**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КР.09.02.07.22.201.07 ПЗ |  |

ВЕБ-ИНТЕРФЕЙС АНАЛИЗА ОТКАЗОВ

КОМПЬЮТЕРНОЙ ЛОКАЛЬНОЙ СЕТИ

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.С.Александрова) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.А Иванов) |

Иркутск 2022

**Содержание**

[Введение 5](#_Toc118736998)

[1 Предпроектное исследование 7](#_Toc118736999)

[1.1 Описание предметной области 7](#_Toc118737000)

[1.2 Обзор инструментальных средств для разработки 10](#_Toc118737001)

[2 Техническое задание 13](#_Toc118737002)

# Введение

Локальная сеть – коммуникационная система, состоящая из нескольких компьютеров, соединенных между собой посредством кабелей (телефонных линий, радиоканалов), позволяющая пользователям совместно использовать ресурсы компьютера: программы, файлы, папки, а также периферийные устройства: принтеры, плоттеры, диски, модемы.

Основным преимуществом работы в локальной сети является использование в многопользовательском режиме общих ресурсов сети: дисков, принтеров, модемов, программ и данных, хранящихся на общедоступных дисках, а также возможность передавать информацию с одного компьютера на другой, соответственно из этого вытекает главная задача локальной компьютерной сети.

Главная задача локальной компьютерной сети – это реализация совместного доступа всех пользователей к данным, устройствам и программам. Таким образом, клиентам системы доступно выполнять операции одновременно, а не поочередно.

Локальная компьютерная сеть не может полноценно функционировать без специального оборудования. Для нее основными составляющими являются: компьютеры, принтеры, серверы, рабочие станции, сканеры, маршрутизаторы, коммутаторы (свитчи) и многие другие.

Веб-интерфейс анализа отказов компьютерной локальной сети будет предоставлять информацию об активности компьютеров в локальной сети, отмечать последнюю активность компьютера в данной сети, проверять возможность использования совместного доступа всех пользователей к данному компьютеру и будет мониторить состояние всех компьютеров в данной локальной сети.

Целью данного курсового проекта является разработка и проектирование веб-интерфейса анализа отказов компьютерной локальной сети, которая позволит более эффективно выполнять процессы, происходящие в локальной сети.

Основными задачами данной работы являются:

– произвести исследование процессов в локальной компьютерной сети;

– на основании полученных знаний разработать веб-интерфейс анализа отказов локальной компьютерной сети.

Для того, чтобы достичь результата – разработки веб-интерфейса анализа отказов компьютерной сети нам понадобится решить ряд задач, а именно:

– исследовать предметную область;

– сделать анализ инструментов для разработки;

– произвести разработку приложения для анализа отказа компьютерной сети;

– произвести разработку веб-интерфейса для данного приложения;

–написать документацию по веб-интерфейсу, а именно руководство пользователя.

# 1 Предпроектное исследование

# 1.1 Описание предметной области

Именно передача информации – один из важнейших процессов в локально компьютерной сети. Благодаря тому, что локальная сеть объединяет все компьютеры, установленные в одном или нескольких помещениях в одно единое целое, пользователи в ней могут получать совместный доступ к ресурсам и информации компьютеров, а также ко всем периферийным устройствам, подключенным к данному компьютеру.

Именно поэтому в локальной сети нужен анализ отказов, в пример можно взять то, что большинству пользователей не важно почему информация с одного компьютера не отправляется на другой компьютер в этой же локальной сети им важен результат, но тем же временем информация почему не происходит обмен информацией между компьютерами важен для того кто будет администрировать данную сеть, ведь удобнее знать в чем проблема, а не “чинить все подряд”.

Таким образом анализ отказов компьютерной локальной сети очень важен, так как в локальных сетях могут быть важные компьютеры, работа которых должна осуществляется круглосуточно, а анализ будет гарантировать надежность его работы. Однако невозможно полностью избавится от отказов в работе локальной компьютерной сети, поэтому необходимо своевременно знать работает ли тот или иной важный элемент данной сети. Но стоит отметить, прежде чем начать анализ, необходимо собрать подробности об обстоятельствах и симптомах отказа. Сюда входит исследование изменений характеристик и других данных предшествующие отказу данных, среды, условия нагрузки, монтаж и возможность ошибок человеческого фактора. Анализ этих факторов позволит сделать предположение о потенциальном характере и механизме отказа. В свою очередь на основе этого предположения определяются наиболее подходящие процедуры и методы починки оборудования.

Данный процесс можно представить таким образом:

– Есть локальная компьютерная сеть с множеством устройств, связанных между собой которую, использует множество пользователей.

– Пользователь один хочет отправить какую-то информацию с компьютера, а на компьютер б другому пользователю.

– Информация которую отправил пользователь не дошла получателю и надо узнать почему это произошло.

– Администратор данной сети заходит на веб-интерфейс анализа отказов локальной компьютерной сети, смотри работает ли данный компьютер, отвечает ли он на запросы и так далее. А анализ данных содержит:

– сведения о состоянии компьютера (включен он или же нет);

– сведения о его последнем сеансе в сети;

– сведенья о его доступности в сети;

– сведенья о конфигурации операционной системы, включая уровни пакета обновления.

– Благодаря данному анализу администратор очень быстро находит причину отказа и начинает работать над её устранением.

– При отсутствии анализа отказов администратору нужно будет лично сходить к компьютеру а, полностью проверить его, убедится включен ли он, подключен к сети и доступен ли в ней другим пользователям, а также еще сходить к компьютеру б и сделать тоже самое с ним и только после данных манипуляций уже начать устранять данную проблему.

Для того что-то бы понять, насколько эффективно будет работать данный веб-интерфейс давайте рассмотрим возможные проблемы локальной сети:

– стихийное развитие кабельного хозяйства сети – пока сеть небольшая она будет работать без сбоев, но как только начнётся хотя бы её не большое расширение, администратор столкнется с проблемой того, что изначально такого количества устройств не планировалось, следовательно начнутся сбои в сети;

– плохое качество системы питания компьютеров, включенных в локальную сеть и статическое электричество – из-за плохого качества питающего напряжения, использования не пригодных ограничителей напряжения и всевозможные другие факторы;

– проблемы физической передачи данных по линиям связи (LAN кабелю или оптоволокну) – даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно увидеть многие проблемы, присущие любой вычислительной сети, в том числе проблемы, связанные с физической передачей сигналов по линиям связи, без решения которой невозможен любой вид связи;

– адресация компьютеров – еще одной проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации;

– отсутствие топологии физической связи – под топологией вычислительной сети понимается конфигурация графа, вершинам которого соответствуют компьютеры сети (иногда и другое оборудование, например концентраторы), а ребрам - физические связи между ними. Компьютеры, подключенные к сети, часто называют станциями или узлами сети;

– человеческий фактор – лень или некомпетентность обслуживающего персонала, также не достаточная квалификация персонала контролирующих данную сеть.

Это только основные проблемы локальной сети, большинство из них сможет решить веб-интерфейс анализа отказа локальной компьютерной сети. Данный веб-интерфейс будет предоставлять обширные возможности по мониторингу отказа компьютерной локальной сети, обслуживающему персоналу данной сети.

# 1.2 Обзор инструментальных средств для разработки

Для разработки приложения для анализа отказа компьютерной сети и его веб–интерфейса можно использовать следующие инструментальные средства разработки и языки программирования:

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.

Node или Node.js – программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

WebStorm – интегрированная среда разработки (IDE) на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями.

Visual Studio Code – это один из наиболее популярных редакторов кода, разработанный корпорацией Microsoft. Он распространяется в бесплатном доступе и поддерживается всеми актуальными операционными системами: Windows, Linux и macOS.

MongoDB – документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Написана на языке C++. (Не является реляционной)

MySQL – это реляционная система управления базами данных (СУБД), которая распространяется как свободное программное обеспечение. Является одной из наиболее популярных, так как отличается гибкостью, легкостью, удобством в использовании. Слово «реляционный» означает, что базы представлены в виде связанной информации и описываются как набор связей. MySQL работает с языком запросов SQL, который традиционно используется в базах данных.

PostgreSQL – свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Таблица 1 – Сравнение языков программирования для разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика/язык | JavaScript | Node | Python |
| Объектно-ориентированный язык | – | **+** | **+** |
| Обширная документация | + | **+** | + |
| Кроссплатформенность | + | + | + |
| Интенсивное использование памяти | – | + | \_ |
| Асинхронное программирование | + | + | \_ |
| Масштабируемость | + | + | + |

Как видно из таблицы лучше всего подойдет язык Node.js, так как на нем можно будет реализовать и разработку самого приложения и способов дальнейшего взаимодействия с веб-интерфейсом.

Таблица 2 – Сравнение БД

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика/СУБД | MySQL | PostgreSQL | MongoDB |
| Масштабируемость | + | **+** | **+** |
| Безопасность | + | **+** | **–** |
| Популярность | + | **+** | **+** |
| Скорость работы | + | **–** | **+** |
| Легкость интеграции | + | **–** | **+** |
| Обширная документация | + | **+** | **+** |
| Реляционные принципы хранения информации | + | **+** | **–** |
| Открытость и гибкость запросов информации | + | **+** | **+** |
| Надежность БД | + | **+** | **–** |

Так как нам нужно хранить только список ПК, которые будут проверятся в локальной сети, нет смысла использовать реляционные БД, поэтому была выбрана mongoDB (noSQL БД).

# 2 Техническое задание

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 34.602-2020.

Согласно ГОСТ 34.602-2020 техническое задание должно включать следующие разделы:

1. общие сведения;
2. назначение и цели создания системы;
3. требования к системе в целом;
   1. требования к структуре и функционированию системы;
   2. требования к надежности;
   3. требования к безопасности;
   4. требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
4. требования к документированию;
5. состав и содержание работ по созданию системы.

Техническое задание на разработку веб-приложения представлено в приложении А.

# 3 Проектирование веб-приложений

# 3.1 Структурная схема веб-приложения

Одним из важнейших этапов разработки является проектирование диаграмм, которые помогут нам лучше понять структуру нашего приложения и его работоспособность в целом.

Диаграмма вариантов использования, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент – возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

На рисунке 1 изображена Use Case View, которая показывает структурную схему веб-интерфейса для пользователей «Администратор» и «Пользователь».

Она отображает действия, выполняемые администратором и информационной системой. «Администратор» и «Пользователь» являются – актерами.

Пользователь – может регистрироваться, авторизоваться и просматривать информацию о устройствах.

Администратор – помимо регистрации, авторизации и просмотра информации может: изменять, удалять, добавлять список устройств, управлять запросами, а именно изменять время запроса, количество запросов, изменение промежутка времени между запросами.

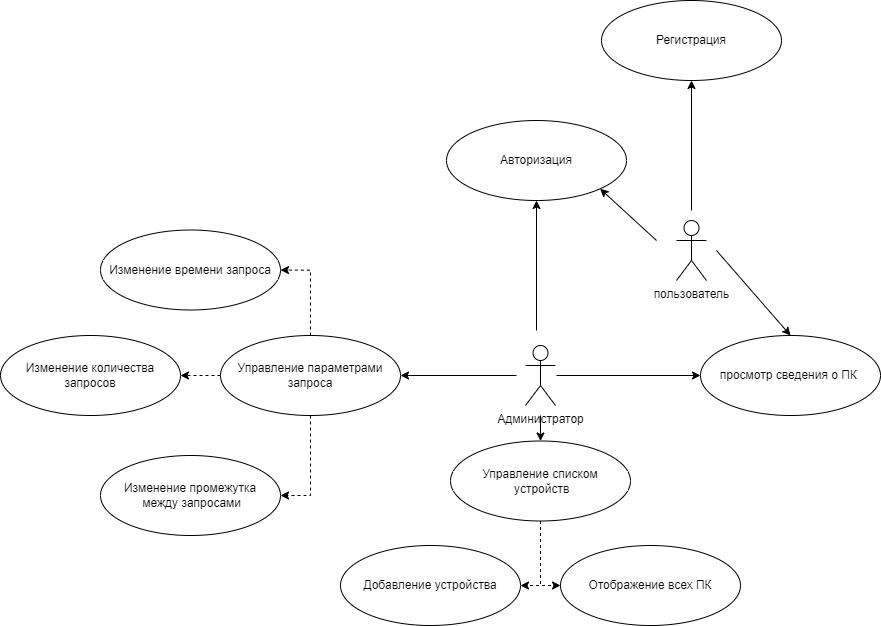
  
Рисунок 1 – Диаграмма прецедентов

Диаграмма деятельности - UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов - вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. На рисунке 2 изображена диаграмма деятельности.

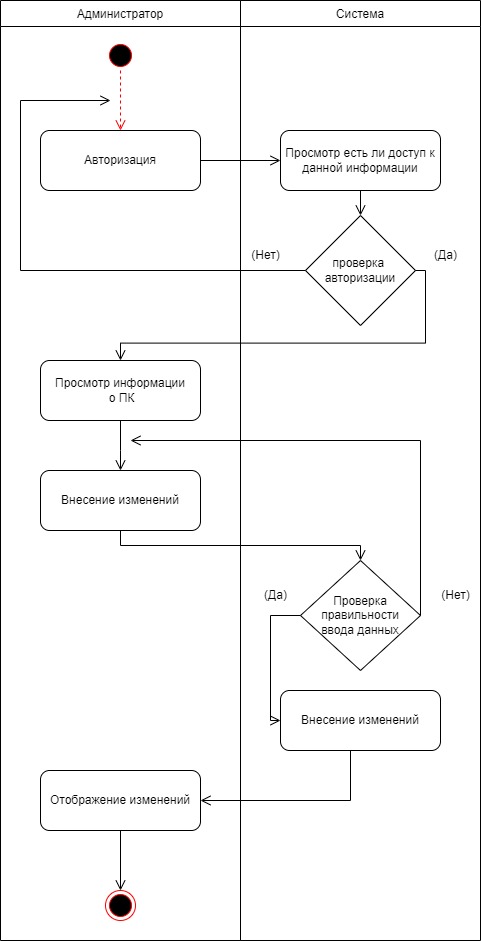


Рисунок 2 – Диаграмма деятельности.

В итоге проектирования диаграммы деятельности были выделены основные возможные действия пользователя с программным продуктом.

# 3.2 Функциональная схема веб-приложение

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Также данная диаграмма является неотъемлемой часть любого веб-приложения. Диаграмма классов разрабатываемого веб интерфейса предоставлена на рисунке 3

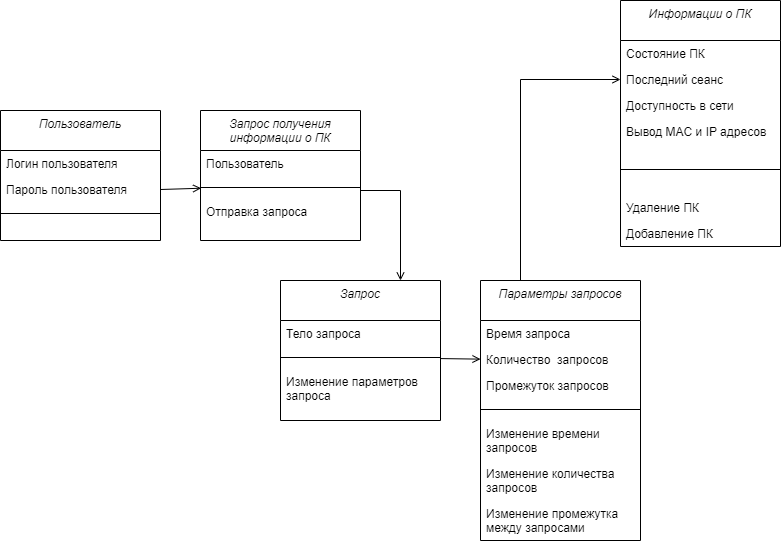


Рисунок 3 – Диаграмма деятельности.

Диаграммы потоков данных (Data Flow Diagrams – DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. Диаграмма потоков данных на рисунке 4 полностью демонстрирует каждый процесс веб-приложения.

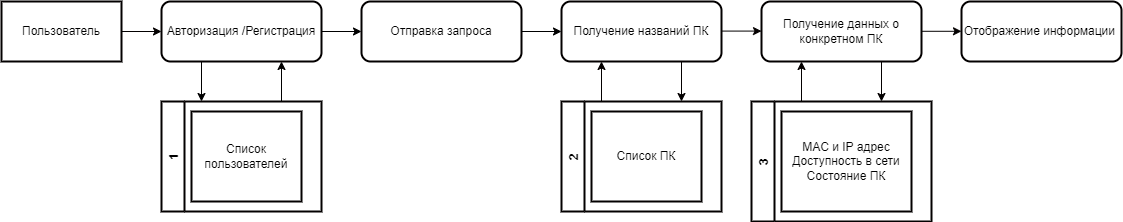


Рисунок 4 – Диаграмма потоков данных.

Из данной диаграммы видно, что веб-интерфейс будет обращаться к базе данных три раза, а именно при регистрации или авторизации, получении имени пк и получении данных о конкретном компьютере.

# 3.3 Проектирование базы данных

Прежде чем приступить разработке программного обеспечения необходимо спроектировать базу данных, а именно, определить с какими данными будут работать участники системы, и чем данные связаны между собой. В этом заключается процесс проектирования.

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных.

Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты). Инфологическая модель на рисунке 5 наглядно показывает объект, его свойства и отношения между другими объектами. Объекты: пользователь, информация о пк, пк, сетевые интерфейсы, MAC, IP. Свойства – это атрибут объекта (овал). Отношения показаны с помощью ромба.

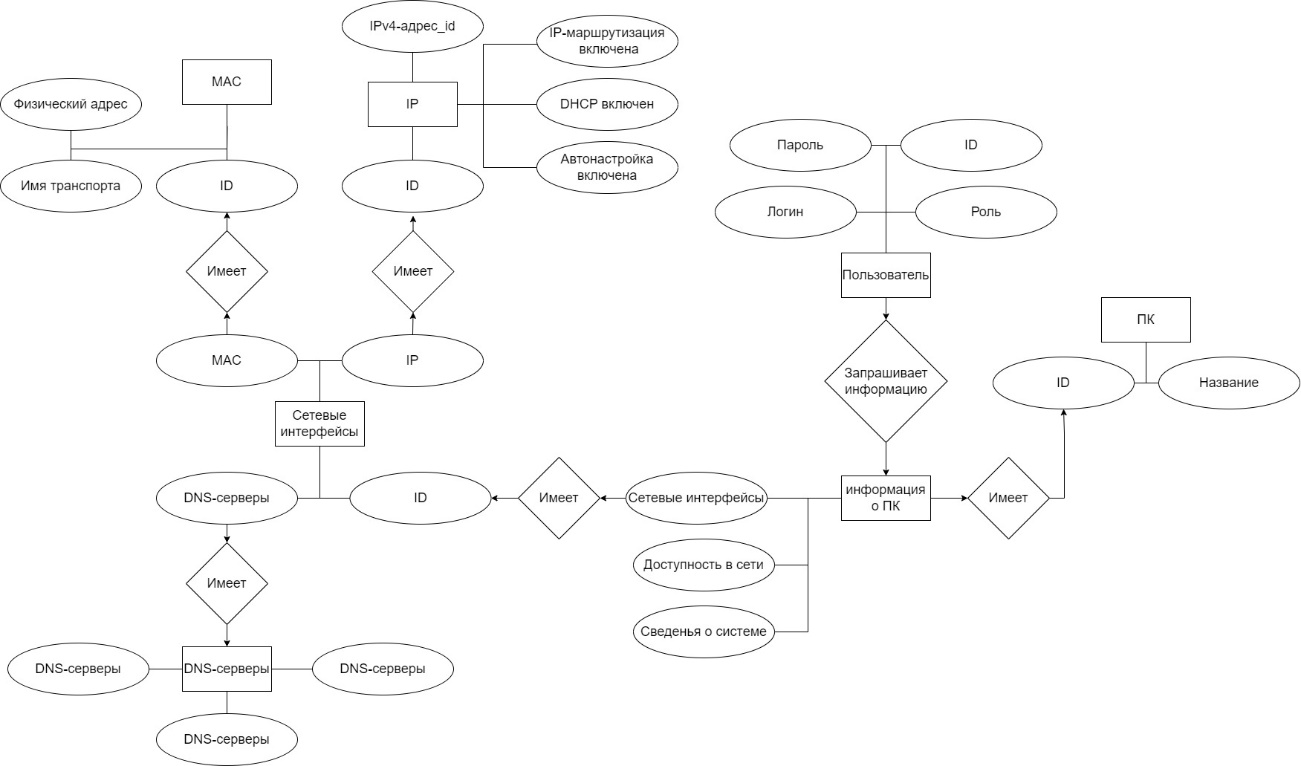


Рисунок 5 – Инфологическая модель базы данных

На модели видно семь сущностей, которые имеют свои уникальные атрибуты. Также сущности связаны между собой и в сумме получается структура под названием – инфологическая модель.

Проектирование базы данных не ограничивается только этим этапом разработки, а проходит на протяжении всей разработки, до того момента пока в системе не появятся данные, которые нельзя потерять. Результатом проектирования базы данных является ER-модель. На рисунке 6 предоставлена ER-модель базы данных.

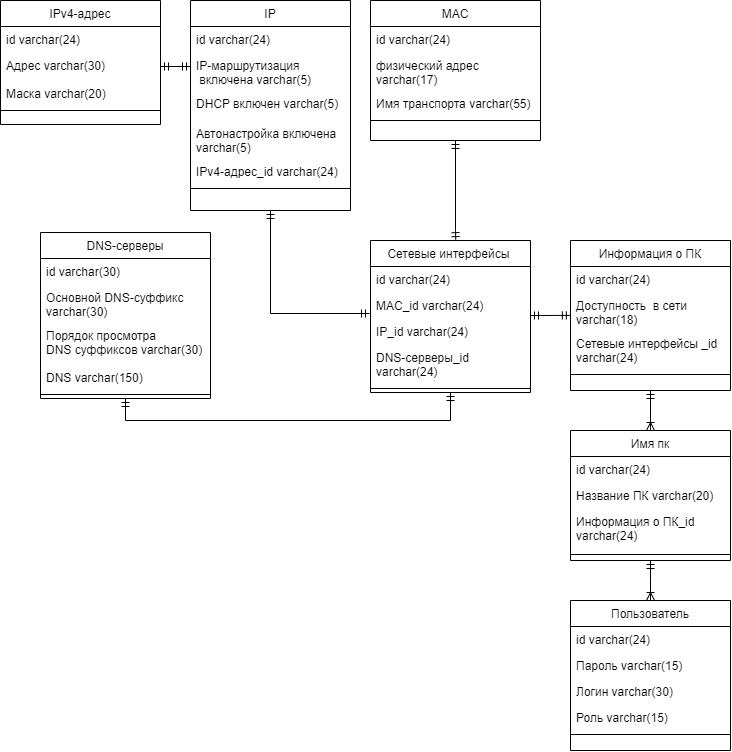


Рисунок 6 – ER-модель базы данных

Данная ER-модель базы данных содержит восемь таблиц.

1. Таблица «Пользователь», предназначена для хранения данных, как:

* Id (идентификатор)
* Пароль (пароль пользователя)
* Логин (username пользователя)
* Роль (роль пользователя)

1. Таблица «Имя пк», предназначена для хранения данных, как:

* Id (идентификатор)
* Название пк (наименование пк)
* Информация о пк (идентификатор информации о пк)

1. Таблица «Информация о пк», предназначена для хранения данных, как:

* Id (идентификатор)
* Доступность в сети (видимость в сети)
* Сетевые интерфейсы (идентификатор сетевые интерфейсы)

1. Таблица «Сетевые интерфейсы», предназначена для хранения данных, как:

* Id (идентификатор)
* MAC (идентификатор MAС)
* DNS-серверы (идентификатор DNS-серверов)

1. Таблица «DNS-серверы», предназначена для хранения данных, как:

* Id (идентификатор)
* Основной DNS-суффикс
* Порядок просмотра DNS суффиксов (список DNS-серверов для просмотра)
* DNS (DNS сервера)

1. Таблица «IP», предназначена для хранения данных, как:

* Id (идентификатор)
* IP-маршрутизация (определение параметра ip-маршрутизации)
* DHCP включен (определение параметра DHCP)
* Автонастройка включена (определение параметра автонастройка)
* IPv4-адрес (идентификатор ip-адрес)

1. Таблица «IPv4-адрес», предназначена для хранения данных, как:

* Id (идентификатор)
* Адрес (ip-адрес)
* Маска (маска подсети)

1. Таблица «MAC», предназначена для хранения данных, как:

* Id (идентификатор)
* Физический адрес (адрес DNS)
* Имя транспорта (имя транспорта DNS адреса)

Сама база данных была доведена до второй нормальной формы. Но прежде, чем она стала таковой пришлось пройти два этапа, это:

1. Нормализация до первой нормальной формы.
2. Нормализация до второй нормальной формы.

Далее рассмотрим, что такое нормальная форма. Нормальная форма– свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение. Существует шесть нормальных форм, и имеющаяся база данных была доведена до второй.

При нормализации в первую нормальную форму были соблюдены условия, которые делают ее таковой. А именно все данные должны быть атомарными. То есть одно поле – одно значение. Это условие было соблюдено и таблица в первой нормальной форме, идем дальше.

Для нормализации во вторую нормальную форму необходимо: во-первых привести форму базы данных к первой нормальной форме, а во-вторых таблицы должны быть связаны между собой, для того чтобы данные в таблицах не дублировались. Ведь если иметь одинаковые записи в разных таблицах, то придется изменять данные в одной таблице и в другой тоже, а это затрачивает много сил и ресурсов, поэтому для более быстрого и эффективного редактирования данных в базе данных лучше вынести повторяющуюся информацию.

Таким образом, можно увидеть всю необходимую информацию для понимания системы хранения данных.

Именно таким образом, можно представить всю необходимую информацию для понимания системы хранения данных.

# 3.4 Проектирование пользовательского интерфейса

Пользовательский интерфейс – это все, что помогает людям управлять устройствами и программами с помощью голоса, нажатий, жестов, через командную строку

Интерфейсы являются основой взаимодействия всех современных информационных систем. Если интерфейс какого-либо объекта не изменяется (стабилен, стандартизирован), это даёт возможность модифицировать сам объект, не перестраивая принципы его взаимодействия с другими объектами.

Прототип интерфейса на рисунке 7 был разработан в онлайн конструкторе Figma. На нем изображено страница вывода данных «Все устройства» на нем отображено меню и все основные блоки. Информация выводится в виде таблицы.

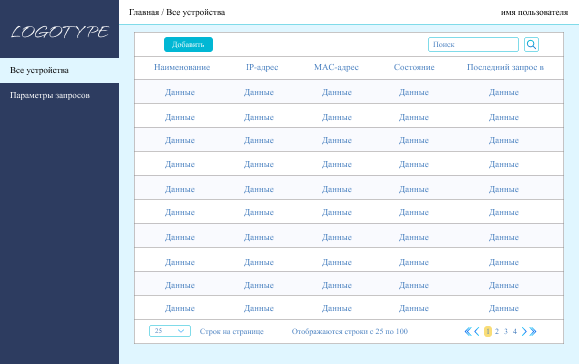


Рисунок 7 – Страница все устройства

На рисунке 8 представлено отображение формы ввода информации, которые будут использоваться для добавления устройств.

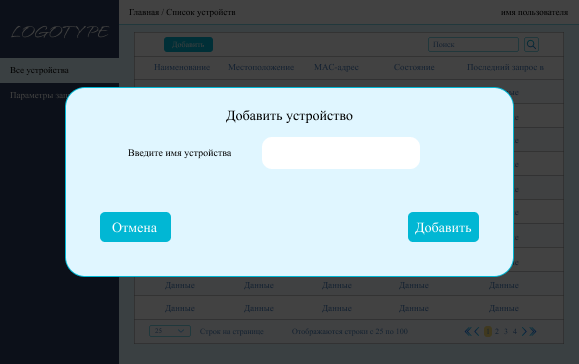


Рисунок 8 – Всплывающее окно добавления устройства

На рисунке 9 предоставлено еще одно отображение формы ввода информации, которые будут использоваться для изменения параметров запроса времени запроса, количества запросов, времени жизни запроса, объёма данных, отправляемых запросом и другие.

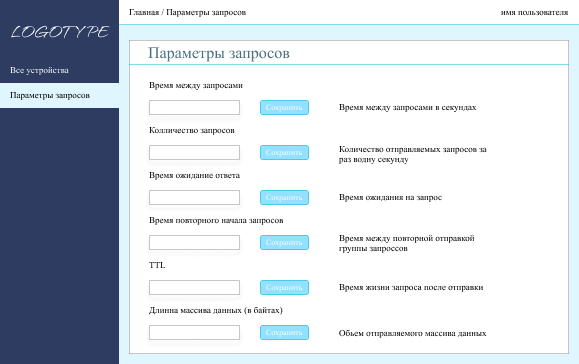


Рисунок 9 – Страница параметров запросов

Во время разработки прототипа интерфейса был учтен проведенный анализ аналогов, что помогло реализовать представленный веб интерфейс.

**Приложение А Техническое задание**

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| ТЕХНИЧЕСКОЕ ЗАДАНИЕ |  |

ВЕБ-ИНТЕРФЕЙС АНАЛИЗА ОТКАЗОВ

КОМПЬЮТЕРНОЙ ЛОКАЛЬНОЙ СЕТИ

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.С.Александрова) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.А Иванов) |

Иркутск 2022

# 1 Общие сведения

Наименование работы: веб-интерфейс анализа отказов компьютерной локальной сети.

Исполнитель: студент иркутского авиационного техникума, группы ВЕБ-20-1, Иванов Д.А.

Разработка веб-интерфейса проходит в рамках курсового проекта по МДК.09.01 «Проектирование и разработка веб-приложений», на основании приказа №46-у от 15 сентября 2022 года.

Сроки разработки веб-интерфейса с 15.09.2022 по 28.06.2023 года.

# 2 Назначение и цели создания системы

Назначение веб-интерфейса анализа отказов компьютерной локальной сети заключается в оптимизации, контроля, централизованности сети, а также её мониторинга. Таким образом данный веб-интерфейс будет облегчать работу администраторам данной сети.

# 3 Требования к системе в целом

# 3.1 Требования к структуре и функционированию системы

Функции веб-интерфейса:

1. раздел «Все устройства»;
   1. Просмотр устройств;
   2. Отображение информации о MAC И IP адресе;
   3. Отображение состояния ПК;
   4. Отображение даты последнего опроса ПК в сети;
   5. Отображение информации о количестве пк;
   6. Добавление нового ПК;
   7. Фильтрация списка по имени пк
2. Раздел «Настройка запросов»;
   1. отображение времени, которое будет затрачено на отправку запросов;
   2. Количество отправляемых запросов;
   3. Промежуток времени между отправляемыми запросами.

# 3.2 Требования к надежности

Для обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях таблицы. Также необходимо обеспечить постоянное подключение к сети.

# 3.3 Требования к безопасности

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа.

# 3.4 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Минимальные системные требования для рабочей станции:

1. Процессор: Intel Core 2 duo 2.0Ghz / AMD phenom x4;
2. Оперативная память: 2048 Мб;
3. Жёсткий диск: 12000мб;
4. Операционная система: Windows, Linux, MacOS.

Минимальные системные требования для сервера:

1. Процессор: Intel Core 2 duo 2.0Ghz / AMD phenom x4;
2. Оперативная память: 3072 Мб;
3. Жёсткий диск: 18000мб;
4. Операционная система: Windows.

# 4 Требования к документированию

Основным документам, регламентирующими использование информационной системы является руководство пользователя.

Основным документам, регламентирующими разработку информационной системы является техническое задание.

# 5 Состав и содержание работ по созданию системы

В таблице 1 представлены плановые сроки начала и окончания работы по созданию информационной системы

Таблица 1 – Плановые сроки по созданию информационной системы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Содержание этапа или стадии  выполнения КП | Сроки выполнения | | Форма  отчетности |
| Начало | Окончание |
| 1 | Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 15.09.22 | 30.09.22 | ПЗ содержание, предметное исследование |
| 2 | Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 30.09.22 | 01.10.22 | ПЗ, ТЗ  обзор средств разработки и проектирования |
| 3 | Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 01.10.22 | 17.11.22 | Проектирование проекта, добавление пунктов в ПЗ |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Оформление пунктов пояснительной записки (введение, предпроектное исследование, техническое задание и проектирование веб-приложения) | 17.11.22 | 18.12.22 | Оформление ПЗ |
| 5 | Разработка (программирование) и отладка программного продукта | 18.12.22 | 04.05.23 | Программный продукт, добавление пунктов в ПЗ |
| 6 | Составление программной документации (оформление ПЗ, руководство пользователя и презентации)  Оформление пунктов пояснительной записки (Разработка веб-приложения, технологическая документация, заключение, список используемых источников и приложения) | 05.06.23 | 14.06.23 | Готовое ПЗ и руководство пользователя |