



ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения заочная

«К ЗАЩИТЕ ДОПУЩЕН(А)»
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

2024

Выпускная квалификационная работа

Обучающегося Тимофеев Александр Александрович

(фамилия, имя, отчество)

Вид работы выпускная квалификационная работа бакалавра

(выпускная квалификационная работа бакалавра, специалиста, магистра)

Пояснительная записка

Тема Разработка приложения по учету рабочего времени сотрудников предприятия (на примере Филиал ФБУЗ «Центр гигиены и эпидемиологии»)

(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы к. ф.-м. н., доцент Черняева С. Н. 21.06.24

(должность, подпись, фамилия, инициалы, дата)

Консультант _____

(при наличии)

(должность, подпись, фамилия, инициалы, дата)

Консультант _____

(должность, подпись, фамилия, инициалы, дата)

Обучающийся Тимофеев А.А. 21.06.24

(подпись, фамилия, инициалы, дата)

Воронеж

2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)

Форма обучения заочная

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись)

Черняева С. Н.
(ФИО)

2024

**Задание
на выпускную квалификационную работу**

Вид работы ВКР бакалавра
(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Тимофееву Александру Александровичу
(фамилия, имя, отчество)

Тема Разработка приложения по учету рабочего времени сотрудников предприятия (на примере на примере Филиал ФБУЗ «Центр гигиены и эпидемиологии»

Утверждена приказом ректора университета от _____ 20____, № _____

Срок сдачи законченной работы _____ 20____

Исходные данные (или цель ВКР):

Разработать приложение по учету рабочего времени сотрудников предприятия

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):

(актуальность темы, цели и задачи ВКР; аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)

– Введение. Актуальность выбранной темы, цель и задачи ВКР
(наименование вопроса, раздела и его краткое содержание)

– Исследовательский раздел.
(наименование вопроса, раздела и его краткое содержание)

Общая характеристика организации, организация проведения работ, порядок и методы проведения работ, теоретические основы информационных систем учета рабочего времени сотрудников, постановка задачи, анализ существующих решений, обоснование выбора разрабатываемых решений, обоснование технического обеспечения.

– Проектный раздел.

(наименование вопроса, раздела и его краткое содержание)

Информационное обеспечение задачи, информационная модель и ее описание, используемые классификаторы и системы кодирования, характеристика входной информации, характеристика базы данных, характеристика выходной информации, программное обеспечение задачи, создание проекта, описание программных модулей, руководство пользователя.

– Заключение. Выводы по работе в целом. Оценка степени решения поставленных задач

(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Обзор предметной области
4. Теоретические основы информационных систем учета рабочего времени сотрудников
5. Постановка задачи
6. Анализ существующих решений
7. Проектный раздел
8. Проектный раздел Диаграмма потоков Гейна/Сарсона
9. Структура базы данных
10. Описание программных модулей
11. Реализация
12. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
2. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
3. _____
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: _____ 20____

Задание согласовано и принято к исполнению: _____ 20____

Руководитель ВКР: к. ф.-м. н., доцент Черняева С. Н.

(должность, ученая степень, ученое звание, ФИО)

_____ (подпись)

Обучающийся: ИТ- 4-2 Тимофеев А.А.

(учебная группа, ФИО)

_____ (подпись)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	7
1.1 Общая характеристика организации	7
1.2 Организация проведения работ	9
1.2.1 Порядок и методы проведения работ	9
1.3 Теоретические основы информационных систем учета рабочего времени сотрудников	10
1.4 Постановка задачи	12
1.5 Анализ существующих решений	13
1.6 Обоснование выбора разрабатываемых решений	18
1.7 Обоснование технического обеспечения	18
2 ПРОЕКТНЫЙ РАЗДЕЛ	22
2.1 Информационное обеспечение задачи	22
2.1.1 Информационная модель и ее описание	22
2.1.2 Используемые классификаторы и системы кодирования	24
2.2 Характеристика входной информации	25
2.3 Характеристика базы данных	26
2.4 Характеристика выходной информации	30
2.5 Программное обеспечение задачи	32
2.6 Описание программных модулей	33
2.7 Руководство пользователя	35
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	41
ПРИЛОЖЕНИЕ А	45

ВВЕДЕНИЕ

В современной информационной эпохе разработка приложений для учета рабочего времени сотрудников становится неотъемлемой составляющей процесса автоматизации бизнеса. Такие приложения предоставляют возможность не только отслеживать отработанные часы, но и анализировать их распределение, учитывать перерывы и отпуска, контролировать выполнение рабочих норм и стандартов. Более того, они обеспечивают удобство для сотрудников и руководства предприятия, делая процесс учета времени более прозрачным и эффективным.

Многие российские компании уже давно пришли к осознанию необходимости учета рабочего времени сотрудников. А вследствие внедрения информационной системы нереализованная функция регулирования рабочего времени персонала стала воплощенным достоинством, упрощая контроль затрат на управление персоналом и повышая эффективность использования рабочего времени.

Согласно данным различных исследований, от 30 до 50% рабочего времени тратится не на рабочий процесс. Почти половина временных потерь происходит по причине неграмотного планирования, а еще примерно треть – от слабого надзора за сотрудниками. Установленная в организации система учета рабочего времени помогает сразу же дисциплинировать сотрудника – ведь если сотрудник при приеме на работу знает, что время его прихода и ухода фиксируется автоматически, и за все опоздания и отлучки ему придется нести ответственность, он внимательнее относится к своему рабочему времени.

Из вышесказанного можно сделать вывод, что учет рабочего времени сотрудников на предприятиях является одним из основополагающих элементов результативной работы и достижения поставленных предприятием целей в кратчайшие сроки. Этот факт еще раз подчеркивает актуальность выбранной темы.

Объектом данного исследования является Филиал федерального бюджетного учреждения здравоохранения «Центр гигиены и эпидемиологии в Липецкой области» в Лебедянском районе.

Предмет исследования: информационная система учета рабочего времени сотрудников предприятия.

Цель проекта: Разработка приложения по учету рабочего времени сотрудников предприятия.

Для достижения поставленной цели необходимо решить следующие задачи:

- выполнить анализ предметной области;
- исследовать теоретические основы учета рабочего времени сотрудников;
- провести анализ существующих разработок;
- провести обоснование выбранных проектных решений;
- провести обоснование технического обеспечения;
- разработать информационную модель;
- спроектировать структуру базы данных;
- описать программные модули;
- разработать приложения по учету рабочего времени сотрудников предприятия;
- разработать руководство пользователя.

При выполнении выпускной квалификационной работы были использованы следующие методы исследования: анализ, синтез, дедукция и моделирование.

Данная выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников и приложений.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Общая характеристика организации

Организация Филиал федерального бюджетного учреждения здравоохранения «Центр гигиены и эпидемиологии в Липецкой области» является некоммерческой организацией, обеспечивающей деятельность Управления Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека по городу Лебедянь.

Место нахождения: 399610 Липецкая область, г. Лебедянь, ул. Антонова, 12 А.

Сайт: www.cge48.ru

Виды деятельности:

Основной (по коду ОКВЭД): 85.14.5 – Деятельность учреждений санитарно-эпидемиологической службы.

Предоставляет следующие виды услуг:

1. Лабораторные исследования, испытания, измерения;
2. Специальная оценка условий труда;
3. Производственный контроль;
4. Оценка профессиональных рисков на рабочих местах;
5. Оформление, проверка, выдача и учет личных медицинских книжек;
6. Дистанционное гигиеническое обучение;
7. Безопасность строящегося жилья и экспертиза строительных материалов;
8. Консультационный центр по защите прав потребителей;
9. Разработка и помощь во внедрении методики НАССР на производстве;
10. Санитарно-эпидемиологическая экспертиза проектной документации;
11. Оценка эффективности систем вентиляции.

На рисунке 1.1 представлена организационная структура Филиала федерального бюджетного учреждения здравоохранения «Центр гигиены и эпидемиологии в Липецкой области».

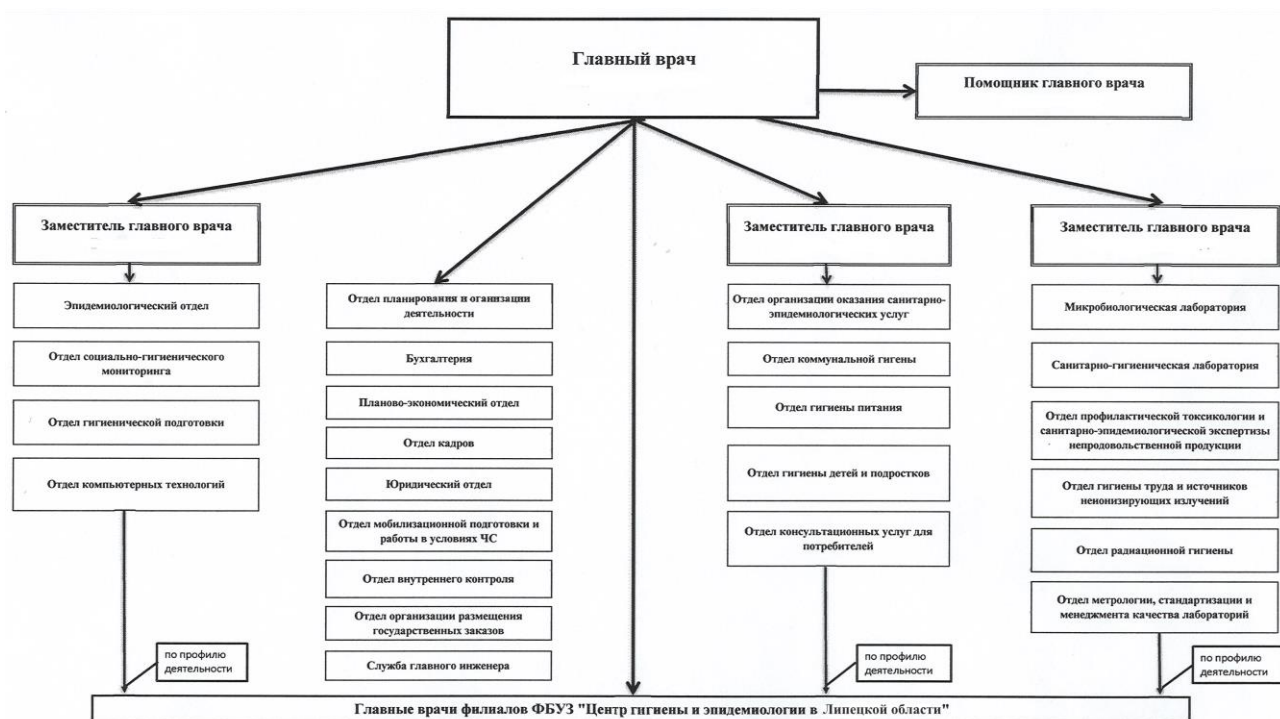


Рисунок 1.1 – Организационная структура

ФБУЗ «Центр гигиены и эпидемиологии в Липецкой области» обладает уникальными технологиями по многим направлениям профилактической медицины является как первопроходцем, так и признанным лидером. Так, например, система регистрации случаев инфекционных и паразитарных заболеваний (отравлений) имеются лишь в нескольких субъектах РФ.

ФБУЗ «Центр гигиены и эпидемиологии в Липецкой области» выполняет любые исследования и измерения, необходимые для: плана производственного контроля вашего объекта; проведения гигиенического обучения, аттестации и выдачи личных медицинских книжек; специальной оценки условий труда на рабочих местах персонала вашей фирмы; проведения оценки риска; создания безопасных и комфортных условий жизнедеятельности; обеспечения санитарно-эпидемиологического благополучия рабочих коллективов; защиты прав потребителей и потребительского рынка.

1.2 Организация проведения работ

Для успешной разработки приложения учета рабочего времени (далее УРВ) в ФБУЗ «Центр гигиены и эпидемиологии в Липецкой области» необходимо провести определенные организационные мероприятия.

Основные этапы, которые будут предприняты:

– Постановка задачи: определение конкретных целей и задач разработки приложения УРВ, уточнение требований к функциональности и определение критериев успешной реализации.

– Изучение предметной области: анализ существующих систем учета рабочего времени и приложений для управления персоналом. Это позволит лучше понять особенности и требования к разрабатываемому приложению УРВ.

– Сбор и анализ требований: проведение опросов и интервью с потенциальными пользователями для выявления и уточнения требований к функциональности и интерфейсу приложения УРВ.

– Проектирование архитектуры и дизайна: разработка общей архитектуры приложения УРВ, определение основных компонентов и модулей. Также будет разработан дизайн пользовательского интерфейса, учитывающий принципы удобства использования и легкости навигации.

– Оценка эффективности и преимуществ: оценка эффективности разработанного приложения УРВ на основе обратной связи от пользователей и анализа данных о рабочем времени сотрудников. Это позволит оценить вклад приложения в улучшение процессов учета времени и управления персоналом на предприятии.

1.2.1 Порядок и методы проведения работ

Для разработки приложения учета рабочего времени сотрудников используется определенный порядок и метод проведения работ, который представлен основными этапами.

Этапы проведения работ:

– Изучение предметной области.

- Сбор требований.
- Проектирование архитектуры.
- Реализация приложения.
- Тестирование приложения.

Для успешной разработки приложения необходимо выполнить все этапы.

1.3 Теоретические основы информационных систем учета рабочего времени сотрудников

Информационная система – это совокупность, состоящая из одного либо нескольких компьютеров, соответствующих средств программирования, операторов, физических процессов, средств телекоммуникаций и других, образующих автономное целое, способное осуществлять обработку или передачу данных. Другими словами, информационная система - это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Информационные системы могут быть использованы в различных сферах деятельности. Одной из таких областей является учет рабочего времени сотрудников.

Учет рабочего времени – очень трудоемкий процесс, особенно для крупных предприятий с большой численностью сотрудников, имеющих разный уровень квалификации. Именно поэтому для упрощения этой процедуры применяются автоматизированные системы учета рабочего времени. Данная система позволяет своевременно и оперативно упорядочить данные о сотрудниках на предприятии, вне зависимости от сферы и масштабов деятельности.

Основной принцип работы таких систем – учет общего количества отработанных сотрудником часов, опозданий, командировок и отпусков, отгулов и больничных. В любой момент система позволяет получать отчеты по всем вышеперечисленным параметрам.

Информационные системы учета рабочего времени решает следующие задачи:

- экономия денежных средств;
- повышение уровня трудовой дисциплины;
- обеспечение справедливого начисления заработной платы;
- упрощение процедуры подготовки отчетности;
- сокращение времени и трудоемкости планирования рабочего времени;
- объективность в принятии решений по управлению персоналом;
- содействие рациональной организации и эффективной реализации бизнес-процессов компании.

Функциональность автоматизированных систем учета рабочего времени:

- точный автоматизированный учет рабочего времени;
- формирование графиков рабочего времени;
- автоматизированный учет опозданий и ранних уходов;
- возможность задавать дополнительные параметры настройки работы системы под специфику деятельности компании;
- определение каждому работнику как нормированного, так и не нормированного графика работы.

Для каждого сотрудника определяется внутри предприятия, время его нахождения, которое считается в системе в качестве рабочего, и выдается идентификатор (как правило, пластиковая карта-пропуск), хотя учет рабочего времени может вестись и с использованием в качестве идентификаторов отпечатка пальца сотрудника или набора PIN-кода. Учет рабочего времени осуществляется по средствам сравнения зафиксированных приходов и уходов в системе с заданным режимом рабочего времени конкретного сотрудника.

Процедура регистрации занимает всего несколько секунд: для этого сотруднику достаточно лишь поднести свою карточку к терминалу или нажать кнопку. Система учета рабочего времени также ведет статистику и формирует отчеты. Результаты учета времени могут импортироваться в программу расчета

заработной платы. В зависимости от постановки задачи, система может формировать отчеты по различным критериям: по приходам и уходам, по типам нарушений (приход после начала рабочего времени, ранний уход, и т.д.), по переработкам, по интервалам, по сотрудникам, по подразделениям.

Анализируя информацию, представленную выше, можно сделать вывод, что использование информационной системы автоматизированного учета рабочего времени сотрудников имеет следующие положительные стороны:

- Экономия времени. Когда старые способы учета рабочего времени забирают очень много времени, с программой его можно значительно сэкономить;

- Данные собираются автоматически. Руководителям не нужно делать постоянные обходы, фиксировать все данные самостоятельно или полагаться на совесть персонала в заполнении отчетностей. Все объективные данные собираются программой автоматически;

- Контролируются все нарушения. Опоздания, прогулы, уходы раньше положенного – каждая минута с помощью программы будет под контролем. Когда человек может что-то упустить, или это сотрудник, или даже руководитель, то программа зафиксирует все.

1.4 Постановка задачи

Цель выпускной квалификационной работы заключается в разработке приложения направленное на улучшение управления рабочим временем сотрудников и оптимизацию процессов работы организации. Основные требования к проектируемой системе обработки данных состоят в следующем:

- Устранение недостатков в учете рабочего времени;
- Повышение эффективности управления персоналом;
- Сокращение времени и затрат на обработку данных;
- Повышение качества обработки информации.

Этапы решения задачи включают в себя:

- Подготовка и установка приложения на сервере и клиентских устройствах.
- Регистрация пользователей и выдача им учетных записей.
- Разработка и реализация функционала для регистрации входов и выходов сотрудников.
- Создание административных функций для управления пользователями и администрирования приложения.

На таблице 1.1 представлено описание входных показателей.

Таблица 1.1 – Описание входных показателей

№п/п	Наименование входного показателя	Идентификатор входного показателя
1	Время входа	Дата и время
2	Время выхода	Дата и время

Входными показателями являются время входа сотрудника на работу и время выхода с работы, данные сохраняются в базу данных в виде даты и времени в формате день, месяц, год, часы, минуты и секунды.

На таблице 1.2 представлено описание выходного показателя.

Таблица 1.2 – Описание выходного показателя

№п/п	Наименование выходного показателя	Идентификатор выходного показателя
1	Время на работе	Время

Выходным данным является учет времени, сколько сотрудник провел времени на работе. Выходным результатом отображается в формате часы, минуты и секунды.

1.5 Анализ существующих решений

В процессе разработки нового приложения важно изучить существующие программные продукты в той же сфере деятельности. Рассмотрим следующие программные продукты: «AntiBAG Tahograf» и «Intempus Timeregistrering».

1. Приложение «AntiBAG Таhограф»

В приложении есть три активных области.

При запуске отображается главное окно, представленное на рисунке 1.2.

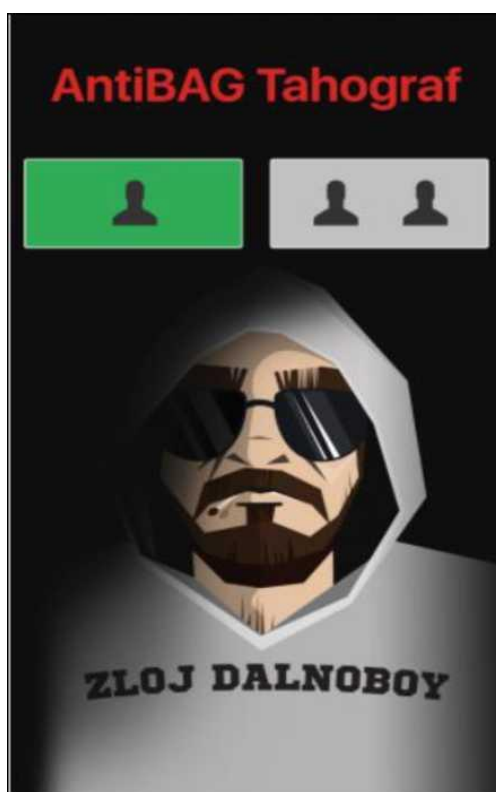


Рисунок 1.2 – Главное окно

Первая область – это напоминание ежедневного ввода страны начала и конца работы, представлено на рисунке 1.3.



Рисунок 1.3 – Первая область

Вторая область – это все необходимые таймеры для работы, представлено на рисунке 1.4.

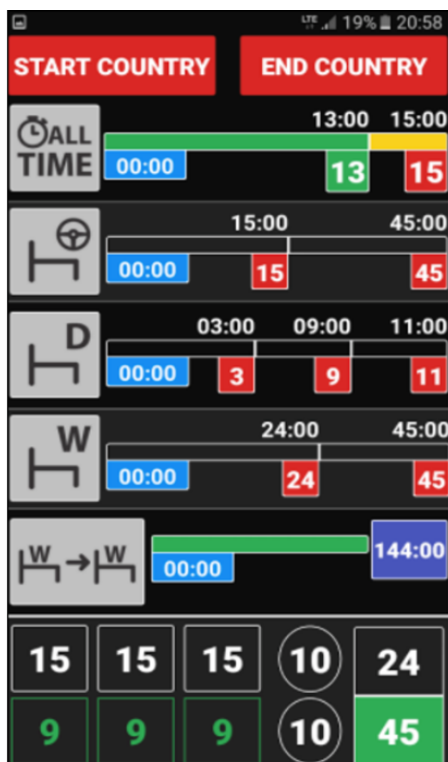


Рисунок 1.4 – Вторая область

Третья область – это рабочий календарь, представлен на рисунке 1.5.

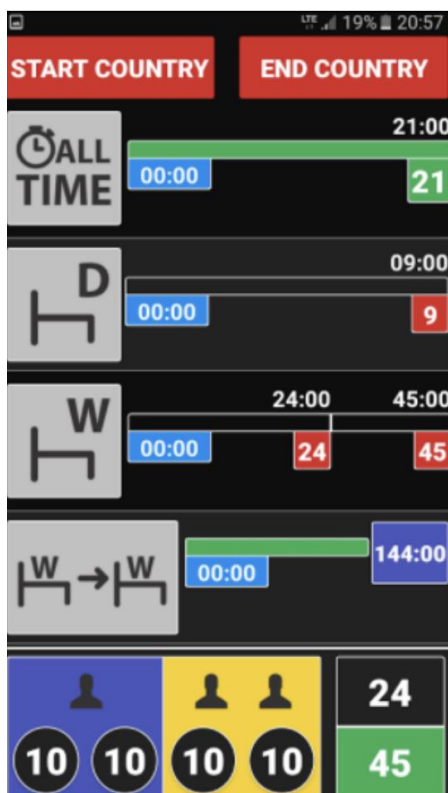


Рисунок 1.5 – Третья область

2. Приложение «Intempus»

«Intempus» – приложение для регистрации таймера, которое может записывать всю работу – и немного больше. Администрация Intempus, веб-система администрирования, предоставляет администрации Intempus обзор времени регистрации ваших сотрудников. Обзор позволяет легко извлекать файлы Excel. Оригинальный язык Датский, перевод выполнен с помощью Google переводчика.

Приложение «Intempus» представлено на рисунке 1.6.

При изучении этих аналогов проведен анализ их преимуществ и недостатков, с целью определить, какие возможности и функции могут быть реализованы лучше в новом приложении, и как их можно улучшить.

При помощи данного приложения руководители получают информацию о работе сотрудников в офисе или удалено.

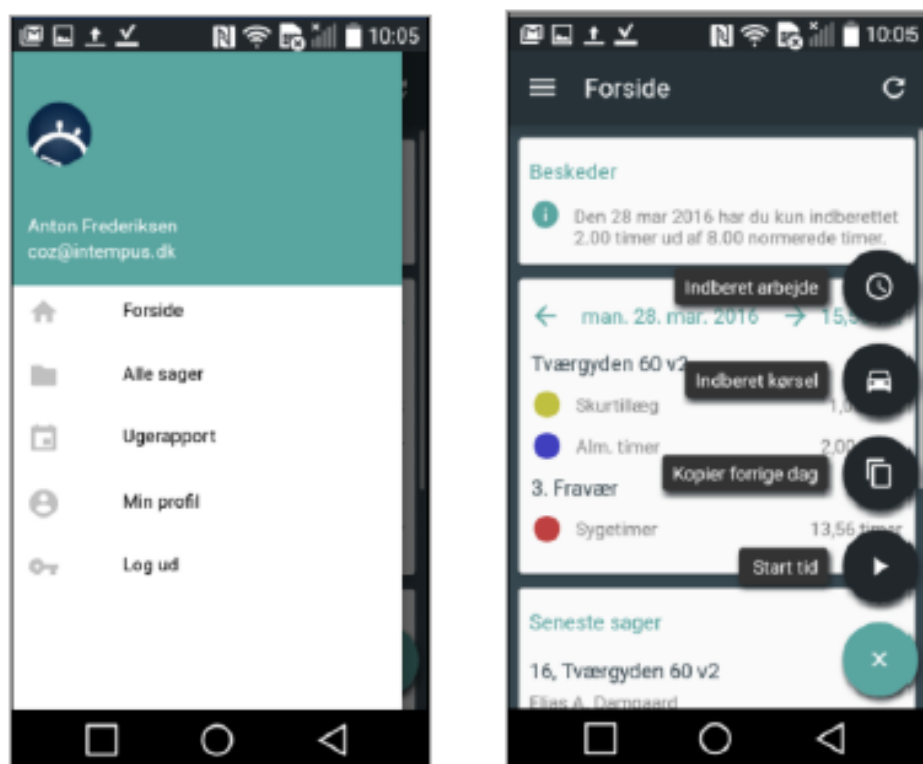


Рисунок 1.6 – Приложение «Intempus»

Руководитель может просматривать чем занимается сотрудник в настоящее время, получать скриншоты с его компьютера или телефона,

заблокировать сотруднику доступ к вредоносным сайтам, отслеживать время, которое он проводит на сайте или в приложении.

Достоинства и недостатки сайта приложения представлены в таблице 1.3.

Таблица 1.3 – Достоинства и недостатки приложения

Достоинства	Недостатки
Можно отслеживать работу сотрудника	Присутствует платная версия
Отчетность	Работает только онлайн
	Не подходит для предприятия
	Сложен в использовании

Для оценки особенностей перечисленных программных продуктов и разрабатываемое приложение соответствия их функциональных возможностей требованиям, произведем сравнительный анализ вышеуказанных программных продуктов, представлен в таблице 1.4.

Таблица 1.4 – Сравнительный анализ

Функции/Характеристики	Приложение «AntiBAG Tahograf»	Приложение «Intempus»	разрабатываемое приложение
Учет времени	+	+	+
Удобство пользования	-	-	+
Работа оффлайн	-	-	+
Дизайн	5	5	9
Удобство	7	7	9
Автоматизация	7	8	8

Исходя из таблицы сравнения, разрабатываемое приложение по учету рабочего времени сотрудников получило высокую оценку по характеристикам дизайна, автоматизации и удобства использования.

Его дизайн оценивается на уровне 9 из 10, что указывает на соответствие современным стандартам и аккуратности в оформлении интерфейса. Оценка автоматизации составляет 8 из 10.

Приложение получило высокую оценку 9 из 10 в категории удобства использования, что указывает на комфортность и практичность в работе с ним.

1.6 Обоснование выбора проектных решений

В процессе выбора технологий для разработки приложения были учтены ряд факторов, включая гибкость, производительность и удобство использования. В результате анализа были выбраны следующие технологии, которые наилучшим образом соответствуют требованиям проекта:

Технология C# Windows Forms была выбрана благодаря своей простоте и эффективности в создании графических пользовательских интерфейсов.

Windows Forms предоставляет обширный набор инструментов для разработки функциональных и привлекательных по внешнему виду приложений. Благодаря этой технологии, можно быстро создать приложение с интуитивно понятным и удобным интерфейсом для пользователей.

Среда разработки Visual Studio была выбрана из-за своей широкой популярности и обширного набора интегрированных инструментов. Visual Studio предоставляет удобное и эффективное рабочее окружение для разработчиков, позволяя с легкостью создавать, отлаживать и развертывать приложения.

Выбор базы данных SQLite обоснован его легковесностью, скоростью и простотой в использовании. SQLite обеспечивает надежное хранение данных приложения и обладает хорошей производительностью даже при работе с большим объемом информации. Благодаря интеграции с C# Windows Forms, использование SQLite позволяет создать компактное и быстродействующее приложение, не требующее дополнительной установки серверов баз данных.

1.7 Обоснование технического обеспечения

Техническое обеспечение информационных систем – это совокупность технических средств, которые обеспечивают работу ИС, соответствуют документации на эти средства и технологические процессы [26]. От того насколько рационально будет спроектирован данный технологический процесс, зависит дальнейшее снижение стоимостных и трудовых затрат.

К элементам технического обеспечения относятся:

- комплекс технических средств;
- организационные формы использования технических средств;
- инструкция по использованию техники.

Комплекс технических средств (КТС) – это совокупность взаимосвязанных технических средств, предназначенных для автоматизированной обработки данных.

КТС является одной из основных составных частей АИС, ее материально-технической базой, на которой реализуются все задачи системы и поэтому к комплексу технических средств предъявляются следующие требования:

- высокая производительность;
- надежность;
- защита от несанкционированных доступов;
- эффективность функционирования при допустимых стоимостных характеристиках;
- минимизация затрат на приобретение и эксплуатацию;
- рациональное распределение по уровням обработки.

С помощью технических средств (ТС) реализуются функции автоматизированного сбора информации от первоисточников, ее анализ и представление, хранение, обработка, отображение, передача.

Для осуществления основных функций технические средства должны отвечать следующим требованиям:

- быть информационно совместимыми между собой и обслуживающим персоналом, что обеспечивается совпадением форм представления информации, видов машинных носителей, языков, кодов, вводом данных в ТС;

- структура КТС должна соответствовать структуре управления объектом, обеспечивая автоматизированное управление выполняемых функций, в том числе функций контроля;

- для обеспечения быстрого решения задач ТС должны быть качественной конструкции, современного и удобного дизайна, иметь дружественный интерфейс для работы пользователя;

– должен соблюдаться принцип экономичности выбора и использования ТС, т. е. минимум затрат на создание (приобретение) технических средств, их эксплуатацию и используемых для размещения площадей [27].

В состав технического обеспечения входит документация, которую можно условно разделить на три группы:

– общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;

– специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;

– нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение.

При предполагаемом внедрении подсистемы, необходимо учесть важнейший фактор, на основе которого предполагается соответствие потребностей в ресурсах имеющемуся на предприятии техническому обеспечению.

Характеристики одного из компьютеров, используемого в организации представлены в таблице 1.5.

Таблица 1.5 – Технические характеристики

№	Наименование	Характеристика
1	Системный блок	CityC60
2	Блок питания	675 Вт
3	Материнская плата	Intel(R) i5 G3220T @ 2.60GHz, 2594 МГц
4	Операционная система	Microsoft Windows 7
5	Оперативная память	DDR4 2x4Гб Kingmax PC10600
6	Процессор	IntelCorei3–3220 3.3ГГц 3М6LGA1155 55WBOX
7	Жесткий диск	WD5000LPVX-22V0T (465 Гб)
9	Видеокарта	Intel(R) HD Graphics (1024 Мб)
11	Монитор	АОС E2060SW 49,5 см (19.5") 1600 x 900 пик
12	Модемы	Модем SHDSL ZyXEL Prestige 791R v2,100 Mbps
13	Клавиатура	Gemius, IBM enhanced (101- or 102-key) keyboard
14	Мышь	Gemius, HID
15	Принтер лазерный	HP LaserJet 400 M401 PCL 6

Вышеприведенные требования к аппаратной части являются стандартными и не требуют больших издержек. Техника с указанными характеристиками фактически уже используется в организации.

2 ПРОЕКТНЫЙ РАЗДЕЛ

2.1 Информационное обеспечение задачи

2.1.1 Информационная модель и ее описание

При разработке информационной модели для приложения учета рабочего времени сотрудников была применена методика, которая предполагает моделирование нового варианта организации информационной системы предметной области, то есть «КАК ДОЛЖНО БЫТЬ». Основные аспекты информационной модели включают:

- Полный состав информации, необходимой для решения комплекса задач данного приложения. Это включает в себя данные о времени входа и выхода сотрудников, а также информацию об активности администратора.

- Отражение информации на всех типах носителей, таких как база данных SQLite для хранения данных о сотрудниках и их рабочем времени.

- Отражение процесса преобразования информации, начиная от получения первичных данных о входе и выходе сотрудников до формирования отчетов об их рабочем времени.

- Состав исходных первичных документов, таких как логины и пароли сотрудников, а также информация об их регистрации и удалении.

- Источники и способы получения первичной информации, которые включают в себя ввод данных сотрудниками через интерфейс приложения.

- Способы выдачи результатной информации, включая предоставление администратору отчетов о рабочем времени сотрудников.

- Состав результатных документов, таких как отчеты и статистика о времени, проведенном на работе, и их адресаты.

- Взаимосвязи между входными, промежуточными и результатными информационными потоками, а также задачами, реализуемыми в приложении учета рабочего времени сотрудников, представлены на структурно-функциональной диаграмме.

Модель «КАК ДОЛЖНО БЫТЬ» показана на рисунке 2.1.

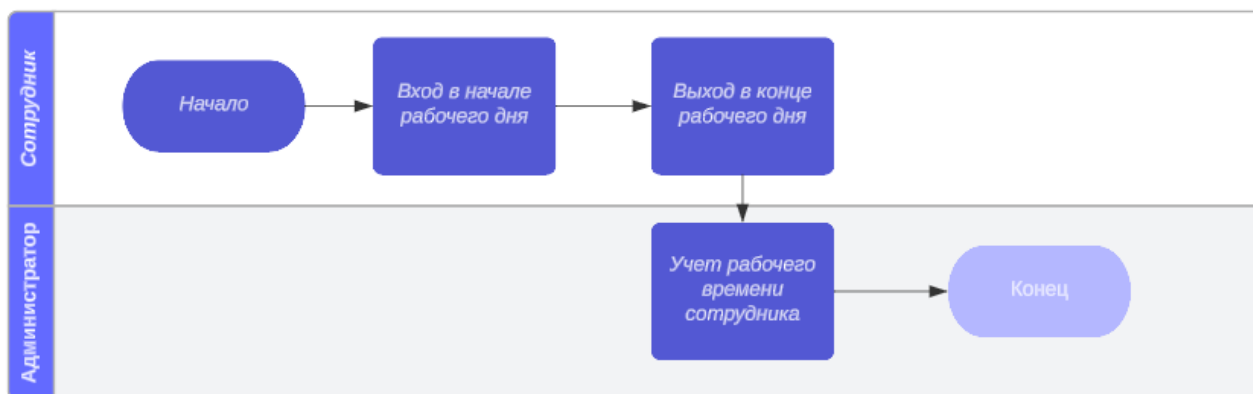


Рисунок 2.1 – Модель «КАК ДОЛЖНО БЫТЬ»

Информационная модель – это модель объекта, которая определяет параметры и переменные объекта, их взаимосвязи, входы и выходы объекта, и может имитировать возможные состояния объекта, предоставляя информацию об изменении входных значений. Информационная модель приложения, является основным инструментом для анализа и понимания работы системы. Она позволяет определить ключевые параметры и переменные, которые влияют на функционирование объекта. Информационная модель приложения показана на рисунке 2.2.



Рисунок 2.2 – Информационная модель

Диаграмма потоков по методологии Гейна/Сарсона представлена на рисунке 2.3.

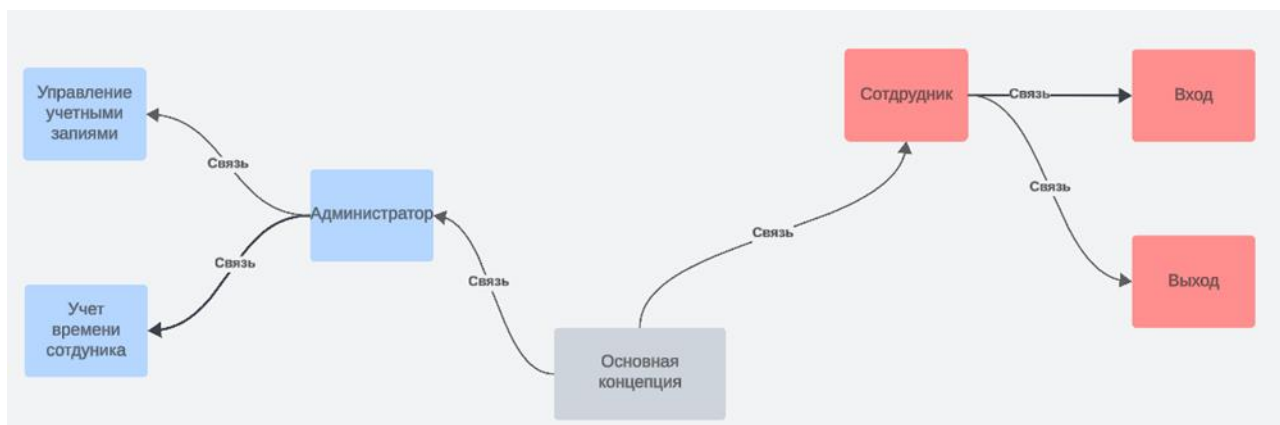


Рисунок 2.3 – Диаграмма потоков Гейна/Сарсона

2.1.2 Используемые классификаторы и системы кодирования

Для решения задачи учета рабочего времени сотрудников в приложении используются следующие классификаторы и системы кодирования.

1. Классификатор сотрудников.

– Наименование кодируемого множества объектов: сотрудники предприятия.

– Значность кода: уникальный идентификатор сотрудника.

– Система кодирования: порядковая.

– Система классификации: иерархическая.

– Вид классификатора: внутренний предприятий.

Описание: Классификатор сотрудников используется для идентификации каждого сотрудника предприятия. Каждому сотруднику присваивается уникальный табельный номер, который используется в системе для идентификации и отслеживания его активности. Классификатор рабочего времени.

2. Классификатор рабочего времени.

– Наименование кодируемого множества объектов: рабочее время.

– Значность кода: уникальный идентификатор записи времени.

– Система кодирования: серийная.

– Система классификации: многоаспектная.

– Вид классификатора: специализированный.

Описание: Классификатор рабочего времени используется для отслеживания временных записей сотрудников. Каждая запись времени имеет уникальный идентификатор, который содержит информацию о дате, времени начала и окончания работы сотрудника.

2.2 Характеристика входной информации

Приложение учета рабочего времени сотрудников использует следующую входную информацию:

1. Регистрационная форма сотрудника:

- Форма документа: форма для регистрации нового сотрудника.
- Показатели: Должность, логин, пароль.
- Источник: пользовательский ввод и всплывающее меню.
- Структура: текстовые поля для ввода должности, поля для ввода логина

и пароля.

- Объем данных: небольшой
- Частота возникновения: при добавлении нового сотрудника.

2. Форма записи рабочего времени:

- Форма документа: форма для ввода записей рабочего времени.
- Показатели: дата, время начала и окончания работы.
- Источник: автоматическая запись по кнопке
- Структура: кнопки для начала и окончания рабочего дня

- Объем данных: небольшой
- Частота возникновения: ежедневно для каждого сотрудника.

3. Экранная форма регистрационной формы сотрудника:

– Макет экранной формы: представлен на экране в виде окна с текстовыми полями для ввода должности, а также полями для ввода логина и пароля.

– Особенности организации макета: четкое разделение на рабочую и служебную зоны, подсказки для заполнения каждого поля.

– Подключаемые справочники: автоматически подключается справочник должностей предприятия для выбора должности сотрудника.

Первичные данные предназначены для отображения процессов в реальном времени и представляет всю постоянную и оперативную информацию, необходимую для решения экономических задач и выработки управленческих решений. К числу основных требований, предъявляемых к первичным данным, можно отнести следующие: не избыточность и полноту информации для решения задач, высокую достоверность и своевременность собираемой информации.

Кроме того, первичная информация должна быть сгруппирована таким образом, чтобы учитывались требования удобства для последующей обработки данных.

2.3 Характеристика базы данных

Системы управления базами данных очень удобны в использовании. Это достигается благодаря наличию следующих свойств [11]:

Быстродействие системы. Система поиска позволяет легко находить нужную информацию даже в больших массивах данных. В обычных электронных таблицах это делать сложнее.

Простота получения и изменения данных. Для добавления и обновления информации оператору требуется совершить минимальное количество действий.

Безопасность. Большая часть баз данных настраивается под разные уровни доступа. Возможно разрешение либо запрет просмотра или редактирования записей определенным пользователям.

Многопользовательский доступ. Современные СУБД поддерживают одновременную работу нескольких пользователей с одной базой.

Поддержка очень больших объемов данных. В БД можно хранить действительно огромные массивы информации. Далее будет подробно разъяснено, что такое запись. База данных позволяет обрабатывать информацию, содержащуюся в миллионах таких записей.

Планирование структуры базы данных – это процесс, который включает создание схемы базы данных и установление ограничений целостности, необходимых для ее правильного функционирования.

Для создания конкретного вида и содержания концептуальной модели базы данных выбирается соответствующий формальный аппарат. Чаще всего используются графические нотации, похожие на ER-диаграммы.

Обычно концептуальная модель базы данных включает в себя [21]:

- описание информационных объектов или понятий предметной области и связей между ними;
- описание ограничений целостности, которые определяют допустимые значения данных и связей между ними.

Логическое проектирование представляет собой создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель представляет собой набор схем отношений, включающих в себя первичные ключи, а также связи между отношениями, которые представляют собой внешние ключи.

Преобразование концептуальной модели в логическую модель обычно выполняется с помощью формальных правил. Этот процесс может значительно автоматизироваться.

Во время логического проектирования учитывается конкретная модель данных, но может не учитываться специфика конкретной СУБД.

Физическое проектирование представляет собой создание схемы базы данных для конкретной СУБД. Некоторые специфичные особенности конкретной СУБД, такие как ограничения на названия объектов базы данных и поддерживаемые типы данных, могут быть учтены. Кроме того, физическое проектирование включает в себя решения, связанные с физической средой хранения данных, такие как выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, а также создание индексов и т.д.

База данных учета рабочего времени сотрудников состоит из двух таблиц.

В первой таблице хранятся данные о пользователях, представлено в таблице 2.1.

Таблица 2.1 – Таблица данные о пользователях «Users»

Поле	Тип	Ключ	Характеристика
Id	int	*	Id пользователя
Username	text		Логин пользователя
Password	text		Пароль пользователя
IsAdmin	int		Проверка на админа

Во второй таблице данные учета времени работника, представлено в таблице 2.2.

Таблица 2.2 – Таблица учет времени работника «Entries»

Поле	Тип	Ключ	Характеристика
Id	int	*	Id пользователя
Username	text		Логин пользователя
EntryType	text		Вход/Выход пользователя
EntryTime	datetime		Время входа и выхода

Структура базы данных представлена на рисунке 2.4.

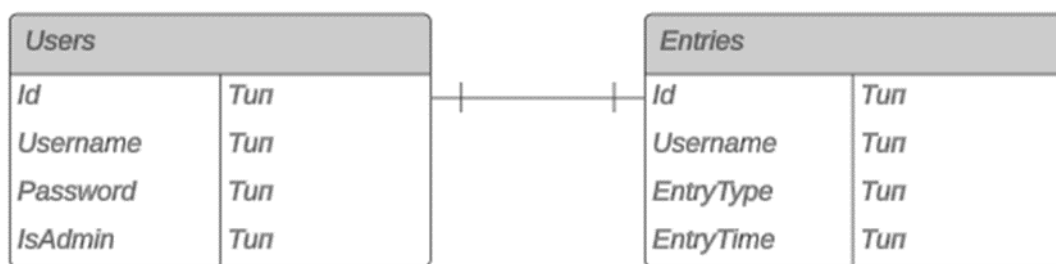


Рисунок 2.4 – Структура базы данных

База данных в приложении создается автоматически при первом запуске.

За хранение и создание таблицы Users отвечает часть кода:

```

private void InitializeDatabase()
{
    connection = new SQLiteConnection("Data Source=users.db;Version=3;");
    connection.Open();

    string createUserTableQuery = "CREATE TABLE IF NOT EXISTS Users (Id INTEGER
PRIMARY KEY AUTOINCREMENT, Username TEXT UNIQUE, Password TEXT, IsAdmin INTEGER)";

```

```

        ExecuteQuery(createUserTableQuery);
    }

```

При вводе данных проверяется база и открывается следующее окно:

```

private bool AuthenticateUser(string username, string password)
{
    string query = $"SELECT * FROM Users WHERE Username='{username}' AND
Password='{password}'";
    using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
    {
        using (SQLiteDataReader reader = cmd.ExecuteReader())
        {
            return reader.Read();
        }
    }
}

```

С помощью `bool isAdmin = IsAdminUser(username);` проверяем, является ли пользователь администратором, при неправильном вводе выводится сообщение `MessageBox.Show("Неверный логин или пароль.");`

Для создания таблицы `Entries` отвечает часть кода:

```

private void InitializeDatabase()
{
    connection = new SQLiteConnection("Data Source=users.db;Version=3;");
    connection.Open();

    string createEntryTableQuery = "CREATE TABLE IF NOT EXISTS Entries (Id
INTEGER PRIMARY KEY AUTOINCREMENT, Username TEXT, EntryType TEXT, EntryTime DATETIME)";
    ExecuteQuery(createEntryTableQuery);
}

```

С помощью данной части кода, записывается статус, что сотрудник вошел и вышел:

```

private void PopulateListBoxes()
{
    listBox1.Items.Clear();
    listBox2.Items.Clear();

    string selectQuery = $"SELECT EntryType, EntryTime FROM Entries WHERE
Username = '{username}'";
    using (SQLiteCommand cmd = new SQLiteCommand(selectQuery, connection))
    {
        using (SQLiteDataReader reader = cmd.ExecuteReader())
        {

```

```

while (reader.Read())
{
    string entryType = reader["EntryType"].ToString();
    string entryTime = reader["EntryTime"].ToString();

    if (entryType == "Вход")
    {
        listBox1.Items.Add(entryTime);
    }
    else if (entryType == "Выход")
    {
        listBox2.Items.Add(entryTime);
    }
}
}
}
}

```

Данная часть отвечает за запись даты и времени в таблицу:

```

private void RecordEntry(string entryType)
{
    string entryTime = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
    string query = $"INSERT INTO Entries (Username, EntryType, EntryTime)
VALUES ('{username}', '{entryType}', '{entryTime}')";
    ExecuteQuery(query);
    PopulateListBoxes();
}

```

2.4 Характеристика выходной информации

Таблица «Entries»:

- Поля: Id (идентификатор записи), Username (имя пользователя), EntryType (тип записи: Вход или Выход), EntryTime (время события).
- Источник данных: записи о входах и выходах сотрудников в систему, хранятся в базе данных.
- Основана на: информации, полученной от пользователей через интерфейс приложения.
- Объем данных: каждая запись содержит информацию о конкретном событии входа или выхода сотрудника.

– Подлежит дальнейшему хранению: да, таблица используется для сохранения истории входов и выходов.

Таблица «Users»:

– Поля: Id (идентификатор пользователя), Username (имя пользователя), Password (пароль пользователя), IsAdmin (флаг администратора).

– Источник данных: данные о зарегистрированных пользователях, хранятся в базе данных.

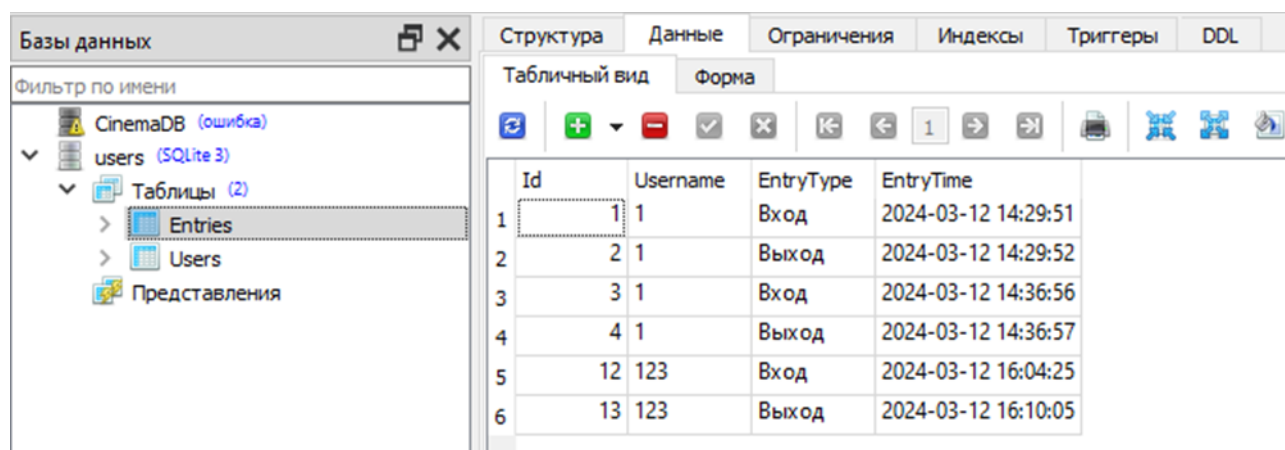
– Основана на: информации, введенной администратором при регистрации новых пользователей.

– Объем данных: каждая запись содержит информацию о пользователе, включая его учетные данные и статус администратора.

– Подлежит дальнейшему хранению: да, таблица используется для аутентификации пользователей и управления доступом.

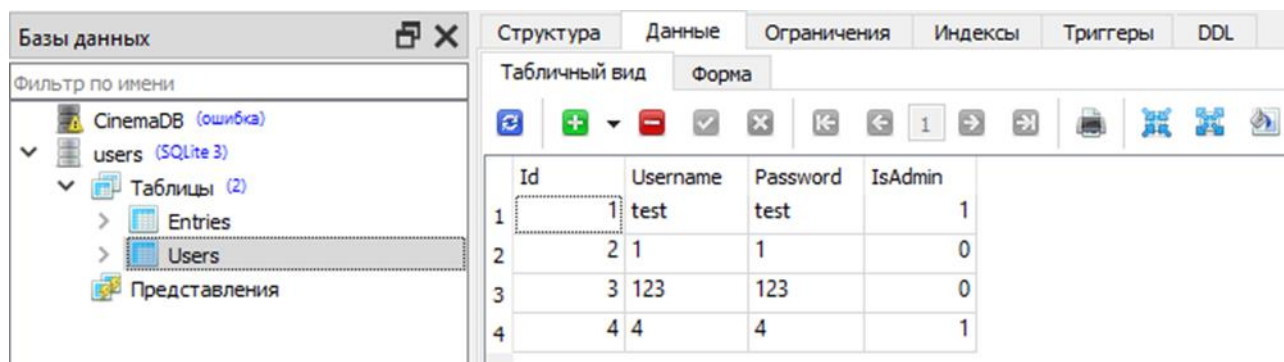
Эти таблицы представляют собой основу для формирования отчетов и ведомостей о рабочем времени сотрудников и обеспечивают доступ к информации об учетных записях пользователей приложения.

Результат таблиц представлен на рисунках 2.5 и 2.6.



	Id	Username	EntryType	EntryTime
1	1	1	Вход	2024-03-12 14:29:51
2	2	1	Выход	2024-03-12 14:29:52
3	3	1	Вход	2024-03-12 14:36:56
4	4	1	Выход	2024-03-12 14:36:57
5	12	123	Вход	2024-03-12 16:04:25
6	13	123	Выход	2024-03-12 16:10:05

Рисунок 2.5 – Результат таблицы «Entries»



Id	Username	Password	IsAdmin
1	test	test	1
2	1	1	0
3	123	123	0
4	4	4	1

Рисунок 2.6 – Результат таблицы «Users»

2.5 Программное обеспечение задачи

Для приложения учета рабочего времени сотрудников выделены следующие функции управления и обработки данных:

1) Служебные функции:

- Проверка пароля пользователя.
- Ведение календаря рабочего времени.
- Архивация баз данных.

2) Основные функции управления и обработки данных:

- Ввод первичной информации о входе и выходе сотрудников.
- Обработка и анализ введенных данных.
- Ведение справочников пользователей.
- Формирование ответов на запросы пользователей.

3) Структура сценария диалога позволяет определить следующие составляющие:

- Работа с экранными формами:
- Ввод данных о рабочем времени.
- Корректировка введенных данных.
- Просмотр введенной информации.

4) Формирование выходных документов:

- Формирование отчетов о рабочем времени сотрудников.

5) Работа с таблицами нормативно-справочной информации:

- Обновление данных о пользователях и их правах.

– Просмотр списка пользователей и их активности.

б) Протоколирование действий пользователя:

– Запись всех событий входа и выхода пользователей.

Диалог в приложении учета рабочего времени сотрудников формализуется в структурной форме, так как это позволяет эффективно управлять и обрабатывать операции в приложении.

2.6 Описание программных модулей

Алгоритм окна авторизации работает с проверками логина, пароля и статуса пользователя. В конечном итоге, администратор переключается на инструменты, то есть окно администратора, а сотрудник на окно сотрудника.

На рисунке 2.7 представлен алгоритм авторизации.

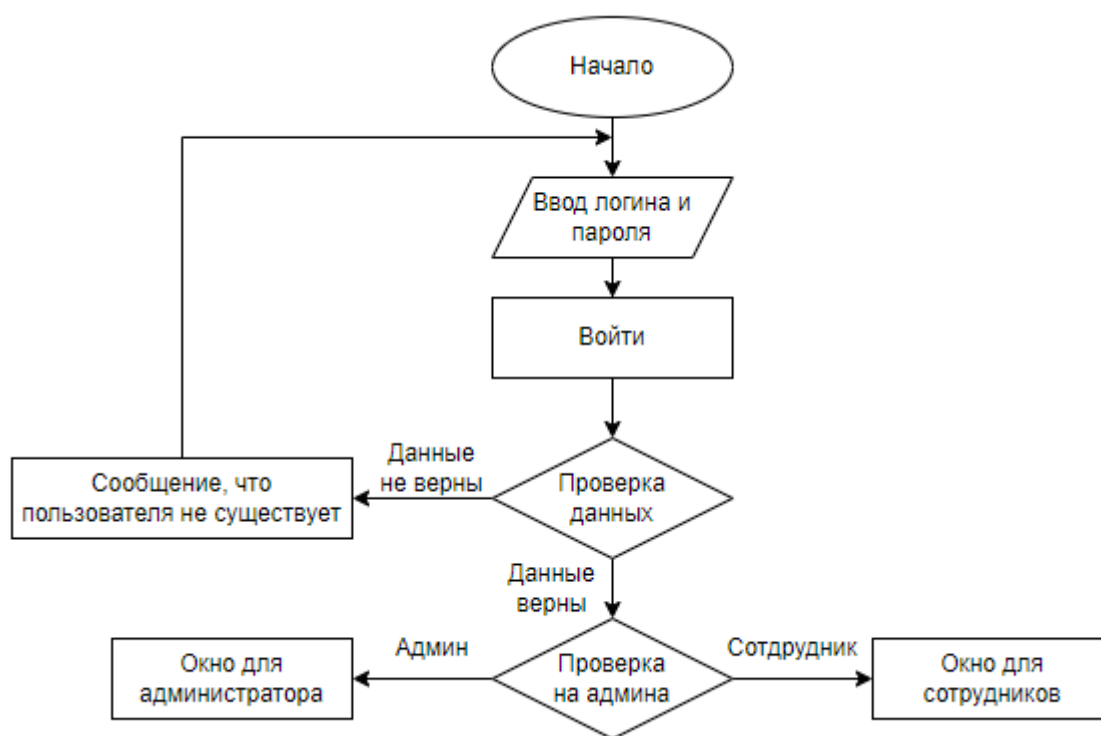


Рисунок 2.7 – Алгоритм авторизации

Если успешно авторизовались за сотрудника, присутствует алгоритм записи и даты и времени начала и конца рабочего дня. Данный алгоритм представлен на рисунке 2.8.



Рисунок 2.8 – Алгоритм сотрудника

Если успешно авторизовались за администратора, можно выполнять действия регистрации, удаления и просмотра учета времени сотрудников. Данный алгоритм представлен на рисунке 2.9.

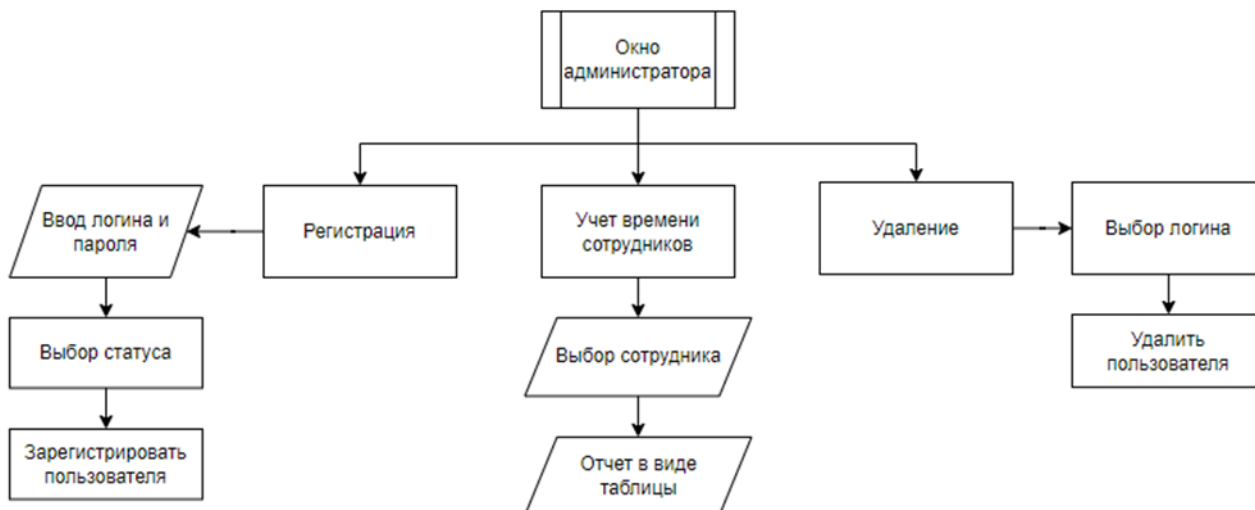


Рисунок 2.9 – Алгоритм администратора

Тема приложения была разработана в стиле Material Skin, также был использован внутренний пакет Visual Studio NuGet для установки пакета «System.Data.SQLite» для стабильной работы базы данных. На рисунке 2.10 показаны формы и среда разработки.

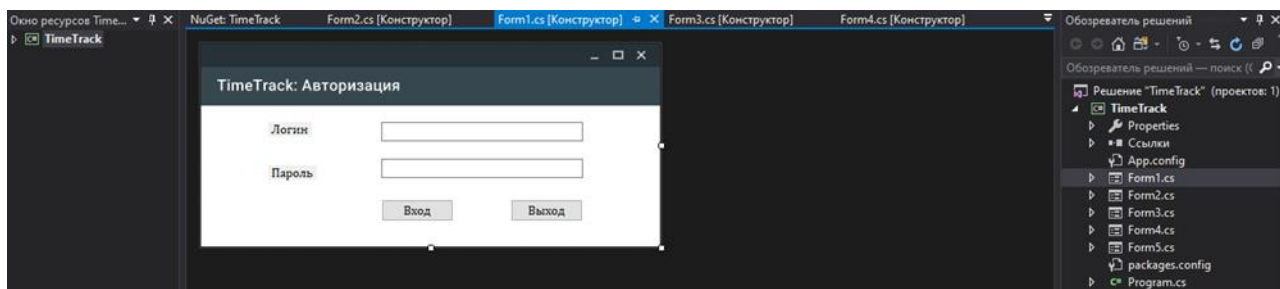


Рисунок 2.10 – Среда разработки и готовые формы

2.7 Руководство пользователя

При входе в приложение встречается окно авторизации, которая представлена на рисунке 2.11. Пользователь должен ввести логин и пароль, далее нажать «Вход», в противном случае «Выход».

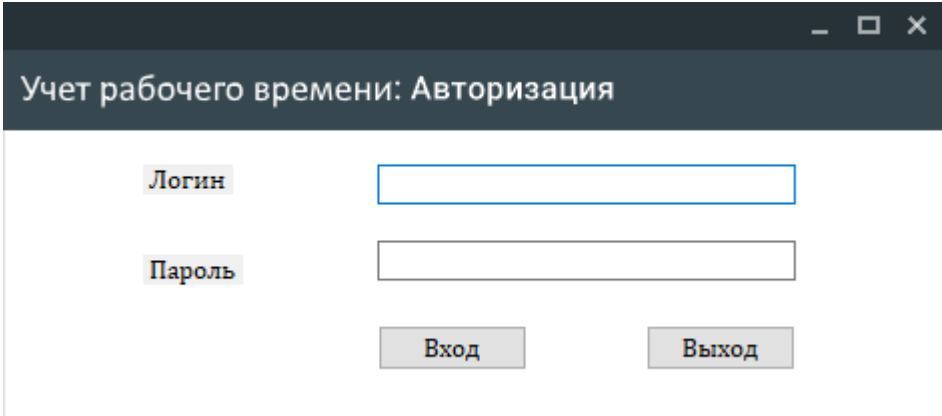
The image shows a software window titled "Учет рабочего времени: Авторизация". It features a dark header bar with the title in white text. Below the header, there are two input fields: the first is labeled "Логин" and the second is labeled "Пароль". At the bottom of the window, there are two buttons: "Вход" on the left and "Выход" on the right. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Рисунок 2.11 – Окно авторизации

В случае, если пользователь вводит неправильные данные, открывается всплывающее окно с предупреждением, окно представлено на рисунке 2.12.

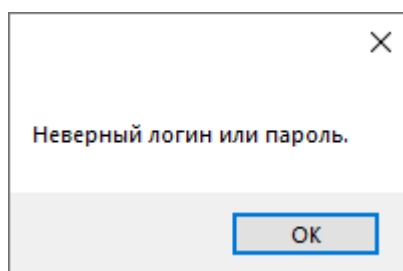


Рисунок 2.12 – Предупреждение

Пользователь заполнил данные верно, если у пользователя статус «Сотрудник» открывается соответствующее окно, где сотрудник в начале дня должен нажать «Вход» и в конце дня «Выход». Сверху указывается логин пользователя. Данное окно представлено на рисунке 2.13. Логин пользователя передается из формы авторизации в формы сотрудника и администратора через программный код. При нажатии на «Вход» в окне ниже записывается дата и время в полном формате и сразу же сохраняется в базу данных, далее идет отсчет, пока пользователь не нажмет кнопку «Выход», чтобы зафиксировать конец рабочего дня.

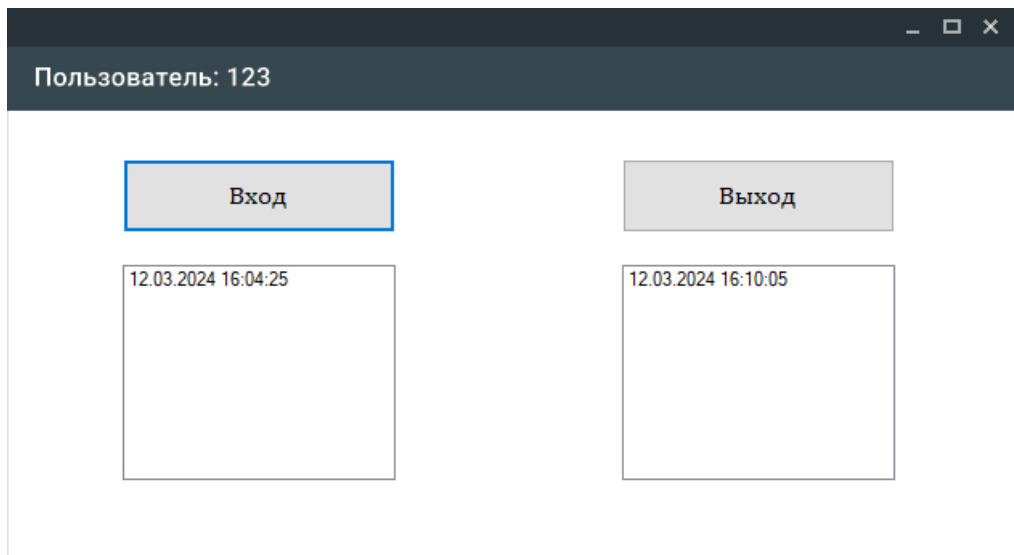


Рисунок 2.13 – Окно сотрудника

В следующем окне администратора, который представлен на рисунке 2.14, пользователь может выполнять действия «Регистрация», «Удаление» и «Фильтрация» для просмотра отчета по времени работы сотрудников.

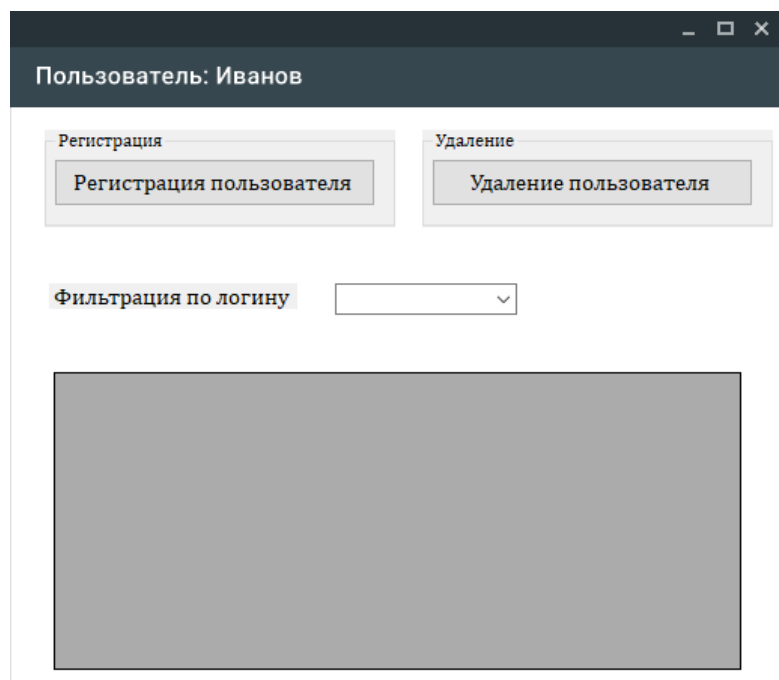


Рисунок 2.14 – Окно администратора

После выбора логина для фильтрации и просмотра в таблице появляется данные о сотруднике, его логин, время входа, время выхода и сколько времени сотрудник провел на работе. Данное действие представлено на рисунке 2.15.

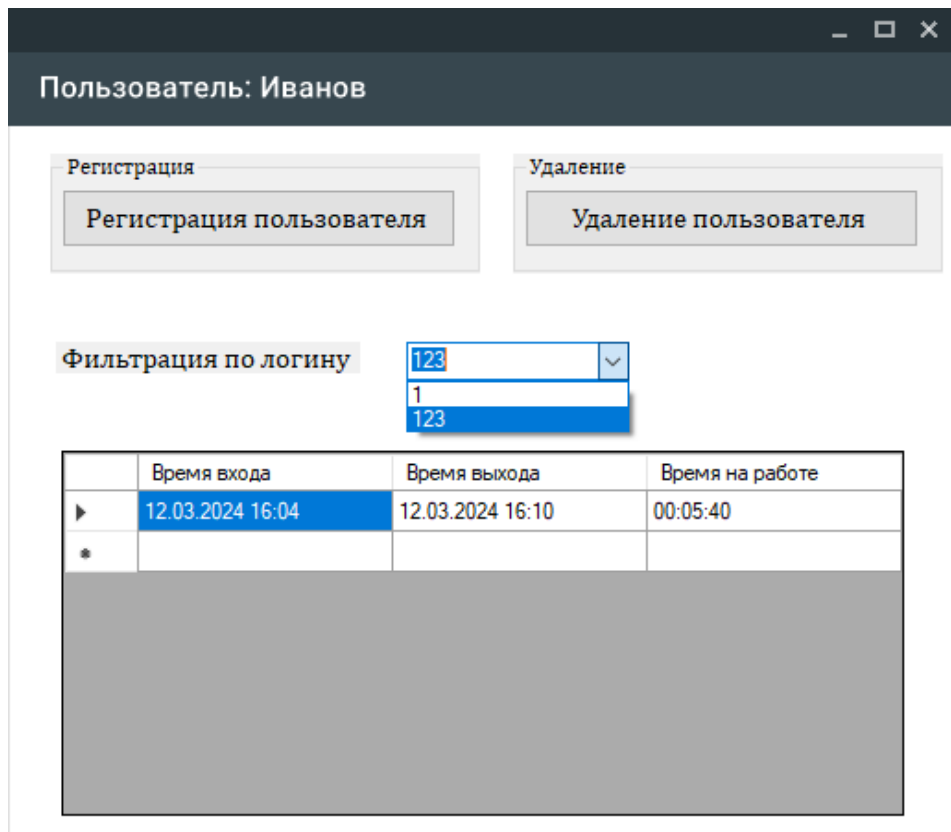


Рисунок 2.15 – Фильтрация и просмотра отчета сотрудников

Регистрация представлена на рисунке 2.16. Администратор должен ввести логин нового пользователя, также пароль и выбрать его статус для дальнейшей работы. Введенный запрос выполняется и сохраняется в базу данных.

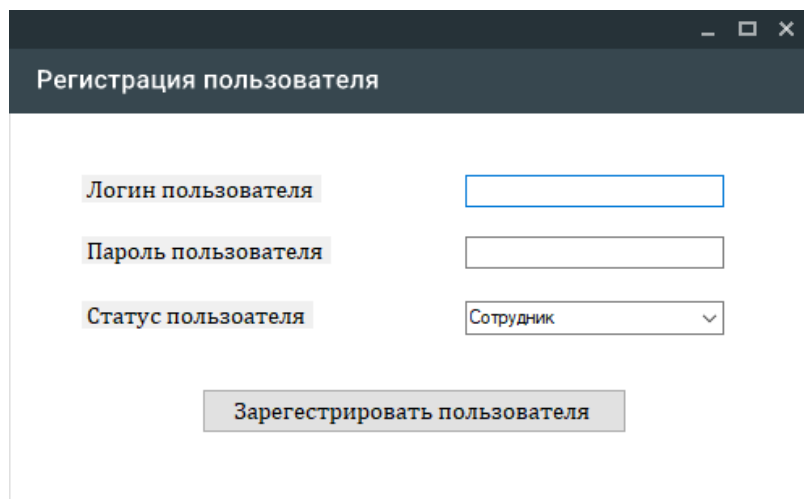


Рисунок 2.16 – Окно регистрации сотрудника

Удаление представлено на рисунке 2.17. Администратор должен выбрать в всплывающем меню логин пользователя, чтобы удалить его из базы данных.

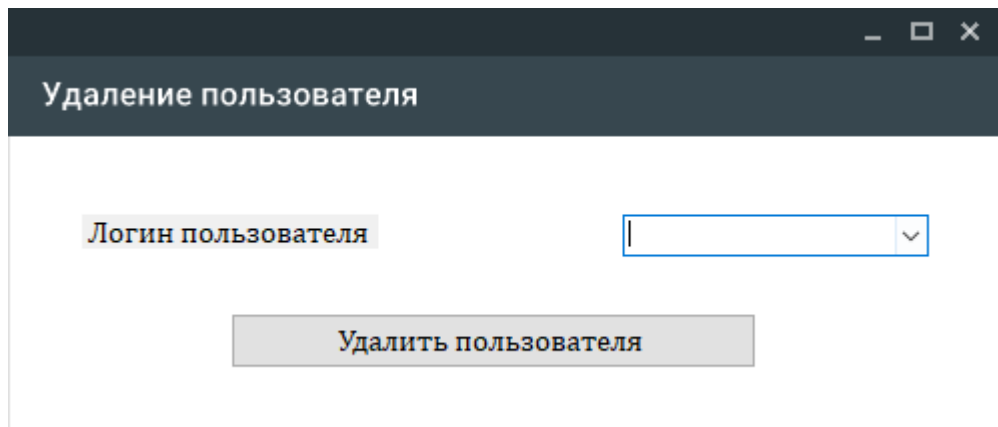


Рисунок 2.17 – Окно удаления сотрудника

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было создано приложение по учету рабочего времени сотрудников предприятия филиала федерального бюджетного учреждения здравоохранения «Центр гигиены и эпидемиологии в Липецкой области».

Создание данного приложения является актуальной задачей в условиях современного бизнеса, где эффективное использование трудовых ресурсов играет ключевую роль в успешной деятельности организации.

В процессе работы над проектом были изучены основные принципы учета рабочего времени, а также существующие подходы к автоматизации данного процесса. Разработанное приложение представляет собой удобный инструмент, позволяющий как сотрудникам, так и руководителям эффективно контролировать и учитывать отработанное время.

Основное внимание было уделено функциональности приложения, его удобству использования с уже существующими системами учета в организации. В результате исследования было показано, что использование разработанного приложения значительно упрощает процесс учета рабочего времени, сокращает временные затраты на административные процедуры и повышает общую эффективность управления персоналом.

В ходе разработки были решены следующие задачи:

- был выполнен анализ предметная область;
- изучены аналогичные решения, что помогло выявить их преимущества и недостатки, а также определить ключевые требования к разрабатываемому приложению.
- спроектирована база данных для хранения информации о пользователях, их входах и выходах, что обеспечило надежное хранение и быстрый доступ к данным.
- разработан функционал для сотрудников, позволяющий им удобно записывать свое время входа и выхода, а также для администраторов,

предоставляющий возможность управлять пользователями и просматривать статистику рабочего времени.

– организована технология сбора, передачи, обработки и выдачи информации, обеспечивающая эффективный поток данных от сбора до предоставления результатов.

Выводы работы подтверждают значимость внедрения современных технологий в управлении трудовыми ресурсами. Разработанное приложение может стать незаменимым инструментом для организации любого масштаба, стремящихся к оптимизации бизнес-процессов и повышению производительности труда. Дальнейшее развитие и совершенствование данного приложения открывает перед предприятиями новые перспективы в организации трудового процесса и повышении конкурентоспособности на рынке труда.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Васюткина, И. А. Разработка клиент-серверных приложений на языке C#: учебное пособие / И. А. Васюткина. – Новосибирск: НГТУ, 2020. – 112 с. – ISBN 978-5-7782-2932-7. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/118210> (дата обращения: 15.03.2024). – Режим доступа: для авториз. пользователей.

2. Гушин, А. Н. Применение библиотеки SDL для разработки программ на языке C: учебное пособие / А. Н. Гушин. – Санкт-Петербург : БГТУ "Военмех" им. Д.Ф. Устинова, 2019. – 118 с. – ISBN 978-5-85546-833-5. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/63684> (дата обращения: 15.03.2024). – Режим доступа: для авториз. пользователей.

3. Кулинча, П. В. Безопасность в C#: методы обеспечения безопасности при разработке приложений на C# / П. В. Кулинча // Дневник науки. – 2023. – № 6(78). – EDN SBGDIV.

4. Казанский, А. А. Программирование на Visual C# 2013: Учебное пособие / А. А. Казанский. – 1-е изд.. – Москва: Издательство Юрайт, 2020. – 191 с.

5. Анализ данных: Учебник / В. С. Мхитарян, М. Ю. Архипова, Т. А. Дуброва [и др.]. – 1-е изд.. – Москва: Издательство Юрайт, 2020. – 490 с. – (Бакалавр. Академический курс).

6. Татарникова, Т. М. Анализ данных: учебное пособие / Т. М. Татарникова. – Санкт-Петербург: Санкт-Петербургский государственный экономический университет, 2020. – 82 с.

7. Майо, Д. Самоучитель Microsoft Visual Studio 2010 / Д. Майо ; Джо Майо ; [пер. с англ. и ред. О. Кокоревой]. – Санкт-Петербург: БХВ-Петербург, 2021. – 445 с.

8. Копырин, А. С. Программирование на C# в Visual Studio 2013: учебное пособие / А. С. Копырин, Т. Л. Салова. – Сочи: СГУ, 2019. – 54 с. // Лань:

электронно-библиотечная система. – URL: <https://e.lanbook.com/book/147662> (дата обращения: 20.05.2024).

9. Параллельное программирование на языке C#: учебно-методическое пособие / составитель Р. Х. Ахмадулин. – Тюмень: ТюмГНГУ, 2020. – 37 с. // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/88569> (дата обращения: 20.05.2024).

10. Курбанисмаилов, З. М. Основы языка программирования C#: учебно-методическое пособие / З. М. Курбанисмаилов, Е. В. Кашкин. – Москва: РТУ МИРЭА, 2019. – 93 с (дата обращения: 20.05.2024).

11. Заборовский, Г. А. Программирование на языке C#: учебно-методическое пособие / Г. А. Заборовский, В. В. Сидорик. – Минск: БНТУ, 2020. – 84 с.

12. Климов, А. C#: Советы программистам / А. Климов. – Санкт-Петербург : БХВ-Петербург, 2022. – 544 с.

13. Мальцев, В. Л. Практическое применение опубликованного nuget-пакета Maltsev.requestredirector при разработке ASP.NET Core веб-приложения / В. Л. Мальцев // Наука. Наследие. Университет : Сборник материалов Международной 56-й научной студенческой конференции. Посвящается Году культурного наследия народов Российской Федерации, Году выдающихся земляков в Чувашской Республике, Чебоксары, 08–15 апреля 2022 года. – Чебоксары: Чувашский государственный университет имени И.Н. Ульянова, 2022. – С. 273-276.

14. Щукова, К. Б. Разработка парсера для извлечения и анализа данных в области судопроизводства / К. Б. Щукова // Современная техника и технологии. – 2020. – № 11(51). – С. 90-92 (дата обращения: 20.05.2024).

15. Мурлин, А. Г. Программирование с использованием компонентов Windows Forms : учебное пособие / А. Г. Мурлин, В. А. Мурлина, М. В. Янаева. – Краснодар : Общество с ограниченной ответственностью "Издательский Дом - Юг", 2022. – 126 с.

16. Петцольд, Ч. Программирование с использованием Microsoft Windows Forms: мастер-класс / Ч. Петцольд ; Чарльз Петцольд ; [пер. с англ. под общ. ред. А. Р. Врублевского]. – Москва [и др.] : Рус. ред., 2020. – 410 с.

17. Антонова, В. А. Создание приложения Windows - Forms для работы с элементами дискретной математики / В. А. Антонова, К. А. Балакин // Colloquium-Journal. – 2019. – № 15-1(39). – С. 7-9.

18. Alekseev, K. Relational database problems / K. Alekseev // Кибернетика и программирование. – 2020. – No. 2. – P. 7-18.

19. Ильясов, А. База данных и требования к базе данных / А. Ильясов, Г. Чарыева // Ceteris Paribus. – 2022. – № 12. – С. 28-30.

20. Ackermans, H. Electronic literature in the database and the database in electronic literature / H. Ackermans, I. Van De Ven // Communications. Media. Design. – 2019. – Vol. 4, No. 4. – P. 5-18.

21. Винник, О. Г. Использование метода «дерево целей» для визуализации целей учреждения высшего образования / О. Г. Винник // Экономика. Бизнес. Финансы. – 2023. – № 4. – С. 11-15 (дата обращения: 20.06.2023).

22. Тычкина, Ю. И. Использование методики "Дерево проблем" для определения проблем, влияющих на мотивацию обучения студентов технического вуза / Ю. И. Тычкина, М. А. Щекалева, Г. М. Гринберг // Актуальные проблемы авиации и космонавтики. – 2022. – Т. 2, № 8. – С. 489-491 (дата обращения: 20.05.2024).

23. Бурькин, А. Д. Организационная структура управления предприятием и факторы, влияющие на ее эффективность / А. Д. Бурькин, В. А. Кваша // Экономика и управление: проблемы, решения. – 2019. – Т. 2, № 7. – С. 55-64. – <https://www.elibrary.ru/item.asp?id=35384866> (дата обращения: 20.05.2024).

24. Ткачук, Е. О. Применение СУБД sqlite в образовательном процессе современного вуза / Е. О. Ткачук // НАУЧНЫЕ ИССЛЕДОВАНИЯ ВЫСШЕЙ ШКОЛЫ: сборник статей IV Международной научно-практической

конференции, Пенза, 10 сентября 2020 года. – Пенза: "Наука и Просвещение" (ИП Гуляев Г.Ю.), 2020. – С. 12-15. – EDN ENTVKP.

25. Гусев, И. В. Использование базы данных sqlite в корпоративных информационных системах / И. В. Гусев, В. В. Гусев // Научное сообщество студентов. Междисциплинарные исследования : Электронный сборник статей по материалам XXVII студенческой международной научно-практической конференции. Том 16 (27) : Ассоциация научных сотрудников "Сибирская академическая книга", 2020. – С. 7-10. – EDN ZEZYGZ.

26. Абдулкадырова, Н. А. Организация табельного учета и ведения табеля учета рабочего времени / Н. А. Абдулкадырова, С. А. Шалиева, З. М. Батырмурзаева // Теория и практика современной науки. – 2020. – № 5(59). – С. 70-74. – EDN FRRKGX.

27. Попов, К. С. Система учета рабочего времени сотрудников / К. С. Попов, С. А. Ляшева // Информационные технологии на службе общества : материалы Международной научно-практической конференции, Нижнекамск, 18 апреля 2020 года. – Нижнекамск: Издательство Казанского государственного технического университета, 2020. – С. 33-34. – EDN TDHZEV.

Листинг программы

Форма авторизации:

```
using System;
using System.Data;
using System.Data.SQLite;
using System.Windows.Forms;
namespace TimeTrack
{
    public partial class Form1 : MaterialSkin.Controls.MaterialForm
    {
        private SQLiteConnection connection;
        public Form1()
        {
            InitializeComponent();
            InitializeDatabase();
        }

        private void InitializeDatabase()
        {
            connection = new SQLiteConnection("Data Source=users.db;Version=3;");
            connection.Open();

            string createUserTableQuery = "CREATE TABLE IF NOT EXISTS Users (Id INTEGER
PRIMARY KEY AUTOINCREMENT, Username TEXT UNIQUE, Password TEXT, IsAdmin
INTEGER)";

            ExecuteQuery(createUserTableQuery);
        }

        private void ExecuteQuery(string query)
        {
            using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
            {
```

```

        cmd.ExecuteNonQuery();
    }
}

private bool AuthenticateUser(string username, string password)
{
    string query = $"SELECT * FROM Users WHERE Username='{username}' AND Password='{password}'";
    using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
    {
        using (SQLiteDataReader reader = cmd.ExecuteReader())
        {
            return reader.Read();
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    string username = textBox1.Text;
    string password = textBox2.Text;
    if (AuthenticateUser(username, password))
    {
        bool isAdmin = IsAdminUser(username); //проверяем является ли пользователь администратором
        if (isAdmin)
        {
            Form2 form2 = new Form2(username);
            form2.Show();
        }
        else
        {

```

```

        Form3 form3 = new Form3(username);
        form3.Show();
    }
}
else
{
    MessageBox.Show("Неверный логин или пароль.");
}
}

```

```

private bool IsAdminUser(string username)
{
    string query = $"SELECT IsAdmin FROM Users WHERE Username='{username}'";
    using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
    {
        object result = cmd.ExecuteScalar();
        if (result != null && result != DBNull.Value)
        {
            return Convert.ToBoolean(result);
        }
        return false; //по умолчанию считаем, что пользователь не администратор
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    this.Close();
}

```

```

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    connection.Close();
}

```

```

    }
    private void textBox1_TextChanged(object sender, EventArgs e)
    {
    }
    private void textBox2_TextChanged(object sender, EventArgs e)
    {
    }
}
}

```

Форма сотрудника:

```

using System;
using System.Data;
using System.Data.SQLite;
using System.Windows.Forms;
namespace TimeTrack
{
    public partial class Form3 : MaterialSkin.Controls.MaterialForm
    {
        private SQLiteConnection connection;
        private string username;
        public Form3(string username)
        {
            InitializeComponent();
            this.username = username;
            this.Text = "Пользователь: " + username;
            InitializeDatabase();
            PopulateListBoxes();
        }

        private void InitializeDatabase()
        {

```



```

connection = new SQLiteConnection("Data Source=users.db;Version=3;");
connection.Open();

string createEntryTableQuery = "CREATE TABLE IF NOT EXISTS Entries (Id
INTEGER PRIMARY KEY AUTOINCREMENT, Username TEXT, EntryType TEXT,
EntryTime DATETIME)";

ExecuteQuery(createEntryTableQuery);
}

private void ExecuteQuery(string query)
{
using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
{
cmd.ExecuteNonQuery();
}
}

private void PopulateListBoxes()
{
listBox1.Items.Clear();
listBox2.Items.Clear();

string selectQuery = $"SELECT EntryType, EntryTime FROM Entries WHERE
Username = '{username}'";

using (SQLiteCommand cmd = new SQLiteCommand(selectQuery, connection))
{
using (SQLiteDataReader reader = cmd.ExecuteReader())
{
while (reader.Read())
{
string entryType = reader["EntryType"].ToString();
string entryTime = reader["EntryTime"].ToString();
if (entryType == "Вход")
{
listBox1.Items.Add(entryTime);
}
}
}
}
}

```

```

    }
    else if (entryType == "Выход")
    {
        listBox2.Items.Add(entryTime);
    }
}
}
}
}

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    RecordEntry("Вход");
}

private void button2_Click(object sender, EventArgs e)
{
    RecordEntry("Выход");
}

private void RecordEntry(string entryType)
{
    string entryTime = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");

```

```

        string query = $"INSERT INTO Entries (Username, EntryType, EntryTime) VALUES
        ({username}', '{entryType}', '{entryTime}')";

        ExecuteQuery(query);

        PopulateListBoxes(); //обновляем списки после добавления новой записи
    }

    private void Form3_FormClosing(object sender, FormClosingEventArgs e)
    {
        connection.Close();
    }

}
}

```

Форма администратора:

```

using System;
using System.Data;
using System.Data.SQLite;
using System.Windows.Forms;
namespace TimeTrack
{
    public partial class Form2 : MaterialSkin.Controls.MaterialForm
    {
        private string username;

        public Form2(string username)
        {
            InitializeComponent();
            this.username = username;
            this.Text = "Пользователь: " + username;
            //заполнение ComboBox при загрузке формы
            PopulateComboBox();
        }
    }
}

```

```

//заполнение DataGridView1 при загрузке формы
PopulateDataGridView();
}

private void PopulateComboBox()
{
    comboBox1.Items.Clear();

    using (SQLiteConnection connection = new SQLiteConnection("Data
Source=users.db;Version=3;"))
    {
        connection.Open();

        string query = "SELECT Username FROM Users WHERE IsAdmin = 0";
        using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
        {
            using (SQLiteDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    string username = reader["Username"].ToString();
                    comboBox1.Items.Add(username);
                }
            }
        }
    }
}

private void PopulateDataGridView()
{
    using (SQLiteConnection connection = new SQLiteConnection("Data
Source=users.db;Version=3;"))

```

```

{
    connection.Open();
    // проверяем был ли выбран пользователь из ComboBox
    if (!string.IsNullOrEmpty(comboBox1.Text))
    {
        string selectedUser = comboBox1.Text;

        //выполнение запроса для получения данных из таблицы Entries для
        выбранного пользователя

        string query = $"SELECT EntryTime AS 'Время входа', " +
            $"(SELECT EntryTime FROM Entries e2 WHERE e2.Username =
            Entries.Username AND e2.EntryTime > Entries.EntryTime AND e2.EntryType = 'Выход'
            ORDER BY e2.EntryTime LIMIT 1) AS 'Время выхода' " +
            $"FROM Entries WHERE EntryType = 'Вход' AND Username =
            '{selectedUser}'";

        SQLiteDataAdapter dataAdapter = new SQLiteDataAdapter(query, connection);
        DataTable dataTable = new DataTable();
        dataAdapter.Fill(dataTable);

        //столбец Время на работе

        DataColumn workTimeColumn = new DataColumn("Время на работе",
        typeof(string));
        dataTable.Columns.Add(workTimeColumn);

        //вычисляем и заполняем значения в столбце Время на работе
        foreach (DataRow row in dataTable.Rows)
        {
            DateTime entryTime = Convert.ToDateTime(row["Время входа"]);
            DateTime exitTime = Convert.ToDateTime(row["Время выхода"]);
            TimeSpan workTime = CalculateWorkTime(entryTime, exitTime);
            row["Время на работе"] = workTime.ToString();
        }

        //привязка данных к DataGridView1
        dataGridView1.DataSource = dataTable;
    }
}

```

```

        }
        connection.Close();
    }
}

private TimeSpan CalculateWorkTime(DateTime entryTime, DateTime exitTime)
{
    return exitTime - entryTime;
}

private void button1_Click(object sender, EventArgs e)
{
    Form4 form4 = new Form4();
    form4.Show();
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs
e)
{
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //при изменении выбранного пользователя обновляем DataGridView
    PopulateDataGridView();
}

private void button2_Click(object sender, EventArgs e)
{
    Form5 form5 = new Form5();
    form5.Show();
}
}

```

```
}
```

Форма регистрации пользователя:

```
using System;
using System.Data.SQLite;
using System.Windows.Forms;
namespace TimeTrack
{
    public partial class Form4 : MaterialSkin.Controls.MaterialForm
    {
        private SQLiteConnection connection;
        public Form4()
        {
            InitializeComponent();
            InitializeDatabase();
            InitializeComboBox();
        }

        private void InitializeDatabase()
        {
            connection = new SQLiteConnection("Data Source=users.db;Version=3;");
            connection.Open();

            string createUserTableQuery = "CREATE TABLE IF NOT EXISTS Users (Id INTEGER
PRIMARY KEY AUTOINCREMENT, Username TEXT UNIQUE, Password TEXT, IsAdmin
INTEGER)";

            ExecuteQuery(createUserTableQuery);
        }

        private void ExecuteQuery(string query)
        {
            using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
            {
```

```

        cmd.ExecuteNonQuery();
    }
}

private void InitializeComboBox()
{
    comboBox1.Items.Add("Сотрудник");
    comboBox1.Items.Add("Администратор");
    comboBox1.SelectedIndex = 0; //по умолчанию выбран статус "Сотрудник"
}

private void button1_Click(object sender, EventArgs e)
{
    {
        string username = textBox1.Text;
        string password = textBox2.Text;
        int isAdmin = comboBox1.SelectedIndex == 1 ? 1 : 0;
        if (!string.IsNullOrEmpty(username) && !string.IsNullOrEmpty(password))
        {
            string insertUserQuery = $"INSERT INTO Users (Username, Password, IsAdmin)
VALUES ('{username}', '{password}', '{isAdmin}')";
            ExecuteQuery(insertUserQuery);
            MessageBox.Show("Пользователь успешно зарегистрирован.");
        }
        else
        {
            MessageBox.Show("Введите логин и пароль.");
        }
    }
}

private void Form4_FormClosing(object sender, FormClosingEventArgs e)
{

```



```

        connection.Close();
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
}
}

```

Форма удаления сотрудника:

```

using System;
using System.Data.SQLite;
using System.Windows.Forms;
namespace TimeTrack
{
    public partial class Form5 : MaterialSkin.Controls.MaterialForm
    {
        public Form5()
        {
            InitializeComponent();
            PopulateComboBox();
        }

        private void PopulateComboBox()

```

```

    {
        comboBox1.Items.Clear();

        using (SQLiteConnection connection = new SQLiteConnection("Data
Source=users.db;Version=3;"))
        {
            connection.Open();
            string query = "SELECT Username FROM Users";
            using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
            {
                using (SQLiteDataReader reader = cmd.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        string username = reader["Username"].ToString();
                        comboBox1.Items.Add(username);
                    }
                }
            }
            connection.Close();
        }
    }

private void button1_Click(object sender, EventArgs e)
{
    string selectedUsername = comboBox1.SelectedItem.ToString();

    if (!string.IsNullOrEmpty(selectedUsername))
    {
        if (MessageBox.Show($"Вы уверены, что хотите удалить пользователя
{selectedUsername}?", "Удаление пользователя", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
        {
            DeleteUser(selectedUsername);

            MessageBox.Show($"Пользователь {selectedUsername} успешно удален.");
        }
    }
}

```

```

        PopulateComboBox(); //обновляем ComboBox после удаления пользователя
    }
}
else
{
    MessageBox.Show("Выберите пользователя для удаления.");
}
}

private void DeleteUser(string username)
{
    using (SQLiteConnection connection = new SQLiteConnection("Data
Source=users.db;Version=3;"))
    {
        connection.Open();
        string query = $"DELETE FROM Users WHERE Username = '{username}'";
        using (SQLiteCommand cmd = new SQLiteCommand(query, connection))
        {
            cmd.ExecuteNonQuery();
        }
        connection.Close();
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}
}
}

```