Министерство просвещения ПМР

ГОУ СПО «Тираспольский техникум информатики и права»

**КУРСОВАЯ РАБОТА**

**по дисциплине:** Прикладное программирование

**на тему:** Разработка информационной системы учета компьютерной техники

**Выполнил обучающийся**

Репида Владислав Александрович

**Специальность 09.02.03**

Программирование в

компьютерных системах

**Руководитель**

преподаватель информационных

дисциплин

Балашова Юлия Владимировна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись руководителя)

Тирасполь, 2023

**Оглавление**

[Введение 3](#_Toc134464927)

[1. Раздел первый: Теоретические аспекты программной разработки 4](#_Toc134464928)

[1.1. Область применения программного обеспечения - описание 4](#_Toc134464929)

[1.2. Изучение существующих программных продуктов 4](#_Toc134464930)

[1.3. Достоинство и недостатки существующих разработок 6](#_Toc134464931)

[1.4. Технология программной разработки 8](#_Toc134464932)

[1.5. Формулирование общей задачи разработки программы 10](#_Toc134464933)

[2. Раздел второй: Практическая реализация программного обеспечения 11](#_Toc134464934)

[2.1. Требования к программе 11](#_Toc134464935)

[2.2. Руководство для программистов 12](#_Toc134464936)

[2.3. Руководство для пользователей программного обеспечения 13](#_Toc134464937)

[Заключение 16](#_Toc134464938)

[Список использованных источников 17](#_Toc134464939)

# Введение

Данная курсовая работа посвящена разработке информационной системы учета компьютерной техники. Современные технологии играют важную роль в облегчении работы в различных сферах деятельности, в том числе и в учете компьютерной техники. Целью данной работы является изучение материалов по учету компьютерной техники, выявление актуальности разработки информационной системы для упрощения учета и управления компьютерной техникой в организации.

Для достижения поставленной цели в работе рассматриваются следующие задачи: изучение учебной справочной литературы и нормативных документов по теме исследования; сбор, систематизация и обработка цифрового и теоретического материала; разработка информационной системы учета компьютерной техники с целью упрощения организации учета компьютерной техники.

В работе используются такие методы, как аналитический метод, метод сравнения, метод анализа и синтеза. Для решения поставленных задач использовались учебная и справочная литература, первичные документы, регистры аналитического и синтетического учета компьютерной техники.

Результатом данной работы будет разработка информационной системы учета компьютерной техники, которая позволит упростить учет компьютерной техники в организации и повысить эффективность управления ресурсами компьютерной техники.

# 1. Раздел первый: Теоретические аспекты программной разработки

## 1.1. Область применения программного обеспечения - описание

В современном бизнесе компьютеры и другая электронная техника являются важнейшими инструментами работы. Компании тратят значительные средства на покупку, обслуживание и модернизацию компьютерной техники, и важно иметь информационную систему учета, которая позволит контролировать расходы и обеспечивать эффективное использование ресурсов.

Информационная система учета компьютерной техники предназначена для управления компьютерным оборудованием в организации. Эта система позволяет контролировать инвентарь, отслеживать изменения состояния и технических характеристик компьютеров, обнаруживать неисправности и своевременно проводить ремонт или замену оборудования.

Одним из главных преимуществ информационной системы учета компьютерной техники является возможность упрощения процесса управления и обслуживания компьютеров. Автоматизация учета компьютерной техники позволяет снизить вероятность ошибок и повысить качество обслуживания, сократить расходы на обслуживание и замену оборудования. Кроме того, информационная система учета компьютерной техники может помочь в принятии решений по модернизации и обновлению парка компьютеров, на основе анализа данных о состоянии оборудования и его использовании.

## 1.2. Изучение существующих программных продуктов

На сегодняшний день существует множество информационных систем учета компьютерной техники. Некоторые из них являются платными, другие - бесплатными, однако все они имеют схожие функциональные возможности.

Одной из наиболее популярных и функциональных систем учета компьютерной техники является программное обеспечение Spiceworks. Эта система позволяет автоматизировать процессы учета компьютерной техники, отслеживать изменения и обновления ПО и оборудования, а также предоставляет множество отчетов о состоянии техники и работы сотрудников.

Spiceworks - это бесплатное программное обеспечение для управления ИТ-активами, которое включает в себя инвентаризацию компьютеров, сетевого оборудования, программного обеспечения, а также управление заявками на техническую поддержку.

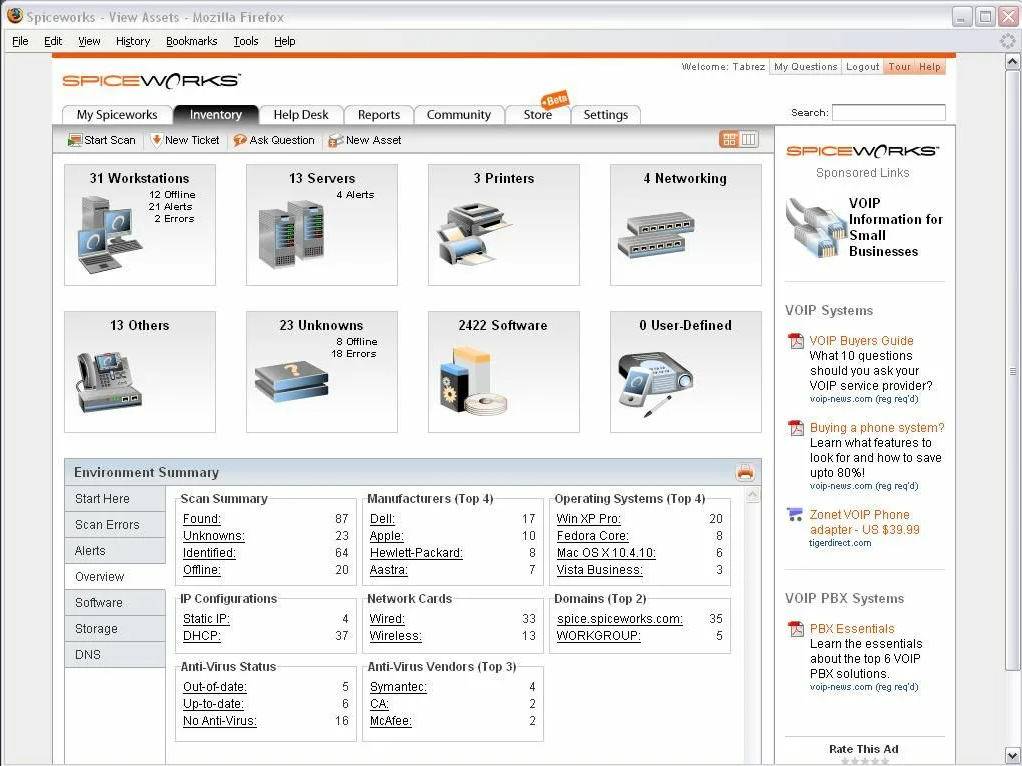


Рис. 1 - Spiceworks

Еще одним примером системы учета компьютерной техники является IT Asset Management от SolarWinds. Она позволяет автоматизировать учет и управление оборудованием, программным обеспечением и лицензиями, предоставляет возможность оптимизировать бюджет на закупку новой техники и обновление ПО.

IT Asset Management (ITAM) от SolarWinds - это интегрированное программное обеспечение для учета и управления компьютерной техникой и программным обеспечением.

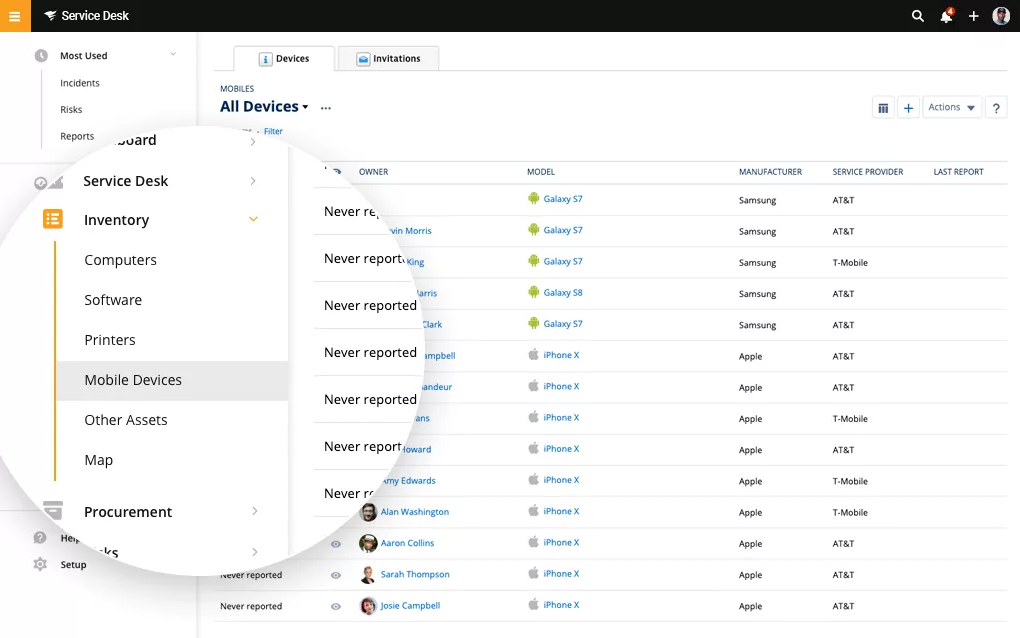


Рис. 2 - IT Asset Management (ITAM)

Важно отметить, что каждая информационная система учета компьютерной техники имеет свои преимущества и недостатки, и выбор конкретной системы зависит от потребностей и возможностей компании.

## 1.3. Достоинство и недостатки существующих разработок

После изучения и анализа существующих аналогов программ, были выявлены как достоинства, так и недостатки.

**Основные достоинства программы “Spiceworks”:**

* Бесплатное использование, что делает Spiceworks доступным для небольших компаний и организаций с ограниченным бюджетом.
* Легкость установки и использования, что позволяет начать использовать программное обеспечение практически сразу после установки.
* Удобный интерфейс и интуитивно понятные функции, что делает Spiceworks доступным для пользователей без технических знаний.
* Интеграция с почтовыми сервисами для автоматического создания заявок на техническую поддержку.
* Возможность мониторинга сетевого оборудования и серверов.

**Основные недостатки программы “Spiceworks”:**

* Ограниченный функционал по сравнению с платными аналогами.
* Отсутствие поддержки пользователей, которые не говорят на английском языке.
* Проблемы с производительностью при работе с большим количеством устройств.
* Возможны проблемы с безопасностью при использовании в больших организациях.
* Отсутствие опций автоматического обновления программного обеспечения.

**Основные достоинства программы “IT Asset Management”:**

* Простота использования: ITAM имеет интуитивно понятный пользовательский интерфейс, что упрощает работу с ним и ускоряет процесс управления компьютерной техникой.
* Широкий функционал: ITAM позволяет отслеживать аппаратное и программное обеспечение на компьютерах, управлять лицензиями на ПО, проводить аудиты ПО и многое другое. Благодаря этому, ITAM может использоваться как для управления компьютерами в небольших компаниях, так и для больших корпораций.
* Интеграция с другими продуктами SolarWinds: ITAM может легко интегрироваться с другими продуктами SolarWinds, такими как Network Performance Monitor, Server & Application Monitor и другие, что позволяет создать полностью интегрированную систему управления ИТ-инфраструктурой.
* Гибкость настройки: ITAM предоставляет возможность настройки и конфигурации почти всех его функций, что делает его гибким инструментом для управления ИТ-инфраструктурой любого масштаба.

**Основные достоинства программы “IT Asset Management”:**

* Высокая стоимость: ITAM относится к категории дорогих программных продуктов, что может быть непозволительно для небольших компаний.
* Сложность настройки: ITAM имеет множество настроек и параметров, что может создать определенные сложности при первоначальной настройке системы.
* Ограниченная поддержка ПО: ITAM не всегда может обеспечить полную поддержку для всего используемого ПО, что может создать определенные проблемы при управлении лицензиями.
* Требует больших ресурсов: ITAM может потребовать значительных ресурсов для работы на больших корпоративных сетях, что может повлиять на производительность системы.

## 1.4. Технология программной разработки

Технология разработки программного обеспечения (ПО) является важным аспектом в процессе создания любого приложения. В моем проекте, где я работал над клиентской частью, я использовал такие технологии, как Vue 3 и TypeScript, применяя ООП подход.

Vue 3 - это современный JavaScript-фреймворк, который используется для создания пользовательских интерфейсов (UI). Он предлагает удобный и легкий способ для создания веб-приложений. Vue 3 имеет множество функций, таких как виртуальный DOM, реактивность, компонентный подход, и многие другие. В моем проекте я использовал Vue 3 для создания пользовательского интерфейса и манипуляции данными.

TypeScript - это язык программирования, который расширяет возможности JavaScript, добавляя типы. Он обеспечивает более строгую типизацию, что упрощает отладку кода, повышает его читаемость и понимаемость. Я использовал TypeScript в проекте для повышения качества кода и упрощения его поддержки в дальнейшем.

ООП - это подход в программировании, в котором реализуется принцип наследования и создания классов. Это упрощает создание и поддержку кода, позволяет избежать дублирования кода и упрощает его понимание и изменение в дальнейшем. В моем проекте я использовал ООП, чтобы создать классы, которые были связаны с логикой пользовательского интерфейса и обработки данных.

Комбинация этих технологий позволила мне создать высококачественный и простой в поддержке код, а также предоставила богатый набор инструментов для создания пользовательских интерфейсов и обработки данных в веб-приложении.

Так как система изначально задумывалась как локальная для хранения данных был использовал Local Storage.

Local storage - это механизм хранения данных в браузере, который позволяет сохранять данные между сеансами работы пользователя. В нашем случае, я использовал его для хранения данных о технике, чтобы их можно было легко получить и обновить при каждом запуске приложения.

В качестве сборщика модулей я использовали Vite, который позволял нам эффективно управлять зависимостями и собирать проект в единую сборку. Мы также использовали npm для управления пакетами и их зависимостями.

Для верстки интерфейса я использовал препроцессор Less, который позволил нам ускорить процесс написания CSS и сделать его более читаемым и удобным для сопровождения.

Наконец, я использовали систему контроля версий Git, чтобы управлять версиями нашего кода. Git также позволил мне легко откатываться к предыдущим версиям кода в случае необходимости.

В целом, использование таких технологий и подходов позволило мне создать эффективное, гибкое и легко сопровождаемое приложение для учета компьютерной техники.

## 1.5. Формулирование общей задачи разработки программного обеспечения

Основной задачей данной работы является реализация программы учета компьютерной техники. Программа должна быть просто и удобной для использования.

Основные функции программы:

* Добавление техники с указанием типа
* Удаление техники
* Редактирование информации
* Вывод информации
* Сохранение данных

# 2. Раздел второй: Практическая реализация программного обеспечения

## 2.1. Требования к программе

Программный должена удовлетворять следующим требованиям:

- иметь возможность добавления техники

- иметь возможность редактирования информации о техники

- удалить соревнование/участников

- сохранение данных в файл

- загрузка данных из файла

Минимальная конфигурация аппаратного и программного обеспечения компьютера, на котором будет использоваться программа:

* Процессор: 1,6 ГГц или более быстрый.
* Оперативная память (RAM): 2 Гб или больше.
* Жесткий диск: 500 Мб свободного места или больше.
* Видеокарта: поддержка графического интерфейса с разрешением экрана не менее 1024x768 пикселей.
* Операционная система: Windows 7 SP1 или более поздняя версия, macOS 10.12 или более поздняя версия, Ubuntu 18.04 или более поздняя версия.
* Браузер: последняя версия Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari или Opera.

Программный продукт был разработан с использованием современных технологий, таких как фреймворк **Vue3** и интегрированной среды разработки **VSCode**, что позволило обеспечить высокую производительность и удобство при работе над проектом.

Конфигурация аппаратного и программного обеспечения компьютера, на котором велась разработка программного продукта:

- процессор *AMD Ryzen 5 2600 Six-Core Processor 3.40 GHz*;

- оперативная память 16 ГБ типа *DDR4*;

- видеокарта *NVIDIA GeForce 1660*, видеопамять 6 ГБ;

- операционная система *Windows 11 Pro* 64 бит.

- браузер *YandexBrowser*

## 2.2. Руководство для программистов

Программный продукт представляет собой веб-сайт для учета компьютерной техники. В разработанном программном продукте реализовано базовая функциональность для данного типа программы.

Класс ***EquipmentModel*** является программным представлением компьютерной техники. Данный класс содержит в себе несколько полей: *id* – идентификатор, *type* – тип техники, *name* – наименование, desc? – описание.

Класс ***ComputerModel*** является программным представлением персонального компьютера (ПК). Данный класс содержит в себе несколько полей: *cpu* – модель процессора, *gpu* – модель графического процессора, *ramSize* – объем оперативной памяти (ОЗУ), *storageSize* – объем хранилища (ТБ).

Класс ***MonitorModel*** является программным представлением монитора. Данный класс содержит в себе несколько полей: *resolution* – разрешение монитора, *screenSize* – диагональ экрана в дюймах, *refreshRate* – частота обновления экрана в Гц.

Класс ***LocalStorageService*** (сервис) необходим для удобной работы с браузерным хранилищем localStorage. В конструкторе принимает имя ключа по которому мы обращаемся к хранилищу. Класс имеет 2 метода, *GetAll* и *WriteAll*. Метод *GetAll* позволяет получать данные из хранилища. Метод *WriteAll* позволяет записывать данные в хранилище.

Перечисление (enum) ***EnumTypeOfEquipment*** – необходим для определения типа техники, т.к при сохранении данные хранятся в json строке. Включает в себя *Computer (0)* и *Monitor (1).*

Функция обертка ***LcReactiveArray*** необходимая для связки реактивного состояния vue и localStorage. Принцип работы прост, при любом действии над массивом хранимой техники, мы обновляем данные в localStorage.

Функция ***uuid*** необходима для генерации уникальной строки идентификатора в формате xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx.

Так же в своем проекте я использовал специализированное хранилище состояние из Npm пакета под названием “***pinia***”

**Pinia** - это библиотека для состояния веб-приложений на Vue.js. Она предоставляет простой и интуитивно понятный API для управления состоянием вашего приложения. В отличие от других библиотек управления состоянием, Pinia не использует Vuex и полностью сосредоточена на реактивности Vue 3.

Методы которые используются в хранилище pinia:

* Метод ***AddEquipment*** – добавляет технику в общее хранилище.
* Метод ***EditEquipment*** – редактирует технику.
* Метод ***RemoveEquipment*** – удаляет технику.

В проекте были использованы множество различных компонентов для отображения интерфейса:

* ***ViewEquipments*** – отображает список техники.
* ***BaseEquipmentForm*** – Базовая форма для заполнения основных данных о технике.
* ***ComputerForm*** – Форма для заполнения данных компьютера.
* ***MonitorForm*** – Форма для заполнения данных монитора.
* ***AddEquipment*** – Модальное окно для добавления новой техники.
* ***RemoveEquipment*** – Модальное окно для удаления техники.
* ***EditEquipment*** – Модальное окно для редактирования техники.

Для создания интерфейса использовались так называемые UI компоненты:

***BaseButton*** – Компонент кнопки

***BaseInput*** – Компонент поля ввода

***BaseLabel*** – Компонент подписи

***BaseModal*** – Модальное окно

***BaseSelect*** – Компонент для выбора значения из списка

***BaseTextarea*** – Компонент ввода текста

***TableDescription*** – Таблица для описания каких либо значений

***FormWrap*** – Обертка для формы (заготовка)

Все компоненты были созданы самостоятельно.

## 2.3. Руководство для пользователей программного обеспечения

При открытии веб-сайта будет отображена страница с темой курсовой работы и кнопкой “Добавить технику”.

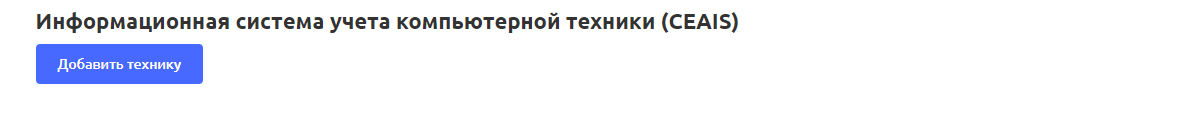


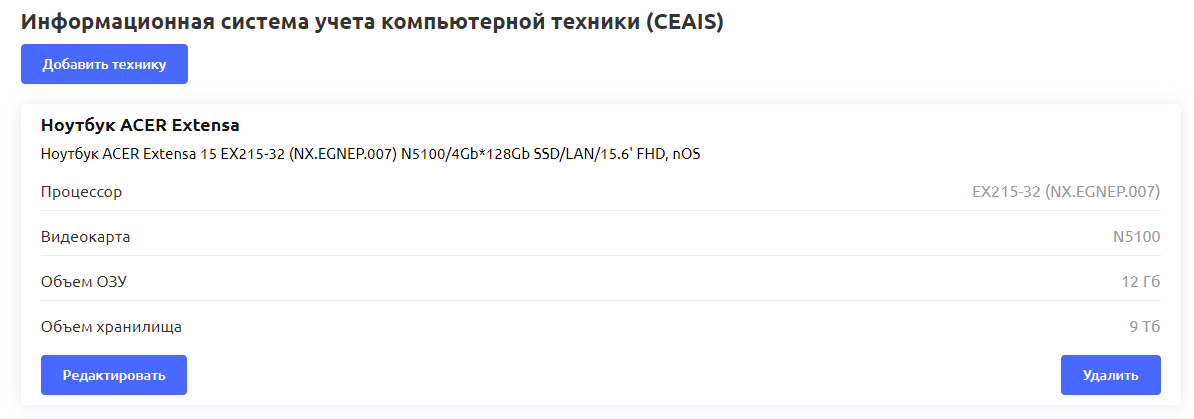
Рис. 3. Страница информационной системы

После нажатия на кнопку “Добавить технику” будет открыто модальное окно с формой для добавления техники.

|  |  |
| --- | --- |
|  |  |

Рис. 5. Формы добавления техники

После добавления техники она появится на экране



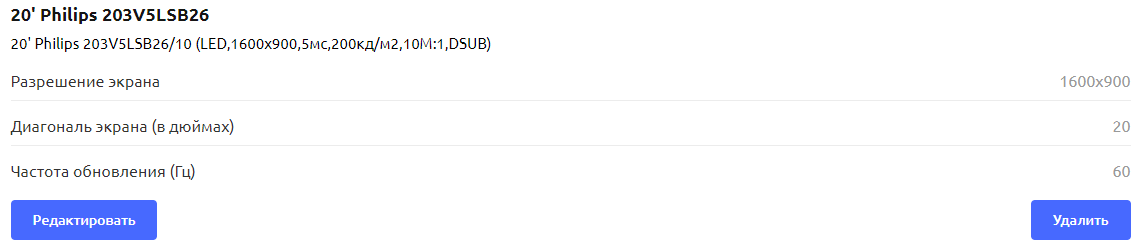


Рис. 2. Отображение добавленной техники

Для редактирования техники нажимаем на кнопку “Редактировать”. Откроется модальное окно с формой редактирования техники.

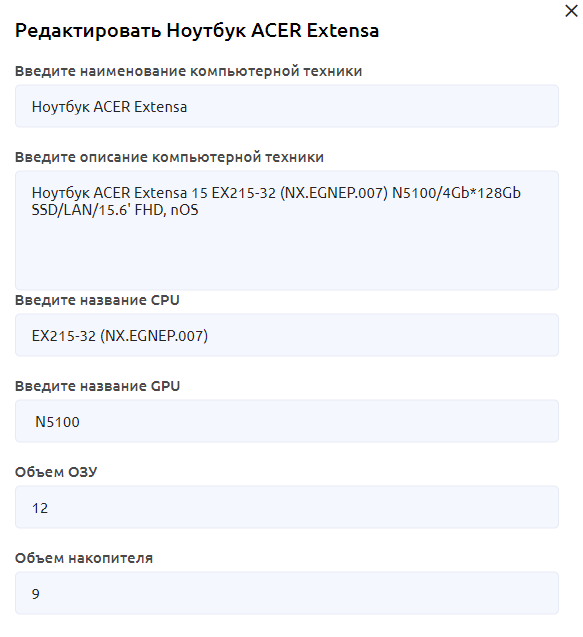


Рис. 3. Форма редактирования техники

Для удаления техники нажимае на кнопку “Удалить”. Откроется модальное окно с подтверждение удаления. Для подтверждения опять нажимае на кнопку “Удалить”, но уже в модальном окне. Если вы передумали удалять, нажимаем на кнопку “Отмена” и модальное окно закроется.

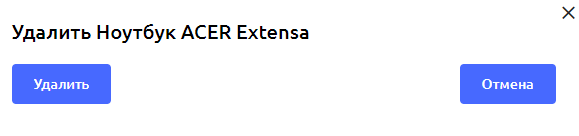


Рис. 2. Модальное окно удаления техники

# Заключение

В результате выполнения курсовой работы на тему "Разработка информационной системы учета компьютерной техники" была проведена глубокая изучение предметной области. Были проведены анализ и оценка существующих аналогов информационных систем учета компьютерной техники, выявлены их недостатки и преимущества.

Были разработаны соответствующие руководства пользователя и программиста.

В ходе выполнения курсовой работы на тему “Разработка информационного программного комплекса реестра участников соревнований” была изучена предметная область. Был произведён анализ существующих аналогов, выявлены недостатки существующих аналогов и принято решение создать собственный программный продукт.

При исследовании предметно области были выявлены требования к программе, а также составлены руководство пользователю и программисту.

Программный продукт удовлетворяет следующим требованиям:

1. создавать технику

2. редактировать технику

3. удалить технику

4. сохранение данных между сеансами в браузере

# Список использованных источник

1. Официальная документация JavaScript: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
2. Официальная документация Vue.js 3: <https://v3.vuejs.org/guide/introduction.html>
3. Официальная документация TypeScript: <https://www.typescriptlang.org/docs/>
4. Официальная документация Less: <https://lesscss.org/>
5. Официальная документация Vite: <https://vitejs.dev/>