посторонним лицам;
- работать с заданным алгоритмом функционирования;
- поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей;
- производить бесперебойную работу по преобразованию информации.

В качестве нефункционального требования можно считать следующее:

- требования к интерфейсу.

Требования к реализации

Данная АИС должна быть реализована с использованием языка программирования C# и принципов объектно ориентированного программирования.

Требования к надёжности выполняемого проекта.

В программном продукте предусмотрена защита от действий, пользователя которые могли бы были привести к выдаче ошибки и экстренного закрытия программного продукта:

- защита от SQL- инъекций;



— защита от некорректного ввода значений в программе;

Требования к интерфейсу клиентского приложения.

Интерфейс данной АИС должен быть понятен пользователю и не иметь в себе что-то раздражающее так как данный программный продукт пользователь будет использовать большую часть рабочего времени.

Требования к программной документации

Программная документация оформляется в соответствии с требованиями документа «МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОФОРМЛЕНИЮ КУРСОВОЙ И ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ» и указанными в нём ГОСТами.

## 2.2 Проектирование программного обеспечения

### 2.2.1 Проектирование баз данных

Стоит отметить, что для программы были разработаны сразу две схемы баз данных:

- основная база данных, используется на сервере «Microsoft SQL Server» в качестве основной базы данных;
- в качестве дополнительной базы данных на компьютере клиента.

Рассмотрим обе базы представленные в качестве диаграмм сущность-связь.

Начнем с локальной базы данных схема которой представлена на рисунке 20.

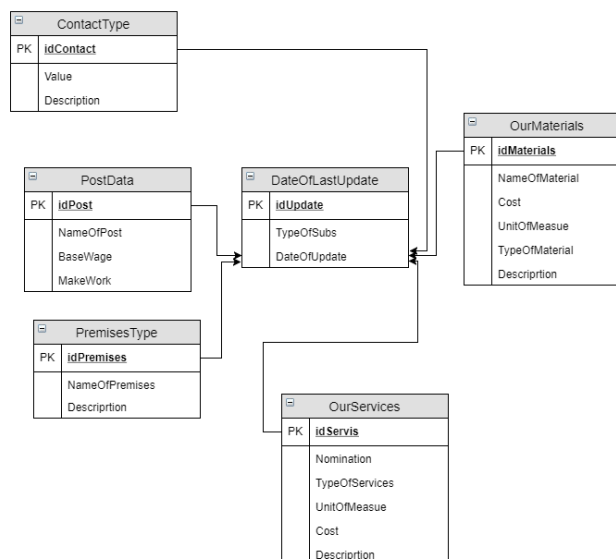


Рисунок 20 – Схема локальной базы данных

Данная база данных разработана на языке SQLite, для отображения данных на тестировании используется редактор «DB Browser for SQLite».

На данной схеме представлены 6 таблиц, которые связывает таблиц с датой последнего обновления данных на данном компьютере.

Стоит отметить, что данные таблицы создаются на основе JSON файла которые прикреплён в ресурсах клиентской части продукта и при желании может быть изменена.

Как уже говорилось выше, база данных на момент написания данной документации состоит из 6 таблиц:

- ContactType – Тип контактов – содержит данные о типах контактах в качестве справочной информации;
- DateOfLastUpdate – Дата последнего обновления – содержит данные о последнем обновлении данных с сервера и типе объектов обновления;
- OurMaterials – Наши материалы – содержит данные о материалах, которые используются в качестве справочных данных в программе;
- OurServices – Наши услуги – содержит данные об услугах, которые используются в качестве справочных данных в программе;

- postData – Данные о должностях – содержит информацию о должностях и используется, когда происходит работа кадрового отдела;
- PremisesType – Данные о типах помещений – содержит информацию о типах помещений и используется в качестве справочных данных.

Для проектирования основной базы данных был выбран способ проектирования с использованием специализированного проекта в среде разработки Visual Studio 2019. Данный проект называется проект базы данных и с помощью данного проекта возможно спроектировать всю базу данных после чего загрузить с использованием специальных средств на сервер «Microsoft SQL Server». Схема сущность-связь разработанной базы данных представлена на рисунке 21.

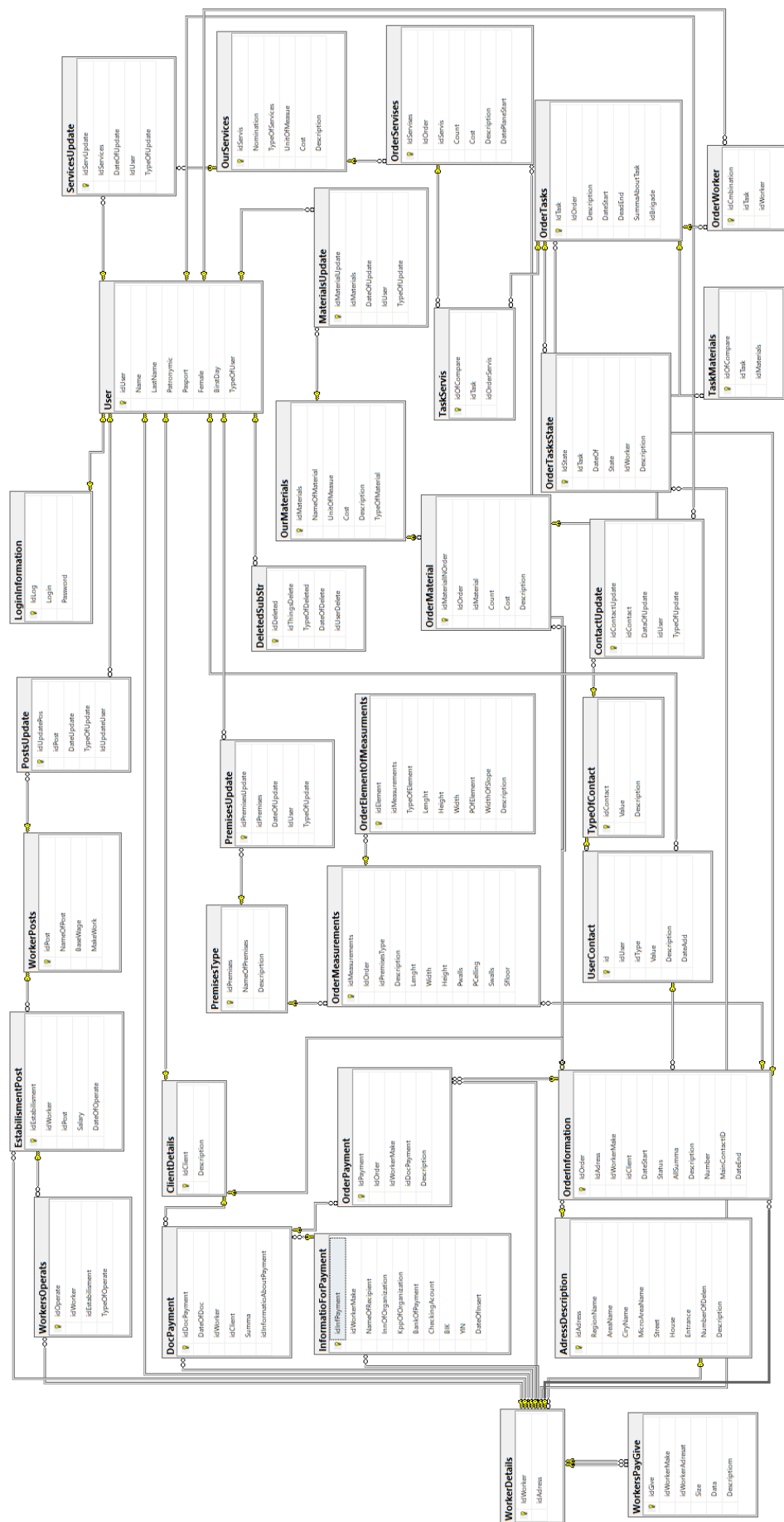


Рисунок 21– Схема основной базы данных

Основная база данных состоит из 33 таблиц:

- AdressDescription – Данные о адресе – данные об адресе как работника, так и адрес выполнения заказа;
- ClientDetails – Данные о клиенте – детальная информация о клиенте данной фирмы;
- ContactUpdate – Обновление справочной информации о типах контактной информации – информация о том, что типы контактной информации были обновлены и кто это сделал;
- DeletedSubStr – Удаленная справочная информация – данные о том, какая справочная информация была удалена и когда это произошло;
- DocPayment – Документы к оплате – документы, выданные клиентам для оплаты;
- EstabilishmentPost – Установление должности за работником – информация о том, что за определенным работником была установлена определенная заработная плата и определенный должностной пост;
- InformatioForPayment – Данные для оплаты – содержит реквизиты для оплаты клиентами заказов;
- LoginInformation – Данные для авторизации – содержит информацию для авторизации в том числе и клиентов при необходимости;
- MaterialsUpdate – Данные об обновлении материалов – содержит данные о том, что материалы были обновлены и когда это произошло;
- OrderElementOfMeasurments – Информация об элементах помещения – содержит данные о том какие элементы помещения содержаться в данном помещении;
- OrderInformation – Информация о заказе – содержит основную информацию о заказе, который поступил от клиента;
- OrderMaterial – Информация о материалах для заказа – содержит данные о материалах для заказа;

- OrderMeasurements – Информация о помещениях для заказа – содержит данные о помещениях для заказа;
- OrderPayment – Данные об оплате заказов – информация о том, какую оплату произвел клиент;
- OrderServices – Данные об услугах для заказа – информация об услугах для всех заказов;
- OrderTasks – Данные о заданиях для заказа – информация о заданиях для заказов;
- OrderTasksState – Степень готовности заказов – информация о том на какой стадии в данный момент находится определенное задание;
- OrderWorker – Данные о работниках, которые работают над заказом – информация о том какие рабочие выполняют работу над определенным заказом;
- OurMaterials – Справочные данные о материалах – данные о том, какие материалы доступны для работы;
- OurServices – Справочные данные об услугах – данные о том, какие услуги доступны для работы;
- PostsUpdate – Данные об обновлении должностей – данные о том, когда были обновлены определенные должности;
- PremisesType – Тип помещений – данные о типах помещений;
- PremisesUpdate – Данные об обновлении типа помещений – информация о том, когда происходило обновление определенного типа помещений;
- ServicesUpdate – Данные об обновлении услуг – информация о том, когда происходило обновление определенных услуг;
- TaskMaterials – Материалы для заданий – данные о материалах, которые используются в заданиях;
- TaskServis – Услуги для заданий – информация об услугах, которые используются в заданиях;

- TypeOfContact – Тип контактов – информация о типах контактов;
- User – Пользователь – информация о пользователях, как о сотрудниках, так и о клиентах;
- UserContact – Контактная информация пользователя – информация о контактах определенного пользователя;
- WorkerDetails – Детальная информация о работнике – информация необходимая для работы с кадрами;
- WorkerPosts – Доступные должности работников – информация о должностях доступных работникам;
- WorkersOperats – Операции над должностями работников – определенные операции над должностями работников;
- WorkersPayGive – Данные о выдаче заработной платы работникам – информация о выдаче заработной платы работникам, принятым на работу.

### 2.3 Разработка пользовательских интерфейсов программного обеспечения

Разработка пользовательского интерфейса являлась одной из основной стадией работы над проектом по причине необходимости работы приложения в адаптивном режиме.

Для реализации данного требования со стороны заказчиков программного обеспечения была выполнена разработка основного окна который повторял бы все стандартные функции окна WPF проекта C#.

Среди основных функций можно считать следующие:

- возможность перемещать окно путем зажатия шапки окна;
- возможность сворачивать окно путем нажатия на соответствующую кнопку в шапке окна;
- возможность разворачивать окно на весь экран путем нажатия на соответствующую кнопку в шапке окна;
- возможность разворачивать окно путем двойного нажатия на его шапку.

Все вышеперечисленные требования удалось решить путем использования следующих библиотек:

- Microsoft.Windows.Shell – предоставляет возможность использовать стандартные функции окна windows там, где это не предусмотрено изначально, что позволяет использовать данную библиотеку как средство работы с окном;

- Fody – необходима для синхронизации работы окна и внутренних свойств некоторых классов, которые будут использоваться в качестве классов для расчета некоторых свойств окна.

При использовании данного набора библиотек было создано адаптивное окно, представленное на рисунке 22.

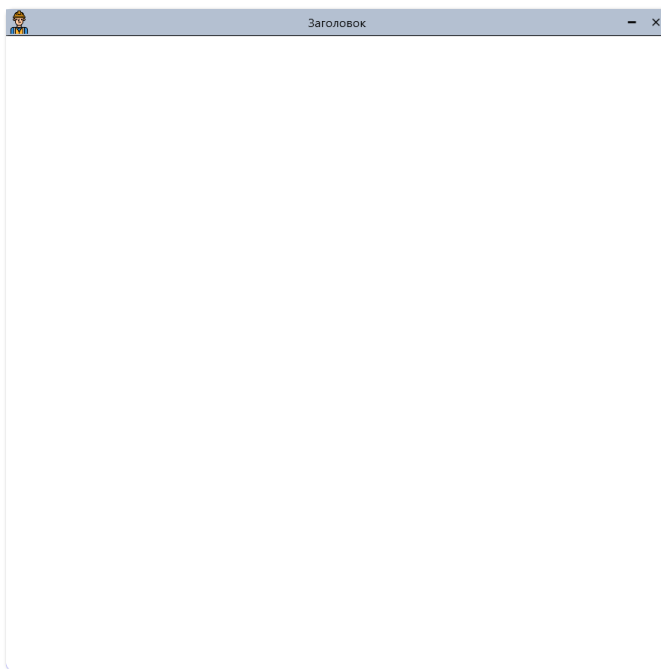


Рисунок 22 – Базовое окно приложения

Пример адаптивности окна представлена на рисунке 23-24.



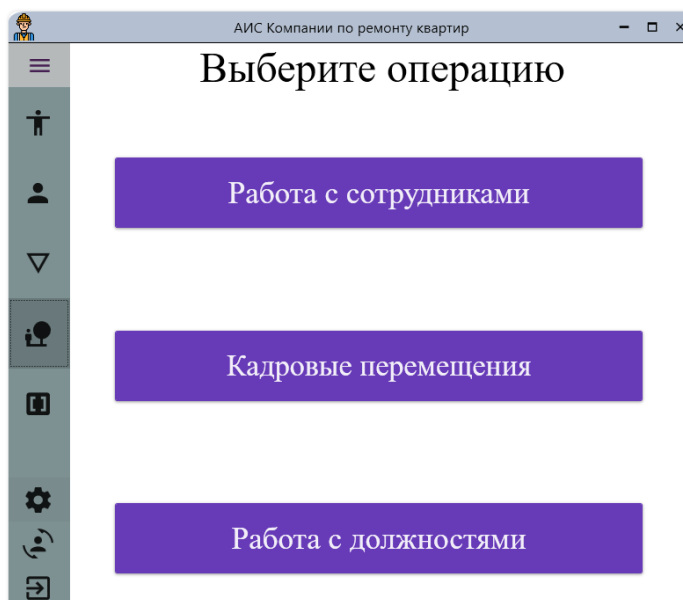


Рисунок 23 – Пример адаптивности окна

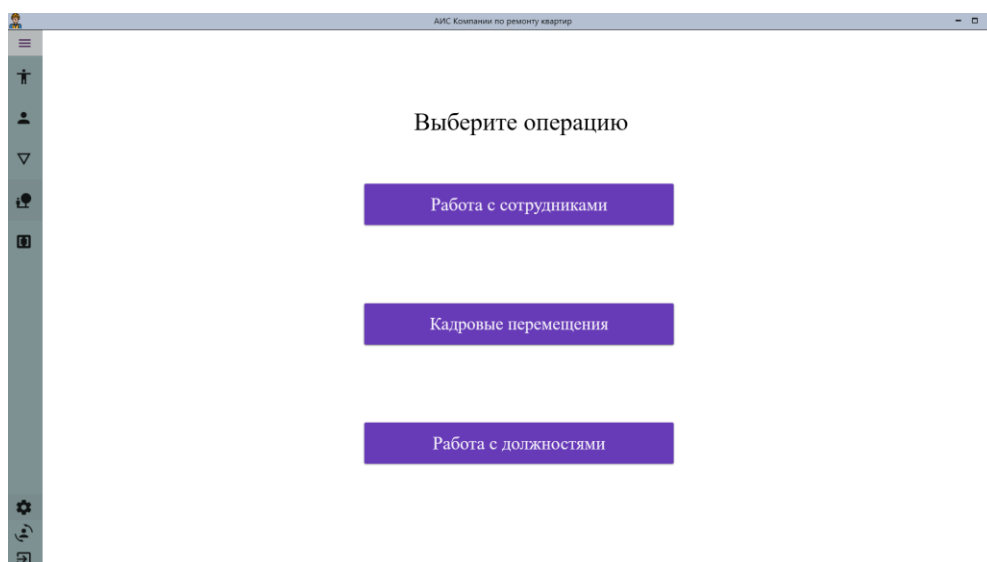


Рисунок 24 – Пример адаптивности окна

В качестве элемента для отображения содержимого окна был использован стандартный элемент WPF User Control.

Данный элемент управления представляет из себя некий скелет, в который можно встроить различные элементы управления, что позволяет сделать вообще любой интерфейс для работы.

Для отображения данного элемента управления можно использовать стандартный элемент управления Grid. В который в качестве ребенка можно

вставить User Control. В итоге пользователь увидит не 2 элемента управления, а один единственный который будет отображать содержимое User control.

С помощью User Control можно сделать и элементы управления адаптивные, что позволит сделать интерфейс изменяемым. Пример отображения User control в редакторе элементов управления WPF представлен на рисунке 25. При этом при отображении данного элемента управления на форме пользователь будет видеть примерно то, что представлено на рисунке 26.

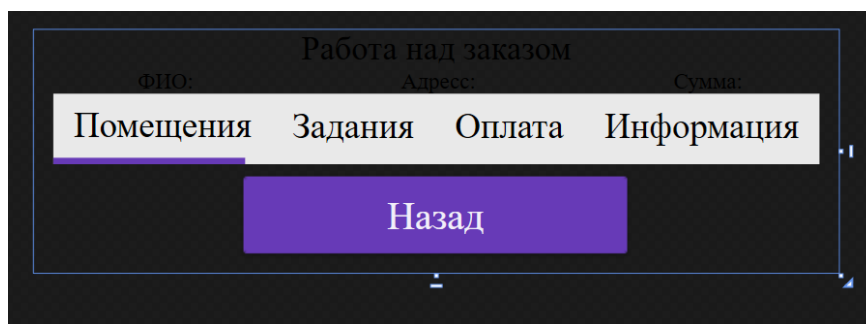


Рисунок 25 – Элемент управления User Control при отображении в редакторе содержимого

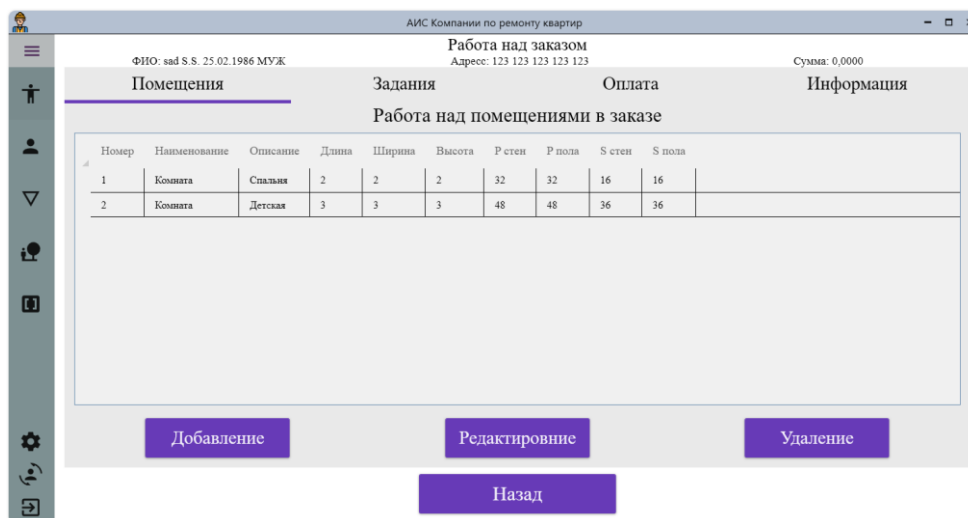


Рисунок 26 – Элемент управления User Control при отображении в окне

Для работы пользователей был создан целый набор User Controlov который подставляются в зависимости от необходимости работы с определенными операциями.

Также по причине необходимости работы разных типов людей была создана система логирования пример отображения которой представлен на рисунке 27.

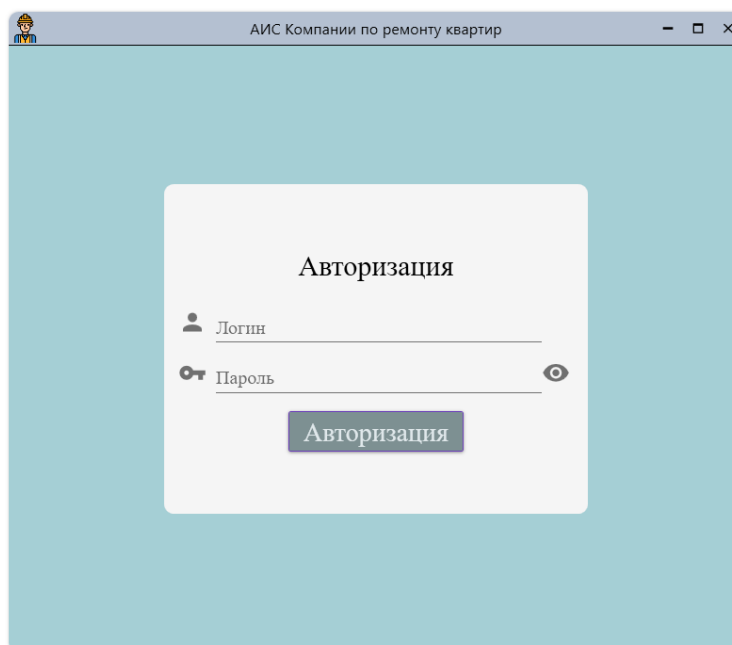


Рисунок 27 – Окно авторизации в приложении

После авторизации перед пользователем появляется окно загрузки, которое отображает степень загрузки определенных данных с сервера для ускорения работы при работе с этими данными, по причине того, что эти данные подставляются автоматически из локальной базы данных, которая создается автоматически. Окно загрузки представлено на рисунке 28.

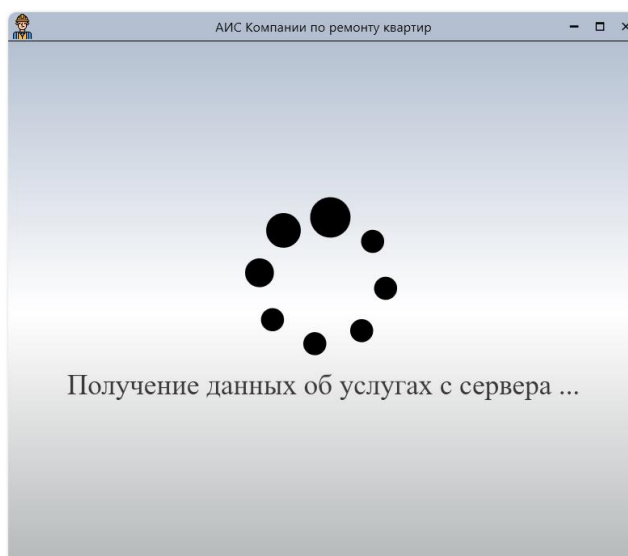


Рисунок 28 – Окно загрузки

Меню данного программного обеспечения представлено на рисунке 29. И представляет из себя боковую панель, которая при нажатии на кнопку будет увеличиваться.

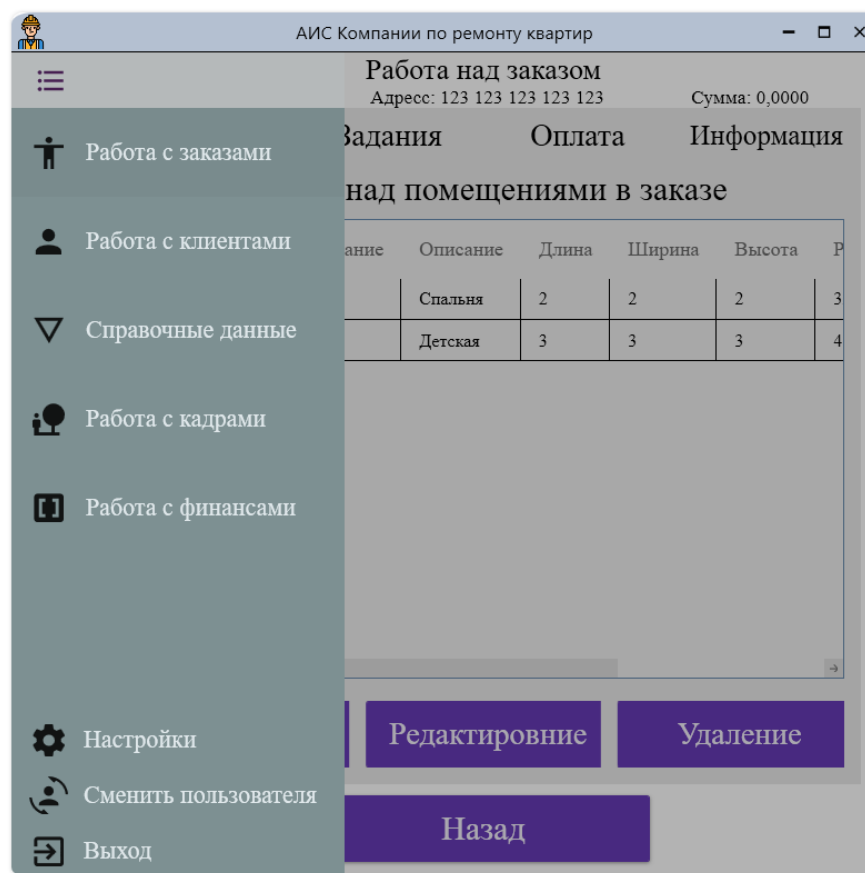


Рисунок 29 – Меню приложения

Также приведу пример некоторый интерфейс приложения, который чем-либо примечателен. Пример данных окон представлен на рисунках 30-32.

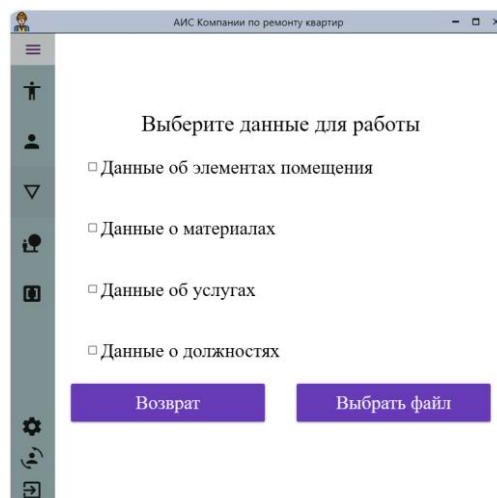


Рисунок 30 – Интерфейс приложения

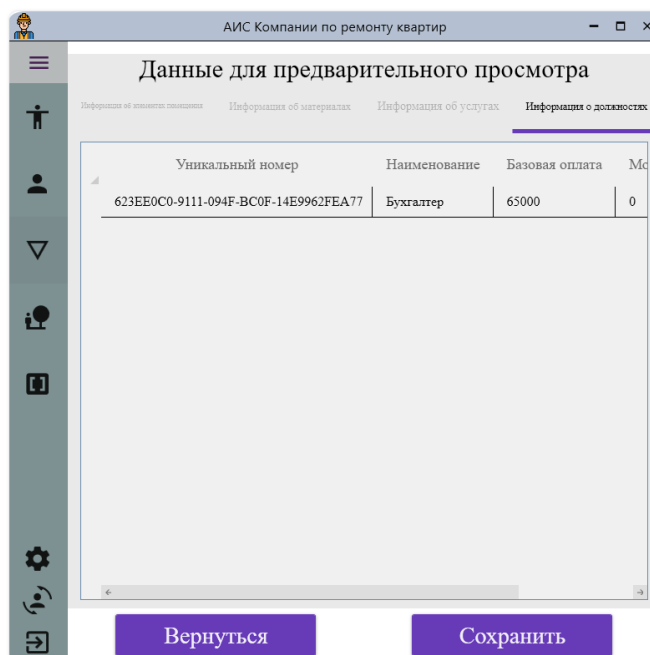


Рисунок 31 – Интерфейс приложения

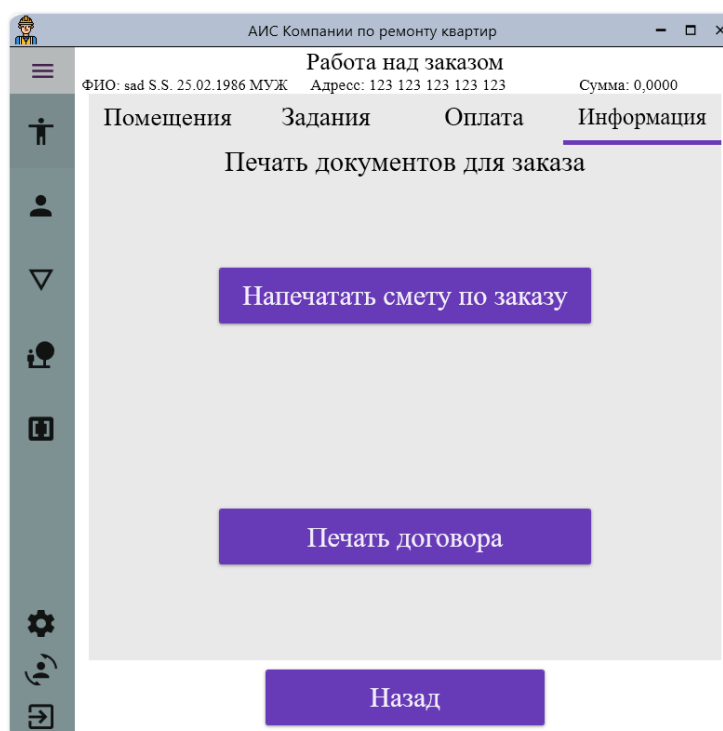


Рисунок 32 – Интерфейс приложения

## 2.4 Тестирование и отладка программного обеспечения

Тестирование и отладка серверной части программного обеспечения происходило посредством программы Postman подробности которой описаны в разделе «1.2.1 Характеристика среды программирования».

Данная программа позволяет отправлять запросы на сервер, который в данном случае представлен ASP.Net проектом на языке C#, который был разработан в «Visual Studio 2019» описание которого также приведено в разделе «1.2.1 Характеристика среды программирования».

Visual Studio предоставляет очень сильный инструмент для тестирования и отладки программного обеспечения. Основным инструментом для этого можно считать так называемые «точки останова».

Данные «точки останова» помогают в демонстрировании мест, где программное обеспечение принимает тот либо иной тип информации и значение данной информации.

Основным механизмом тестирования серверной части можно считать следующий:

- отправка запросов программой Postman;
- проверка на отсутствие ошибок при работе серверной части программы;
- при возникновении неточностей ставиться точка останова в месте, где начинается работа определенного метода и смотреться причина данной ошибки.

После создания и отладки серверной части программного продукта настало время клиентской программы. Отладка данной части программного продукта также проходила в «Visual Studio 2019».

При разработке интерфейсов возникла особенность при работе основного окна при работе на новых компьютерах по причине того, что использованная библиотека прописывает некоторые подробности на компьютере. По этой причине окно изначально выглядит немного не так, как это задумано изначально.

Основной проблемой при разработке была в том, что при разработке моделей, которые преобразуются в JSON возникали опечатки, которые

препятствовали получению корректной информации либо вообще ошибки при работе со стороны серверной части.

При наличии такого количества объектов при разработке избежать ошибок крайне сложно и не редко установление причин данных ошибок может занять большое количество времени сил и моральных затрат по причине большого количества абсурда в ошибках, которые возникали при разработке.

## 2.5 Руководство по использованию программного обеспечения

### 2.5.1 Руководство системного программиста

#### 2.5.1.1 Общие сведения о программе

Данная программа предназначена для автоматизации работы фирмы по ремонту квартир и позволяет автоматизировать некоторые рутинные процессы, которые выполняются в типичной компании по данной тематике.

Программа позволяет следующие процессы:

- составление сметы на материалы и услуги по определенным заданиям, которые были созданы;
- составление договора на оказание услуг;
- расчет площади элементов помещений с учетом элементов данных помещений;
- генерация отчетов в Word о различных хозяйственных операциях, происходящих в данной фирме;
- возможность загрузки справочной информации из файла Excel;
- разделенный режим доступа к программе;
- возможность работы с сотрудниками.

#### 2.5.1.2 Проверка программы

Проверка работоспособности программы заключается в процессе отображения на форме авторизации сообщения, о том, что связь с сервером была установлена.

### 2.5.1.3 Требование к исполняемому серверу

Для работы данного программного комплекса необходимо наличие некоторого удаленного сервера, на котором предполагается выполнение серверной части программного обеспечения.

Среди требований к исполняемому серверу можно отметить следующие:

- наличие .Net Framework версии не ниже 4.6;
- наличие не менее 6 гб свободного пространства на диске;
- наличие не менее 2 гб оперативной памяти;
- частота процессора не менее 1,4 ГГц;
- тип процессора x64: AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T.

### 2.5.1.4 Сообщения системному администратору

Сообщения системному администратору не предусмотрены в данной программе.

## 2.5.2 Руководство программиста

### 2.5.2.1 Назначения и условия выполнения программы

Данная программа предназначена для автоматизации работы фирмы по ремонту квартир и позволяет автоматизировать некоторые рутинные процессы, которые выполняются в типичной компании по данной тематике.

Программа позволяет следующие процессы:

- составление сметы на материалы и услуги по определенным заданиям, которые были созданы;
- составление договора на оказание услуг;
- расчет площади элементов помещений с учетом элементов данных помещений;
- генерация отчетов в Word о различных хозяйственных операциях, происходящих в данной фирме;
- возможность загрузки справочной информации из файла Excel;



- разделенный режим доступа к программе;
- возможность работы с сотрудниками.

Необходимыми программными средствами для работы будут являться:

- операционная система Windows 10;
- приложение Microsoft Word и Microsoft Excel;
- наличие .Net Framework 4.7 версии и выше.

Необходимыми аппаратными средствами является:

- процессор с тактовой частотой 2,7 ГГц или выше. Рекомендуется использовать двухъядерный процессор;
- 1 ГБ ОЗУ (3 ГБ при выполнении в виртуальной машине);
- 10 ГБ доступного пространства на жестком диске;
- видеоадаптер с поддержкой DirectX 9 (разрешение 1600 x 900 или больше);
- монитор с разрешением не меньше, чем (1600 x 900);
- клавиатура и мышь.

#### 2.5.2.2 Обращение к программе

Обращение к программе происходит посредством запуска приложения «RepairFlatWPF.exe», после чего отобразиться окно авторизации, после которой пользователь попадает внутрь программы.

#### 2.5.2.3 Входные и выходные данные

Входные и выходные данные можно посмотреть в разделе 2.1.2 Описание входной информации и в разделе 2.1.3 Описание выходной информации данной пояснительной записки.

#### 2.5.2.4 Сообщения

Сообщения программисту в данной автоматизированной информационной системе не предусмотрено.

## 2.5.3 Руководство пользователя

### 2.5.3.1 Назначение программы

Данная программа предназначена для автоматизации работы фирмы по ремонту квартир и позволяет автоматизировать некоторые рутинные процессы, которые выполняются в типичной компании по данной тематике.

Программа позволяет следующие процессы:

- составление сметы на материалы и услуги по определенным заданиям, которые были созданы;
- составление договора на оказание услуг;
- расчет площади элементов помещений с учетом элементов данных помещений;
- генерация отчетов в Word о различных хозяйственных операциях, происходящих в данной фирме;
- возможность загрузки справочной информации из файла Excel;
- разделенный режим доступа к программе;
- возможность работы с сотрудниками.

### 2.5.3.2 Условия выполнения программы

Необходимыми программными средствами для работы будут являться:

- операционная система Windows 10;
- приложение Microsoft Word и Microsoft Excel;
- наличие .Net Framework 4.7 версии и выше.

Необходимыми аппаратными средствами является:

- процессор с тактовой частотой 2,7 ГГц или выше. Рекомендуется использовать двухъядерный процессор;
- 1 ГБ ОЗУ (3 ГБ при выполнении в виртуальной машине);
- 10 ГБ доступного пространства на жестком диске;
- видеоадаптер с поддержкой DirectX 9 (разрешение 1600 x 900 или больше);

- монитор с разрешением не меньше, чем (1600 x 900);
- клавиатура и мышь.

### 2.5.3.3 Выполнение программы

После запуска программы на выполнение перед пользователем демонстрируется окно, представленное на рисунке 33, на котором необходимо будет ввести логин и пароль работника.

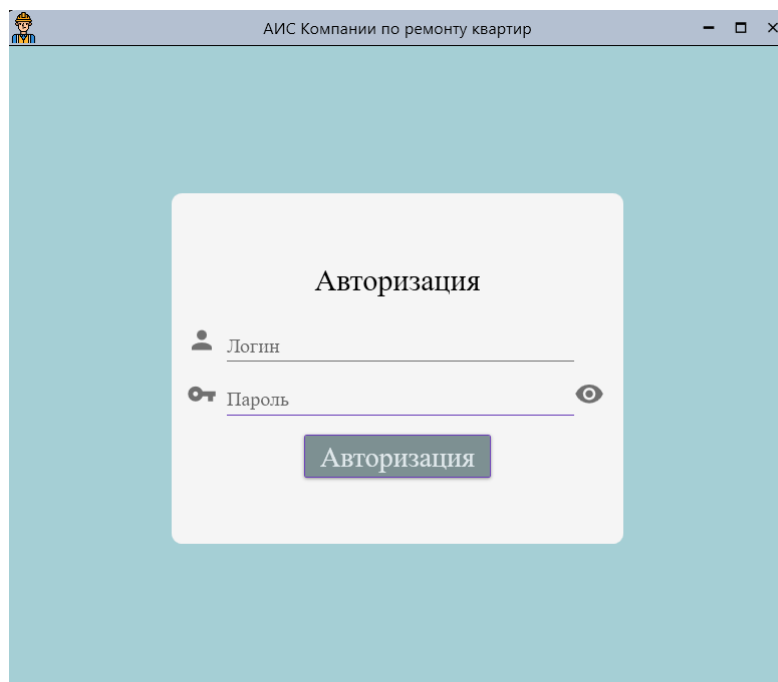


Рисунок 33 – Окно авторизации в системе

При неверном логине либо пароле пользователь получит сообщение, представленное на рисунке 34.

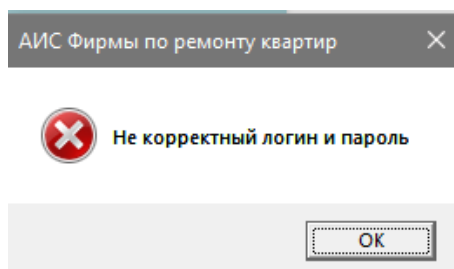


Рисунок 34 – Сообщение о неверном логине либо пароле

При работе с программой возможно использование в многопользовательском режиме с использованием следующего списка ролей:

- работник отдела кадров;

- менеджер;
- администратор;
- директор.

Рассмотрим особенности каждой роли на примере работы программы.

Работник отдела кадров имеет возможность работы только с кадрами фирмы. После авторизацией под данной ролью перед пользователем представляется окно, представленное на рисунке 35.

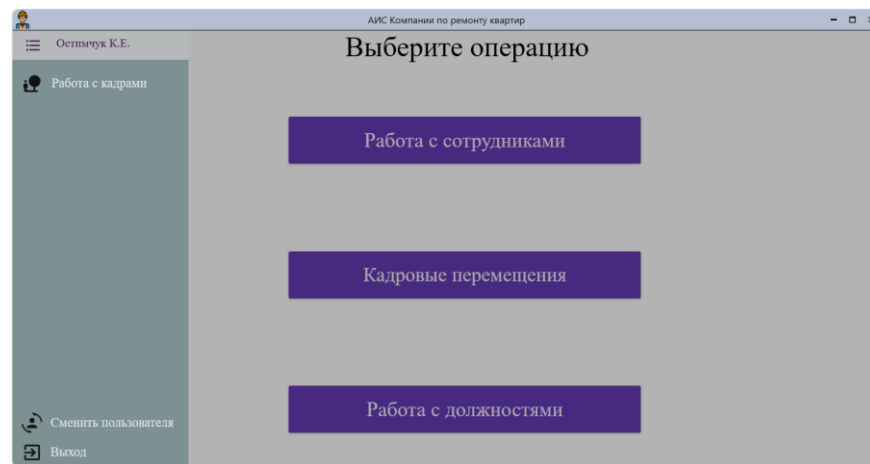


Рисунок 35 – Главное меню приложения при авторизации под правами кадрового работника

После закрытия бокового меню пользователь может выполнить следующие действия:

- создание нового работника;
- редактирования данных о работниках;
- принятие на работу нового сотрудника;
- изменить должность работника;
- указать данные о логине и пароле для работника.

Для примера создадим нового работника. Все совпадения с реальными людьми являются совпадением.

Чтобы добавить нового работника в данную программу необходимо зайти в основное окно работы с пользователем после чего выбрать пункт

работа с сотрудниками. Далее необходимо нажать на кнопку добавить нового работника, появится окно, представленное на рисунке 36

Рисунок 36 – Создание нового работника

При создании работника либо клиента появляется возможность добавления данных о контактной информации. Пример окна добавления контактной информации представлен на рисунке 37.

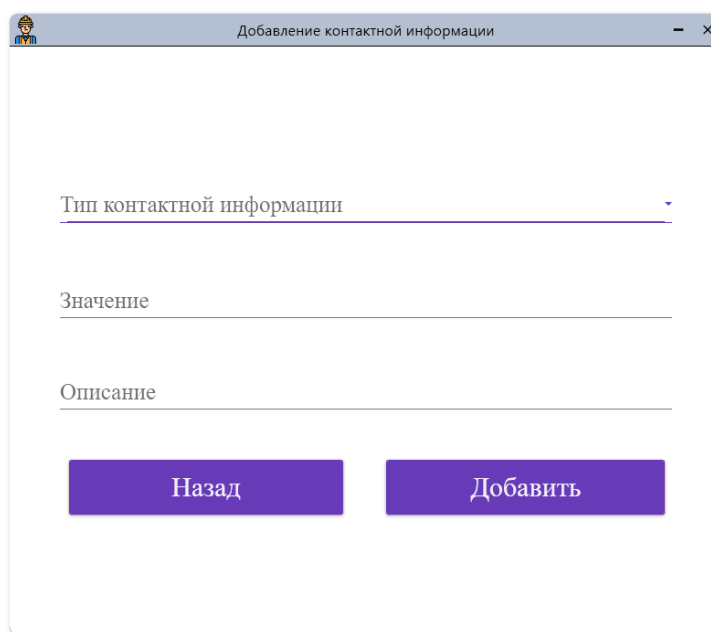
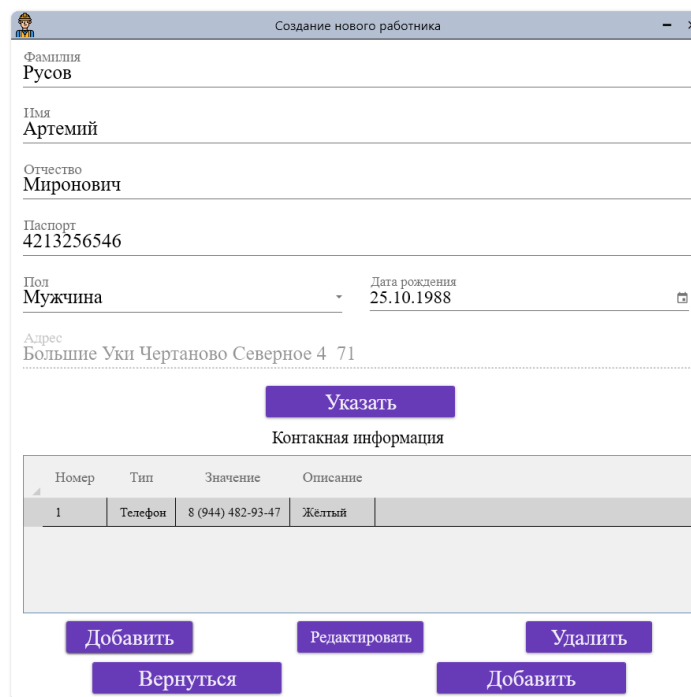


Рисунок 37 – Создание нового контакта

Также, при создании работника добавляется информация об адресе. Окно добавления/ редактирования адреса представлен на рисунке 38.

Рисунок 38 – Создание нового контакта

После заполнения данных о работнике (пример заполнения данных о работнике представлен на рисунке 39), появляется сообщение, представленное на рисунке 40.



Создание нового работника

Фамилия  
Русов

Имя  
Артемий

Отчество  
Миронович

Паспорт  
4213256546

Пол  
Мужчина

Дата рождения  
25.10.1988

Адрес  
Большие Уки Чертаново Северное 4 71

Указать

Контактная информация

Номер	Тип	Значение	Описание
1	Телефон	8 (944) 482-93-47	Желтый

Добавить Редактировать Удалить

Вернуться Добавить

Рисунок 39 – Пример добавления данных о работнике

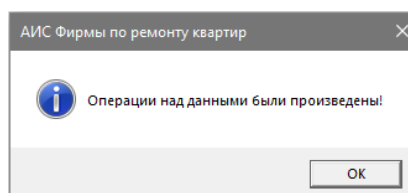
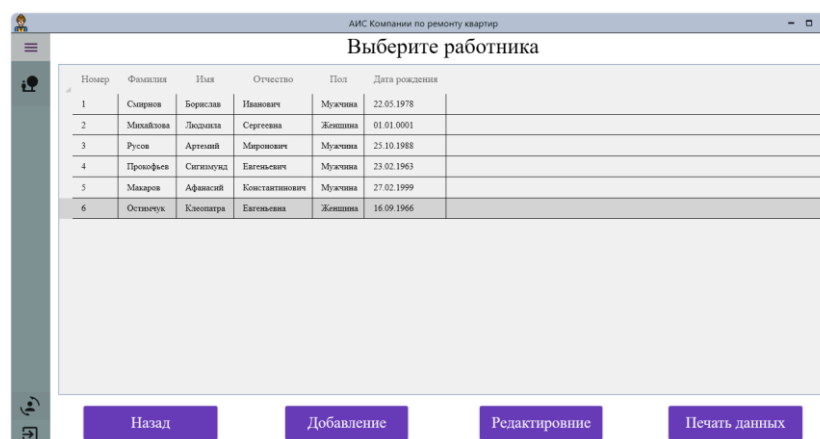


Рисунок 40 – Сообщение о после добавления нового работника

Окно с выбором работников представлен на рисунке 41.



АИС Компании по ремонту квартир

Выберите работника

Номер	Фамилия	Имя	Отчество	Пол	Дата рождения
1	Смирнов	Борислав	Иванович	Мужчина	22.05.1978
2	Михайлова	Людмила	Сергеевна	Женщина	01.01.0001
3	Русов	Артемий	Миронович	Мужчина	25.10.1988
4	Прокофьев	Сигизмунд	Евгеньевич	Мужчина	23.02.1963
5	Макаров	Афанасий	Константинович	Мужчина	27.02.1999
6	Осташук	Клеопатра	Евгеньевна	Женщина	16.09.1966

Назад Добавление Редактирование Печать данных

Рисунок 41 – Сообщение о после добавления нового работника

Также на данном окне можно распечатать данные о работнике в документ WORD. Пример вывода представлен на рисунке 42.

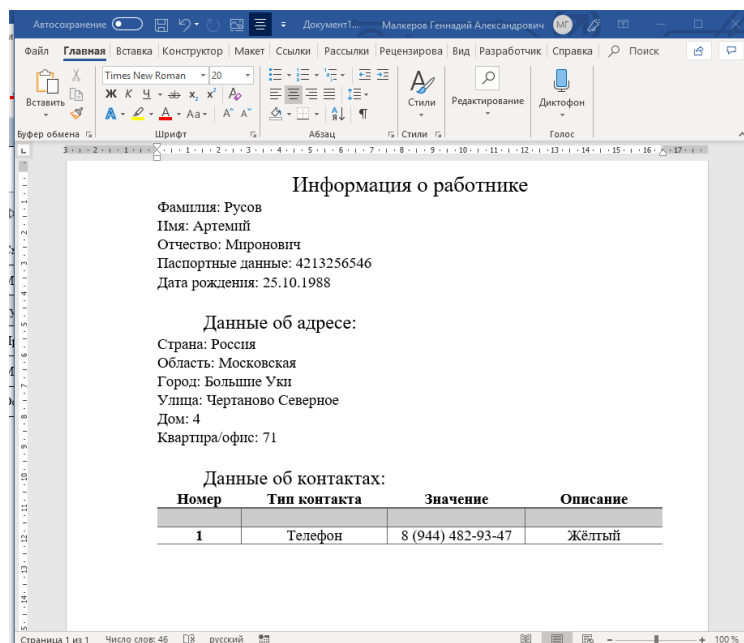


Рисунок 42 – Данные о работнике в WORD

При возврате назад путем нажатия на кнопку назад и выбора пункта «Кадровые перемещения», перед пользователем представляется окно, представленное на рисунке 43.

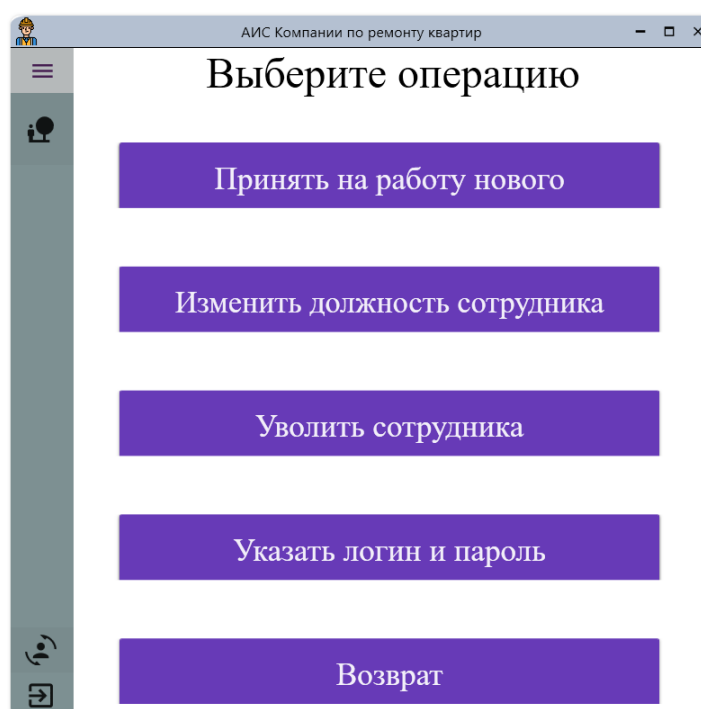


Рисунок 43 – Пункт кадровые перемещения

При выборе пункта «Принятие на работу» перед пользователем представлена информация, пример отображения данного окна представлен на рисунке 44.

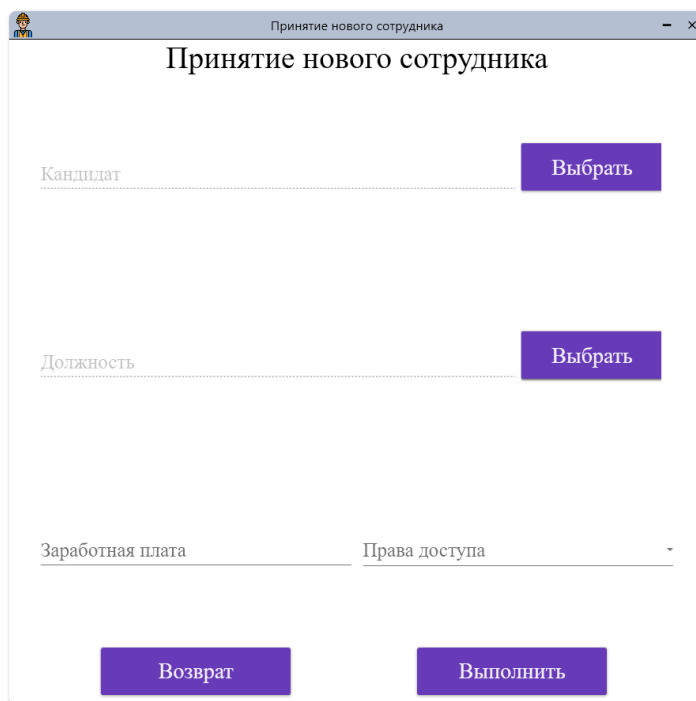
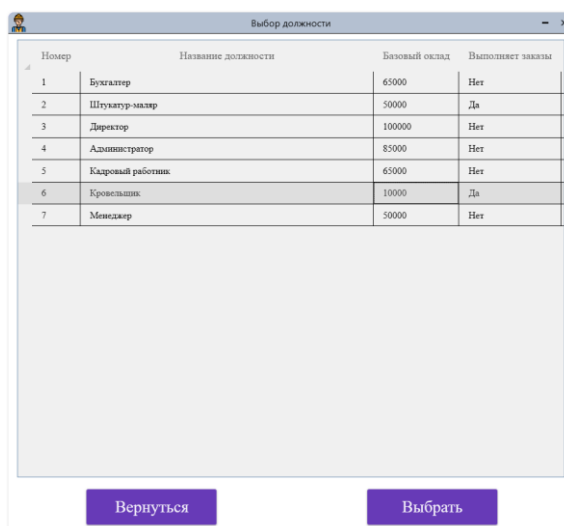


Рисунок 44 – Пункт принятие на работу

При необходимости выбора данных о должностях, перед пользователем представлено появляется окно, представленное на рисунке 45.



Номер	Название должности	Базовый оклад	Выполняет заявки
1	Бухгалтер	45000	Нет
2	Штукатур-маляр	50000	Да
3	Директор	100000	Нет
4	Администратор	85000	Нет
5	Кадровый работник	65000	Нет
6	Кровельщик	10000	Да
7	Менеджер	50000	Нет

Рисунок 45 – Окно выбора должности

После выбора должности и работника, а также указания данных о заработной плате и указания сферы влияния на программу и выбора пункта



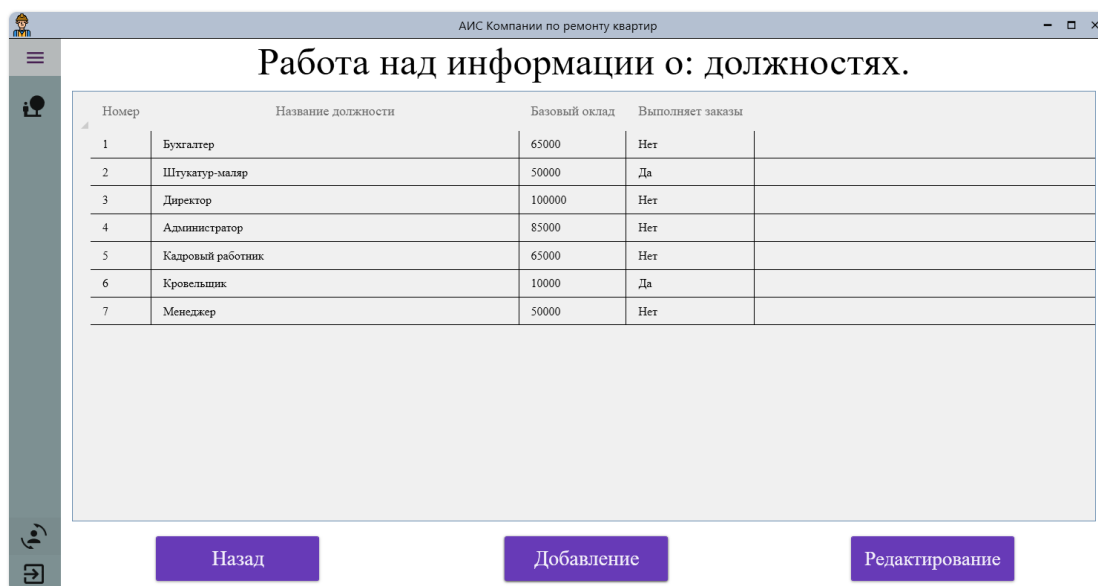
выполнить перед пользователем появляется сообщение, представленное на рисунке 40.

Также в меню, представленном на рисунке 43, можно сделать следующие действия:

- принятие на работу нового сотрудника;
- изменение должности для работающего сотрудника;
- увольнение сотрудника;
- указание логина и пароля для авторизации;

Также помимо выполнения описанных выше пунктов кадровый работник также может редактировать данные о должностях и дополнять их.

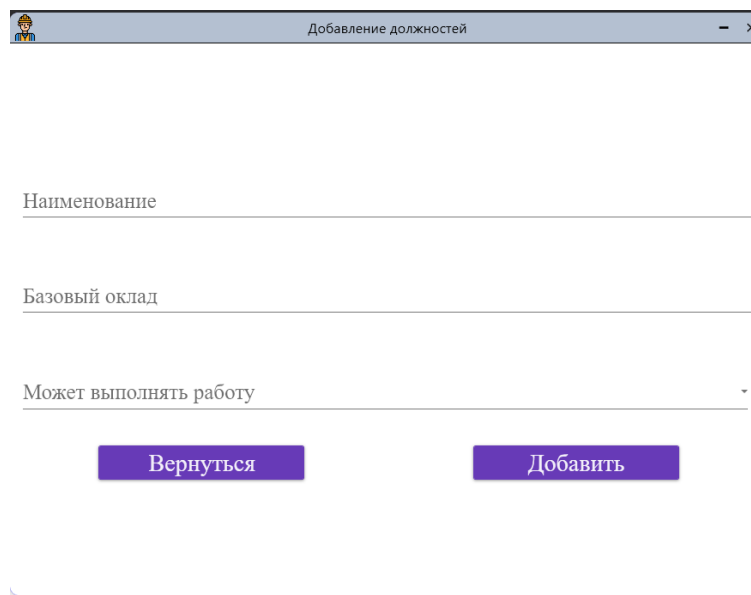
Для выполнения данных операция пользователь должен перейти в пункт «Работа над должностями» меню кадрового работника, перед пользователем появиться окно, представленное на рисунке 46. При выборе пункта добавление перед пользователем появиться окно, представленное на рисунке 47.



Номер	Название должности	Базовый оклад	Выполняет заказы
1	Бухгалтер	65000	Нет
2	Штукатур-маляр	50000	Да
3	Директор	100000	Нет
4	Администратор	85000	Нет
5	Кадровый работник	65000	Нет
6	Кровельщик	10000	Да
7	Менеджер	50000	Нет

Назад      Добавление      Редактирование

Рисунок 46 – Окно просмотра должностей



Добавление должностей

Наименование

Базовый оклад

Может выполнять работу

Вернуться

Добавить

Рисунок 47 – Окно добавления данных о должностях

Роль администратора предполагает, что человек специально имеющий данные права сможет настраивать данную программу при работе в некоторых важных аспектах.

При авторизации в данной роли перед пользователем демонстрируется меню, представленное на рисунке 48. Данному пользователю типу пользователей предоставлен доступ к пункту:

- настройки;
- справочные данные.

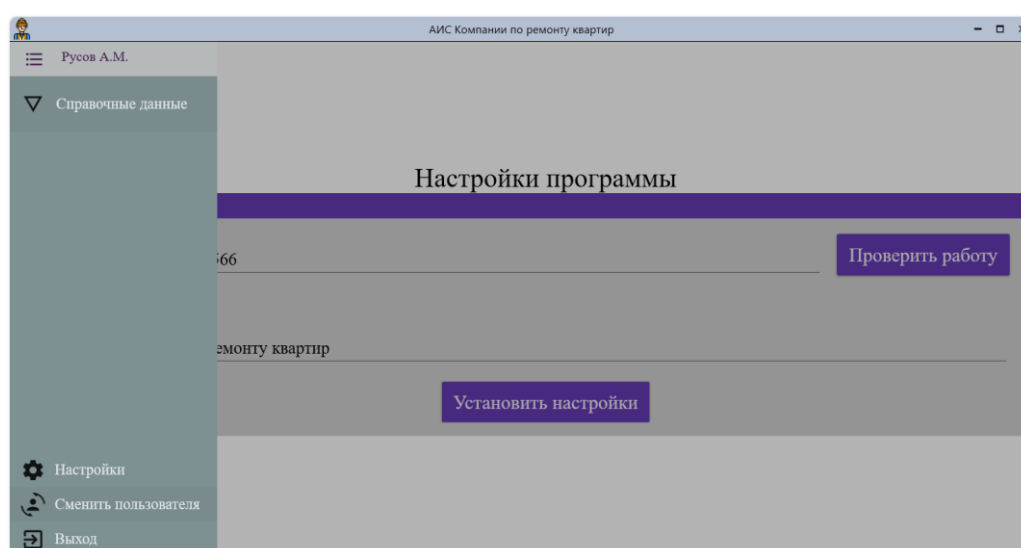


Рисунок 48 – Главное меню приложения при авторизации под правами администратора

Раздел настроек данной программы включает в себя следующие настройки:

- адреса базового сервера для подключения к нему;
- текста заголовка при открытии диалогового окна по типу MessageBox.

Пример отображения данного окна редактирования данных представлен на рисунке 49.

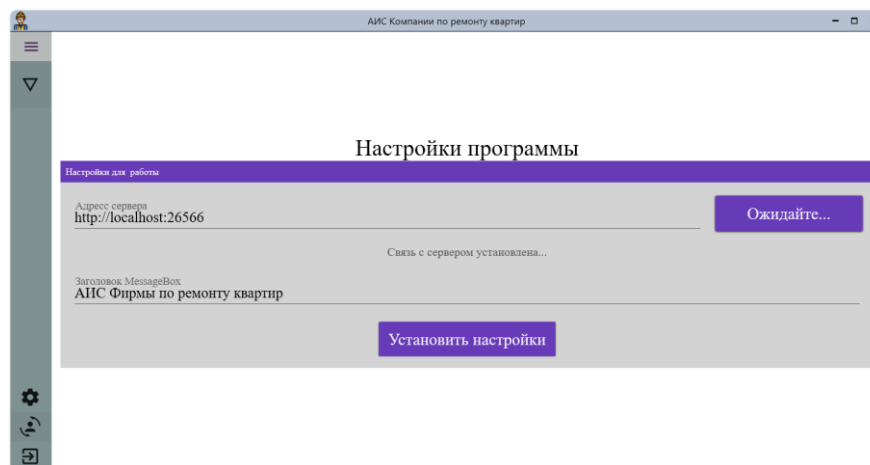


Рисунок 49 – Окно настроек программы

Для сохранения указанных настроек необходимо указать актуальный сервер, для проверки данного факта также имеется соответствующая функция.

Также администратор может открыть окно работы со справочными данными, пример отображения данного окна представлен на рисунке 50. Данная возможность необходима для контролирования работы локальной базы данных.

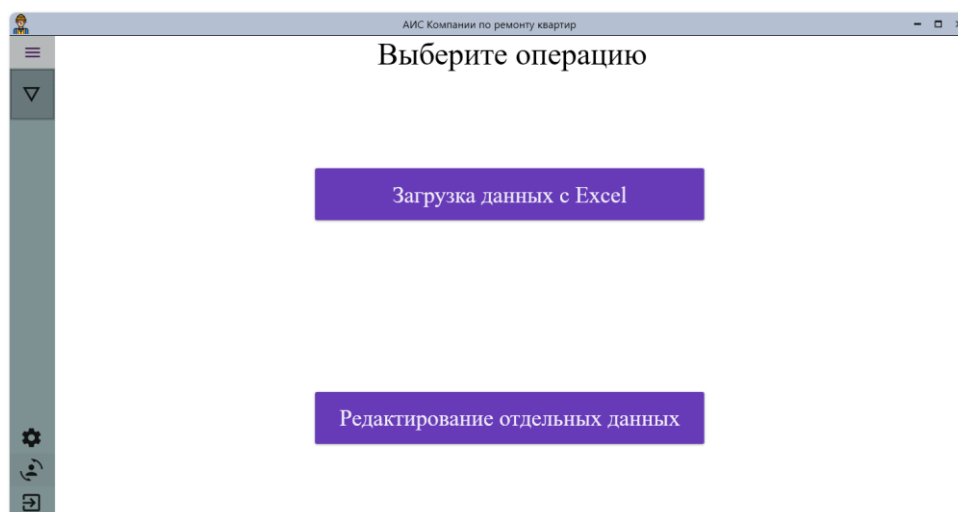


Рисунок 50 – Справочные данные

Далее рассмотрим роль менеджера. Данная роль предусматривает возможность работы:

- с данными клиентов;
- с данными о заказах;
- со справочными данными.

После авторизации пользователя с правами доступа менеджер перед пользователем представляется окно, представленное на рисунке 51.

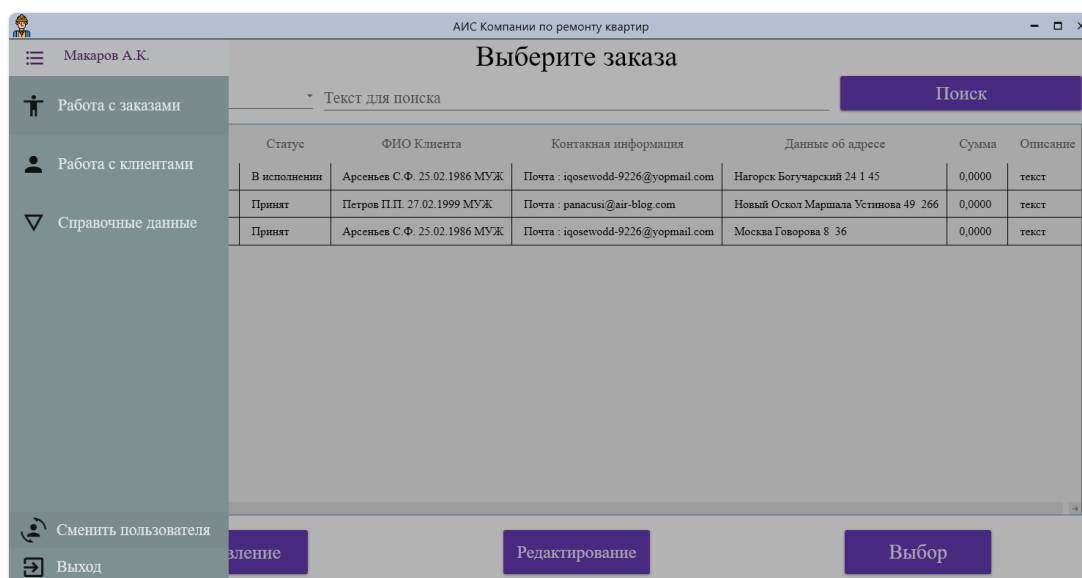


Рисунок 51 – Вид программы при авторизации с правами менеджера

При выборе пункта работа со справочными данными перед пользователем появляется окно, представленное на рисунке 52. При выборе

пункта «Загрузка данных с Excel» перед пользователем появляется окно, представленное на рисунке 53 в котором необходимо выбрать тип данных для загрузки. После выбора пунктов пользователь должен выбрать файл для загрузки, после чего произойдет считывание данных и представление этих данных перед пользователем. Пример считывания данных представлен на рисунке 54. Пример исходного файла представлен на рисунке 55.

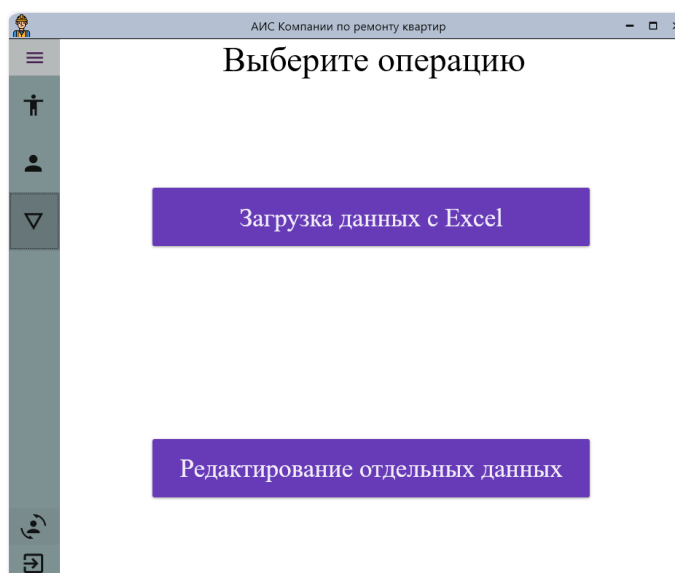


Рисунок 52 – Окно справочные данные

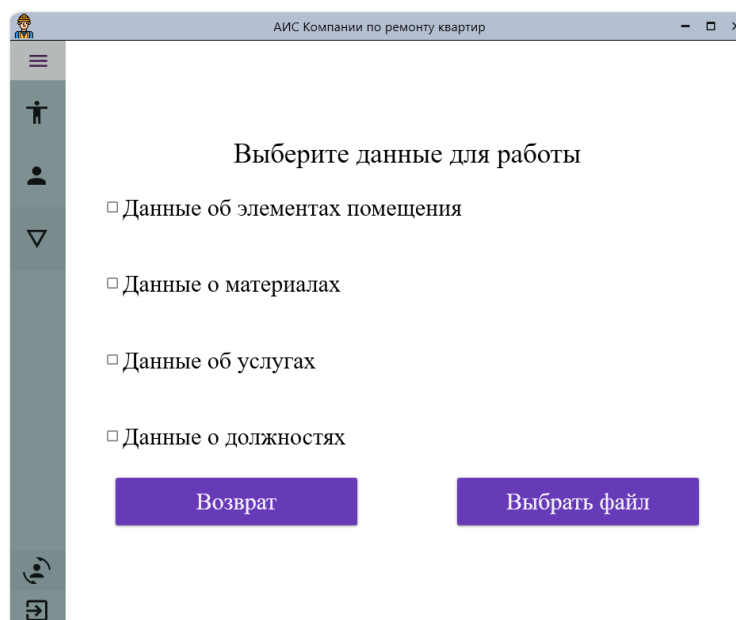


Рисунок 53 – Окно выбора типа справочных данных

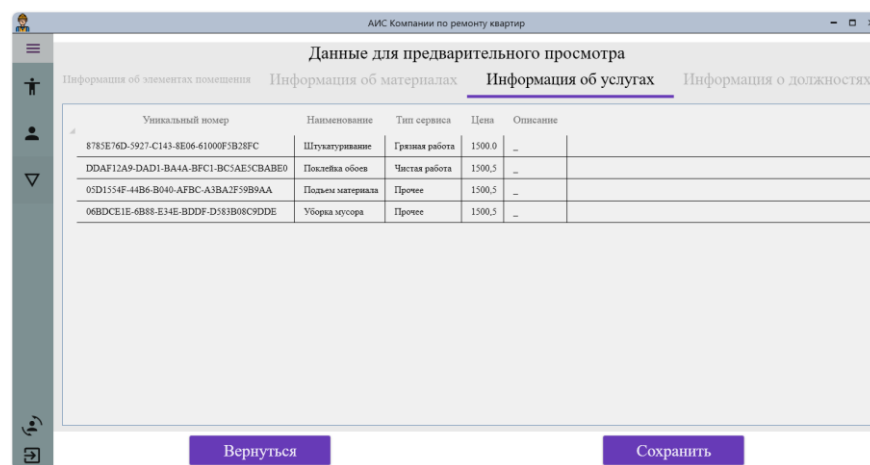


Рисунок 54 – Окно предварительного просмотра данных

Уникальный номер	Наименование	Тип сервиса	Цена	Описание
8785E76D-5927-C143-8E06-61000F5B28FC	Штукатуривание	Грязная работа	1500.0	—
DDAF12A9-DAD1-BA4A-BFC1-BC5AE5CBABE0	Поклейка обоев	Чистая работа	1500,5	—
05D1554F-44B6-B040-AFBC-A3BA2F59B9AA	Подъем материала	Прочее	1500,5	—
06BDCE1E-6B88-E34E-BDDF-D583B08C9DDE	Уборка мусора	Прочее	1500,5	—

Рисунок 55 – Исходный файл Excel

После проверки данных на клиенте их можно загрузить на сервер с помощью кнопки «Сохранить», после чего появиться окно загрузки аналогичное окну при старте программы.

При необходимости работы с клиентами можно перейти в соответствующий пункт бокового меню. С помощью данного меню пользователь может работать с данными о клиентах.

Пример формы создания нового клиента представлен на рисунке 56.

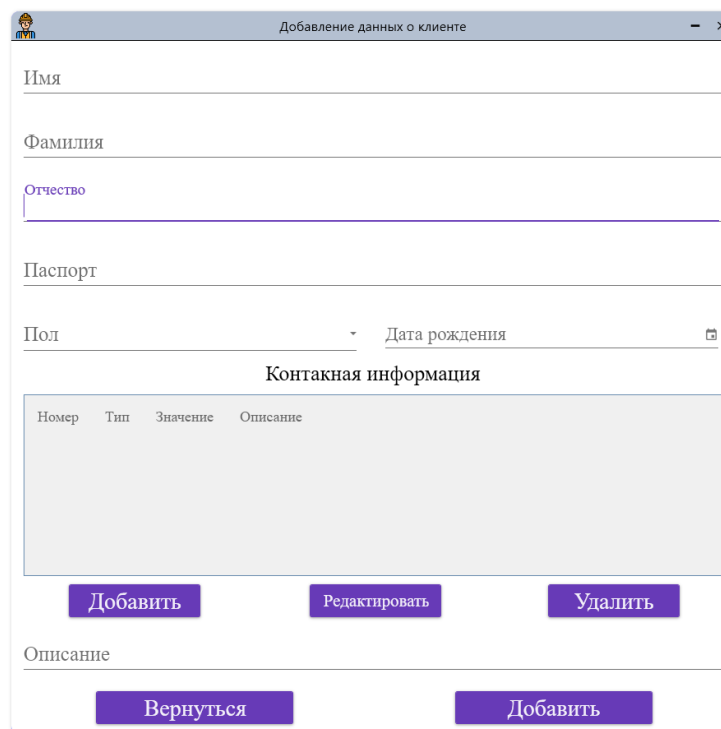


Рисунок 56 – Пример окна по созданию нового пользователя

Для создания заказа необходимо перейти в пункт меню работы с заказами после чего нажать на кнопку создать. Пример формы по созданию заказов представлен на рисунке 57.

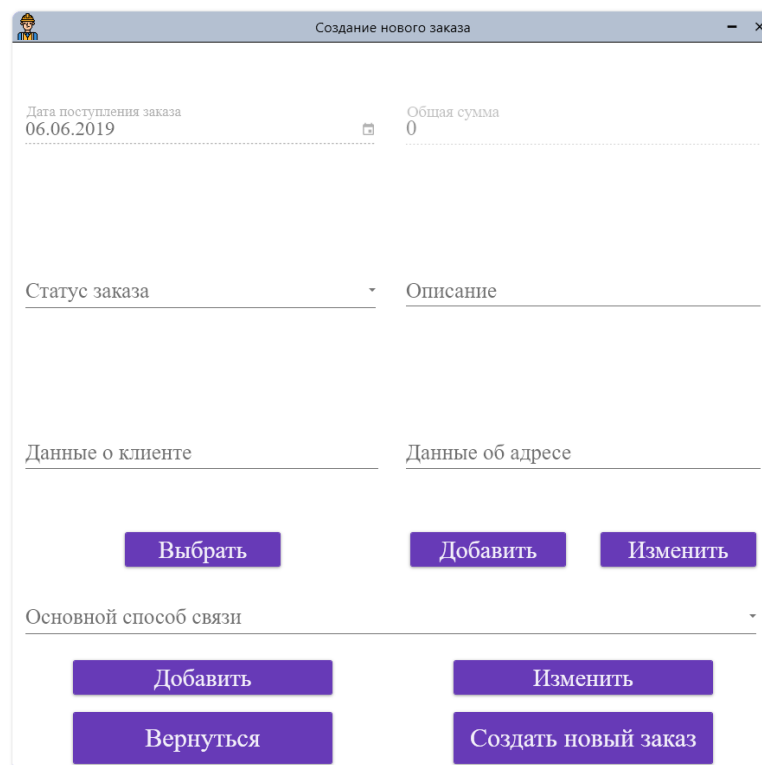


Рисунок 57 – Окно созданию заказа

После создания заказа пользователь видит некоторое сообщение аналогичное сообщению, представленному на рисунке 40. При переходе по средствам кнопки «Выбор» формы выбора заказов, которая представлена на рисунке 58. Пользователю демонстрируется окно, представленное на рисунке 59.

Номер	Дата начала	Статус	ФИО Клиента	Контактная информация	Данные об адресе	Сумма	Описание
1	29.05.2019	В исполнении	Арсеньев С.Ф. 25.02.1986 МУЖ	Почта: arsenovod4-9226@yornmail.com	Нагорск Богучарский 24 1 45	0,0000	текст
2	30.05.2019	Принят	Петров П.П. 27.02.1999 МУЖ	Почта: rpnascini@air-blog.com	Новый Оскол Маршала Устинова 49 266	0,0000	текст
3	29.05.2019	Принят	Арсеньев С.Ф. 25.02.1986 МУЖ	Почта: arsenovod4-9226@yornmail.com	Москва Говорова 8 36	0,0000	текст

Рисунок 58 – Окно выбора заказа для работы

Номер	Наименование	Описание	Длина	Ширина	Высота	P стен	P пола	S стен	S пола
-------	--------------	----------	-------	--------	--------	--------	--------	--------	--------

Рисунок 59 – Окно выбора заказа для работы

На данной форме необходимо поочередно указывать информацию о деталях заказах после завершения заполнения данных пользователь получает возможность напечатать договор и смету по заказам при необходимости.

Данные операции можно выполнить посредством использования вкладки информация пример отображения которой представлен на рисунке 60.



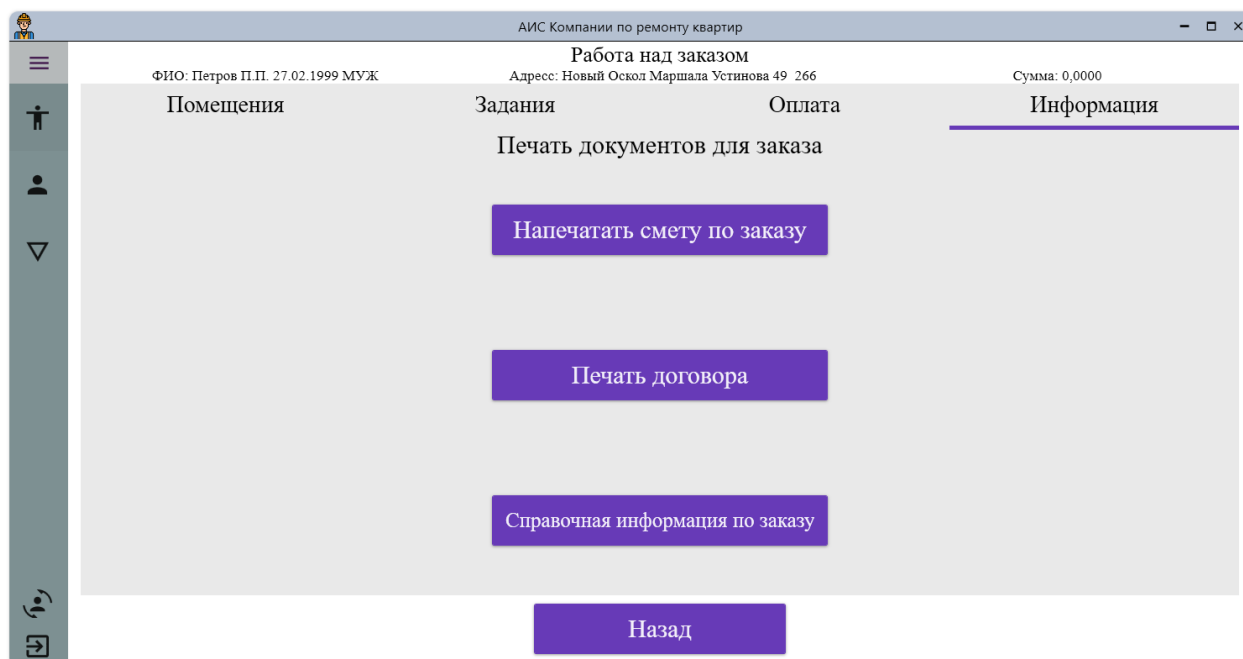


Рисунок 60 – Вкладка «Информация» при работе с заказами

#### 2.5.3.4 Сообщения пользователю

Пример сообщения пользователю представлен на рисунке 61-63.

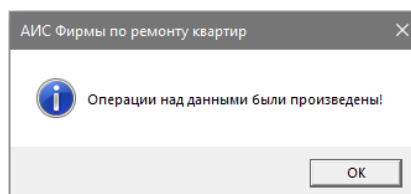


Рисунок 61 – Сообщение пользователю

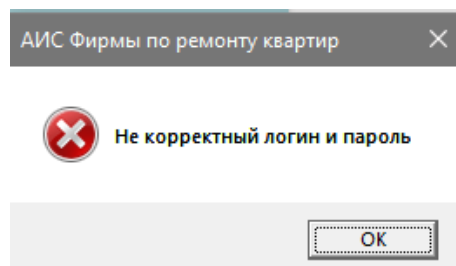


Рисунок 62 – Сообщение пользователю

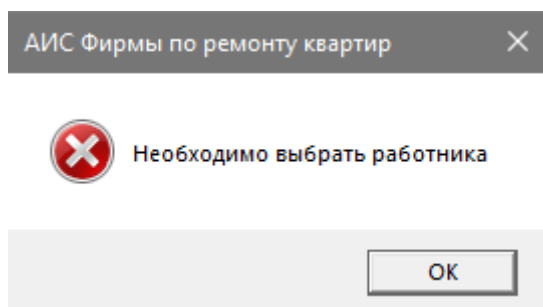


Рисунок 63 – Сообщение пользователю

## Заключение

В ходе разработки выпускной квалификационной работы было написано клиент-серверное приложение «Автоматизированная информационная система фирмы по ремонту квартир». При подготовке к написанию выпускной квалификационной работы была изучена предметная область разрабатываемого проекта, а также изучена специализированная литература, которая необходима была при написании данной выпускной квалификационной работы.

После изучения предметной области были составлены схемы баз данных, которые используются в качестве способа хранения информации. При разработке базы данных были использованы некоторые специализированные средства, которые также были изучены.

После составления схем баз данных была начата разработка серверной части продукта, которая как раз и использовала основную базу данных.

Параллельно с разработкой серверной части было начато создание интерфейсов клиентского приложения, с которым и взаимодействует пользователь.

Вместе с разработкой было выполнено тестирование программного продукта, что позволило выявить большое количество недостатков и недочетов, но они все были исправлены.

- среди достоинств программы можно считать следующие:
- приятный глазу и удобный пользователю интерфейс;
- полная настройка добавленных опций в заказ;
- вывод отчетов в Word;
- импорт данных пользователя из Excel;
- простота дальнейшего расширения данной программы;
- клиент-серверная технология работы программы.

Тесты данной программы показали достаточную надежность и корректность работы программы, что позволяет использовать данное

программно обеспечение в реальной компании занимающейся организацией ремонтов квартир.

## Список использованных источников

### *Законодательные и нормативные акты*

1. ГОСТ 2.316-2008 Правила нанесения надписей, технических требований и таблиц на графических документах.
2. ГОСТ 7.1. – 2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2004. – 169 с.
3. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание.
4. ГОСТ 7.32 – 2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: ИПК Издательство стандартов, 2001. – 21 с.
5. ГОСТ\_2.105-95 Общие требования к текстовым документам
6. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с.

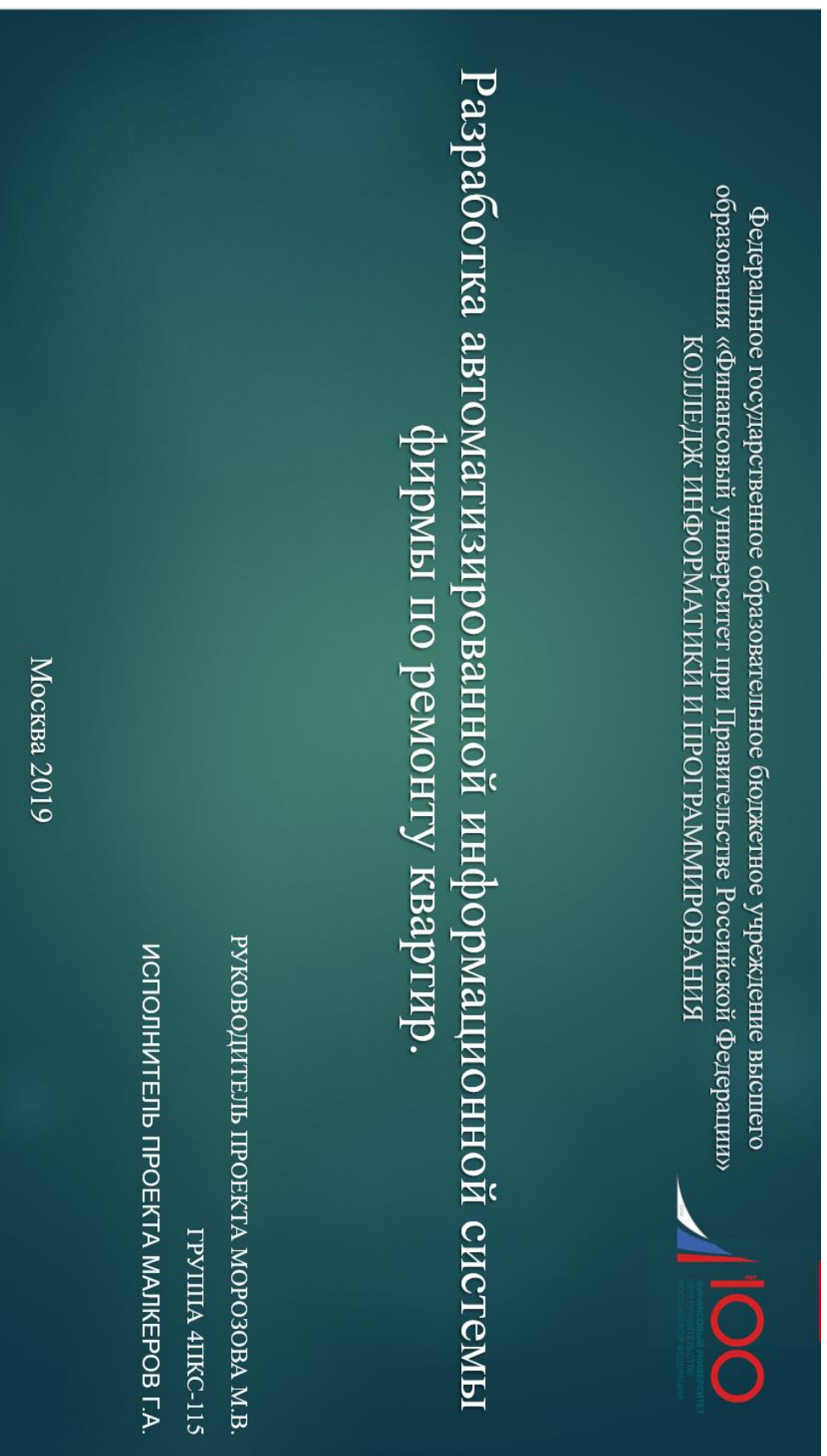
### *Интернет-документы*

7. Рынок ремонтно-строительных работ: тенденции и прогнозы – [Электронный ресурс]. – Режим доступа: <https://www.vira.ru/exp/news/detail.php?ID=48005>
8. Простой Софт - разработка программ– [Электронный ресурс]. – Режим доступа: <http://www.prostoysoft.ru/index.htm>
9. Конфигурация "Ремонт квартир" – [Электронный ресурс]. – Режим доступа: <http://www.prostoysoft.ru/FlatRepair.htm>
10. Draw.io – [Электронный ресурс]. – Режим доступа: <https://www.draw.io/>
11. DB Browser for SQLite – [Электронный ресурс]. – Режим доступа: <https://sqlitebrowser.org/>
12. SQL Server 2017 — Выпуски – [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017-editions>

13. Скачивание SQL Server Management Studio (SSMS) – [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>
14. Описание среды программирования VS – [Электронный ресурс]. – Режим доступа: <https://studfiles.net/preview/4114690/>
15. Скачать приложение Postman – [Электронный ресурс]. – Режим доступа: <https://www.getpostman.com/downloads/>
16. C# Язык программирования – [Электронный ресурс]. – Режим доступа: <http://progopedia.ru/language/csharp/>
17. SQL Язык программирования – [Электронный ресурс]. – Режим доступа: <http://progopedia.ru/language/sql/>
18. Введение в Entity Framework 6– [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/entityframework/1.1.php>

## Приложение А

Титульный лист презентации к защите ВКР:



## Приложение Б

Код программы

