

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»
Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Конструирование программного обеспечения (КПО)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

«АРМ работника ЖЭС»

Студент:

гр. 381573 Ридецкая А.М.

Руководитель:

ассистент кафедры ИСиТ ИИТ БГУИР
Медведев С.А.

Минск 2024

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

УТВЕРЖДАЮ

Заведующий кафедрой ИСиТ

(подпись)

2024 г.

ЗАДАНИЕ
на курсовой проект

студенту Ридецкой Анне Михайловне

1. Тема проекта «АРМ работника ЖЭС»

2. Срок сдачи студентом законченного проекта _____

3. Исходные данные к проекту

Тип операционной системы – ОС Windows 10 и выше; язык
программирования – C#; среда разработки – Microsoft Visual Studio; вид
программы – Windows Form;

4. Содержание пояснительной записки

Введение

1. Постановка задачи

2. Программная реализация

3. Руководство пользователя

Заключение

Список используемых источников

Приложение А. (обязательное) Листинг программы

5. Перечень графического материала

1. "АРМ работника ЖЭС", схема программы, чертеж – формат А3, лист 1.

2. "Алгоритм пользователя", схема алгоритма, чертеж – формат А3, лист 1.

3. "Алгоритм работника", схема алгоритма, чертеж – формат А3, лист 1.

6. Консультант по курсовому проекту Медведев С.А.

7. Дата выдачи задания 02.09.2024 г.

8. Календарный график работы над курсовым проектом на весь период
(с обозначением сроков выполнения и процентом от общего объёма работы):

№	Содержание работ	Срок выполнения	% от общего объёма работы
1	Раздел 1	16.09.2024	25 %
2	Раздел 2	30.09.2024	50 %
3	Раздел 3	21.10.2024	75 %
5	Оформление пояснительной записки и графического материала	18.11.2024	100 %
6	Защита курсового проекта		

РУКОВОДИТЕЛЬ С.А. Медведев
(подпись)

Задание принял к исполнению А. М. Ридецкая
(дата и подпись студента)

РЕФЕРАТ

АРМ РАБОТНИКА ЖЭС: курсовой проект / А. М. Ридецкая – Минск: БГУИР, 2024, – п.з. – 44 с., плакатов – 3 л. Формата А3.

В данном курсовом проекте рассматривается разработка программного обеспечения для удобного и функционального будильника на платформе Windows 10 и выше, реализованного с использованием языка программирования C# и среды разработки Microsoft Visual Studio. Основной целью работы является создание приложения для автоматизации рабочего места работников жилищно-эксплуатационной служба.

Основное внимание уделено удобству использования и интуитивно понятному интерфейсу, что делает процесс взаимодействия пользователя с приложением максимально простым. Программная реализация основана на монолитной архитектуре, что позволяет интегрировать все функциональные компоненты в единую систему, упрощая разработку и тестирование.

Разработанная «АРМ работника ЖЭС» представляет собой полезный инструмент для работников ЖЭС, позволяя им управлять заявками, контролировать выполнение работ и минимизировать количество ошибок при учете.

Ключевые слова: АРМ работника ЖЭС, алгоритмы, C#, Visual Studio
2022

СОДЕРЖАНИЕ

Введение	5
1 Постановка задачи.....	6
1.1 Описание задачи.....	6
1.2 Блок-схема программы	7
2 Программная реализация	10
2.1 Краткая характеристика инструментальной программной среды.....	10
2.2 Структура программы.....	11
2.3 Описание программных модулей.....	13
3 Руководство пользователя	15
Заключение	27
Список используемых источников.....	28
Приложение А (обязательное) Листинг программы	29

ВВЕДЕНИЕ

В условиях современного общества управление жилищно-эксплуатационными службами (ЖЭС) требует высокоэффективных решений. Работники ЖЭС сталкиваются с различными задачами, связанными с обслуживанием жилых фондов, и автоматизация процессов может значительно улучшить качество их работы и повысить уровень обслуживания населения.

Работники Жилищно-эксплуатационных служб (ЖЭС) выполняют множество задач, направленных на обеспечение качественного обслуживания жилых зданий и комфортных условий проживания для граждан. Вот основные задачи, с которыми они сталкиваются:

- регулярные осмотры: проведение плановых проверок состояния инженерных систем (водоснабжение, отопление, электроснабжение);
- профилактика: выполнение профилактических работ для предотвращения аварийных ситуаций и поломок оборудования;
- капитальный и текущий ремонт: организация и выполнение ремонтных работ в жилых помещениях и общем имуществе (крыши, фасады, подъезды);
- устранение аварийных ситуаций: быстрое реагирование на аварии, такие как протечки, отключения электроэнергии;
- прием заявок: регистрация и обработка заявок от жильцов на выполнение работ или устранение неисправностей;
- контроль выполнения: мониторинг выполнения заявок и информирование жильцов о статусе их обращений;
- консультирование: предоставление жильцам информации по вопросам эксплуатации жилья, прав и обязанностей;
- организация собраний: участие в общих собраниях жильцов для обсуждения насущных вопросов и проблем;
- учет работ: ведение отчетности по выполненным работам, расходам и доходам;
- документация: оформление необходимых документов, связанных с техническим обслуживанием и ремонтом;
- уборка и содержание территории: обеспечение чистоты и порядка на придомовых территориях, организация уборки снега, вывоз мусора;
- благоустройство: работа по озеленению и благоустройству придомовых участков;
- распределение задач: назначение задач между работниками ЖЭС в зависимости от их квалификации и загруженности;
- контроль выполнения: мониторинг выполнения работниками своих обязанностей и качества их выполнения.

Разработка АРМ работника ЖЭС значительно повысит эффективность процессов в данной сфере, улучшит качество обслуживания граждан и обеспечит более прозрачное управление задачами и ресурсами.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Описание задачи

Целью является разработка АРМ работника ЖЭС.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить порядок работы и учет задач работников ЖЭС;
- выполнить проектирование системы;
- апробировать разработанную автоматизированную систему.

Результатом проекта станет автоматизированная система управления работой работников ЖЭС, которая обеспечит:

- упрощение работы с данными для администратора;
- работнику принимать заявки, просматривать задачи на день;
- пользователю подавать заявки;
- возможность создания акта выполненных и назначенных работ;
- эффективное распределение задач и мониторинг их выполнения;
- сокращать количество ошибок учета и неполноты сведений;
- добавлять, редактировать и удалять информацию;
- просматривать данные с возможностью фильтрации и поиска;
- разграничивать права доступа по ролям: «Администратор», «Работник», «Пользователь»;
- проводить анализ данных и предоставлять статистическую отчетность.

Работа с заявками в ЖЭС должна быть интуитивно понятной и удобной для пользователей, обеспечивая гибкость и широкие возможности для управления задачами. Важно, чтобы система функционировала стабильно, позволяя работнику эффективно принимать заявки и просматривать свои задачи на день.

Работник ЖЭС может принимать заявки, создавать акты выполненных и назначенных работ, а также эффективно распределять задачи и отслеживать их выполнение. Это упрощает работу с данными и помогает сократить количество ошибок учета и неполноты сведений.

Система должна предоставлять возможность добавления, редактирования и удаления информации, а также просмотра данных с функциями фильтрации и поиска. Разграничение прав доступа по ролям — «Администратор», «Работник», «Пользователь» — обеспечивает безопасность и контроль над процессами.

Кроме того, система должна поддерживать анализ данных и предоставлять статистическую отчетность, что поможет в принятии обоснованных решений и планировании работы ЖЭС.

1.2 Блок-схема программы

Блок-схема является удобным и наглядным инструментом для записи алгоритма. Это графическое представление последовательности операций, необходимых для решения задачи или выполнения алгоритма. Она состоит из различных блоков, представляющих действия или операции, и стрелок, указывающих порядок их выполнения. Блок-схемы находят применение в проектировании программного кода, описании бизнес-процессов, а также в анализе и оптимизации работы систем и процессов [1].

На рисунке 1.1 предоставлена блок-схема алгоритма пользователя.

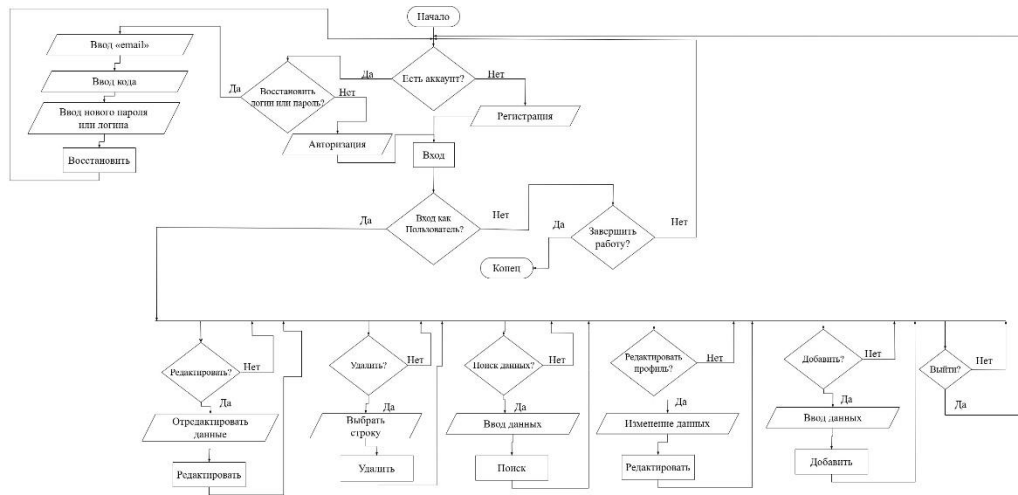


Рисунок 1.1 — Блок-схема алгоритма «Алгоритм пользователя»

Описание работы алгоритма:

- создаем аккаунт или входим в уже существующий, или восстанавливаем доступ;
- выбираем функцию;
- проводим работу с данными;
- ВЫХОД.

На рисунке 1.2 предоставлена блок-схема алгоритма работника

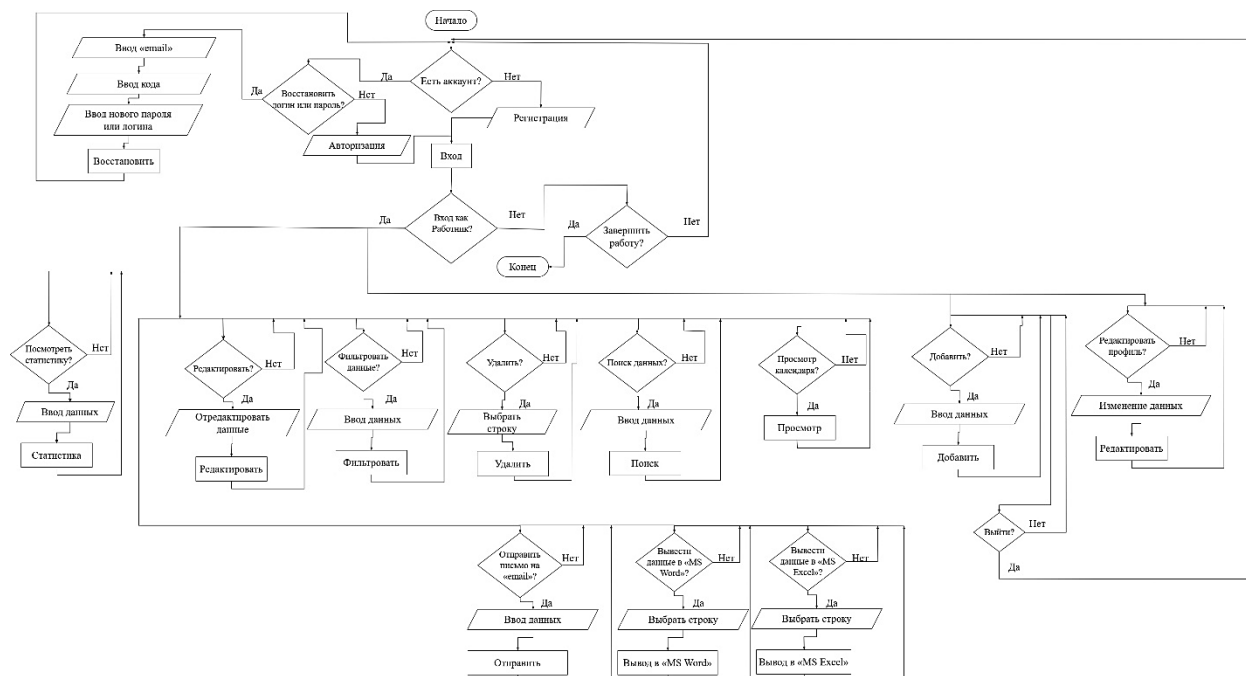


Рисунок 1.2 — Блок-схема алгоритма «Алгоритм работника»

Описание работы алгоритма:

- создаем аккаунт или входим в уже существующий, или восстанавливаем доступ;
- выбираем функцию;
- проводим работу с данными;
- ВЫХОД.

На рисунке 1.3 предоставлена блок-схема программы «АРМ работника ЖЭС».

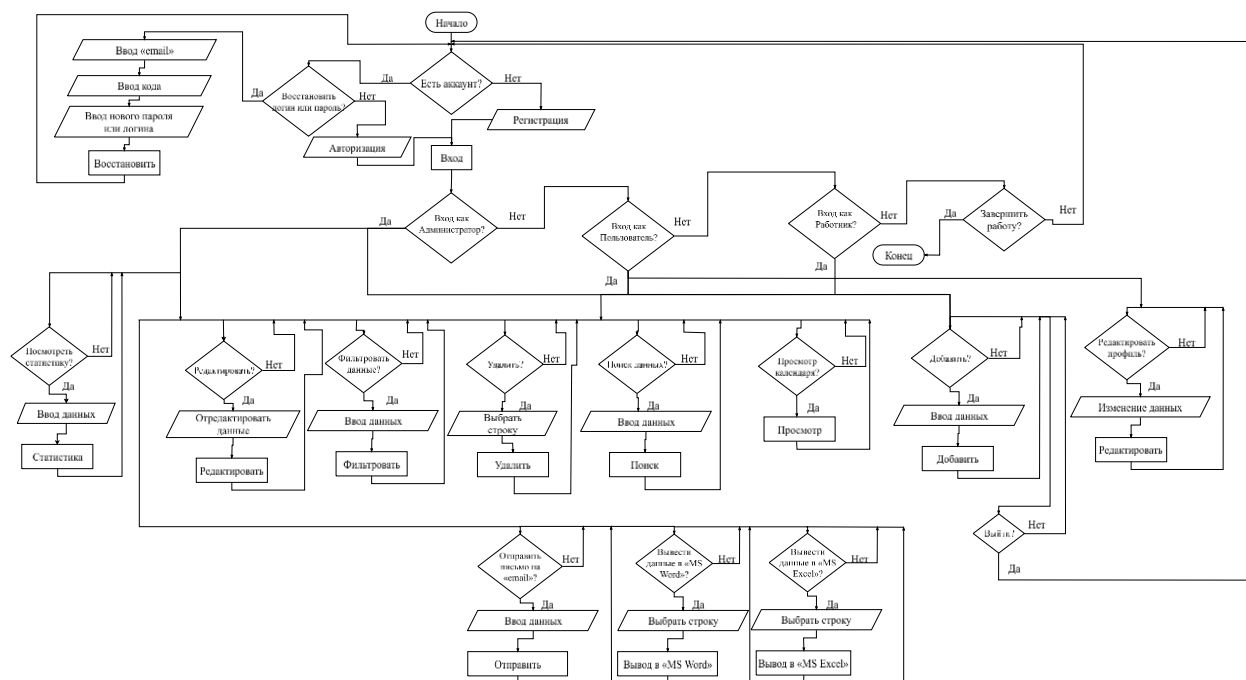


Рисунок 1.3 — Блок-схема программы «АРМ работника ЖЭС»

Алгоритм программы начинается с создания аккаунта или входа в уже созданный. Так же можно восстановить доступ к аккаунту. Далее функционал программы зависит от уровня доступа.

Уровень доступа «Пользователь» имеет функционал:

- редактирование профиля;
- просмотр данных таблиц «услуги», «объекты», «оплата», «заявка», «отзыв»;
- поиск;
- добавление, редактирование, удаление заявки и отзыва;
- оплата выполненной заявки.

Уровень доступа «Пользователь» имеет функционал:

- редактирование профиля;
- просмотр данных таблиц «услуги», «объекты», «оплата», «заявка», «отзыв», «задачи»;
- редактирование, удаление заявок;
- поиск и фильтрация данных;
- прием заявок;
- вывод данных в «Word» и «Excel»;
- просмотр статистических данных, календаря задач;
- отправка акта выполненных работ на почту пользователя.

Уровень доступа «Администратор» имеет функционал:

- просмотр данных таблиц «услуги», «объекты», «оплата», «заявка», «отзыв», «задачи», «пользователь», «работник»;
- поиск и фильтрация данных;
- добавление, редактирование и удаление всех данных;
- просмотр статистических данных, календаря задач;
- вывод данных в «Word» и «Excel»;
- отправка акта выполненных работ на почту пользователя и работника.

2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

2.1 Краткая характеристика инструментальной программной среды

Для разработки «АРМ работника ЖЭС» использован язык программирования «С#».

«С#» — это объектно-ориентированный язык программирования [2].

Следующие особенности, которые определили его как средство разработки:

- поддержание полиморфизма;
- поддержание перегрузки операторов;
- использовании делегатов, атрибутов, событий, обобщенных типов и анонимных функций.

«С#» предоставляет широкие возможности работы с платформой «.Net» и имеет большое количество библиотек и шаблонов [3].

Для организации хранилища данных автоматизированной системы определена система управления реляционными базами данных «Microsoft SQL Server».

«Microsoft SQL Server» — система управления реляционными базами данных, разработанная корпорацией «Microsoft». Основным используемый язык запросов — «Transact-SQL», создан совместно «Microsoft» и «Sybase». «Transact-SQL» является реализацией стандарта ANSI/ISO по структурированному языку запросов с расширениями.

Преимущества «Microsoft SQL Server»:

- высокоскоростной доступ к данным, обеспечиваемый надежной клиент-серверной архитектурой системы управления базы данных;
- возможность работы с приложениями «Microsoft Excel» и «Microsoft Word»;
- простота работы и администрирования, обусловленные понятной структурой языка «SQL»;
- безопасность хранения информации в базе данных — благодаря возможности шифрования данных и резервного копирования [4].

К достоинствам реляционных систем управления базами данных можно отнести:

- наличие развитой теории реляционной модели данных;
- возможность и наличие аппарата сведения других моделей данных к реляционной;
- поддержка индексирования и кэширования данных;
- возможность манипуляции данными без необходимости знаний физической организации базы данных [5].

Для работы с «Microsoft SQL Server» может использоваться «Microsoft SQL Server Management Studio». «Microsoft SQL Server Management Studio» — утилита для конфигурирования, управления и администрирования всех компонентов «Microsoft SQL Server». Утилита включает скриптовый редактор

и графическую программу, которая работает с объектами и настройками сервера [6].

Для разработки «АРМ работника ЖЭС» использована среда разработки «Microsoft Visual Studio 2022».

«Microsoft Visual Studio» — это лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для «Windows», «Mac», «Linux», «iOS» и «Android» [7].

Преимущества «Microsoft Visual Studio 2022»:

- редактирование, профилирование и отладка кода;
- нумерация строк в редакторе;
- поддержка языков «Visual Basic», «C#», «C++», «Java Script», «HTML5», «Python»;
- автоматическое завершение кода и подсказки;
- система скачиваемых расширений для дополнительных функций;
- встроенные тесты производительности;
- редактируемое меню внешних инструментов.

Таким образом, технологиями разработки выбраны язык программирования «C#», система управления базами данных «Microsoft SQL Server» и среда разработки «Microsoft Visual Studio 2022». Данные технологии являются наиболее подходящими для реализации поставленных задач.

2.2 Структура программы

Программа разрабатывается с использованием языка программирования «C#» в среде «Visual Studio 2022» и использует базу данных «Microsoft SQL Server» для хранения информации.

Монолитная архитектура — это подход, при котором все компоненты приложения объединены в одну единую структуру, что делает систему проще для разработки на начальных этапах, но затрудняет масштабирование и гибкость в дальнейшем. В программе также используется монолитная архитектура. Это проявляется в том, что все основные компоненты, такие как взаимодействие с базой данных, обработка бизнес-логики и пользовательский интерфейс, интегрированы в одну систему [8].

Одна из главных особенностей монолитной архитектуры — тесная связь компонентов. В программе все бизнес-операции, такие как работа с заявками, оплатами и профилями, выполняются в одном классе. Например, класс «Pays» выполняет сразу несколько задач: отображает данные из базы данных, взаимодействует с интерфейсом пользователя и управляет логикой, связанной с заявками и оплатами. Всё это интегрировано в одном месте, что облегчает разработку и тестирование, однако затрудняет внесение изменений и масштабирование системы.

Программа использует единую кодовую базу, где все функциональные компоненты (обработка данных, пользовательский интерфейс, доступ к базе данных) находятся в одном проекте. В частности, методы работы с базой данных, такие как выполнение SQL-запросов для получения данных о заявках или оплатах, реализованы внутри класса. Это упрощает процесс разработки, так как все изменения в логике приложения происходят в одном месте.

В монолитных приложениях управление состоянием обычно осуществляется внутри одного процесса. В программе проявляется в том, что состояние интерфейса, видимость панелей или кнопок, управляется в одном месте, в классе. Видимость панели для добавления оплаты или возможность выбрать заявку контролируются непосредственно внутри этого класса, без необходимости взаимодействовать с отдельными сервисами или процессами.

Еще одной характеристикой монолитной архитектуры является прямое взаимодействие с базой данных. В программе доступ к базе данных осуществляется прямо из методов классов, таких как «Pays_load», где выполняются SQL-запросы для получения данных о платежах и заявках. Все операции — от получения информации о заявках до записи данных в таблицы базы данных — происходят в рамках одного класса.

База данных представлена на рисунке 2.1.

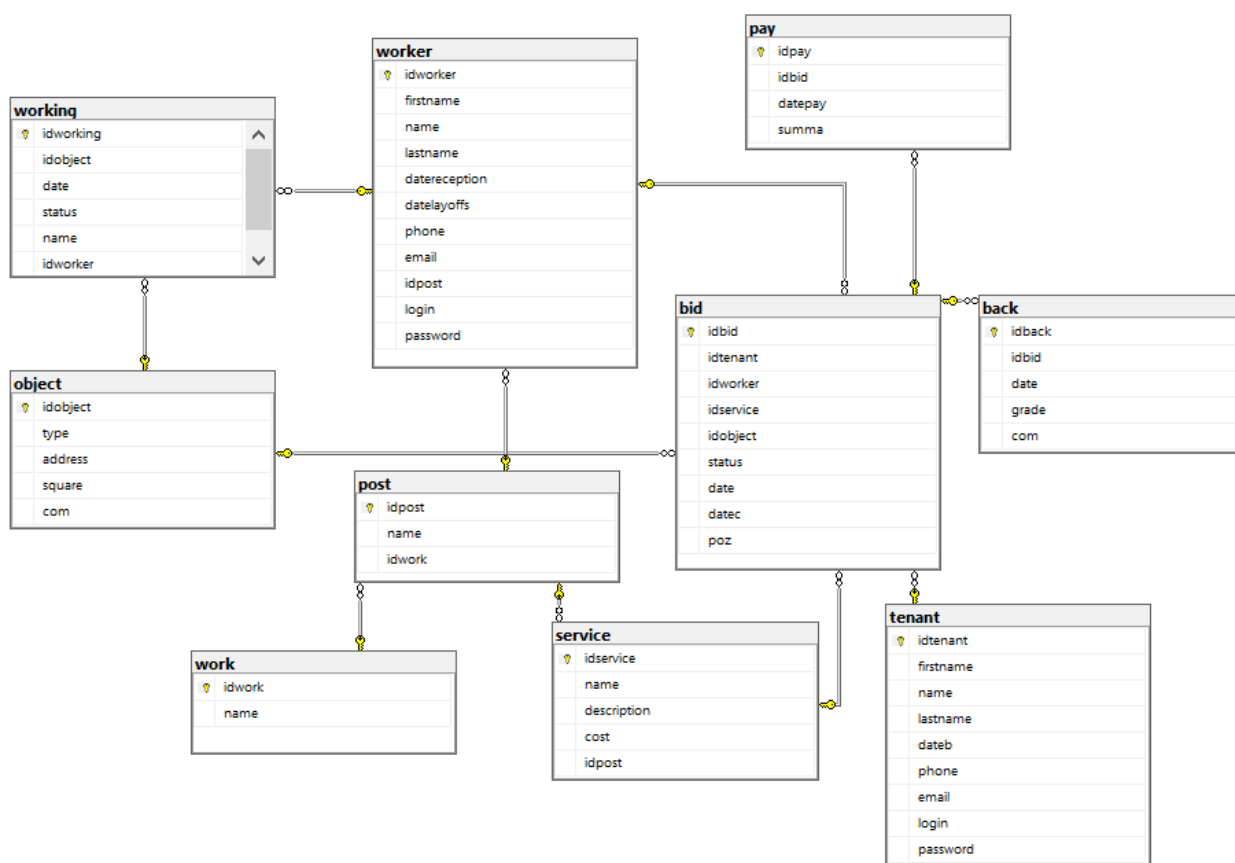


Рисунок 2.1 — База данных

Листинг программы указан в Приложении А.

2.3 Описание программных модулей

Программа построена так, чтобы каждая её часть выполняла свою роль, но все модули работают как единое целое. Программа поделена на несколько модулей:

- модуль «Вход»;
- модули «Работы с данными»;
- модуль «Статистических данных»;
- модуль «Календарь»;
- Модуль «Обработки ошибок и уведомлений»;
- Модуль «Управление уведомлениями и отправка сообщений».

Первый модуль предназначен для управления процессами регистрации и восстановления доступа пользователей. Он обеспечивает удобный интерфейс для входа в систему, регистрации новых пользователей и восстановления забытых паролей или логинов. Модуль включает в себя обработку пользовательского ввода, валидацию данных и отправку необходимых уведомлений по электронной почте. Его главная задача — обеспечить безопасность и удобство работы с системой.

Второй модуль предназначен для управления данными обеспечивает пользователям возможность добавления и редактирования различных записей. Пользователи могут легко добавлять новые записи, выбирая необходимые параметры из выпадающих списков, а также редактировать существующие данные через интуитивно понятный интерфейс, где требуется указать основные атрибуты, такие как название и связь с другими элементами. Удаление записей также реализовано с подтверждением действия, что минимизирует риск случайных ошибок. Важной частью функционала является валидация данных, которая проверяет наличие пустых полей и уникальность вводимых значений, предотвращая дублирование записей. После выполнения операций интерфейс автоматически обновляется, обеспечивая пользователей актуальной информацией. Удобство поиска и навигации реализовано через возможность быстро находить нужные записи в таблицах, что значительно упрощает работу с данными и повышает общую эффективность взаимодействия с формой.

Третий модуль предназначен для анализа статистических данных и представления их в графическом виде. Он позволяет пользователю выбирать диапазон дат для анализа выполненных задач и отображает результаты в виде графиков. Модуль включает в себя загрузку данных из базы данных, их обработку и визуализацию, что помогает пользователям принимать обоснованные решения на основе анализа статистики.

Четвёртый модуль фокусируется на визуализации данных в виде календаря. Он позволяет пользователю просматривать и управлять задачами, связанными с работой и услугами, а также осуществлять поиск по заданным критериям. Модуль предоставляет удобный интерфейс для навигации по месяцам и отображает важную информацию о задачах в удобном формате.

Пятый модуль отвечает за вывод ошибок и уведомлений пользователю в случае неправильного ввода данных (например, неверный логин или пароль), а также уведомляет о успешных операциях, таких как смена пароля или логина. Модуль управляет состоянием сообщений, информируя пользователя о проблемах или успехах.

Шестой модуль отвечает за управление уведомлениями и отправку писем пользователям, таким как пользователи и работники. Он отображает данные о выполненных заявках или работах в интерфейсе, позволяя пользователю выбирать конкретные записи для отправки уведомлений. В зависимости от выбора (пользователь или работник), подгружаются соответствующие данные из базы данных. Модуль реализует функционал отправки уведомлений по электронной почте, с возможностью прикрепления документов (актов или рабочих документов). Для отправки используется SMTP-сервер, и в случае корректности электронной почты, письмо отправляется с прикрепленным файлом. В процессе работы предусмотрены проверки и обработка ошибок, чтобы обеспечить корректную работу и информирование пользователя о возможных проблемах.

Каждый из этих модулей играет важную роль в обеспечении функциональности и удобства работы с приложением, позволяя пользователям легко взаимодействовать с системой, получать необходимую информацию и управлять своими задачами.

3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для использования данного приложения необходимо его запустить, нажать на ярлык два раза.

В приложении предусмотрены режимы доступа трех ролей:

- администратор;
- пользователь;
- работник.

Начальная форма позволяет войти в программу указав идентификационные данные пользователя или выполнить регистрацию нового пользователя. После запуска программы запускается начальная форма. Начальная форма программы представлена на рисунке 3.1.

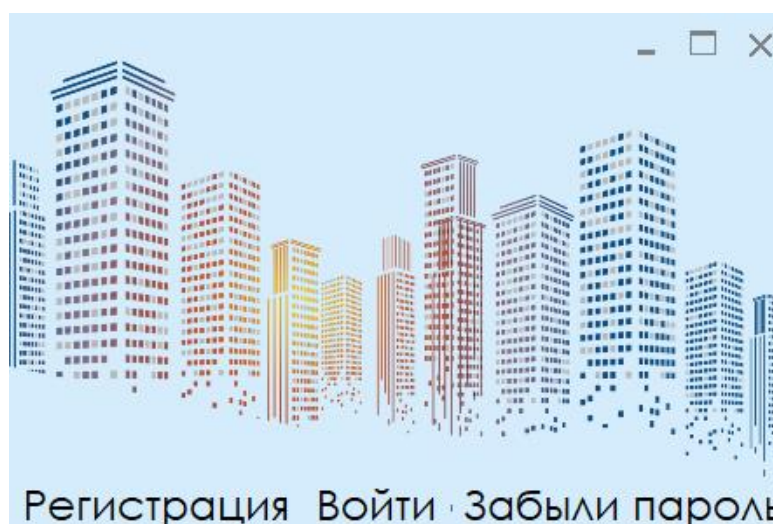


Рисунок 3.1 — Начальная форма

Пользователь может войти в аккаунт, зарегистрироваться или восстановить пароль или логин. После нажатия «Регистрация» необходимо выбрать регистрироваться как пользователь или работник. Далее нажать кнопку «Регистрация», откроется форма регистрации. После необходимо ввести данные и нажать кнопку «Зарегистрироваться». Процесс регистрации представлен на рисунке 3.2.

Регистрация

Фамилия: Петрова

Имя: Анна

Отчество: Сергеевна

Дата рождения: 15 августа 1990

Телефон: +375(29) 123-45-67

Логин: a.petrova

Пароль: Petrova#1990

Почта: anna.petrova@gmail.com

Зарегистрироваться

Рисунок 3.2 — Процесс регистрации

Регистрация работника происходит аналогично. После ввода кода доступа.

При нажатии на кнопку «Профиль» загружаются данные о пользователе. Данные можно редактировать. Для сохранения результата необходимо нажать кнопку «Редактировать». Редактирование профиля страхователя представлено на рисунке 3.3.

Редактировать профиль

Фамилия: Петрова

Имя: Анна

Отчество: Сергеевна

Дата рождения: 15 августа 1990

Телефон: +375(29) 123-45-67

Логин: a.petrovana

Пароль: Petrova#1990

Почта: anna.petrova@gmail.com

Редактировать

Сохранено

Изменения сохранены

OK

Рисунок 3.3 — Редактирование профиля пользователя

В режиме работы «Пользователь» доступны следующие функции:

- просмотр информации о своих заявках, оплатах и отзывах;
- просмотр справочника «Услуги», «Объекты»;
- оформление заявки;
- оплата заявки и написание отзыва;
- поиск данных по указанному критерию;
- редактирование профиля.

В режиме работы «Работник» доступны следующие функции:

- отклик на заявки;
- печать акта выполненных работ;
- работа со справочными данными;
- отправка писем и копий актов выполненных работ по указанному «email»;
- редактирование профиля;
- фильтрации выполненных работ по дате;
- поиск данных по указанному критерию;
- просмотр статистических данных;
- просмотр календаря задач.

Режим работы «Администратор» выполняет все функции работы с данными, а также ведение справочников, отправка писем и копии документов пользователям и работникам, получение выходных форм и отчетности.

При нажатии кнопки «Войти» необходимо выбрать уровень доступа — «Пользователь», «Работник» или «Администратор», ввести логин и пароль. Вход страхователя представлен на рисунке 3.4.

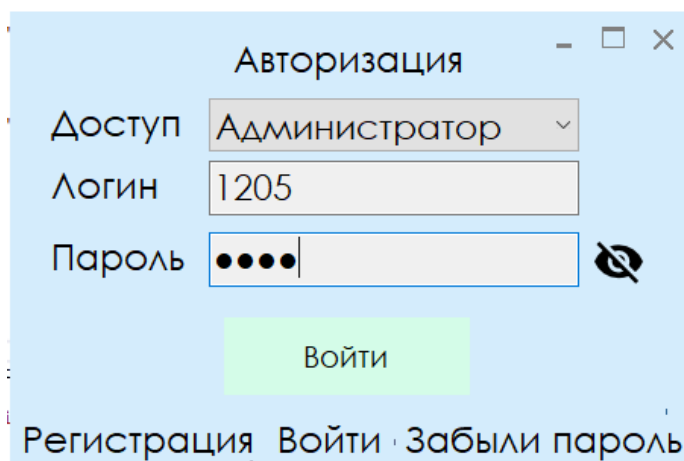


Рисунок 3.4 — Вход страхователя

После входа в программу при вводе корректных идентификационных данных открывается основное окно программы. Основное окно программы с режимом доступа «Администратор», представлено на рисунке 3.5.

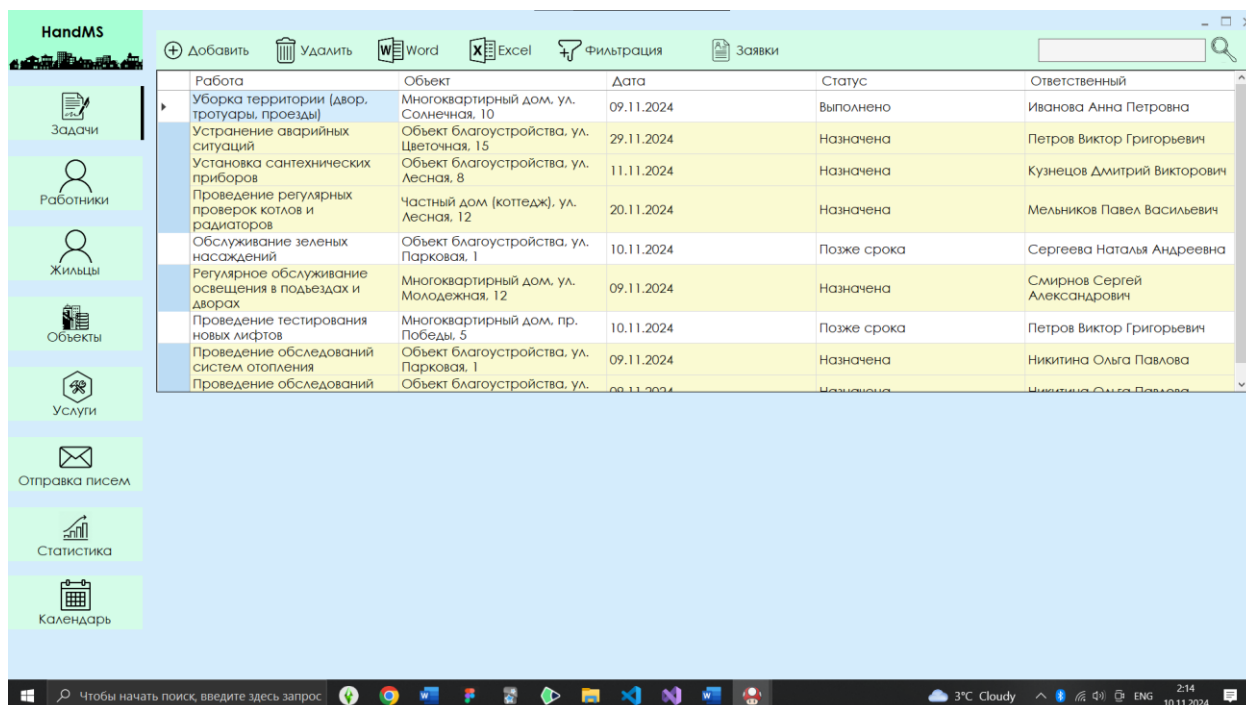


Рисунок 3.5 — Основное окно программы с режимом доступа «Администратор»

Основное окно программы с режимом доступа «Пользователь», представлено на рисунке 3.6.

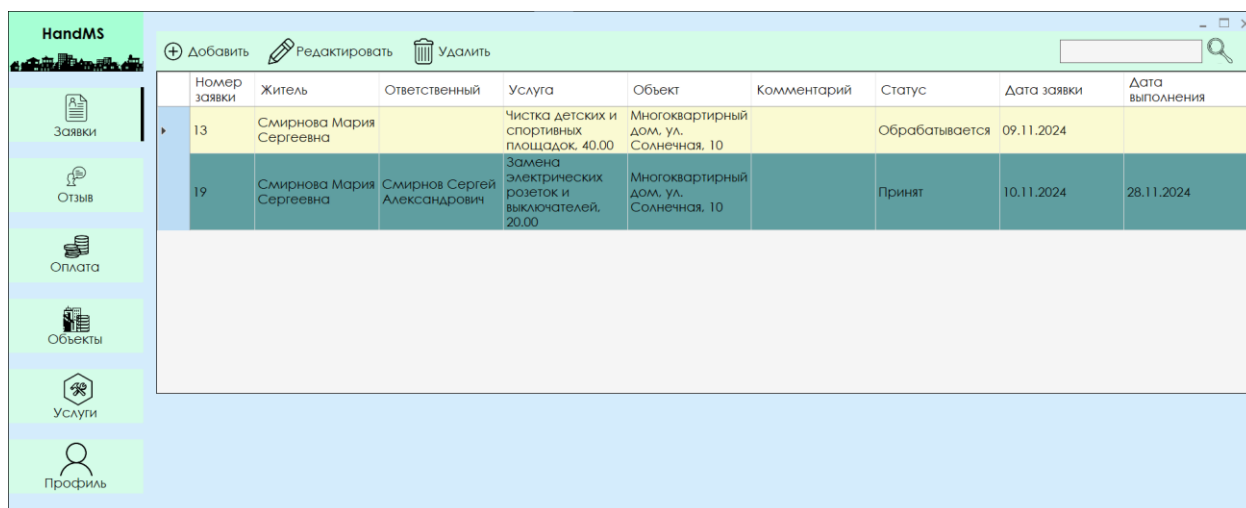


Рисунок 3.6 — Основное окно программы с режимом доступа «Пользователь»

Основное окно программы с режимом доступа «Работник», представлено на рисунке 3.7.

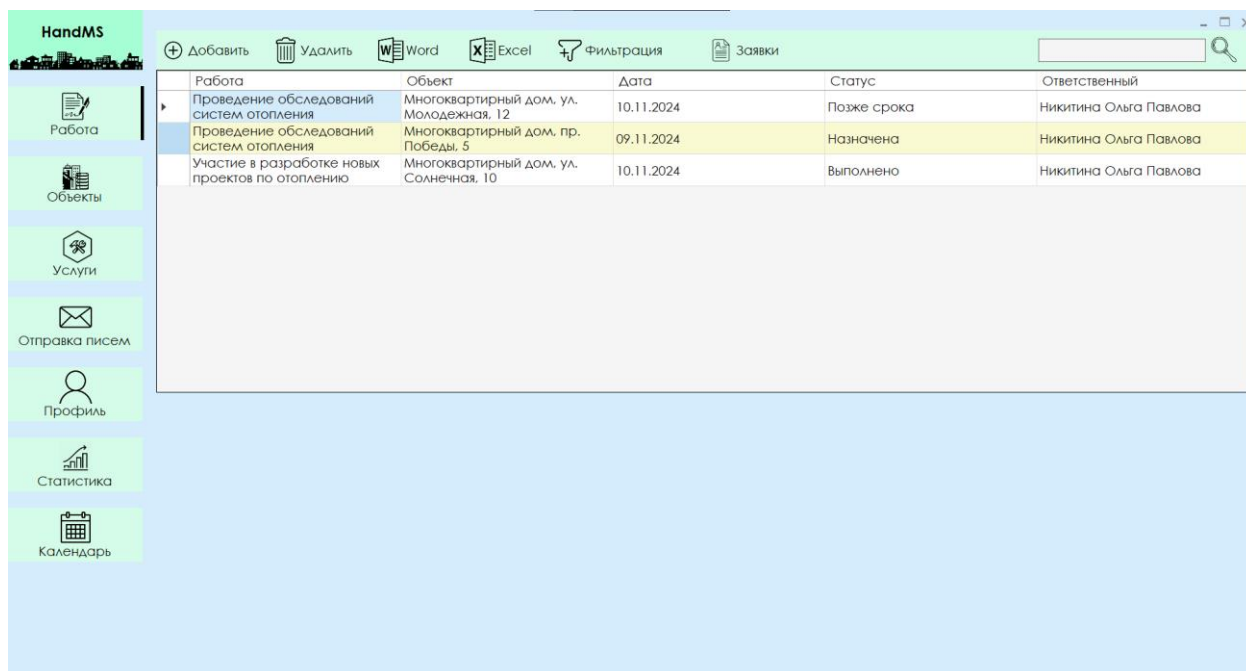


Рисунок 3.7 — Основное окно программы с режимом доступа «Работник»

Кнопочное меню в левой области определено уровнем доступа пользователя.

В приложении предусмотрены следующие справочные таблицы:

- работники;
- пользователи;
- услуги;
- отдел;
- объект;
- должности.

Для добавления объекта необходимо нажать кнопку «Объект». Далее нажать «Добавить», далее выбрать вид объекта и ввести всю необходимую информацию и нажать кнопку «Добавить». Добавление объекта представлено на рисунке 3.8.

Многоквартирный дом	ул. Молодежная, 12	12,000	- Количество этажей: 12; Количество квартир: 120; Тип строения: Монолитный; - Коммуникации: Водоснабжение: централизованное; Отопление: централизованное; Электричество: центральное; Канализация: централизованная; Газ: централизованный газопровод; - Наличие лифтов: Пассажирский: 3; - Технические помещения: Электрощитовая;
---------------------	--------------------	--------	--

Добавить многоквартирный дом

Тип строения: Кирпичный

Адрес: ул. Дружная, 16

Площадь: 11,000

Количество этажей: 7

Количество квартир: 70

+ Добавить

Коммуникации

Водоснабжение: ☒ Централизованное ☐ Автономное

Электричество: ☒ Центральное ☐ Автономное

Газ: ☒ Централизованный газопровод ☐ Сниженный газ в баллонах

Отопление: ☒ Централизованное ☐ Автономное

Канализация: ☒ Централизованная ☐ Автономная

Наличие лифтов

☒ Пассажирский Количество: 2

☒ Грузопассажирский Количество: 1

☐ Пожарный Количество:

Технические помещения

☒ Подвал ☐ Чердак ☒ Электросчетовая

☐ Машинное отделение лифтов

Придомовая территория

☒ Детская площадка ☒ Спортивная площадка

☐ Зеленые зоны ☒ Места для парковки

Рисунок 3.8 — Добавление вида страхования

Для редактирования информации о услугах, необходимо нажать кнопку «Услуги». Далее выбрать строку и нажать кнопку «Редактировать». Ввести требуемые изменения в выбранной строке и нажать кнопку «Редактировать». Процесс редактирования записи в таблице «Города» представлен на рисунках 3.9-3.10.

Уборка придомовой территории	мусора, очистка от листвы и снега.	50,00	Уборщик
Уборка мест общего пользования	Уборка лестничных клеток, коридоров и мест общего пользования в многоквартирных домах.	30,00	Уборщик
Чистка детских и спортивных площадок	Очистка и дезинфекция площадок для детей и спортивных объектов.	40,00	Уборщик
Устранение засоров	Устранение засоров в трубах и сантехнических устройствах.	40,00	Слесарь сантехник
Замена сантехнического оборудования	Замена унитазов, раковин и другой сантехники.	50,00	Слесарь сантехник
Ремонт системы горячего водоснабжения	Ремонт и обслуживание систем горячего водоснабжения.	60,00	Слесарь сантехник
Устранение неисправностей в электросистемах	Диагностика и устранение неисправностей в электрических системах.	60,00	Электрик
Замена электрических розеток и выключателей	Замена неисправных розеток и выключателей.	20,00	Электрик
	Полная проверка состояния		

Редактировать услугу

Название

Уборка мест общего пользования

Работник

Уборщик

Цена

80,00

Описание

Уборка лестничных клеток, коридоров и мест общего пользования в многоквартирных домах.

Редактировать

Рисунок 3.9 — Редактирование записи в таблице «Услуга»

Название	Описание	Цена	Работник
Уборка придомовой территории	Регулярная уборка территории, вывоз мусора, очистка от листвы и снега.	50,00	Уборщик
Уборка мест общего пользования	Уборка лестничных клеток, коридоров и мест общего пользования в многоквартирных домах.	80,00	Уборщик
Чистка детских и спортивных площадок	Очистка и дезинфекция площадок для детей и спортивных объектов.	40,00	Уборщик
Устранение засоров	Устранение засоров в трубах и сантехнических устройствах.	40,00	Слесарь сантехник
Замена сантехнического оборудования	Замена унитазов, раковин и другой сантехники.	50,00	Слесарь сантехник
Ремонт системы горячего водоснабжения	Ремонт и обслуживание систем горячего водоснабжения.	60,00	Слесарь сантехник

Рисунок 3.10 — Редактирование записи в таблице «Услуга»

Для удаления записи в таблице «Отзыв» необходимо нажать кнопку «Отзыв». Выбрать строку и нажать кнопку «Удалить». Подтвердить удаление выбранной записи в диалоговом окне. Удаление записи в таблице «Отзыв» представлено на рисунке 3.11.

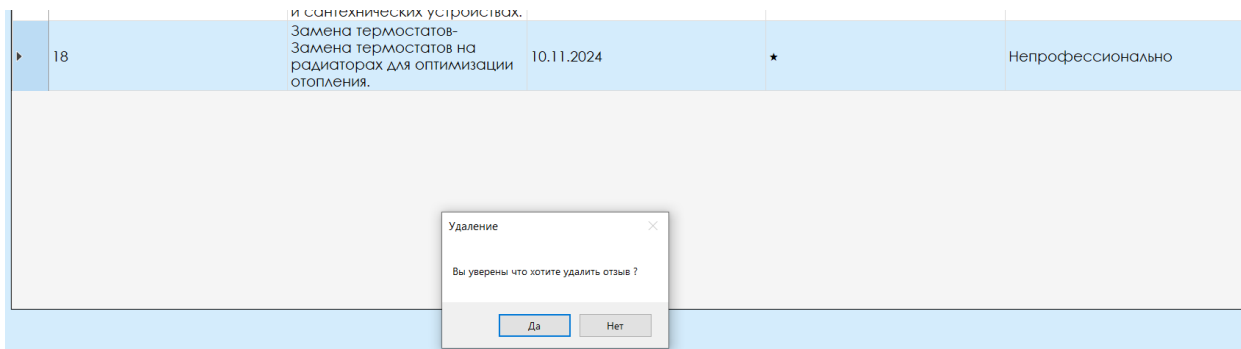


Рисунок 3.11 — Удаление строки в таблице «Отзыв»

Функции удаления, добавления и редактирования записей в таблицах базы данных реализованы аналогично.

Для отправки письма или копии документа пользователю необходимо нажать кнопку «Отправка писем». Далее выбрать пользователя, за тем выбрать если необходимо документ и поставить галочку в «Копия документа». Так же можно написать текст к письму, нажать кнопку «Отправить письмо». Об успешной отправке письма свидетельствует появление соответствующего сообщения. Пример отправки письма страхователю представлен на рисунке 3.12.

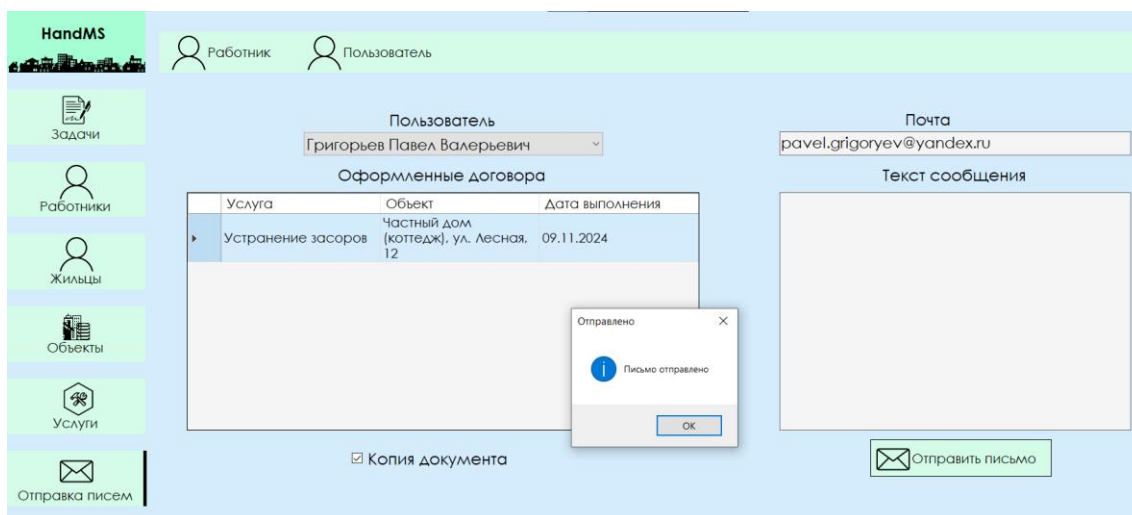


Рисунок 3.12 — Отправка письма страхователю

Для получения печатной формы акта выполненных работ нужно нажать на кнопку «Word». Вывод печатной формы акта выполненных работ представлен на рисунке 3.13.

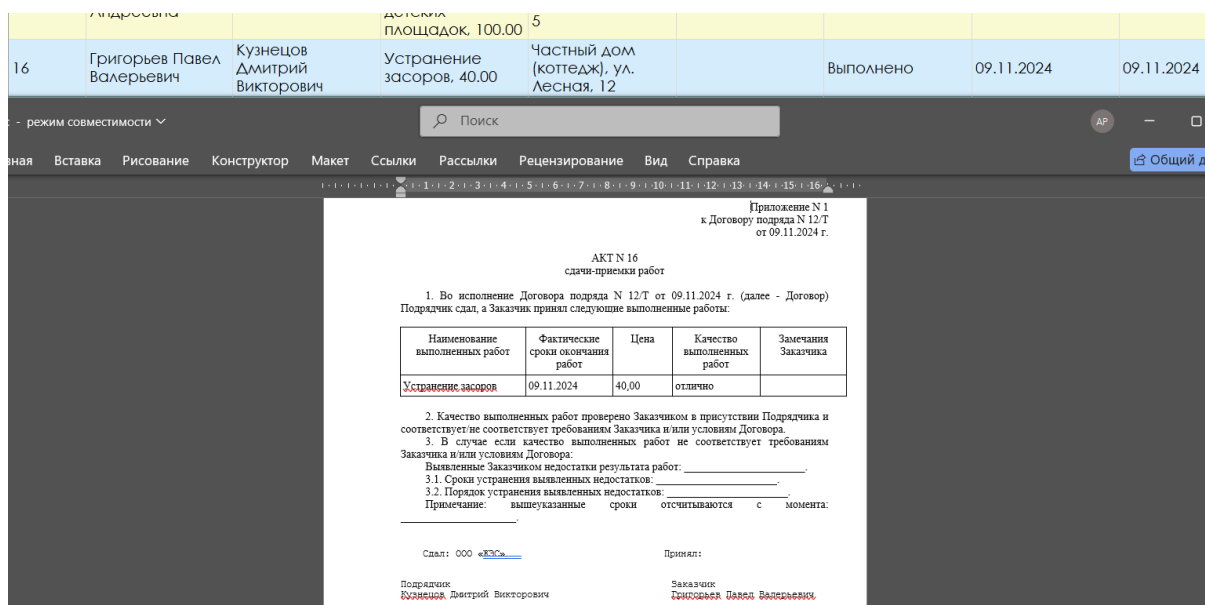


Рисунок 3.13 — Печатная форма акта выполненных работ «Microsoft Word»

В программе реализована статистика по количеству выполненных заявок в разрезе работников, количество выполненных задач в разрезе работников, динамика выполненных заявок, количество выполненных заявок в разрезе типы объекта и за указанный временной промежуток. Для просмотра статистических данных необходимо нажать кнопку «Статистика». Пример вывода статистических данных представлен на рисунке 3.14.

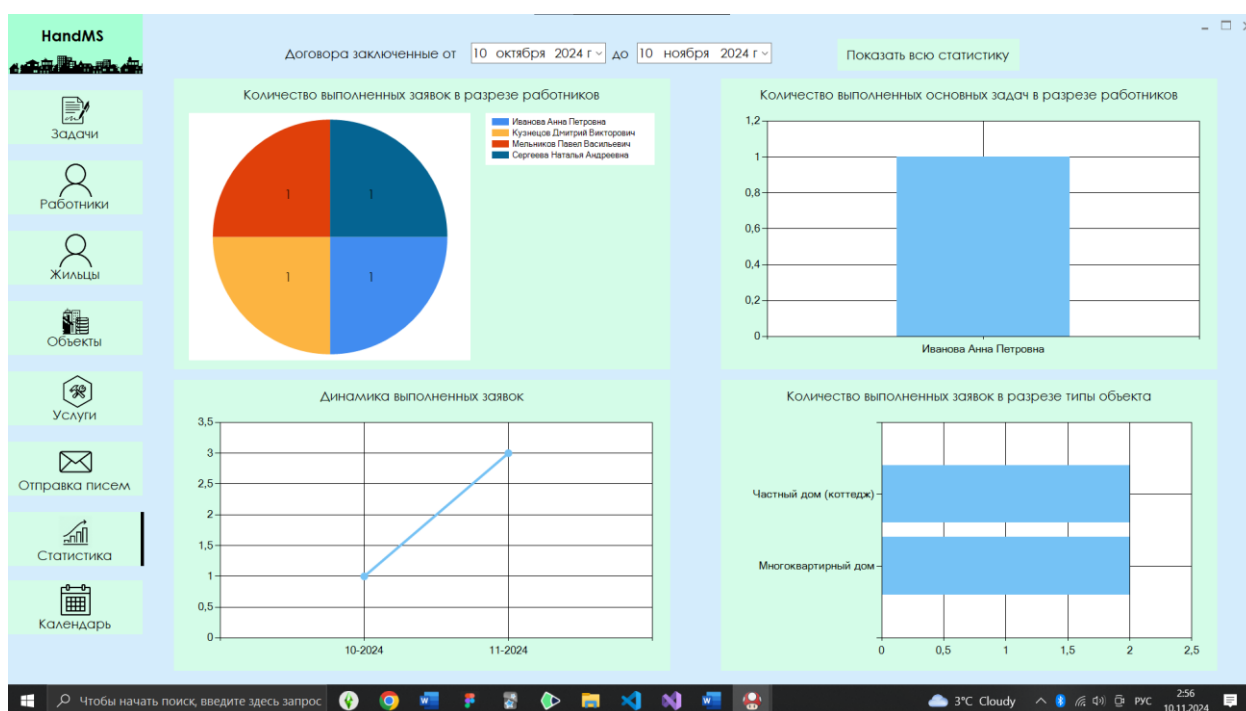


Рисунок 3.14 — Окно вывода статистических данных

Далее необходимо ввести полученный код и нажать кнопку «Проверить». После этого появится поле, в которое необходимо ввести новый логин и нажать кнопку «Сменить логин». Об успешной смене логина свидетельствует появление соответствующего сообщения. Смена логина представлена на рисунке 3.17.

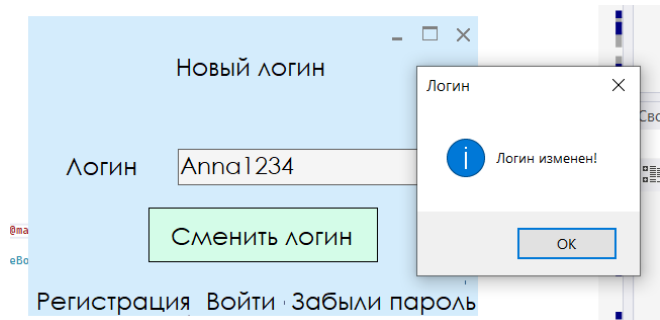


Рисунок 3.17 — Смена логина

Смена пароля осуществляется аналогично. Для поиска данных необходимо в соответствующем поле ввести текст. Строки с совпадением окрасятся в голубой цвет. Результат поиска представлен на рисунке 3.18.

A screenshot of a web application showing a table of user data. The table has columns: Фамилия (Surname), Имя (Name), Отчество (Patronymic), Дата рождения (Date of birth), Телефон (Phone), Почта (Email), Логин (Login), and Пароль (Password). The table is displayed in a window with a toolbar containing icons for adding, editing, deleting, and exporting to Excel. Several rows in the table are highlighted in blue, indicating search results. The data is as follows:

Фамилия	Имя	Отчество	Дата рождения	Телефон	Почта	Логин	Пароль
Петров	Алексей	Иванович	15.04.1985	+375(29) 111-22-33	aleksey.petrov@y...	a.petrov	Petrov#2024
Смирнова	Мария	Сергеевна	22.08.1990	+375(29) 222-33-44	maria.smirnova@...	m.smirnova	Smirnova#1990
Иванов	Дмитрий	Викторович	05.12.1978	+375(29) 333-44-55	dmitry.ivanov@q...	d.ivanov	Ivanov#1978
Кузнецова	Елена	Андреевна	30.01.1988	+375(29) 444-55-66	elena.kuznecova@...	e.kuznecova	Kuznecova#1988
Федоров	Сергей	Николаевич	10.06.1982	+375(29) 555-66-77	sergey.fedorov@...	s.fedorov	Fedorov#1982
Мельникова	Наталья	Петров	18.03.1995	+375(29) 666-77-88	natalya.melnikov...	n.melnikova	Melnikova#1995
Сергеева	Анна	Васильевна	25.11.1989	+375(29) 777-88-99	anna.sergeeva@...	a.sergeeva	Sergeeva#1989
Никитин	Виктор	Григорьевич	14.09.1980	+375(29) 888-99-00	viktor.nikitin@mail.ru	v.nikitin	Nikitin#1980
Сидорова	Ольга	Андреевна	02.07.1992	+375(29) 999-00-11	olga.sidorova@q...	o.sidorova	Sidorova#1992
Григорьев	Павел	Валерьевич	09.05.1987	+375(29) 101-12-13	pavel.grigoryev@...	p.grigoryev	Grigoryev#1987
Петрова	Анна	Сергеевна	15.08.1990	+375(29) 123-45-67	anna.petrova@q...	a.petrovana	Petrova#1990

Рисунок 3.18 — Результат поиска

Поиск во всех таблицах приложения происходит аналогично. Также в программе реализован другой вид поиска для таблицы «Объект». Для поиска данных необходимо в соответствующем поле ввести текст. Столбцы с совпадением окрасятся в желтый цвет и строки без информации скроются. Результат поиска представлен на рисунке 3.19.

Добавить

Редактировать

Удалить

Excel

12

Тип объекта	Адрес	Площадь	Описание
Многоквартирный дом	ул. Молодежная, 12	12,000	- Количество этажей: 12; Количество квартир: 120; Тип строения: Монолитный; - Коммуникации: Водоснабжение: централизованное; Отопление: централизованное; Электричество: центральное; Канализация: централизованная; Газ: централизованный газопровод; - Наличие лифтов: Пассажирский: 3; - Технические помещения: Электрощитовая; - Придомовая территория: Детская площадка; Зеленые зоны; Спортивная площадка; Места для парковки;
Частный дом (коттедж)	ул. Лесная, 12	150,000	Тип строения: одноэтажный; - Коммуникации: Водоснабжение: автономное; Отопление: автономное; Электричество: центральное; Канализация: автономная; Газ: централизованный газопровод; - Придомовая территория: Сад; Хозяйственная постройка; Газон;

Рисунок 3.19 — Результат поиска

Предусмотрена фильтрация по следующим критериям:

- дате задач, в указанном диапазоне дат;
- работающие и уволенные работники.

Для фильтрации задач необходимо нажать кнопку «Фильтровать». Выбрать диапазон дат и нажать нижнюю кнопку «Фильтровать». Пример работы функции фильтрации данных в таблице «Задачи» представлен на рисунке 3.20.

Добавить

Удалить

Word

Excel

Фильтрация

Заявки

Работа	Объект	Дата	Статус	Ответственный
Уборка территории (двор, тротуары, проезды)	Многоквартирный дом, ул. Солнечная, 10	09.11.2024	Выполнено	Иванова Анна Петровна
Обслуживание зеленых насаждений	Объект благоустройства, ул. Парковая, 1	10.11.2024	Позже срока	Сергеева Наталья Андреевна
Регулярное обслуживание освещения в подъездах и дворах	Многоквартирный дом, ул. Молодежная, 12	09.11.2024	Назначена	Смирнов Сергей Александрович
Проведение тестирования новых лифтов	Многоквартирный дом, пр. Победы, 5	10.11.2024	Позже срока	Петров Виктор Григорьевич
Проведение обследований систем отопления	Многоквартирный дом, ул. Молодежная, 12	10.11.2024	Позже срока	Никитина Ольга Павлова
Проведение обследований систем отопления	Многоквартирный дом, пр. Победы, 5	09.11.2024	Назначена	Никитина Ольга Павлова
Участие в разработке новых проектов по отоплению	Многоквартирный дом, ул. Солнечная, 10	10.11.2024	Выполнено	Никитина Ольга Павлова

Фильтрация

От 9 ноября 2024 г. до 10 ноября 2024 г.

Фильтровать

Рисунок 3.20 — Пример работы функции фильтрации

Фильтрация остальных данных происходит аналогично.

В приложении предусмотрено получение следующих печатных форм документов:

- акт выполненных работ;
- содержимое таблиц.

Для вывода содержимого таблиц средствами «Microsoft Excel» необходимо нажать «Excel». Пример импорта данных «Работники» представлен на рисунке 3.21.

Фамилия	Имя	Отчество	Дата приема	Дата увольнения	Должность	Отдел	Телефон	Почта
Иванова	Анна	Петровна	01.03.2022		Уборщик	Уборка и благоустройства	+375(29) 123-45-67	ridetskaya.anka@ex.ru
Смирнов	Сергей	Александрович	15.05.2021		Электрик	Электротехнический	+375(29) 234-56-78	sergey.smirnov@com
Кузнецов	Дмитрий	Викторович	10.07.2022		Слесарь сантехник	Сантехнический	+375(29) 345-67-89	dmitry.kyznetsov.com
Федоров	Алексей	Николаевич	20.04.2023		Кровельщик	Кровельный	+375(29) 456-78-90	alexei.fedorov@ru
Волкова	Ирина	Сергеевна	05.06.2021		Механик по обслуживанию	Механического обслуживания	+375(29) 567-89-01	irina.volkova@m
Петров	Виктор	Григорьевич	12.01.2020		Оператор по эксплуатации лифтов	Эксплуатации лифтов	+375(29) 678-90-12	viktor.petrov@gm
Сергеева	Наталья	Андреевна	30.08.2022		Специалист по благоустройству территории	Благоустройство территории	+375(29) 789-01-23	natalya.sergeeva@ex.ru
Никитина	Ольга	Павлова	18.11.2021		Инженер по теплоэнергетике	Теплотехнический	+375(29) 901-23-45	olga.nikitina@gm

Рисунок 3.21 — Импорт данных в «Microsoft Excel»

Импорт данных таблиц в «Microsoft Excel» происходит аналогично. Для сортировки данных в любой таблице необходимо нажать на заголовок столбца. Появится указатель, если указатель направлен в низ сортировка по убыванию (От Я До А), направлен вверх сортировка происходит по возрастанию (От А До Я). Сортировка по убыванию по столбцу «Статус» (От Я До А) представлена на рисунке 3.22.

Номер заявки	Житель	Ответственный	Услуга	Объект	Комментарий	Статус	Дата заявки	Дата выполнения
14	Кузнецова Елена Андреевна	Мельников Павел Васильевич	Очистка радиаторов, 30.00	Многоквартирный дом, ул. Солнечная, 10		Принят	09.11.2024	21.11.2024
17	Мельникова Наталья Петровна	Иванова Анна Петровна	Чистка детских и спортивных площадок, 40.00	Многоквартирный дом, ул. Молодежная, 12		Принят	09.11.2024	21.11.2024
19	Смирнова Мария Сергеевна	Смирнов Сергей Александрович	Замена электрических розеток и выключателей, 20.00	Многоквартирный дом, ул. Солнечная, 10		Принят	10.11.2024	28.11.2024
18	Сергеева Анна Васильевна	Мельников Павел Васильевич	Замена термостатов, 40.00	Частный дом (коттедж), ул. Лесная, 12		Позже срока	09.11.2024	09.11.2024
13	Смирнова Мария Сергеевна	Иванова Анна Петровна	Чистка детских и спортивных площадок, 40.00	Многоквартирный дом, ул. Солнечная, 10		Выполнено	09.10.2024	10.10.2024
	Смирнова Мария Сергеевна	Сергеева	Ремонт и обслуживание	Многоквартирный				

Рисунок 3.22 — Сортировка по убыванию (От Я До А)

ЗАКЛЮЧЕНИЕ

Разработка автоматизированной системы управления работой работников жилищно-эксплуатационных служб (ЖЭС) с использованием языка программирования С#, системы управления базами данных Microsoft SQL Server и среды разработки Microsoft Visual Studio 2022 представляет собой эффективное решение для повышения качества обслуживания населения.

Результатом курсовой стала эффективная автоматизированная система, которая предоставляет множество возможностей. К достоинствам системы можно отнести возможность добавления, редактирования и удаления заявок, а также управление данными о работниках и выполненных задачах. Пользователи могут просматривать информацию из базы данных с функциями фильтрации и поиска, что значительно упрощает взаимодействие с системой.

Система также реализует разграничение прав доступа по ролям «Администратор», «Работник», «Пользователь», что повышает безопасность и контроль. Анализ данных и предоставление статистической отчетности помогают в принятии обоснованных решений и планировании работы ЖЭС.

Программа обладает простым и понятным интерфейсом, что делает работу с ней интуитивно понятной для пользователей. Надежное хранение данных и достоверность предоставленной информации способствуют снижению количества ошибок в учете и повышению качества обслуживания граждан.

Проверка оригинальности пояснительной записки выполнена с использованием платформы «antiplagiat.ru». Результаты показали, что работа обладает 86.17% оригинальности, что удовлетворяет установленным требованиям по оригинальности текста.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Блок схема [Электронный ресурс] – URL—
<https://www.lucidchart.com/pages/ru/%D0%B1%D0%BB%D0%BE%D0%BA-%D1%81%D1%85%D0%B5%D0%BC%D0%B0> (дата обращения 06.11.2024).
2. Шарп, Джон Microsoft Visual C#. Подробное руководство. 8-е изд./ Джон Шарп — СПб.: Питер, 2017 — 848 с.
3. Джозеф, Албахари C# 5.0. Справочник. Полное описание языка; Диалектика / Вильямс - М., 2014. - 545 с.
4. Бьюли, А. Изучаем SQL/ А. Бьюли — СПб.: Символ-Плюс, 2007. — 312 с.
5. Реляционная база данных [Электронный ресурс] – URL—
<https://metanit.com/sql/tutorial/1.1.php> (дата обращения 05.11.2024).
6. Петкович Душан. «Microsoft SQL Management Studio»/ Душан Петкович. — СПб. Символ-Плюс, 2009 — 467с.
7. Visual Studio 2022 [Электронный ресурс]. – URL.
<https://metanit.com/sharp/tutorial/1.2.php> (дата обращения 01.11.2024).
8. Монолитная и микросервисная архитектура: чем они отличаются [Электронный ресурс]. – URL. <https://appmaster.io/ru/blog/monolitnaia-i-mikroservisnaia-arkhitektura> (дата обращения 10.11.2024).

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

```
using Agent.Form;
using Agent.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.Mail;
using System.Data.SqlClient;
using System.Net;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using static
System.Windows.Forms.VisualStyles.VisualStyleElement.ToolBar;
using static MetroFramework.Drawing.MetroPaint.BorderColor;
using System.Diagnostics.Eventing.Reader;
namespace Agent
{
    public partial class Startcs : System.Windows.Forms.Form
    {
        database db = new database();
        public SqlConnection sqlConnection = null;
        public Startcs()
        {
            InitializeComponent();
            sqlConnection = new SqlConnection(db.connection);
        }
        public Boolean press = true;
        private System.Drawing.Point mouseOffset;
        private bool isMouseDown = false;
        private void Form1_KeyUp(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Escape) this.Close();
        }
        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
            int xOffset;
```

```

int yOffset;
if (e.Button == MouseButtons.Left)
{
    xOffset = -e.X - SystemInformation.FrameBorderStyle.Width;
    yOffset = -e.Y - SystemInformation.CaptionHeight -
        SystemInformation.FrameBorderStyle.Height;
    mouseOffset = new System.Drawing.Point(xOffset, yOffset);
    isMouseDown = true;
}
}
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    if (isMouseDown)
    {
        System.Drawing.Point mousePos = Control.MousePosition;
        mousePos.Offset(mouseOffset.X, mouseOffset.Y);
        Location = mousePos;
    }
}
private void Form1_MouseUp(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        isMouseDown = false;
    }
}
public int idak = 0;
private void button1_Click(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex == 1)
    {
        try
        {
            //Фрагмент кода для входа в программу
            string query1 = $"Select idtenant from tenant
            where password='{textBox1.Text}' and
            login='{textBox3.Text}'";
            DataTable data = new DataTable();
            SqlDataAdapter command1 = new SqlDataAdapter(query1,
            sqlConnection);
            command1.Fill(data);
            DataColumn column = data.Columns[0];
            DataRow row = data.Rows[0];
            idak = Convert.ToInt32(row[column].ToString());
            textBox1.Text = "";
        }
        catch { }
    }
}

```

```

        textBox3.Text = "";
        Tenantak policyholderak = new Tenantak(this);
        policyholderak.Width = 1920;
        policyholderak.Height = 1040;
        policyholderak.panel6.Visible = true;
        policyholderak.panel11.Visible = false;
        policyholderak.ShowDialog();
    }
    catch
    {
        MessageBox.Show("Неверный логин или пароль!",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}
else if (comboBox1.SelectedIndex == 2)
{

    try
    {
        string query1 = $"Select idworker from worker where
password='{textBox1.Text}' and login='{textBox3.Text}'";
        DataTable data = new DataTable();
        SqlDataAdapter command1 = new SqlDataAdapter(query1,
sqlConnection);

        command1.Fill(data);
        DataColumn column = data.Columns[0];
        DataRow row = data.Rows[0];
        idak = Convert.ToInt32(row[column].ToString());

        string query2 = $"Select datelayoffs from worker where
idworker={idak}";
        DataTable data2 = new DataTable();
        SqlDataAdapter command2 = new SqlDataAdapter(query2,
sqlConnection);

        command2.Fill(data2);
        DataColumn column2 = data2.Columns[0];
        DataRow row2 = data2.Rows[0];
        if (row2[column2].ToString() != null &&
row2[column2].ToString() != "")
        {
            MessageBox.Show("Нет доступа. Причина: вы
уволены!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else

```



```

        {
            textBox1.Text = "";
            textBox3.Text = "";
            Workerak insurerak = new Workerak(this);
            insurerak.Width = 1920;
            insurerak.Height = 1040;
            insurerak.panel1.Visible = false;
            // insurerak.button9.Visible = false;
            insurerak.button6.BringToFront();
            insurerak.button6.Visible = true;
            insurerak.button10.BringToFront();
            insurerak.button10.Visible = true;
            insurerak.button9.BringToFront();
            insurerak.button9.Visible = true;
            insurerak.ShowDialog();
        }
    }

    catch
    {
        MessageBox.Show("Неверный логин или пароль!",
            "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

else
{
    //      if (textBox1.Text == "1205" && textBox3.Text == "1205")
    //      {
        textBox1.Text = "";
        textBox3.Text = "";
        Form1 form1 = new Form1();
        form1.ShowDialog();
        //      }
        //      else
        //      {
            MessageBox.Show("Неверный логи или пароль!",
                "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            //      }
        }
}

private void label5_Click(object sender, EventArgs e)
{
    if (textBox1.UseSystemPasswordChar == true)
    {
        textBox1.UseSystemPasswordChar = false;
    }
}

```

```

        label5.Image =
Image.FromFile("D:/Diplom/proga/Agent/Agent/Resources/eye.png");
    }
    else
    {
        textBox1.UseSystemPasswordChar = true;
        label5.Image =
Image.FromFile("D:/Diplom/proga/Agent/Agent/Resources/1.png");
    }
}
private void linkLabel2_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    panel1.Visible = true;
    panel2.Visible = false;
    panel3.Visible = false;
}
private void linkLabel3_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    label9.Text = "Забыли пароль или логин";
    button4.Visible = true;
    button3.Visible = true;
    button11.Visible = false;
    label8.Visible = false;
    label7.Visible = false;
    comboBox2.Visible = false;
    textBox10.Visible = false;
    panel2.Visible = true;
    panel1.Visible = false;
    panel3.Visible = false;
}

private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    panel3.Visible = true;
    panel1.Visible = false;
    panel2.Visible = false;
}
private void Startcs_Load(object sender, EventArgs e)
{
    panel2.Visible = false;
    panel3.Visible = false;
    panel1.Visible = false;
}

```

```

    }
    private void button2_Click(object sender, EventArgs e)
    {
        if (comboBox3.SelectedIndex == 0)
        {
            Tenantak policyholderak = new Tenantak(this);
            policyholderak.panel1.Visible = true;
            policyholderak.panel1.Left = 12;
            policyholderak.panel1.Top = 0;
            policyholderak.Width = 520;
            policyholderak.Height = 524;
            policyholderak.panel6.Visible = false;
            policyholderak.panel11.Visible = false;
            policyholderak.panel14.Visible = false;
            policyholderak.label10.Text = "Регистрация";
            policyholderak.button11.Text = "Зарегистрироваться";
            policyholderak.panel6.Visible = false;
            policyholderak.clear();
            policyholderak.ShowDialog();
        }
        else
        {
            if (textBox2.Text=="1205") {
                Workerak insurerak = new Workerak(this);
                insurerak.panel1.Visible = true;
                insurerak.panel1.Left = 0;
                insurerak.panel1.Top = 0;
                insurerak.Width = 484;
                insurerak.Height = 572;
                insurerak.panel2.Visible = false;
                insurerak.panel6.Visible = false;
                insurerak.button11.Width = 280;
                insurerak.metroSetControlBox2.Visible = false;
                insurerak.label6.Text = "Регистрация";
                insurerak.button11.Text = "Зарегистрироваться";
                insurerak.metroSetControlBox1.Visible = true;
                insurerak.ShowDialog();
                textBox2.Text = "";
            }
            else { MessageBox.Show("Нет доступа!", "Ошибка",
                MessageBoxButtons.OK, MessageBoxIcon.Error); }
        }
    }
    public void three()
    {

```

```

        //Код отправки кода
        MailAddress fromadress = new MailAddress("mih2023@mail.ru",
"Admin");
        MailAddress toadress = new MailAddress(textBox10.Text,
comboBox2.Text);
        MailMessage Message = new MailMessage(fromadress, toadress);
        Random random = new Random();
        kod = Convert.ToString(random.Next(100000, 1000000));
        if (text == "password")
        {
            Message.Subject = "Забыли пароль Insurance";
            Message.Body = "Код для смены пароля: " + kod;
        }
        else
        {
            Message.Subject = "Забыли логин Insurance";
            Message.Body = "Код для смены логина: " + kod;
        }
        SmtpClient smtpClient = new SmtpClient();
        smtpClient.Host = "smtp.mail.ru";
        smtpClient.Port = 587;
        smtpClient.EnableSsl = true;
        smtpClient.DeliveryMethod = SmtpDeliveryMethod.Network;
        smtpClient.UseDefaultCredentials = false;
        smtpClient.Credentials = new
NetworkCredential("mih2023@mail.ru", "nzYCLzK1x7hjginEbApV");
        smtpClient.Send(Message);
        MessageBox.Show("Проверьте почту", "Отправлено",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        textBox10.Text = "";
        label9.Text = "Введите код";
        label8.Text = "Код";
        label7.Visible = false;
        comboBox2.Visible = false;
        button11.Text = "Проверить";
    }
    string kod = "";
    int id = 0;
    string email = "";
    private void button11_Click(object sender, EventArgs e)
    {
        if (comboBox2.SelectedIndex != -1)
        {
            if (comboBox2.SelectedIndex == 0)
            {

```

```

        if (button11.Text == "Отправить код")
        {
            textBox10.MaxLength = 6;
            Regex r2 = new
Regex(@"(\S*(@mail\.ru|@gmail\.com|@yandex\.ru))$");
            if (r2.IsMatch(textBox10.Text) && textBox10.TextLength >
8)
            {
                try
                {
                    string query1 = $"Select idtenant from tenant where
email='{textBox10.Text}'";
                    DataTable data = new DataTable();
                    SqlDataAdapter command1 = new
SqlDataAdapter(query1, sqlConnection);
                    command1.Fill(data);
                    DataColumn column = data.Columns[0];
                    DataRow row = data.Rows[0];
                    id = Convert.ToInt32(row[column].ToString());
                    email = textBox10.Text;
                }
                catch
                {
                    three();
                }
                catch { MessageBox.Show("Ошибка!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error); }
            }
            catch
            {
                MessageBox.Show("Нет страхователя с такой
почтой!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Некорректная почта!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else if (button11.Text == "Проверить")
    {
        four();
    }
    else if (button11.Text == "Сменить пароль")

```

```

    {
        textBox10.MaxLength = 30;
        if (textBox10.Text.Any(char.IsLower) &&
textBox10.Text.Any(char.IsUpper) && textBox10.Text.Length > 8)
        {
            sqlConnection.Open();
            SqlCommand command = new
SqlCommand($"@\"UPDATE tenant SET password=@pa WHERE idtenant=@id",
sqlConnection);
            command.Parameters.AddWithValue("@pa",
(textBox10.Text));
            command.Parameters.AddWithValue("@id", (id));
            command.ExecuteNonQuery();
            sqlConnection.Close();
            MessageBox.Show("Пароль изменен!", "Пароль",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            two();
        }
        else
        {
            MessageBox.Show("Пароль должен быть больше 8
символов, содержать одну букву верхнего и нижнего регистра!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else if (button11.Text == "Сменить логин")
    {
        textBox10.MaxLength = 30;
        if (textBox10.Text.Length > 5 &&
textBox10.Text.Any(char.IsLetter))
        {
            //Код на смену логина
            sqlConnection.Open();
            SqlCommand command = new
SqlCommand($"@\"UPDATE tenant SET login=@pa
WHERE idtenant=@id", sqlConnection);
            command.Parameters.AddWithValue("@pa",
(textBox10.Text));
            command.Parameters.AddWithValue("@id", (id));
            command.ExecuteNonQuery();
            sqlConnection.Close();
            MessageBox.Show("Логин изменен!", "Логин",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            two();
        }
    }

```

```

else
{
    MessageBox.Show("Логин должен быть больше 5
символов и содержать хотя бы одну букву!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    if (button11.Text == "Отправить код")
    {
        textBox10.MaxLength = 6;
        Regex r2 = new
Regex(@"(\S*(@mail\.ru|@gmail\.com|@yandex\.ru))$");
        if (r2.IsMatch(textBox10.Text) && textBox10.TextLength >
8)
        {
            try
            {
                string query1 = $"Select idworker from worker where
email='{textBox10.Text}'";
                DataTable data = new DataTable();
                SqlDataAdapter command1 = new
SqlDataAdapter(query1, sqlConnection);
                command1.Fill(data);
                DataColumn column = data.Columns[0];
                DataRow row = data.Rows[0];
                id = Convert.ToInt32(row[column].ToString());
                email = textBox10.Text;
                string query2 = $"Select datelayoffs from worker
where idworker={id}";
                DataTable data2 = new DataTable();
                SqlDataAdapter command2 = new
SqlDataAdapter(query2, sqlConnection);
                command2.Fill(data2);
                DataColumn column2 = data2.Columns[0];
                DataRow row2 = data2.Rows[0];
                if (row2[column2].ToString() != null &&
row2[column2].ToString() != "")
                {
                    MessageBox.Show("Нет доступа. Причина: вы
уволены!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else

```

```

        {
            try
            {
                three();
            }
            catch { MessageBox.Show("Ошибка!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error); }
        }
    }
    catch
    {
        MessageBox.Show("Нет страховщика с такой
почтой!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    MessageBox.Show("Некорректная почта!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else if (button11.Text == "Проверить")
{
    four();
}
else if (button11.Text == "Сменить пароль")
{
    textBox10.MaxLength = 30;
    if (textBox10.Text.Any(char.IsLower) &&
textBox10.Text.Any(char.IsUpper) && textBox10.Text.Length > 8)
    {
        sqlConnection.Open();
        SqlCommand command = new
SqlCommand($"UPDATE worker SET password=@pa WHERE
idworker=@id", sqlConnection);
        command.Parameters.AddWithValue("@pa",
(textBox10.Text));
        command.Parameters.AddWithValue("@id", (id));
        command.ExecuteNonQuery();
        sqlConnection.Close();
        MessageBox.Show("Пароль изменен!", "Пароль",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        two();
    }
}
else

```



```

        {
            MessageBox.Show("Пароль должен быть больше 8
символов, содержать одну букву верхнего и нижнего регистра!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else if (button11.Text == "Сменить логин")
    {
        textBox10.MaxLength = 30;
        if (textBox10.Text.Length > 5 &&
textBox10.Text.Any(char.IsLetter))
        {
            sqlConnection.Open();
            SqlCommand command = new
SqlCommand($"UPDATE worker SET login=@pa WHERE idworker=@id",
sqlConnection);
            command.Parameters.AddWithValue("@pa",
(textBox10.Text));
            command.Parameters.AddWithValue("@id", (id));
            command.ExecuteNonQuery();
            sqlConnection.Close();
            MessageBox.Show("Логин изменен!", "Логин",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            two();
        }
        else
        {
            MessageBox.Show("Логин должен быть больше 5
символов и содержать хотя бы одну букву!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}
}
public void four()
{
    textBox10.MaxLength = 30;
    if (kod == textBox10.Text)
    {
        if (text == "password")
        {
            label8.Text = "Пароль";
            label9.Text = "Новый пароль";
            button11.Text = "Сменить пароль";
        }
    }
}
}

```

```

        textBox10.Text = "";
    }
    else
    {
        label8.Text = "Логин";
        label9.Text = "Новый логин";
        button11.Text = "Сменить логин";
        textBox10.Text = "";
    }
}
else
{
    MessageBox.Show("Код неверный!", "Ошибка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    label8.Text = "Почта";
    if (text == "password")
    {
        label9.Text = "Забыли пароль";
    }
    else
    {
        label9.Text = "Забыли логин";
    }
    button11.Text = "Отправить код";
    comboBox2.SelectedIndex = -1;
    textBox10.Text = email;
    label7.Visible = true;
    comboBox2.Visible = true;
}
}
public void two()
{
    label8.Text = "Почта";
    label9.Text = "Забыли пароль или логин";
    button11.Text = "Отправить код";
    comboBox2.SelectedIndex = -1;
    textBox10.Text = "";
    label8.Visible = false;
    textBox10.Visible = false;
    button3.Visible = true;
    button4.Visible = true;
    panel2.Visible = false;
    panel1.Visible = true;
}
private void textBox10_KeyPress(object sender, KeyPressEventArgs e)

```

```

        {
            if (button11.Text == "Отправить код")
            {
                char c = e.KeyChar;
                e.Handled = !((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c
== '@' || c == 8 || c == '.'));
                if (e.KeyChar == '@')
                {
                    if (((sender as
System.Windows.Forms.TextBox).Text.IndexOf('@') != -1))
                    {
                        e.Handled = true;
                        return;
                    }
                }
            }
            else if (label9.Text == "Забыли логин")
            {
                char c = e.KeyChar;
                e.Handled = !((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c
>= '0' && c <= '9') || (c == '@' || c == 8 || c == '.' || c == '_' || c == ','));
            }
            else if (button11.Text == "Проверить")
            {
                if ((e.KeyChar <= 47 || e.KeyChar >= 58) && e.KeyChar != 8)
                    e.Handled = true;
            }
            else
            {
                char c = e.KeyChar;
                e.Handled = !((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c
>= '0' && c <= '9') || (c == '@' || c == 8 || c == '.' || c == '_' || c == ',' || c == '#' || c ==
'+' || c == '!' || c == '$' || c == ':' || c == ';' || c == '%' || c == '^' || c == '&' || c == '*' || c
== ') || c == '(' || c == '-'));
            }
        }
        string text = "";
        public void one()
        {
            button11.Visible = true;
            label8.Visible = true;
            label7.Visible = true;
            comboBox2.Visible = true;
            textBox10.Visible = true;
            button3.Visible = false;
        }
    }

```

```

        button4.Visible = false;
    }
    private void button4_Click(object sender, EventArgs e)
    {
        label9.Text = "Забыли логин";
        text = "login";
        one();
    }
    private void button3_Click(object sender, EventArgs e)
    {
        label9.Text = "Забыли пароль";
        text = "password";
        one();
    }

    private void comboBox3_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        if (comboBox3.SelectedIndex==1)
        {
            textBox2.Text = "";
            textBox2.Visible=true;
            label11.Visible=true;
        }
        else
        {
            textBox2.Visible = false;
            label11.Visible = false;        }    }    }}

```