ОПИСАНИЕ СИСТЕМЫ

Обоснование технических и программных средств.

Для создания системы управления сетевой инфраструктурой обоснованы следующие технические и программные средства:

Серверная часть:

Язык программирования: C# - надежный и широко распространенный язык с отличной поддержкой разработки на стороне сервера, что делает его идеальным выбором для создания REST API.

Фреймворк REST API: C# Rest API может быть реализован с использованием ASP.NET Core, зрелого и многофункционального фреймворка для построения современных веб-приложений и API.

Доступ к базам данных: Entity Framework Core - мощный фреймворк Object-Relational Mapping (ORM), который упрощает взаимодействие с базами данных и обеспечивает бесшовную интеграцию с ними.

Клиентская часть:

Язык программирования: C# - универсальный язык, подходящий для разработки на стороне клиента. Он обеспечивает согласованность с языком серверной части, что упрощает сопровождение кода.

Пользовательский интерфейс: C# WPF (Windows Presentation Foundation) - надежный фреймворк для создания настольных приложений с богатыми и интерактивными пользовательскими интерфейсами.

Связь с сервером: C# WPF может использовать JSON-запросы для взаимодействия с REST API на сервере и эффективного получения или обновления данных.

Состав и структура системы.

Система управления сетевой инфраструктурой состоит из двух основных частей: серверной и клиентской.

Серверная часть:

REST API с использованием C# и ASP.NET Core:

Он обрабатывает HTTP-запросы от клиентского приложения и предоставляет соответствующие ответы.

Использует Entity Framework Core для доступа к данным и манипулирования ими.

Реализует различные конечные точки API для выполнения CRUD-операций (Create, Read, Update, Delete) над коммутаторами, почтовыми адресами и типами оборудования.

Клиентская часть:

C# WPF-приложение:

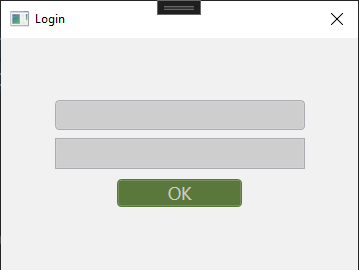
Использует C# для разработки логики и пользовательского интерфейса на стороне клиента.

Взаимодействует с REST API на сервере для получения данных и выполнения операций.

Обеспечивает удобный и отзывчивый интерфейс для поиска коммутаторов, просмотра подробной информации, редактирования и добавления данных.

Руководство пользователя: Действия пользователя программы:

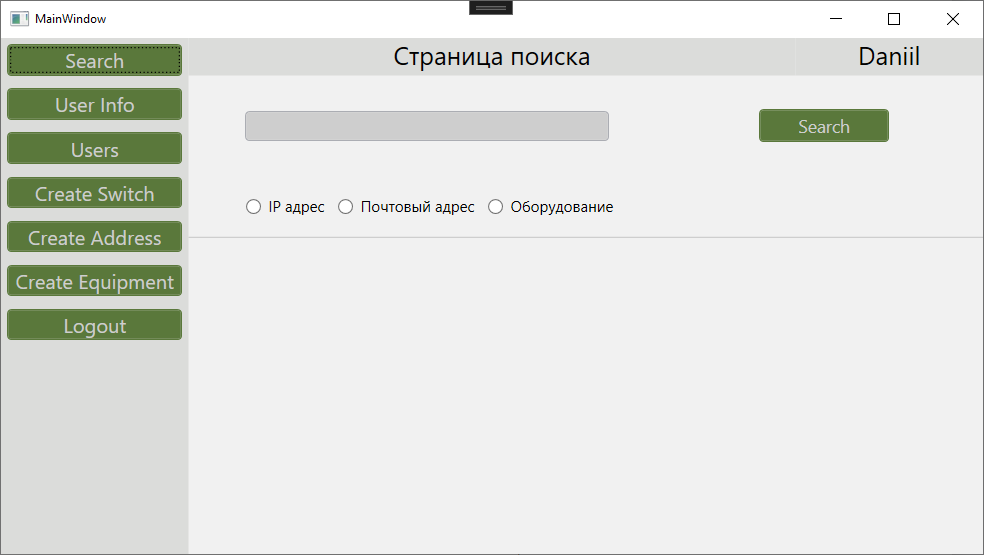
Вход в систему:



Введите имя пользователя и пароль на экране входа в систему.

Для авторизации нажмите кнопку "OK".

Переключатели поиска:

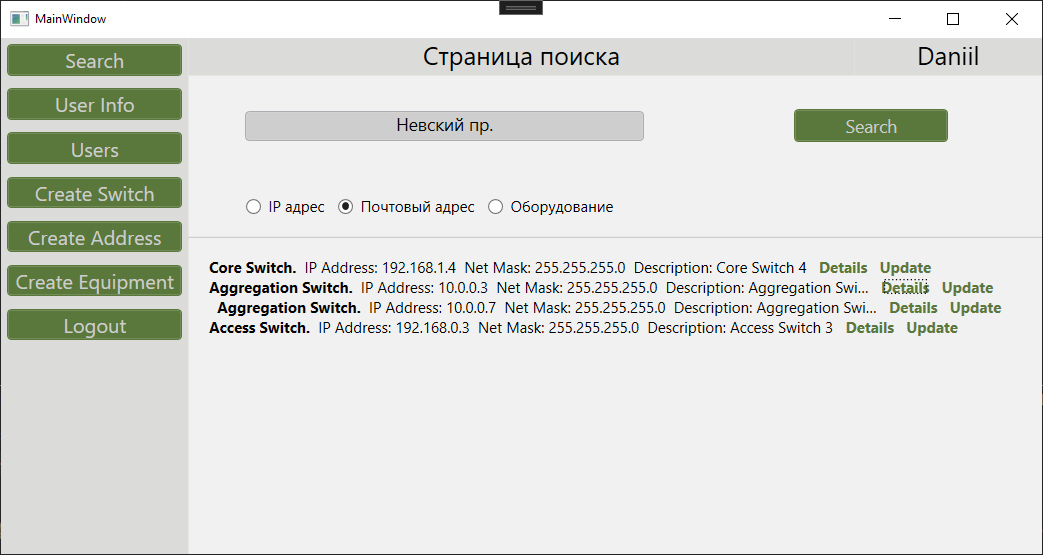


Выберите один из вариантов поиска: "Поиск по IP-адресу", "Поиск по почтовому адресу" или "Поиск по типу оборудования".

Введите соответствующие данные в зависимости от выбранного варианта поиска (например, IP-адрес, почтовый адрес, тип оборудования).

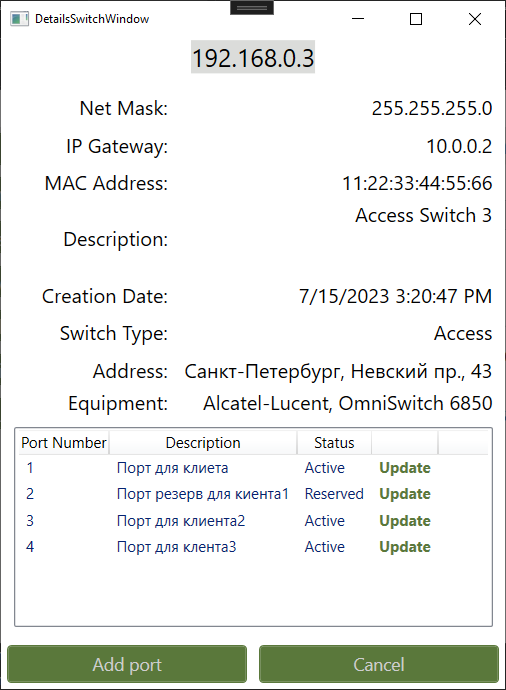
Нажмите кнопку "Search".

Приложение отобразит список коммутаторов, соответствующих критериям поиска.

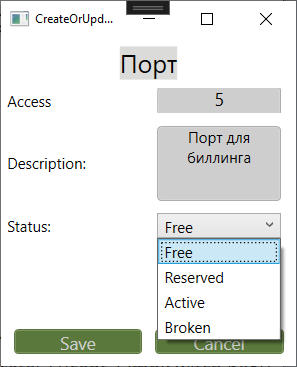


Просмотр подробной информации:

Щелкните на ссылку “Details” из результатов поиска, чтобы просмотреть его подробную информацию.



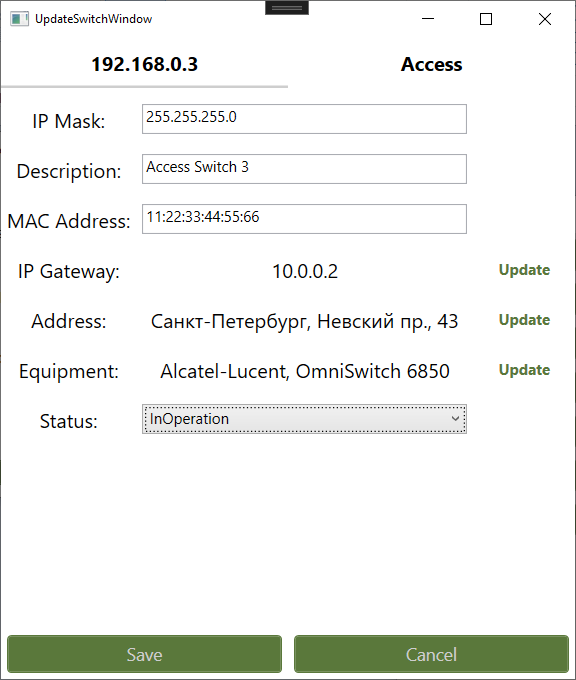
Приложение отображает исчерпывающую информацию о выбранном коммутаторе, такую как IP-адрес, тип оборудования, производитель и т.д., информацию о портах, где возможно редактировать задействованный порт, добавить описание для нового порта, такое как номер порта, описание, статус.



Редактировать коммутатор:

Просмотр подробной информации о коммутаторе.

Для изменения сведений о коммутаторе нажмите кнопку "Update".

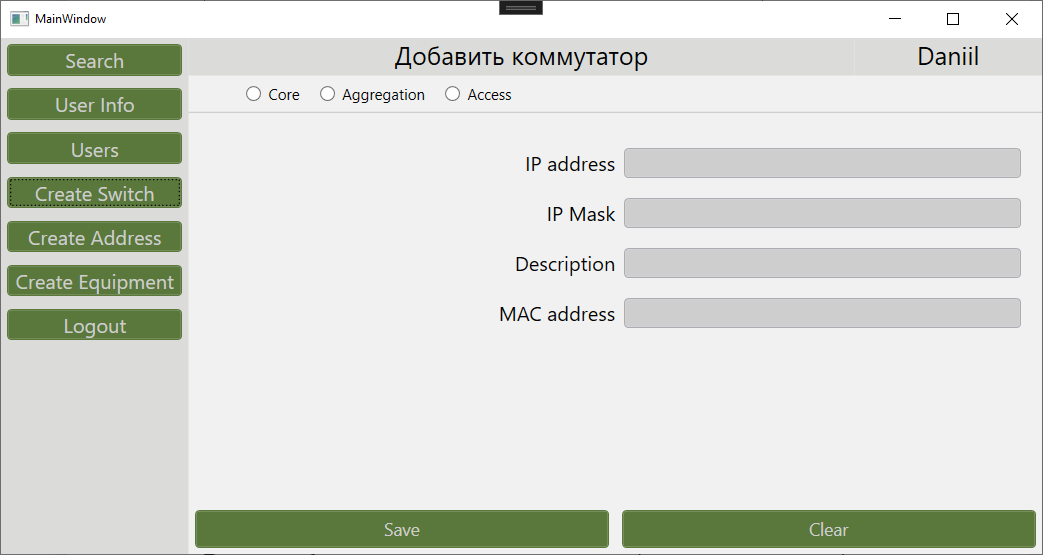


Внесите необходимые изменения в форму редактирования.

Нажмите кнопку "Save", чтобы обновить информацию о коммутаторе.

Добавить новый коммутатор:

Нажмите кнопку "Create Switch" на приборной панели.



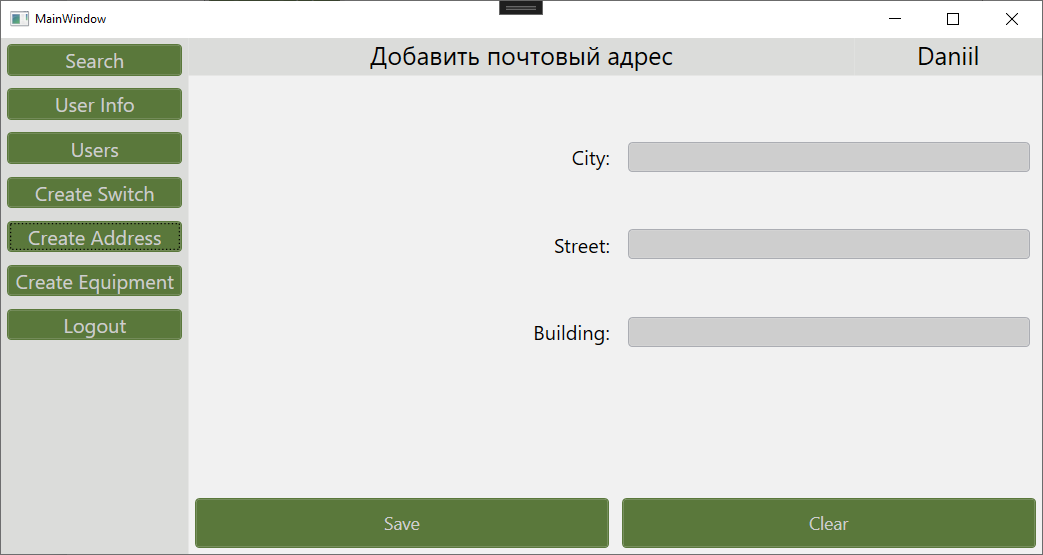
Заполните необходимые данные в форме добавления нового коммутатора.

При попытки добавить коммутатор с уже существующим IP адресом в базе данных возникнет ошибка.

Нажмите кнопку "Save", чтобы добавить новый коммутатор.

Добавить новый почтовый адрес:

Нажмите кнопку "Create Address" на приборной панели.

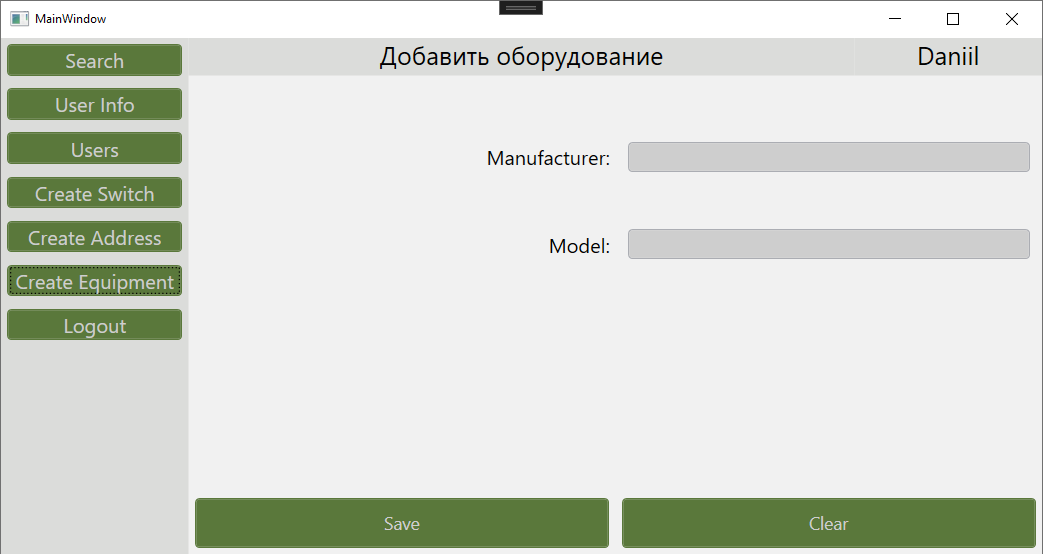


Введите в форму данные о почтовом адресе (город, улица, здание).

При попытки добавить уже существующий адрес в базе данных, возникнет ошибка.

Нажмите кнопку "Save", чтобы добавить новый почтовый адрес.

Добавить новое оборудование:



Нажмите кнопку "Create Equipment" на приборной панели.

Введите данные о новом типе оборудования.

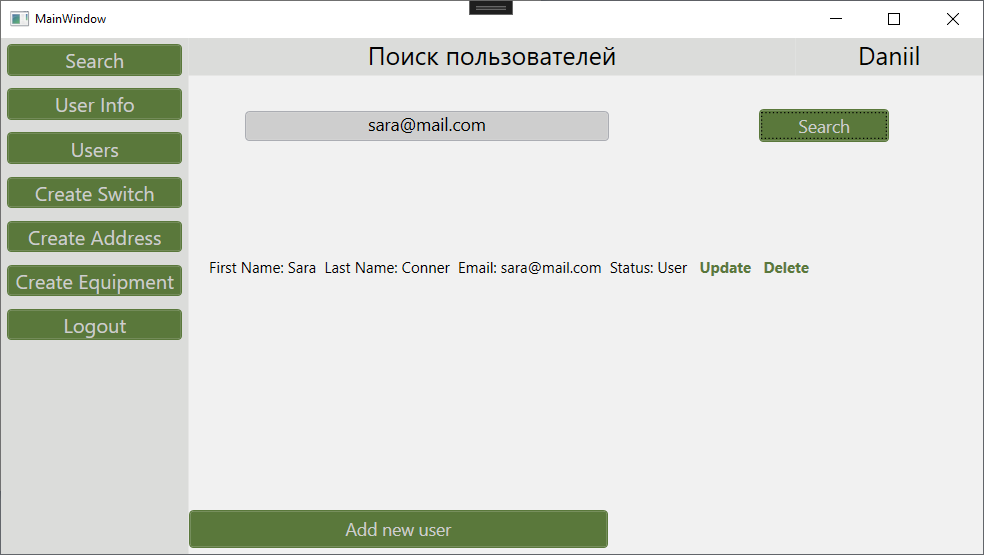
Нажмите кнопку "Save", чтобы добавить новое оборудование.

При попытки добавить уже существующее оборудование в базе данных, возникнет ошибка.

Работа с пользователями:

Нажмите кнопку "Users" на приборной панели.

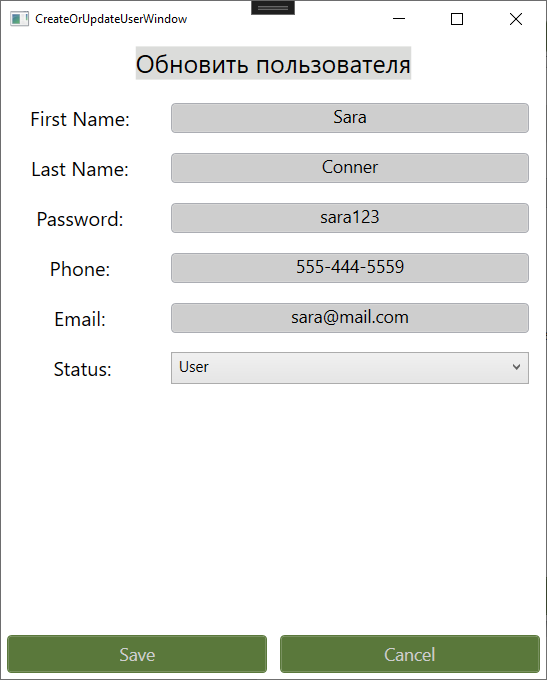
Введите логин (email) пользователя



Для создания нового пользователя нажмите кнопку “Add new user”

Для удаление пользователя нажмите “Delete”

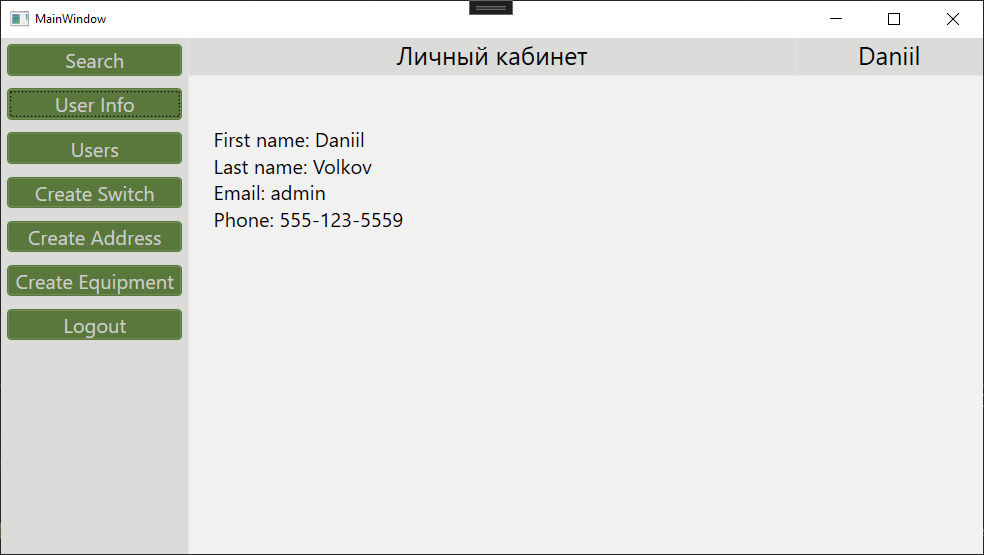
Для обновления пользователя нажмите “Update”



Внесите необходимые изменения в форму редактирования.

Нажмите кнопку "Save", чтобы обновить информацию о пользователе.

Информация о текущем пользователе:



Практическая значимость работы:

Система управления сетевой инфраструктурой, построенная с использованием C# Rest API, Entity Framework Core и C# WPF, имеет практическое значение для управления и организации сложных сетевых инфраструктур:

Эффективный доступ к данным: C# Rest API с Entity Framework Core обеспечивает беспрепятственный и эффективный доступ к данным, позволяя пользователям беспрепятственно получать и изменять информацию о сетевой инфраструктуре.

Опыт работы с настольными приложениями: C# WPF предоставляет богатый пользовательский интерфейс с интерактивными элементами, обеспечивая привычный и эффективный опыт для пользователей настольных систем.

Обновления в реальном времени: Приложение взаимодействует с сервером в режиме реального времени, что позволяет пользователям получать самую свежую информацию о сети.

Масштабируемость: Модульная и хорошо структурированная архитектура обеспечивает легкую масштабируемость, что позволяет использовать ее в растущих сетевых инфраструктурах.

Безопасность данных: Механизмы аутентификации и авторизации пользователей обеспечивают доступ и модификацию критически важной сетевой информации только авторизованным сотрудникам.

Оптимизированное управление: Удобный интерфейс и интуитивно понятные действия упрощают задачи управления сетевой инфраструктурой, повышая производительность и эффективность.

В целом система управления сетевой инфраструктурой, разработанная на базе C# Rest API и C# WPF, представляет собой надежное и практичное решение для эффективного управления и обслуживания сложных сетевых инфраструктур, способствующее повышению производительности, безопасности и оптимизации работы сети.