

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной практике

наименование вида и типа практики

на (в) Юго-Западном государственном университете

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-016

курса, группы

Арнаутова Ярослава Владиславовича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка

должность, звание, степень

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка

должность, звание, степень

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск, 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ предметной области	8
1.1 Описание предметной области	8
1.2 Основные принципы удаленного администрирования	9
1.3 Задачи и функции удаленного администратора	10
1.4 Особенности администрирования в корпоративной среде	11
2 Техническое задание	13
2.1 Основание для разработки	13
2.2 Цель и назначение разработки	13
2.3 Требования пользователя к программному продукту	14
2.3.1 Требования к данным программного продукта	14
2.3.2 Функциональные требования к программному продукту	15
2.3.3 Требования пользователя к интерфейсу в программном продукте	16
2.3.4 Варианты использования программного продукта	17
2.3.4.1 Добавление Ip	17
2.3.4.2 Удаление Ip	18
2.3.4.3 Подключение к серверу	18
2.3.4.4 Открытие файловых систем компьютеров	18
2.3.4.5 Получение и просмотр файловой системы клиента	19
2.3.4.6 Получение и просмотр файловой системы сервера	19
2.3.4.7 Отправка файла	19
2.3.4.8 Получение файла	20
2.3.4.9 Открытие консольного управления сервером	20
2.3.4.10 Отправка команды	20
2.3.4.11 Получение ответа	21
2.3.4.12 Отключение от сервера	21
2.4 Нефункциональные требования к программному продукту	21
2.4.1 Требования к надежности	21
2.4.2 Требования к программному обеспечению	21

2.4.3	Требования к оформлению документации	22
3	Технический проект	23
3.1	Общие сведения о программной систем	23
3.2	Обоснование выбора технологии проектирования	23
3.2.1	Описание используемых технологий и языков программирования	24
3.2.1.1	Язык программирования C#	24
3.2.1.2	Протокол TCP/IP	26
3.3	Проектирование архитектуры программной системы	27
3.3.1	Архитектура программной системы	27
3.4	Проектирование пользовательского интерфейса	27
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПО – программное обеспечение.

ТСР – протокол передачи данных.

ПО – программное обеспечение.

РП – рабочий проект.

ТЗ – техническое задание.

ТП – технический проект.

ВВЕДЕНИЕ

В современном мире информационных технологий, где компьютерные сети проникают во все сферы бизнеса и жизни, эффективное администрирование становится неотъемлемой частью успешного функционирования организаций. Однако, с увеличением масштабов бизнеса и географического разнообразия его структур, возникают новые трудности работы IT-специалистов. В этом контексте, удаленное администрирование компьютеров становится не просто дополнительной опцией, а необходимостью.

Удаленное администрирование позволяет IT-администраторам, без физического присутствия, эффективно управлять компьютерами на больших. Данная функция становится крайне важной и необходимой в условиях удаленной работы, где сотрудники могут находиться в разных частях города, страны или даже континента.

Также стоит отметить что время становится одним из самых ценных ресурсов, наравне с техническими и человеческими. В ситуации, когда необходимо провести оперативные действия с компьютером, даже краткая задержка может оказать существенное воздействие на процессы бизнеса. Взаимодействие с данными или настройка системы, осуществляемые IT-администратором, порой вынуждают обычного пользователя останавливать свою работу. Тем самым создавая диссонанс в отточенный механизм корпоративных процессов. В свете этих обстоятельств, удаленное администрирование выступает важным инструментом, призванным не просто обеспечивать функциональность, но и минимизировать воздействие на оперативную деятельность. Воплощаясь в максимально незаметных процессах, оно позволяет пользователям беспрепятственно продолжать свои деловые задачи, не отвлекаясь на технические моменты.

Таким образом разработка приложения удаленного администрирования компьютеров становится актуальной задачей для разработчиков программного обеспечения. Такое приложение должно обеспечивать не только простоту и удобство в использовании, но и надежность, безопасность и

высокую производительность. Кроме того, оно должно быть гибким и масштабируемым, способным адаптироваться к изменяющимся потребностям и условиям бизнеса.

Цель настоящей работы – разработка приложения удаленного администрирования компьютеров организации для улучшения эффективности работы IT-администраторов. Для достижения поставленной цели необходимо решить *следующие задачи*:

- провести анализ предметной области;
- разработать концептуальную модель приложения;
- спроектировать приложение;
- реализовать приложение.

Структура и объем работы. Отчет состоит из введения, 4 разделов основной части, заключения, списка использованных источников, 2 приложений. Текст выпускной квалификационной работы равен 30 страницам.

Во введении сформулирована цель работы, поставлены задачи разработки, описана структура работы, приведено краткое содержание каждого из разделов.

В первом разделе на стадии описания технической характеристики предметной области приводится сбор информации о потребностях IT-администраторов и данные о компьютерах в организации.

Во втором разделе на стадии технического задания приводятся требования к разрабатываемому приложению.

В третьем разделе на стадии технического проектирования представлены проектные решения для приложения.

В четвертом разделе приводится список классов и их методов, использованных при разработке приложения, производится тестирование разработанного приложения.

В заключении излагаются основные результаты работы, полученные в ходе разработки.

В приложении А представлен графический материал. В приложении Б представлены фрагменты исходного кода.

1 Анализ предметной области

1.1 Описание предметной области

Удаленное администрирование представляет собой метод управления информационными системами, компьютерными сетями и серверами без необходимости физического присутствия администратора на месте. Этот подход становится все более востребованным в современном мире, где компьютеры компании расположены на большом удалении друг от друга, а технологии позволяют обеспечить надежное удаленное управление.

Основной целью удаленного администрирования является обеспечение надёжности, доступности и безопасности работы информационных систем и сетей. В современных условиях, когда организации все более активно используют облачные решения и развивают глобальные сети, удаленное администрирование становится необходимым компонентом эффективного функционирования бизнеса.

Удаленное администрирование позволяет оперативно реагировать на возникающие проблемы и инциденты, минимизируя время простоя систем и сокращая потери бизнеса. Благодаря возможности удаленного мониторинга и управления, администраторы могут непрерывно следить за состоянием систем, выявлять угрозы безопасности и проводить профилактические мероприятия, что способствует повышению общей надёжности и стабильности работы информационной инфраструктуры.

Кроме того, удаленное администрирование снижает операционные расходы организации за счёт уменьшения необходимости в дорогостоящем обслуживании и обновлении аппаратного обеспечения, а также сокращения затрат на поездки и проживание администраторов на объектах.

С учётом растущей цифровизации бизнеса и повсеместного использования удаленных рабочих мест, удаленное администрирование становится неотъемлемой составляющей успешного функционирования компаний в современном информационном пространстве.

1.2 Основные принципы удаленного администрирования

В основе удаленного администрирования лежат фундаментальные принципы, на которых строится эффективное управление информационными системами и сетями в удаленном режиме. Они определяют основные правила работы и подходы, которые необходимо учитывать при реализации и использовании удаленных административных решений. Соблюдение данных принципов является важным условием для успешного и безопасного функционирования систем удаленного администрирования. Ниже перечислены основные из них:

Безопасность: Одним из важнейших принципов удаленного администрирования является обеспечение безопасности системы. Это включает в себя использование шифрование данных во время передачи, контроль доступа к ресурсам, а также мониторинг и реагирование на потенциальные угрозы и атаки.

Доступность: Основным принципом удаленного администрирования является обеспечение постоянного доступа к информационным ресурсам независимо от времени и местоположения. Это достигается за счёт высокой надёжности сетевых соединений, резервирования каналов связи и резервного копирования данных.

Производительность: Для эффективного удаленного администрирования необходима высокая производительность средств управления и мониторинга. Это включает в себя оптимизацию сетевых протоколов, использование высокопроизводительного оборудования и программного обеспечения, а также оптимизацию алгоритмов и процессов администрирования.

Автоматизация: Одним из ключевых принципов удаленного администрирования является автоматизация рутинных операций и процессов. Это позволяет сократить время на выполнение задач, уменьшить вероятность ошибок и повысить эффективность работы администраторов.

Гибкость: Принцип гибкости предполагает, что системы удаленного администрирования должны быть гибкими и адаптивными к различным

условиям и требованиям. Это позволяет эффективно реагировать на изменения в бизнес-процессах и технологической инфраструктуре.

1.3 Задачи и функции удаленного администратора

Удаленный администратор играет ключевую роль в обеспечении надёжной и работы систем компании. Его задачи и функции включают в себя:

Управление службами: Администратор может использовать консоль для управления службами, такими как запуск, остановка, приостановка или изменение конфигурации служб.

Управление процессами: Администратор, при необходимости, может просматривать запущенные процессы на компьютере, завершать нежелательные процессы или управлять приоритетами процессов.

Управление пользователями и группами: Администратор может осуществлять управление пользователями и группами, включая создание, удаление, изменение учётных записей и назначение прав доступа.

Настройка и диагностика сети: Администратор выполняет различные сетевые задачи, такие как проверка состояния сетевого подключения, настройка параметров сети, выполнение утилит диагностики сети и много другое.

Управление файловой системой: Также администратор осуществляет работу с файлами и папками, такую как создание, копирование, редактирование, замена, перемещение и удаление файлов.

Мониторинг ресурсов: Администратору необходим доступ к различным утилитами и командам для мониторинга использования ресурсов компьютера, таких как процессор, память, дисковое пространство.

Управление политиками безопасности: Администратор может осуществлять настройку различных политик безопасности, включая управление правами доступа, настройку брандмауэра, а также аудит и журналирование событий безопасности.

1.4 Особенности администрирования в корпоративной среде

Администрирование в корпоративной среде имеет свои уникальные особенности и требования, которые отличаются от администрирования в других типах организаций. Рассмотрим основные из них:

Масштабность: В корпоративной среде часто существует большое количество компьютерных систем, серверов и сетевых устройств, что создает необходимость в масштабируемых решениях для управления всей инфраструктурой.

Системы управления конфигурацией: В корпоративной среде широко используются системы управления конфигурацией (Configuration Management Systems), которые позволяют централизованно управлять конфигурацией компьютерных систем.

Политики безопасности: В корпоративной среде обычно существуют строгие политики безопасности, которые требуют соблюдения определенных стандартов и процедур для обеспечения защиты информации и сетевой инфраструктуры.

Автоматизация и оркестрация: Для эффективного управления корпоративной средой часто применяются инструменты автоматизации и оркестрации, которые позволяют автоматизировать рутинные задачи, координировать работу между различными системами и обеспечивать высокую степень автоматизации.

Централизованный мониторинг: Важным аспектом администрирования в корпоративной среде является централизованный мониторинг состояния систем и сетей, который позволяет оперативно обнаруживать и реагировать на проблемы, а также проводить анализ производительности и использования ресурсов.

Управление доступом и аутентификация: В корпоративной среде особое внимание уделяется управлению доступом и аутентификации пользователей, что включает в себя настройку политик доступа, использование механизмов одноразовых паролей и многофакторной аутентификации.

Бизнес-процессы и требования: При администрировании в корпоративной среде необходимо учитывать специфические бизнес-процессы и требования заказчиков, что требует гибкости и адаптивности в реализации технических решений.

Все эти особенности делают администрирование в корпоративной среде сложным и ответственным процессом, требующим высокой квалификации и профессионализма со стороны администраторов. Однако, правильное управление корпоративной информационной инфраструктурой способствует повышению эффективности бизнеса, обеспечивает защиту от угроз и сбоев, а также снижает операционные риски.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки программно-информационной системы для удаленного администрирования компьютеров организации является задание на выпускную квалификационную работу приказ ректора ЮЗГУ от « » 2024 года № 0000-0 «Об утверждении тем выпускных квалификационных работ и руководителей выпускных квалификационных работ».

2.2 Цель и назначение разработки

Целью является разработка приложение удаленного администрирования компьютеров организации для улучшения эффективности работы IT-администраторов.

Функциональное назначение разрабатываемого приложения заключается в предоставлении IT-администраторам организации удаленного доступа к консоли и файловой системе компьютеров сотрудников компании.

Предполагается, что данной программой будут пользоваться работники IT-отдела организации для облегчения выполнения своих обязанностей и сокращения затрат человека-часов на решения возникших проблем.

Задачами разработки данного приложения являются:

- создание клиентской части приложения;
- создание серверной части приложения;
- реализация клиент-серверного соединения;
- реализация обмена данными.

На рисунке 2.1 представлена диаграмма логики действия приложения



Рисунок 2.1 – Диаграмма логики действия приложения

2.3 Требования пользователя к программному продукту

2.3.1 Требования к данным программного продукта

Входными данными для приложения являются:

- Ip компьютера пользователя;
- команда для выполнения cmd;
- пути к папкам для отправки/получения файла;
- пути к файлам для отправки/получения.

Выходными данными для приложения являются:

- файловая структура клиента;
- файловая структура сервера;
- результат выполнения команды в cmd.

2.3.2 Функциональные требования к программному продукту

На основании анализа предметной области в разрабатываемой программно-информационной системе удаленного администрирования компьютеров организации должны быть реализованы следующие функции:

Для клиентской части программы:

- добавление Ip сервера;
- удаление Ip сервера;
- подключение к серверу;
- открытие файловых систем компьютеров;
- получение и просмотр файловой системы клиента;
- получение и просмотр файловой системы сервера;
- отправка файла на сервер;
- получение файла от сервера;
- открытие консольного управления сервером;
- отправка команды на сервер;
- получение и просмотр ответа от сервера;
- отключение от сервера.

Для серверной части программы:

- ожидание подключения клиента;
- ожидание запроса от клиента;
- выполнение запроса клиента;
- отправка ответа клиенту;

На рисунке 2.2 представлены функциональные требования к системе в виде диаграммы прецедентов.

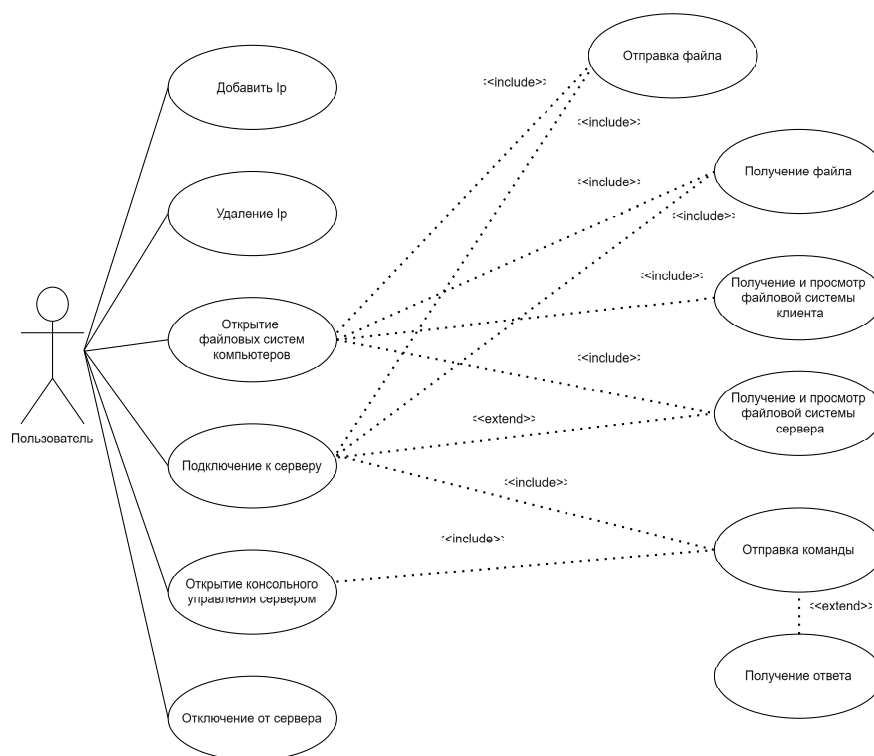


Рисунок 2.2 – Диаграмма прецедентов

2.3.3 Требования пользователя к интерфейсу в программном продукте

Интерфейс программного продукта должен соответствовать следующим требованиям: Для клиентской части программы должны быть реализованы следующие графические элементы:

- Кнопка добавления Ip;
- Кнопка удаления Ip;
- Поле ввода Ip;
- Выпадающий список доступных Ip;
- Кнопка открытие файловых систем компьютеров;
- Поле просмотра файловой системы клиента;
- Поле просмотра файловой системы сервера;
- Кнопка отправки файла на сервер;
- Кнопка получения файла от сервера;
- Кнопка открытие консольного управления сервером;
- Поле ввода команды для отправки на сервер;

- Поле просмотра ответа от сервера;
- Кнопка подключения к серверу;
- Кнопка отключения от сервера.

Серверная часть программы должна представлять из себя службу.

На рисунках 2.3 - 2.4 представлены функциональные требования к системе в виде диаграммы прецедентов.

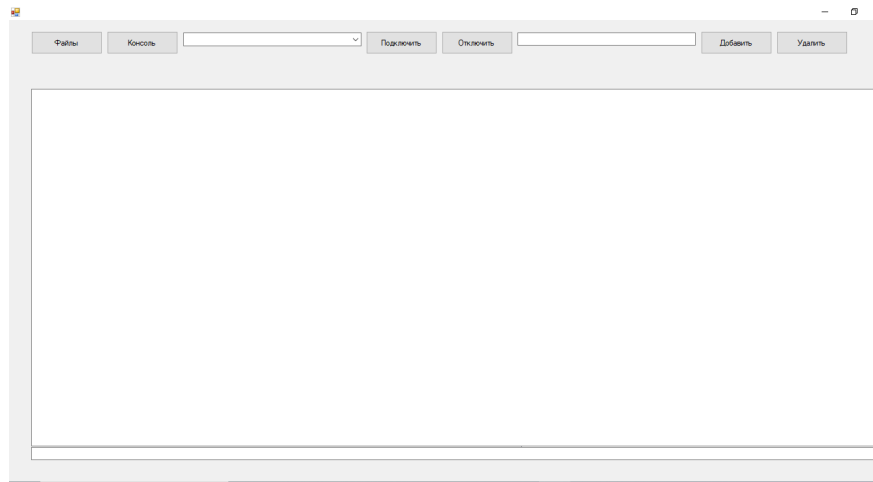


Рисунок 2.3 – Макет окна с консолью

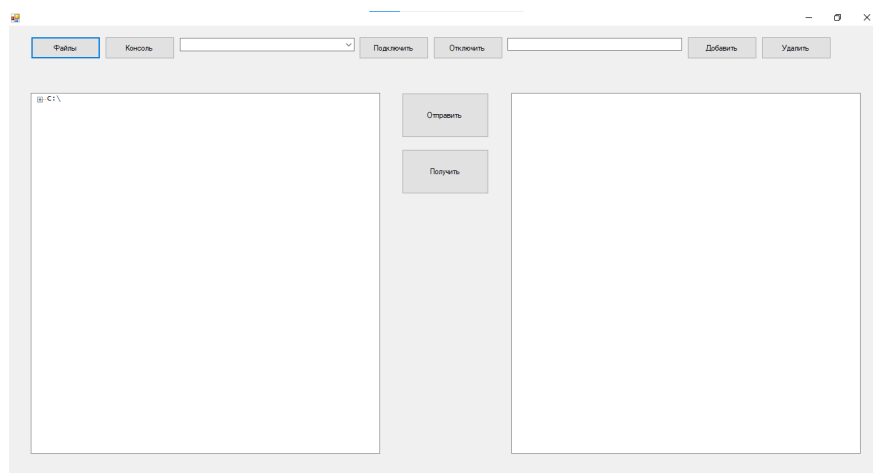


Рисунок 2.4 – Макет окна с файловой системой

2.3.4 Варианты использования программного продукта

2.3.4.1 Добавление Ip

Заинтересованные лица и их требования: IT-администратор, который хочет добавить Ip для дальнейшей возможности подключения к серверу.

Предусловие: Пользователь запускает клиентскую часть приложения. Постусловие: Пользователь добавляет Ip. Основной успешный сценарий:

1. Пользователь вводит Ip в строку.
2. Пользователь нажимает кнопку "Добавить".
3. Ip появляется в выпадающем списке доступных Ip.

2.3.4.2 Удаление Ip

Заинтересованные лица и их требования: IT-администратор, который хочет удалить Ip. Предусловие: Пользователь запускает клиентскую часть приложения, Ip был ранее добавлен. Постусловие: Пользователь удаляет Ip. Основной успешный сценарий:

1. Пользователь вводит Ip в строку.
2. Пользователь нажимает кнопку "Удалить".
3. Ip пропадает из выпадающего списка доступных Ip.

2.3.4.3 Подключение к серверу

Заинтересованные лица и их требования: IT-администратор, который хочет подключиться к серверу. Предусловие: Пользователь запускает клиентскую часть приложения, Ip был ранее добавлен, на сервере запущена серверная часть программы. Постусловие: Пользователь подключается. Основной успешный сценарий:

1. Пользователь выбирает Ip из выпадающего списка.
2. Пользователь нажимает кнопку "Подключить".
3. Система выводит сообщение об успешном подключении.

2.3.4.4 Открытие файловых систем компьютеров

Заинтересованные лица и их требования: IT-администратор, который хочет открыть файловую систему компьютеров. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу. Постусловие: Система отображает поля для просмотра и взаимодействия с файловой системой компьютеров. Основной успешный сценарий:

1. Пользователь нажимает кнопку "Файлы".
2. Система отображает поля для просмотра и взаимодействия с файловой системой компьютеров.

2.3.4.5 Получение и просмотр файловой системы клиента

Заинтересованные лица и их требования: IT-администратор, который хочет посмотреть свою файловую систему. Предусловие: Пользователь запускает клиентскую часть приложения, была нажата кнопка "Файлы". Постусловие: Система отображает файловую систему клиента. Основной успешный сценарий:

1. Система отображает файловую систему клиента.
2. Пользователь может с ней взаимодействовать.

2.3.4.6 Получение и просмотр файловой системы сервера

Заинтересованные лица и их требования: IT-администратор, который хочет посмотреть файловую систему сервера. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу, была нажата кнопка "Файлы". Постусловие: Система отображает файловую систему клиента. Основной успешный сценарий:

1. Система делает запрос на получение файловой системы сервера.
2. Система получает ответ и выводит его в поле.
3. Пользователь может с ней взаимодействовать.

2.3.4.7 Отправка файла

Заинтересованные лица и их требования: IT-администратор, который хочет отправить файл на сервера. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу, была нажата кнопка "Файлы". Постусловие: Система отображает файловую систему клиента. Основной успешный сценарий:

1. Пользователь выбирает файл для отправки.
2. Пользователь выбирает путь для получения.

3. Пользователь нажимает кнопку "Отправить".
4. Система отправляет файл.

2.3.4.8 Получение файла

Заинтересованные лица и их требования: IT-администратор, который хочет получить файл от сервера. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу, была нажата кнопка "Файлы". Постусловие: Система отображает файловую систему клиента. Основной успешный сценарий:

1. Пользователь выбирает файл для получения.
2. Пользователь выбирает путь для сохранения.
3. Пользователь нажимает кнопку "Получить".
4. Система отправляет запрос на получение файла.
5. Система получает файл.

2.3.4.9 Открытие консольного управления сервером

Заинтересованные лица и их требования: IT-администратор, который хочет открыть консольное управление сервером. Предусловие: Пользователь запускает клиентскую часть приложения. Постусловие: Система отображает поля для консольного управления сервером. Основной успешный сценарий:

1. Пользователь нажимает кнопку "Консоль".
2. Система отображает поля для консольного управления сервером.

2.3.4.10 Отправка команды

Заинтересованные лица и их требования: IT-администратор, который хочет отправить команду на сервера. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу, была нажата кнопка "Консоль". Постусловие: Система отправляет команду на сервер. Основной успешный сценарий:

1. Пользователь вводит для отправки.
2. Пользователь нажимает Enter.

3. Система отправляет команду.

2.3.4.11 Получение ответа

Заинтересованные лица и их требования: IT-администратор, который хочет отправить команду на сервера. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу, была нажата кнопка "Консоль была отправлена команда. Постусловие: Система отправляет команду на сервер. Основной успешный сценарий:

1. Серверная часть приложения получает команду.
2. Серверная часть выполняет команду в cmd.
3. Серверная часть отправляет результат выполнения команды.
4. Система получает и выводит ответ от сервера.

2.3.4.12 Отключение от сервера

Заинтересованные лица и их требования: IT-администратор, который хочет отключиться от сервера серверу. Предусловие: Пользователь запускает клиентскую часть приложения, было осуществлено подключение серверу. Постусловие: Пользователь отключается. Основной успешный сценарий:

1. Пользователь нажимает кнопку "Отключить".
2. Система выводит сообщение об успешном отключении.

2.4 Нефункциональные требования к программному продукту

2.4.1 Требования к надежности

В приложении не должно возникать критических ошибок, приводящих к экстренному завершению работы.

2.4.2 Требования к программному обеспечению

Для реализации программной системы должен быть использован язык программирования высокого уровня C# для разработки поведения компонентов программы.

2.4.3 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общие сведения о программной системе

Необходимо спроектировать и разработать программно-информационную систему для удаленного администрирования компьютеров организации.

Разрабатываемая программная система предназначена для использования IT-администраторами организации с целью повышения эффективности и качества их работы. А также оптимизации рабочего времени пользователей и уменьшения затрат человеко-часов на решение возникших проблем.

Основной принцип работы системы заключается в установке устойчивого соединения между компьютерами IT-администратора и пользователя с последующей возможностью передачи данных.

Целью разработки приложения удаленного администрирования компьютеров организации является создание эффективного инструмента для повышения уровня обслуживания IT-администраторами компьютера пользователя.

3.2 Обоснование выбора технологии проектирования

Для разработки был выбран проект WinForms так как он предоставляет широкий спектр необходимых условий для успешного создания программного продукта в соответствии с требованиями пользователя:

1. Простота и быстрота разработки: WinForms предоставляет интуитивно понятный инструментарий для создания пользовательского интерфейса. Благодаря этому, разработка приложения может быть выполнена быстро и эффективно, что особенно важно в условиях ограниченного времени на проект.

2. Широкая поддержка в среде разработки: Технология WinForms широко поддерживается в среде разработки .NET, что обеспечивает доступ к богатому набору инструментов и ресурсов для разработчиков. Это включает

в себя возможность использования Visual Studio, богатый выбор компонентов пользовательского интерфейса и библиотеки классов для работы с сетью и другими аспектами системы.

3. Стабильность и надежность: WinForms является проверенной и стабильной технологией, которая широко используется в промышленных приложениях на протяжении многих лет. Это обеспечивает надежность работы приложения и минимизирует риск возникновения ошибок или проблем в процессе эксплуатации.

4. Привычный пользовательский интерфейс: WinForms предлагает привычный пользовательский интерфейс для пользователей Windows, что может уменьшить время обучения и повысить удобство использования приложения.

5. Поддержка Windows-ориентированных функций: WinForms хорошо поддерживает различные функции операционной системы Windows. Это позволяет легко интегрировать приложение с окружением Windows и использовать его функциональные возможности.

3.2.1 Описание используемых технологий и языков программирования

В рамках разработки приложения удаленного администрирования компьютеров организации было принято решение использовать язык программирования C# в сочетании с протоколом TCP/IP для обеспечения сетевого взаимодействия. Этот выбор обусловлен рядом факторов, которые обеспечивают эффективность, надежность и удобство в разработке и использовании приложения.

3.2.1.1 Язык программирования C#

C# - это объектно-ориентированный язык программирования, разработанный корпорацией Microsoft. C# является частью платформы .NET и широко используется для создания различных типов приложений, включая веб-

приложения, мобильные приложения, приложения для настольных компьютеров и многое другое.

Особенности языка C#, подходящие для данного проекта:

1. Богатый набор стандартных библиотек: C# предоставляет доступ к обширному набору стандартных библиотек .NET Framework, включая классы и компоненты для работы с сетью, управления потоками, обработки исключений и многого другого. Это позволяет разработчикам эффективно использовать готовые решения для реализации функциональности удаленного администрирования.

2. Удобство работы с GUI: C# обладает удобным синтаксисом и широкими возможностями для создания графического пользовательского интерфейса (GUI). Это особенно важно для разработки приложения с удобным и интуитивно понятным интерфейсом, который позволит администраторам эффективно управлять компьютерами организации.

3. Поддержка многопоточности: C# имеет встроенную поддержку многопоточности, что позволяет разработчикам создавать многопоточные приложения для эффективной работы с параллельными задачами, такими как обработка запросов от нескольких компьютеров одновременно.

4. Высокая производительность: C# компилируется в нативный машинный код с использованием JIT (Just-In-Time) компиляции, что обеспечивает высокую производительность выполнения кода. Это особенно важно для приложения удалённого администрирования, где требуется быстрый отклик на запросы администраторов.

5. Поддержка современных парадигм программирования: C# поддерживает современные парадигмы программирования, такие как асинхронное программирование, что позволяет разработчикам создавать эффективные и гибкие приложения с использованием современных подходов к разработке.

3.2.1.2 Протокол TCP/IP

Протокол TCP/IP (Transmission Control Protocol/Internet Protocol) является основным протоколом сетевого взаимодействия в интернете и локальных сетях. Он обеспечивает надёжную и универсальную передачу данных между компьютерами и сетевыми устройствами.

Особенности протокола TCP/IP, подходящие для данного проекта:

1. Надёжность передачи данных: TCP обеспечивает гарантированную доставку данных, контролируя потери, дублирование и порядок получения пакетов. Это особенно важно для приложения удаленного администрирования, где надёжность передачи данных имеет высокий приоритет.

2. Установка соединения: TCP устанавливает соединение между отправителем и получателем перед началом передачи данных, что обеспечивает надёжность и порядок доставки. Это позволяет обеспечить безопасность и целостность передаваемой информации.

3. Поддержка потоковой передачи данных: TCP обеспечивает передачу данных в виде потока байтов, что удобно для приложений, требующих непрерывного потока данных. В случае приложения удаленного администрирования, где требуется передача команд и ответов от удаленных компьютеров, потоковая передача данных является эффективным подходом.

4. Универсальность и широкое распространение: Протокол TCP/IP является стандартом сетевого взаимодействия и широко поддерживается на различных платформах и устройствах. Это обеспечивает совместимость и возможность взаимодействия с различными компьютерами и сетевыми устройствами в организации.

Использование протокола TCP/IP для сетевого взаимодействия в приложении удаленного администрирования компьютеров организации обеспечивает надёжность, безопасность и эффективность передачи данных между администратором и пользователем.

3.3 Проектирование архитектуры программной системы

3.3.1 Архитектура программной системы

На рисунке 3.1 представлена диаграмма классов программной системы.

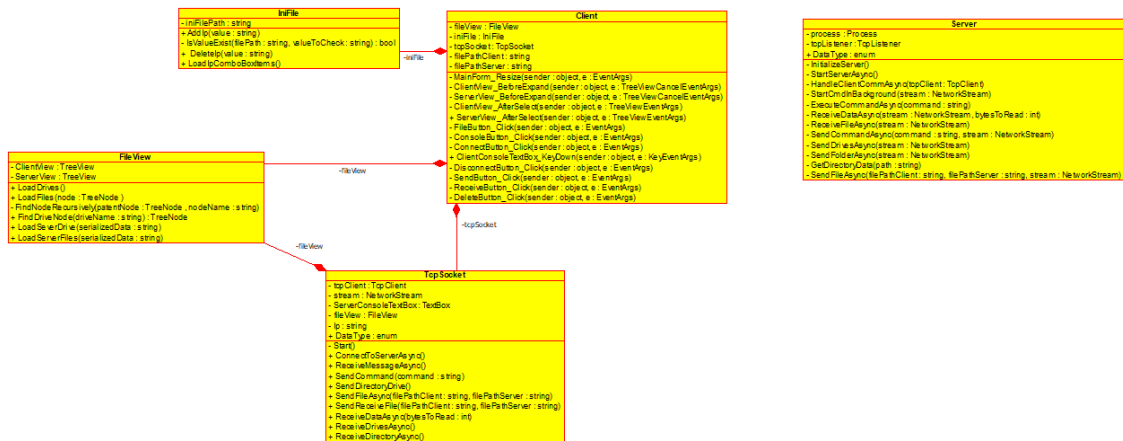


Рисунок 3.1 – Диаграмма классов

3.4 Проектирование пользовательского интерфейса

На основе требований к пользовательскому интерфейсу, представленных в пункте 2.3.3 технического задания, был разработан интерфейс управления программной системой. Для разработки пользовательского интерфейса было выбрано использование компонентов формы в Winform.

Числами на рисунках обозначены номера объектов интерфейса.

На рисунках 3.2 - 3.3 представлен интерфейс программной системы.

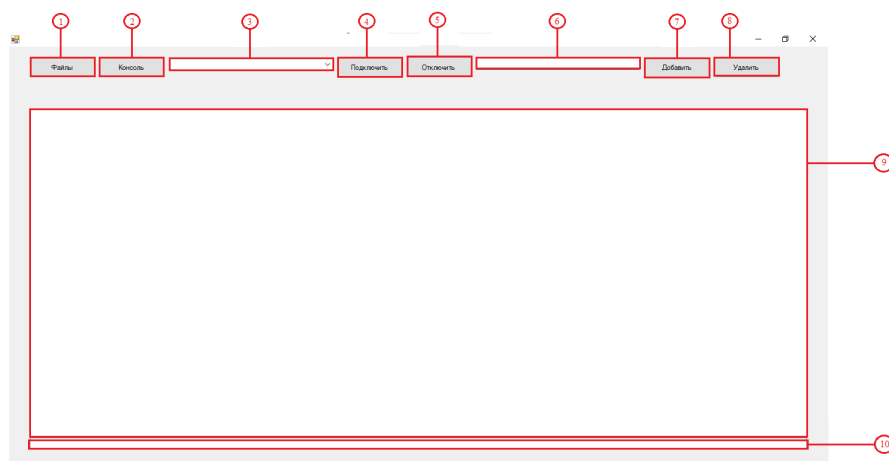


Рисунок 3.2 – Окно с отображением консоли

Элементами данного окна являются:

1. Кнопка открытие файловых систем компьютеров;
2. Кнопка открытие консольного управления сервером;
3. Выпадающий список доступных Ip;
4. Кнопка подключения к серверу;
5. Кнопка отключения от сервера;
6. Поле ввода Ip;
7. Кнопка добавления Ip;
8. Кнопка удаления Ip;
9. Поле просмотра ответа от сервера;
10. Поле ввода команды для отправки на сервер.

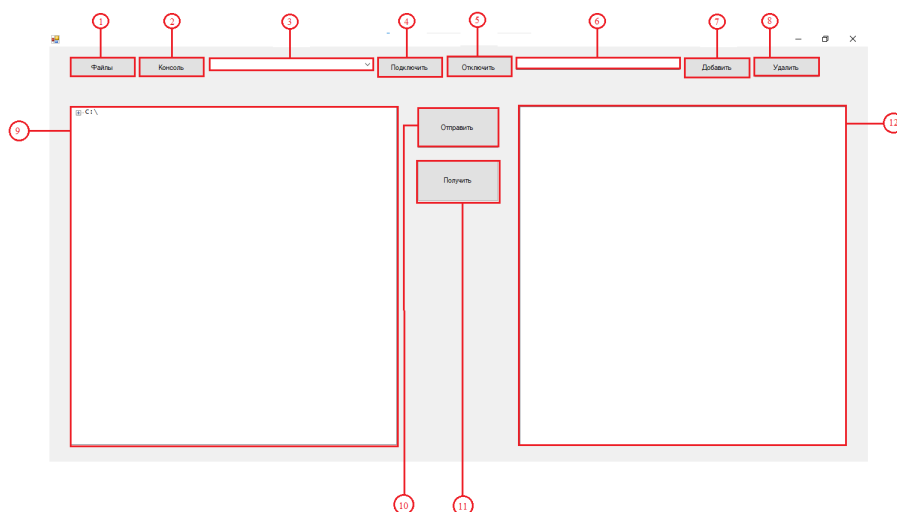


Рисунок 3.3 – Окно с отображением файловой системы

Элементами данного окна являются:

1. Кнопка открытие файловых систем компьютеров;
2. Кнопка открытие консольного управления сервером;
3. Выпадающий список доступных Ip;
4. Кнопка подключения к серверу;
5. Кнопка отключения от сервера;
6. Поле ввода Ip;
7. Кнопка добавления Ip;
8. Кнопка удаления Ip;
9. Поле просмотра файловой системы клиента;
10. Кнопка отправки файла на сервер;
11. Кнопка получения файла от сервера;
12. Поле просмотра файловой системы сервера.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Троелсен, Э., Джеппесен, П. Программирование на языке C# 7.0 и платформы .NET Core./ Э. Троелсен, П. Джеппесен.– М.: ДМК Пресс, 2018.– 960 с.– ISBN 978-5-97060-566-2.
2. Лав, Э., Карнахан, Б. Сетевое программирование в системе UNIX./ Э. Лав, Б. Карнахан.– М.: ДМК Пресс, 2018.– 832 с.– ISBN 978-5-97060-571-6.
3. Фаулер, М. Рефакторинг: Улучшение существующего кода./ М. Фаулер.– СПб.: Питер, 2018.– 576 с.– ISBN 978-5-496-02621-5.
4. Хёрст, Г., Альбахари, Дж. Эффективное программирование на C#./ Г. Хёрст, Дж. Альбахари.– Пер. с англ. – СПб.: Символ-Плюс, 2019.– 768 с. – ISBN 978-5-9903398-2-1.
5. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования./ Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес.– М.: Питер, 2018.– 400 с.– ISBN 978-5-496-02723-6.
6. Костарев, Д. C#, .NET и платформа Microsoft./ Д. Костарев.– М.: БХВ-Петербург, 2016.– 528 с.– ISBN 978-5-9775-2005-9.
7. Комаровский, В. Сокеты программирование в примерах./ В. Комаровский.– М.: Издательский дом «Питер», 2019.– 432 с.– ISBN 978-5-9915-1072-7.
8. Эванс, Э. Язык шаблонов доменной модели./ Э. Эванс.– СПб.: Питер, 2016.– 384 с.– ISBN 978-5-496-02554-6.
9. Сейтман, Дж. C# 7.0 и .NET Core 2.0 – рецепты программирования./ Дж. Сейтман.– М.: ДМК Пресс, 2018.– 640 с.– ISBN 978-5-97060-596-9.