Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана» (национальный исследовательский университет)

Московский техникум космического приборостроения

УТВЕРЖДАЮ

Заместитель директора по УР

Н.Н. Ковзель

(подпись, дата)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗРАБОТКА СИСТЕМЫ УДАЛЕННОГО УПРАВЛЕНИЯ ФАЙЛАМИ И ОПЕРАЦИОННЫМИ СИСТЕМАМИ**

Пояснительная записка

Группа ТИП-83 Листов

Председатель предметной

Митрошенкова Е.А.

(цикловой) комиссии

(подпись, дата) (ФИО)

Руководитель разработки

от техникума

(подпись, дата) (ФИО)

Руководитель разработки

от предприятия

(подпись, дата) (ФИО)

Консультант по Ковалев С.А.

экономической части

(подпись, дата) (ФИО)

Студент

(подпись, дата)

(ФИО)

Москва 2023

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана» (национальный исследовательский университет)

Московский техникум космического приборостроения

Срок окончания работы « » июня 2023 г.

Зав. отделением О.В. Коротченко Дата « » 20 г.

(подпись)

ЗАДАНИЕ

на выполнение дипломного проекта Студенту

(фамилия, имя, отчество)

Тема работы Разработка системы удаленного управления фалйами и операционными системами

Введение

1. Теоретическое обоснование разрабатываемого программного продукта
   1. Описание предметной области
   2. Сравнительный анализ программ-аналогов
   3. Моделирование проектируемой системы
   4. Разработка функциональных требований к программе
   5. Обоснование выбора средств реализации программной системы
   6. Вывод по разделу
2. Разработка программного продукта
   1. Разработка архитектура программной системы
   2. Разработка структуры данных
   3. Конструирование пользовательского интерфейса
   4. Схемы алгоритма программы и подпрограмм
   5. Отладка и тестирование программы
   6. Руководство пользователя
   7. Вывод по разделу 3 Охрана труда

4 Экономическая часть Заключение

Список использованных источников Приложения

Презентация

Дата выдачи задания « » 20 г.

Руководитель дипломного проекта от техникума

(подпись, дата) (ФИО)

Руководитель работы от предприятия Шулькин А. В.

(подпись, дата) (ФИО)

Срок сдачи обучающимся готовой работы: г. Задание принял к исполнению:

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc134621467)

[1 Теоретическое обоснование разрабатываемого программного продукта 7](#_Toc134621468)

[1.1 Описание предметной области 7](#_Toc134621469)

[1.2 Сравнительный анализ программ-аналогов 9](#_Toc134621470)

[1.3 Моделирование проектируемой системы 12](#_Toc134621471)

[1.4 Разработка функциональных требований программной системе 14](#_Toc134621472)

[1.4.1 Назначение программы 14](#_Toc134621473)

[1.4.2 Требования к функциональным характеристикам 14](#_Toc134621474)

[1.4.3 Требования к надежности 15](#_Toc134621475)

[1.4.4 Требования к составу и параметрам технических средств 15](#_Toc134621476)

[1.4.5 Требования к информационной и программной совместимости 15](#_Toc134621477)

[1.4.6 Требования к программной документации 16](#_Toc134621478)

[1.4.7 Обоснование выбора средств в реализации программной системы 16](#_Toc134621479)

[1.5 Вывод по разделу 18](#_Toc134621480)

[2 Разработка программного продукта 19](#_Toc134621481)

[2.1 Разработка архитектуры программной системы 19](#_Toc134621482)

[2.2 Разработка структуры данных 21](#_Toc134621483)

[2.3 Конструирование пользовательского интерфейса 23](#_Toc134621484)

[2.4 Схемы алгоритма программы и подпрограмм 26](#_Toc134621485)

[2.4.1 Схема алгоритма функции авторизации 26](#_Toc134621486)

[2.4.2 Схема алгоритма функции регистрации 27](#_Toc134621487)

[2.4.3 Схема алгоритма функции создания заявки 28](#_Toc134621488)

[2.5 Отладка и тестирование программы 29](#_Toc134621489)

[2.5.1 Отладка программы 29](#_Toc134621490)

[2.5.2 Тестирование программы 30](#_Toc134621491)

[2.5.2.1 Тестирование программы в нормальных условиях 31](#_Toc134621492)

[2.5.2.2 Тестирование программы в экстремальных условиях 34](#_Toc134621493)

[2.5.2.3 Тестирование программы в исключительных ситуациях 35](#_Toc134621494)

[2.5.2.4 UnitTest 36](#_Toc134621495)

[Листинг проекта UnitTest1 37](#_Toc134621496)

[2.5.2.5 Сообщения 38](#_Toc134621497)

[2.6 Руководство пользователя 39](#_Toc134621498)

[2.6.1 Назначение программы 39](#_Toc134621499)

[2.6.2 Условия работы программы 39](#_Toc134621500)

[2.6.3 Входные данные 40](#_Toc134621501)

[2.6.4 Выполнение программы 40](#_Toc134621502)

[2.6.5 Выходные данные 48](#_Toc134621503)

[2.7 Вывод по разделу 49](#_Toc134621504)

[3 Охрана труда 50](#_Toc134621505)

[3.1 Техника безопасности при работе на персональном компьютере 50](#_Toc134621506)

[3.2 Требования к помещению (машинного зала, ВЦ) 53](#_Toc134621507)

[3.3 Мероприятия по противопожарной охране 54](#_Toc134621508)

[4 Технико-экономическая часть 55](#_Toc134621509)

[4.1 Технико-экономические расчеты экономической целесообразности программы 55](#_Toc134621510)

[4.2 Расчетная часть 58](#_Toc134621511)

[4.2.1 Расчет трудоемкости разработанной программы 58](#_Toc134621512)

[4.2.2 Расчет себестоимости разработанной программы 61](#_Toc134621513)

[4.2.3 Анализ возможных путей снижения себестоимости 67](#_Toc134621514)

[4.3 Графическая часть 68](#_Toc134621515)

[ЗАКЛЮЧЕНИЕ 71](#_Toc134621516)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 72](#_Toc134621517)

[ПРИЛОЖЕНИЕ А 73](#_Toc134621518)

[Листинг программы 74](#_Toc134621519)

[Форма «Главное меню» 76](#_Toc134621520)

[ПРИЛОЖЕНИЕ Б 78](#_Toc134621521)

# ВВЕДЕНИЕ

Тема: Разработка системы удаленного управления и операционными системами. В современном мире удаленная работа и управление компьютерными системами становятся неотъемлемой частью работы многих компаний и организаций. Благодаря быстрому развитию технологий и широкому доступу к интернету, удаленный доступ к файлам и возможность управления операционными системами стали необходимостью для обеспечения эффективности и безопасности рабочих процессов. Это особенно актуально для компаний с географически распределенными командами, филиалами или сотрудниками, работающими в удаленном режиме.

Однако, при осуществлении удаленной работы возникают ряд проблем, таких как безопасность данных, удобство доступа и управления системами, а также совместная работа над проектами. Для решения этих проблем и обеспечения эффективного удаленного рабочего процесса необходимы специализированные инструменты и системы управления файлами и операционными системами.

Системы удаленного управления позволяют пользователям получать доступ к компьютерам и серверам из любой точки мира через интернет. Они предоставляют возможность удаленного управления и контроля над операционной системой, файлами, приложениями и другими ресурсами компьютерной системы. Такие системы могут быть основаны на различных протоколах и технологиях, таких как Remote Desktop Protocol (RDP), Virtual Network Computing (VNC) и Secure Shell (SSH), и обеспечивают шифрование и аутентификацию для защиты конфиденциальности данных.

Системы управления файлами позволяют эффективно организовывать и обмениваться файлами между удаленными пользователями. Они обеспечивают возможность загружать, скачивать, редактировать и совместно работать над файлами, а также устанавливать права доступа и контролировать версии файлов. Такие системы часто включают в себя функции облачного хранения, синхронизации и резервного копирования данных, обеспечивая надежность и доступность файлов в любое время и из любого места.

В целом, разработка систем удаленного управления и операционными системами является актуальной задачей, которая позволяет компаниям и организациям обеспечить безопасность, эффективность и гибкость в рамках удаленной работы. Такие системы способствуют повышению производительности и сокращению затрат, а также обеспечивают комфортные условия для сотрудников, позволяя им эффективно совмещать работу и личные обязанности.

С развитием технологий удаленного управления и операционными системами, компании и организации получают ряд преимуществ. Во-первых, удаленный доступ к файлам и операционным системам обеспечивает гибкость и мобильность. Сотрудники могут работать из любой точки мира, не привязываясь к конкретному рабочему месту или офису. Это особенно полезно для компаний с дистанционными или международными командами, а также для сотрудников, работающих в командировках или на удаленных объектах.

Во-вторых, удаленное управление и операционные системы позволяют быстро реагировать на проблемы и проводить техническую поддержку удаленных пользователей. Администраторы систем могут удаленно настраивать и обновлять программное обеспечение, решать проблемы сети и диагностировать ошибки без необходимости физического присутствия на месте.

Кроме того, системы удаленного управления и операционными системами способствуют повышению безопасности данных. Они обеспечивают шифрование и аутентификацию, защищая информацию от несанкционированного доступа и внешних угроз. Благодаря возможностям удаленного удаления данных в случае утери или кражи устройства, компании могут снизить риск утечки конфиденциальной информации.

Кроме того, удаленное управление операционными системами способствует более эффективному использованию ресурсов. Администраторы могут мониторить и оптимизировать производительность систем, распределять вычислительные мощности и управлять доступом к ресурсам в режиме реального времени. Это позволяет снизить издержки и повысить эффективность работы компьютерных систем.

Однако, несмотря на все преимущества удаленного управления и операционными системами, существуют и некоторые вызовы и риски. Важно обеспечить надежность и защиту системы от возможных угроз, таких как хакерские атаки или вирусы. Также следует учесть вопросы приватности и соблюдения законодательства о защите данных при удаленной работе и доступе к конфиденциальной информации.

В целом, разработка систем удаленного управления и операционными системами становится все более важной и актуальной. Компании и организации, осознавая преимущества таких систем, стремятся использовать современные инструменты и технологии для обеспечения безопасности, эффективности и гибкости рабочих процессов в условиях удаленной работы.

Целью дипломной работы является разработка системы удаленного управления файлами и операционными системами для обеспечения эффективной и безопасной удаленной работы.

Для создания такой системы нужно решить следующие задачи:

1. Произвести анализ предметной области;
2. Осуществить проектирование программного продукта с построением диаграмм потоков данных, функциональной диаграммы;
3. Осуществить проектирование БД;
4. Разработать интерфейс;
5. Определить экономическую эффективность жизненного цикла системы;
6. Разработать руководство по эксплуатации системы.

Объектом исследования данного проекта является компания ООО «СМАРТ СИСТЕМС».

Предметом исследования является процесс технической поддержки сотрудников в компании ООО «СМАРТ СИСТЕМС».

ООО «СМАРТ СИСТЕМС» - это IT компания, специализирующаяся на разработке и предоставлении инновационных решений в области информационных технологий. Компания занимается широким спектром деятельности, охватывающим различные аспекты IT сферы. Вот некоторые из ключевых направлений работы компании:

Разработка программного обеспечения: ООО «СМАРТ СИСТЕМС» разрабатывает пользовательские приложения, веб-приложения, мобильные приложения и другое программное обеспечение для различных платформ и операционных систем. Компания имеет опыт в создании индивидуальных решений под конкретные потребности клиентов.

Интеграция и внедрение систем: Компания предлагает услуги по интеграции различных информационных систем и платформ, а также их внедрению в организации. Это включает в себя настройку и адаптацию системы под требования клиента, обучение пользователей и поддержку в процессе внедрения.

Системная инфраструктура и сетевые решения: ООО «СМАРТ СИСТЕМС» специализируется на проектировании, развертывании и поддержке корпоративной сетевой инфраструктуры. Компания предлагает услуги по настройке серверов, сетевого оборудования, виртуализации, облачных решений, а также обеспечивает безопасность сети и защиту от угроз.

Консалтинг и IT-стратегия: Компания оказывает консультационные услуги по вопросам информационных технологий и разработке IT-стратегии для организаций. Это включает в себя анализ текущей IT инфраструктуры, определение потребностей и целей клиента, разработку плана действий и рекомендаций по оптимизации использования технологий.

Безопасность и защита данных: ООО «СМАРТ СИСТЕМС» предлагает решения и услуги по обеспечению безопасности информации и защите данных. Компания разрабатывает стратегии безопасности, реализует системы защиты от внешних угроз, проводит аудит безопасности, внедряет системы резервного копирования и восстановления данных.

Техническая поддержка и обслуживание: Компания предоставляет техническую поддержку и обслуживание IT инфраструктуры клиентов. Это включает в себя решение текущих технических проблем, обновление и мониторинг систем, а также предоставление консультаций и помощи пользователям.

ООО «СМАРТ СИСТЕМС» стремится предоставлять клиентам современные и инновационные решения, помогая им оптимизировать бизнес-процессы, повысить эффективность работы и обеспечить безопасность информации. Компания ориентирована на индивидуальный подход к каждому клиенту, адаптируя свои услуги под конкретные требования и потребности заказчика.

Разработка системы удаленного управления файлами и операционными системами позволит сотрудникам компании эффективно и безопасно работать удаленно. С помощью этой системы сотрудники смогут получить доступ к нужным файлам и приложениям с любого устройства и из любой точки мира, что позволит им быть более мобильными и продуктивными. Также система обеспечит высокий уровень безопасности и защиты данных, что важно для компаний, работающих с конфиденциальной информацией.

# Теоретическое обоснование разрабатываемого программного продукта

## Описание предметной области

Система удаленного управления файлами и операционными системами - область IT и компьютерных сетей, позволяющая удаленно управлять операционными система-ми и файлами на удаленных компьютерах. Разработка такой системы включает в себя проектирование архитектуры, разработку ПО, настройку безопасности, тестирование и оптимизацию производительности. Она используется для управления компьютерными парками, удаленного обучения, виртуализации рабочих мест и т.д.

В текущее время, если у сотрудников компании появляются проблемы технического плана, они приходят в отдел техподдержки лично.

Требуется разработать ИС которая предоставит функционал для удаленного подключения к операционной системе

Данная ИС будет представлен в качестве приложения.

Для более наглядного представления на рисунке 1.1 приведена диаграмма IDEF0, модель «AS-IS»

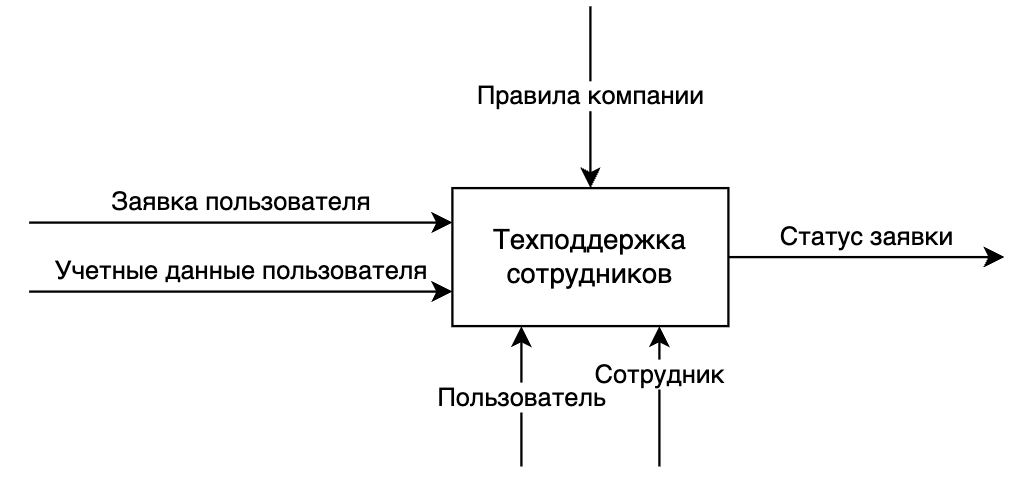


Рисунок 1.1 ‒ Диаграмма IDEF0, модель «AS-IS» для процесса «Техподдержка сотрудников»

Результат построения декомпозиции диаграммы IDEF0, модель «AS-IS» приведен на рисунке 1.2.

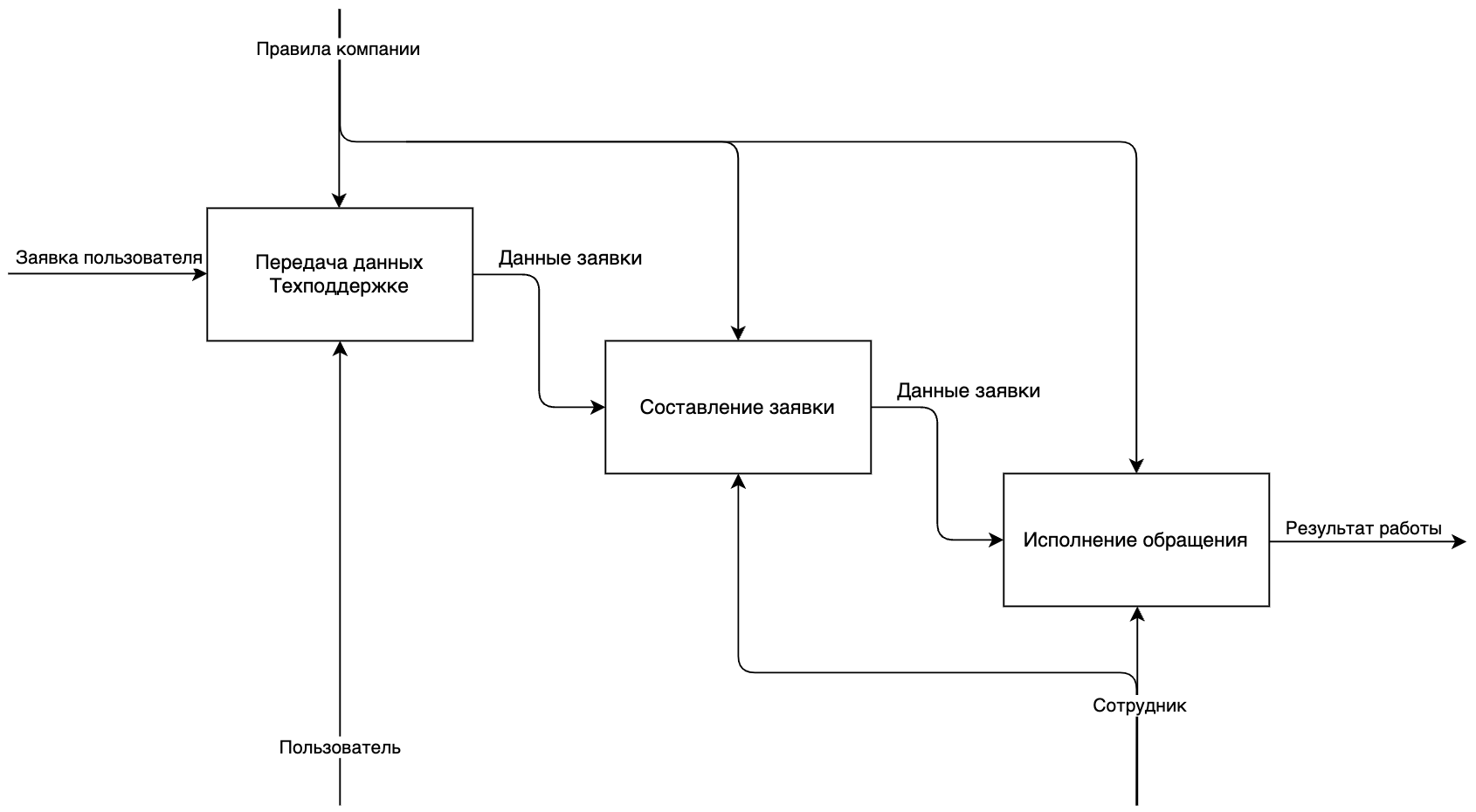


Рисунок 1.2 – Декомпозиция диаграммы IDEF0, модель «AS-IS»

## Сравнительный анализ программ-аналогов

Примерами программ-аналогов разрабатываемой системы являются следующие решения:

* «Termius».
* «Putty».

Ниже будет рассмотрена каждая из них.

1. «Termius» - эта программа является одной из наиболее распространенных систем для удаленного управления файлами и операционными системами.

Достоинства программы:

* позволяет быстро выполнять подключение и манипулирование файлами;
* красивый интерфейс;

Недостатки программы:

* имеет довольно высокую стоимость;
* требует наличия определенных навыков для ее настройки и администрирования;

Для более наглядного представления на рисунке 1.1 приведен интерфейс приложения:

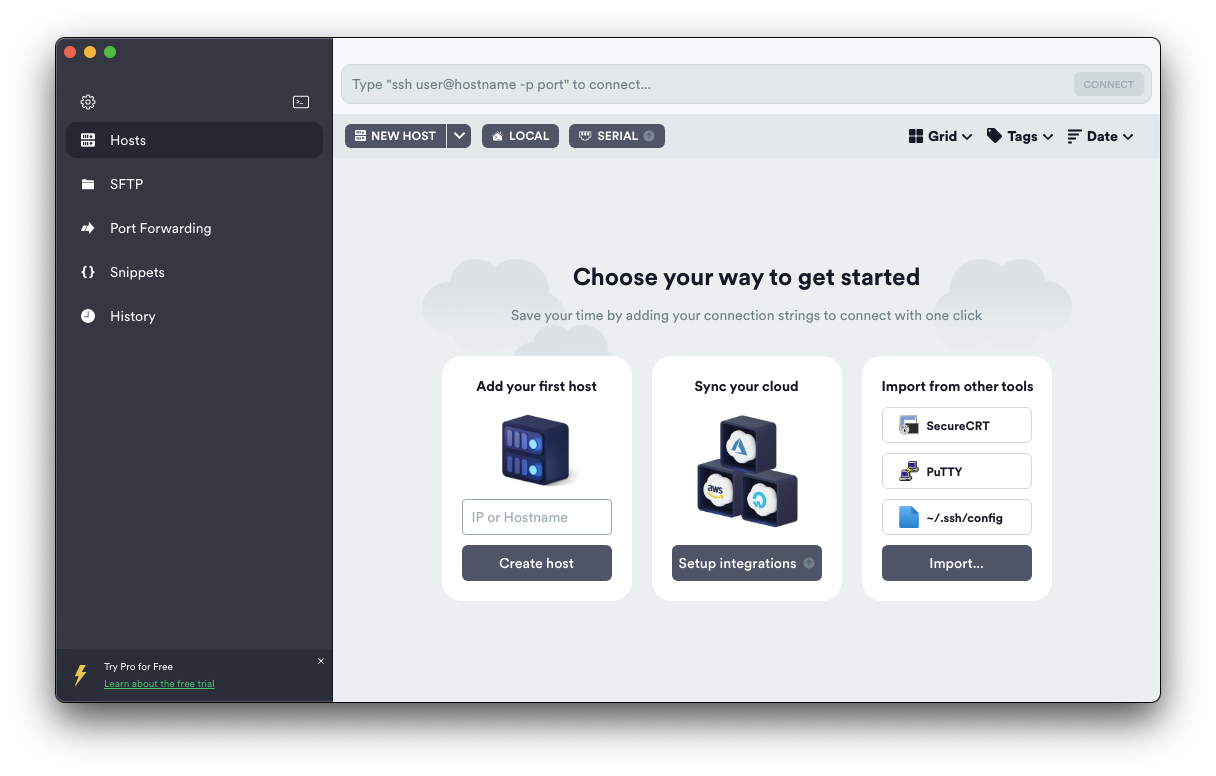


Рисунок 1.1 ‒ Интерфейс приложения «Termius»

1. «Putty» - эта программа является бесплатной и открытой системой для удаленного управления файлами и операционными системами.

Достоинства программы:

* бесплатная;

Недостатки программы:

* устаревшая;
* не продуманный интерфейс.

Для более наглядного представления на рисунке 1.2 приведен интерфейс приложения:

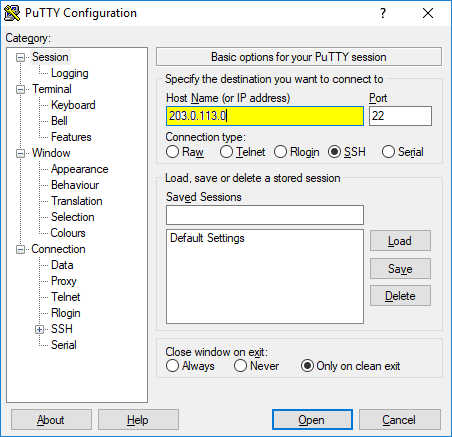


Рисунок 1.2 ‒ Интерфейс приложения «Putty»

Ниже представлена таблица 1.3, в которой приведен анализ программ-аналогов.

Таблица 1.3 — Анализ программ-аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Termius | Putty | Дипломный проект |
| Цена | Высокая | Бесплатная | Бесплатная |
| Установка и настройка | Требуются навыки | Требуются навыки | Простая |
| Скорость работы | Высокая | Средняя | Высокая |
| Масштабируемость | Средняя | Средняя | Средняя |

Как видно из таблицы, каждая из программ-аналогов имеет свои преимущества и недостатки, и для разработки приложения было принято решение использовать собственное решение с учетом конкретных потребностей.

## Моделирование проектируемой системы

На рисунке 1.6 приведена диаграмма IDEF0, модель «TO-BE».

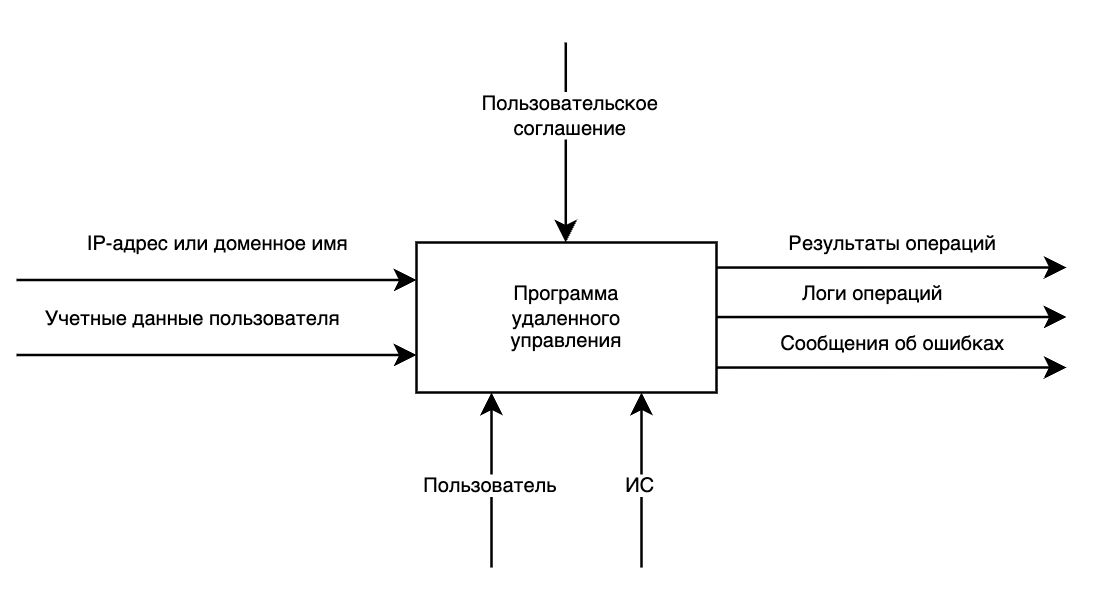


Рисунок 1.6 ‒ Диаграмма IDEF0, модель «TO-BE» для процесса «Техподдержка сотрудников компании»

Управление процессом включает в себя политику пользовательского соглашения, которые необходимы для обеспечения правил использования.

Процесс начинается с добавления пользователем данных для подключения, после чего ИС проверяет доступность использую текущие данные.

После проверки доступности, выполняется команда, которую ввел пользователь и ее результат. Результат построения декомпозиции диаграммы IDEF0, модель «TO-BE» приведен на рисунке 1.7.

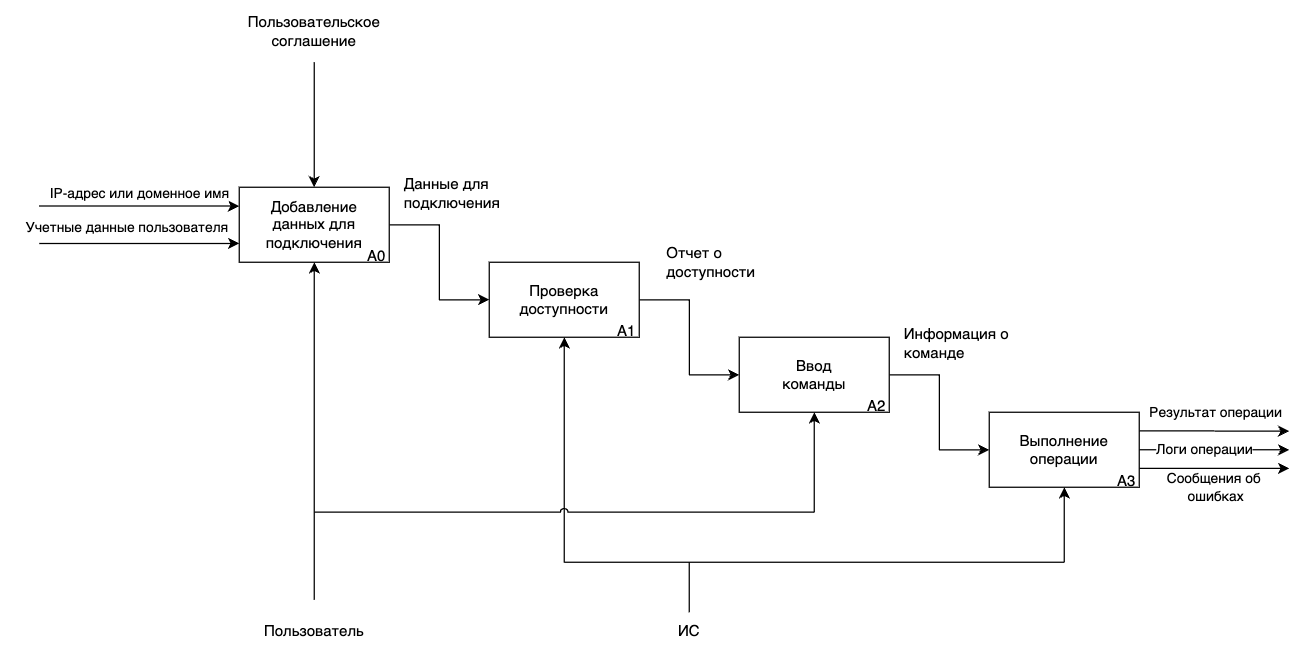


Рисунок 1.7 ‒ Декомпозиция диаграммы IDEF0, модель «TO-BE»

## Разработка функциональных требований программной системе

### Назначение программы

Назначением программы разработки системы удаленного управления файлами и операционными системами является обеспечение эффективной и безопасной удаленной работы сотрудников компании, что позволит повысить их производительность и мобильность, а также обеспечить высокий уровень защиты данных. Программа позволит сотрудникам получать доступ к необходимым файлам и приложениям с любого устройства и из любой точки мира, а также обеспечит быстрый и удобный доступ к технической поддержке. Кроме того, система удаленного управления позволит оптимизировать работу IT-отдела компании, ускорив процессы технической поддержки и повысив качество обслуживания.

### Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения следующих функций:

* Управление файлами: система должна обеспечивать возможность удаленного управления файлами, включая загрузку, выгрузку, создание, изменение, удаление и перемещение файлов между удаленным и локальным компьютером.
* Управление процессами: система должна обеспечивать возможность удаленного управления процессами, включая запуск, остановку, приостановку и возобновление процессов на удаленном компьютере.
* Удаленный доступ: система должна обеспечивать безопасный удаленный доступ к удаленному компьютеру с помощью протоколов, таких как SSH.
* Многопользовательский доступ: система должна обеспечивать возможность многопользовательского доступа к удаленному компьютеру, чтобы несколько пользователей могли работать на одном удаленном компьютере одновременно.
* Автоматическое обновление: система должна обеспечивать возможность автоматического обновления удаленной операционной системы и программного обеспечения для обеспечения безопасности и надежности системы.
* Управление правами доступа: система должна обеспечивать возможность управления правами доступа для пользователей удаленного компьютера, чтобы администратор мог определить, какие пользователи имеют доступ к каким файлам и приложениям.

### Требования к надежности

Надежность технических средств обеспечивается использованием сертифицированных средств вычислительной техники и их комплектующих. При разработке ПО важным критерием является наделжность. Это означает, что программа должна работать стабильно и без сбоев, а также обеспечивать безопасность и конфиденциальность данных. Для обеспечения надежности программы следует учитывать следующие требования:

1. Предусмотреть обработку исключений, связанных с вводом пользователем неправильных данных;
2. Стабильность работы: программа должна работать стабильно и без сбоев;
3. Резервное копирование: программа должна иметь механизмы резервного копирования данных, чтобы в случае сбоя или потери данных можно было быстро восстановить работоспособность системы.

### Требования к составу и параметрам технических средств

Минимальные системные требования:

* + Процессор с тактовой частотой не менее 2,0 ГГц;
  + Объем оперативного запоминающего устройства – 4 Гб и более;
  + Объем постоянного запоминающего устройства – 16 Гб и более.

### Требования к информационной и программной совместимости

* + Приложение должно быть разработано на языке программирования TypeScript.
  + Данные пользователей и заказов должны храниться в базе данных PostgreSQL.
  + Приложение должно поддерживать передачу данных через протокол HTTPS и использовать механизмы шифрования для защиты конфиденциальной информации.
  + Приложение должно быть совместимо с операционными системами Windows и Linux.

### Требования к программной документации

Программа должны быть интуитивно понятной и расширяемой при необходимости. Разработанные программные модули (программа) должны быть самодокументированы, т.е. листинг программы должен содержать все необходимые комментарии.

В состав сопровождающей документации должны входить:

1. пояснительная записка, содержащая описание разработки:
   * схемы алгоритма программы;
   * отладка программы;
   * тестирование программы;
   * руководство пользователя;
2. приложения:
   * листинг программы;
   * результаты выполнения программы.

Разрабатываемая программа должна включать подсказки пользователю для более удобной работы.

### Обоснование выбора средств в реализации программной системы

Для реализации серверной части был выбран язык программирования TypeScript и хостинг firstvds.ru. Для реализации клиентской части - библиотека React. Данный выбор обусловлен рядом преимуществ, которые позволят эффективно и быстро создать новое ПО.

В первую очередь, TypeScript был выбран из-за его типизации, которая позволя-ет уменьшить количество ошибок в коде и сделать его более понятным и легко под-держиваемым. TypeScript также является совместимым с JavaScript, что позволит ис-пользовать существующий код и библиотеки.

Вторым выбранным инструментом является хостинг firstvds.ru. Он был выбран благодаря его надежности и доступности. Регистрация и настройка домена производит-ся просто и быстро, а цены являются конкурентными на рынке.

Библиотека React, который позволяет быстро создавать интерактивные пользо-вательские интерфейсы. React также обладает хорошей производительностью и мас-штабируемостью, что позволит удовлетворить потребности кондитерского предприятия и его клиентов.

В итоге, выбранный инструментарий является оптимальным для реализации программной системы для кондитерского предприятия. TypeScript и React позволят со-здать качественный и интерактивный пользовательский интерфейс, а хостинг firstvds.ru обеспечит доступность и надежность системы.

## Вывод по разделу

В разделе 1.1 была рассмотрена предметная область разработки приложения для разработки системы удаленного управления файлами и операционными системами. Были представлены диаграммы IDEF0, которые показывают процесс обработки данных в информационной системе и политику управления процессом.

В разделе 1.2 был проведен анализ существующих программ-аналогов и определены их основные преимущества и недостатки.

В разделе 1.3 были приведены диаграммы:

* Последовательности
* Деятельности
* USE CASE

В разделе 1.4 были разработаны функциональные требования к программной системе. Была определена функциональность каждого модуля, а также определены требования к интерфейсу приложения.

В разделе 1.4.7 был обоснован выбор средств реализации программной системы. Были рассмотрены различные технологии и языки программирования, которые могут быть использованы для разработки приложения. Также были определены требования к хостингу и базе данных.

Таким образом, проектирование и разработка является актуальной задачей, которая позволит упростить и оптимизировать работу. Реализация данного проекта требует тщательного анализа и выбора средств реализации, а также разработки функциональных требований к программной системе и моделирования проектируемой системы.

# Разработка программного продукта

## Разработка архитектуры программной системы

Каждый пользователь может перемещаться по разделам приложения из любого модуля. Единственное, если роль пользователя – администратор, то он будет иметь доступ к админ панели. Увидеть схему архитектуры приложения, можно на рисунке 2.1.

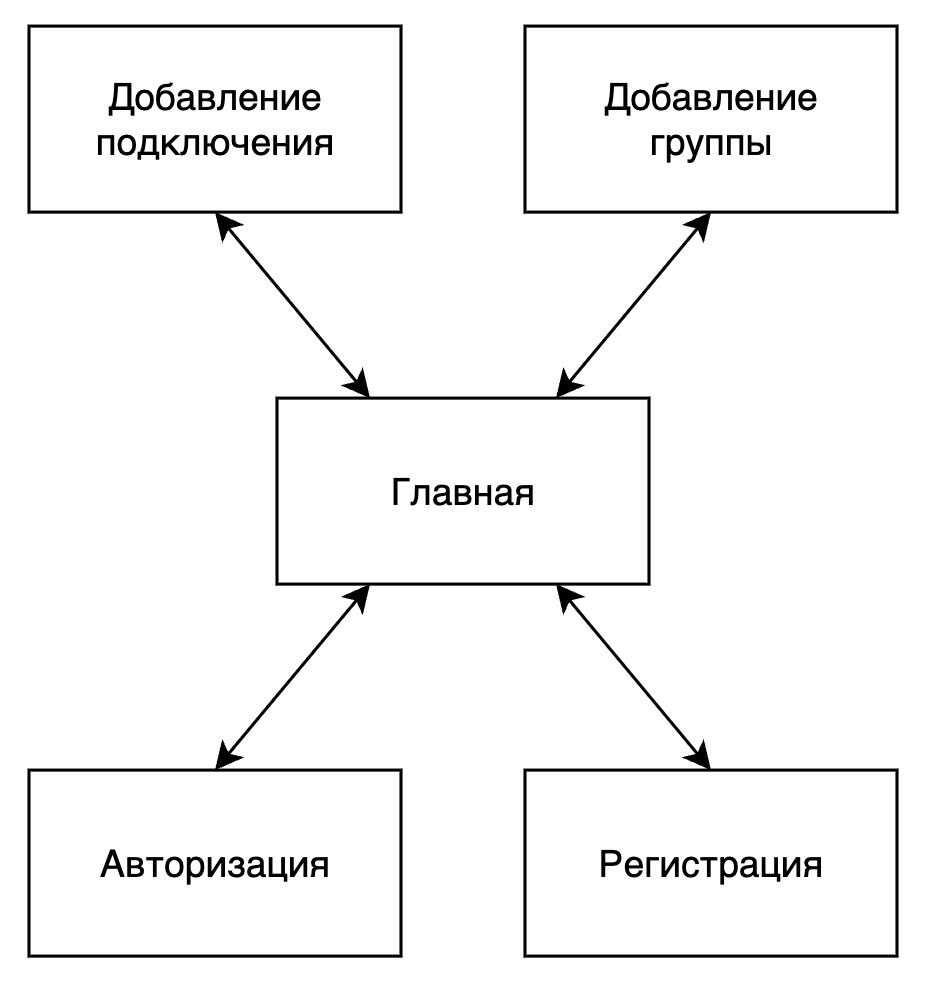


Рисунок 2.1 ‒ Структура программы

В таблице 2.1 представлены методы, используемые в программе и их назначение.

Таблица 2.1 — Методы, используемые в программе

|  |  |
| --- | --- |
| Метод | Назначение |
| register | Используется для регистрации пользователя |

Продолжение таблицы 2.1— Методы, используемые в программе

|  |  |
| --- | --- |
| login | Используется для входа пользователя |
| getUser | Используется для получения пользователя |
| updateUser | Используется для обновления данных пользователя |
| deleteUser | Используется для удаления пользователя |
| createConnection | Используется для создания соединения |
| deleteConnection | Используется для удаления соединения |
| getConnections | Используется для получения соединения |
| updateConnection | Используется для обновления соединения |
| createGroup | Используется для создания группы |
| deleteGroup | Используется для удаления группы |
| getGroups | Используется для получения групп |
| updateGroup | Используется для обновления группы |

Формы, используемые в основной программе, приведены в таблице 2.2.

Таблица 2.2 — Формы, используемые в основной программе

|  |  |
| --- | --- |
| Форма | Назначение |
| Auth | Экран «Авторизация» с возможностью авторизации |
| Register | Экран «Регистрациии» с возможностью регистрации |
| CreateGroup | Экран «Добавления группы» с возможностью добавления группы |
| CreateConnection | Экран «Добавления соединения» с возможностью добавления соединения |

## 

## Разработка структуры данных

Схема базы данных, указанная на рисунке 2.1, содержит информацию о пользователях, продуктах, заказах, корзинах, статистике продаж и сотрудниках.

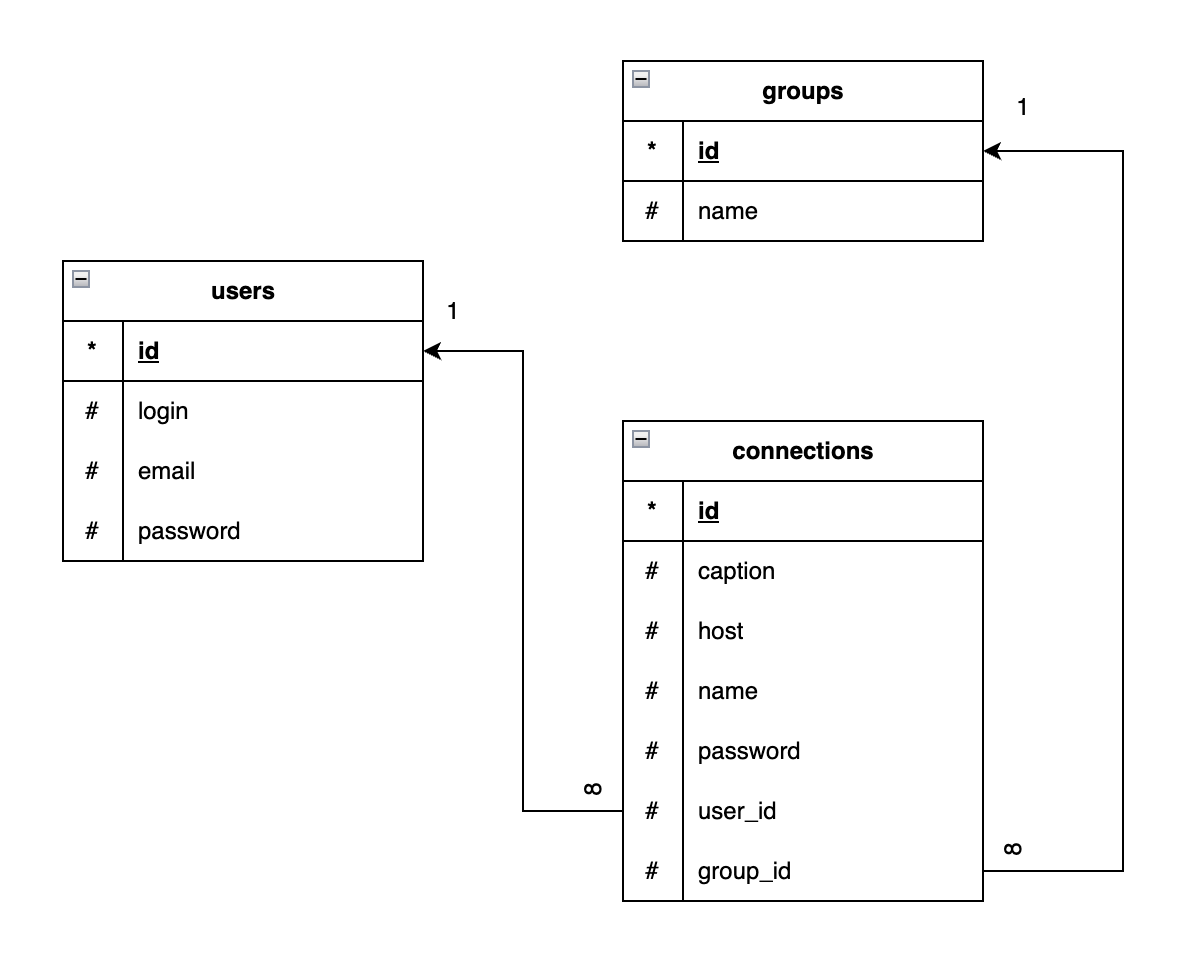


Рисунок 2.1 — Логическая схема базы данных

В таблицах 2.2, 2.3 и 2.4 представлены структуры таблиц разрабатываемой базы данных terix.

Таблица 2.2 – Описание таблицы users

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип поля | Описание поля |
| id | INT | Код пользователя |
| login | VARCHAR(255) | логин пользователя |
| email | VARCHAR(255) | Почта пользователя |
| password | VARCHAR(255) | Пароль пользователя |

Таблица 2.3 – Описание таблицы groups

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип поля | Описание поля |
| id | INT | Код группы |
| name | VARCHAR(255) | Название группы |

Таблица 2.4 – Описание таблицы connections

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип поля | Описание поля |
| id | INT | Код соединения |
| caption | VARCHAR(255) | Название соединения |
| host | VARCHAR(255) | Хост соединения |
| login | VARCHAR(255) | Логин соединения |
| password | VARCHAR(255) | Пароль соединения |
| User\_id | INT | Код пользователя |
| Group\_id | INT | Группа соединения |

## Конструирование пользовательского интерфейса

Интерфейс является важной частью любого приложения, и его создание должно основываться на удобстве и простоте использования. Чтобы пользователь мог легко получить нужный результат, необходимо, чтобы интерфейс был интуитивно понятен.

На рисунке 2.3 представлена форма «Авторизация», которая предоставляет пользователю возможность зарегистрироваться или войти в свой аккаунт и перейти к главному меню.



Рисунок 2.3 — Экран «Авторизация»

На рисунке 2.4 представлен экран «Регистрация», где пользователь может вернуться на экран авторизации или заполнить необходимую информацию, чтобы включилась синхронизация между аккаунтами.



Рисунок 2.4 — Экран «Регистрация»

На рисунке 2.6 представлен экран «Главная». С помощью левой панели навигации пользователю предоставляется возможность перейти на экраны: «Главная», «SFTP», увидеть открытые соединения с серверами, а так же историю подключений. В верхней панели вкладок пользователь может произвести, а так же добавить новое соединение и группу.

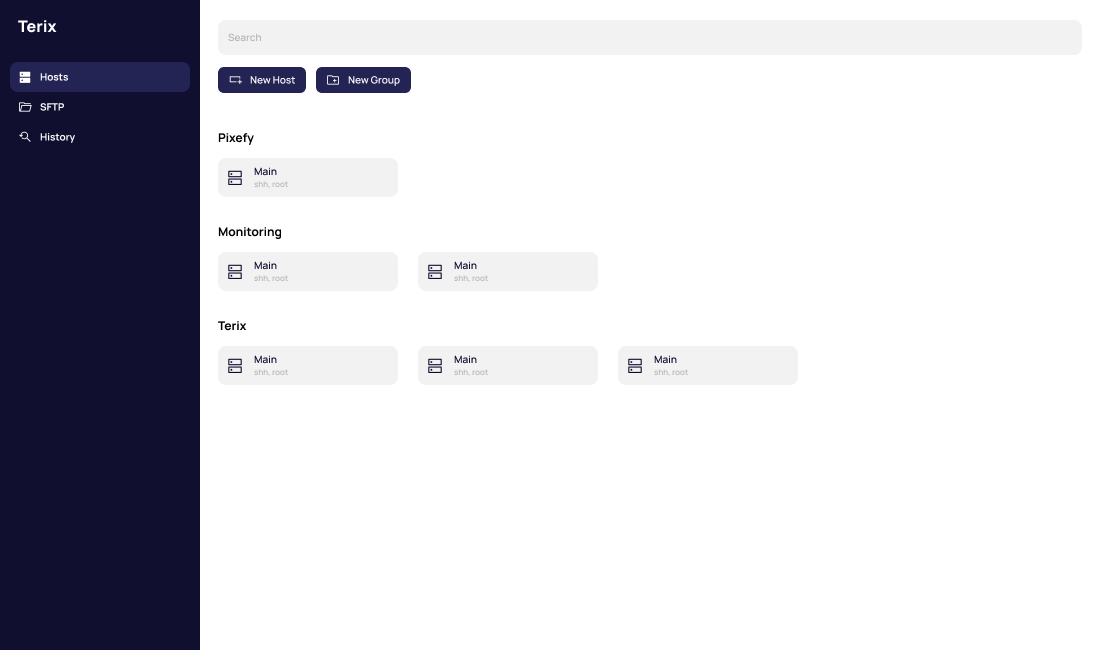


Рисунок 2.6 — Экран «Главная»

## Схемы алгоритма программы и подпрограмм

### Схема алгоритма функции авторизации

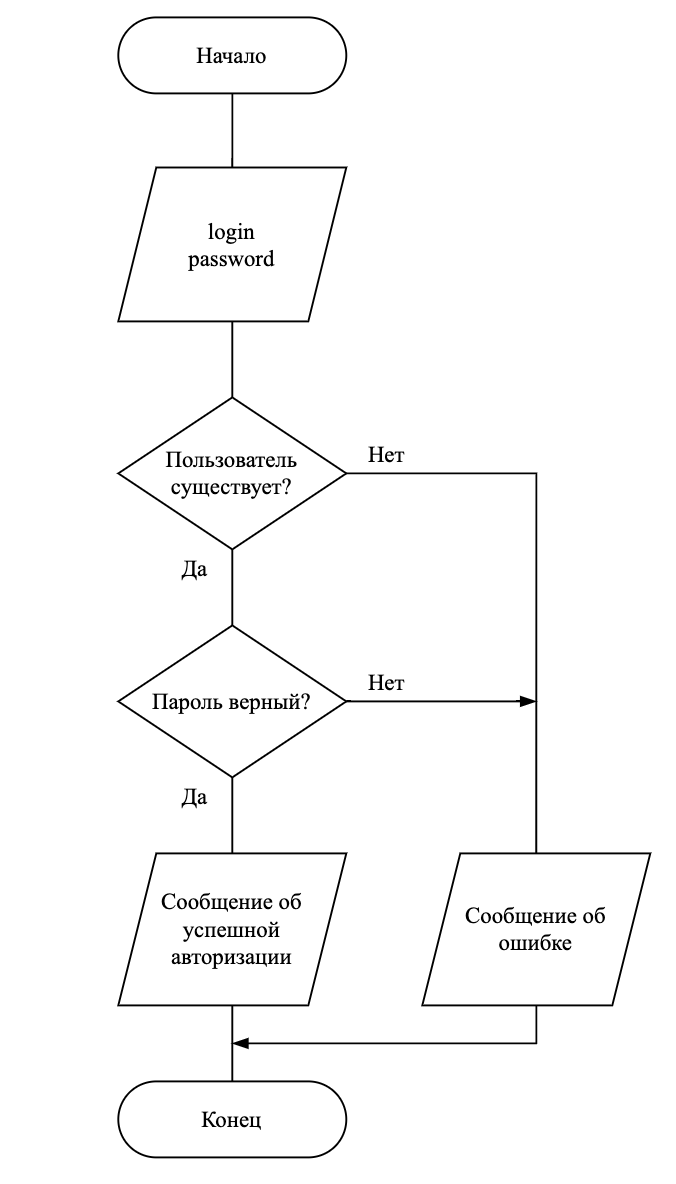


Рисунок 2.7 — Схема алгоритма функции «Авторизация». login – логин; password – пароль.

### Схема алгоритма функции регистрации

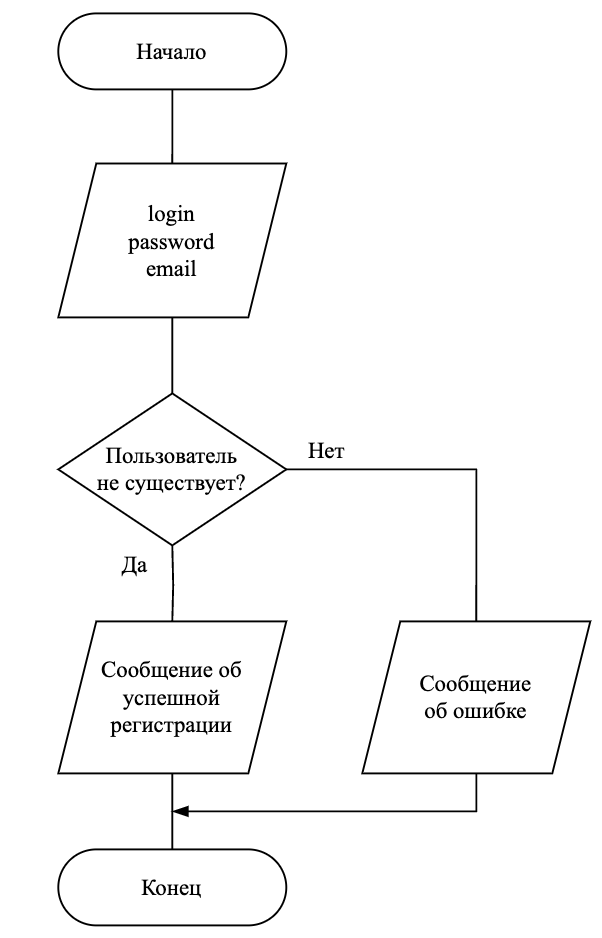


Рисунок 2.8 — Схема алгоритма функции «Регистрация». login – логин; password – пароль; email – электронная почта.

### Схема алгоритма функции создания группы

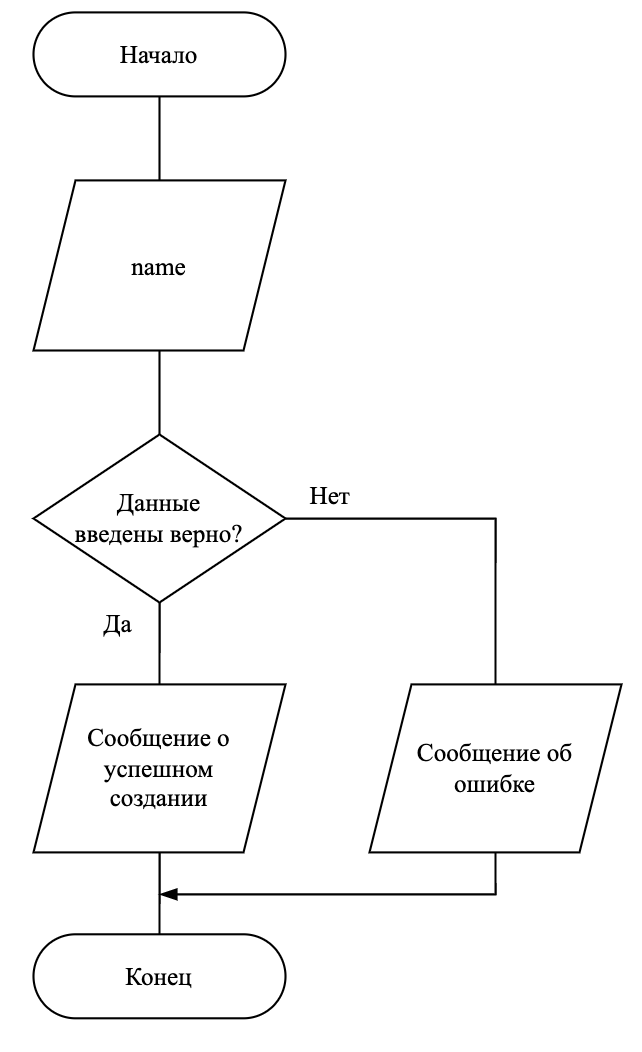


Рисунок 2.9 — Схема алгоритма функции «Создания группы». name – название.

### Схема алгоритма функции создания соединения

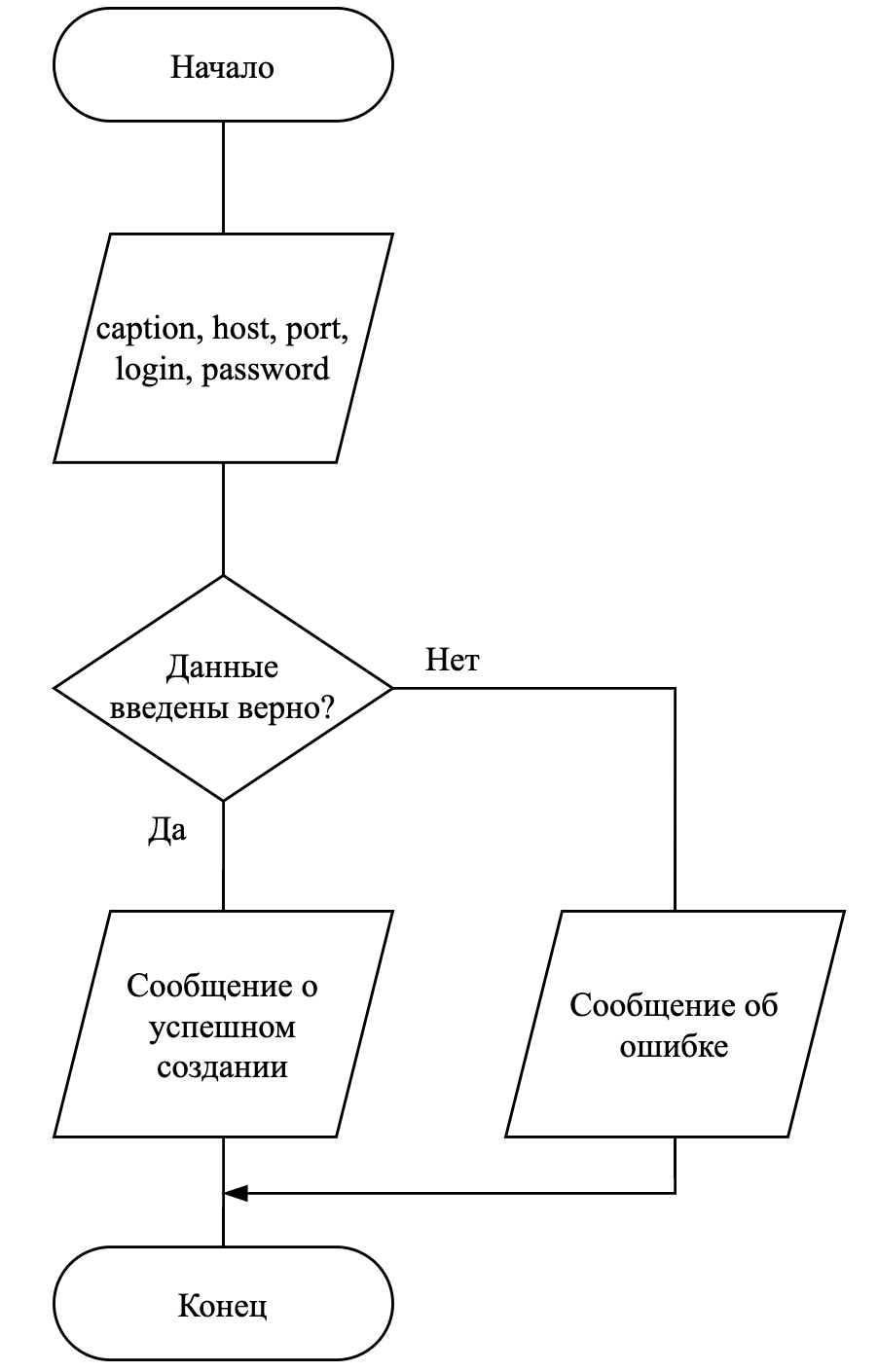


Рисунок 2.10 — Схема алгоритма функции «Создания соединения». caption – название; host – адрес; port – порт; login – логин; password – пароль.

## Отладка и тестирование программы

### Отладка программы

Отладка программы – это процесс обнаружения и исправления ошибок в программном коде с целью устранения несоответствий и неправильных действий. Этот этап разработки обычно занимает значительное время и может быть даже более трудоемким, чем написание самой программы. Почти каждая программа содержит ошибки до начала процесса отладки.

Синтаксические ошибки – это ошибки, которые компилятор (транслятор, интерпретатор) обнаруживает при анализе синтаксиса программы и, в некоторой степени, её семантики. Они возникают, когда исходный код не соответствует синтаксису выбранного языка программирования.

Ошибки компоновки – это ошибки, которые возникают при объединении модулей программы компоновщиком (редактором связей). Они происходят, когда связи между модулями и подпрограммами не совпадают или при обращении к ним возникают проблемы.

Ошибки выполнения – это ошибки, которые возникают во время выполнения программы и часто приводят к её преждевременному завершению. Наиболее распространённым примером такой ошибки является деление на ноль.

Ошибки логики – это ошибки, которые препятствуют программе выполнять ожидаемые действия. Код может быть скомпилирован и выполнен без ошибок, но результат операций может быть непредсказуемым и неверным.

Во время отладки данной программы были допущены следующие ошибки:

1. синтаксические ошибки:
   * ошибки в названиях переменных
   * неправильное имя оператора Пример ошибки:

d = SELECT \* FROM Users; Исправленный вариант:

d = "SELECT \* FROM Users";

1. ошибки компоновки:
   * неверный список перечисляемых параметров.

Пример ошибки:

string d = String.Format("INSERT INTO ? VALUES (?)", dto); Исправленный вариант:

string d = String.Format("INSERT INTO ? VALUES (?)", "Users", dtos);

1. ошибки выполнения:
   * несовпадение полученных результатов с ожидаемыми;
   * неверное использование неполного типа;
   * неправильно организованный цикл с параметром. Пример ошибки:

// Несовпадение полученных результатов с ожидаемыми

function addNumbers(a: number, b: number): number {

return a - b; // Неправильно вычитаем вместо сложения

}

const result = addNumbers(2, 3);

console.log(result); // Ожидалось 5, но получили -1

Исправленный вариант:

function addNumbers(a: number, b: number): number {

return a + b; // Исправлено на сложение

}

const result = addNumbers(2, 3);

console.log(result); // Выводится 5, как ожидалось

Все ошибки были исправлены. Для того, чтобы убедиться в правильности работы программы, необходимо провести её тестирование.

### Тестирование программы

Тестирование – это набор процедур и действий, предназначенных для демонстрации правильности работы программы в заданных режимах и внешних условиях.

Цель тестирования – выявить наличие ошибок или убедительно

продемонстрировать их отсутствие. Процесс тестирования проходит в 3 этапа:

* проверка поведения программы в нормальных условиях;
* проверка поведения программы в экстремальных условиях;
* проверка поведения программы в исключительных ситуациях.

Каждый из этапов предполагает задание определенного, характерного для данного этапа набора данных.

### Тестирование программы в нормальных условиях

Тестирование в нормальных условиях было проведено со всеми формами. Были протестированы переходы от экрана к экрану, предоставление возможности входа в систему и регистрации, автоматическое обновление списков. Тестирование процедуры добавления, удаления и обновления соединения представлено на рисунках 2.16, 2.17, 2.18.

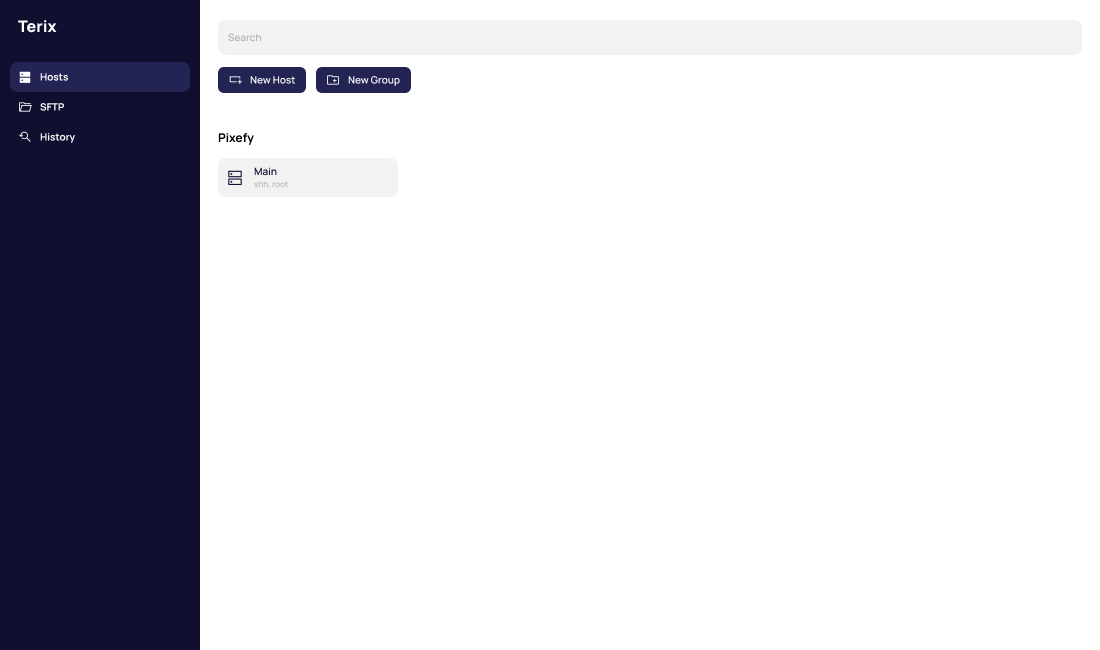


Рисунок 2.16 — Добавление нового соединения

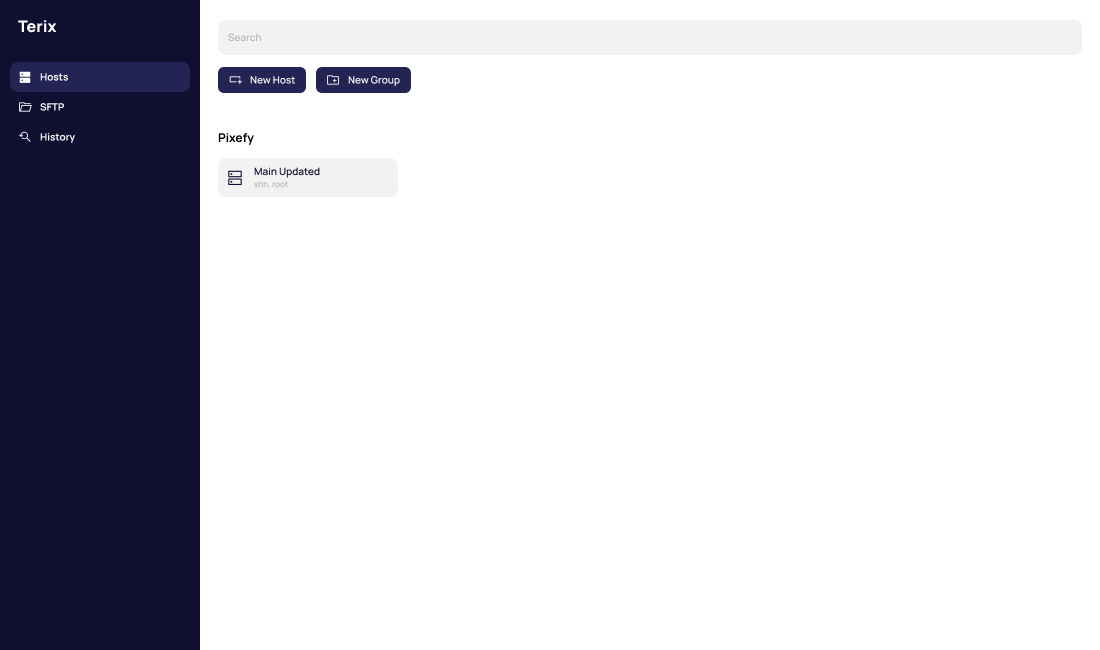


Рисунок 2.17 — Обновление соединения

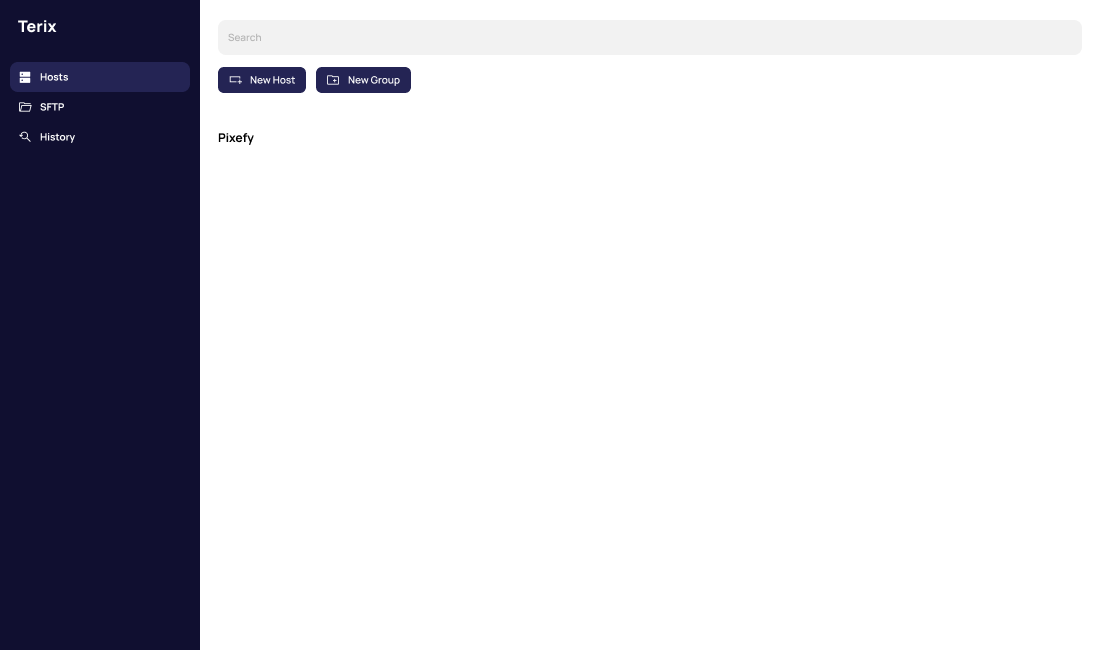


Рисунок 2.18 — Удаление соединения

Тестирование процедуры проверки занятости логина при регистрации новых пользователей с помощью функций, представленных в классе и делегатов на рисунках 2.19 и 2.20.

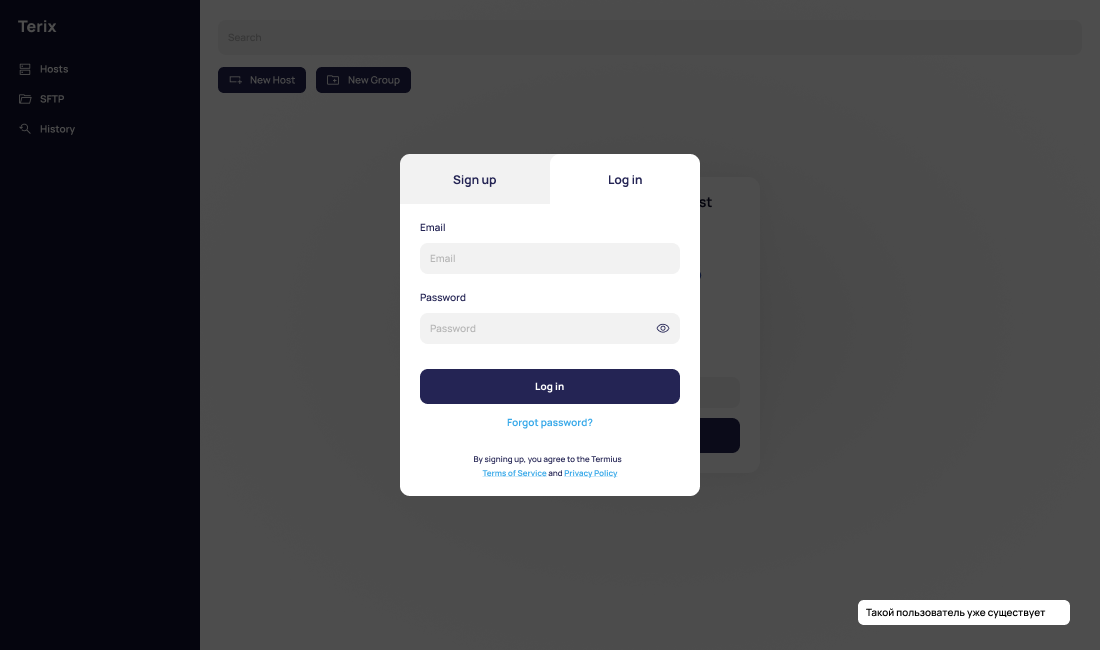


Рисунок 2.19 — Выдача сообщения о том что пользователь уже существует

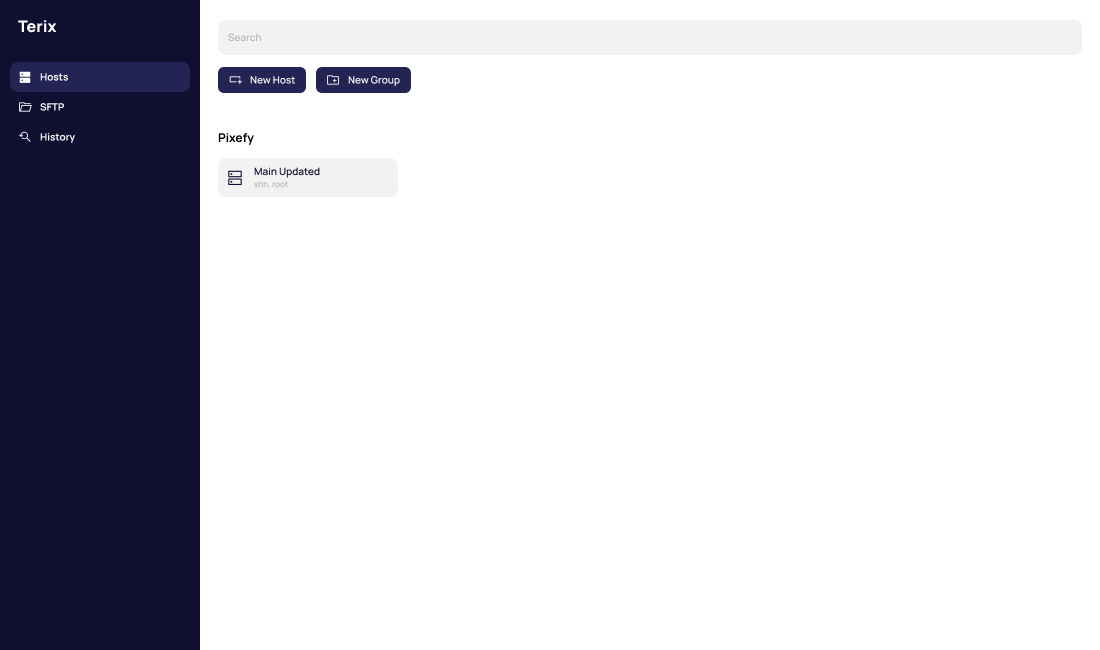


Рисунок 2.20 — Главный экран приложения

Данные рисунки показывают, что программа в нормальных условиях работает правильно.

### Тестирование программы в экстремальных условиях

На рисунке 2.21 показано, что при попытке оставить пустые данные в полях ввода и нажать кнопку «Добавить» будет выдано сообщение «Заполните все поля!».

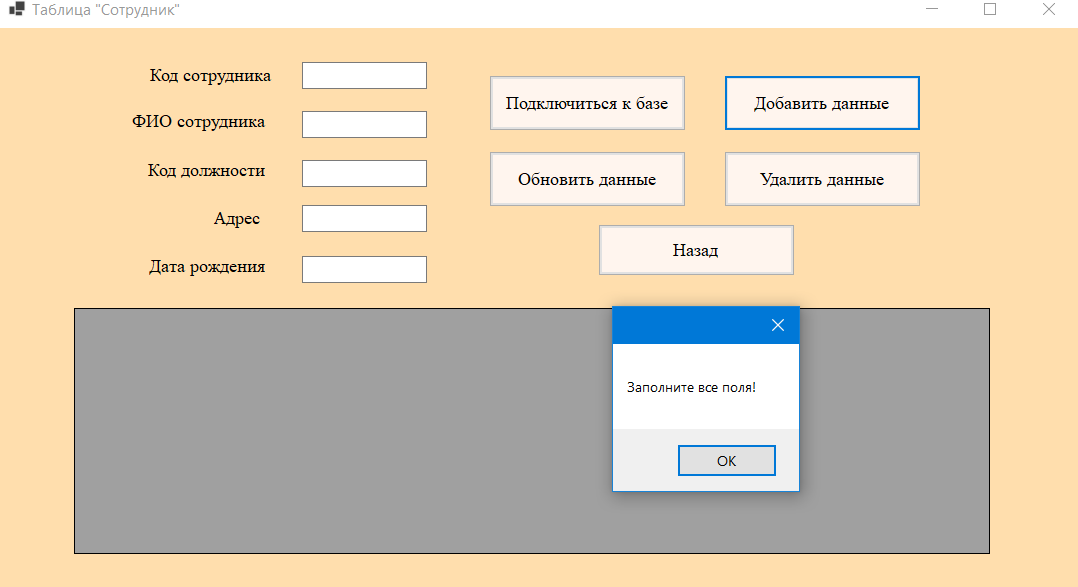


Рисунок 2.21 — Сообщение «Заполните все поля!»

Данный рисунок показывает, что программа в экстремальных условиях работает правильно.

### Тестирование программы в исключительных ситуациях

На рисунках 2.22 и 2.23 проведена демонстрация выдачи сообщений при попытке заполнения полей ввода некорректными данными и нажатия на кнопку «Добавить».

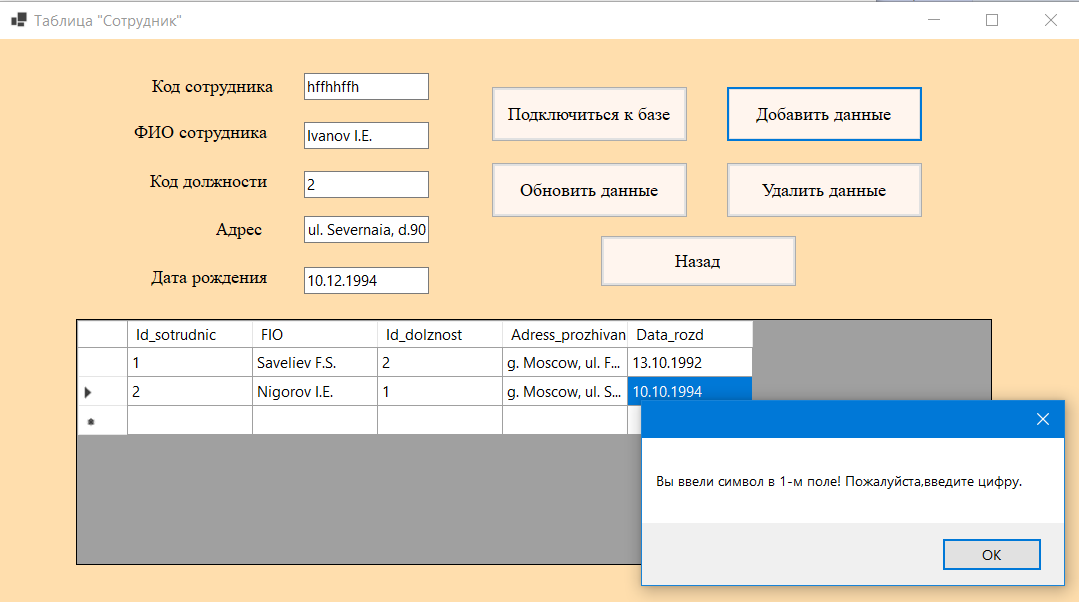


Рисунок 2.22 — Выдача сообщения об ошибке

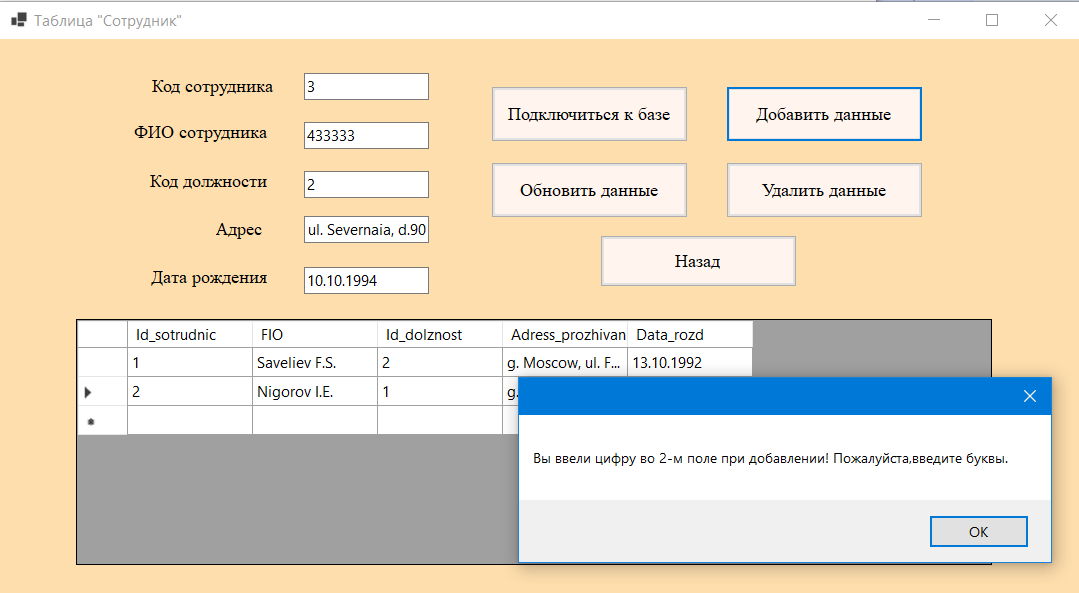


Рисунок 2.23 — Выдача сообщения об ошибке

Данные рисунки показывают, что программа в исключительных условиях работает правильно.

Так как тестирование в нормальных, экстремальных условиях и исключительных ситуациях не выявило наличие ошибок, можно сказать, что программа работает правильно. Листинг программы представлен в приложении А.

### UnitTest

Для тестирования работы программы регистрации были написаны модульные тесты. Листинги модульных тестов приведены ниже. На вход тест получает два варианта пароля, один из которых проходит проверку, а второй – нет. Полученные результаты теста сравниваются с ожидаемыми и при совпадении результатов, тест считается успешно пройденным. Все тесты были пройдены успешно, значит можно сделать вывод, что программа регистрации работает верно. Результаты тестирования программы представлены на рисунке 2.24.

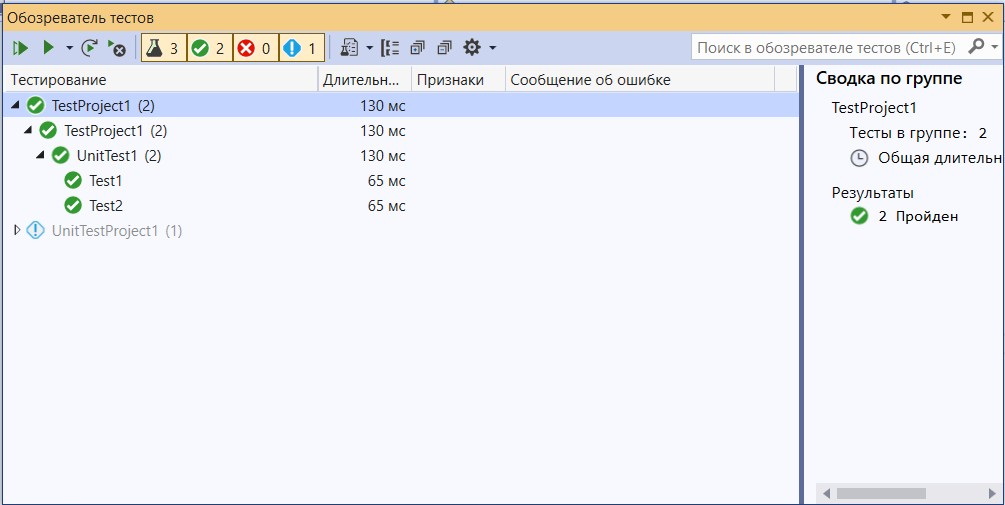


Рисунок 2.24 — Прохождение UnitTest

### Листинг проекта UnitTest1

using System; using Xunit; using Praktica;

namespace TestProject1

{

public class UnitTest1

{

[Fact]

public void Test1()

{

form\_reg m = new form\_reg (); bool res = m.password("Qwerty1"); Assert.Equal(true, res);

}

[Fact]

public void Test2()

{

form\_reg m = new form\_reg ();

bool res = m.password("DEl1fdfhgftyklgfx"); Assert.Equal(false, res);

}

}

### Сообщения

В таблице 2.17 представлены тексты сообщений с их описанием и действий, которые необходимо предпринять после их получения.

Таблица 2.17 — Выдача сообщений

|  |  |  |
| --- | --- | --- |
| Текст сообщения | Описание | Действия для совершения |
| Регистрация прошла успешно! | Пользователь зарегистрирован, данные сохранены в таблице  «Пользователь». | Необходимо нажать на кнопку ниже «Войти в систему» для входа в систему. |
| Проверьте связку пароля и логина, а также введенную роль, она может не совпадать с действительной. | Введен неверная связка логина и пароля или указанная роль не соответствует роли, которая была выбрана при регистрации, и пользователь не сможет войти в систему. | Необходимо изменить роль на противоположную для входа в систему, а также проверить связку логина и пароля. |
| Пользователь уже существует в системе! | При регистрации пользователя введен логин пользователя, уже  существующего в системе. | Необходимо выбрать иной логин и ФИО и попробовать снова. |
| Заполните все поля! | Поля ввода в формах не заполнены или не заполнены до конца. | Необходимо заполнить все поля, представленные на форме. |
| Вы ввели цифру в поле! Пожалуйста, введите буквы. | Выдается при попытке заполнить поля числовыми выражениями вместо буквенных. | Необходимо заполнить поле с ошибкой одно- или двухзначными числами. |
| Вы ввели символ в поле! Пожалуйста, введите цифру. | Выдается при попытке заполнить поля буквенными выражениями вместо  числовых. | Необходимо заполнить поле буквенными выражениями. |
| Не подходит! Попробуйте ввести иной пароль. | Введенный в поле пароль не соответствует требованиям. | Придумайте новый пароль и введите в поле еще раз. |

## Руководство пользователя

### Назначение программы

Данная программа предназначена для системы удаленного управления файлами и операционными системами. Она позволяет зарегистрированным пользователям авторизоваться в системе и получить доступ к функциям удаленного управления файлами и операционными системами.

Система обеспечивает возможность просмотра, редактирования, перемещения и удаления файлов на удаленных компьютерах или серверах. Создавать группы для удобного хранения подключений, а так же получать доступ к созданным подключениям на другом устройстве, за счет синхронизации данных между ними.

### Условия работы программы

Минимальные системные требования:

* + процессор с тактовой частотой не менее 2,0 ГГц;
  + объем оперативного запоминающего устройства – 2 Гб и более;
  + объем постоянного запоминающего устройства – 16 Гб и более;
  + устройство вывода (монитор);
  + устройства ввода (клавиатура, компьютерная мышь). Требования к программному обеспечению:
  + создание подключения;
  + создание группы;
  + восстановление пароля;
  + синхронизация данных;
  + управление серверами;
  + управление файлами;
  + регистрация новых пользователей в системе;
  + вход в систему;

### Входные данные

Ограничения на входные данные представлены в таблице ниже.

Таблица 2.18 — Ограничения на входные данные

|  |  |  |
| --- | --- | --- |
| Таблица | Поле | Количество символов |
| Пользователи | Логин | 50 |
| Пароль | 50 |
| Почта | 100 |
| Соединения | Название | 80 |
| Хост | 100 |
| Логин | 50 |
| Пароль | 50 |
| Группа | Название | 80 |

### Выполнение программы

Для пользования программой необходимо выполнить следующие действия:

1. для запуска программы, необходимо запустить файл «Diplom.exe»;
2. при запуске программы откроется «Главное меню» (рисунок 2.25), в котором можно нажать на кнопки «Вход в систему», «Регистрация» и закрыть приложение;

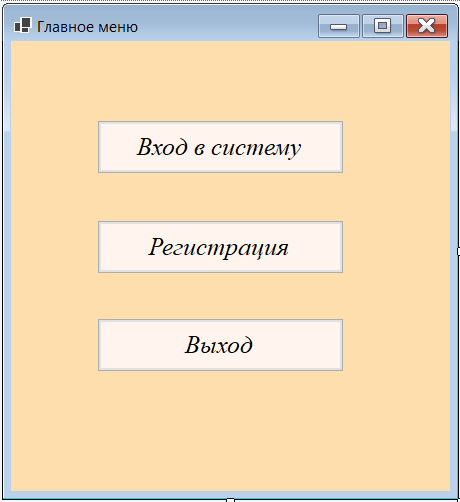


Рисунок 2.25 — Форма «Главное меню»

1. для входа в систему необходимо нажать кнопку «Вход в систему» и появится соответствующее окно (рисунок 2.26);

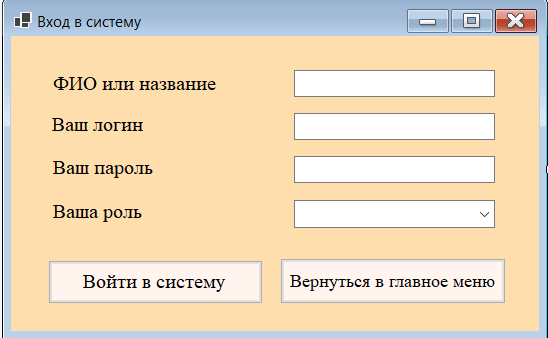


Рисунок 2.26 — Форма «Вход в систему»

1. после ввода данных в поля, выбора роли и при совпадении данных в таблице базы данных «Пользователь» будет осуществлен вход в систему и переход на форму «Работа с БД» (рисунок 2.27) и выдана возможность открытия форм, в зависимости от предоставленной роли;

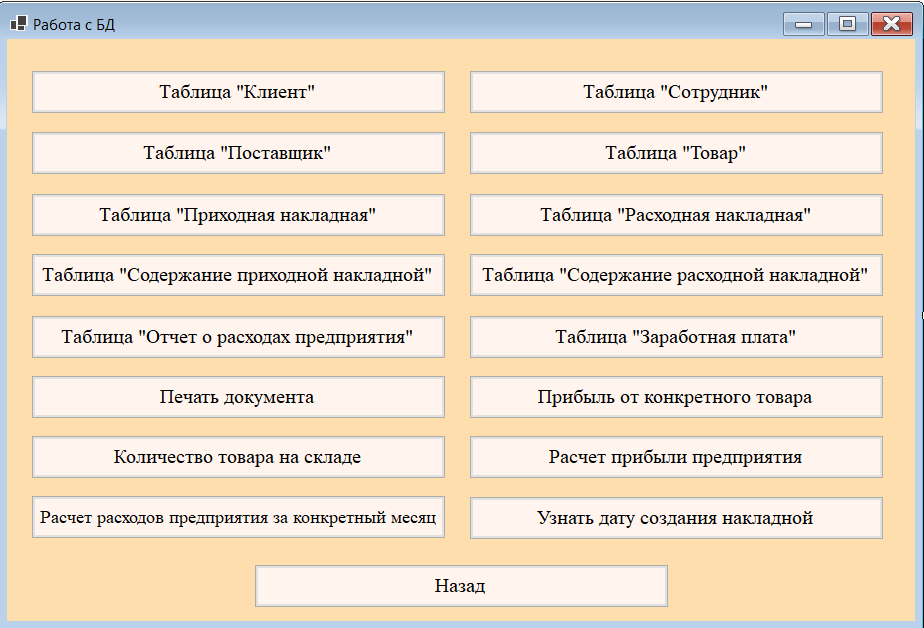


Рисунок 2.27 — Форма «Работа с БД»

1. При входе в систему с ролью бухгалтера откроется следующее окно с сообщением (рисунок 2.28):

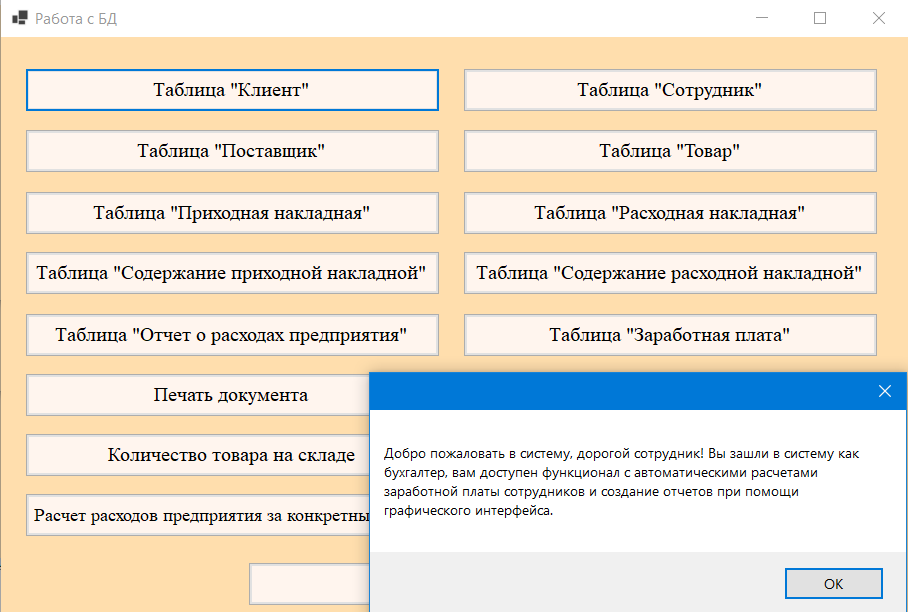


Рисунок 2.28 — Вход в систему как бухгалтер

1. при нажатии на форме «Работа с БД» на кнопку «Печать документа» совершится перенаправление на форму «Получение даты создания расходной накладной», в которой при вводе названия компании будет выдана информация о дате создания накладной (рисунок 2.29, 2.30), доступная только бухгалтеру:

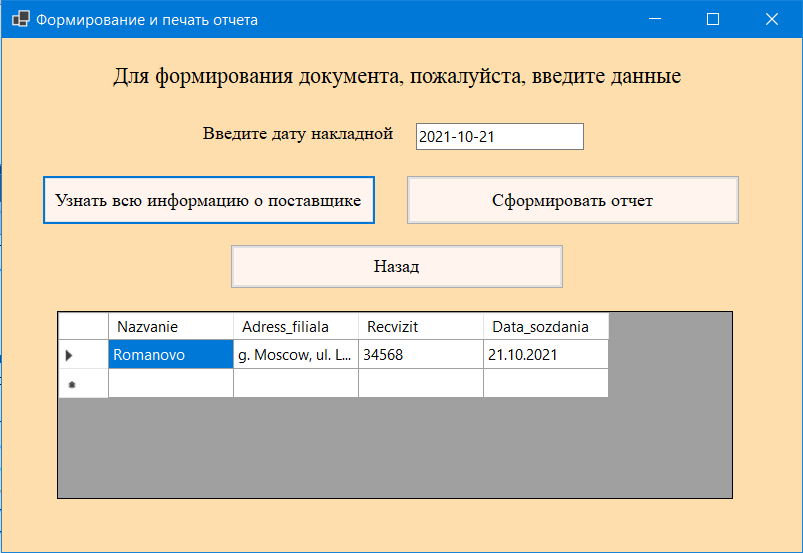


Рисунок 2.29 — Выдача информации о поставщике

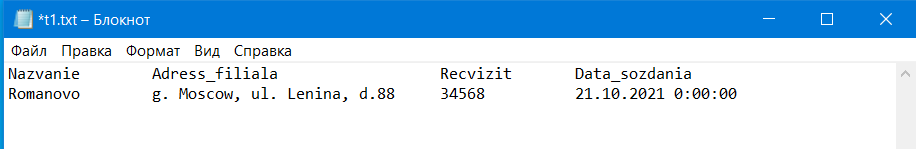


Рисунок 2.30 — Формирование документа

1. при нажатии на форме «Работа с БД» на кнопку «Расчет прибыли предприятия» совершится перенаправление на форму «Расчет прибыли предприятия», в которой при вводе периода продаж предприятия рассчитывается прибыль (рисунок 2.31);

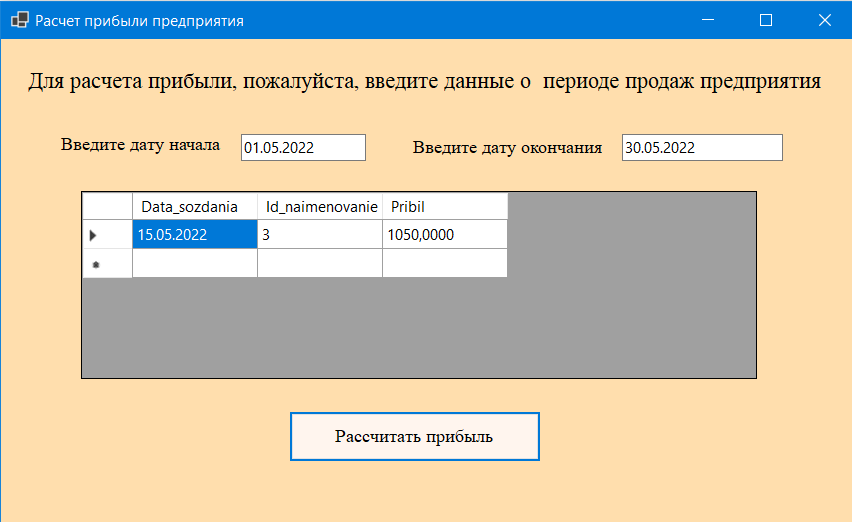


Рисунок 2.31 — Расчет прибыли предприятия

1. при нажатии на форме «Работа с БД» на кнопку «Расчет расходов предприятия» совершится перенаправление на форму «Расчет расходов предприятия», в которой при вводе периода рассчитывается сумма расходов (рисунок 2.32);

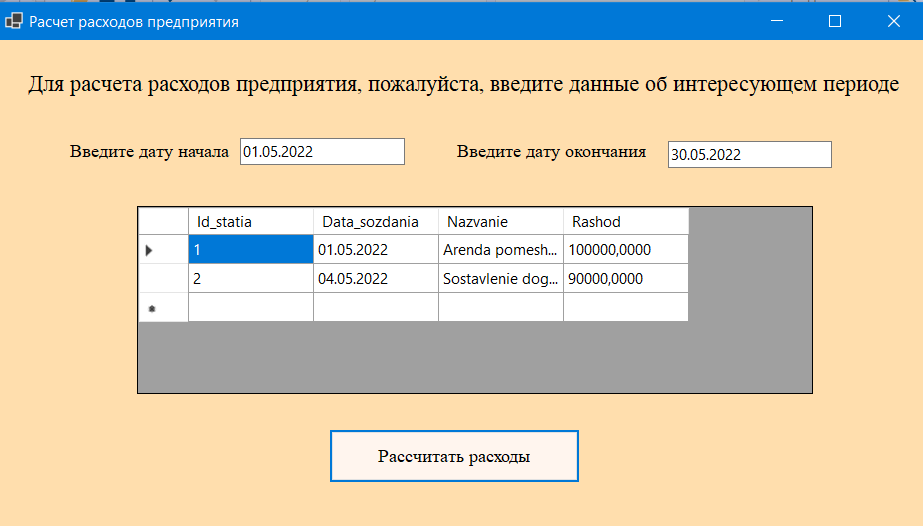


Рисунок 2.32 — Расчет расходов предприятия

1. При входе в систему с ролью менеджера откроется следующее окно с

сообщением (рисунок 2.33):

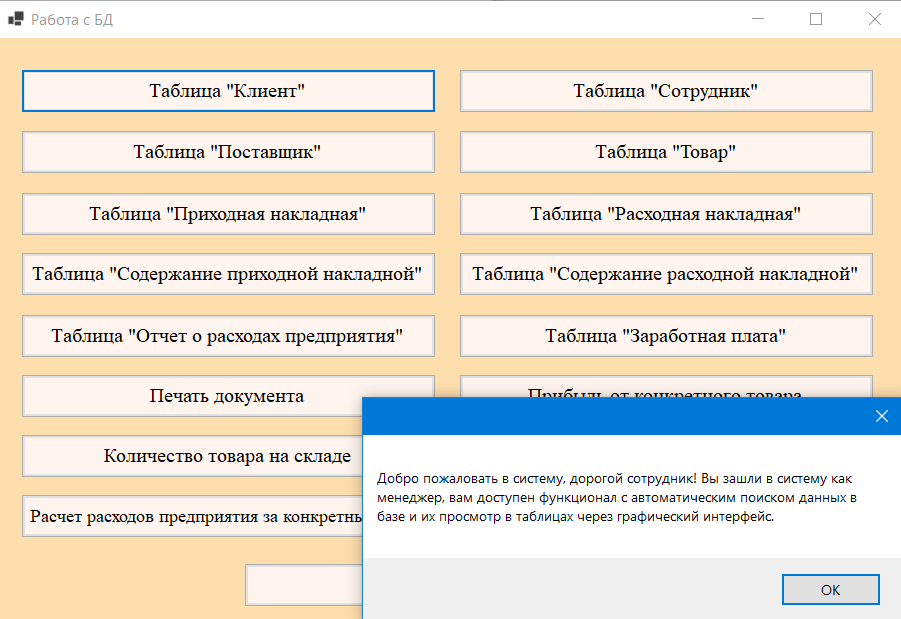


Рисунок 2.33 — Вход в систему как менеджер

1. при нажатии на форме «Работа с БД» на кнопку «Показ количества товара на складе» совершится перенаправление на форму «Показ количества товара на складе», в которой при вводе наименования товара выводятся данные с упоминанием даты последних обновлений (рисунок 2.34), доступная только менеджеру;

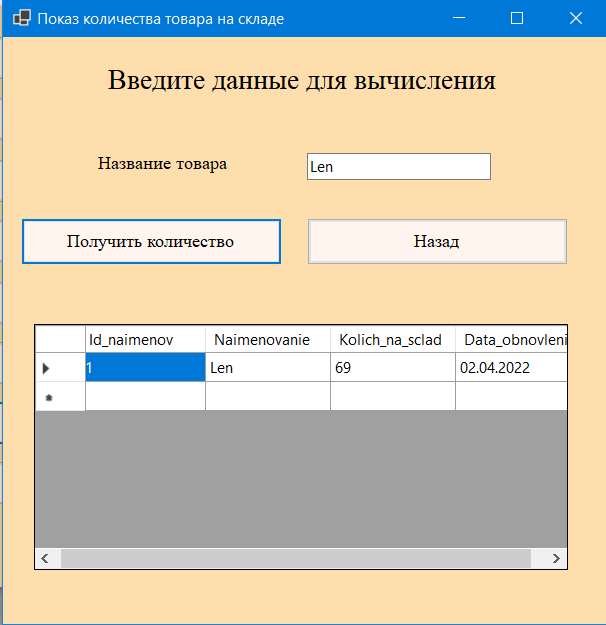


Рисунок 2.34 — Расчет расходов предприятия

1. Ниже будут рассмотрены таблицы, к которым имеет доступ как бухгалтер, так и менеджер. При нажатии на кнопку «Сотрудник» откроется форма «Таблица «Сотрудник»», на которой возможно произвести просмотр данных в таблице, их добавление, обновление и удаление (рисунок 2.35, 2.36, 2.37);

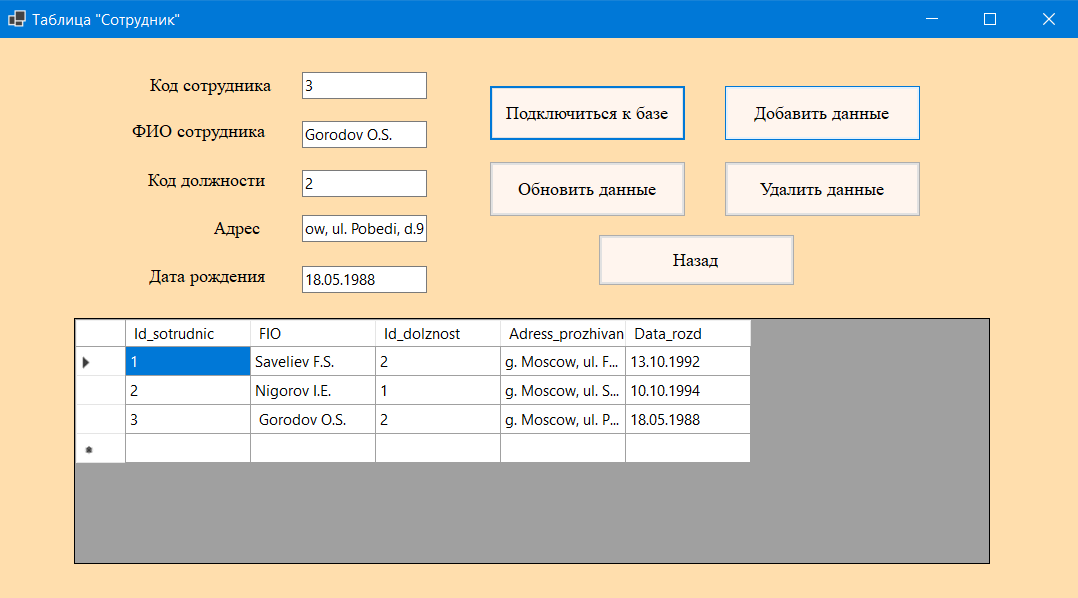


Рисунок 2.35 — Добавление данных в таблицу «Сотрудник»

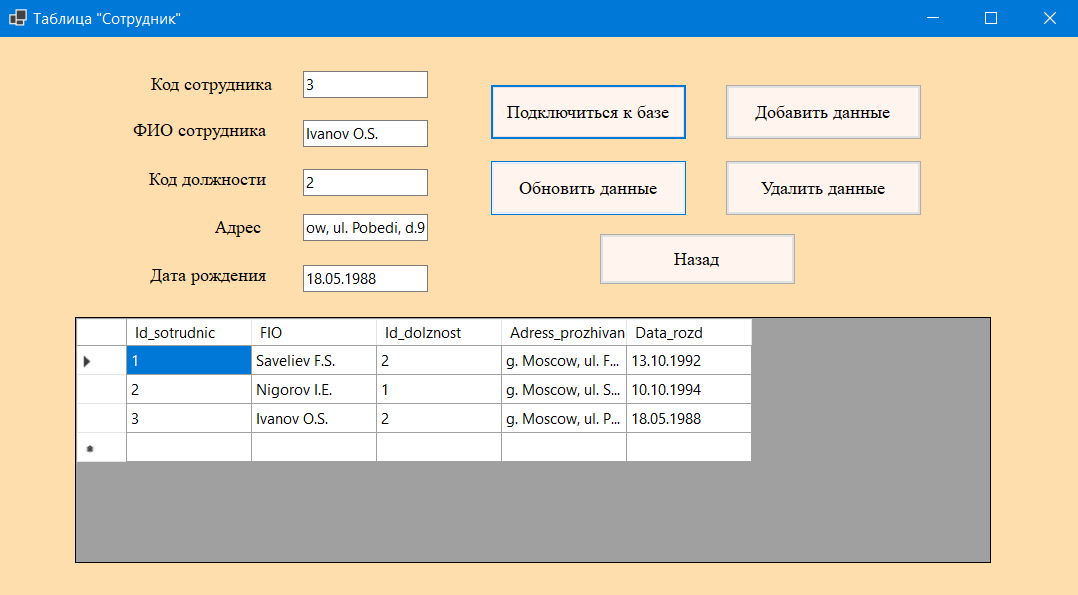


Рисунок 2.36 — Обновление данных в таблице «Сотрудник»

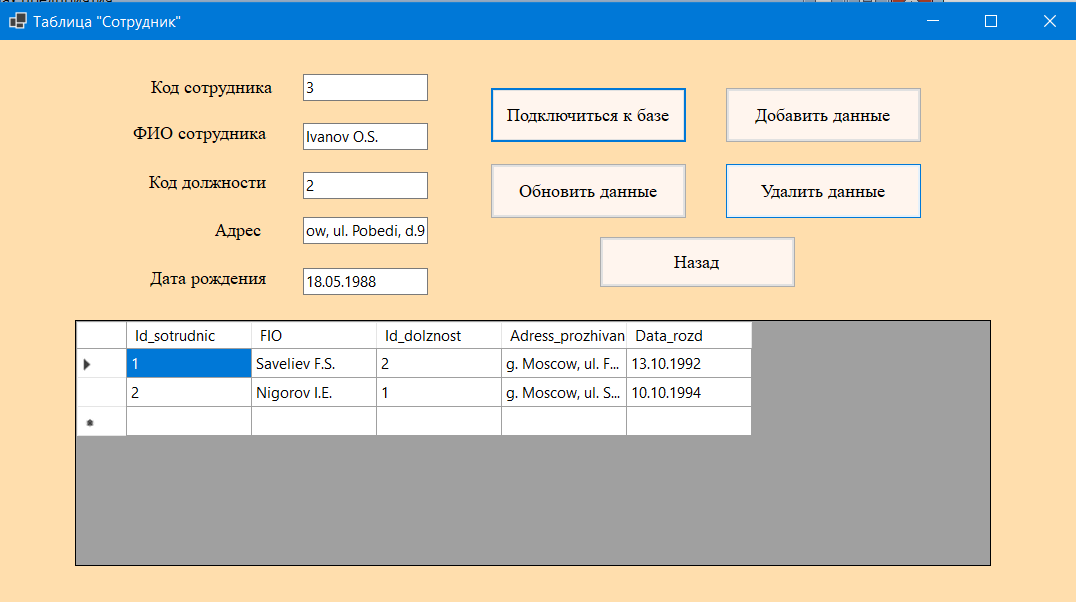


Рисунок 2.37 — Удаление данных в таблице «Сотрудник»

1. при нажатии на кнопку «Назад» будет осуществлено закрытие формы и переход к форме «Вход в систему»;
2. работа с таблицами «Клиент», «Товар», «Поставщик», «Приходная накладная»,

«Расходная накладная», «Содержание приходной накладной», «Содержание расходной накладной», «Отчет о расходах предприятия», «Заработная плата» происходит аналогично;

1. при нажатии на форме «Работа с БД» на кнопку «Узнать дату создания накладной» совершится перенаправление на форму «Получение даты создания расходной накладной», в которой при вводе названия компании будет выдана информация о дате создания накладной (рисунок 2.38);

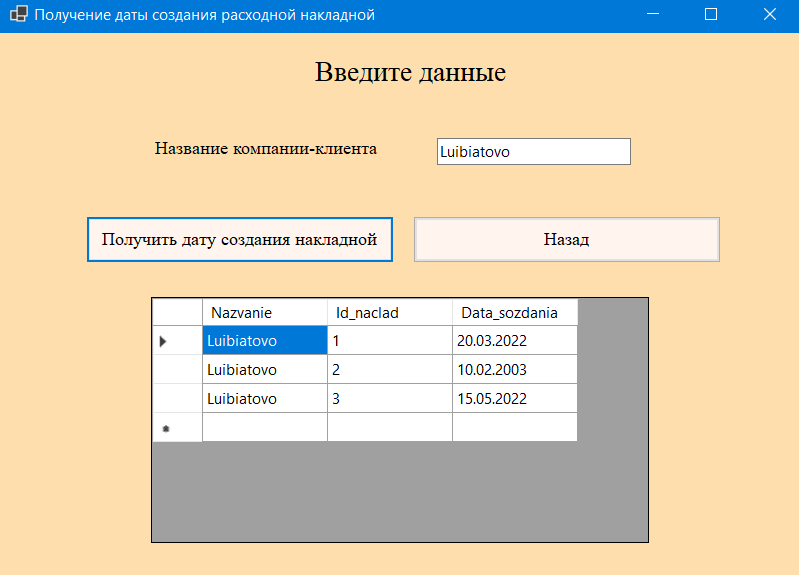


Рисунок 2.38 — Получение информации о дате создания расходной накладной

1. при нажатии на форме «Работа с БД» на кнопку «Прибыль от конкретного товара» совершится перенаправление на форму «Получение выручки проданных товаров», в которой при вводе названия компании будет выдана информация о дате создания накладной (рисунок 2.39);

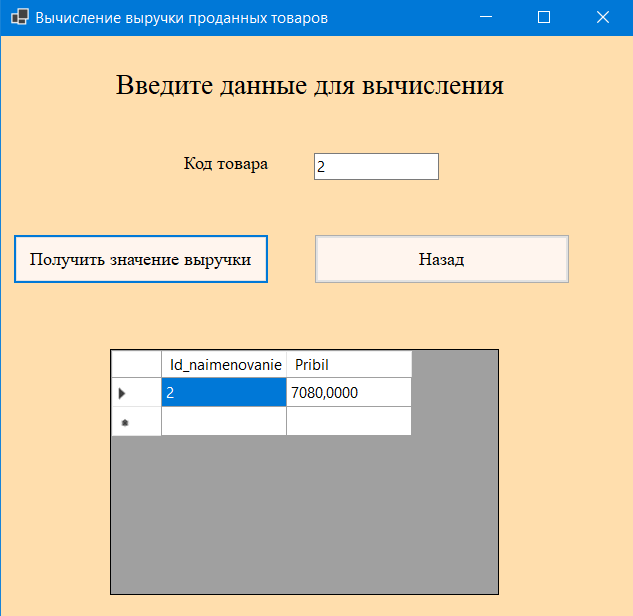


Рисунок 2.39 — Вычисление выручки от проданных товаров

### Выходные данные

Выходными данными в программе являются:

* + занесение зарегистрированных пользователей в таблицу «Пользователь» при выполнении соответствующих требований;
  + осуществление добавления\удаления\обновления данных при работе с таблицами;
  + расчет суммы выручки от определенного товара;
  + определение количества товара на складе;
  + получение кода менеджера;
  + вывод даты создания расходной накладной;
  + вывод всей информации о поставщике;
  + формирование выходного документа
  + расчет доходов и расходов за определенный период.

## Вывод по разделу

В данном разделе были осуществлены следующие задачи:

1. Разработана архитектура программной системы.
2. Создана структура данных.
3. Сконструирован пользовательский интерфейс.
4. Разработаны схемы алгоритма программы и подпрограмм.
5. Программа отлажена. Выполнено успешное тестирование в нормальных и экстремальных условиях, исключительных ситуациях.
6. Создано руководство пользователя.

# Охрана труда

## Техника безопасности при работе на персональном компьютере

Техника безопасности при работе с компьютером на предприятии предусматривает наличие общедоступной инструкции, в которой указаны обязательные требования к обустройству рабочего места и процессу использования техники. Эти правила едины для всех организаций, их выполнение контролируется руководящими органами. Из-за возможности продолжительной работы ПК без отключения от электросети следует уделить особое внимание качеству организации электропитания.

Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать санитарные правила и нормы, гигиенические требования к персональным электронно-вычислительным машинам и организации работы.

С целью уменьшения отрицательного влияния монотонности необходимо применять чередование операций.

Персональный компьютер – электроприбор. От прочих электроприборов он отличается тем, что для него предусмотрена возможность длительной эксплуатации без отключения от электрической сети. В связи с возможностью продолжительной работы компьютера без отключения от электросети следует уделить особое внимание качеству организации электропитания.

Для исключения поражения электрическим током запрещается:

1. работать на ПК и периферийном оборудовании мокрыми руками;
2. работать на ПК и периферийном оборудовании, имеющих нарушения целостности конструкции, изоляции проводов, а также имеющим признаки электрического напряжения на корпусе;
3. часто включать и выключать ПК без необходимости. Перед началом работы с ПК работник обязан:
4. проветрить рабочее помещение;
5. проверить:
   * устойчивость положения оборудования на рабочем столе;
   * отсутствие видимых повреждений оборудования;
   * исправность мебели;
   * исправность и целостность питающих и соединительных кабелей, разъемов, защитного заземления.
6. отрегулировать:
   * положения стула, стола, экрана, клавиатуры;
   * освещенность на рабочем месте.

Запрещается приступать к работе при:

1. выраженном дрожании изображения на мониторе;
2. обнаружении неисправности оборудования;
3. наличии поврежденных кабелей или проводов, разъемов, штепсельных соединений;
4. отсутствии или неисправности защитного заземления оборудования. Во время работы с ПК работник обязан:
5. содержать в порядке и чистоте свое рабочее место;
6. соблюдать оптимальное расстояние от экрана монитора до глаз;
7. соблюдать требования охраны труда.

Работу за экраном монитора следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья или заменять другой работой с целью сокращения рабочей нагрузки у экрана.

Запрещается очищать оборудование от пыли и различных загрязнений, когда оно находится под напряжением.

Недопустимо под напряжением проводить ремонт средств вычислительной техники. Ремонт аппаратуры производится только специалистами с соблюдением необходимых требований.

Не следует оставлять оборудование включенным без наблюдения. При необходимости прекращения на некоторое время работы корректно закрываются все активные задачи и оборудование выключается.

По окончании работы работник обязан:

1. корректно закрыть все активные окна и процессы;
2. выключить питание системного блока;
3. выключить питание всех периферийных устройств;
4. отключить блок бесперебойного питания;
5. отключить питающий кабель от сети;
6. осмотреть и привести в порядок рабочее место.

О неисправности оборудования и других замечаний по работе с ПК сообщить непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования.

## Требования к помещению (машинного зала, ВЦ)

Помещения ВЦ, их размеры (площадь, объем) должны в первую очередь соответствовать количеству работающих и размещаемому в них комплекту технических средств. Должны быть предусмотрены соответствующие параметры температуры, освещения, чистоты воздуха, производственных шумов и др.

Помещения с постоянным пребыванием людей должны иметь естественное освещение. Освещение помещений должно обеспечивать зрительный комфорт, предупреждать появление зрительного и общего утомления.

Снижение шума, создаваемого на рабочих местах ВЦ внутренними источниками, а также шума, проникающего извне, является очень важной задачей. Снижение шума в источнике излучения можно обеспечить применением упругих прокладок между основанием машины, прибора и опорной поверхностью.

Рациональная планировка помещения, размещения оборудования в ВЦ является важным фактором, позволяющим снизить шум при существующем оборудовании ЭВМ.

При планировке ВЦ машинный зал и помещение для сервисной аппаратуры необходимо располагать в дали от шумящего и вибрирующего оборудования. Снижение уровня шума, проникающего в производственное помещение извне, может быть достигнуто увеличением звукоизоляции ограждающих конструкций, уплотнением по периметру притворов окон, дверей.

Кондиционирование воздуха должно обеспечивать автоматическое поддержание параметров микроклимата в необходимых пределах в течение всех сезонов года, очистку воздуха от пыли и вредных веществ, создание небольшого избыточного давления в чистых помещениях для исключения поступления неочищенного воздуха. Температура воздуха, подаваемого в помещения ВЦ, должна быть не ниже 19 градусов.

## Мероприятия по противопожарной охране

При эксплуатации ПК возможны возникновения следующих аварийных ситуаций:

1. короткие замыкания;
2. перегрузки;
3. повышение переходные сопротивлений в электрических контактах;
4. перенапряжение;
5. возникновение токов утечки.

При возникновении аварийных ситуаций происходит резкое выделение тепловой энергии, которая может явиться причиной возникновения пожара.

Для организации пожарной безопасности можно выделить следующие мероприятия:

1. обеспечить безопасность людей и разработать инструкцию по противопожарной технике для каждого помещения;
2. следует допускать работников к работе только после инструктажа;
3. регламентировать порядок проведения противопожарных работ, осмотра помещений, действий коллектива при возникновении пожара;
4. проводить учебную эвакуацию не реже, чем раз в полгода;
5. при ремонтно-профилактических работах строго соблюдать правила пожарной безопасности;
6. каждое из помещений, где производится эксплуатацию ПК, должно быть оборудовано первичными средствами пожаротушения и обеспечено инструкциями по их применению.

Данные мероприятия разработаны для обеспечения безопасности сотрудников предприятия. Соблюдая их, можно сократить риск возникновения пожара до минимума.

# Технико-экономическая часть

## Технико-экономические расчеты экономической целесообразности программы

Задача состоит в разработке системы учета деятельности торгового предприятия.

Программа написана для повышения быстродействия и качества обслуживания клиентов организации, упрощения расчетов и доступа к данным с помощью использования графического интерфейса. В данной программе предусмотрены подсказки и защита от ошибок.

Для написания программы использовался язык программирования С#.

Программа должна быть разработана для компьютеров со следующей конфигурацией:

* процессор от 2,0 ГГц;
* ОЗУ от 10 МБ;
* свободное место на жестком диске 100 МБ;

Для того, чтобы говорить об экономической эффективности разработанного проекта, необходимо рассчитать его себестоимость.

Себестоимость – это затраты предприятий, связанные с производством и реализацией продукции. Калькуляция себестоимости выпускаемой продукции включает в себя затраты на сырье и материалы, оплату труда, страховые взносы, амортизацию основных фондов и прочие расходы. Расчет себестоимости показан на формуле (4.1).

С = М + КИ(ПФ) + Тр−з + Зосн + Здоп + Св + Нцех + Нзав + Впр, (4.1)

где С – себестоимость, руб.;

М – стоимость материалов, основы для полуфабрикатов, разделенных на вспомогательные (не входящие в состав готовой продукции) и основные, руб.;

КИ(ПФ) – стоимость комплектующих материалов, готовых элементов конечного продукта, используемых для сборки, ремонта и упаковки, руб.;

Тр-з – транспортно-заготовительные расходы, связанные с заготовкой, доставкой и хранением материальных ценностей (товаров, сырья, материалов, инструментов), руб.;

Зосн – заработная плата основная основных производственных рабочих, определяется в зависимости от трудоемкости, сложности и действующей формы оплаты труда, руб.;

Здоп – заработная плата дополнительная основных производственных рабочих, расходуется на оплату отпусков, отдельных видов премирования, пособия для уходящим в армию, выплат государственных и общественных обязанностей, руб.;

Св – страховые взносы основных производственных рабочих, расходуется на пенсии и медицину, руб.;

Нцех – накладные расходы цеховые, расходуются на амортизацию оборудования цеха, заработную плату всех работников цеха (исключая основных производственных рабочих), ЖКХ, текущий и капитальный ремонт помещений цеха, руб.;

Нзав – накладные расходы завода, расходуются на амортизацию общезаводского оборудования, заработную плату всех работников завода (исключая работников цехов), ЖКХ, текущий и капитальный ремонт здания завода, аренду, отчисления вышестоящим организациям, руб.;

Впр – внепроизводственные расходы, расходуются рекламу, упаковку, командировки, презентации новых разработок, руб.

Ввиду специфики работы программиста и отсутствия материальной части формула приобретает вид (4.2).

С = Зосн + Здоп + Св + Нцех + Нзав + Впр, (4.2)

где Зосн – заработная плата основная программистов, определяется в зависимости от трудоемкости задачи и оклада, руб.;

Здоп – заработная плата дополнительная программистов, расходуется на оплату отпусков, отдельных видов премирования, пособия для уходящим в армию, выплат государственных и общественных обязанностей, руб.;

Св – страховые взносы программистов, расходуется на пенсии и медицину, руб.;

Нцех – накладные расходы отдела, расходуются на амортизацию компьютеров и другого оборудования, заработную плату всех работников отдела (исключая программистов), ЖКХ, текущий и капитальный ремонт помещений отдела, руб.;

Нзав – накладные расходы фирмы, расходуются на амортизацию общефирменного оборудования, заработную плату всех работников фирмы (исключая работников отделов),

ЖКХ, текущий и капитальный ремонт здания фирмы, аренду, отчисления вышестоящим организациям, руб.;

Впр – внепроизводственные расходы, расходуются на антивирусные программы, оплату интернета, покупку нового оборудования, сопровождение программы во время ее выполнения, руб.

В связи с высокой стоимостью оборудования для программиста, необходимостью быстрой амортизации и большим объемом потребляемой электроэнергии формула приобретает вид (4.3).

С = Зосн + Здоп + Св + А + Э + Нцех + Нзав + Впр, (4.3)

где А – сумма амортизации оборудования, руб.; Э – плата за потребленную электроэнергию, руб.

## Расчетная часть

### Расчет трудоемкости разработанной программы

Для определения себестоимости решения задачи необходимо, прежде всего, найти трудоемкость решения задачи.

Трудоемкость – это сумма затрат труда (по времени), необходимых для изготовления единицы продукции, которая предназначена для решения разработки программы в соответствии с содержанием задания.

Трудоемкость рассматривается как сумма затрат времени на разных этапах решения задачи. Общая трудоемкость рассчитывается по формуле (4.4).

То = Ти + Та + Тбс + Тп + Тот + Тд + Тмр + Тэвм, (4.4)

где Tо – трудоемкость общая, час;

Ти – затраты труда на изучение материала, описание задачи, час; Та – затраты труда на разработку алгоритмов решения задачи, час;

Тбс – затраты труда на разработку блок-схем алгоритма программы, час; Тп – затраты труда на программирование, час;

Тот – затраты труда на отладку программы, час; Тд – затраты на оформление документации, час;

Тмр – затраты труда на машинно-ручные работы, час; Тэвм – время машинного счета на ЭВМ, час.

Слагаемые трудоемкости определяются через количество программных команд данной стадии разработки.

Затраты труда на изучение и описание задачи определяются по формуле (4.5).

Т = 𝑄

и

𝐵×Ккв

× 𝛽, (4.5)

где Q – предполагаемое число форм данной стадии разработки;

β – коэффициент, учитывающий качество описания задачи, равный 1,3 по данным предприятия;

В – скорость программиста, количество команд в час;

Ккв – коэффициент квалификации исполнителя, равный 0,8 по данным предприятия.

#### Расчет трудоемкости на этапе изучения и описания задачи

Число форм данной программы равно 22. По формуле (4.5) трудоемкость изучения материала и описания задачи.

B =1,48;

β = 1,3;

Ккв = 0,8.

22

Ти = 1,48 × 0,8 × 1,3 = 24,16

#### Расчет затрат труда на этапе разработки алгоритма решения задачи

Величина Ta находится по формуле, идентичной Tи. B =0,44;

β = 1,3;

Ккв = 0,8.

22

Та = 0,44 × 0,8 × 1,3 = 81,25

#### Расчет затрат труда на этапе разработки схем алгоритма программы

Величина Tбс находится по формуле, идентичной Tи. B =0,26;

β = 1,3;

Ккв = 0,8.

22

Тбс = 0,26 × 0,8 × 1,3 = 137,5

**Расчет затрат труда на этапе программирования** Величина Tп находится по формуле, идентичной Ти. B =0,44;

β = 1,3;

Ккв = 0,8.

22

Тбс = 0,44 × 0,8 × 1,3 = 81,25

#### Расчет затрат труда на этапе отладки

Величина Tо находится по формуле, идентичной Tи. B =0,087;

β = 1,3;

Ккв = 0,8.

22

Тбс = 0,087 × 0,8 × 1,3 = 410,92

**Расчет затрат труда на этапе документации** Величина Tд находится по формуле, идентичной Tи. B =0,35;

β = 1,3;

Ккв = 0,8.

22

Тбс = 0,35 × 0,8 × 1,3 = 102,14

(4.6).

#### Расчет затрат труда на этапе машинно-ручных операций

Трудоемкость на этапе машинно-ручных операций рассчитывается по формуле

Тм−р

= 𝑡 3600

, (4.6)

где t – время ввода информации, сек.

Время ввода информации находится по формуле (4.7).

𝑡 = С×1,5

, (4.7)

4

где С – объем программы в символах.

По формуле (4.7) необходимо найти время ввода информации при С = 70 014.

𝑡 =

70014 × 1,5

4

= 26255,25

Зная время ввода информации, можно найти трудоемкость на этапе машинно- ручных операций по формуле (4.6).

Тм−р =

26255,25

= 7,29

3600

#### Расчет затрат труда на этапе машинного времени

Время машинного счета на ЭВМ вычисляется по формуле (4.8).

Тэвм = 𝑡вв + 𝑡выв + 𝑡сч, (4.8)

где tвв – время ввода, час; tвыв – время вывода, час; tсч – время счета, час.

По статистическим данным Тэвм = 0,02 часа.

#### Расчет общей трудоемкости

Зная все затраты труда на всех этапах решения задачи, можно вычислить общую трудоемкость решаемой задачи по формуле (4.4).

То = 24,16 + 81,25 + 137,5 + 81,25 + 410,92 + 102,14 + 7,29 + 0,02 = 844,53

Трудоемкость решения задачи на ЭВМ 844,53 часа.

### Расчет себестоимости разработанной программы

Для нахождения себестоимости разработки программы при решении задачи на ЭВМ необходимо использовать формулу (4.9).

С = Зосн + Здоп + Св + А + Э + Нцех + Нзав + Впр (4.9)

#### Расчет основной заработной платы

Заработная плата – это часть национального дохода, переданного в личное распоряжение работника в соответствии с количеством и качеством затраченного им труда. Заработная плата состоит из основной, дополнительной заработной платы и страховых взносов.

Основная заработная плата зависит от степени квалификации работника, уровня сложности, количества и качества выполняемой работы, а также условий, в которых выполняется работа и определяется по формуле (4.10).

Зосн = Сч × То, (4.10)

где Сч – часовая тарифная ставка, руб.;

То – общая трудоемкость решения задачи, час

Часовая тарифная ставка определяется по формуле (4.11).

С = Оклад , (4.11)

ч

22,8×8

где оклад – месячный штатный оклад программиста по данным предприятия, равен 60 000 рублей;

22,8 – среднее количество рабочих дней в месяц; 8 – количество рабочих часов в смену.

Найдем часовую тарифную ставку по формуле (4.11).

(4.10).

60000

Сч = 22,8 × 8 = 328,94

Для нахождения основной заработной платы необходимо использовать формулу

Зосн = 328,94 × 844,53 = 277 799,7

#### Расчет дополнительной заработной платы

Далее, необходимо найти дополнительную заработную плату по формуле (4.12), которая равна 80% от основной заработной платы по данным предприятия.

Зосн = Здоп × 80%, (4.12)

Здоп = 277 799,7 × 0,8 = 222 239,76

#### Расчет страховых взносов

Зная размер основной и заработной платы, можно вычислить сумму страховых взносов, которая составляет 30,2% от размера фонда оплаты труда (ФОТ) и является обязательным государственным налогом. ФОТ рассчитывается по формуле (4.13).

ФОТ = Зосн + Здоп, (4.13)

ФОТ = 277 799,7 + 222 239,76 = 450 039,46

Зная ФОТ можно рассчитать страховые взносы по формуле (3.14).

Св = ФОТ × 30,2%, (4.14)

Св = 450 039,46 × 0,302 = 135 911,92

#### Расчет амортизационных отчислений

Амортизация — это ежемесячное денежное отчисление для возмещения износа оборудования. Амортизация рассматриваемой программы вычисляется в три этапа. Сначала рассчитывается годовая амортизация по формуле (4.15).

Агод

= Сперв, (4.15)

4

где Сперв – первоначальная себестоимость оборудования, использованного для разработки программы, по данным предприятия 96 000 руб;

4– срок окупаемости оборудования, год.

Агод =

96 000

= 24 000 руб.

4

Далее рассчитывается амортизация дневная по формуле (4.16).

Адня

= Агод, (4.16)

247

где 247 – количество рабочих дней в 2022 году.

24 000

Адня =

= 97,17 руб.

247

Чтобы найти амортизацию программы необходимо вычислить амортизацию часовую, исходя из формулы (4.17).

Ачас

= Адня, (4.17)

8

где 8 – количество рабочих часов в смену.

Ачас =

97,17

= 12,15 руб.

8

Итого на каждый час работы программиста приходится 20,8755 рубля амортизации.

Вычислим амортизацию рассматриваемой программы по формуле (4.18).

Апрог = Ачас × То

Апрог = 12,15 × 844,53 = 10 261,04 руб.

#### Расчет затрат на электроэнергию

Затраты на электроэнергию рассчитываются по формуле (4.19).

Э = Тариф × То, (4.19)

где Тариф – действующий тариф на электроэнергию в Москве, по данным предприятия 6,17руб/кВт-час.

Э = 6,17 × 844,53 = 5 210,75

#### Расчет накладных цеховых расходов

Накладные расходы отдела определяются в процентном отношении от основной зарплаты, составляют 200% по данным предприятия. В состав накладных расходов отдела включаются такие затраты как заработная плата аппарата управления отдела (начальника отдела, заместителя начальника, системного администратора), амортизационные отчисления на текущий и капитальный ремонт заданий и оборудования, сооружений, на охрану труда в данном отделе и на непроизводительные затраты. Рассчитывается по формуле (4.20).

Нцех = Зосн × 200%

Нцех = 277 799,7 × 2,00 = 555 599,4

#### Расчет накладных цеховых расходов

Себестоимость отдела – это показатель затратности производства, его анализ позволяет выявить фактор отклонения от процентного объема расходов и сравнить эффективность работы разных отделов. Рассчитывается по формуле (3.21).

Сцех = Зосн + Здоп + Св + А + Э + Нцех (4.21)

Сцех = 277 799,7 + 222 239,76 + 135 911,92 + 10 261,04 + 5 210,75 +

+555 599,4 = 1 207 022,57

#### Расчет накладных цеховых расходов

Накладные расходы фирмы определяются в процентном отношении от основной зарплаты и составляют 100% по данным предприятия. Накладные расходы фирмы – это расходы по управлению фирмой, содержание общефирменного персонала с отчислением на страховые взносы, расходы по командировкам, амортизационные отчисления общефирменного оборудования, на текущий и капитальный ремонт зданий, отчисление вышестоящим организациям. Рассчитываются по формуле (4.22).

Нзав = Зосн × 100%, (4.22)

Нзав = 277 799,7 × 1,00 = 277 799,7

#### Расчет накладных цеховых расходов

Производственная себестоимость определяется путем суммирования общезаводских и целевых расходов с себестоимостью цеховой. Она включает производственные затраты всех отделов, занятых производством продукции или услуг, и расходы по общему управлению предприятием. Себестоимость производственная рассчитывается по формуле (4.23).

Спр = Сцех + Нзав, (4.23)

Спр = 1 207 022,57 + 277 799,7 = 1 484 822,27

#### Расчет внепроизводственных расходов

Внепроизводственные расходы являются неотъемлемой частью затрат предприятия и могут включать в себя рекламу, сопровождение программы во время ее исполнения на предприятии, оплату интернета, первоначальную настройку оборудования и сетей, поддержание работоспособности системы ЭВМ. По данным предприятия они составляют 10% от стоимости заводской и рассчитываются по формуле (4.24).

Впр = Спр × 10%, (4.24)

Впр = 1 484 822,27 × 0,10 = 148 482,227

#### Расчет полной себестоимости решения задачи

Таким образом, зная все необходимые величины, можно вычислить полную себестоимость. Она включает затраты организации не только на выпуск продукции и организацию производственного процесса, но и на ее реализацию, иначе – на ее поставку

на рынок конечного товара и услуг. Она учитывается при формировании цены реализации этой продукции и служит показателем для определения суммы прибыли, получаемой от ее продажи. Вычисляется по формуле (4.25).

Сп = Спр + Впр, (4.25)

Сп = 1 484 822,27 + 148 482,227 = 1 633 304,5

Себестоимость решения задачи на ЭВМ равна 1 633 304,5 руб.

### Анализ возможных путей снижения себестоимости

Данная программа является трудоемкой, так как затраты, связанные с составлением и реализацией программы высоки.

Прямые расходы составляют 51,44% (22,47% + 17,98% + 10,99%). Снижение себестоимости возможно путем:

* увеличения производительности труда за счет повышения квалификации разработчика программы;
* изменения текущего штатного расписания.

По косвенным расходам наибольший удельный вес себестоимости полной занимают накладные цеховые = 44,94%. Снизить себестоимость в данном случае можно при помощи:

* перевода сотрудников на удаленную работу, что позволит уменьшить затраты на электроэнергию;
* аренды более выгодного помещения для организации рабочего пространства сотрудников.

## Графическая часть

В таблице 4.1 представлена производительность труда программиста.

Таблица 4.1 – Производительность труда программиста

|  |  |
| --- | --- |
| **Характер работы** | **Производительность, количество форм/час** |
| Изучение описания задачи | 1,48 |
| Разработка алгоритмов решения | 0,44 |
| Разработка схем алгоритма | 0,26 |
| Программирование по готовой схеме алгоритма с использованием алгоритмического языка | 0,44 |
| Автономная отладка программы | 0,087 |
| Оформление документации | 0,35 |

В таблице 4.2 представлена трудоемкость при решении задачи на ЭВМ.

Таблица 4.2 – Трудоемкость при решении задачи на ЭВМ

|  |  |  |
| --- | --- | --- |
| **Наименование затрат** | **Единица измерения** | **Трудоемкость в часах** |
| Трудоемкость решения задачи на ЭВМ | час | 844,53 |
| В том числе: | | |
| Время изучения описания задачи | час | 24,16 |
| Время на разработку алгоритма задачи | час | 81,25 |
| Время на разработку схемы алгоритма | час | 137,50 |
| Время на программирование | час | 81,25 |
| Время на отладку программы | час | 410,92 |

Продолжение таблицы 4.2

|  |  |  |
| --- | --- | --- |
| Время на оформление документации | час | 102,14 |
| Время на машинно-ручные работы | час | 7,29 |
| Время машинного счета | час | 0,02 |

В таблице 4.3 представлена калькуляция разработки программы.

Таблица 4.3 – Таблица калькуляции

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование статьи калькуляции** | **Сумма в руб** | **Процент к итогу** | **Процент к основной зарплате** |
| Прямые затраты: | | | |
| Основная заработная плата | 277 799,70 | 22,47 | 100,00 |
| Дополнительная заработная плата | 222 239,76 | 17,98 | 80,00 |
| Страховые взносы | 135 911,92 | 10,99 | 48,92 |
| Амортизация оборудования | 10 261,04 | 0,83 | 3,69 |
| Затраты на электроэнергию | 5 210,75 | 0,42 | 1,88 |
| Косвенные затраты: | | | |
| Накладные цеховые расходы | 555 599,40 | 44,94 | 200,00 |
| Себестоимость цеховая | 1 207 022,57 | 97,64 | 434,49 |
| Накладные заводские расходы | 277 799,70 | 22,47 | 100,00 |
| Себестоимость производственная | 1 484 822,27 | 120,11 | 534,49 |
| Внепроизводственные расходы | 148 482,22 | 148 482,23 | 53,45 |
| Себестоимость полная | 1 633 304,50 | 100,00 | 587,94 |

В таблице 4.4 представлены технико-экономические показатели.

Таблица 4.4 – Технико-экономические показатели

|  |  |  |  |
| --- | --- | --- | --- |
| **Технико- экономический показатель** | **Единица измерения** | **Формула** | **Результат** |
| Трудоёмкость решения задачи на ЭВМ | час | То = Ти + Та + Тбс +  +Тп + Тот + Тд + Тмр +  +Тэвм | 833,53 |
| Себестоимость решения задачи на ЭВМ | руб. | С = Зосн + Здоп + Св + А +  +Э + Нцех + +Нзав + Впр | 1 633 304,50 |

# ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы была разработана информационная система учета деятельности торгового предприятия. В результате выполнения работы все требования были выполнены. Итоги выполненной работы:

1. Создана база данных, с использованием наиболее подходящей к выданному заданию платформу (Microsoft SQL Server).
2. Создано приложение, с использованием наиболее приемлемой для решения задачи платформы: .NET.
3. Реализована возможность авторизации зарегистрированных пользователей.
4. В системе, в зависимости от предметной области выделены 2 типа пользователей: бухгалтер и менеджер.
5. Осуществлена проверка пароля заданным условиям для входа в систему.
6. Созданы основные формы для работы с сущностями, реализованными в базе данных.
7. Реализована автоматизация 2 бизнес-процессов для каждого из видов пользователей и необходимых для этого форм: вычисление выручки от проданных товаров и показ количества товара на складе.
8. Реализована форма вывода отчета с возможностью печати документа.

При разработке программы использовался высокоуровневый язык программирования C#.

Программа хорошо структурирована, в листинге приведены удобные комментарии, что позволяет быстро понять код программы для дальнейшей разработки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и защите информации» (с изменениями и дополнениями от 27.07.2010, 06.04.2011, 21.07.2011, 28.07.2012, 05.04.2013, 07.06.2013, 02.07.2013, 28.12.2013,

05.05.2014).

1. Бурцева Е.В. Оценка экономической эффективности информационной системы- Компьютер Пресс, 2011.
2. Грофф Джеймс Р., Вайнберг Пол Н. SQL: Полное руководство- Вильямс, 2015.
3. Черников В.Н. Разработка мобильных приложений на C# для IOS и Android- ДМК Пресс, 2020.
4. Кондрашов Ю.Н. Анализ данных и машинное обучение на платформе MS SQLServer- Кнорус, 2019.
5. Кригель, А. SQL. Библия пользователя / А. Кригель. - М.: Диалектика / Вильямс,2016.
6. Матрин Р. Чистый код - Питер, 2012.
7. Макконелл С. Совершенный код- БХВ-Петербург, 2017.
8. Моргунов Е.П. Язык SQL. Базовый курс- Postgres Professional, 2017.
9. Плаксин М. Тестирование и отладка программ для профессионалов будущих инастоящих- Лаборатория знаний, 2020.
10. Советов Б.Я., Цехановский В.В., Чертовской В.Д. Базы данных. Теория и практика– М.: Юрайт, 2013.
11. Трисуб Н.А. Базы данных – МИСиС, 2019.
12. Методы в C# //URL: [https://metanit.com](https://metanit.com/) (дата обращения: 31.03.2022)
13. Персональные данные пользователей: как не нарушить закон при разработке приложения //URL: [https://cmsmagazine.ru](https://cmsmagazine.ru/) (дата обращения: 04.04.2022)
14. Информационная безопасность России в условиях глобализации //URL: <https://www.dissercat.com/>(дата обращения: 12.04.2022)

# ПРИЛОЖЕНИЕ А

(обязательное) Листинг программы

# Листинг программы

/\*Выпускная квалификационная работа по теме: «Разработка системы учета деятельности торгового предприятия»

\*Название: МС-Продукт

\*Разработал:

\*Группа: ТИП-81

\*Дата: 11.05.2022 v.1.0.

\*Язык: С#

\*Краткое описание: данная программа является информационной системой учета деятельности торгового предприятия

\*Задание:

* 1. создание базы данных для информационной системы «МС-Продукт»;
  2. создание форм для просмотра, обновления, добавления и удаления данных в базе с помощью использования графического интерфейса;
  3. реализация возможности регистрация новых пользователей в системе и их авторизации при различном доступе к данным;
  4. реализация возможности формирования отчетов по проданным товарам и данным о клиентах для работы менеджеров и отдела бухгалтерии, пользующихся данным программным продуктом;
  5. создание бизнес-процессов для каждого типа пользователя;

\*Используемые формы:

form\_menu- «Главное меню»; form\_vhod- «Вход в систему»; form\_db- «Работа с БД»; form\_client- «Таблица «Клиент»»;

form\_sotrudnic- «Таблица «Сотрудник»»; form\_postavchic- «Таблица «Поставщик»»; form\_tovar- «Таблица «Товар»»;

form\_prih\_nacl- «Таблица «Приходная накладная»»; form\_rash\_nacl- «Таблица «Расходная накладная»»; form\_s\_prih\_nacl- «Таблица «Содержание приходной накладной»»;

form\_s\_rash\_nacl - «Таблица «Содержание расходной накладной»»; form\_pechat- «Формирование и печать документа»;

form\_reg- «Регистрация новых пользователей»; form\_virucka- «Вычисление выручки проданных товаров»; form\_sclad- «Показ количества товаров на складе»;

form\_data\_nacl- «Получение даты создания расходной накладной»; form\_zp- «Таблица «Заработная плата»»;

form\_pribil - «Расчет прибыли предприятия»;

form\_rashod - «Расчет расходов предприятия за конкретный месяц»; form\_otchet\_rashod- «Таблица «Отчет о расходах предприятия»»;

# Форма «Главное меню»

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

namespace Praktica

{

public partial class form\_menu : Form

{

public form\_menu()

{

InitializeComponent();

}

//Вызов формы «Вход в систему»

private void button1\_Click(object sender, EventArgs e)

{

form\_vhod fm = new form\_vhod(); fm.Show();

}

//Вызов формы «Вход в систему» private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

//Вызов формы «Регистрация новых пользователей»

private void button2\_Click(object sender, EventArgs e)

{

form\_reg = new form\_reg(); fm.Show();

}

}

}

# ПРИЛОЖЕНИЕ Б

(обязательное) Результаты выполнения программы

На рисунке Б.1 продемонстрирован интерфейс формы «Вход в систему».

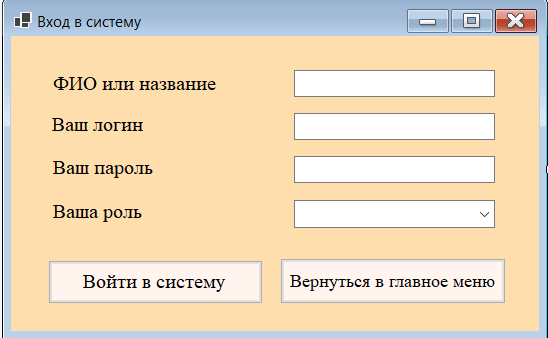


Рисунок Б.1 — Интерфейс формы «Вход в систему»

На рисунке Б.2 представлена форма «Работа с БД».



Рисунок Б.2 — Интерфейс формы «Работа с БД»

На рисунке Б.3 изображена форма «Таблица «Сотрудник»».

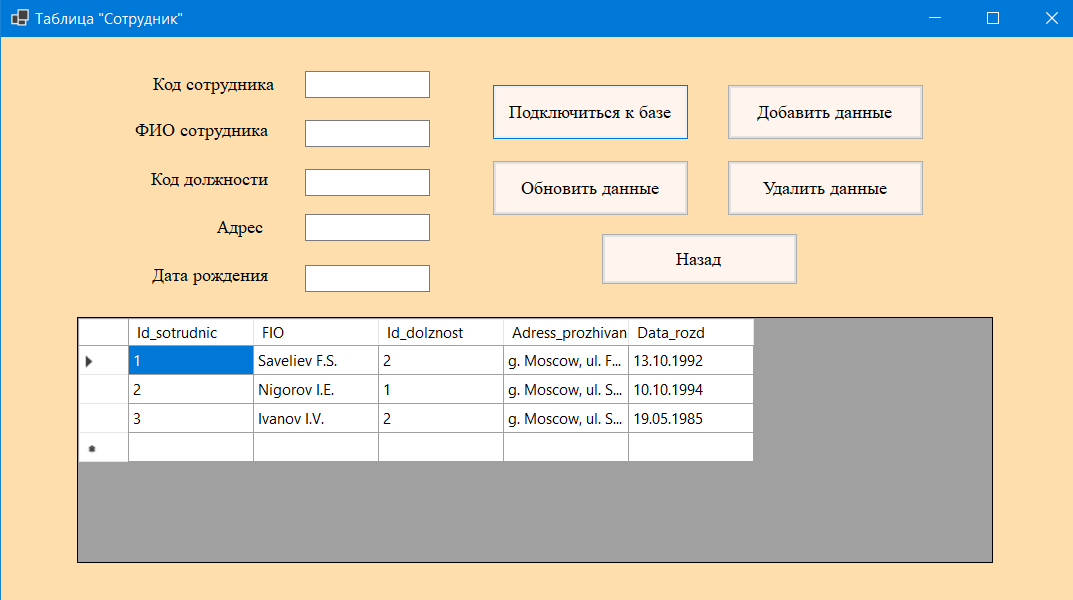


Рисунок Б.3 — Просмотр данных в таблице

На рисунке Б.4 изображено осуществление добавление строки в таблицу на форме

«Таблица «Сотрудник»».

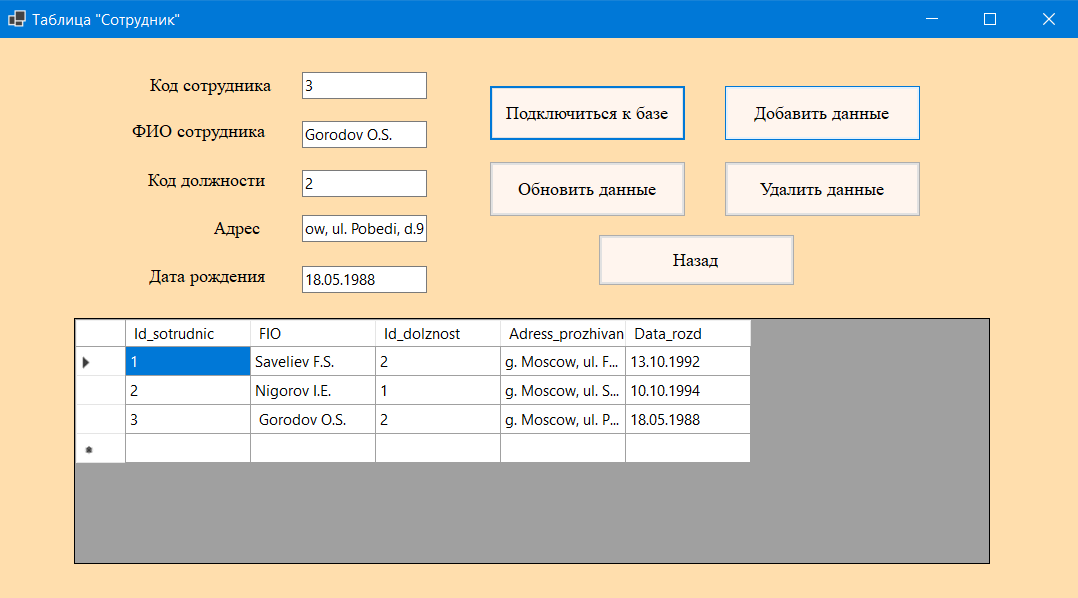


Рисунок Б.4— Просмотр данных в таблице после добавления

На рисунке Б.5 показано разграничение доступа к формам, соответствуя роли, указанной в базе данных.

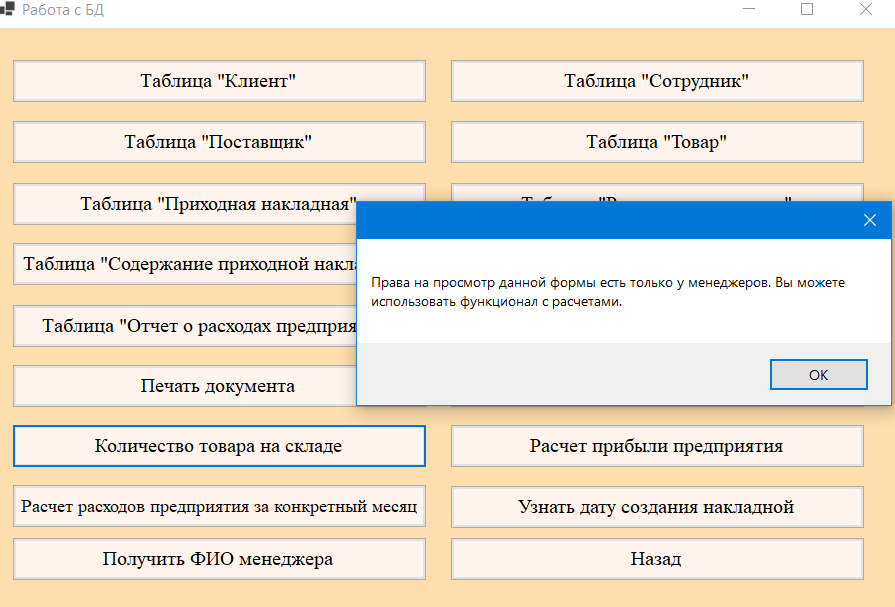


Рисунок Б.5 — Сообщение об отсутствии прав на просмотр формы

На рисунке Б.6 представлена форма «Прибыль от конкретного товара».

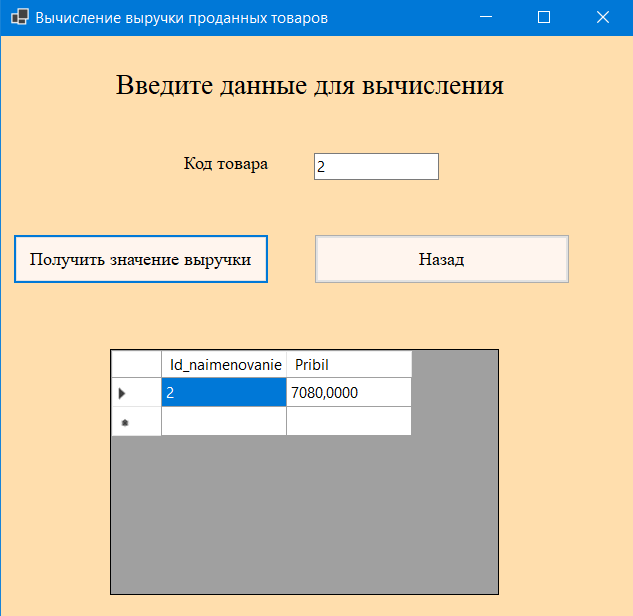


Рисунок Б.5 — Интерфейс формы «Прибыль от конкретного товара»

На рисунке Б.6 представлена форма «Получение даты создания расходной накладной».

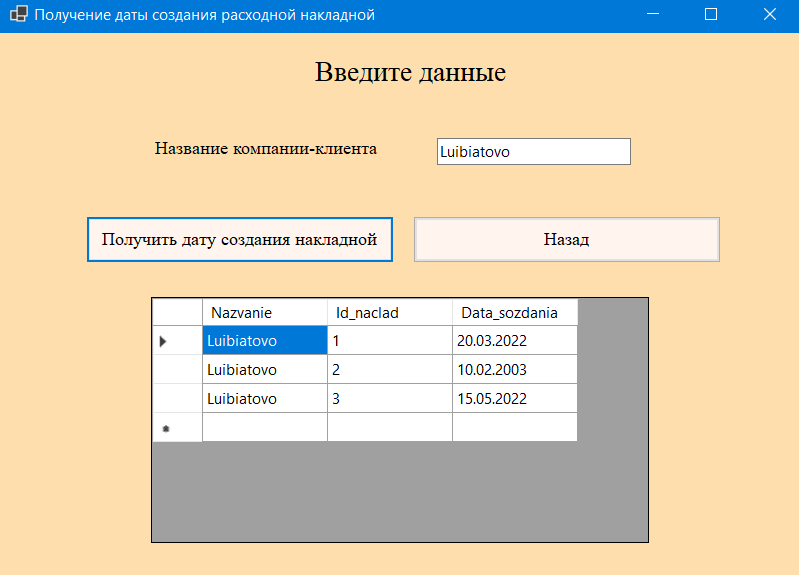


Рисунок Б.6 — Интерфейс формы «Получение даты создания расходной накладной»

На рисунке Б.7 изображена форма «Показ количества товара на складе».

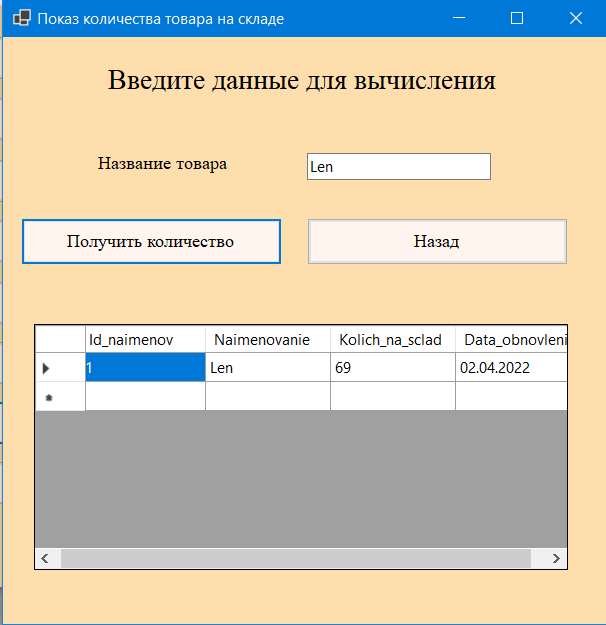


Рисунок Б.7 — Интерфейс формы «Показ количества товара на складе»

На рисунке Б.8 показана форма «Расчет расходов предприятия».

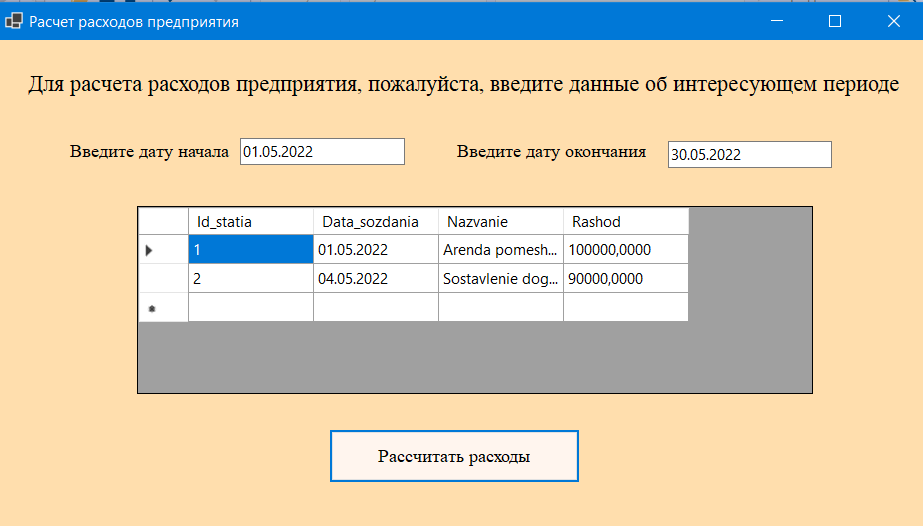


Рисунок Б.8 — Интерфейс формы «Расчет расходов предприятия»

На рисунке Б.9 показана форма «Расчет прибыли предприятия».

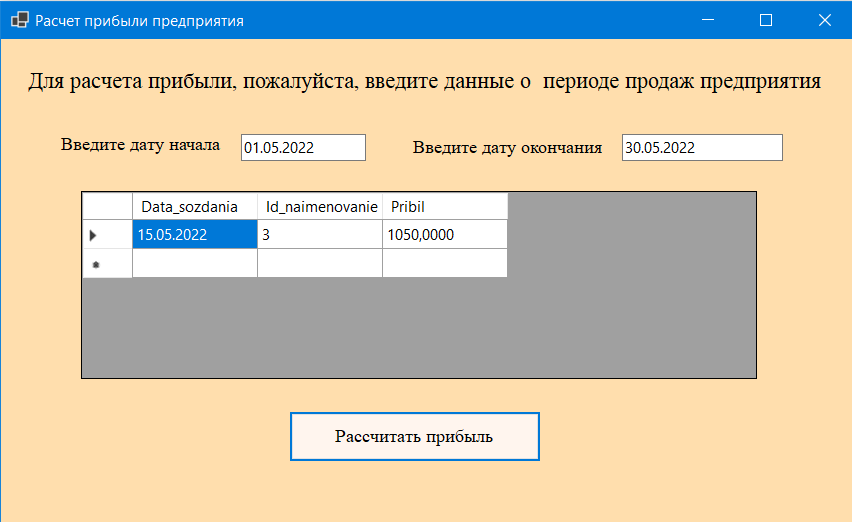


Рисунок Б.9 — Интерфейс формы «Расчет прибыли предприятия»

На рисунке Б.10 показана форма «Формирование и печать документа».

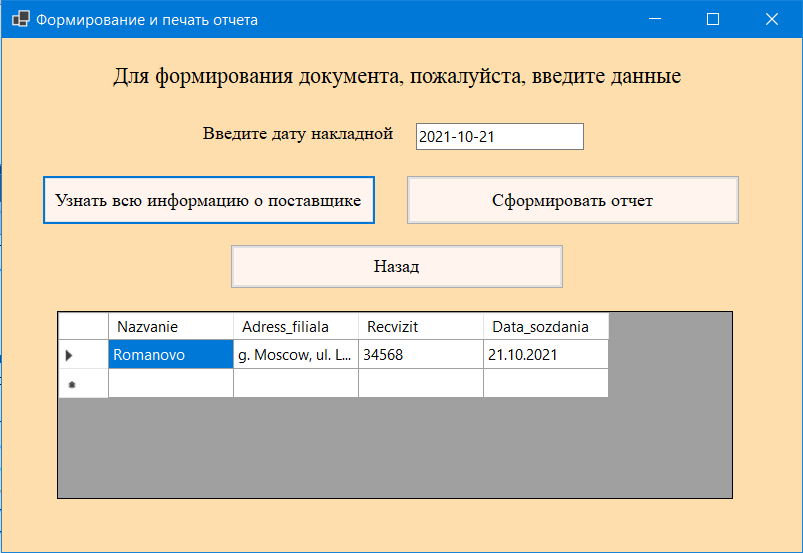


Рисунок Б.10 — Интерфейс формы «Формирование и печать документа»

На рисунке Б.11 представлен пример формирования выходного документа.

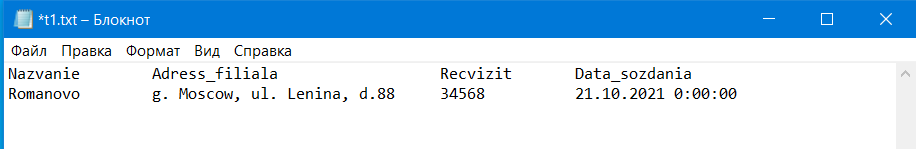


Рисунок Б.11 — Формирование документа