

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Вычислительная техника»

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий

«\_\_» \_\_\_\_\_ 2022 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника  
код и наименование направления

**Тема: Система тестирования телекоммуникационного оборудования**

Руководитель \_\_\_\_\_ старший преподаватель И.Н. Рыженко  
подпись, дата

Выпускник \_\_\_\_\_ Г.М. Турзунов  
подпись, дата

Нормоконтролер \_\_\_\_\_ старший преподаватель А.Ю. Сидоров  
подпись, дата

Красноярск 2022

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Система тестирования телекоммуникационного оборудования»

**Цель работы:** написание программного обеспечения для удаленного тестирования и конфигурирования телекоммуникационного оборудования на этапе производства.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к программному обеспечению.

Объект работы – приложение, которое по описанному сценарию формирует набор команд для системы управления, конфигурирует модем и проверяет результат.

### Задачи:

- осуществить выбор программных средств моделирования и разработки программного обеспечения;
- выполнить моделирование разрабатываемого программного обеспечения;
- выполнить программную реализацию приложения для тестирования телекоммуникационного оборудования;
- проанализировать полученные результаты работы;

					КП-09.03.01 ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата				
Разработал		Турзунов Г.М.			Система тестирования телекоммуникационного оборудования	Лит.	Лист	Листов
Проверил		Сидоров А.Ю.					2	35
						09.03.01		
Н. контроль								

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Система тестирования телекоммуникационного оборудования»

**Цель работы:** написание программного обеспечения для удаленного тестирования и конфигурирования телекоммуникационного оборудования на этапе производства.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к программному обеспечению.

Объект работы – приложение, которое по описанному сценарию формирует набор команд для системы управления, конфигурирует модем и проверяет результат.

### **Задачи:**

- осуществить выбор программных средств моделирования и разработки программного обеспечения;
- выполнить моделирование разрабатываемого программного обеспечения;
- выполнить программную реализацию приложения для тестирования телекоммуникационного оборудования;
- проанализировать полученные результаты работы;

					КП-09.03.01 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

## Содержание

Введение.....	2
1. <u>Анализ задания</u> .....	2
1.1 <u>Оборудование</u> .....	2
1.2 <u>Выбор инструментов</u> .....	2
1.2.1 <u>Vue.js</u> .....	2
1.2.2 <u>.net 5 (ASP.NET Core 5.0 Web API)</u> .....	2
1.2.3 <u>SQLite</u> .....	2
1.3 <u>Анализ существующих аналогов</u> .....	2
1.4 <u>Функциональные требования</u> .....	2
1.5 <u>Выводы по главе</u> .....	2
2 <u>Этапы проектирования</u> .....	2
2.1 <u>Архитектура приложения</u> .....	2
2.2 <u>Прецеденты</u> .....	2
2.3 <u>База данных</u> .....	2
2.4 <u>Выводы по главе</u> .....	2
3 <u>Программная реализация</u> .....	2
3.1 <u>Главный экран Web-приложения</u> .....	2
3.2 <u>Вкладка информации об устройстве</u> .....	2
3.3 <u>Модуль тестирования</u> .....	2
3.4 <u>Модуль просмотра истории тестирования</u> .....	2
3.5 <u>Функции для работы с базой данных</u> .....	2
3.5.1 <u>Функция GetAll</u> .....	2
3.5.2 <u>Функция Post</u> .....	2
3.5.3 <u>Функция Update</u> .....	2
3.5.4 <u>Функция Delete</u> .....	2
3.5.5 <u>Функция сохранения результата тестирования в БД</u> .....	2
3.6 <u>Пример тестирования устройства</u> .....	2
3.6.1 <u>Передача данных Python скрипту</u> .....	2
3.6.2 <u>Сценарий тестирования и пример тестирования</u> .....	2
<u>Заключение</u> .....	2

					КП-09.03.01 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

<u>Список сокращений</u> .....	2
<u>Список использованных источников</u> .....	2

					КП-09.03.01 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

## Введение

Пандемия коронавируса и последовавшая за этим самоизоляция показала какую большую роль играет интернет в жизни людей по всему миру. Благодаря ему стала возможна удаленная работа, общение с родными и близкими людьми и родственниками, запись в медицинские учреждения, учеба. Но также это непростое время стало для многих пользователей настоящим испытанием.

В такой огромной стране, как Россия, невозможно обеспечить стопроцентное покрытие территории только за счет оптоволокну и сотовых сетей. Важную роль в телекоммуникационной жизни страны играет также спутниковый интернет. Особенно очевидно это стало во время пандемии коронавируса. Во время самоизоляции люди уезжали за город, на дачу, а базовые станции сотовой связи были сильно загружены, поэтому многие подключали спутниковый интернет. Мобильный интернет сотовых операторов попросту не смог справиться с возросшей нагрузкой на базовые станции. Когда все, кто мог, переехали из города в область, то часы наибольшей нагрузки сместились на полное дневное время, с 9 утра до 9 вечера, – и вот здесь спутник оказался незаменимой альтернативой, сравнимой по стоимости даже с оптоволокну.

Основа удаленной работы команды – это обеспечение сотрудников, живущих далеко за городом, надежным спутниковым интернетом. География покрытия спутникового интернет вся территории страны, поэтому даже если сотрудники живут в отдаленной местности, им можно обеспечить надежную связь, а в результате бизнес не потеряет эффективности.

Самый высокий спрос на спутниковый интернет фиксируется на Дальнем Востоке, но его уверенно догоняют жители центральной части России. Особую потребность в спутниковом интернете ощутили семьи, где есть школьники и студенты, в связи с переходом на онлайн-обучение.

					КП-09.03.01 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

Задачи, которые спутниковый интернет помогал решать корпоративным клиентам, в период пандемии не изменились. Это услуги доступа к цифровым ресурсам компаний, использование корпоративных приложений, почтовых сервисов, телефонии, доступа в интернет, получение данных мониторинга, удаленное управление технологическими процессами и видеонаблюдение. Основной запрос в частном использовании (не для бизнеса) – это удаленная работа из дома, онлайн-банкинг и покупка товаров, работа с государственными сервисами, например порталом Госуслуг.

Порой не особо разбирающиеся в технике пользователи не могут оперативно определить причины медленной работы или того хуже, отсутствия интернет-соединений. Причин, конечно, может быть очень много, но некоторые неисправности, могут быть связаны с работой телекоммуникационного оборудования и определить их можно довольно легко, используя систему тестирования. Раньше эта процедура занимала некоторое время, требовала постоянного ручного ввода. Использование системы тестирования позволяет, довольно легко и быстро, всего в несколько кликов, провести полное тестирование, с возможностью выбора различных сценариев. Любой пользователь с легкостью может получить нужную информацию о корректности работы тестируемого устройства.

					КП-09.03.01 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 1. Анализ задания

Целью работы является разработать приложение, которое по описанному сценарию формирует набор команд для системы управления, конфигурирует модем и проверяет результат. Система будет универсальная, тестирование можно будет проводить не только для этого модема, но и любого другого телекоммуникационного оборудования.

### 1.1 Оборудование

Тестирование будет производиться спутникового маршрутизатора ЯР-1040. Модем представляет собой универсальное оборудование систем передачи данных с программно-изменяемыми функциями. Его общий вид представлен на рисунке 1, а задняя панель изделия представлена на рисунке 2.

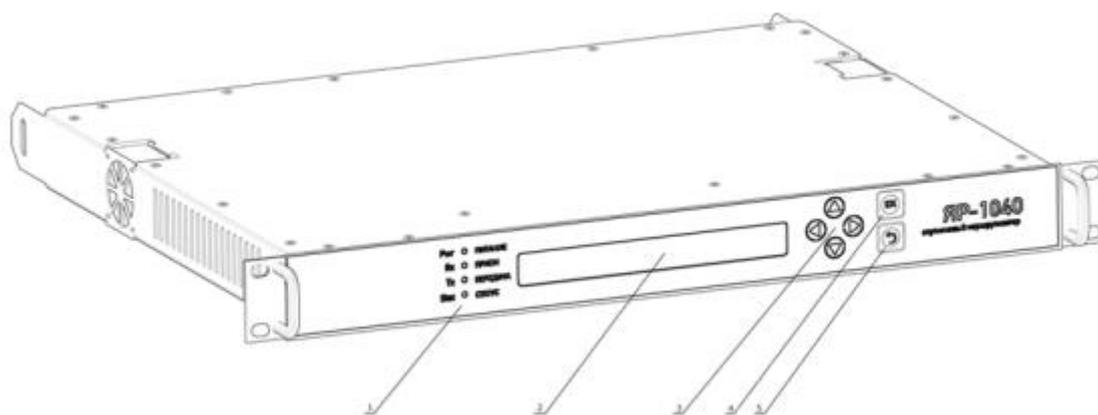


Рисунок 9 – Общий вид изделия ЯР-1040

					КП-09.03.01 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

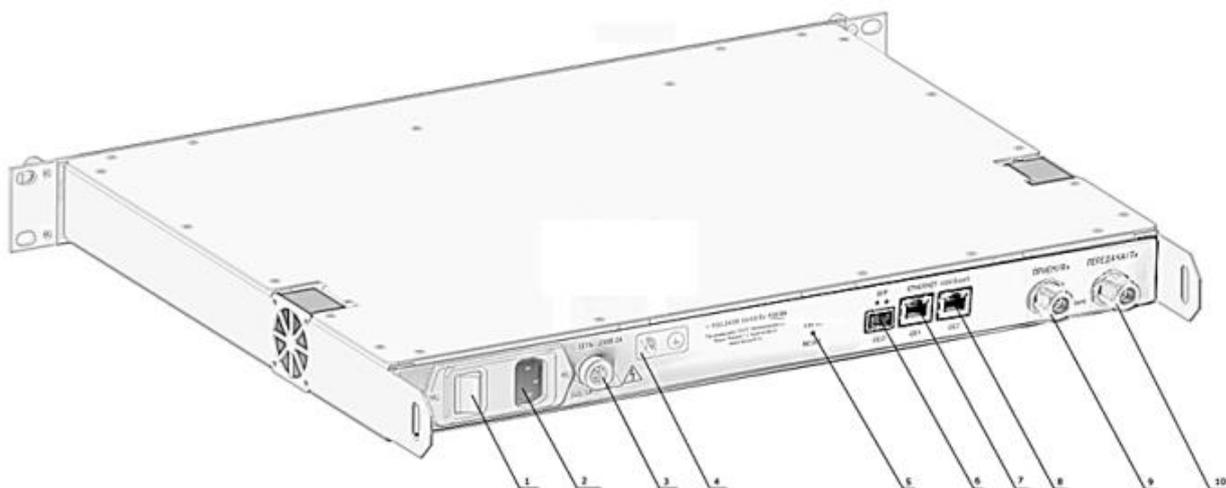


Рисунок 10 – Задняя панель изделия

Система управления изделия предоставляет возможность выгрузки рабочей или загрузочной конфигурации на удалённый сервер протокола FTP или TFTP либо загрузки конфигурации с него (при использовании командного интерпретатора), а также выгрузки (загрузки) по протоколу HTTPS (при использовании веб-интерфейса). Например, у оператора имеется возможность сформировать требуемую рабочую конфигурацию на одной единице изделия, выгрузить её на сервер или свою рабочую станцию, отредактировать в текстовом редакторе, загрузить на ту же единицу изделия, протестировать, и затем загрузить на весь имеющийся у него парк.

					КП-09.03.01 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

## 1.2 Выбор инструментов

### 1.2.1 Vue.js

В качестве инструмента для реализации vue мной был выбран фреймворк vue.js – это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления, что упрощает интеграцию с другими библиотеками и существующими проектами.

### 1.2.2 .net 5 (ASP.NET Core 5.0 Web API)

Backend был написан с помощью фреймворка .net 5 – небольшая коллекция скриптов для создания интерактивной документации для API веб-приложений с REST протоколом. Очень полезно, если вы пишете приложение, которое должно взаимодействовать с внешней системой, а договориться друг с другом в текстовом формате мало. Интерактивность проявляется в том, что из документации можно делать HTTP-запросы.

					КП-09.03.01 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

### 1.2.3 SQLite

База данных реализована с помощью SQLite – компактная встраиваемая СУБД, которая удовлетворяет текущим требованиям к программе



Рисунок 11 – .net 5 / VueJS / SQLite

					КП-09.03.01 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

### 1.3 Анализ существующих аналогов

Прямого конкурента нет, но было найдено похожее по функционалу ПО. Zabbix – это бесплатное программное обеспечение, предназначенное для мониторинга многочисленных параметров сети, жизнеспособности и целостности серверов. Важным достоинством является множество полезных функций в одном пакете. Сбор данных, хранение данных в истории, множество настроек оповещений.

Однако, «Zabbix» имеет ряд недостатков, он сложен в освоении, далеко не каждый пользователь сможет освоить принципы его работы. «Zabbix» поддерживает не все операционные системы и требует установки на устройство. Это усложняет процесс внедрения и сопровождения, т.к. каждому человеку нужно будет установить систему на свой компьютер. Из этого следует, что нам необходима система, реализованная как web-приложение. Это обеспечит возможность входа в неё из любого устройства, имеющего доступ в сеть Интернет.

Таким образом было принято решение разработать приложение для тестирования телекоммуникационного оборудования.



Рисунок 12- ZABBIX

					КП-09.03.01 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

## 1.4 Функциональные требования

Выбор сценария тестирования.

Установление соединения с тестируемым устройством.

Ведение баз данных устройств: добавление/удаление устройства, изменение конфигурации, хранение логов/результатов тестирования.

Запуск сценария тестирования.

Сохранение результатов тестирования.

## 1.5 Выводы по главе

В результате анализа были сформулированы спецификация требования к разрабатываемой системе.

Поставлены цели и задачи для разработки ПО.

Рассмотрены аналоги.

Так же были выбраны инструменты для разработки ПО.

Изучено оборудование, которое мы будем тестировать.

					КП-09.03.01 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 Этапы проектирования

Варианты использования приложения представлены на рисунке 5.



Рисунок 5 – Диаграмма вариантов использования приложения

## 2.1 Архитектура приложения

Архитектура приложения изображена на рисунке 6.

**Client** – клиентское приложение, написанное на фреймворке VueJS

**Server** – серверное приложение на ASP.NET Core (5)

**Data base** – локальный файл базы данных, который хранится на том же сервере, где расположено серверное приложение

**Библиотека тестирования** – скрипт, написанный моим напарником, на языке программирования Python, который так же хранится на сервере. Интеграция .net и python осуществляется через запуск фонового процесса с передачей необходимых параметров для скрипта и чтением соответствующего окна вывода

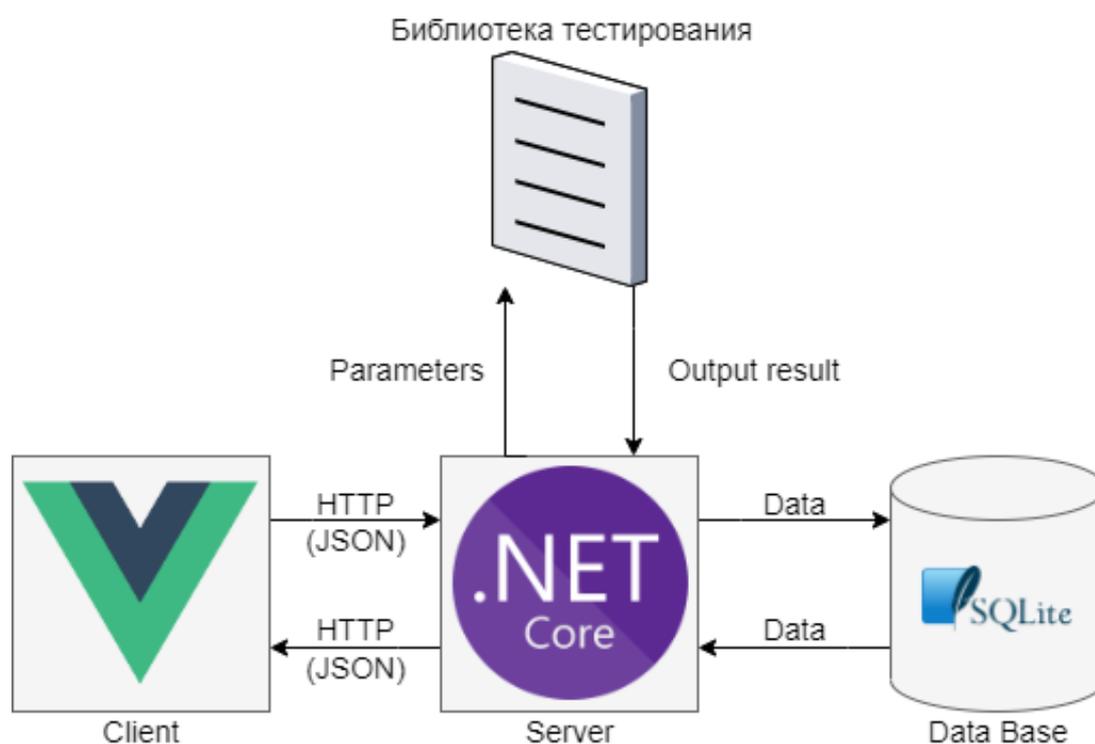


Рисунок 6- Архитектура приложения

## 2.2 Прецеденты

Прецеденты описывают поведение пользователя в системе. Для визуализации взаимодействия компонентов между собой, а также с пользователем используется диаграмма последовательностей, отражающая динамическую модель системы.

**Прецедент 1.** Добавить устройство.

**Цель сценария:** Добавить информацию о модеме в БД.

**Предусловия:** Пользователь нажал кнопку “Добавление устройства”.

**Основной сценарий:** После того, как пользователь ввел всю необходимую информацию об устройстве (название, адрес, порт, логин) и нажал кнопку “ОК”, то устройство добавляется в БД.

Добавление / отображения устройства представлены на рисунке 7.

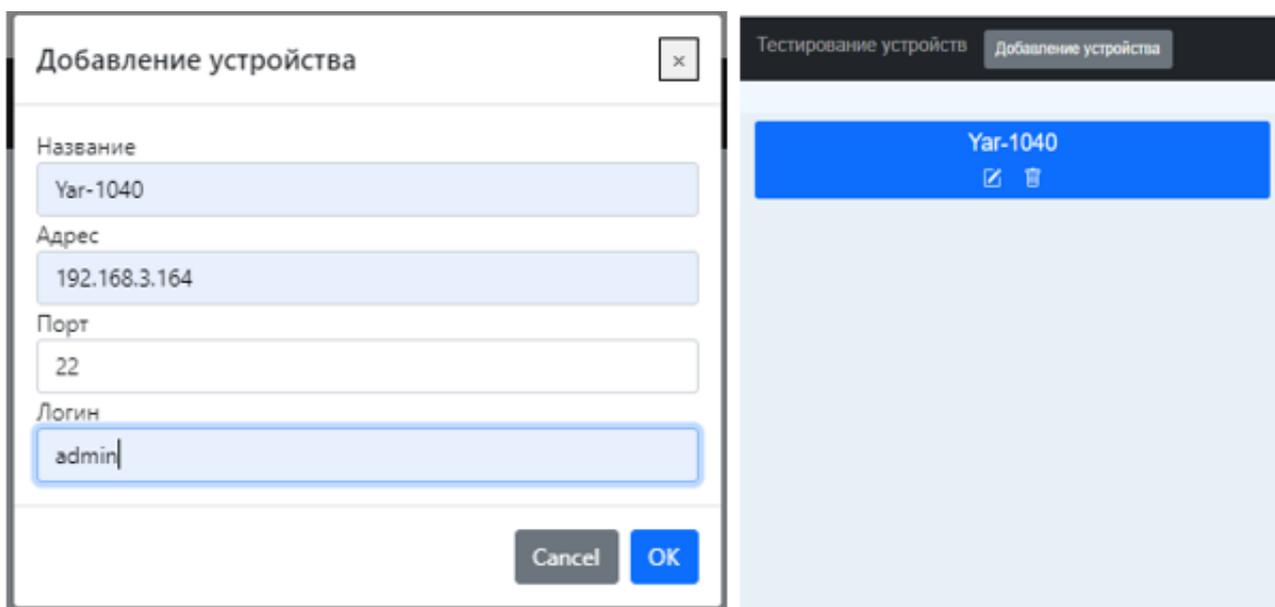


Рисунок 7 – Добавление / отображения устройства

**Прецедент 2.** Отредактировать устройство.

**Цель сценария:** Обновить информацию об устройстве.

**Предусловия:** Пользователь нажал кнопку “Редактировать устройство”.

**Основной сценарий:** После нажатия на кнопку для редактирования, всплывает окно, в котором можно изменить данные об устройстве, при нажатии на кнопку “ОК” срабатывает функция Update и данные обновляются в БД.

На рисунке 8 представлено редактирование устройства.

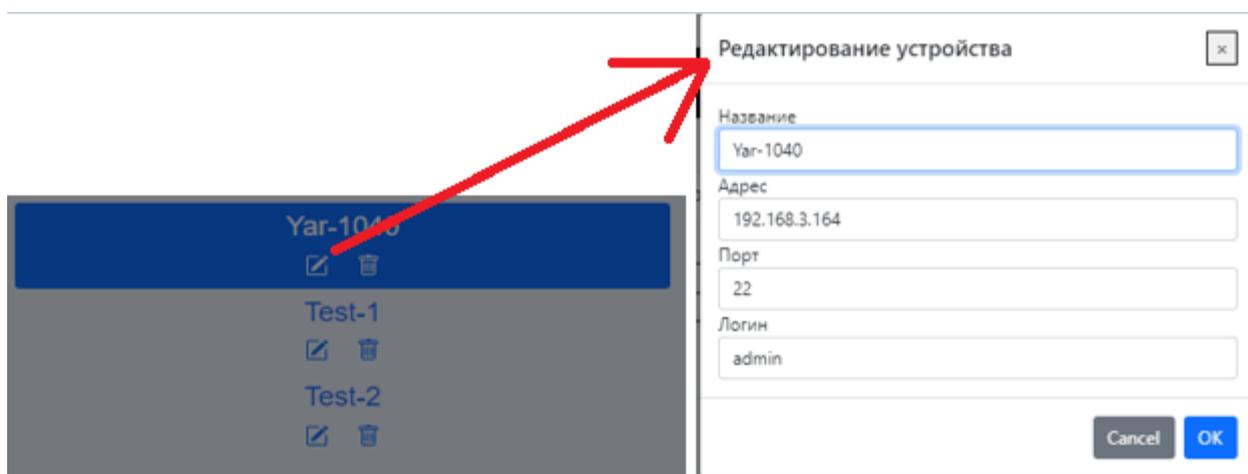


Рисунок 8 – Редактирование устройства

**Прецедент 3.** Удаление устройства.

**Цель сценария:** Удалить устройство из БД.

**Предусловия:** Пользователь нажал кнопку “Удаление устройства”.

**Основной сценарий:** После того, как пользователь нажал на кнопку удаления, всплывает окно подтверждения, при нажатии на кнопку “Удалить” срабатывает функция Delete и данные удаляются из БД.

На рисунке 9 представлено удаление устройства.

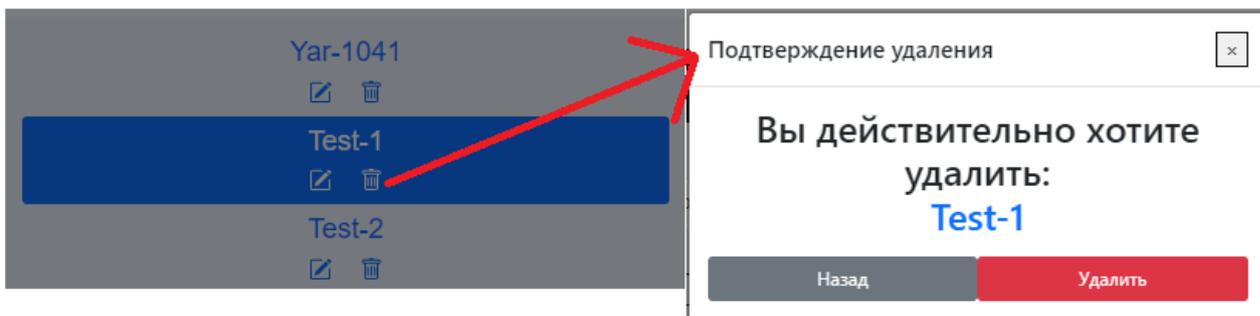


Рисунок 9 – Удаление устройства

**Прецедент 5.** Установка соединения с устройством.

**Цель сценария:** Установить соединение с устройством и вывести сообщение о том, удалось ли это сделать или нет.

**Предусловия:** Пользователь нажал кнопку “Установка соединения”.

**Предусловия:** Пользователь нажал на кнопку “Установка соединения”.

**Основной сценарий:** Отправляются данные о модеме параметрами в библиотеку тестирования, и исходя, от того, что вернет нам библиотека выведет сообщение о удачном или неудачном подключении к устройству.

**Условие 1.** Соединение установлено.

**Условие 2.** Соединение не установлено.

На рисунке 10 представлено удачное или неудачное подключение.

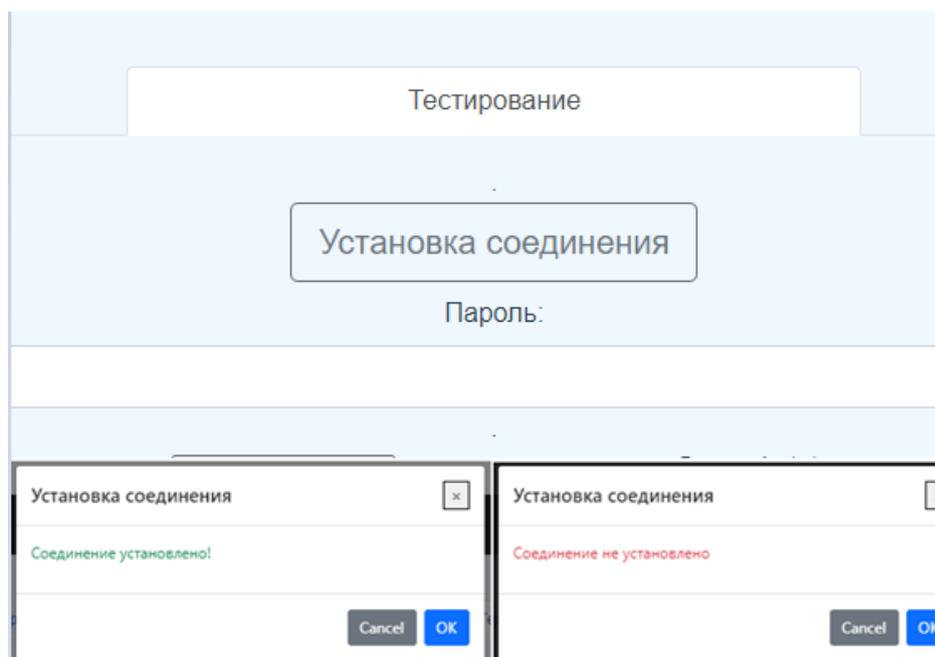


Рисунок 10 – Удачное или неудачное подключение

## Прецедент 6. Тестирование устройства.

**Цель сценария:** Произвести тестирование устройства и получить результат.

**Предусловия:** Пользователь выбрал сценарий, ввел пароль от устройства и начал тестирование.

**Основной сценарий:** После установки соединения с устройством, можно начать тестирование, для этого пользователю нужно выбрать txt файл со сценарием тестирования, ввести пароль от устройства, затем нажать кнопку “Начать тестирование”, вся информация о модеме (адрес, порт, логин, пароль) параметрами передается в Python script и результат отобразится в вкладке “История” и запишется в базу данных.

На рисунке 11 представлено тестирование устройства.

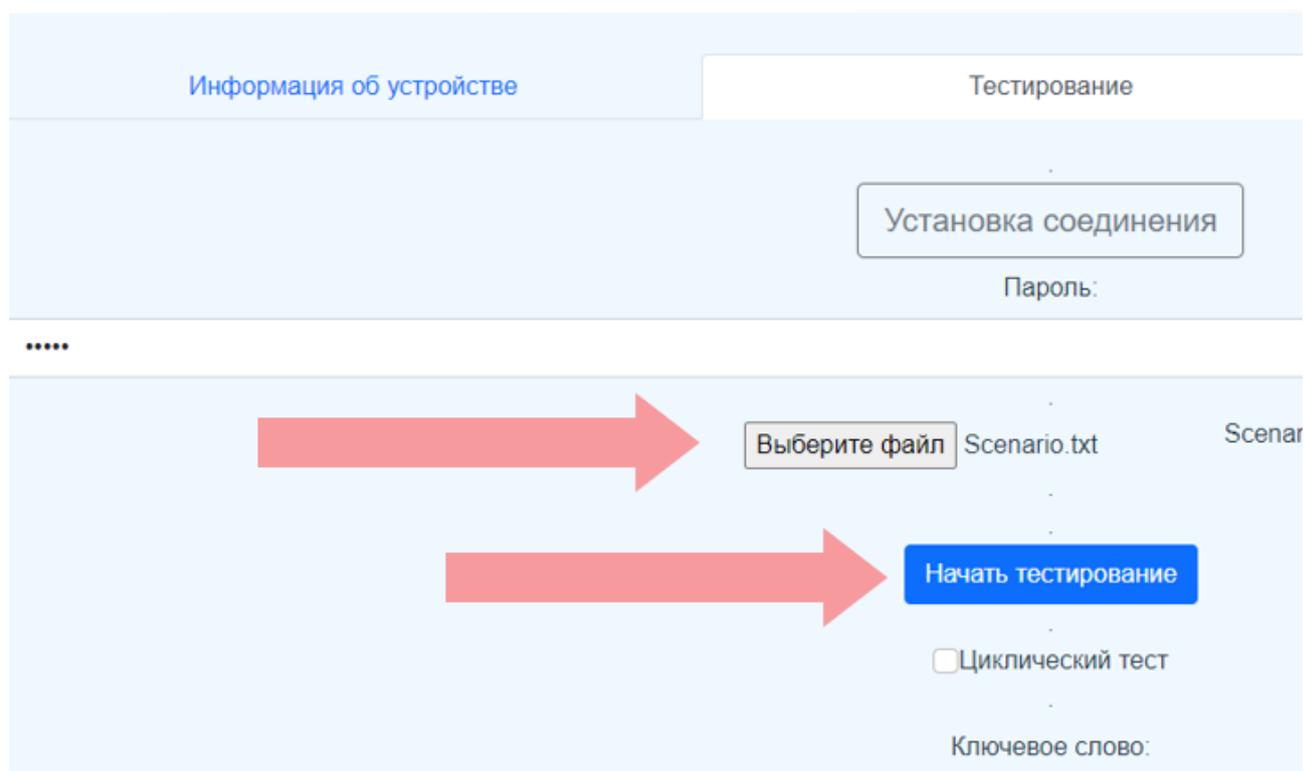


Рисунок 11 – Тестирование устройства

## Прецедент 7. История тестирования.

**Цель сценария:** Просмотр результата тестирования.

**Основной сценарий:** Пользователь зашел во вкладку “История” и нажал на кнопку “Результаты тестирования”.

На рисунке 12 представлен результат тестирования.

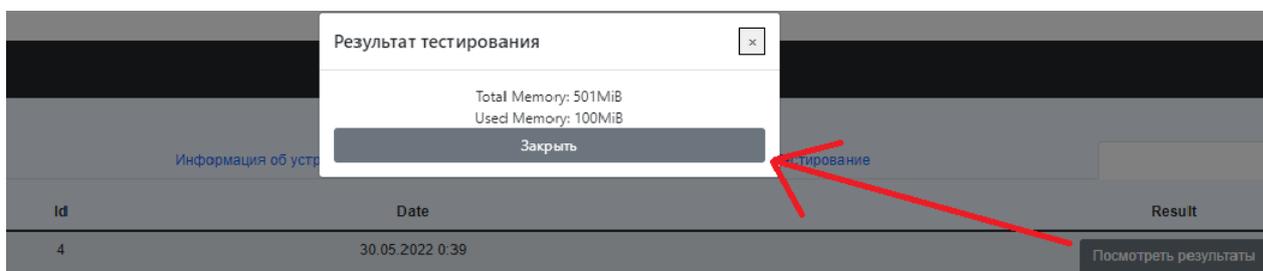


Рисунок 12 – Результат тестирования

Проводится тестирование модема по данному сценарию. Пользователю нужно выбрать сценарий, ввести пароль от устройства.

## 2.3 База данных

Для хранения информации о модеме, результатов тестирования используется база данных.

На рисунке 13 представлена диаграмма базы данных.

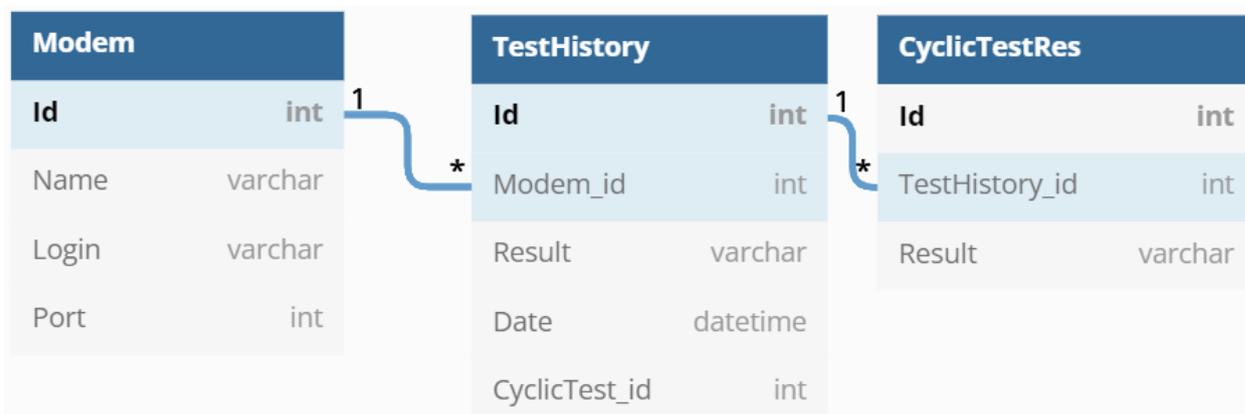


Рисунок 13 – Диаграмма базы данных

База данных состоит всего из трех основных таблиц. Modem хранит в себе информацию об устройстве. TestHistory сохраняет результат тестирования модема и дату, когда устройство было проверено. CyclicTestRes требуется, чтобы записывать результаты циклических тестов. Для варианта циклических тестов формируется N записей типа CyclicTestRes. Поле CyclicTest\_id таблицы TestHistory используется для связи со всеми результатами циклического теста (поле id таблицы CyclicTestRes). Поле TestHistory\_id указывает номер теста в последовательности тестов и изменяется от 1 до N.

## 2.4 Выводы по главе

В соответствии с техническим заданием:

- предложена архитектура системы и структура базы данных;
- с помощью нотации диаграмм последовательностей задокументированы наиболее сложные отношения в системе, более детально проработано взаимодействие объектов;
- разработан вывод результата тестирования с использованием ключевого слова.

					КП-09.03.01 ПЗ	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

### 3 Программная реализация

В процессе программной реализации были решены следующие задачи:

- выполнена верстка страниц;
- реализованы основные функции для клиент-серверного взаимодействия;
- реализована навигация между модулями.

#### 3.1 Главный экран Web-приложения

При запуске приложения мы видим главный экран, после добавления устройства у нас будет доступ к меню навигации, меню навигации создано с помощью компонента из The Bootstrap – tabs. Tabs – виджет панелей локального содержимого с вкладками. Компонент вкладок построен на основе navs и cards и обеспечивает полное управление навигацией по вкладкам. Главный экран и навигация представлены на рисунках 14 и 15.

					КП-09.03.01 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

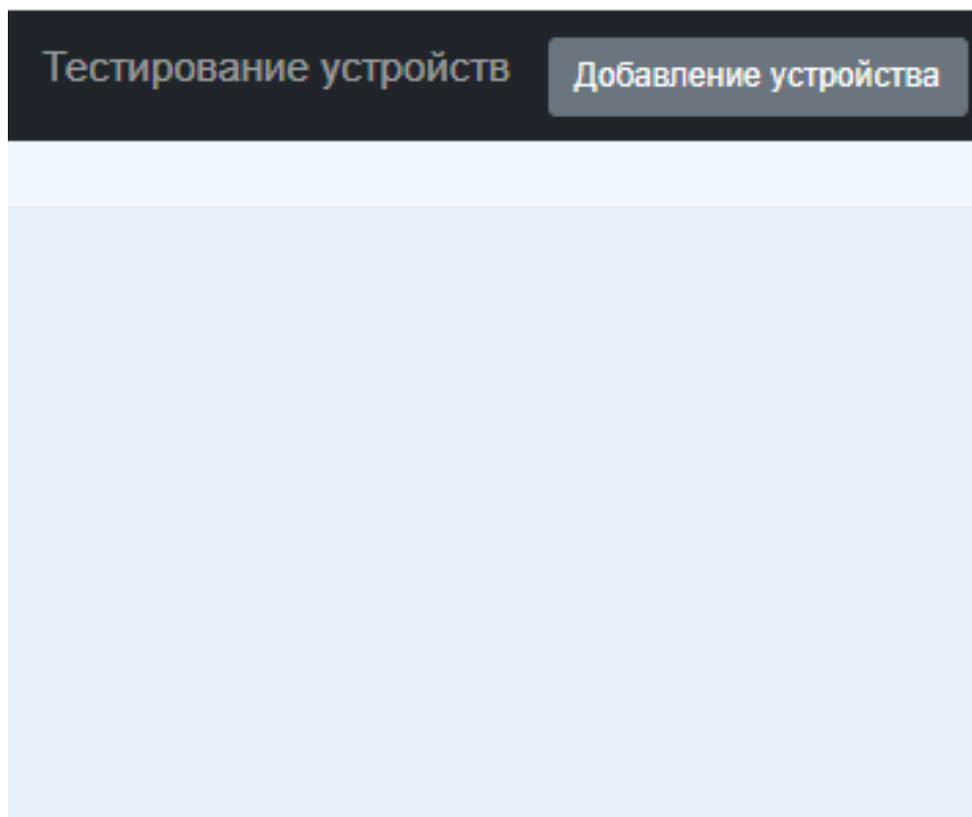


Рисунок 14 – Экран при запуске приложения

На рисунке 15 представлена навигация в приложении, реализованная с помощью tabs.



Рисунок 15 – Навигация в приложении реализована с помощью tabs

### 3.2 Вкладка информации об устройстве

При открытии модуля информации об устройстве данные загружаются при get-запросе из базы данных и отображаются в таблице.

Устройство можно удалить или отредактировать, нажав соответствующую кнопку возле устройства (смотри рисунок 16).

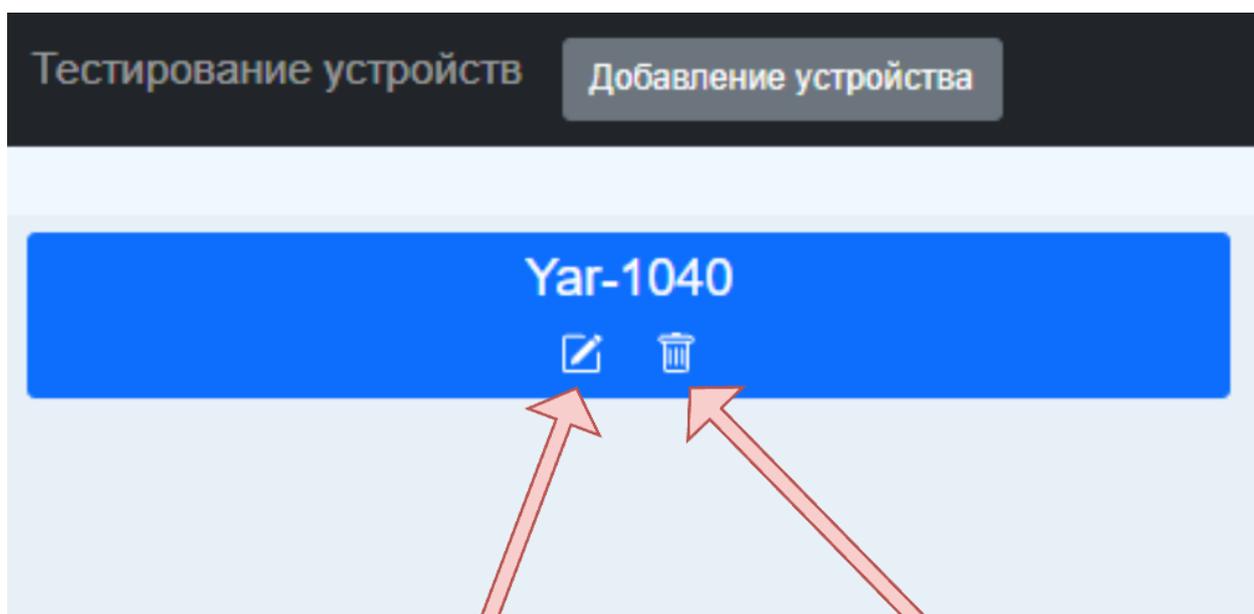


Рисунок 16 – Редактирование / удаления устройства

На рисунке 17 представлена вкладка информации об устройстве.



Рисунок 17 – Вкладка информации об устройстве

### 3.3 Модуль тестирования

При открытии модуля тестирования можно увидеть кнопку установки соединения, кнопку для выбора файла со сценарием, поле для ввода пароля и поле для ввода ключевого слова. Ключевое слово требуется для того, чтобы вывести нужную информацию с результата тестирования, если ключевое слово не введено, то пользователь получает весь результат тестирования.

На рисунке 18 представлена вкладка информации устройства.

Рисунок 18 – Вкладка тестирования устройства

### 3.4 Модуль просмотра истории тестирования

При открытии модуля истории можно увидеть дату, когда было проведено тестирования устройства и кнопку для просмотра результата тестирования. Детальнее вкладка истории тестирования представлена на рисунке 19.

Информация об устройстве		Тестирование	История
Id	Date		Result
8	30.05.2022 14:19		Посмотреть результаты
9	30.05.2022 14:20		Посмотреть результаты
10	30.05.2022 14:20		Посмотреть результаты
11	30.05.2022 14:21		Посмотреть результаты

Рисунок 19 – Вкладка истории тестирования

На рисунке 20 представлен результат тестирования устройства.

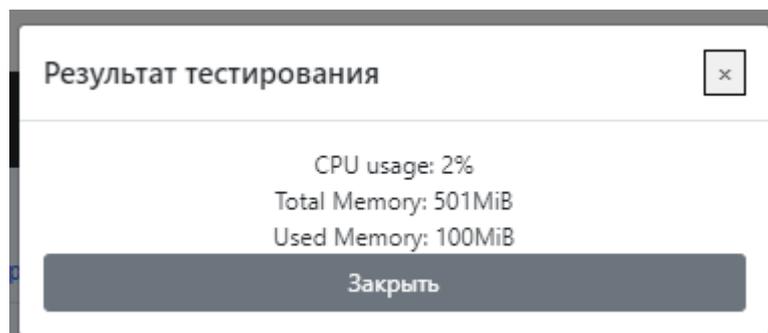


Рисунок 20 – Результат тестирования устройства

## 3.5 Функции для работы с базой данных

### 3.5.1 Функция GetAll

Для того, чтобы взять всю информацию об устройствах из БД используется функция GetAll и отобразить её на Vue.

На рисунке 21 представлена функция GetAll

```
[HttpGet]
[Route("GetAll")]
0 references
public ModemTestingModel[] GetAll()
{
    _context.Database.EnsureCreated();
    return _context.Modems.Select(m => new ModemTestingModel
    {
        Id = m.Id,
        Name = m.Name,
        Address = m.Address,
        Login = m.Login,
        Port = m.Port,
        TestHistory = m.TestHistory.Select(h => new TestHistoryItem
        {
            Id = h.Id,
            Date = h.Date,
            Result = h.Result
        }).ToArray()
    }).ToArray();
}
```

Рисунок 21 – Функция GetAll

					КП-09.03.01 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

### 3.5.2 Функция Post

Для добавления устройства в базу данных используется функция Post, после ввода всей информации об устройстве вызывается функция и вся необходимая информация загружается в БД.

На рисунке 22 представлена функция Post

```
public int Post([FromBody] ModemTestingModel model)
{
    _context.Database.EnsureCreated();
    var modem = new Modem
    {
        Address = model.Address,
        Name = model.Name,
        Login = model.Login,
        Port = model.Port
    };
    _context.Set<Modem>().Add(modem);
    _context.SaveChanges();
    return modem.Id;
}
```

Рисунок 22 – Функция Post

### 3.5.3 Функция Update

Для того, чтобы отредактировать информацию об устройстве используется функция Update, после ввода новых данных об устройстве вызывается данная функция и вся информация обновляется в БД.

На рисунке 23 представлена функция Update

					КП-09.03.01 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

```

public void Update(int id, [FromBody] ModemTestingModel model)
{
    _context.Database.EnsureCreated();
    var modem = _context.Set<Modem>().SingleOrDefault(m => m.Id == id);
    modem.Name = model.Name;
    modem.Login = model.Login;
    modem.Address = model.Address;
    modem.Port = model.Port;
    _context.Set<Modem>().Update(modem);
    _context.SaveChanges();
}

```

Рисунок 23 – Функция Update

### 3.5.4 Функция Delete

Для того, чтобы удалить данные об устройстве из БД используется функция Delete, после нажатия соответствующей кнопки вызывается функция и все данные удаляются из БД.

На рисунке 24 представлена функция Delete

```

public void Delete(int id)
{
    _context.Database.EnsureCreated();
    var modem = _context.Set<Modem>().SingleOrDefault(m => m.Id == id);
    _context.Set<Modem>().Remove(modem);
    _context.SaveChanges();
}

```

Рисунок 24 – Функция Delete

### 3.5.5 Функция сохранения результата тестирования в БД

Для сохранения даты, когда тестировалось устройство, результата тестирования и id модема, используется функция TestHistory.Add

На рисунке 25 представлена функция сохранения результата тестирования в БД.

					КП-09.03.01 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

```

var newTestHistory = new TestHistory
{
    ModemId = model.ModemId,
    Date = DateTime.Now,
    Result = result
};
modem.TestHistory.Add(newTestHistory);
_context.SaveChanges();
return new TestHistoryItem
{
    Id = newTestHistory.Id,
    Date = newTestHistory.Date,
    Result = newTestHistory.Result
};

```

Рисунок 25 – Функция сохранения результата тестирования в БД

### 3.6 Пример тестирования устройства

#### 3.6.1 Передача данных Python скрипту

Чтобы передать нужные параметры в библиотеку тестирования, используется конструкция, которая показана на рисунке 26, во второй строке указывается путь до скрипта, а в 4-ой передаются нужные данные (путь до библиотеки, путь до сценария, адрес, логин, пароль и порт) для корректной работы программы.

```

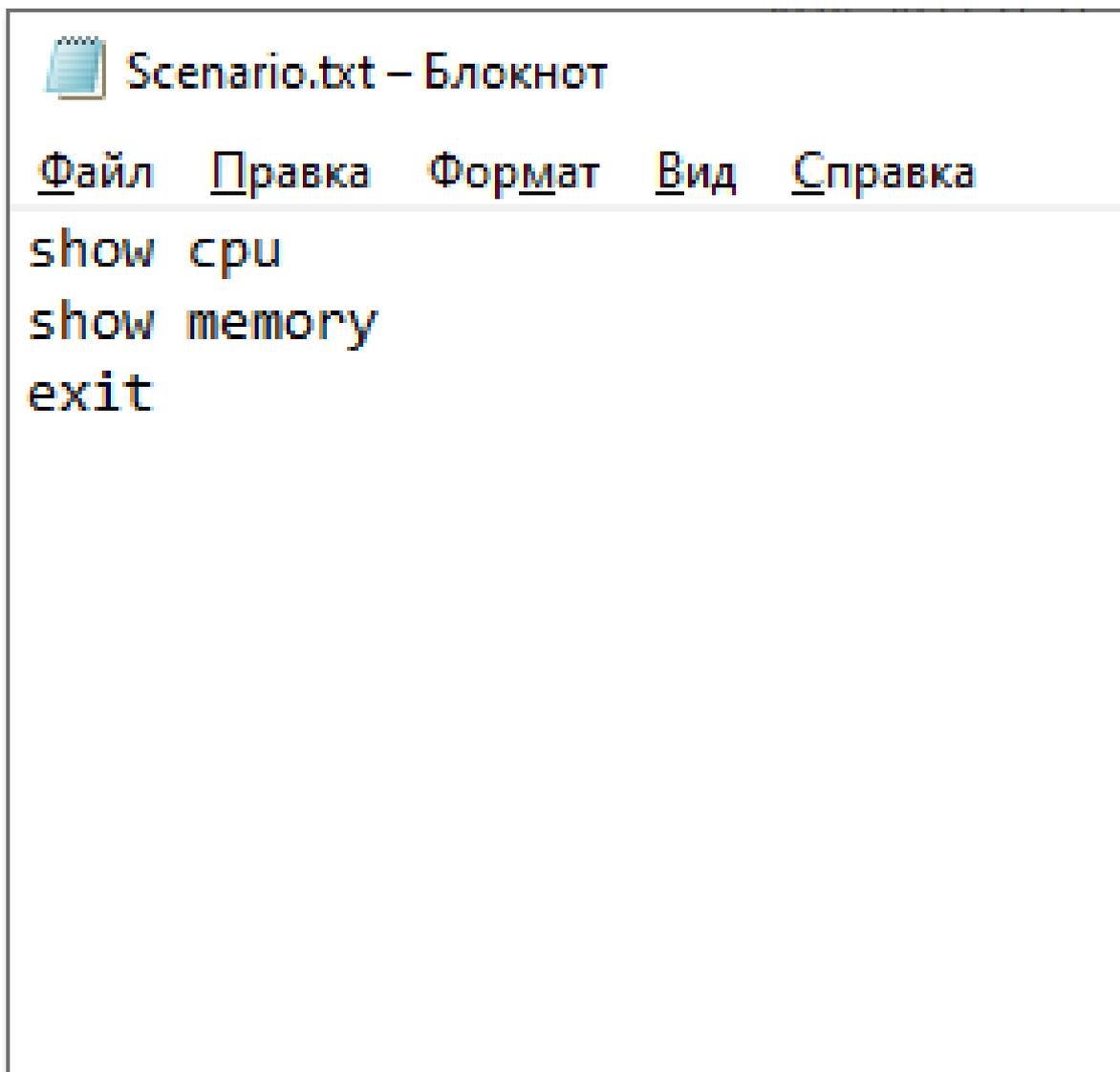
1 var process = new ProcessStartInfo();
2 var path = @"test.py";
3 process.FileName = @"C:\Users\Aseno\AppData\Local\Programs\Python\Python310\python.exe";
4 process.Arguments = $"{path}" "{fileName}" "{model.Address}" "{model.Login}" "{model.Password}" "{model.Port}";
5 if (!string.IsNullOrEmpty(model.Keyword))
6 {
7     process.Arguments = process.Arguments + $" {model.Keyword}";
8 }

```

Рисунок 26 – Конструкция для передачи данных в библиотеку тестирования через параметры

### 3.6.2 Сценарий тестирования и пример тестирования

Простой пример сценария тестирования показан на рисунке 18, предназначен он для того, чтобы просмотреть загруженность процессора модема и сколько оперативной памяти всего и сколько занято. Также можно загрузить большой сценарий и по ключевому слову вывести результат. Например, если нам нужно узнать только загруженность процессора, то в поле “Ключевое слово” нужно ввести CPU и в результате тестирования отобразится только строка с загруженностью процессора.



```
Scenario.txt – Блокнот
Файл  П_равка  Ф_ормат  В_ид  С_правка
show cpu
show memory
exit
```

Рисунок 27 – Пример сценария

На рисунке 28 представлен ввод ключевого слова.

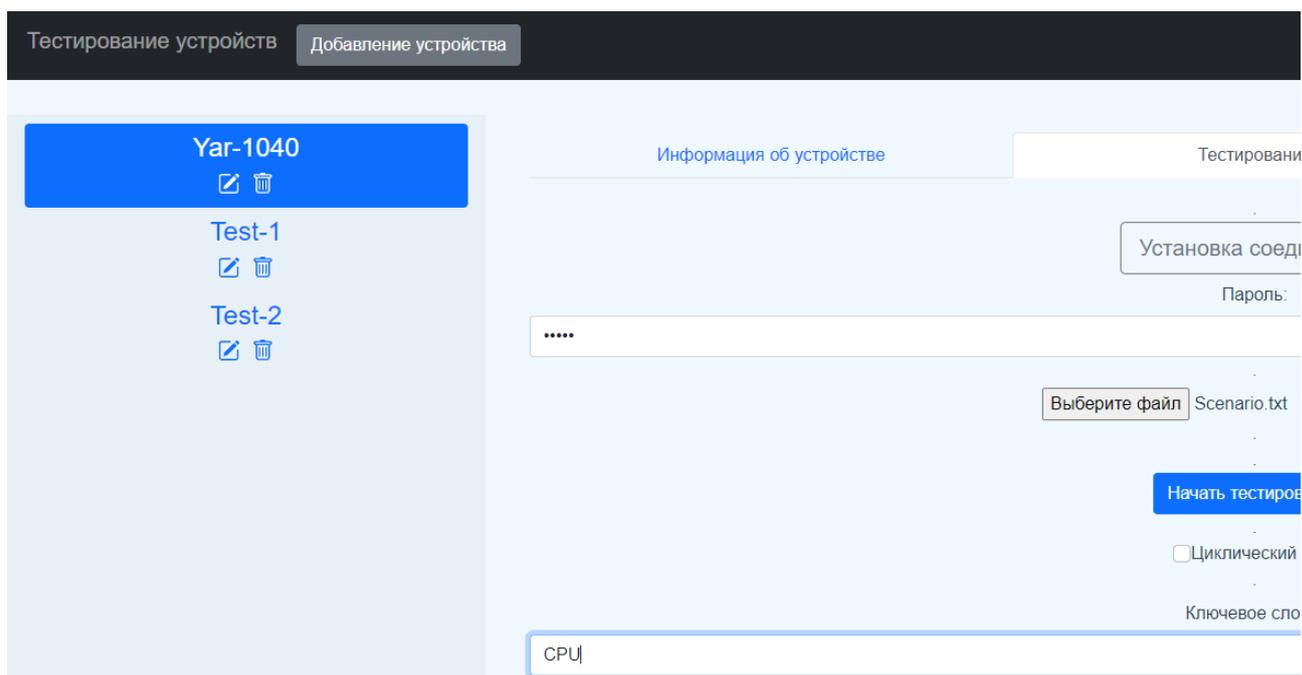


Рисунок 28 – Ввод ключевого слова

На рисунке 29 и 30 представлены результаты тестирования с ключевым словом и без ключевого слова.

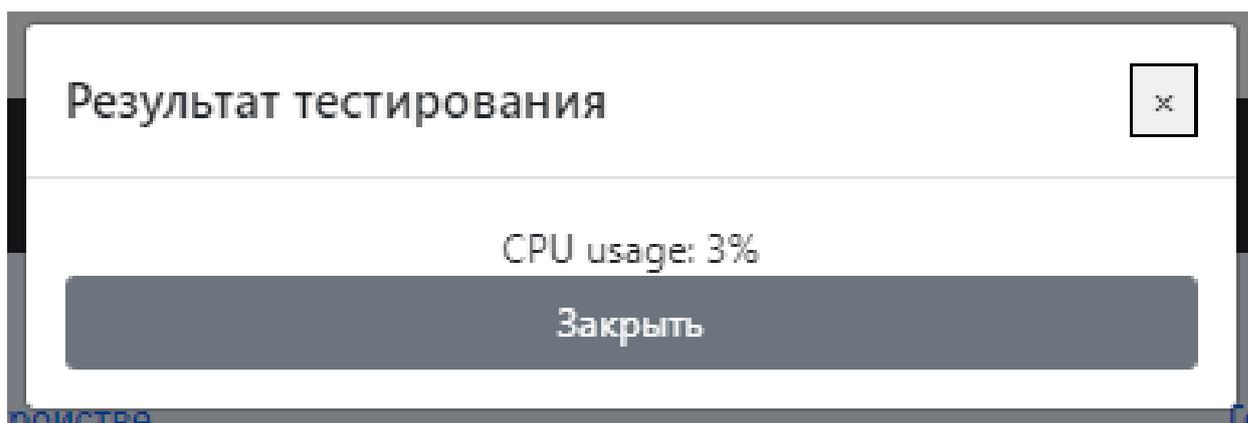


Рисунок 29 – Результат тестирования с ключевым словом

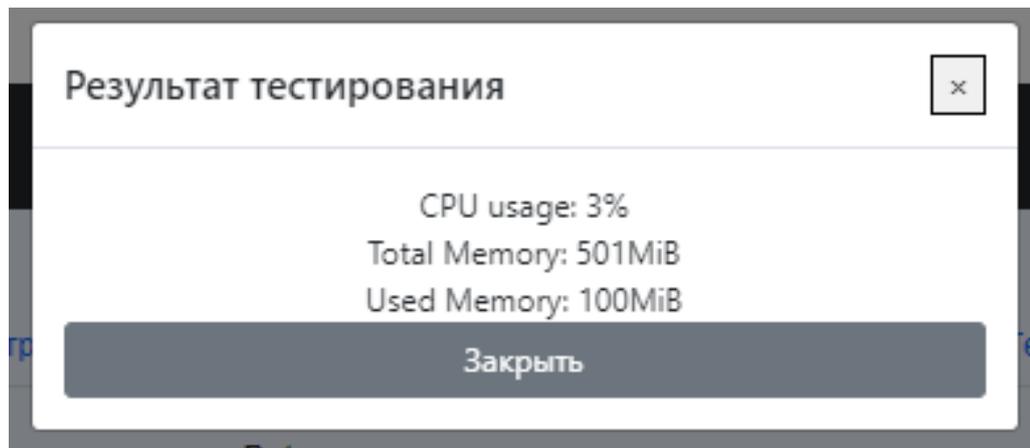


Рисунок 30 – Результат тестирования с ключевым словом

					КП-09.03.01 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

## Заключение

В результате выполнения выпускной квалификационной работы была изучена предметная область.

Был сформулирован ряд требований, предъявленных к Web-приложению. На основе сформулированных требований были определены технологии разработки. Приложение было написано с использованием фреймворков VueJS и .net 5 (ASP.NET Core 5.0 Web API) с компактной встраиваемой СУБД - SQLite.

В результате было разработано Web-приложение, которое позволяет упрощенно и быстро тестировать телекоммуникационные устройства.

В дальнейшем требуется расширение функционала, обработка некоторых ошибок. Также необходимо будет улучшить интерфейс.

					КП-09.03.01 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

## Список сокращений

БД – база данных;

CSS – Cascading Style Sheets;

HTML – Hypertext Markup Language;

JS – JavaScript.

СУБД – Система управления базами данных.

					КП-09.03.01 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

## Список использованных источников

1. Vue.js [Электронный ресурс]: – Режим доступа: <https://vuejs.org/>
2. Zabbix [Электронный ресурс]: – Режим доступа: <https://www.zabbix.com/ru>
3. ASP.NET Core [Электронный ресурс]: – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet>
4. Python [Электронный ресурс]: – Режим доступа: <https://www.python.org/>
5. Клиент – серверная архитектура [Электронный ресурс]: - Режим доступа: <https://zametkinapolyah.ru/servera-i-protokoly/o-modeli-vzaimodejstviya-klient-server-prostymi-slovami-arxitektura-klient-server-s-primerami.html>
6. СТУ 7.5–07–2021 [Электронный ресурс]: - Режим доступа: <https://about.sfu-kras.ru/docs/8127/pdf/119063>

					КП-09.03.01 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

Кафедра вычислительной техники  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой  
О.В.Непомнящий  
подпись                      инициалы, фамилия  
«06»                      06 2022г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 – Информатика и вычислительная техника  
код – наименование направления

Система тестирования телекоммуникационного оборудования  
тема

Руководитель

  
подпись, дата

Старший преподаватель  
должность,  
ученая степень

И. Н. Рыженко  
инициалы, фамилия

Студент

 24.06.2022  
подпись, дата

Г.М. Турзунов  
инициалы, фамилия

Нормоконтролер

  
подпись, дата

Старший преподаватель  
должность,  
ученая степень

И.Н. Рыженко  
инициалы, фамилия

Красноярск 2022