

Министерство образования и науки

донецкой народной республики

Государственное образовательное учреждение

высшего профессионального образования

«донецкий национальный университет»

Физико-технический факультет

Кафедра компьютерных технологий

Направление подготовки 09.03.01 Информатика и вычислительная техника

**К защите допустить:**

Зав. кафедрой компьютерных технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.И. Аверин

(подпись)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**ДИПЛОМНАЯ работа**

на тему: **«Автоматизированная система учёта IT-оборудования ГОУ ВПО «ДОННУ»»**

Студент: **Бардашевский Никита Сергеевич**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Научный руководитель: ст. преподаватель **Котенко В.Н.**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Работа представлена на кафедру «\_\_\_» \_\_\_\_ 2021 г. рег. № \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись принявшего)

Донецк 2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10107996)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10107997)

[1.1 Состояние вопроса 6](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10107998)

[1.2 Моделирование бизнес-процессов 10](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10107999)

[1.3 Актуальность и цель работы 15](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108000)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 18](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108001)

[2.1 Описание области применения и исходных данных приложения 18](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108002)

[2.2 Требования к пользовательским интерфейсам 18](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108003)

[2.3 Требования к аппаратным, программным и коммуникационным интерфейсам 30](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108004)

[2.4 Требования к пользователям продукта 31](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108005)

[2.5 Требования к адаптации на месте 31](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108006)

[2.6 Функции продукта 31](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108007)

[2.7 Ограничения 35](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108008)

[3 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ И ПРОГРАММНЫХ СРЕДСТВ 36](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108009)

[3.1 Язык серверных скриптов PHP 36](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108010)

[3.2 Web-framework Laravel 37](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108011)

[3.3 Редактор исходного кода Visual Studio Code 39](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108012)

[3.4 Front-end framework Vue.js 41](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108012)

[3.5 Сервера баз данных MySQL 42](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108012)

4 РАЗРАБОТКА БАЗЫ ДАННЫХ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЁТА IT-ОБОРУДОВАНИЯ ГОУ ВПО «ДОННУ»……...…………………44

[4.1 Выбор и обоснование CУБД 44](#_Toc513067915)

[4.2 Концептуальное и логическое проектирование БД 45](#_Toc513067916)

[4.3 Физическое проектирование БД 48](#_Toc513067917)

[5 РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЁТА IT-ОБОРУДОВАНИЯ ГОУ ВПО «ДОННУ» 55](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108013)

[5.1 Моделирование новых бизнес-процессов 55](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108014)

[5.2 Проектирование баз данных динамического сайта 16](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108015)

[5.3 Архитектура сайта 17](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108016)

[5.4. Описание скриптов динамического сайта 21](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108017)

[5.4.1 Описание скриптов пользовательской части 21](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108018)

[5.4.2 Описание скриптов администраторской части 22](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108019)

[6 ТЕСТИРОВАНИЕ И ВНЕДРЕНИЕ ПРОГРАММНОГО ПРОДУКТА 23](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108020)

[5.1 Аппаратные, системные и программные требования 23](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108021)

[5.2 Описание установки программы 23](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108022)

[5.2.1 Размещение скриптов динамического сайта 23](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108023)

[5.2.2 Импорт баз данных 26](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108024)

[5.3 Руководство пользователя 26](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108025)

[5.3 Описание контрольных примеров 26](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108025)

[5.4 Внедрение 26](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108026)

[7 ОХРАНА ТРУДА НА РАБОЧЕМ МЕСТЕ 27](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108027)

[ЗАКЛЮЧЕНИЕ 28](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108028)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108029)

[ПРИЛОЖЕНИЕ А Экранные формы 30](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108030)

[ПРИЛОЖЕНИЕ Б Фрагменты листинга 31](file:///D:\Алексей\Работа\Пример%20Диплома\4_kurs_Shablon_Diplom_2019.docx#_Toc10108031)

# ВВЕДЕНИЕ

В современном мире IT-технологии являются неотъемлемой частью любой отрасли. От домашнего компьютера до супер-компьютеров, которые занимаются обработкой данных поступающих с бортовых компьютеров космических дронов. Если дома всегда можно ограничиться несложной системой из компьютера, маршрутизатора и телефона, то на больших предприятиях, где необходимо выполнять большое количество различных задач, количество оборудования стремительно растет.   
 Учебные заведения – одни из тех предприятий, где поддерживается сложная информационная сеть. Огромное количество IT-оборудования работает на протяжении многих месяцев или даже лет, беспрерывно обрабатывая различную информацию. Все это оборудование, например: коммутаторы, маршрутизаторы, системные блоки, мониторы, сервера, принтеры; не только поддерживает рабочий процесс в образовательном учреждении но и непосредственно участвует в обучении будущих специалистов.   
 Однако, техника не идеальна, и она часто выходит из строя. Обычно это небольшая поломка, но замена одной маленькой детали может занимать продолжительное время из-за того что не существует удобных систем учёта оборудования.

В настоящей работе разрабатывалось web-приложение, которое позволит удобно и систематично вести учёт IT-оборудования на предприятии. Одновременно с самим оборудованием, оно будет позволять вести учёт деталей, которые используются для ремонта этого оборудования.   
 Например: в случае поломки системного блока, он будет помечен, как блок «в ремонте», а так же будет отмечено, какие детали были списаны для его ремонта. Если каких-то деталей останется мало, система выделит их в системе и позволит пользователю вовремя заметить недостаток на складе.   
 Объектом исследования является учёт оборудования на складе.  
 Предметом исследования является система для учёта IT-оборудования ГОУ ВПО «ДОННУ».   
 Метод исследования – анализ предметной области; изучение систем управления базами данных, для выбора наиболее подходящей для хранения информации об учтённом оборудовании; изучение возможностей языка серверных скриптов PHP и Framework-а Laravel, для создания web-приложения, которое позволит вести удобны учёт оборудования и деталей к нему; изучение возможностей языка разметки HTML, языка каскадных стилей CSS и framework-а Vue.js для построения удобной в пользовании клиентской части приложения.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Состояние вопроса

Автоматизированная система учета – это специальное [прикладное программное обеспечение](http://usu.kz/wiki/prikladnoe_programmnoe_obespechenie.php), с которым можно провести автоматизацию учета предприятия. Автоматизированные системы, в том числе и автоматизированная система учета, позволяют собрать всю информацию по работе организации в единой [базе данных](http://usu.kz/wiki/baza_dannih.php). В такой базе может храниться информация о финансовых потоках компании то есть вестись [бухгалтерский учет](http://usu.kz/wiki/buhgalterskiy_uchet.php), выполненных работах и заказах, осуществляться планирование будущей работы и многое другое[1].   
 В настоящее время в донецком национальном университете для учёта оборудования, запчастей и материалов не существует какой-либо автоматизированной системы учёта. Учётная работа выполняется вручную с использованием различной отчётной документации, которая и позволяет следить за наличием тех или иных материалов и следить за количеством эксплуатируемого рабочего оборудования.   
 Даже при наличии автоматизированной системы учёта, документация занимала бы лидирующее значение в учёте оборудования, однако, с визуальной точки зрения, документация не даёт возможности увидеть картину целиком. Следовательно, из-за бумажного формата, тяжело уследить за тем, какая техника сейчас работает, а какая пришла в негодность; на какую технику списали детали; не было ли списано слишком много деталей. При работе с бумагами их нужно тщательно перепроверять, так как огромную роль играет человеческий фактор. В автоматизированной системе же, человек никогда не сможет списать деталей больше, чем есть у него на складе, так как система не даст ему этого сделать. К тому же, автоматизированная система предоставляет возможность создания некоторой документации, что в разы ускорит и упростит процесс наблюдения за передвижением оборудования и материалов. Ко всему прочему, автоматизированная система, как было сказано в определении, завязана на базе данных предприятия, а значит, она позволяет избежать излишнего дублирования данных в нескольких местах, и строится на уже существующей базе.  
 В настоящее время сложно найти аналоги систем учёта на таких предприятиях, как высшее учебное заведение, однако таких систем очень много на крупных промышленных предприятиях. Рассмотрим несколько аналогов для полноценного понимания вопроса.

1) «Автоматизированная система комплексного учёта теплоресурсов»

В первую очередь, при рассмотрении предметной области систем учёта, необходимо обратить внимание на тривиальный и обыденный пример подобной системы. Самый распространенный вид – система учёта какого-либо ресурса: энерго-, теплоресурса. Такие системы учёта строятся на SCADA системах и обеспечивают не только учёт, но так же анализ и контроль над оборудованием, которое использует или производит конкретный ресурс. Интерфейс следующих систем сложен для восприятия, так как разрабатывается в виде мнемосхемы под определенные предприятия. Автоматизированная система комплексного учёта теплоресурсов разработана компанией «КРУГ» и создана для учета теплоресурсов объектов энергетики, ЖКХ и промышленных предприятий с паровыми, водяными закрытыми и открытыми системами теплоснабжения, а также с системами теплопотребления, выполненными по зависимым и независимым схемам подключения[2].

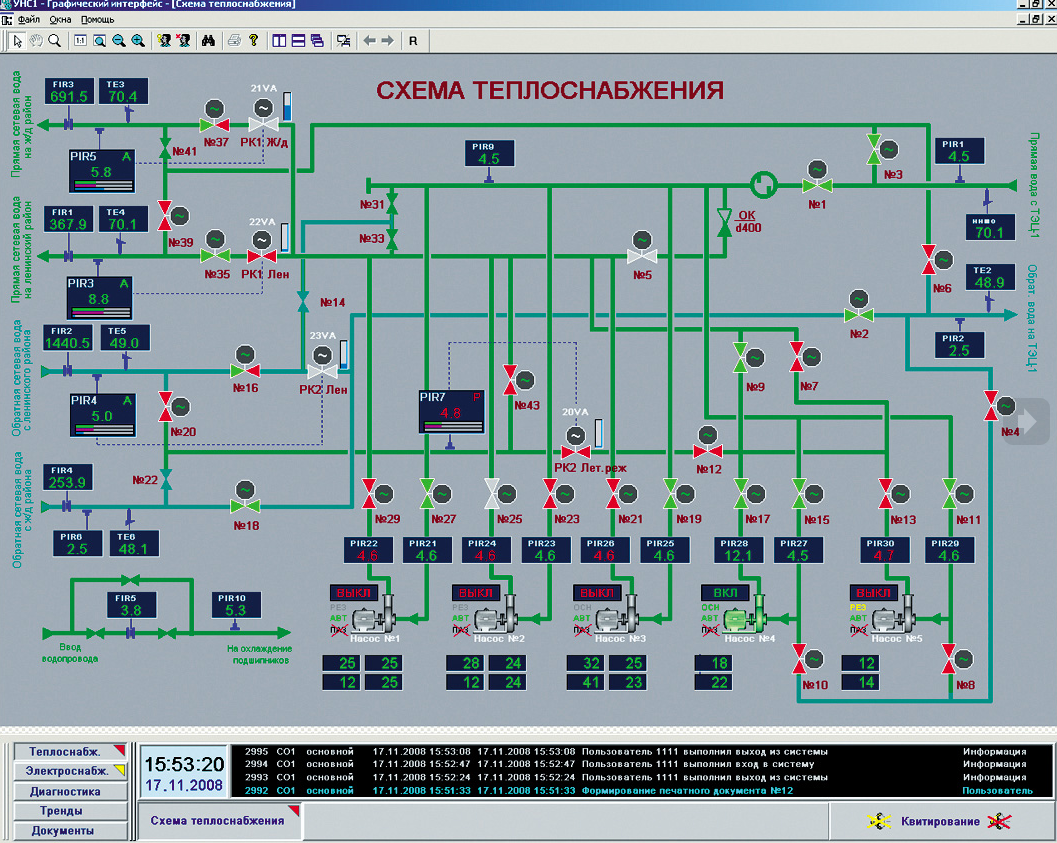


Рисунок 1.1 – Автоматизированная система комплексного учёта теплоресурсов

2) «Автоматизированная система контроля и учёта энергоресурсов»

Несмотря на то, что название настоящей системы созвучно с прошлым аналогом, эта система занимается учётом физически существующих материалов. Под энергоресурсами в этой системе понимаются различные виды топлива. Как можно заметить на рисунке 2, в отличие от системы учёта теплоресурсов, система учёта энергоресурсов построена в табличном виде, что приближает её к виду базы данных и делает её прямым аналогом разрабатываемой в настоящей работе системе. Однако, как можно заметить, за счёт того что интерфейс приложения имеет громоздкий вид – в этой системе сложно разобраться неподготовленному пользователю.

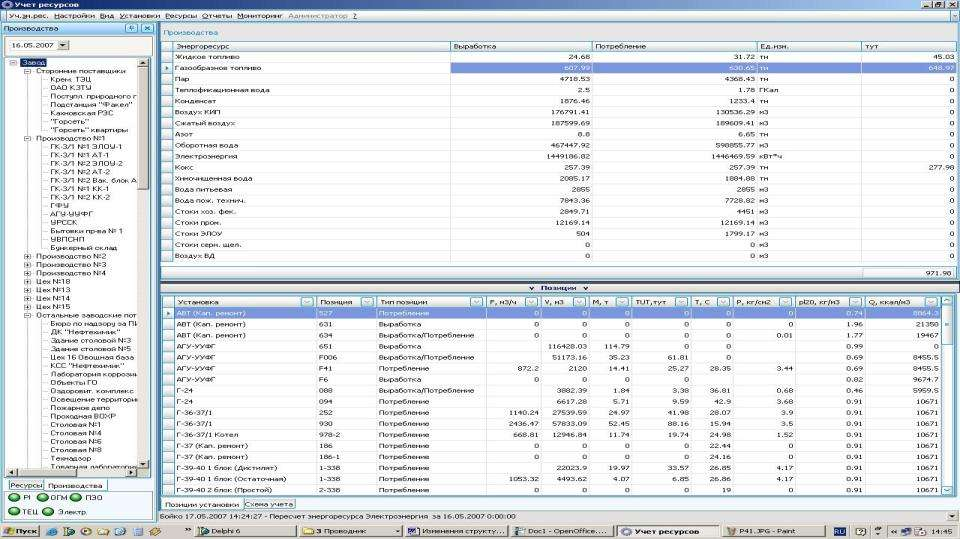


Рисунок 1.2 – Автоматизированная система комплексного учёта энергоресурсов

3) Автоматизированная система управления складом «WMS»

Наиболее близкий к настоящей работе аналог автоматизированной системы это система управления складом «WMS». Разработана система русской компания с использованием ядра «1С». Благодаря предоставляемым 1С ресурсам, программа выглядит удобно и практично, без излишнего нагромождения функционала. В такой системе удобно вести любой простой складской учёт, однако, для задач, где требуется производить какие-либо проверки – рассматриваемая конфигурация 1С не подойдёт. После покупки такую систему необходимо будет тонко настраивать под предприятие, и вносить правки в конфигурацию, тем самым добавляя новый функционал. Не стоит забывать тот факт, что функционал конфигураций 1С ограничен ядром системы. Создать сложную систему валидации данных в такой системе не получится, как и оптимизировать узкие места программы.

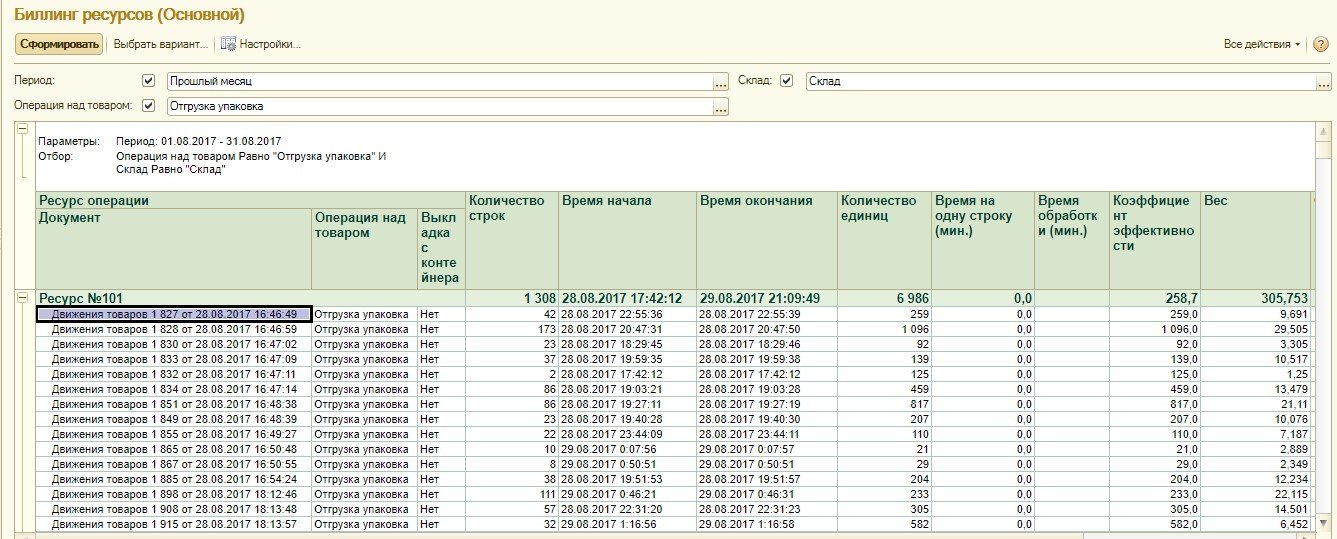


Рисунок 1.3 – Автоматизированная система управления складом «WMS»[3]

Проведя анализ предметной области автоматизированных систем учёта можно сделать вывод, что в настоящее время отсутствие подобного программного обеспечения на предприятиях оказывает сильное влияние на скорость работы предприятия и качество учёта ресурсов. Ко всему прочему, опираясь на предоставленные примеры, можно заметить, что программное обеспечение такого рода делают в виде массивного компьютерного клиента, где сервер занимается лишь хранением информации, а вся её обработка – задача клиента. Системы, построенные таким образом, медленные и сильно зависят от характеристик компьютера, на который они были установлены. Следовательно, web-приложение является наиболее быстрым вариантом «клиента», который не будет зависеть от характеристик компьютера.

## Моделирование существующих бизнес-процессов

Проведем описание бизнес-процессов с помощью методологии IDEF0.

Во время проектирования приложения и базы данных, важно понимать, с какими данными ведется работа, какие данные получаются в итоге и какие функции отвечают за преобразование входящих данных в данные выхода. Для более полного и качественного понимания бизнес-процессов приложения строиться диаграмма по выбранной методологии. Она помогает чётко проследить ход процессов в приложении от начала до конца. На диаграмме IDEF0 отображаются основные функции процесса, входы, выходы, управляющие воздействия и устройства, взаимосвязанные с основными функциями.

Диаграмма IDEF0 имеет всего два элемента: функциональный блок и интерфейсная дуга. Дуги в свою очередь делятся на четыре типа:

1. вход – входные данные (стрелка слева);
2. управление – правила определяющие и разграничивающие работу

системы (стрелка сверху);

1. выход – выходные данные (стрелка справа);
2. механизм – аппарат, выполняющий данную функцию (стрелка снизу).

Составим диаграмму нулевого уровня по методологии IDEF0 (рисунок 1.4).

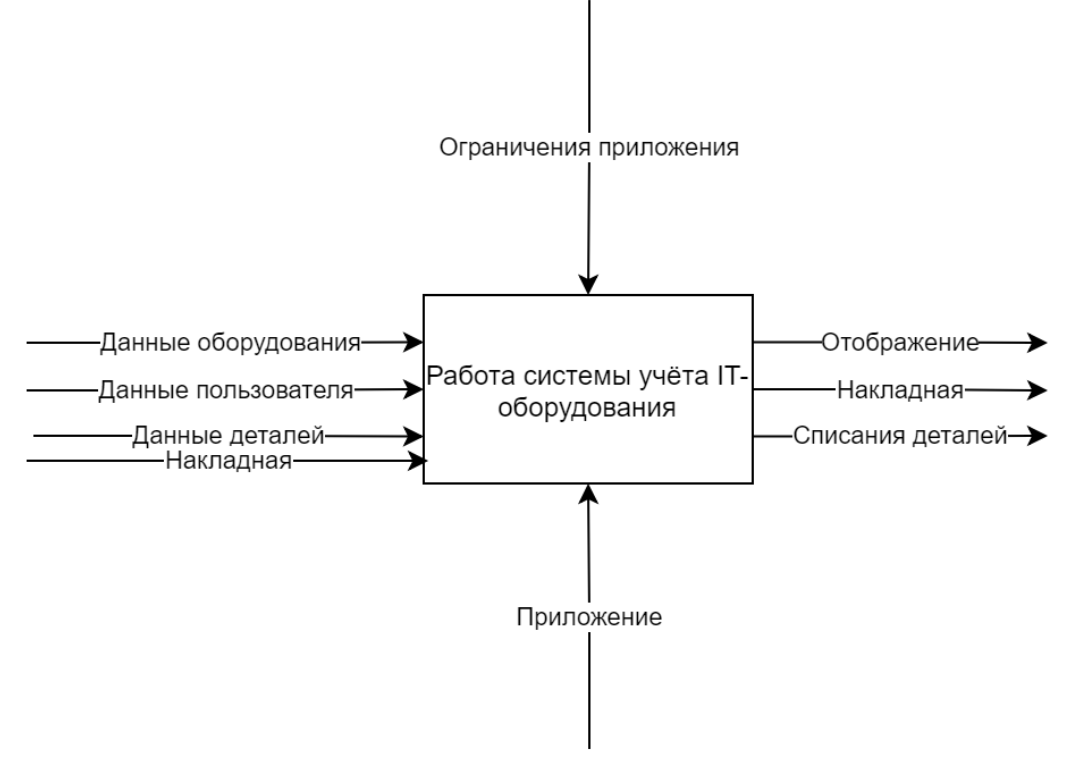


Рисунок 1.4 – Диаграмма нулевого уровня по методологии IDEF0

Верхний уровень рассматривает работу приложения в общих чертах. То есть функциональным блоком является web-приложение в целом, что включает в себя: работу с базой данных, обработка различных запросов к серверу, создание различных отчётов. Входными данными приложения являются: данные сущностей, который существуют в приложении (IT-оборудование, пользователи, детали), так же предусмотрена возможность загрузки одновременно большого количества деталей или оборудования с помощью загрузки накладной. Всем этим управляют ограничения, заложенные в системе, например: ограничение на количество символов в пароле пользователя, или формат номера, или то, что номер накладной должен быть уникальным. Выходными данными являются: отображения списков сущностей в табличном виде; накладные; документы о списании конкретных деталей на ремонт IT-оборудования. Механизм, выполняющий всю работу – web-приложение, где основной логикой занимается сервер, а логикой отображения занимается клиент.

Перейдём на первый уровень и декомпозируем функциональный блок «Работа системы учёта IT-оборудования» на дочерние функциональные блоки (рисунок 1.5).

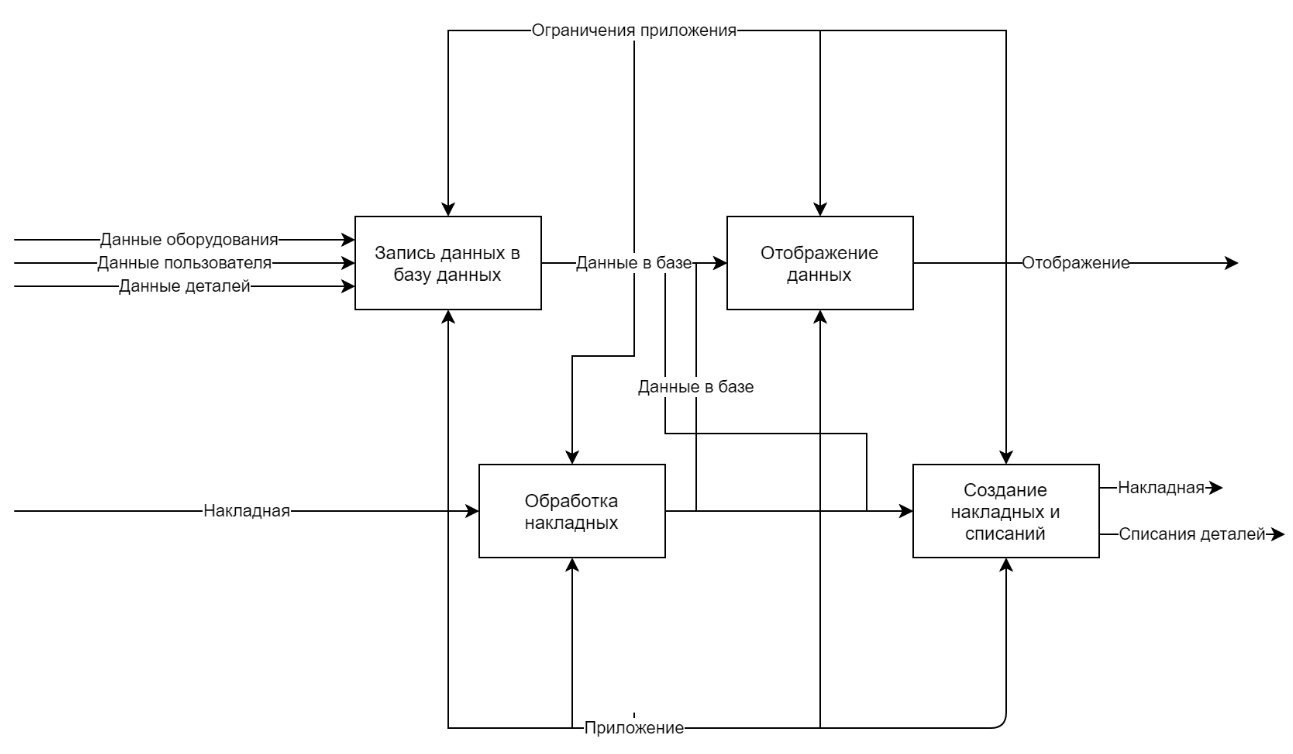


Рисунок 1.5 – Декомпозиция первого уровня по методологии IDEF0

После декомпозиции получаем четыре основных блока: «запись данных в базу данных», который отвечает за обработку запросов на запись данных, их валидацию и непосредственно запись в базу данных; «отображение данных», отвечает за организацию отображения данных в удобном для пользователя виде; «обработка накладных», отвечает за обработку Excel-документов, перед записью их содержимого в базу данных и «создание накладных и списаний», блок отвечающий за создание Excel-документов на основе готовых шаблонов и данных из базы данных. Входными данным для блока «запись в базу данных» являются данные сущностей web-приложения (пользователей, деталей, оборудования), а выходными данным у этого блока являются данные записанные в базу данных. Для блока «обработка накладных» входные данные это накладные в Excel-формате, а выходные – так же данные записанные в базу данных. Блок «отображение» принимает на вход данные из базы и компонует их, создавая View для отображения, которое является выходными данными. «Создание накладных и списаний» - блок, который на вход, принимает данные из базы данных и на их основе формирует Excel-файл, который возвращает пользователю. На работу выше описанных блоков оказывают влияние ограничения приложения. Выполняются функциональные блоки web-приложением. Для понимания работы блока «запись в базу данных», как основного блока системы, проведем его декомпозицию и перейдём ко второму уровню (рисунок 1.6).

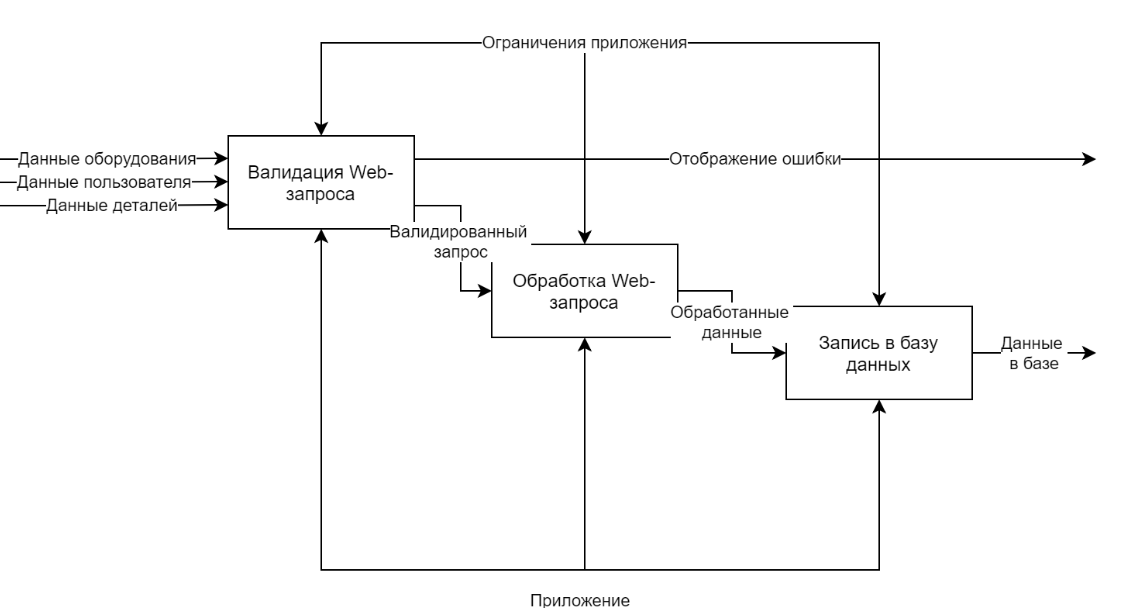


Рисунок 1.6 – Декомпозиция второго уровня по методологии IDF0

На втором уровне есть три функциональных блока. Первые два - это предварительная обработка поступившего web-запроса, а последний – работа с базой данных. Способ обработки запроса для разных видов входящих данных ничем не отличается на концептуальном уровне. Сначала в блок «Валидация Web-запроса» происходит первичная проверка данных содержащихся в запросе и, если их достаточно и они правильно оформлены, валидированный запрос поступает в «Обработку Web-запроса», иначе на обработку отображению будет отправлена ошибка валидации. Следующий функциональный блок занимается проверкой данных перед записью в базу данных, он проверяет необязательные поля, а так же то, что проверить при валидации нельзя, заполняет модель данными. А, после, модель с обработанными данными, записывается в базу данных в функциональном блоке «Запись в базу данных». Управлением занимаются ограничения программного обеспечения, а выполнением занимается web-приложение.

## Актуальность и цель работы

В пункте 1.1 были затронуты следующие преимущества автоматизированной системы в целом, так и если она будет разработана в виде web-приложения как: упрощение учёта оборудования и материалов, защита от ошибок человеческого характера, ускорение работы предприятия за счёт автоматизации учёта, обособление от характеристик рабочей станции. Рассмотрим подробнее каждый из пунктов.   
 За счёт внедрения автоматизированной системы учёта IT-оборудования, можно контролировать эксплуатацию оборудования на предприятии. Если какое-то устройство выйдет из строя – это будет отмечено в системе, и руководство предприятия будет знать, что такое то оборудование на данный момент невозможно использовать. Ко всему прочему, работники, которые занимаются обслуживанием техники, отметят в системе, какие детали были списаны для работы – это позволит заранее докупать недостающие детали, что в свою очередь уменьшит время починки оборудования, что иногда может сыграть большую роль.   
 Во время проведения различных списаний, создания накладных и выгрузка списаний из системы нет места ошибкам человеческого характера. Система занимается анализом данных самостоятельно и блокирует некоторые действия пользователя, например, попытка списать деталей больше чем их наличие на складе закончится неудачей. Ко всему прочему, в подобной системе видно, какой работник, сколько деталей списал, потому списать излишне много деталей и при этом остаться незамеченным не получится.  
 В настоящей работе разрабатывается web-приложение, где основная логика размещена на сервере, а клиент построен в виде сайта, что позволяет использовать программное обеспечение на любой рабочей станции, у которой есть выход в интернет и любой установленный браузер. Работники склада могут зайти на сайт с помощью планшета и внести какие-либо коррективы или ознакомится с настоящим состоянием оборудования на предприятии.   
 Актуальность настоящей работы заключается в упрощении учёта IT-оборудования на предприятии ГОУ ВПО «ДОННУ», а так же в возможности создания простейшего анализа данных. Такое программное обеспечение в разы сократит время и денежные затраты, которые потребляет учёт оборудования в настоящее время.   
 Целью настоящей дипломной работы будет разработка web-приложения(сайта), которое позволит пользователям вносить в базу данных данные об оборудовании, закупленных деталях и пользователях. Проводить операции списания деталей на починку определенного оборудования, а так же проводить выгрузку накладных и списаний в виде Excel-документов.

Задачи, которые необходимо решить для реализации проекта:

1) изучить источники относительно автоматизированных систем учёта оборудования на предприятиях;

2) изучить подходы к реализации автоматизированных систем учёта оборудования и приложениям осуществляющим работу с базами данных;

3) составить предполагаемый функционал приложения;

4) сформировать техническое задание (ТЗ), на основе которого будет разработано приложение;

5) выбрать и изучить технологии для реализации поставленных в ТЗ задач;

6) спроектировать базу данных (БД) на основе технического задания;

7) выбрать подходящую систему для управления базой данных;

8) спроектировать дизайн приложения;

9) разработать приложение;

10) провести тестирование программного продукта;

11) написать руководство пользователя;

Предполагаемый функционал проекта:

1) просмотр списков деталей, оборудования, пользователей;

2) сортировка списков деталей, оборудования, пользователей;

3) фильтрация списков деталей, оборудования, пользователей;

4) просмотр списков списаний деталей на различное оборудование;

5) просмотр деталей, которых не достает на складе, либо осталось слишком мало;

6) логирование действий каждого пользователя в системе;

7) создание пользователей, оборудования и деталей;

8) изменение пользователей, оборудования и деталей;

9) удаление пользователей, оборудования и деталей;

10) загрузка накладных деталей в систему;

11) выгрузка накладных в виде Excel-документов;

12) выгрузка списания в виде Excel-документов.

# 2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## 2.1 Описание области применения и исходных данных приложения

Настоящее web-приложение предназначено для работы с базой данных ГОУ ВПО «ДОННУ» и позволяет производить учёт оборудования и деталей для этого оборудования. Исходными данными для приложения являются:

1. Предустановленные ограничения: максимальное и минимальное количество символов в полях редактируемых пользователем;
2. Настройки по умолчанию: шрифт, цвет, размер шрифта, стиль шрифта;
3. Способы фильтрации и сортировки;
4. Роли пользователей: администратор (чтение, запись, редактирование, создание новых пользователей), модератор (чтение, запись), пользователь (чтение), работник склада (чтение и запись оборудования, списание деталей), работник поставок (чтение и запись деталей);
5. Логин и пароль пользователя-администратора.

## 2.2 Требования к пользовательским интерфейсам

Основные требования к пользовательскому интерфейсу:

1. функциональность;
2. быстрое обучение пользователя;
3. обеспечение высокой скорости работы пользователя;
4. обеспечение защиты от человеческих ошибок;
5. понятность и логичность.

Пользовательский интерфейс должен быть простым. Основным его элементом должны быть таблицы, заполненные необходимыми пользователю данными.

Переключение между страницами приложения должно быть быстрым и осуществляться через пункты меню расположенные в верхней части каждой страницы (за исключением страницы авторизации).   
 На страницах списков должны быть реализованы все фильтры и все возможности сортировки. Должна присутствовать возможность поиска данных по определенному значению.   
 На странице входа должны присутствовать поля для ввода логина и пароля пользователя, кнопка для входа в систему. Регистрация не предусмотрена, так как созданием пользователей занимается администратор.

Данные, загружаемые с помощью Excel-файлов, должны быть оформлены по специальному шаблону. В соответствии с этим шаблоном данные выгружаются из приложения.

Если пользователь не авторизовался в системе, главная страница приложения должна содержать название web-приложения в центре экрана и кнопку перехода к авторизации.

В зависимости от роли пользователя, который зашел в систему, должны изменяться присутствующие на интерфейсе элементы. Например, администратору будет доступно всё, а работнику склада только страница поставок и страница оборудования, а так же кнопка добавления оборудования.

Рассмотрим примерные макеты основных функциональных элементов web-приложения для администратора:

Основные части функционального элемента «страница входа», представленного на рисунке 2.1:

1) название страницы;

2) поле для ввода логина пользователя с подсказкой «Имя пользователя» и иконкой в виде человека;   
 3) поле для ввода пароля с подсказкой «Пароль» и иконкой в виде замка;

4) чек-бокс для запоминания пользователя после входа «Запомнить меня»;

5) кнопка «Вход»;

6) кнопка «Забыли пароль?».



Рисунок 2.1 – Примерный макет страницы «Страница входа»

Основные части функционального элемента «главная страница сайта для неавторизированного пользователя», представленного на рисунке 2.2:

1) текстовая строка «Приложение-Склад»;

2) кнопка «LOGIN».

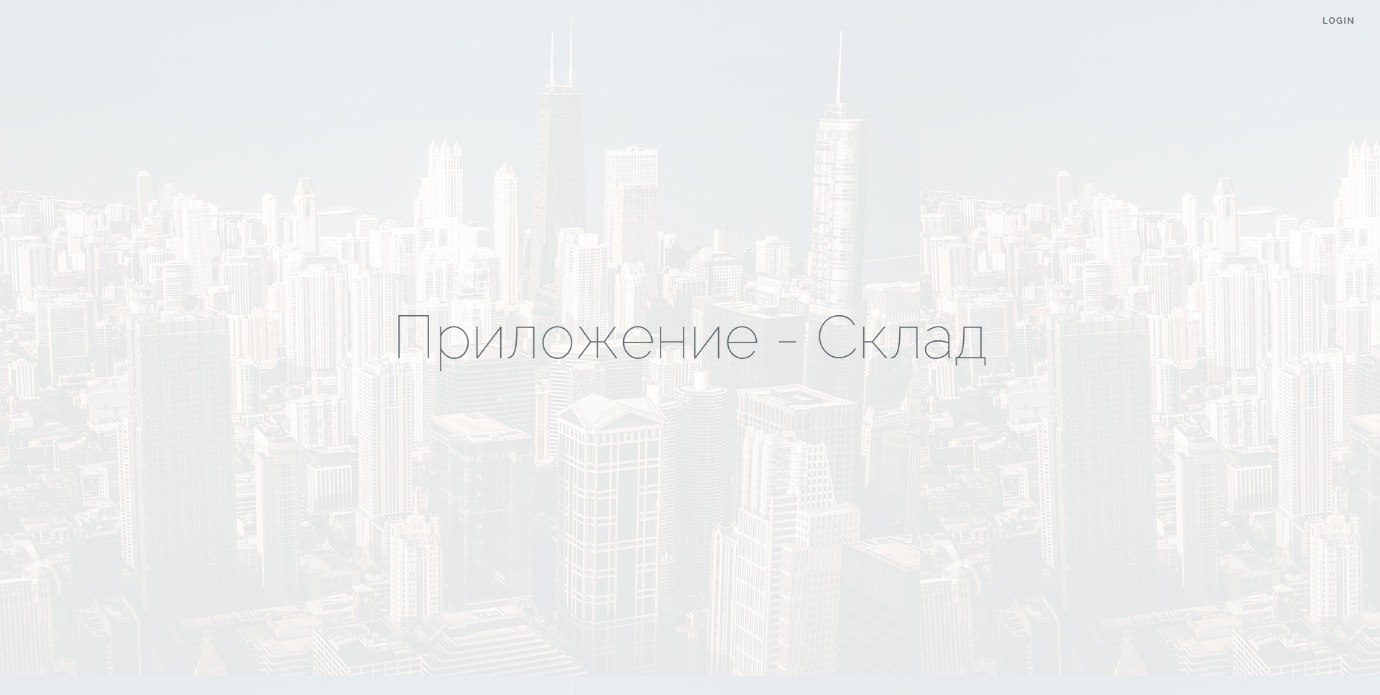


Рисунок 2.2 – Примерный макет страницы «Главная страница сайта для неавторизированного пользователя»

Основные части функционального элемента «Страница поставок деталей», представленного на рисунке 2.3:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Поставки товара»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) выпадающий список для поиска с подсказкой «Накладная»;

6) выпадающий список для поиска с подсказкой «Поставщики»;

7) выпадающий список для поиска с подсказкой «Склад»;

8) кнопка «Добавить» в виде иконки плюса;

9) таблица для отображения деталей со следующими атрибутами: накладная, дата, название товара (товар), код, количество, ед. изм., цена, поставщик, договор, склад.

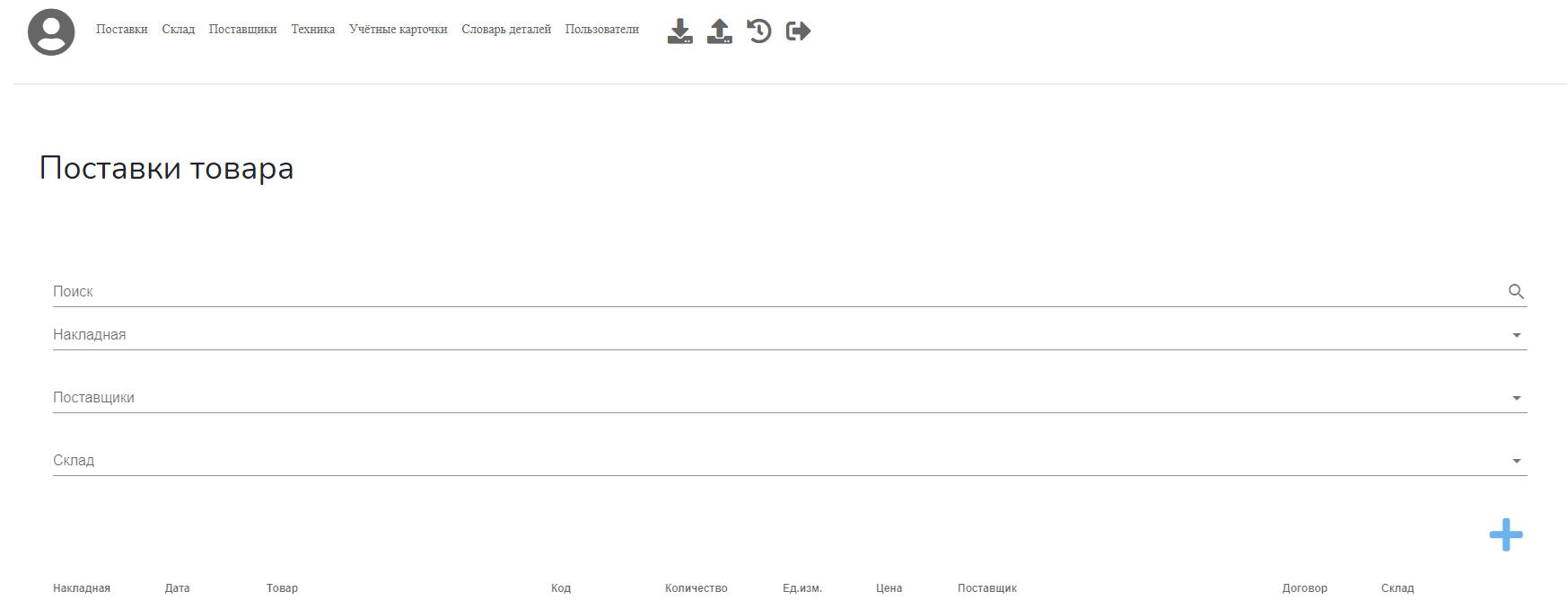


Рисунок 2.3 – Примерный макет страницы «Страница поставок деталей»

Основные части функционального элемента «страница склада», представленного на рисунке 2.4:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Движения товара»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) таблица для отображения перемещений на складе со следующими атрибутами: название з/части, цена за ед., количество, списано.

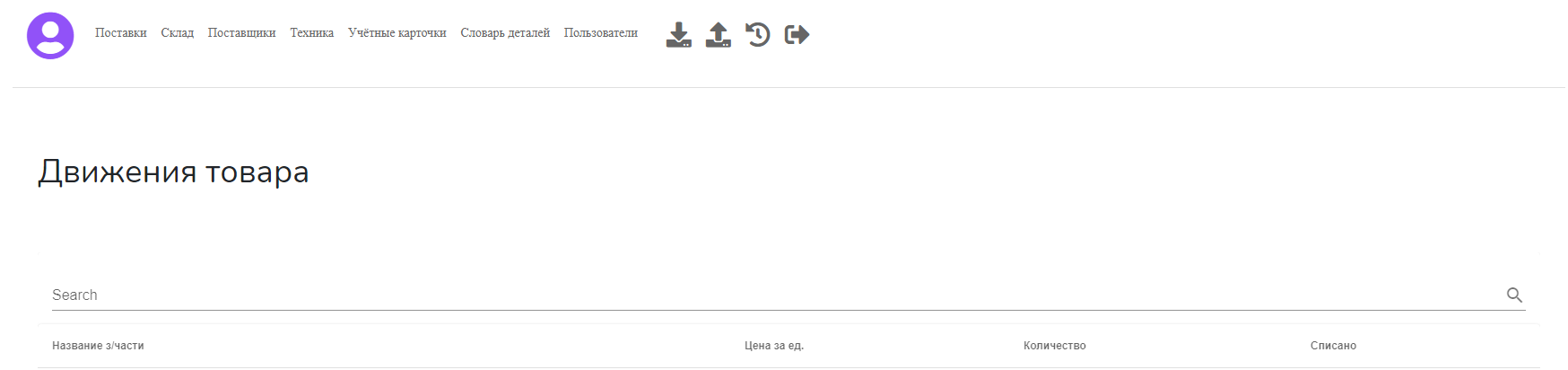


Рисунок 2.4 – Примерный макет страницы «страница склада»

Основные части функционального элемента «страница поставщиков», представленного на рисунке 2.5:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Перечень поставщиков»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) кнопка «Добавить» в виде иконки плюса;

6) таблица для отображения поставщиков со следующими атрибутами: поставщик, телефон, код по ЕГРПОУ, инн, № свид., адрес.

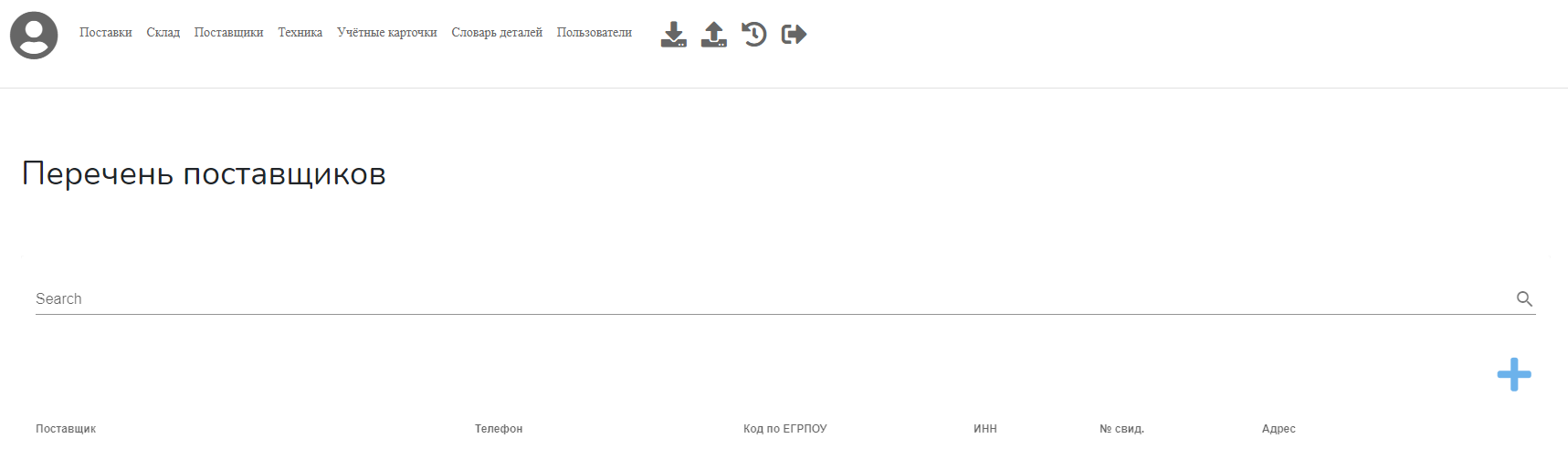


Рисунок 2.5 – Примерный макет «страница поставщиков»

Основные части функционального элемента «страница оборудования», представленного на рисунке 2.6:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Оборудование»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) кнопка «Добавить» в виде иконки плюса;

6) таблица для отображения оборудования со следующими атрибутами: тип оборудования, оборудование, модель, бренд, цвет, производительность cpu, аренда, вес, сертификат, дата выпуска, дата начала эксплуатации, дата последнего ремонта, гос. номер, номер на предприятии, заводской номер.



Рисунок 2.6 – Примерный макет «страница оборудования»

Основные части функционального элемента «страница учётных карточек», представленного на рисунке 2.7:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Учётные карточки деталей»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) выпадающий список для поиска с подсказкой «Оборудование»;

6) выпадающий список для поиска с подсказкой «Деталь»;

7) кнопка «Добавить» в виде иконки плюса;

8) таблица для отображения учётных карточек деталей со следующими атрибутами: наименование оборудования, дата замены, док-т, наименование запчастей, деталей на складе, списано, ед. изм.



Рисунок 2.7 – Примерный макет «страница учётных карточек»

## 

Основные части функционального элемента «страница словаря деталей», представленного на рисунке 2.8:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Справочник по деталям»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) выпадающий список для поиска с подсказкой «Оборудование»;

6) выпадающий список для поиска с подсказкой «Деталь»;

7) кнопка «Добавить» в виде иконки плюса;

8) таблица для отображения справочников по деталям со следующими атрибутами: наименование оборудования, наименование запчастей, количество, ед. изм.

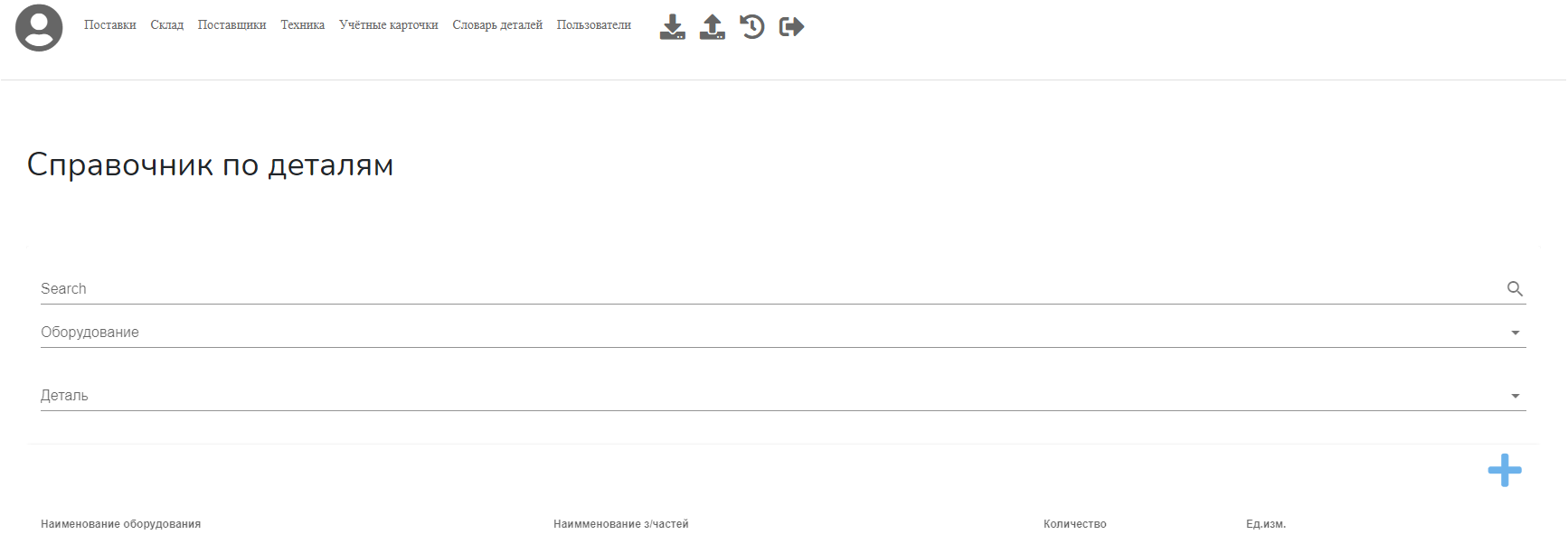


Рисунок 2.8 – Примерный макет «страница справочника по деталям»

Основные части функционального элемента «создание excel», представленного на рисунке 2.9:

1) текстовое поле «Создать excel»;

2) текстовое поле «Тип файла»;   
 3) выпадающий список с типами файла;

4) кнопка «Отмена»;

5) кнопка «Скачать».

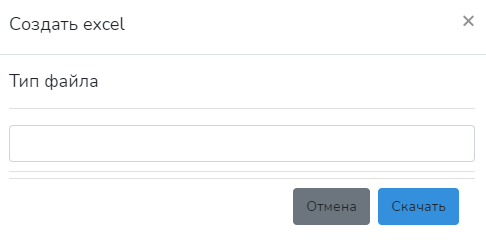


Рисунок 2.9 – Примерный макет «страница создания excel»

Основные части функционального элемента «страница загрузки excel», представленного на рисунке 2.10:

1) текстовая поле «загрузка excel»;

2) кнопка «Выберите файл»;   
 3) текстовое поле с путём к файлу;

4) радиальная кнопка «Накладная»;

5) кнопка «Отмена»;

6) кнопка «Загрузить».

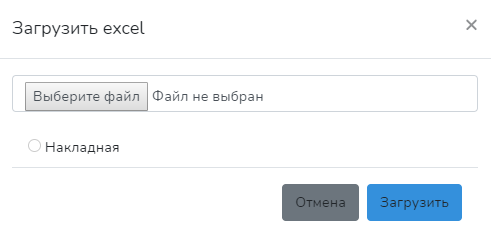


Рисунок 2.10 – Примерный макет «страница лекарства»

Основные части функционального элемента «страница пользователей», представленного на рисунке 2.11:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «Пользователи»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) выпадающий список для поиска с подсказкой «Роль»;

6) выпадающий список для поиска с подсказкой «Компания»;

7) кнопка «Добавить» в виде иконки плюса;

8) таблица для отображения пользователей со следующими атрибутами: №, фамилия, имя, отчество, логин, e-mail, роль, компания.

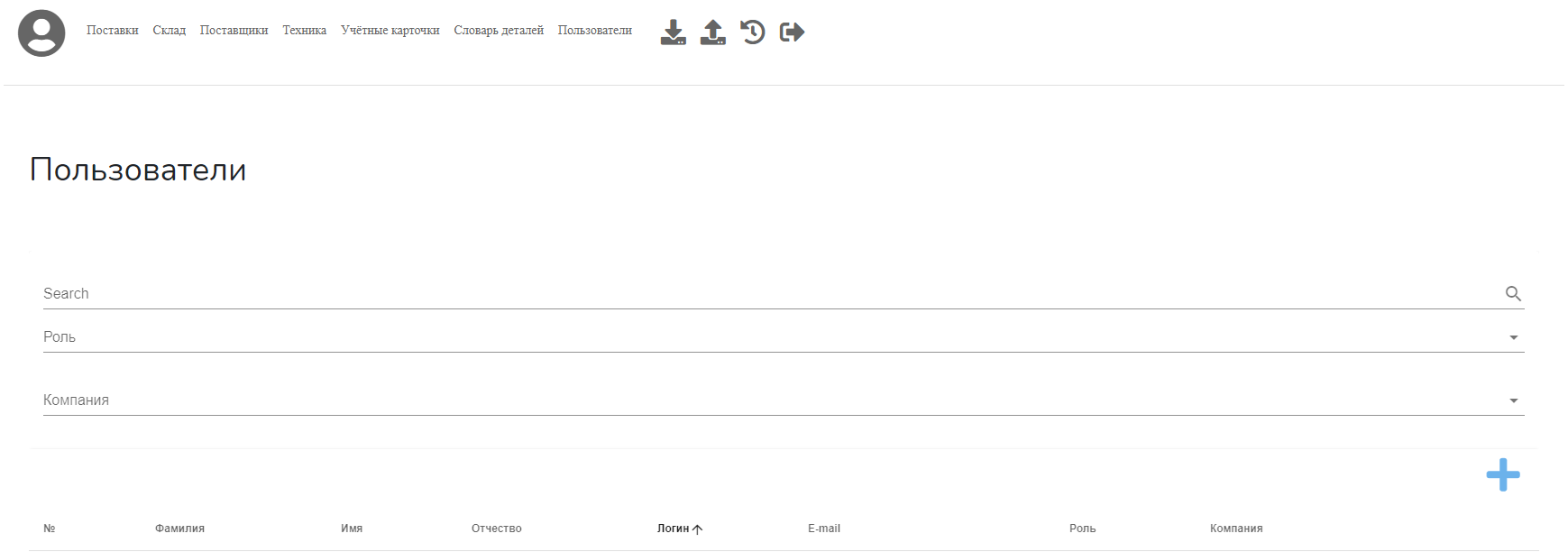


Рисунок 2.11 – Примерный макет «страница пользователей»

Основные части функционального элемента «страница истории изменений», представленного на рисунке 2.12:

1) иконка аккаунта, при наведении на которую высвечивается логин пользователя;

2) меню сайта со следующими элементами: поставки, склад, поставщики, техника (оборудование), учётные карточки, словарь деталей, пользователи, иконка загрузки Excel, иконка выгрузки Excel, иконка перехода к истории использования сайта, иконка выхода;   
 3) текстовая строка «История изменений»;

4) текстовое поле поиска с подсказкой «Поиск» и иконкой лупы;

5) таблица для отображения истории изменений со следующими атрибутами: №, пользователь, дата, изначально, изменено.



Рисунок 2.12 – Примерный макет «страница истории изменений»

## 2.3 Требования к аппаратным, программным и коммуникационным интерфейсам

Для установки и работы приложения необходимо иметь вычислительную систему следующей минимальной аппаратной конфигурации:

1) процессор: 1.3 ГГц;

2) оперативная память: 128 Мб;

3) свободное место на жёстком диске: 150 Мб;

4) видеоадаптер: 64 Мб;

5) разрешение экрана 1280х720 пикселей;

6) наличие основных устройств ввода-вывода (мышь, клавиатура, монитор).

7) выход в интернет;

Необходимое программное обеспечение:

1. операционная система Windows не ниже версии 7;
2. любой браузер.

## 2.4 Требования к пользователям продукта

Пользователями настоящего web-приложения будут являться работники ГОУ ВПО «ДОННУ». В основном те, кто отвечает за поставки оборудования и его эксплуатацию. Так же пользователями настоящего программного обеспечения могут быть те, кто производят обслуживание и ремонт IT-оборудования. Приложением могут использовать в бухгалтерии.

## 2.5 Требования к адаптации на месте

Чтобы начать пользоваться приложением необходимо скачать любой браузер или воспользоваться браузером, встроенным в операционную систему. Открыть его и подключится к серверу с приложением.

## 2.6 Функции продукта

1. Авторизация: если пользователь не внесен в базу данных приложения, он не сможет использовать его;
2. Просмотр списка деталей: для просмотра списка деталей необходимо перейти на страницу «Поставки»;
3. Поиск по ключевым словам: доступен на всех страницах сайта;
4. Поиск по накладным: доступен на странице «Поставки»;
5. Поиск по поставщикам: доступен на странице «Поставки»;
6. Поиск по складам: доступен на странице «Поставки»;
7. Добавление новых деталей: для добавления новых деталей необходимо перейти на страницу «Поставки» и нажать на иконку в виде плюса (только для модераторов, администраторов и работник поставок);
8. Сортировка данных: отсортировать данные можно по любому из атрибутов, доступно три способа сортировки: по возрастанию, по убыванию и без сортировки;
9. Изменение детали: для изменения детали необходимо перейти на страницу «Поставки» и нажать на иконку карандаша рядом с необходимой деталью (только для администраторов и модераторов);
10. Удаление деталей: для удаления детали необходимо перейти на страницу «Поставки» и нажать на иконку корзины рядом с необходимой деталью (только для администраторов);
11. Просмотр движений деталей на складе: для просмотра движения товара необходимо перейти на страницу «Склад»;
12. Просмотр поставщиков: для просмотра поставщиков необходимо перейти на страницу «Поставщики»;
13. Добавление поставщика: для добавления новых поставщиков необходимо перейти на страницу «Поставщики» и нажать на иконку в виде плюса (только для модераторов, администраторов);
14. Изменение поставщика: для изменения поставщика необходимо перейти на страницу «Поставщики» и нажать на иконку карандаша рядом с необходимым поставщиком (только для администраторов и модераторов);
15. Удаление поставщиков: для удаления поставщика необходимо перейти на страницу «Поставщики» и нажать на иконку корзины рядом с необходимым поставщиком (только для администраторов);
16. Просмотр оборудования: для просмотра оборудования необходимо перейти на страницу «Техника»;
17. Добавление оборудования: для добавления нового оборудования необходимо перейти на страницу «Техника» и нажать на иконку в виде плюса (только для модераторов, администраторов);
18. Изменение оборудования: для изменения оборудования необходимо перейти на страницу «Техника» и нажать на иконку карандаша рядом с необходимой техникой (только для администраторов и модераторов);
19. Удаление оборудования: для удаления оборудования необходимо перейти на страницу «Техника» и нажать на иконку корзины рядом с необходимой техникой (только для администраторов);
20. Просмотр деталей списанных на определенное оборудование: для просмотра списка деталей списанных на определенное оборудование, необходимо перейти на страницу «Техника» и нажать на иконку гаечного ключа возле необходимого оборудования;
21. Просмотр учётных карточек: для просмотра учётных карточек необходимо перейти на страницу «Учётные карточки»;
22. Поиск по транспорту: доступен на страницах: «Учётные карточки» и «Словарь деталей»;
23. Поиск по деталям: доступен на страницах: «Учётные карточки» и «Словарь деталей»;
24. Добавление учётных карточек: для добавления новых учётных карточек необходимо перейти на страницу «Учётные карточки» и нажать на иконку в виде плюса (только для модераторов, администраторов);
25. Изменение учётных карточек: для изменения учётных карточек необходимо перейти на страницу «Учётные карточки» и нажать на иконку карандаша рядом с необходимой учётной карточкой (только для администраторов и модераторов);
26. Удаление учётных карточек: для удаления учётной карточки необходимо перейти на страницу «Учётные карточки» и нажать на иконку корзины рядом с необходимой учётной карточкой (только для администраторов);
27. Просмотр словарей деталей: для просмотра словарей деталей необходимо перейти на страницу «Словарь деталей»;
28. Добавление словарей деталей: для добавления новых словарей деталей необходимо перейти на страницу «Словарь деталей» и нажать на иконку в виде плюса (только для модераторов, администраторов);
29. Изменение словарей деталей: для изменения словарей деталей необходимо перейти на страницу «Словарь деталей» и нажать на иконку карандаша рядом с необходимым словарём деталей (только для администраторов и модераторов);
30. Удаление словарей деталей: для удаления словаря деталей необходимо перейти на страницу «Словарь деталей» и нажать на иконку корзины рядом с необходимым словарём деталей (только для администраторов);
31. Просмотр пользователей: для просмотра пользователей необходимо перейти на страницу «Пользователи» (только для администраторов);
32. Поиск по ролям: доступен на странице «Пользователи»;
33. Поиск по компаниям: доступен на странице «Пользователи»;
34. Добавление пользователей: для добавления новых пользователей необходимо перейти на страницу «Пользователи» и нажать на иконку в виде плюса (только для администраторов);
35. Изменение пользователей: для изменения пользователей необходимо перейти на страницу «Пользователи» и нажать на иконку карандаша рядом с необходимым пользователем (только для администраторов);
36. Удаление пользователей: для удаления пользователя необходимо перейти на страницу «Пользователи» и нажать на иконку корзины рядом с необходимым пользователем (только для администраторов);
37. Скачивание накладных: для выгрузки накладной необходимо нажать на иконку «Скачать» в меню сайта и выбрать тип «Накладная»;
38. Скачивание списаний: для выгрузки списаний необходимо нажать на иконку «Скачать» в меню сайта и выбрать тип «Учётная карточка»;
39. Загрузка накладных: для загрузки накладной на сайт необходимо нажать на иконку «Загрузить» в меню сайта и выбрать загружаемый файл в с разрешением .xls;
40. Просмотр истории изменений: для просмотра истории изменений необходимо перейти на страницу «История изменений» с помощью иконки в виде часов в меню сайта (только для администраторов);
41. Выход: для выхода необходимо нажать на иконку выхода в меню сайта.

## 2.7 Ограничения

1. Пользователей может создавать только администратор, как и редактировать их личную информацию;
2. Детали можно добавить только на существующего поставщика;
3. Код детали уникальный и не может повторяться;
4. ИНН поставщика уникальный и не может повторяться;
5. Номер оборудования уникальный и не может повторяться;
6. Нельзя списать на оборудование деталей больше, чем есть на складе в остатке;
7. Нельзя создать двух пользователей с одинаковыми e-mail иди логином;
8. Минимальная длина пароля пользователя – 8 символов;
9. Нельзя загрузить накладную произвольного формата;
10. При создании любой из сущностей все поля являются обязательными для заполнения.

# 3 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ И ПРОГРАММНЫХ СРЕДСТВ

## 3.1 Язык серверных скриптов PHP

PHP - это язык программирования, специально разработанный для написания web-приложений, исполняющихся на web-сервере. PHP достаточно прост для изучения. PHP предоставляет web-разработчикам возможность быстрого создания динамически генерируемых web-страниц. Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP. Этот язык программирования позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем. Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Возможно сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта[4].

PHP не является самым быстрым серверным языком и сильно уступает большинству языков в этом плане, однако его простота в разы ускоряет разработку, что играет большую роль в создании web-приложений. На этом языке написано огромное множество различных framework-ов, одним из которых является Laravel. Для создания настоящего программного обеспечения был выбран именно PHP по той причине, что этот язык используется в Laravel.   
 Однако, PHP не приобрёл свою популярность за счёт написанных для него framework-ов. Это проверенный временем и устоявшийся как основа язык написания серверной логики web-приложений. Если взглянуть на рейтинг языков web-программирования (рисунок 3.1)[5], построенный по различным критериям, однин из которых это анализ кода web-ресурсов, можно убедиться в том, что по данным на конец 2020 года, более 80% сайтов и web-приложений написано на языке программирования PHP.

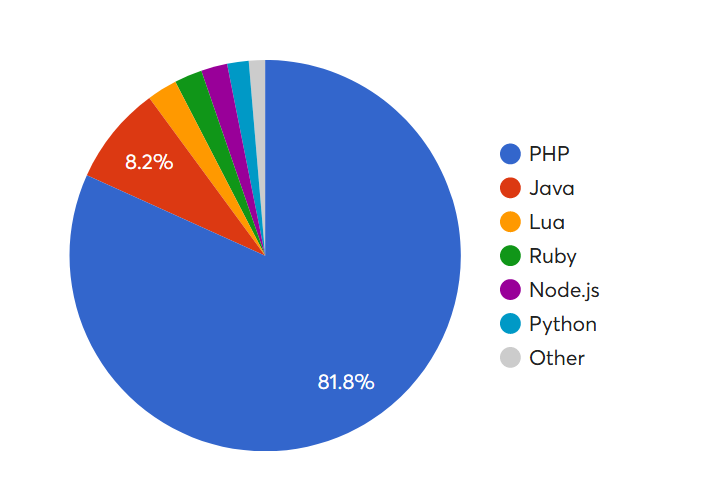


Рисунок 3.1 – Рейтинг языков для web-программирования

## 3.2 Web-framework Laravel

Laravel – это бесплатный PHP фреймворк общего назначения с открытым кодом, который появился на свет сравнительно недавно – в 2011 году, но, благодаря стремительным темпам развития и огромной армии поклонников, сегодня он является одним из самых популярных PHP движков.

Сами создатели Laravel назвали его «framework for artisans», что в переводе означает «фреймворк для ремесленников», намекая на то, что данная платформа даёт разработчикам полную свободу творчества, не создавая перед ними никаких препятствий в процессе разработки[6]. В 2015 и 2016 годах он был признан самым популярным PHP фреймворком. График представлен на рисунке 3.2[6].

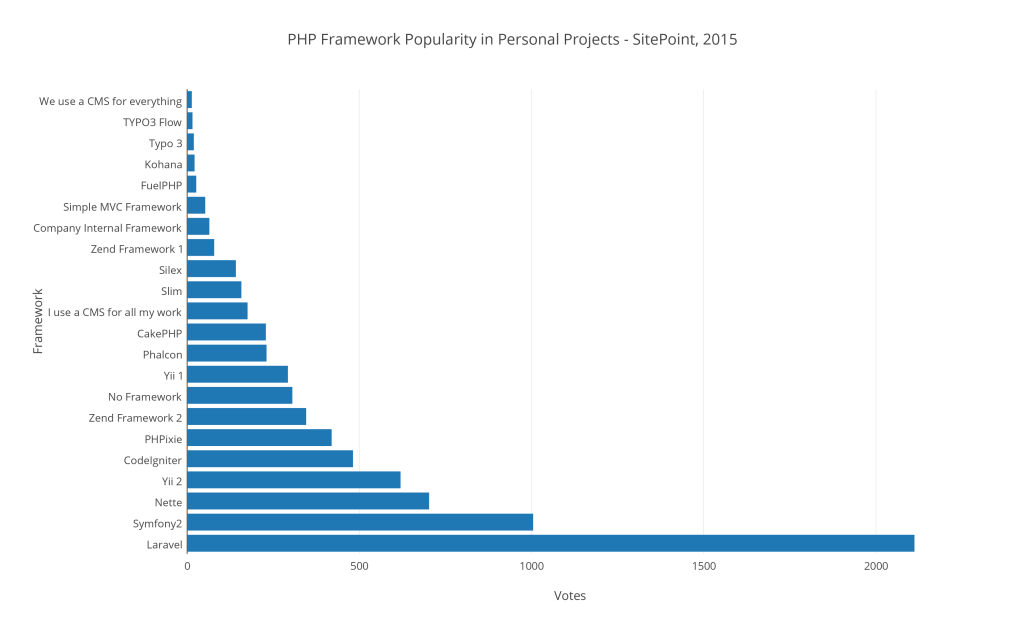


Рисунок 3.2 – Рейтинг популярности php-framework-ов на конец 2015 года

Логика обработки запросов в Laravel построена согласно паттерну MVC (Model-View-Controller). Это основной паттерн в разработке современных web-приложений. Принцип работы в таком случае заключается в следующем: все запросы поступающие на сервер обрабатываются контроллером, который в свою очередь, с помощью ORM модели обращается к базе данных. После получения данных, контроллер возвращает их в удобно виде на View пользователю. Наглядно, с работой паттерна MVC в Laravel можно ознакомиться на рисунке 3.3.

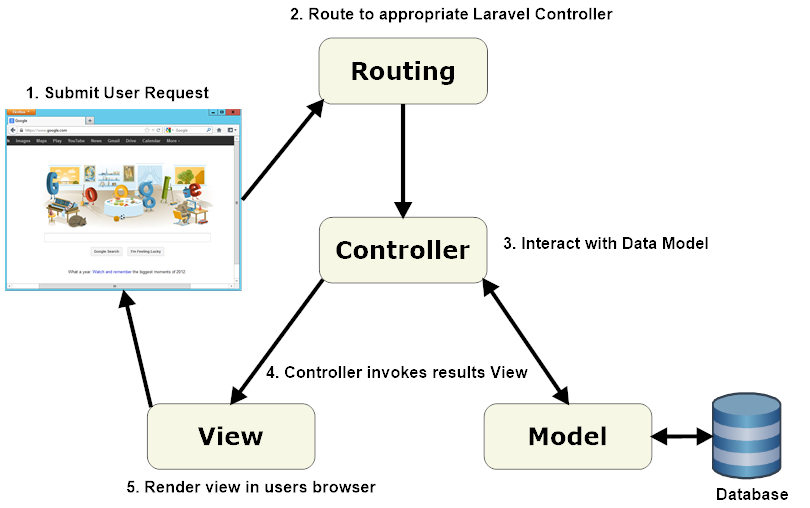


Рисунок 3.3 – Принцип работы паттерна MVC в Laravel

Laravel значительно ускоряет и упрощает разработку web-приложения на PHP. Если ознакомится с «быстрым стартом» указанным на официальной странице разработчика, можно понять, что сайт с базой данных и готовой авторизацией можно создать, используя всего несколько команд. Laravel – настоящий прорыв в web-разработке, предоставляющий гибкие возможности создания web-ресурсов.

## 3.3 Редактор исходного кода Visual Studio Code

[Visual Studio Code —](https://code.visualstudio.com/Docs) это облегченный, но мощный редактор исходных кодов. [Visual Studio Code](https://code.visualstudio.com/Docs) доступен для Windows, Linux и macOS. Он включает встроенную поддержку JavaScript, TypeScript и Node.js, поэтому это отличное средство для веб-разработчиков[6].

Главное преимущество этого редактора является в количестве доступных плагинов. Это и проверка синтаксиса различных языков, и редактор элементов сайта, и даже отладчик для Microsoft Edge. Для программирования на Laravel обычно выбирают из нескольких редакторов исходного кода, это PHP Storm и Visual Studio Code.   
 В то время как PHP Storm является мощной средой разработки, Visual Studio Code предоставляет возможность гибкой настройки установленных плагинов и не нагружает систему. Это лёгкий, продуманный и быстрый инструмент для web-разработки.

Ко всему прочему Visual Studio Code обладает встроенными элементами для работы с системой контроля версий Git и предоставляет пользователю огромное множество визуальных возможностей для понимания, в какие файлы были внесены изменения, какие из файлов уже закоммичены, а какие ждут своей очереди.

Благодаря своей лаконичной и простой для освоения структуре Visual Studio Code занимает лидирующее место среди редакторов кода в web-сегементе. Пример редактора кода с открытым Laravel проектом показан на рисунке 3.4.

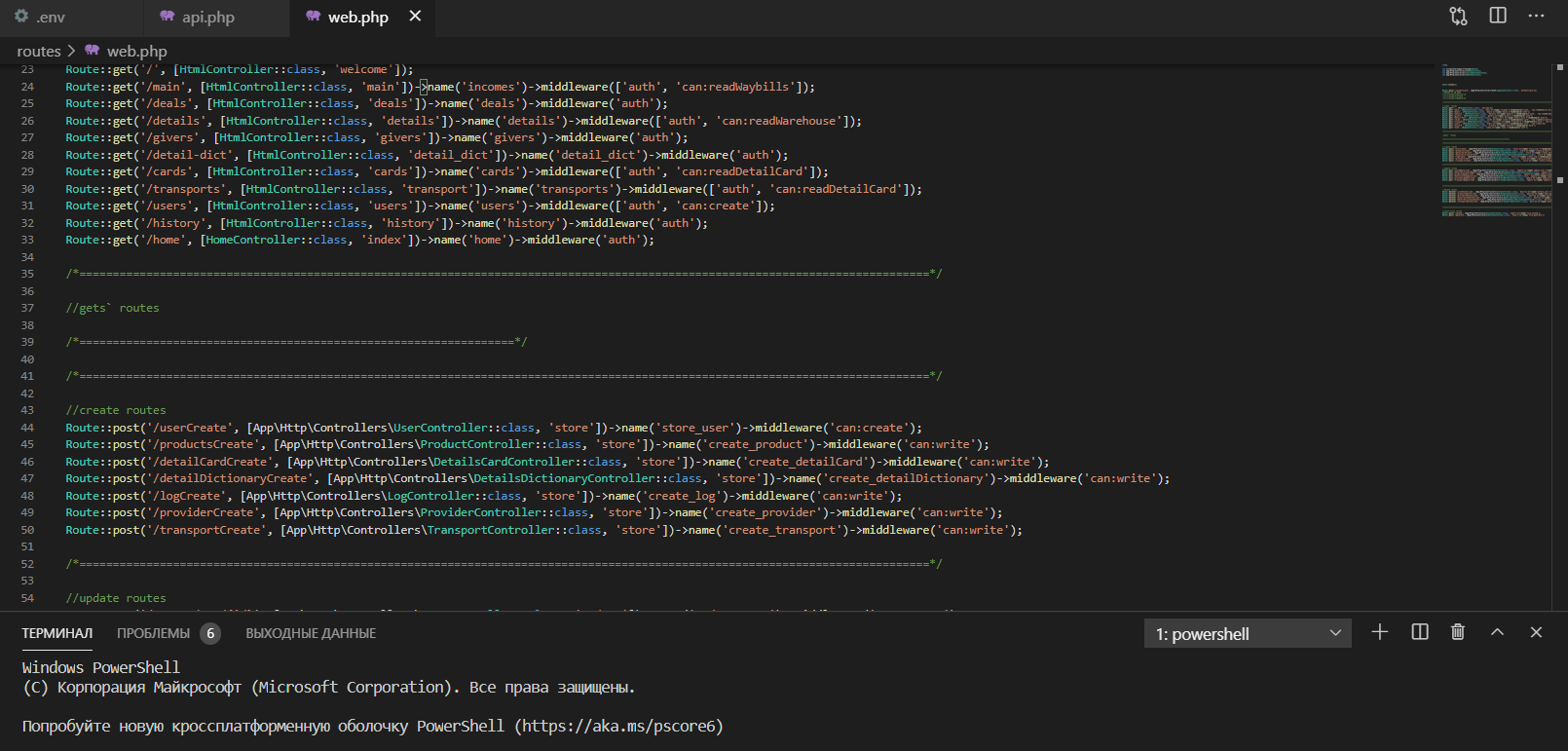


Рисунок 3.4 – окно редактора кода в Visual Studio Code

## 3.4 Front-end framework Vue.js

Vue - это web-фреймворк для разработки пользовательских интерфейсов на языке программирования JavaScript. Vue создан для постепенного внедрения в существующее приложение. Он решает различные задачи уровня представления (view), упрощает работу с другими библиотеками и позволяет создавать сложные одностраничные приложения (SPA, Single-Page Applications)[7].

Проще говоря, Vue.js инструмент web-разработки, который предоставляет возможности для создания визуального интерфейса и логики клиента даже если программист знаком с JavaScript на базовом уровне. Framework полностью охватывает всю клиентскую логику, включая все скрипты на JavaScript и разметку на HTML и CSS.   
 Наравне с Laravel, который является лидирующим back-end framework-ом, Vue.js – занимает лидирующую позицию в front-end составляющей. Вся разработка на Vue состоит в использовании определенных концепций framework-а, а именно: конструкторы, компоненты, директивы и переходы.

Современный Vue-js – очень быстрый framework, что можно увидеть на диаграмме сравнения скорости современных front-end framework-ов (рисунок 3.5)[8].

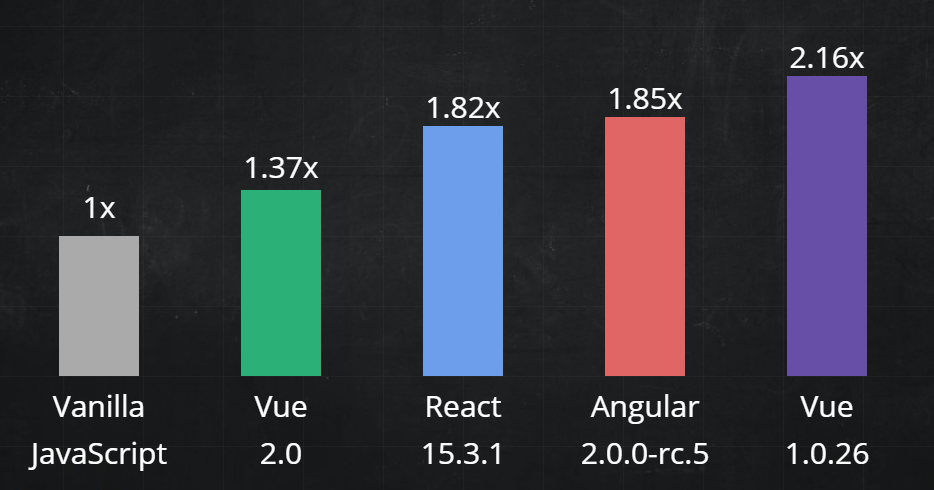


Рисунок 3.5 – сравнение скорости front-end framework-ов

Для разработки настоящего web-приложения Vue.js был выбран из-за скорости обработки и простоты освоения. На современном рынке сложно найти настолько мощный и настолько простой в освоении front-end framework.

## 3.5 Сервера баз данных MySQL

MySql - система управления реляционными базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. СУРБД — это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели. Разработан MySQL был в 1994 году Шведской компанией MySQL AB. Тогда американская технологическая компания Sun Microsystems полностью приобрела право собственности, купив MySQL AB в 2008 году. Американский технологический гигант Oracle в 2010 году приобрёл Sun Microsystems, а с тех пор MySQL практически принадлежала Oracle [9].   
 Почему был выбран именно сервер баз данных MySQL? Какие у него преимущества?

1. Гибкость и простота использования: свободное изменение исходного кода под свои предпочтения и отсутствие ценника на СУБД MySQL.
2. Высокая производительность: MySQL позволяет очень гибко и тщательно настроить производительность сервера баз данных, позволяя достичь заметного ускорения работы по сравнению с другими СУБД.
3. MySQL, как промышленный стандарт: Эта СУБД настолько популярна, что является неотъемлемой частью почти любой разработки. Без знаний языка программирования SQL не возьмут ни на одну работу программистом.

Безопасность: сервера этой базы данных очень хорошо защищены. Ко всему прочему, СУБД даёт возможность настройки этой защиты, увеличивая сохранность данных.

# 4 РАЗРАБОТКА БАЗЫ ДАННЫХ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЁТА IT-ОБОРУДОВАНИЯ ГОУ ВПО «ДОННУ»

## 4.1 Выбор и обоснование СУБД

# СУБД – приложения, либо библиотеки управляющие базами данных. Именно системы управления базами данных покрывают всю работу с базами, их созданием, заполнением, удалением.

В настоящей дипломной работе применяется РСУБД - это реляционная система управления базами данных. То есть такая система, которая организовывает всю работу с таблицами и их кортежами, атрибутами в рассматриваемой базе данных.   
 Рассмотрим варианты существующих РСУБД.   
 Одной из самых популярных на сегодняшний день РСУБД является MySQL. Одно из главных её преимуществ – она бесплатная и позволяет работать с базами данных, как в консоли, так и с помощью специального приложения с интуитивно понятным интерфейсом. Именно эта СУБД, по сравнению со всеми остальными является одной из самых мощных, быстродействующих и безопасных. Основной причиной для этого, является гибкость её работы, которая достигается за счёт возможности контролировать практически все существующие процессы в базе данных. Однако она обладает некоторыми ограничениями, которые позволяют обойти другие СУБД. Ко всему прочему, в случае переноса базы данных на другую СУБД из этой, могут возникунть проблемы совместимости.   
 Ещё одной из самых современных РСУБД является PostgreSQL. Эта РСУБД является самой гибкой и продвинутой на фоне остальных. Также обеспечивает полную совместимость базы данных при переходе не другую РСУБД. Однако, за счёт того что эта РСУБД обладает колоссальным функционалом, она уступает в скорости и производительности другим СУБД. А ещё, из-за обилия этого функционала является сложной для освоения.  
 Рассмотрим ещё одну РСУБД. SQLite – очень мощная встраиваемая система управления базами данных, которая лучше всего подходит для быстрой разработки базы данных и последующего её переноса на другие платформы. Однако, за счёт того что она не обладает внушительным функционалом, она не позволяет гибко настраивать базу данных. То есть у пользователя не получиться сделать свою базу данных более производительной и быстрой.   
 Делая вывод на основании информации предоставленной выше, принято решение, использовать в настоящей дипломной работе РСУБД MySQL, так как эта СУБД не является слишком трудной в освоении, но за счёт предоставляемой гибкости, позволяет эффективно работать с реляционными базами данных.

* 1. **Концептуальное и логическое проектирование БД**

Концептуальная модель данных – это модель данных, которая предоставляет информацию о том, какие отношения и их атрибуты должны существовать в проектируемой базе данных, без какой либо ссылки на логику взаимодействия между ними.   
 Рассмотрим сущности, которые будут использоваться в проектируемой базе данных (рис. 4.1):   
 1. Карточка детали;   
 2. Словарь детали;   
 3. Запись лога;   
 4. Деталь;   
 5. Поставщик;   
 6. Роль;   
 7. Пользователь;   
 8. Тип оборудования;   
 9. Оборудование.

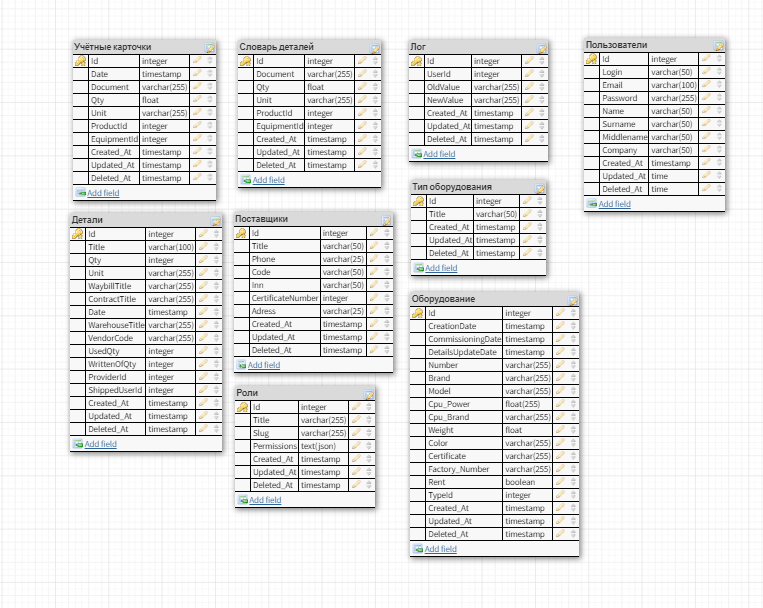


Рисунок 4.1 – Концептуальная модель данных

Логическая модель данных – это такое представление данных, в котором отображены отношения, их атрибуты, а так же указаны связи между отношениями без ссылки на то, как это достигается (рис 4.2).

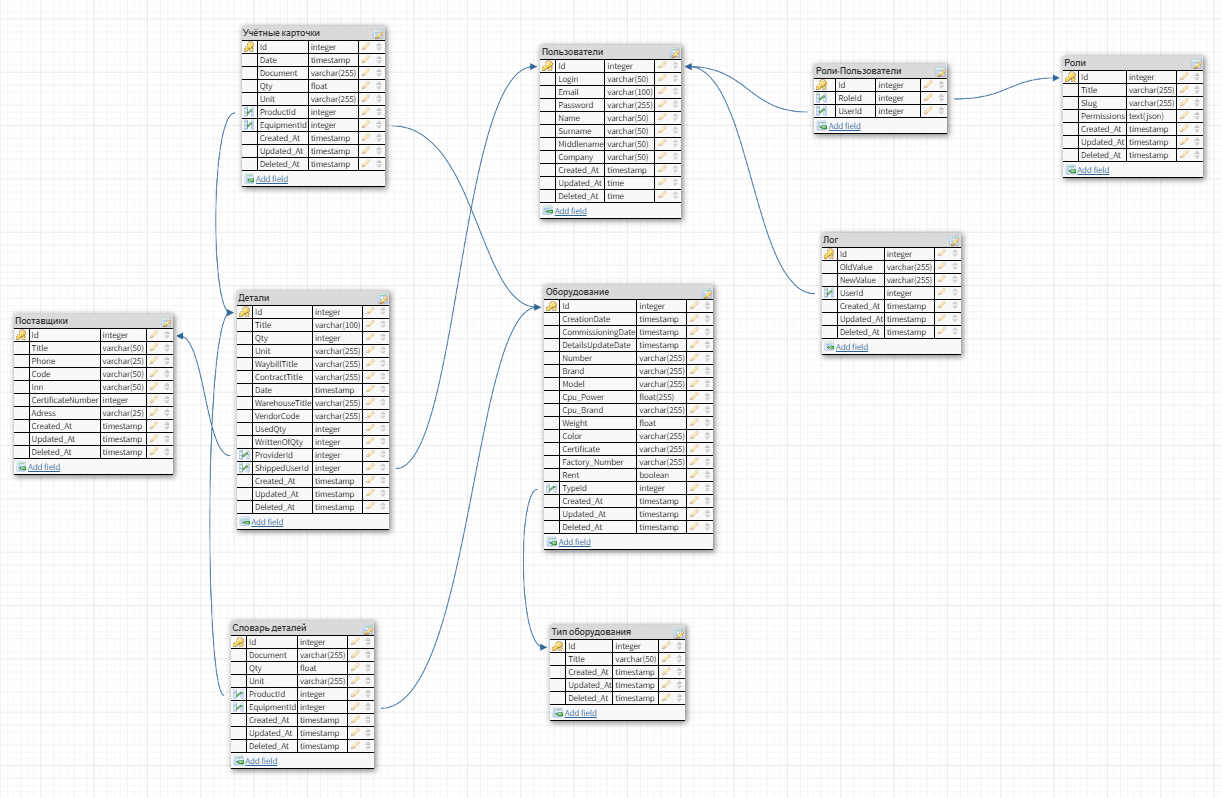


Рисунок 4.2 – Логическая модель данных

В базе данных есть три основных отношения: «Пользователи», «Оборудование» и «Детали». Отношение «Пользователи» описывает информацию связанную с пользователями, отношение «Оборудование» описывает технику на складе, отношение «Детали» описывает все детали на складе. Отношение «Учётные карточки» связано с отношениями «Оборудование» и «Детали» связями один-ко-многим. Это отношение показывает, какие детали были списаны на то или иное оборудование. Отношение «Словарь деталей» имеет аналогичные учётным карточкам связи, но показывает, какие детали в какой технике применяются. Отношение «Поставщики» связано с отношением «Детали» связью один-ко-многим и содержит информацию о поставщиках тех или иных деталей. «Пользователи» связаны с отношением «Детали» связью один-ко-многим. У «Оборудования» есть связь один-ко-многим с отношением «Типы оборудования». У отношения «Лог» есть связь многие-ко-многим с отношением «Пользователи». Это отношение описывает действия производимые пользователями на сайте. Наконец отношение «Роли» имеет связь многие-ко-многим с отношением «Пользователи» через таблицу «Роли-Пользователи».

**4.3 Физическое проектирование БД**

Последним этапом проектирования базы данных, является создание её физической модели – то есть такой модели, в которой подробно описаны все отношения, их атрибуты, а так же идёт подробное описание того, как это достигается. На рисунке (рис. 4.3) показана физическая модель данных, базы данных автоматической системы учёта IT-оборудования ГОУ ВПО «ДОННУ».

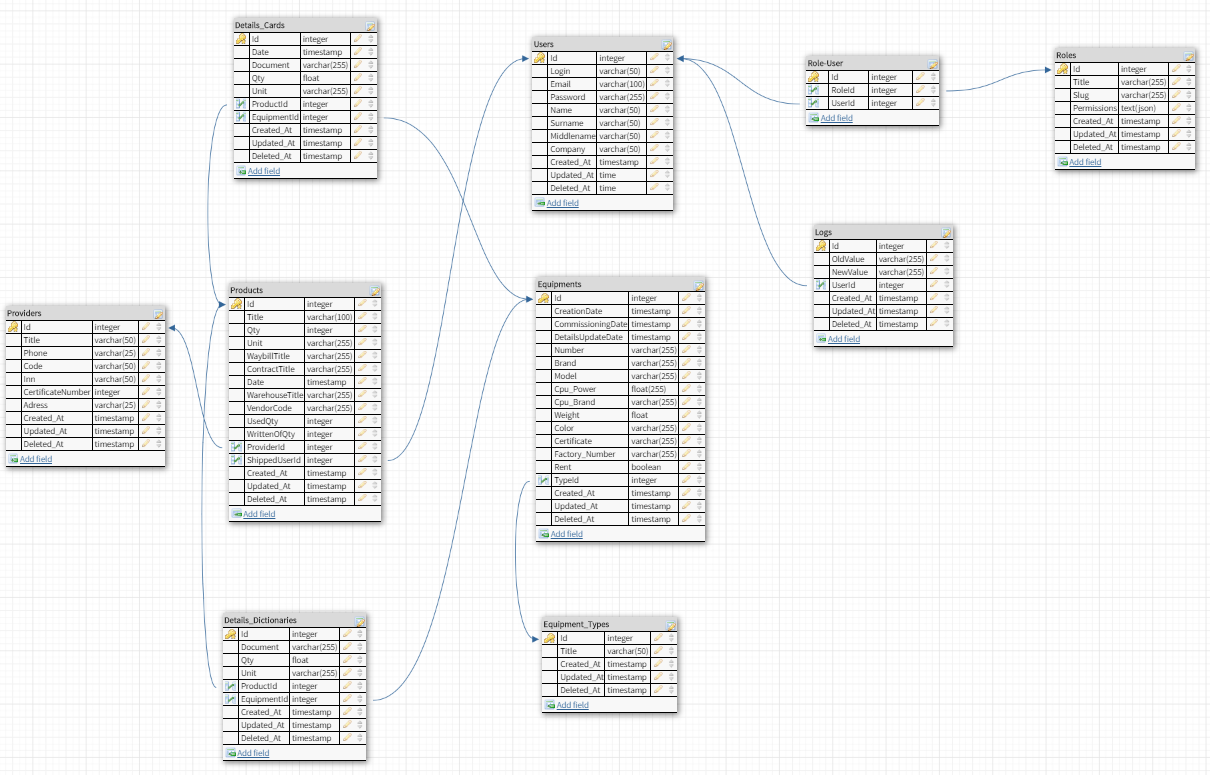


Рисунок 4.3 – Физическая модель данных

В физическом проектировании базы данных, особое внимание уделяется глоссарию – некому словарю с пояснениями всех атрибутов отношений в базе данных, для удобства работы с ними. Данный глоссарий предоставлен по отдельности для каждой таблицы, с описанием того, какую информацию содержит отношение.

Отношение «Details\_Cards» - содержит информацию о списаниях деталей на определенное оборудование:   
 - Id (auto increment primary key) – уникальный идентификатор учётной карточки в отношении;   
 - Date (timestamp) – дата проведения списания;   
 - Document (varchar(255)) – наименование документа подтверждающего списание;  
 - Qty (double) – количество списанных деталей;

- Unit (varchar(255)) – единица измерения для количества списанных деталей;   
 - ProdcutId – id списанной детали (foreign key (ProdcutId) references Products (Id));   
 - EquipmentId - id оборудования, для которого производится списание (foreign key (EquipmentId) references Equipments (Id));   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Details\_Dictionaries» - содержит информацию о словарях деталей. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор словаря деталей в отношении;   
 - Qty (double) – количество деталей;

- Unit (varchar(255)) – единица измерения для количества деталей;   
 - ProdcutId – id детали применяемой в оборудовании (foreign key (ProdcutId) references Products (Id));   
 - EquipmentId - id оборудования, для которого создан словарь (foreign key (EquipmentId) references Equipments (Id));   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Logs» - содержит информацию о болезнях. Его атрибуты:  
 - Id (auto increment primary key) – уникальный идентификатор записи лога в отношении;   
 - OldValue (varchar(255)) – старое значение, которое было изменено;

- NewValue (varchar(255)) – новое значение на которое поменялось старое;   
 - UserId – id пользователя, который совершал действие (foreign key (UserId) references Users (Id));   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Prodcuts» - содержит информацию о лекарствах. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор детали в отношении;   
 - Title (varchar(100)) – название детали;   
 - Price (decimal) – цена за одну единицу детали;   
 - Qty (integer) – количество деталей на складе;   
 - Unit (varchar(255)) – единица измерения количества деталей;   
 - WaybillTitle (varchar(255)) – название накладной, по которой детали поступил на склад;   
 - ContractTilte (varchar(255)) – название договора с поставщиком;   
 - Date (timestamp) – дата поступления деталей на склад;   
 - WarehouseTitle (varchar(255)) – название склада на который поступили детали;   
 - VendorCode (varchar(255)) – код продавца деталей;   
 - UsedQty (integer) – количество использованных деталей;   
 - WrittenOffQty (integer) – количество списанных деталей;   
 - ProviderId (integer) – id поставщика деталей (foreign key (ProviderId) references Providers (Id));   
 - ShippedUserId (integer) – id пользователя, который вносил детали в систему (foreign key (ShippedUserId) references Users (Id));   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Providers» - содержит информацию о поставщиках. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор поставщика в отношении;   
 - Title (varchar(50)) – название поставщика;   
 - Phone (varchar(25)) – телефон поставщика;   
 - Code (varchar(50)) – уникальный код поставщика;   
 - Inn (varchar(50)) – уникальный inn поставщика;   
 - CertificateNumber (integer) – номер сертификата;   
 - Adress (varchar(255)) – адрес поставщика;   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Role-Users» - промежуточное отношение для реализации связи многие-ко-многим между отношениями «Users» и «Roles». Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор связи в отношении;   
 - User\_Id (integer) – id пользователя (foreign key (User\_Id) references Users (Id));   
 - Role\_Id (integer) – id роли (foreign key (Role\_Id) references Roles (Id)).

Отношение «Roles» - содержит информацию о ролях пользователей. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор роли в отношении;   
 - Title (varchar(255)) – наименование роли;   
 - Slug (integer) – краткое название роли;   
 - Permissions (json) – разрешения роли;   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Equipment\_Types» - содержит информацию о типах оборудования. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор типа оборудования в отношении;   
 - Title (varchar(100)) – наименование типа оборудования;   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Equipments» - содержит информацию об оборудовании. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор оборудования в отношении;   
 - Title (varchar(100)) – наименование оборудования;   
 - CreationDate (timestamp) – дата выпуска оборудования;   
 - CommissioningDate (timestamp) – дата введения в эксплуатацию;   
 - DetailsUpdateDate (timestamp) – дата последнего ремонта оборудования;  
 - Number (varchar(255)) – номер оборудования;   
 - Brand (varchar(255)) – бренд оборудования;   
 - Model (varchar(255)) – модель оборудования;   
 - Cpu\_Brand (varchar(255)) – бренд центрального процессора оборудования;  
 - Cpu\_Power (double) – мощность центрального процессора оборудования (ГГц);   
 - Weight (double) – вес оборудования;   
 - Color (varchar(255)) – цвет оборудования;   
 - Certificate (varchar(255)) – сертификат на оборудование;   
 - Factory\_Number (varchar(255)) – номер завода на котором было произведено оборудование;   
 - Rent (boolean) – взято ли оборудование в аренду;   
 - TypeId (integer) - тип оборудования (foreign key (TypeId) references Equipment\_Types (Id));   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

Отношение «Users» - содержит информацию о пользователях. Его атрибуты:   
 - Id (auto increment primary key) – уникальный идентификатор пользователя в отношении;   
 - Login (varchar(50)) – уникальный логин пользователя, используется для авторизации;   
 - Email (varchar(100)) – уникальный адрес электронной почты пользователя;   
 - Password (varchar(255)) – пароль пользователя, используется при авторизации;   
 - Name (varchar(50)) – имя пользователя;   
 - Surname (varchar(50))– фамилия пользователя;   
 - Middlename (varchar(50)) – отчество пользователя;   
 - Company (varchar(50)) – компания пользователя, используется для отображения добавленных деталей;   
 - Created\_At (timestamp) – дата создания записи в отношении;   
 - Updated\_At (timestamp) – дата изменения записи в отношении;   
 - Deleted\_At (timestamp) – дата удаления записи.

1. <http://usu.kz/wiki/avtomatizirovannaya_sistema_ucheta.php>
2. <https://www.krug2000.ru/decisions/solutions_comacc/ackyt.html>
3. <https://abmcloud.com/abm-soft/wms/>
4. <http://www.php.su/php/?php>
5. <https://www.wappalyzer.com/technologies/programming-languages/>
6. <https://docs.microsoft.com/ru-ru/microsoft-edge/visual-studio-code/>
7. <https://unetway.com/tutorial/cto-takoe-vuejs>
8. <https://habr.com/ru/post/329452/>
9. Что такое MySQL: Объяснение MySQL Для Начинающих // Руководства Hostinger URL: <https://www.hostinger.com.ua/rukovodstva/shto-takoje-mysql/> (дата обращения: 27.04.2020).