

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

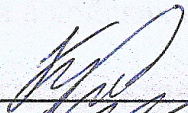
Специальность: 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»
Специализация: «Эксплуатация вычислительных машин, комплексов, систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

**КЛИЕНТСКАЯ ЧАСТЬ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
УЧЕТА ТЕХНИКИ СВЯЗИ И ФОРМИРОВАНИЯ ОТЧЕТНЫХ
ДОКУМЕНТОВ В ВОЙСКАХ СВЯЗИ ВОЕННОГО ОКРУГА**

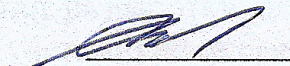
Студент

 14.06.22 Кривцов Михаил Андреевич
(подпись, дата) (ФИО полностью)

Руководитель

 14.06.22 Митрохин М.А.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

 14.06.22 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Рецензент

(должность, место работы)  16.06.22 Афонин А.Ю.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 02.06.2022 г. № 13)

Заведующий кафедрой

 М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от

17 июня 2022 № 5)

Секретарь ГЭК

 Е.И. Гурин
(подпись)

Пенза 2022

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

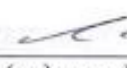
«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«26» апреля 2022 г.

**ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ**

1. Студент Кривцов Михаил Андреевич гр. 17BO2 факультета вычислительной техники.
2. Тема работы «Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа»
Тема утверждена приказом ПГУ №0357/о от "26" апреля 2022 г.
3. Руководитель работы д.т.н., заведующий кафедрой ВТ ПГУ Митрохин М.А.
4. Задание на работу (назначение разработки, исходные данные и т.п.)
Автоматизированная система учета техники связи и формирования отчетных документов в войсках связи военного округа предназначена для упрощения и ускорения взаимодействия между ответственными лицами по вопросам учета техники связи, а также формирования необходимых документов по их обслуживанию и ремонту. Система должна иметь клиент-серверную архитектуру. Клиентская часть автоматизированной системы отвечает за пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.
5. Перечень подлежащих разработке вопросов
Требуется произвести анализ предметной области, сформировать техническое задание, выбрать инструменты для разработки клиентской части автоматизированной системы, выполнить проектирование системы, разработать клиентскую часть системы. Клиентская часть системы должна выполнять функции: отображать всю доступную информацию; делать запросы на добавление, изменение и удаление информации на сервер; обеспечивать обмен сообщениями между пользователями; авторизацию в системе; настраивать параметры для ввода данных в систему. Также выполнить задание по БЖД и осуществить экономическое обоснование разработки системы.
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	8%	26.04.2022 – 29.04.2022	
2.	Составление технического задания	6%	30.04.2022 – 02.05.2022	
3.	Выбор инструментов разработки	8%	03.05.2022 – 06.05.2022	
4.	Проектирование системы	12%	07.05.2022 – 12.05.2022	
5.	Разработка клиентской части системы	40%	13.05.2022 – 01.06.2022	
6.	Выполнение экономического обоснования	10%	02.06.2022 – 06.06.2022	
7.	Выполнение задания по БЖД	8%	07.06.2022 – 10.06.2022	
8.	Оформление пояснительной записки	8%	11.06.2022 – 14.06.2022	

Дата выдачи задания "26" апреля 2022 г.

Руководитель работы  26.04.2022
(подпись, дата)

Задание к исполнению принял студент  26.04.2022
(подпись, дата)

Работу к защите допустить
Декан факультета  14.06.2022
(подпись, дата)

Реферат

Пояснительная записка содержит 148 листов, 40 рисунков, 16 таблиц, 7 источников, 4 приложения.

КЛИЕНТ, ИНТЕРФЕЙС, БРАУЗЕР, АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА ТЕХНИКИ СВЯЗИ И ФОРМИРОВАНИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ В ВОЙСКАХ СВЯЗИ ВОЕННОГО ОКРУГА.

Цель дипломного проектирования - разработка клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа.

Разработанная клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа отвечает за пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

Графическая часть дипломного проекта содержит 5 *UML*-диаграмм формата A1.

Разработанная клиентская часть автоматизированной системы была реализована на языках: *JavaScript*, *HTML*, *CSS*, а также с использованием библиотеки *React.js*.

Для проектирования системы было использовано *CASE*-средство *StarUML*.

					ПГУ 1.090501.07.001 ПЗ				
Изм.	Лист	№ докум.	№	Подпись	Дата				
Разраб.		Крыжов М.А.				Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа Пояснительная записка	Лит.	Лист	Листов
Провер.		Митрохин М.А.						2	148
Н. Контр.		Бычков А.С.					ФВТ, каф. ВТ гр. 17В02		
Утверд.		Митрохин М.А.							

Содержание

Перечень сокращений и обозначений	6
Введение	7
1 Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа ...	9
1.1 Анализ предметной области	9
1.2 Техническое задание	14
1.2.1 Основание для разработки	14
1.2.2 Назначение разработки	14
1.2.3 Требования к программе	14
1.2.3.1 Требования к функциональным характеристикам	14
1.2.3.2 Требования к надежности	15
1.2.3.3 Требования к составу и параметрам технических средств	15
1.2.3.4 Требования к информационной и программной совместимости	16
1.2.4 Требования к программной документации	16
1.2.5 Стадии и этапы разработки	17
1.2.6 Порядок контроля и приемки	17
1.3 Анализ целесообразности использования заданных средств разработки	17
1.4 Проектирование автоматизированной системы	24
1.4.1 UML - моделирование автоматизированной системы	24
1.4.1.1 Диаграмма вариантов использования	25
1.4.1.2 Диаграмма деятельности	27
1.4.1.3 Диаграмма последовательности	34
1.4.1.4 Диаграмма классов	35
1.4.1.5 Диаграмма развертывания	36
1.4.2 Общая архитектура автоматизированной системы	37
1.4.3 Проектирование клиентской части	38

1.5	Описание клиентской части	43
1.5.1	Общие сведения	43
1.5.2	Функциональное назначение	43
1.5.3	Описание логической структуры	43
1.5.4	Используемые технические средства	58
1.6	Программа и методика испытаний	58
1.6.1	Объект испытания	58
1.6.2	Цель испытаний	58
1.6.3	Требования к клиентской части системы	59
1.6.4	Требования к программной документации	59
1.6.5	Средства и порядок испытаний	59
1.6.6	Методы испытания	61
1.7	Описание применения	68
1.7.1	Руководство пользователя	68
1.7.2	Руководство администратора	70
2	Технико-экономическое обоснование разработки программы	73
2.1	Организация производства при разработке клиентской части АС	73
2.2	Календарное планирование разработки программного обеспечения	76
2.3	Расчет себестоимости программного обеспечения	79
2.4	Расчет экономической эффективности	83
3	Безопасность жизнедеятельности	85
3.1	Введение.	85
3.2	Требования к параметрам воздушной среды	86
3.3	Требования к освещению рабочих мест	86
3.4	Требования к уровням шума и вибрации на рабочих местах	88
3.5	Расчет освещенности рабочего места	89
	Заключение	93
	Список использованных источников	94
	Приложение А - Результаты выполнения тестовых примеров	95

Приложение Б - Листинг клиентской части	108
Приложение В - <i>UML</i> - диаграммы	134
Приложение Г - Презентация	140

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

Перечень обозначений и сокращений

АИС - автоматизированная информационная система

ВВТ - вооружение и военная техника

МСВС - мобильная система Вооруженных Сил

Фронтенд - видимая пользователям часть сайта или приложения

Фреймворк - программная платформа, определяющая структуру программной системы

Обфускация - процесс изменения кода программы, в результате которого он приобретает вид, трудный для понимания

DOM - объектная модель документа

ASCII - название таблицы, в которой некоторым распространённым печатным и непечатным символам сопоставлены числовые коды

MVVM - шаблон проектирования архитектуры приложения

CLI - интерфейс командной строки,

БД - база данных

СУБД - система управления базами данных

КТС - комплекс технических средств

Введение

Автоматизированная система - это совокупность программных и аппаратных средств, которые используются для регулярной обработки информации и решения поставленных задач.

Автоматизированная система всегда функционирует в окружении внешней среды, которая является источником как входной, так и выходной информации. К основным этапам функционирования АИС можно отнести:

- сбор, регистрация и первичная обработка;
- передача по каналу связи от источника к компьютеру;
- перенос на машинные носители;
- создание и поддержание информационных фондов;
- внутримашинная обработка и формирование выходных форм;
- передача по каналу связи от компьютера к пользователю;
- преобразование к виду, пригодному для восприятия пользователем.

Для мотивации к разработке и внедрению АИС способны множество факторов, к которым можно отнести: быстрый вывод информации о выполнении конкретных задач, учет единиц продукции, формирование отчетных документов.

В настоящее время автоматизированные системы используются почти во всех сферах деятельности, где необходимо обрабатывать большие объемы информации. К таким можно отнести делопроизводство в Вооруженных Силах Российской Федерации сферам деятельности.

Внедрение автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа существенно облегчит и ускорит взаимодействие между ответственными лицами по вопросам учета техники связи, а также формирования необходимых документов по их обслуживанию и ремонту. Разрабатываемая система должна обеспечить обработку и хранение данных об имеющейся технике, формирование заданных пользователем отчетных документов.

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

Автоматизированная система включает в себя базу данных и программное обеспечение для обработки данных и формирования документации. Для реализации клиентской части приложения применялась библиотека *React*.

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

1 Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа

1.1 Анализ предметной области

Современные войска связи оснащены мобильными, обладающими высокой надежностью радиорелейными, тропосферными, космическими станциями, аппаратурой высокочастотного телефонирования, тонального телеграфирования, телевизионной и фотографической аппаратурой, коммутационным оборудованием и специальной аппаратурой засекречивания сообщений.

Главным направлением развития войск связи является создание и оснащение Сухопутных войск средствами, комплексами связи и автоматизированными системами управления войсками и оружием, обеспечивающими устойчивое, непрерывное, оперативное и скрытное управление объединениями, соединениями и подразделениями в мирное время, угрожаемый период и при ведении боевых действий в самых сложных физико-географических и климатических условиях. При этом особое внимание уделяется внедрению единой системы управления войсками и оружием тактического звена и оснащению войск цифровыми средствами связи, обеспечивающими защищенный, помехоустойчивый режим обмена информацией от отдельного военнослужащего до командира соединения.

Развитие войск связи – один из самых высоких приоритетов в программе обновления армии. По мнению экспертов, по своей эффективности применение более современных средств управления и связи сопоставимо со значительным увеличением количества боевых средств или с повышением их боевых возможностей.

На сегодняшний день, в процессе развития войск связи в Вооруженных Силах Российской Федерации, актуален вопрос об автоматизации учета

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

вооружения и военной техники, и других материальных ценностей.

Образцы техники связи, поступившие в воинскую часть, распределяются по подразделениям и вводятся в эксплуатацию приказом командира воинской части в трехдневный срок.

В приказе указываются: наименования, типы, марки, индексы образцов ВВТ; подразделения, в которые они предназначены; заводские номера образцов ВВТ, корпусов, двигателей; запас ресурса или наработка на день ввода в эксплуатацию; в какой группе эксплуатации содержат образцы ВВТ и нормы расхода ресурса до конца года; присваиваемые условные номера или государственные регистрационные знаки; должности, воинские звания и фамилии членов экипажа (расчета), механиков-водителей (водителей) и других должностных лиц, за которыми закрепляются образцы ВВТ [1].

Номер и дата приказа командира воинской части о вводе ВВТ в эксплуатацию, должности, воинские звания, фамилии членов экипажей (расчетов) или других должностных лиц, за которыми закрепляются образцы ВВТ, заносятся в формуляры (паспорта) образцов ВВТ. При проведении технических обслуживаний и ремонтов их результаты также заносятся в формуляр. Для ремонта техники необходимо формировать заявку в вышестоящий орган [2].

Документы в войсках связи представляют собой немалую ценность, так же, как и материальные ресурсы. Для того, чтобы оперативно реагировать на все качественные и количественные изменения в войсках связи, нужны современные подходы к обработке документов, при этом важно уделять достаточное внимание вопросам информационной безопасности. Создание автоматизированной системы решит ряд проблем, таких как: потеря документов, затраты большого количества времени на поиск нужного документа, создание нескольких копий одного и того же документа и т.д.

На сегодня процесс учета техники в войсках частично автоматизирован с помощью пакета *Microsoft Office*, который исключает необходимость

					ПГУ 1.090501.07.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

ведения имеющихся большие размеры сводных книг учета техники. Но пакет *Microsoft Office* не сможет решить все вышеуказанные проблемы. Данная задача может быть удовлетворена путем внедрения автоматизированной системы, которая сможет обеспечить: слаженную работу всех подразделений, повышение производительность труда за счет сокращения времени на создание, обработку и поиск документов, повышение оперативности доступа к информации, настраивание разграничение доступа к документам.

В настоящее время аналогами системы автоматизации учета техники связи и формирования отчетных документов в войсках связи военного округа являются платформы: «1С Предприятие», «МойСклад», «Okdesk» и «Парус». Плюсы и минусы данных платформ приведены в таблице 1.1.

Таблица 1.1 – Плюсы и минусы аналогов

Название	Плюсы	Минусы
1С Предприятие	<ul style="list-style-type: none"> - возможность изменения конфигурации для решения своих задач; - информационно-технологическое сопровождение; - ежемесячные обновления; - российская разработка; - возможность формирования отчетов. 	<ul style="list-style-type: none"> - для доработки конфигурации нужен обученный специалист; - невозможно работать с графикой; - работает на операционных системах семейства Microsoft Windows - отсутствуют обновления языка; - отсутствует чат [3].
МойСклад	<ul style="list-style-type: none"> - возможность разграничения доступа; - шифрование данных; - российская разработка; - не требует установки на ПК; - простой в использовании. 	<ul style="list-style-type: none"> - для работы требуется интернет; - подходит для малых предприятий; - плохая техническая поддержка; - отсутствует конструктор отчетов; - отсутствует чат.

Название	Плюсы	Минусы
Okdesk	<ul style="list-style-type: none"> - для настройки не требуется специалист; - работает с большим количеством операционных систем; - удобный интерфейс; - присутствует конструктор отчетов; - хорошая техническая поддержка. 	<ul style="list-style-type: none"> - для работы требуется интернет; - сложности с поиском в базе нужной позиции; - отсутствует возможность разграничения прав пользователей; - отсутствует чат.
АИС Парус	<ul style="list-style-type: none"> - используется в государственном секторе; - доступна интеграция с продуктами фирм: 1С, Аскон, Адем, Интермех и прочими; - высокий уровень технической поддержки. 	<ul style="list-style-type: none"> - только разработчики имеют право проводить модификацию базовых модулей и приспособлять их к специфике конкретного предприятия; - стоимость этой программы определяется стоимостью одного рабочего; - отсутствие возможности работы в веб-среде; - отсутствует чат [4].

В последнее время в России поднимается вопрос импортозамещения. Формируются благоприятные условия для развития разработки отечественного конкурентоспособного программного обеспечения, основываясь на приказе Министерства связи и массовых коммуникаций РФ от 1 апреля 2015 г. №96 «Об утверждении плана импортозамещения программного обеспечения» [5].

Проанализировав недостатки существующих систем было принято решение создать новую автоматизированную систему, главные цели которой:

- вести учет техники связи подразделений;
- анализировать данные подразделений с целью эффективного управления войсками связи;

- формировать отчетные документы по заданным пользователями параметрам.

Диаграмма вариантов использования представлена на рисунке 1.1

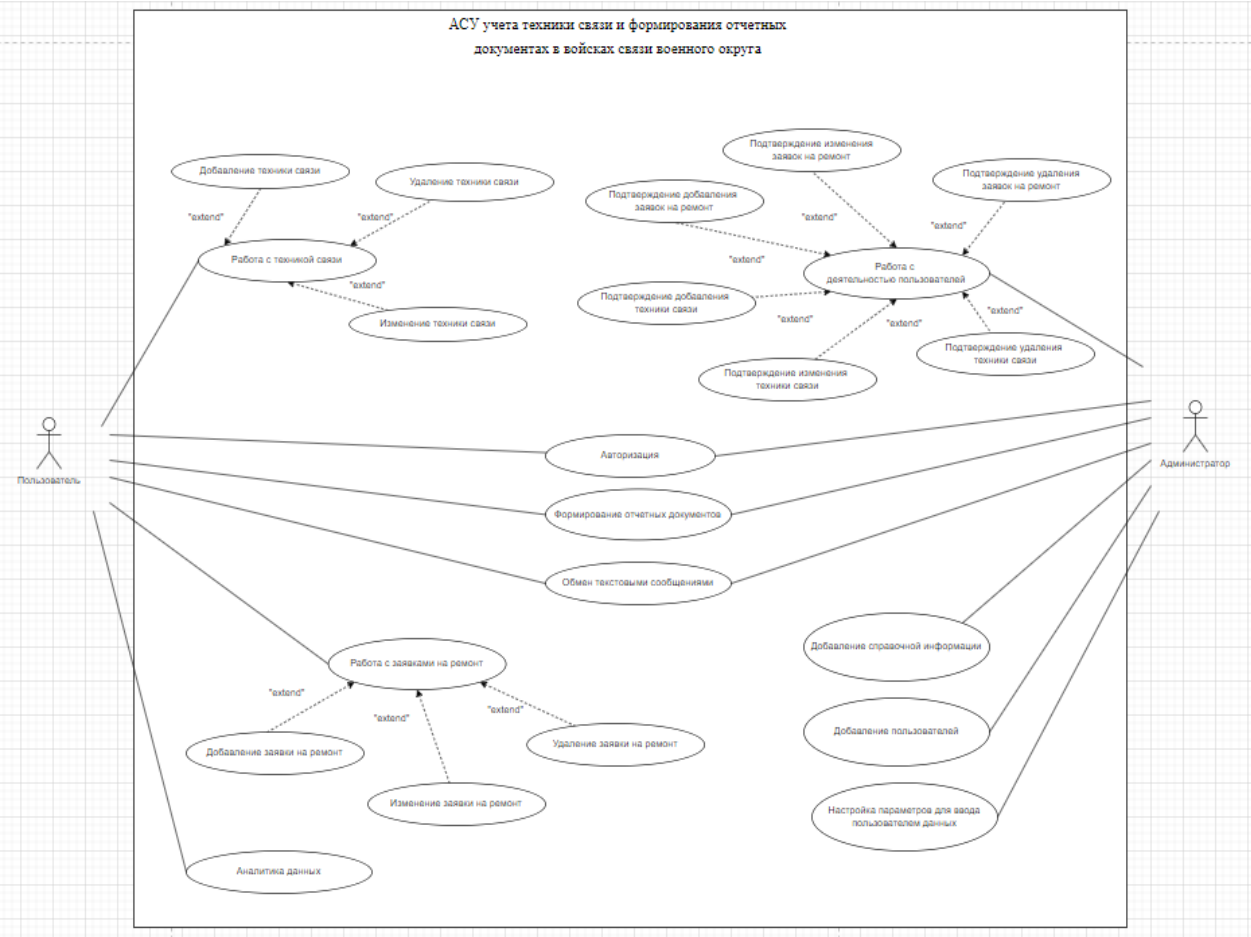


Рисунок 1.1 – Диаграмма вариантов использования

Таким образом, актуальной является задача автоматизации процесса учета техники связи в войсках связи военного округа, решаемая путем разработки новой автоматизированной системы, устраняющая недостатки уже существующих систем и легкой в использовании для должностных лиц.

1.2 Техническое задание

1.2.1 Основание для разработки

Основанием для разработки является задание на дипломное проектирование, выданное и утвержденное д.т.н., заведующим кафедрой ВТ ПГУ Митрохиным М.А.

1.2.2 Назначение разработки

Автоматизированная система учета техники связи и формирования отчетных документов в войсках связи военного округа предназначена для упрощения и ускорения взаимодействия между ответственными лицами по вопросам учета техники связи, а также формирования необходимых документов по их обслуживанию и ремонту. Система должна иметь клиент-серверную архитектуру. Клиентская часть автоматизированной системы отвечает за пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

1.2.3 Требования к программе

1.2.3.1 Требования к функциональным характеристикам

Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа должна обеспечивать выполнение следующих функций:

- отображение, изменение и отправка на сервер сведений о вооружении и военной техники;
- отображение, изменение и отправка на сервер заявок на ремонт;
- отображение, изменение и отправка на сервер справочных данных;
- отображение, изменение и отправка на сервер сведений о новых видах и типах техники;
- отображение информации в зависимости от прав пользователей;
- отображение аналитики данных о вооружении и военной техники;
- отображение и отправка введенных пользователями сообщений;

– отображение уведомлений администратора о новых заявках на ремонт или изменения информации в базе;

– настройка и отправка на сервер параметров для формирования отчетных документов;

– отображение сформированных пользователем отчетных документов.

Входными данными являются:

– сведения о видах и типах техники;

– сведения о параметрах конкретной единицы техники;

– сведения о пользователе;

– сведения о формируемых документах;

– сведения о заявках на ремонт;

– сообщения пользователей.

Выходными данными должны быть:

– хранимые в БД данные;

– сформированные документы, включающие в себя информацию, указанную пользователем.

1.2.3.2 Требования к надежности

При функционировании клиентской части системы должен осуществляться контроль входной и выходной информации.

В случае ошибки ввода-вывода должны выдаваться соответствующие сообщения и предоставляться возможность исправления ошибок и продолжения работы.

1.2.3.3 Требования к составу и параметрам технических средств

Система должна быть предназначена для работы на IBM-совместимых персональных компьютерах со следующей минимальной конфигурацией:

– процессор *Intel Core 2 Duo* 2.4 ГГц и выше;

– оперативная память 1024 Мбайт и выше;

– жесткий диск;

– монитор.

1.2.3.4 Требования к информационной и программной совместимости

Автоматизированная система учета техники связи и формирования отчетных документов в войсках связи военного округа должна работать в следующих операционных системах: MCBC 12.1, MCBC 12.4, *Astra Linux SE*, *Windows XP* и выше. На всех операционных системах должен быть установлен браузер.

1.2.4 Требования к программной документации

Комплект разрабатываемых документов должен включать:

- текст программы;
- описание программы;
- программу и методику испытаний;
- описание применения.

1.2.5 Стадии и этапы разработки

Стадии и этапы разработки клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа представлены в таблице 1.2.

Таблица 1.2 – Стадии и этапы разработки автоматизированной системы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения
1.	Анализ предметной области	8%	26.04.2022 – 29.04.2022
2.	Составление технического задания	6%	30.04.2022 – 02.05.2022
3.	Выбор инструментов разработки	8%	03.05.2022 – 06.05.2022
4.	Проектирование системы	12%	07.05.2022 – 12.05.2022
5.	Разработка клиентской части системы	40%	13.05.2022 – 01.06.2022
6.	Выполнение задания по БЖД	10%	02.06.2022 – 06.06.2022
7.	Выполнение экономического обоснования	8%	07.06.2022 – 10.06.2022
8.	Оформление пояснительной записки	8%	11.06.2022 – 14.06.2022

1.2.6 Порядок контроля и приемки

Для проверки правильности работы модуля клиентской части должно быть проведено тестирование его работы во всех режимах и выявлено функциональное соответствие предъявляемым к нему требованиям.

Проверка правильности работы должна осуществляться путем подсоединения к тестовому серверу и проверки правильности функционирования модуля по всем возможным вариантам работы системы.

Для испытания в качестве входных данных должны использоваться тестовые наборы данных.

Приёмка работоспособной клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа производится после правильного выполнения всех тестовых заданий и при наличии пояснительной записки, включающей полный комплект программной документации.

1.3 Анализ целесообразности использования заданных средств разработки

Перед разработкой клиентской части автоматизированной системы необходимо сделать выбор среды, языка и инструментов. Так как система будет веб-ориентированной, то следует рассмотреть такой язык как *JavaScript*.

Первоначально *Javascript* был разработан *Netscape* как способ добавления динамических и интерактивных элементов на веб-сайты. *JavaScript* находится под влиянием *Java* с аналогичным синтаксисом C. Данный язык соответствует спецификациям *ECMAScript*, которые были разработаны *Sun Microsystems*.

JavaScript может быть языком сценариев на стороне клиента, который предполагает, что текстовый файл *ASCII* обрабатывается браузером клиента, а не стороне сервера. Это позволяет загружать веб-страницу без связи с

основным сервером. Например, функция *JavaScript* может проверять интернет-форму перед ее отправкой, чтобы убедиться, что все указанные поля заполнены. Код *JavaScript* может выдать сообщение об ошибке до того, как какая-либо информация действительно будет передана на сервер.

Преимущества *JavaScript*:

- независимо от того, где вы размещаете *JavaScript*, он всегда выполняется в клиентской среде, чтобы сэкономить большую пропускную способность и ускорить процесс выполнения;
- способность поддерживать все современные браузеры и выдавать эквивалентный результат;
- глобальные компании поддерживают развитие *JavaScript*, создавая важные для разработчика продукты. Примером может служить *Angular* или *React.js*;
- прекрасно сочетается с другими языками программирования и может быть использован в огромном количестве приложений;
- дает возможность создавать богатые и динамичные интерфейсы.

Недостатки *JavaScript*:

- видимость кода является одним из основных недостатков;
- одна ошибка в коде *JavaScript* может остановить рендеринг всего веб-приложения;
- отсутствие множественного наследования: *JavaScript* поддерживает только одиночное наследование.

Для разработки клиентской части автоматизированной системы будет использоваться *JavaScript*, несмотря на свои недостатки, он является единственным языком сценариев на стороне клиента. Защита кода *Javascript* будет производиться с помощью инструментов обфускации.

На сегодняшний день разработка программного продукта с нуля, исключая даже обыденные вещи, может занять чрезвычайно много времени. Довольно часто разработчики тратят время на однотипные вещи, которое они

могли бы использовать на внедрение новых функций. Именно здесь на помощь приходят фреймворки и библиотеки. Они часто охватывают все обычные аспекты приложения, включая доступ к базе данных, аутентификацию, управление сессиями и многое другое.

Начиная новый проект, программист сталкивается с выбором: какой *JavaScript* фреймворк выбрать для веб-приложения – *React*, *Vue* или *Angular*. Статистика использования разработчиками фронтенд фреймворков представлена на рисунке 1.2.



Рисунок 1.2 – Статистика использования фреймворков

React — это библиотека, отвечающая за построение пользовательского интерфейса сайта. Причём такого интерфейса, где есть не только текст, но и интерактивные элементы, динамика, разные события.

Преимущества *React*:

- простота в освоении и использовании. *React* поставляется с хорошим запасом документации, руководств и учебных ресурсов. Любой разработчик, который имеет опыт работы с *JavaScript*, может легко понять и начать создавать веб-приложения с помощью *React* за несколько дней;

- простота в создании динамических веб-приложений. Для создания

динамического веб-приложения со строками *HTML* требовалось много времени, но библиотека *React* решила эту проблему. Она использует *JSX* (расширение *JavaScript*), которое представляет собой особый синтаксис, позволяющий использовать *HTML*-кавычки и синтаксис *HTML*-тегов для отображения определенных подкомпонентов;

- многоразовые компоненты. Веб-приложение, реализованное с использованием *React*, состоит из нескольких компонентов, и каждый компонент имеет свою собственную логику и элементы управления. Эти компоненты отвечают за вывод небольшого, многоразового фрагмента *HTML*-кода, который можно повторно использовать везде, где он вам нужен. Повторно используемый код помогает упростить разработку и обслуживание ваших приложений. Эти компоненты могут быть вложены в другие компоненты, что позволяет создавать сложные приложения из простых частей;

- повышение производительности. В *React* для каждого объекта *RDOM* существует соответствующий объект *VDOM*. *VDOM* — это объектное представление *RDOM*, его легковесная копия. *VDOM* содержит те же свойства, что и *RDOM*, но не может напрямую влиять на то, что отображается на экране. Манипуляции с *RDOM* являются медленными. Манипуляции с *VDOM* намного быстрее, поскольку они не отображаются (отрисовываются) на экране. Манипуляции с *VDOM* похожи на работу с проектом (или планом) здания перед началом его возведения;

- поддержка удобных инструментов. Библиотека *React* завоевала популярность благодаря наличию удобного набора инструментов. Эти инструменты делают задачу разработчиков понятной и более простой. Инструменты разработчика *React* были разработаны в качестве расширения для *Chrome* и *Firefox* и позволяют просматривать иерархии компонентов *React* в виртуальном *DOM*;

- тестирование. Приложения *React* чрезвычайно просты в тестировании.

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

Библиотека предлагает область, в которой разработчик может тестировать и отлаживать свои коды с помощью встроенных инструментов.

Недостатки *React*:

- тяжелый для небольших проектов;
- частые обновления. Для некоторых разработчиков становится трудно поддерживать код, поскольку фреймворк обновляется слишком часто;
- много абстракции. Большое количество действий происходит за кулисами. Иногда новые разработчики не имеют представления о том, как работает код;
- *React* охватывает только слои пользовательского интерфейса приложения.

Vue - это прогрессивный фреймворк для разработки пользовательских интерфейсов и одностраничных веб-приложений на языке *JavaScript*. Он решает задачи уровня представления (*view*) и упрощает работу с библиотеками. *Vue.js* можно внедрять постепенно, этим он отличается от других фреймворков.

Преимущества *Vue*:

- простота. Основная идея, лежащая в основе разработки *Vue* заключается в том, чтобы добиться хороших результатов с минимальными усилиями, насколько это возможно. *Vue.js* также отлично подходит для работы с компонентами, поскольку компоненты могут хранить все коды, такие как *HTML*, *CSS* и *JavaScript*, в одном файле;
- интеграция. Разработчики могут интегрировать *Vue* в другие фреймворки, такие как *React*, позволяющие им настраивать проект в соответствии с их соответствующими требованиями. В дополнение к *JavaScript*, он также может обрабатывать *HTML*-блоки благодаря архитектуре *MVVM*, позволяющей осуществлять двустороннюю связь;
- удобный для разработчика. По мнению различных экспертов, *Vue* не требует долгого обучения, что является большим плюсом для начинающих

программистов. Для разработки на *Vue* требуется только знание основ *JavaScript*, *HTML* и *CSS*. Существует даже плагин для браузера в *Firefox* и *Chrome*, который облегчает быстрое начало работы;

- хорошая документация. В ней показаны все варианты, предлагаемые фреймворком, и соответствующие примеры наилучшей практики. Документация на *Vue* постоянно совершенствуется и обновляется. Она также содержит простое вводное руководство и очень хороший обзор *API*.

Недостатки *Vue*:

- масштабируемость. В настоящее время *Vue* в основном используется для создания легких веб-приложений, таких как одностраничные приложения, а также пользовательских интерфейсов. Ему не хватает масштабируемости, которая позволяет создавать большие сложные приложения. Это может быть связано с тем, что фреймворк все еще является инди-проектом;

- отсутствие плагинов. *Vue* еще не так хорошо зарекомендовал себя, как *Angular* и *React*, поэтому в нем меньше плагинов, чем в последних двух;

- небольшое сообщество и поддержка.

Angular — это фреймворк для работы с *JavaScript*, поддерживаемый Google. Он используется для создания одностраничных приложений и позволяет взаимодействовать с *DOM*.

Преимущества *Angular*:

- реализация архитектуры *MVC*. С помощью архитектуры *MVC* можно изолировать логику приложения от уровня пользовательского интерфейса и поддерживать разделение задач. Контроллер получает все запросы для приложения и работает с моделью, чтобы подготовить любые данные, необходимые для представления;

- модули. Приложение на основе *Angular* можно рассматривать как мозаику, где каждый модуль необходим, чтобы иметь возможность собрать полную картину. *Angular* решает проблему использования глобальных функций, ограничивая область действия всех функций модулем, в котором она

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

была определена и использовалась;

- пользовательские директивы. Они улучшают функциональность *HTML* и подходят для динамических клиентских приложений. Все директивы начинаются с префикса *ng*, чтобы *HTML* мог их идентифицировать.

Недостатки *Angular*:

- основным недостатком использования *Angular* являются ограниченные возможности *SEO* и плохая доступность для поисковых систем;

- *Angular* многословен и сложен. Частая жалоба, которую можно услышать от разработчиков *Angular* - это многословие инструмента. И эта проблема не сильно изменилась со временем;

- сложность обучения. Если вы пригласите новых разработчиков, знакомых с *JavaScript*, использовать новый *Angular*, им будет сложно по сравнению с адаптацией *React* или *Vue*;

- отсутствие документации *CLI*.

Для данного проекта был выбран веб-фреймворк *React*, как наиболее популярный и простой в изучении, помогающий разработать веб-приложение наиболее быстро и имеющий самый большой выбор вспомогательных инструментов.

Одной из особенностей разрабатываемой автоматизированной системы является то, что она включает в себя несколько различных интерактивных графиков разных типов для отображения аналитики данных. Для реализации данного функционала выбрана библиотека *Recharts*. Главными преимуществами *Recharts* является то, что она настроена под *React*, и в ней используется компонентный подход, обеспечивающий хорошую модульность. Реализованные на *Recharts* графики достаточно легко комбинировать и переиспользовать, а также, что очень важно, библиотека позволяет достаточно сильно модифицировать стандартные элементы и переопределять внутренние обработчики.

1.4 Проектирование автоматизированной системы

1.4.1 UML - моделирование автоматизированной системы

Унифицированный язык моделирования *UML* - это стандарт, используемый для визуального описания программы. *UML* не является языком программирования, но на основании *UML*-моделей можно генерировать программный код.

Стандарты разработки *UML* приняты *Object Management Group* и широко используются для моделирования нотаций и артефактов программного обеспечения. Он отличается от других языков программирования тем, что не зависит ни от какой логики, это графический язык, который используется для создания чертежей программного обеспечения.

Для проектирования системы среда *StarUML 5.2.0* позволяет создавать диаграммы десяти типов, формирующие полное представление о проектируемой системе и ее компонентах:

- *Use case diagram* (диаграмма вариантов использования);
- *Deployment diagram* (диаграмма развертывания);
- *Statechart diagram* (диаграмма состояний);
- *Interaction diagram* (диаграмма взаимодействия);
- *Sequence diagram* (диаграмма последовательности);
- *Collaboration diagram* (диаграмма сотрудничества);
- *Class diagram* (диаграмма классов);
- *Component diagram* (диаграмма компонентов);
- *Behavior diagrams* (диаграмма поведения);
- *Activity diagram* (диаграмма деятельности);
- *Implementation diagrams* (диаграмма реализации).

Преимущества *UML*:

- объектно-ориентированным языком моделирования;
- поддержка обратного проектирования, *UML* регенерирует артефакты

уровня дизайна в соответствии с изменениями, которые внесены в структуру отношений классов;

- широко распространен и достаточно быстро развивается.

Модель сложной системы на языке *UML* фиксируются в виде специальных графических конструкций, получивших название диаграмм.

В процессе проектирования автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа целесообразно разработать следующие *UML* диаграммы:

- диаграмму вариантов использования;
- диаграмму деятельности;
- диаграмму последовательности;
- диаграмму классов;
- диаграмму развертывания.

1.4.1.1 Диаграмма вариантов использования

Диаграмма вариантов использования отображает высокоуровневый обзор отношений между вариантами использования, действующими лицами и системами. Каждый вариант использования должен обеспечивать некоторый наблюдаемый и ценный результат для участников или других заинтересованных сторон системы.

Диаграмма вариантов состоит из:

- варианты использования: овалы, обозначающие различные варианты использования, которые могут быть у пользователя;
- действующие лица: фигуры, которые представляют людей, фактически использующих варианты использования;
- ассоциации: линия между действующими лицами и вариантами использования;
- поле системы: поле, которое устанавливает область действия системы для вариантов использования.

Для автоматизированной системы учета техники связи и формирования

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

отчетных документов в войсках связи военного округа необходимо разработать диаграмму вариантов использования.

Автоматизированная система может выполнять следующие основные функции:

- авторизация;
- добавление пользователей;
- работа с техникой связи;
- добавление техники связи;
- изменение техники связи;
- удаление техники связи;
- работа с заявками на ремонт;
- добавление заявки на ремонт;
- изменение заявки на ремонт;
- удаление заявки;
- работа с деятельностью пользователя;
- подтверждение добавления заявки на ремонт;
- подтверждение изменения заявки на ремонт;
- подтверждение удаления заявки на ремонт;
- подтверждение добавления техники связи;
- подтверждение изменения техники связи;
- подтверждение удаления техники связи;
- формирование отчетных документов;
- обмен текстовыми сообщениями;
- добавление справочной информации;
- аналитика данных;
- настройка параметров для ввода пользователем данных.

Диаграмма вариантов использования автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа приведена на рисунке 1.3.

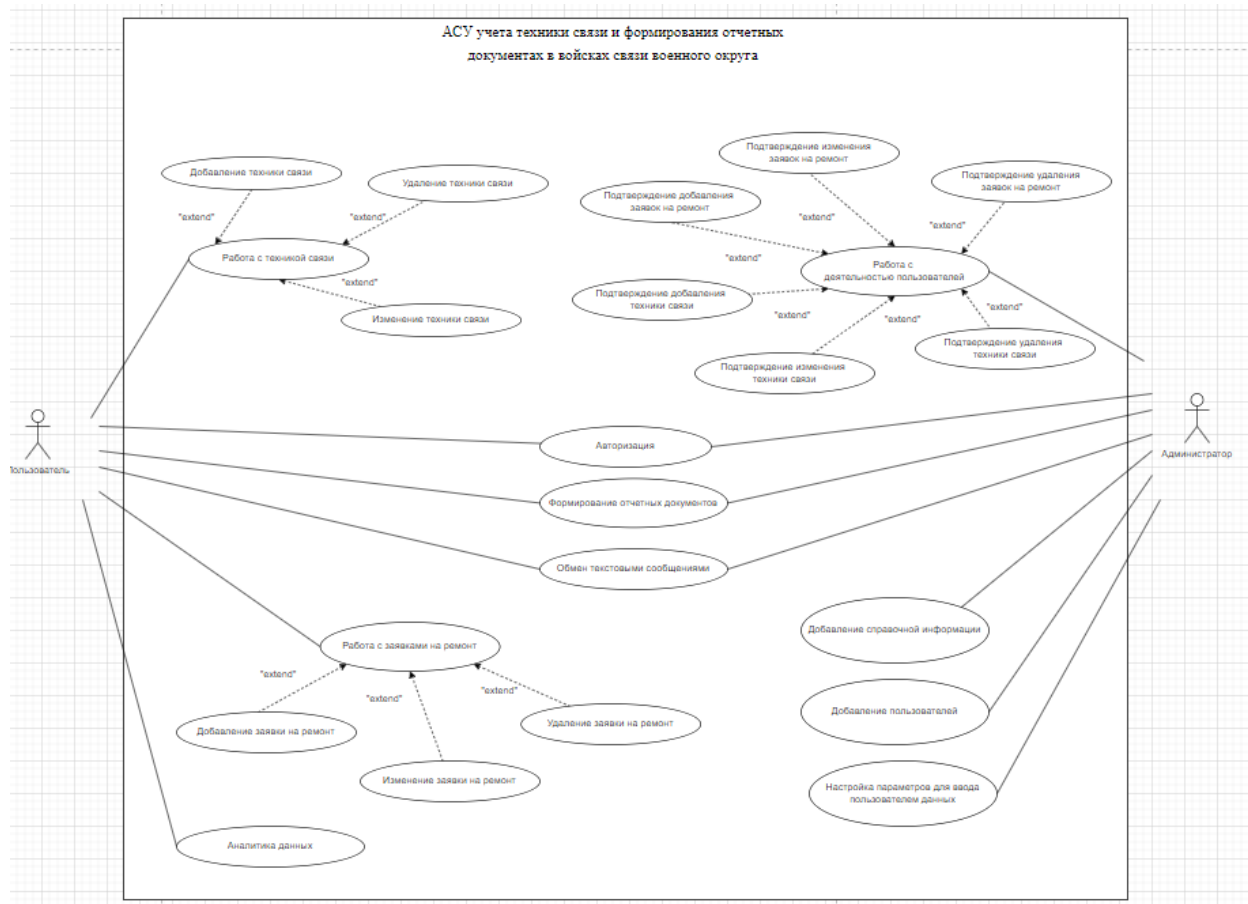


Рисунок 1.4 – Диаграмма вариантов использования

1.4.1.2 Диаграмма деятельности

Диаграмма деятельности - диаграмма в *UML*, предназначенная для описания динамических аспектов системы. Диаграммы деятельности позволяют моделировать сложный жизненный цикл объекта, с переходами из одного состояния (деятельности) в другое. Объектами выступают функции (варианты) использования из разработанной диаграммы вариантов использования.

Диаграммы деятельности основных прецедентов представлены на рисунках 1.5 — 1.26.

Диаграмма деятельности авторизации представлена на рисунке 1.5.

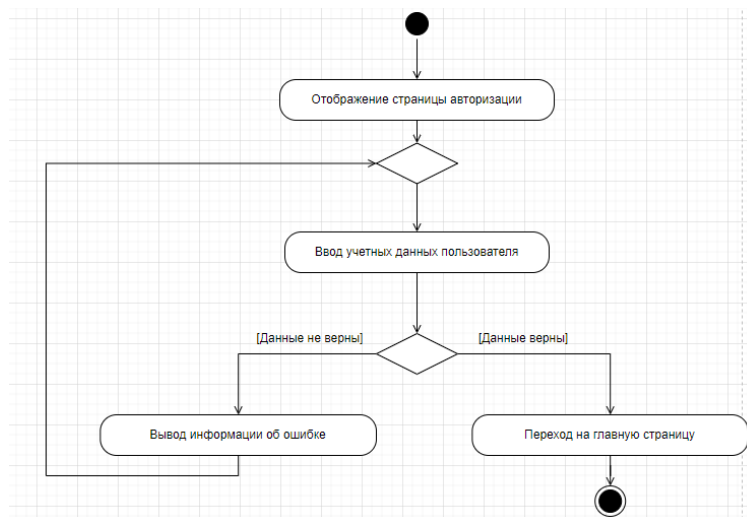


Рисунок 1.5 – Диаграмма деятельности: авторизация

Авторизация заключается в выполнении следующих шагов:

- отображение страницы авторизации;
- ввод учетных данных пользователя;
- если данные верны, то переход на главную страницу, иначе вывод информации об ошибке.

Диаграмма деятельности работа с техникой связи представлена на рисунке 1.6.

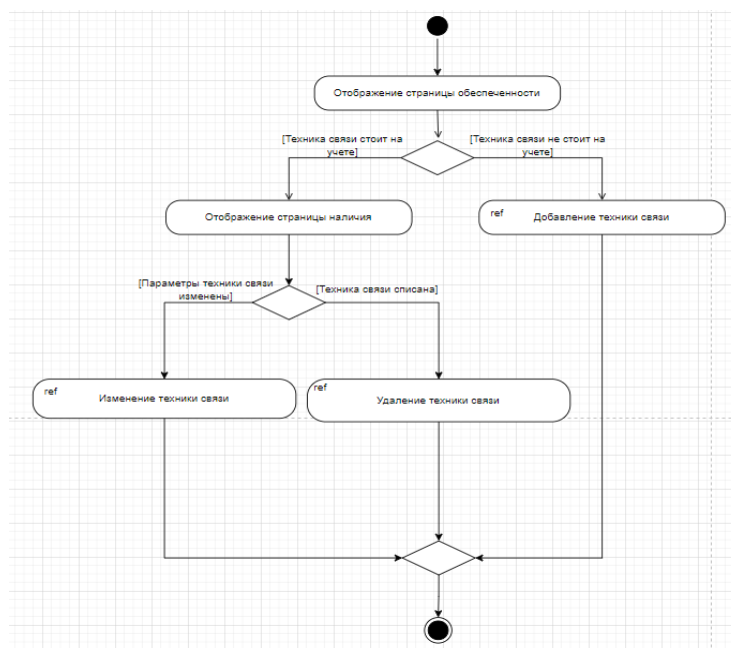


Рисунок 1.6 – Диаграмма деятельности: работа с техникой связи

Работа с техникой связи заключается в выполнении следующих шагов:

- отображение страницы обеспеченности;
- если техника связи стоит на учете, то отображается страницы наличия, иначе происходит добавление;
- если параметры техники изменились, то происходит изменение техники;
- если техника списана, то происходит ее удаление.

Диаграмма деятельности работа с заявками на ремонт представлена на рисунке 1.7.

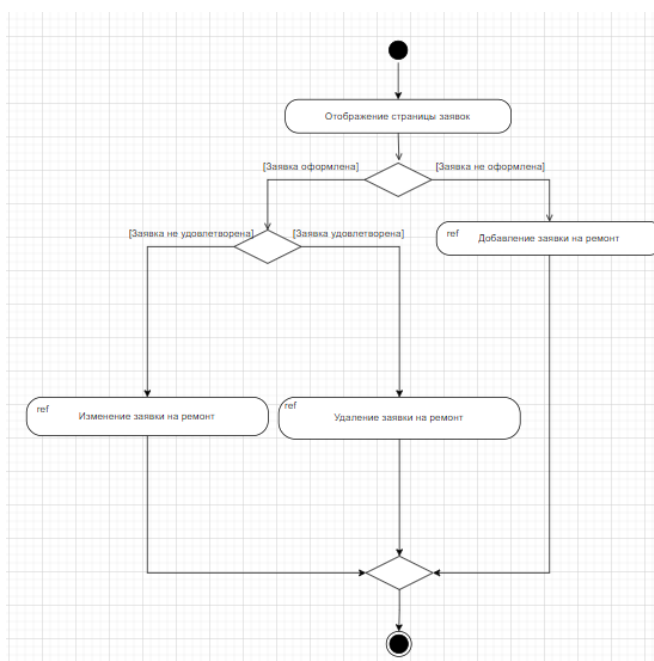


Рисунок 1.7 – Диаграмма деятельности: работа с заявками на ремонт

Работа с заявками на ремонт заключается в выполнении следующих шагов:

- отображение страницы заявок;
- если заявка не оформлена, то происходит добавление;
- если заявка не удовлетворена, то происходит изменение заявки на ремонт;
- если заявка не актуальна, то происходит ее удаление.

Диаграмма деятельности работа с деятельностью пользователей представлена на рисунке 1.8.

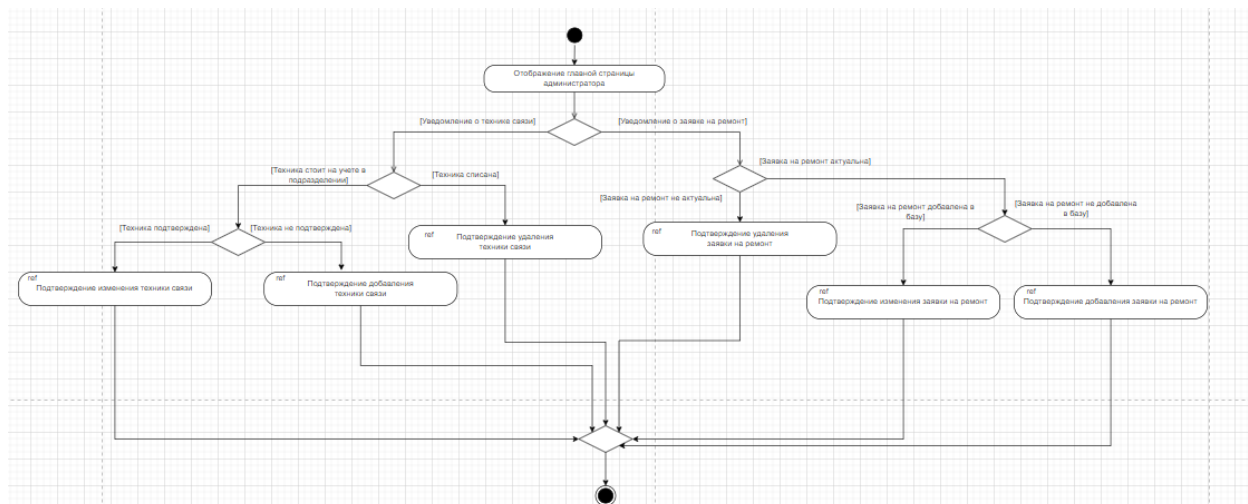


Рисунок 1.8 – Диаграмма деятельности: работа с деятельностью пользователей

Работа с деятельностью пользователей заключается в выполнении следующих шагов:

- отображение главной страницы;
- если пришло уведомление о технике, и она списана, то происходит подтверждение удаления техники;
- если пришло уведомление о технике, она стоит на учете и подтверждена в базе, то происходит подтверждение изменения техники, иначе подтверждение добавления техники;
- отображение главной страницы;
- если пришло уведомление о заявке, и она не актуальна, то происходит подтверждение удаления заявки;
- если пришло уведомление о заявке, она актуальна и подтверждена в базе, то происходит изменения заявки, иначе подтверждение добавления заявки.

Диаграмма деятельности добавления справочной информации представлена на рисунке 1.9.

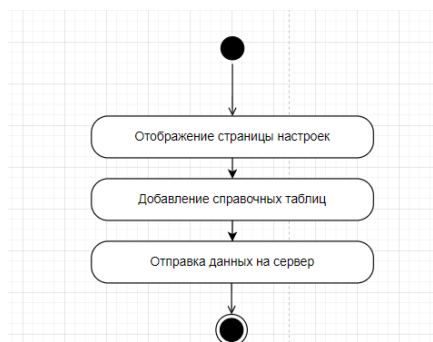


Рисунок 1.9 – Диаграмма деятельности: добавление справочной информации

Работа с деятельностью пользователей заключается в выполнении следующих шагов:

- отображение страницы настроек;
- добавление справочных таблиц;
- отправка данных на сервер.

Диаграмма деятельности добавления пользователей представлена на рисунке 1.10.

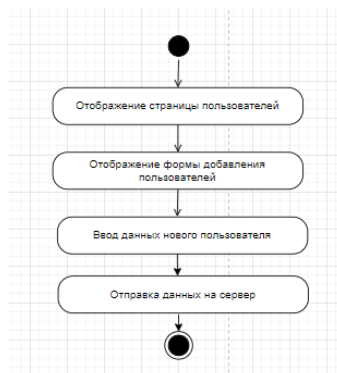


Рисунок 1.10 – Диаграмма деятельности: добавление пользователей

Работа с деятельностью добавления пользователей заключается в выполнении следующих шагов:

- отображение страницы пользователей;
- отображение формы добавления пользователей;

- ввод данных нового пользователя;
- отправка данных на сервер.

Диаграмма деятельности настройки параметров для ввода пользователем данных представлена на рисунке 1.11.

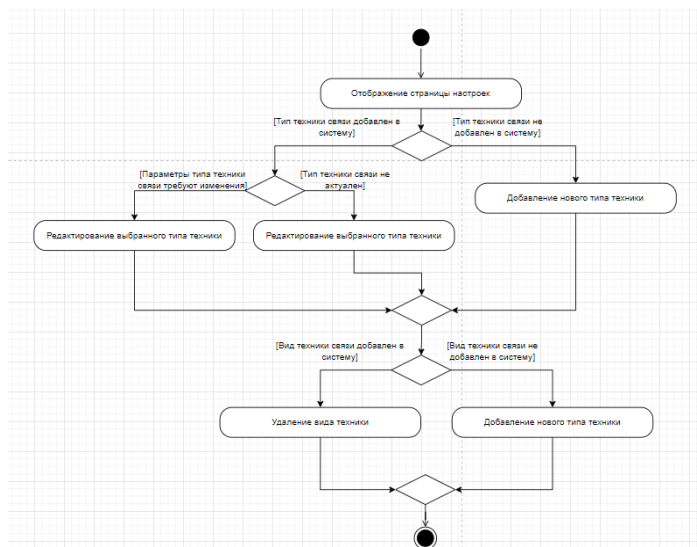


Рисунок 1.11 – Диаграмма деятельности: настройка параметров для ввода пользователем данных

Настройка параметров для ввода пользователем данных заключается в выполнении следующих шагов:

- отображение страницы настроек;
- если тип техники добавлен в систему, и он перестал быть актуальным, то происходит его удаление, иначе его изменение;
- если тип техники не добавлен в систему, то происходит его добавление;
- если вид техники добавлен в систему, то удалить его, иначе добавить.

Диаграмма деятельности формирования отчетных документов представлена на рисунке 1.12.

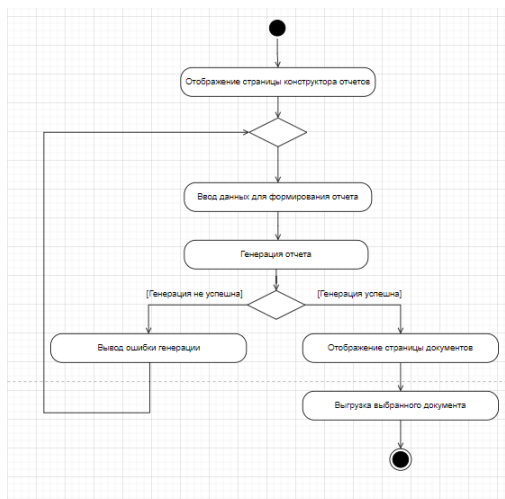


Рисунок 1.12 – Диаграмма деятельности: формирование отчетных документов

Формирование отчетных документов заключается в выполнении следующих шагов:

- отображение страницы конструкторов отчета;
- ввод данных для формирования отчета;
- генерацию отчета;
- если генерация успешна, то переход на страницу документов, иначе вывод об ошибке;
- выгрузка выбранного документа.

Диаграмма деятельности обмена текстовыми сообщениями представлена на рисунке 1.13.

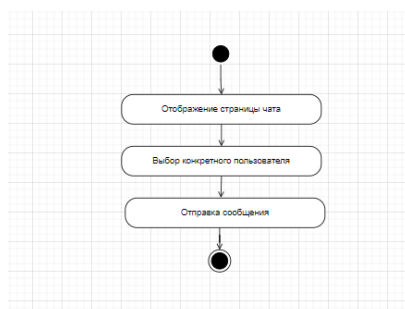


Рисунок 1.13 – Диаграмма деятельности: обмен текстовыми сообщениями

Обмен текстовыми сообщениями заключается в выполнении следующих шагов:

- отображение страницы чата;
- выбор конкретного пользователя;
- генерацию отчета;
- отправка сообщений;
- выгрузка выбранного документа.

Диаграмма деятельности аналитики данных представлена на рисунке 1.14.

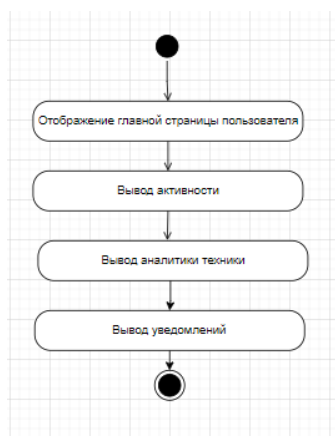


Рисунок 1.14 – Диаграмма деятельности: аналитика данных

Аналитика данных заключается в выполнении следующих шагов:

- отображение главной страницы пользователя;
- вывод активности;
- вывод аналитики техники;
- вывод уведомлений.

1.4.1.3 Диаграмма последовательности

Диаграммы последовательностей, обычно используемые разработчиками, моделируют взаимодействия между объектами в едином сценарии использования. Они иллюстрируют, как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая

ИСПОЛЬЗОВАНИЯ.

Диаграмма последовательности автоматизированной системы представлена на рисунке 1.15.

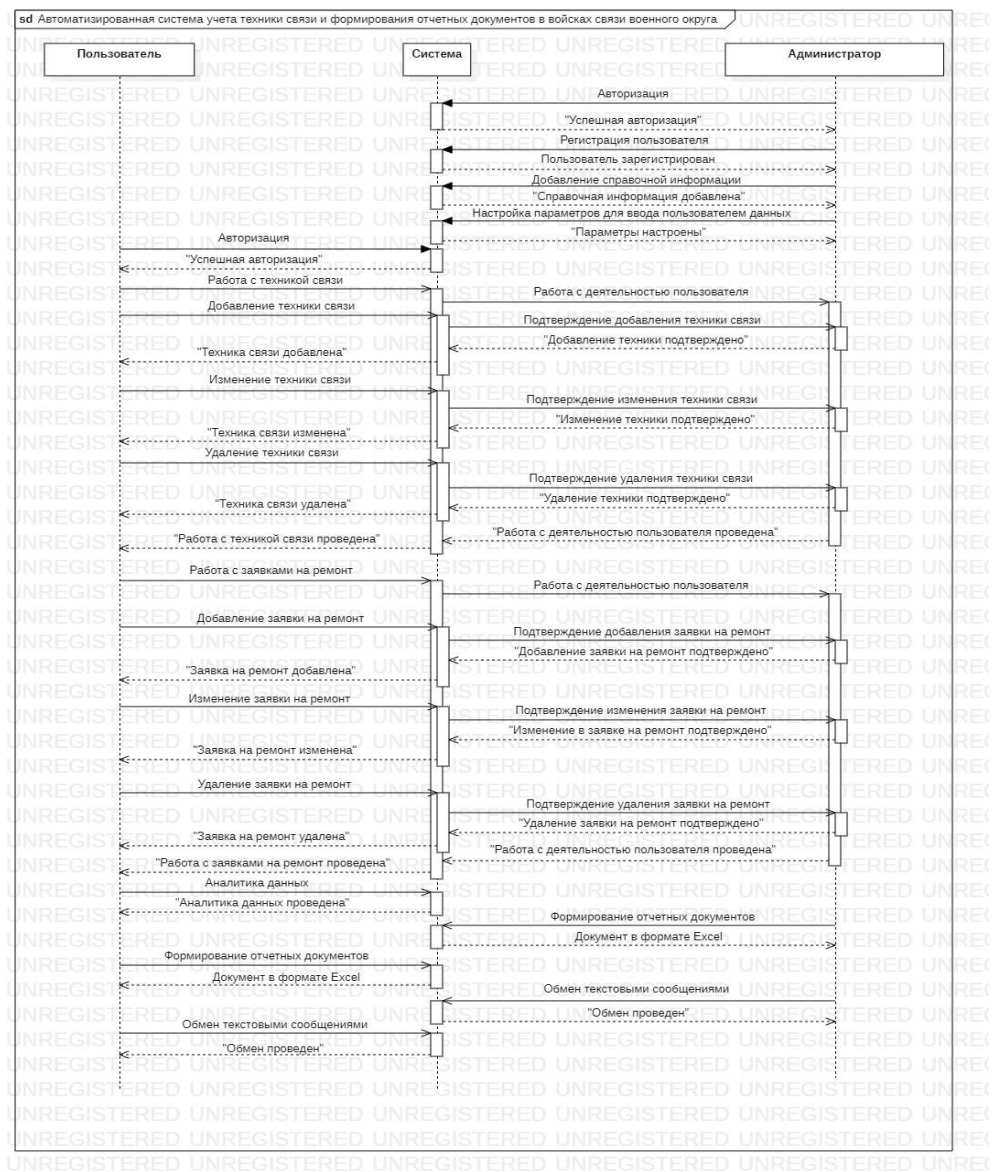


Рисунок 1.15 – Диаграмма последовательности

1.4.1.4 Диаграмма классов

В основе любой объектно-ориентированной системы лежит этап проектирования структуры классов. Именно на данном этапе на помощь приходит диаграмма классов. На диаграмме классов с помощью специальных символов изображаются типы данных программы и отношения между ними, хотя в некоторых случаях могут использоваться и другие элементы — пакеты

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.07.001 ПЗ

Лист

35

и экземпляры классов.

В процессе проектирования разработана диаграмма классов автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа, которая представлена на рисунке 1.16.

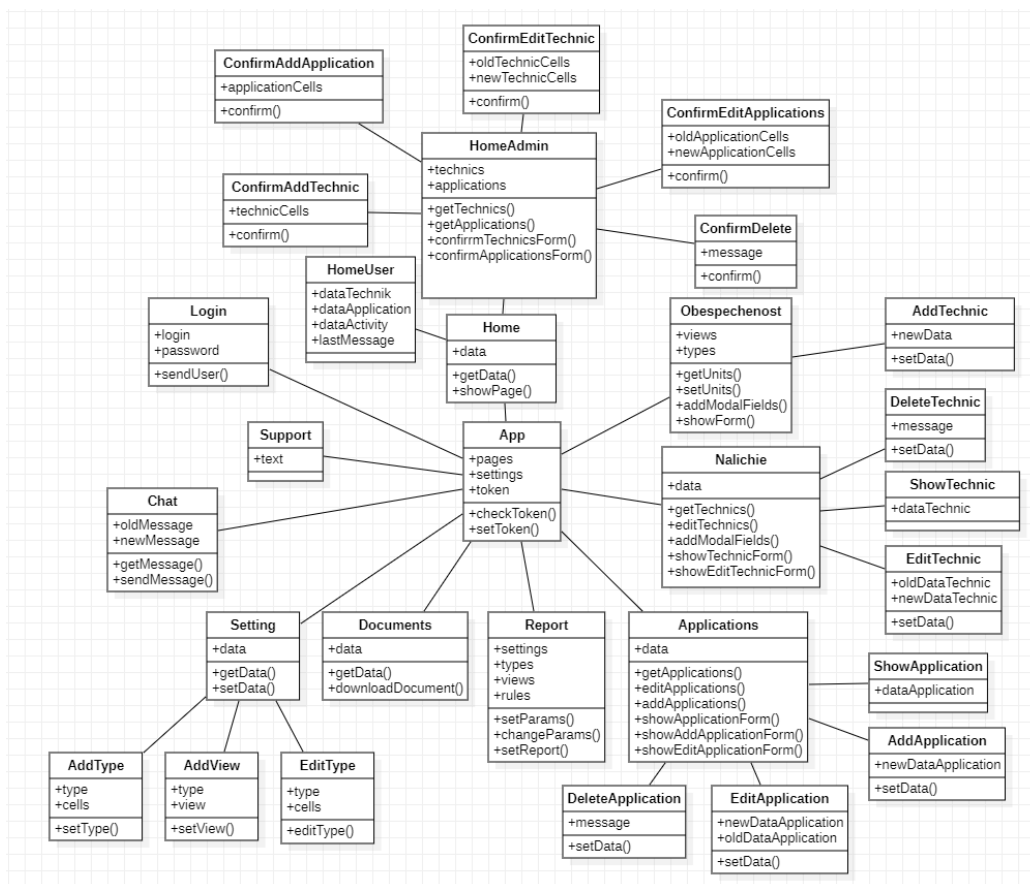


Рисунок 1.16 – Диаграмма классов

1.4.1.5 Диаграмма развертывания

На диаграмме развертывания показаны основные узлы, на которых развернуты программные компоненты. Она отображает общую конфигурацию и топологию шины, показывает расположение компонентов на каждом узле в системе, указывает на наличие физических соединений и информационных путей между аппаратными устройствами.

Согласно техническому заданию была разработана диаграмма развертывания, представленная на рисунке 1.17.

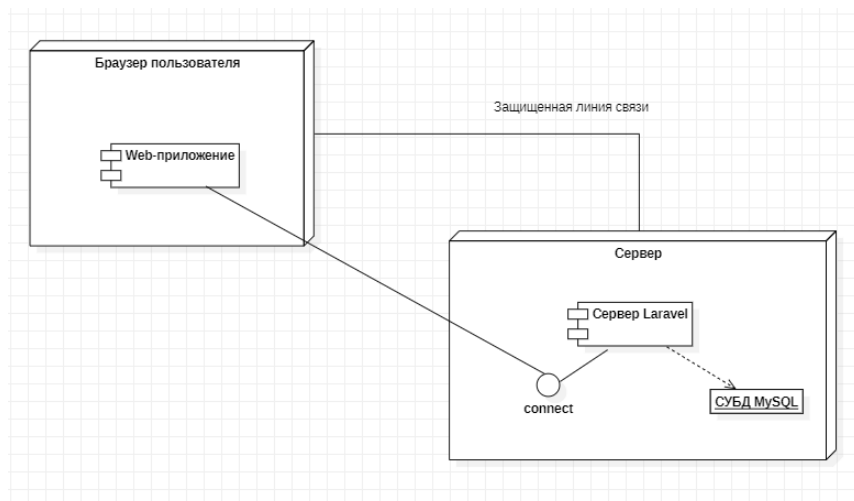


Рисунок 1.17 – Диаграмма развертывания

1.4.2 Общая архитектура автоматизированной системы

В основе разрабатываемой системы лежит архитектура «клиент–сервер», где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций). В ее состав будет входить 3 звена, представленных на рисунке 1.18

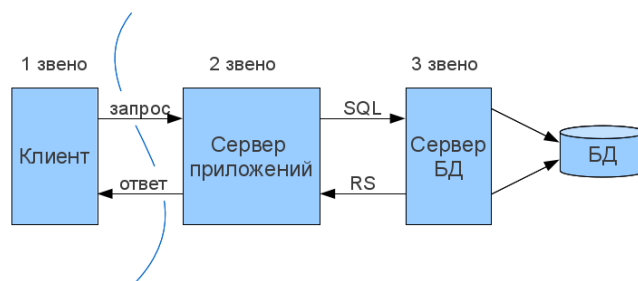


Рисунок 1.18 – Архитектура «клиент–сервер»

В состав трехзвенной архитектуры входит:

- представление данных, на стороне клиента;
- прикладной компонент, на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО);
- управление ресурсами, на сервере БД, который и представляет запрашиваемые данные.

1.4.3 Проектирование клиентской части

При разработке логической структуры приложения нужно учитывать доступность важных разделов для пользователей. Необходимые разделы должны быть всегда на виду. Пользователь не должен совершать более двух-трёх кликов мышкой, чтобы попасть в нужный раздел. Грамотно продуманная организация всех разделов, подразделов, категорий и просто страниц приложения крайне важна для его успеха. Если пользователю будет сложно ориентироваться в приложении, он просто будет ошибаться, так и не совершив целевого действия.

Проектирование структуры страниц приложения представлено на рисунках 1.19 – 1.20. Необходимо учитывать, что разрабатываемая системы имеет разграничение прав пользователей, а именно: «Пользователь», «Администратор».

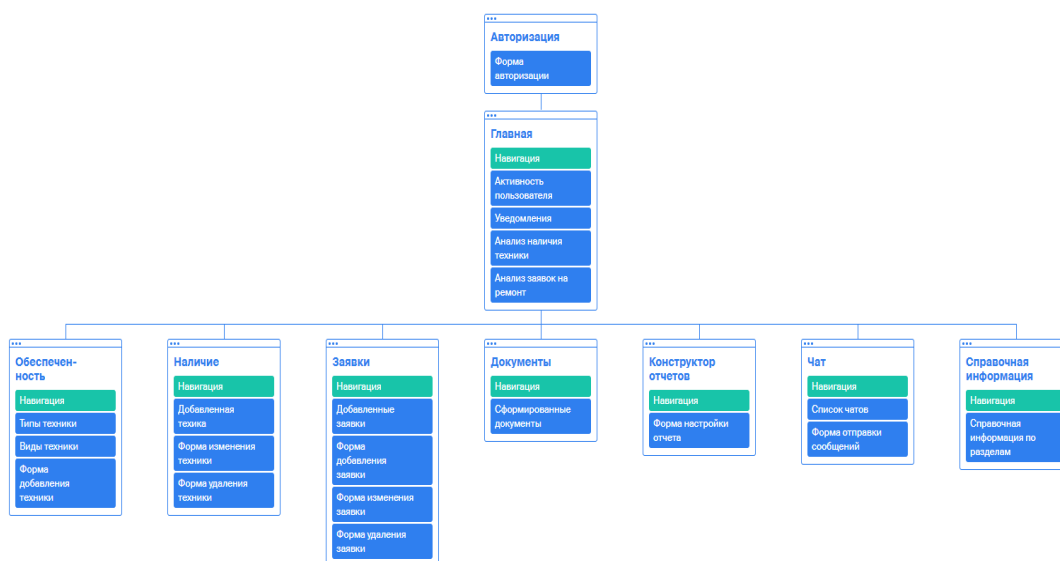


Рисунок 1.19 – Структура страниц для роли «Пользователь»

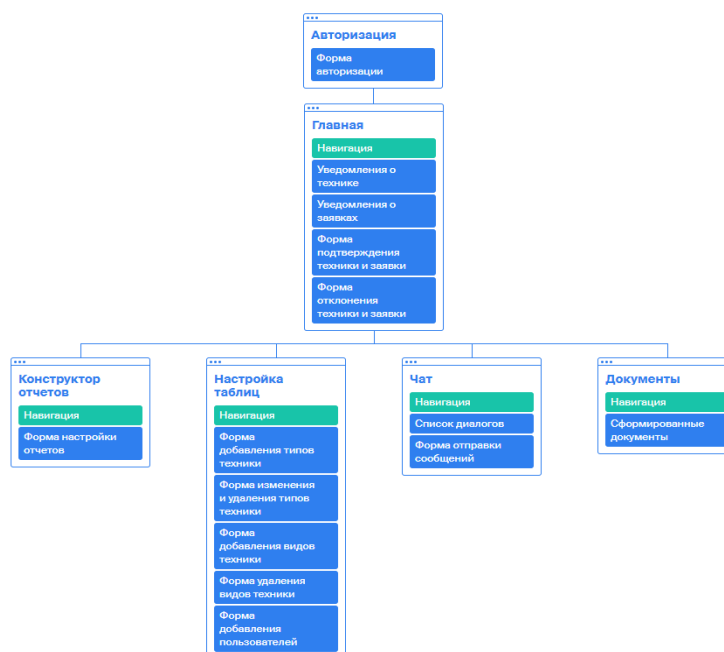


Рисунок 1.20 - Структура страниц для роли «Администратор»

При запуске системы пользователю отображается страница авторизации, которая содержит в себя форму, где расположены поля ввода логина и пароля. После успешной авторизации пользователь автоматически перенаправляется на главную страницу, где расположен компонент навигации. Навигация доступна на всех страницах веб-приложения.

Компонент навигации должен содержать в себе все доступные переходы. Для обычного пользователя доступны следующие переходы:

- главная страница;
- страница обеспеченности;
- страница наличия;
- страница заявок на ремонт;
- страница чата;
- страница настроек отчетов;
- страница справочной информации;
- страница сформированных документов;
- выход из системы.

Для администратора доступны следующие переходы:

- главная страница;
- страница чата;
- страница настроек отчетов;
- страница справочной информации;
- страница настроек данных;
- страница сформированных документов;
- выход из системы.

Необходимо визуально выделять выбранную пользователем страницу.

Структура навигации представлена на рисунках 1.21, 1.22.

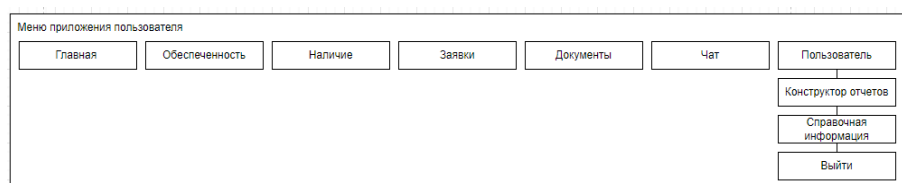


Рисунок 1.21 – Структура навигации для простого пользователя

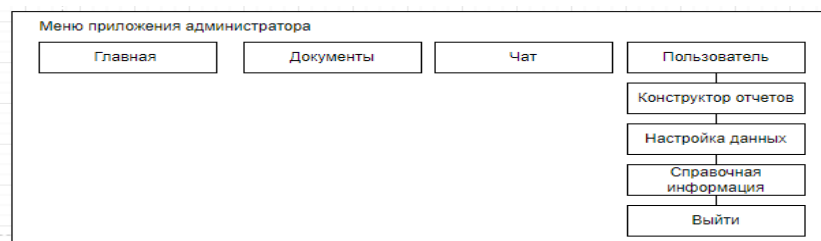


Рисунок 1.22 – Структура навигации для администратора

Главная страница должна содержать в себе всю проанализированную информацию пользователя, в которую входят:

- активность пользователя;
- последние уведомления;
- анализ наличия техники;
- анализ отправленных заявок на ремонт.

Отображение активности пользователя необходимо представить в графике, в котором по оси x будет указываться дата, а по оси y количество добавленных единиц техники и заявок.

Главная страница для администратора будет принимать другой вид и содержать в себе:

- уведомления об изменении техники и заявок;
- уведомления о добавлении техники и заявок;
- уведомления об удалении техники и заявок.

Для добавления техники пользователь переходит на страницу обеспеченности, на которой должна выводиться информация о всех доступных типах и видов техники

Для проверки добавления, редактирования и удаления техники пользователь переходит на страницу наличия, которая должна содержать в себе информацию о всей добавленной техники в подразделении.

Если техника добавлена, то пользователь имеет возможность сформировать заявку на ремонт. Он переходит на страницу заявок, которая должна содержать в себе информацию о всех ранее добавленных заявках на ремонт в подразделении.

Администратор и пользователь должны иметь возможность сформировать отчет по своим параметрам. В параметры необходимо включить:

- название отчета;
- таблица вывода;
- правило вывода;
- диапазон дат;
- пользовательские поля.

Администратор и пользователь должны иметь возможность выгружать сформированные по своим параметрам документы. Они переходят на страницу документов, которая должна содержать в себе список всех сгенерированных ранее документов с указанием даты и названия.

Если в процессе добавления или изменения техники и заявок возникают затруднения, пользователь может перейти на страницу чата и задать вопрос

администратору. Также пользователь может перейти на страницу справочной информации.

При изменении в войсках типов и видов техники, администратор может зайти на страницу настройки данных и в соответствии с изменениями внести поправки в таблицы базы данных. Также при создании новых подразделений администратор на этой же странице может добавить пользователей.

Карта экранных форм приведена на рисунке 1.23.

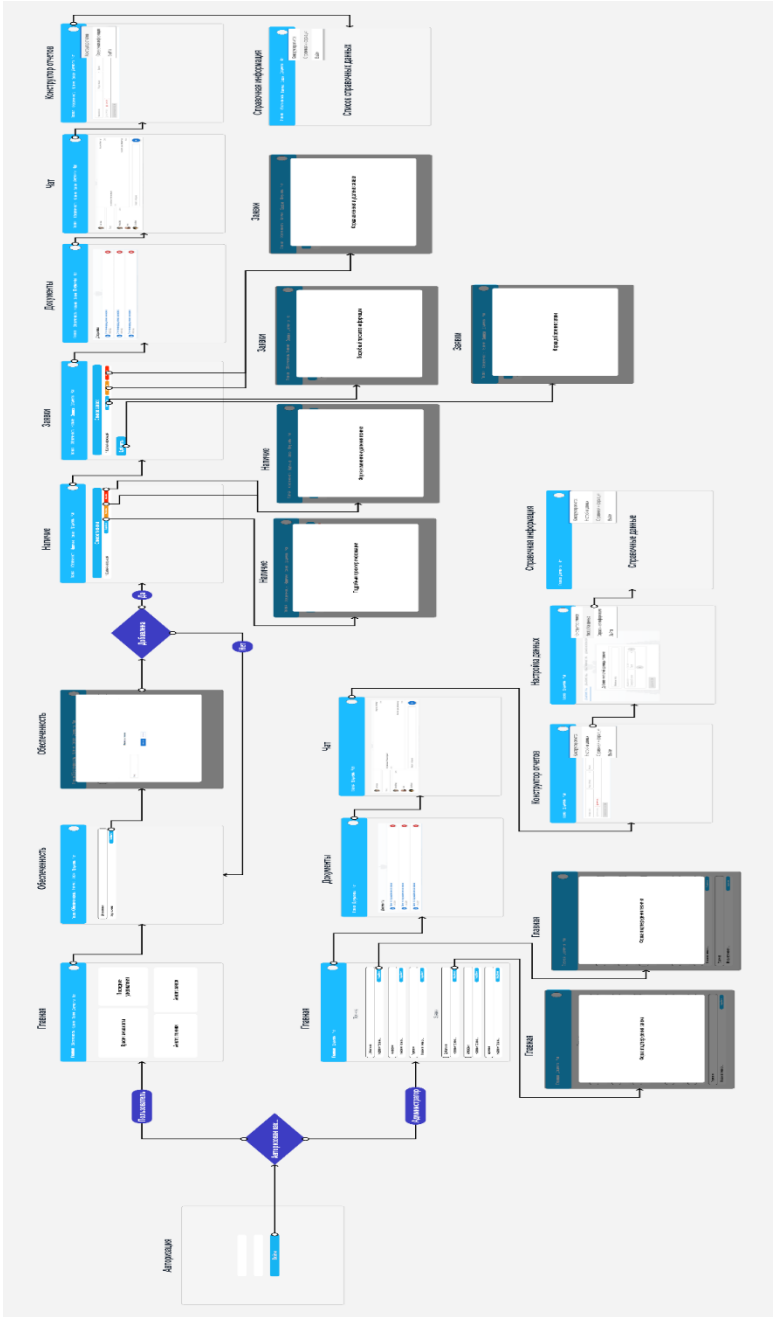


Рисунок 1.23 – Карта экранных форм приложения

1.5 Описание клиентской части

1.5.1 Общие сведения

Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа написана на языке программирования JavaScript, с использованием фреймворка React, в среде визуального программирования VS Code. Для функционирования клиентской части должен быть развернут сервер Laravel. Листинг программы приведен в приложение Б.

1.5.2 Функциональное назначение

Клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа предназначена для упрощения и ускорения взаимодействия между ответственными лицами по вопросам учета техники связи, а также формирования необходимых документов по их обслуживанию и ремонту

1.5.3 Описание логической структуры

Структурная схема клиентской части приложения представлена на рисунке 1.24.

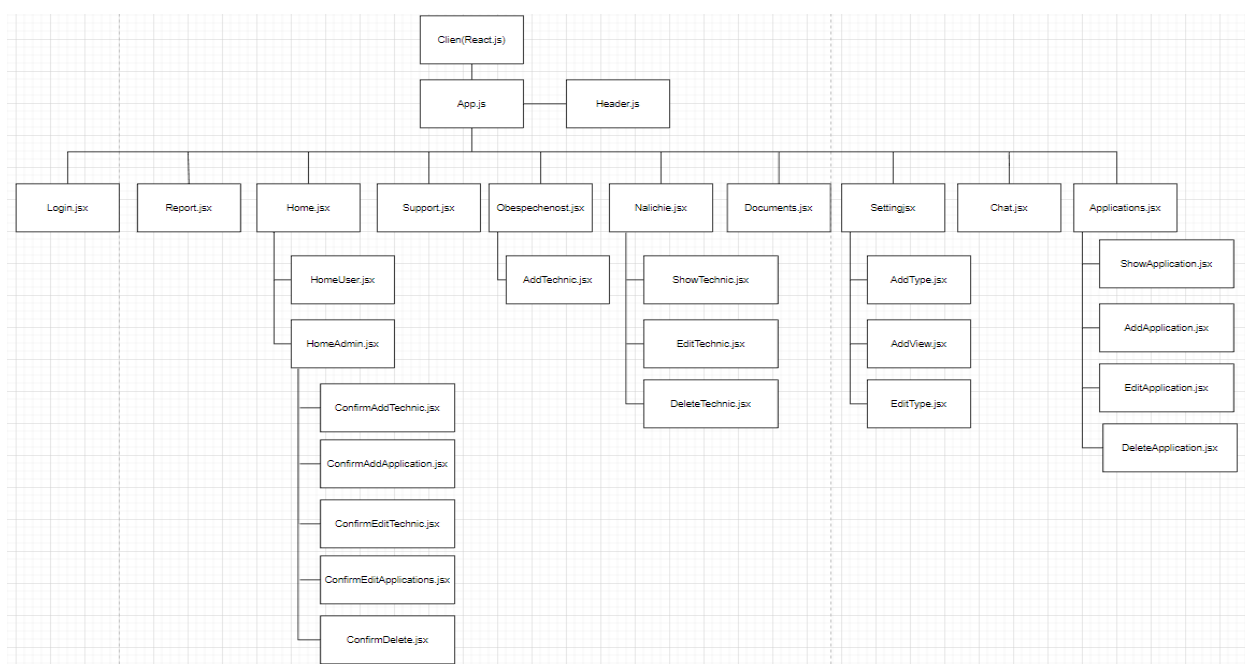


Рисунок 1.24 – Структурная схема клиентской части

Вызов модулей клиентской части приложения производится в соответствии с рисунком 1.36. При запуске программы на выполнение запускается главный модуль App, который связан непосредственно с остальными. Описание основных модулей приведено в таблице 1.3.

Таблица 1.3 – Описание основных модулей клиентской части

Название модуля	Название подмодуля	Назначение
<i>Client</i>	<i>App.js</i>	Вызывает остальные подмодули
<i>App.js</i>	<i>Header.jsx</i>	Навигация по приложению
	<i>Login.jsx</i>	Страница авторизации
	<i>Home.jsx</i>	Главная страница с аналитикой данных пользователя
	<i>Report.jsx</i>	Страница настроек отчетов
	<i>Setting.jsx</i>	Страница настроек таблиц
	<i>Obespechenost.jsx</i>	Страница вывода доступной для добавления техники
	<i>Nalichie.jsx</i>	Страница вывода и редактирования добавленной техники
	<i>Documents.jsx</i>	Страница вывода сформированных документов
	<i>Support.jsx</i>	Страница справочной информации
	<i>Chat.jsx</i>	Страница чата
	<i>Applications.jsx</i>	Страница заявок на ремонт
<i>Home.jsx</i>	<i>HomeUser.jsx</i>	Вывод компонентов главной страницы для пользователя
	<i>HomeAdmin.jsx</i>	Вывод компонентов главной страницы для администратора
<i>HomeAdmin.jsx</i>	<i>ConfirmAddTechnic.jsx</i>	Форма подтверждения добавления техники
	<i>ConfirmAddApplication.jsx</i>	Форма подтверждения добавления заявки на ремонт

Название модуля	Название подмодуля	Назначение
	<i>ConfirmEditTechnic.jsx</i>	Форма подтверждения изменения техники
	<i>ConfirmEditApplications.jsx</i>	Форма подтверждения изменения заявки на ремонт
	<i>ConfirmDelete.jsx</i>	Форма подтверждения удаления
<i>Obespechenost.jsx</i>	<i>AddTechnic.jsx</i>	Форма добавления техники
<i>Nalichie.jsx</i>	<i>ShowTechnic.jsx</i>	Подробное отображение информации о технике
	<i>EditTechnic.jsx</i>	Форма изменения техники
	<i>DeleteTechnic.jsx</i>	Форма удаления техники
<i>Setting.jsx</i>	<i>AddType.jsx</i>	Форма добавления типов техники и справочных таблиц
	<i>AddView.jsx</i>	Форма добавления видов техники
	<i>EditType.jsx</i>	Форма изменения типов техники и справочных таблиц
	<i>AddUser.jsx</i>	Форма добавления пользователей
<i>Applications.jsx</i>	<i>ShowApplication.jsx</i>	Подробное отображение информации о заявке на ремонт
	<i>AddApplication.jsx</i>	Форма добавления заявки на ремонт
	<i>EditApplication.jsx</i>	Форма изменения заявки на ремонт
	<i>DeleteApplication.jsx</i>	Форма удаления заявки на ремонт

Описание созданных обработчиков событий приложения приведено в таблице 1.4

Таблица 1.4 – Описание обработчиков событий

Название модуля	Название функции	Примечание
<i>App.js</i>	<i>checkToken()</i>	Функция выполняет проверку токена пользователя
	<i>setToken()</i>	Функция выполняет создание и запись токена пользователя в локальное хранилище
<i>Header.jsx</i>	<i>logout()</i>	Функция выполняет выход из приложения
	<i>handleOpenNavMenu(event)</i>	Функция выполняет открытие подменю пользователя
	<i>handleCloseNavMenu(event)</i>	Функция выполняет закрытие подменю пользователя
	<i>navigate(page)</i>	Функция выполняет переход на указанную страницу
<i>Login.jsx</i>	<i>sendUser()</i>	Функция выполняет запрос авторизации на сервер
	<i>handleClick()</i>	Функция выполняет обработчик отправки формы авторизации
	<i>navigate(page)</i>	Функция выполняет переход на указанную страницу
<i>Home.jsx</i>	<i>getData()</i>	Функция выполняет запрос данных на сервер о пользователе
	<i>showPage(role)</i>	Функция выполняет выбор вывода страницы в зависимости от роли
<i>HomeAdmin.jsx</i>	<i>getTechnics()</i>	Функция выполняет запрос данных на сервер о технике
	<i>getApplications()</i>	Функция выполняет запрос данных на сервер о заявках
	<i>showAddTechnicsForm()</i>	Функция выполняет отображение формы подтверждения добавления техники
	<i>showAddApplicationsForm()</i>	Функция выполняет отображение формы подтверждения добавления заявки на ремонт

Название модуля	Название функции	Примечание
	<i>showEditTechnicsForm()</i>	Функция выполняет отображение формы подтверждения изменения техники
	<i>showEditApplicationsForm()</i>	Функция выполняет отображение формы подтверждения заявки на ремонт
	<i>showDeleteForm()</i>	Функция выполняет отображение удаления техники или заявки на ремонт
<i>ConfirmAddTechnic.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о конкретной технике
	<i>confirm()</i>	Функция выполняет подтверждение добавления техники
<i>ConfirmAddApplication.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о конкретной заявке на ремонт
	<i>confirm()</i>	Функция выполняет подтверждение добавления заявки на ремонт
<i>ConfirmEditTechnic.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о конкретной технике
	<i>confirm()</i>	Функция выполняет подтверждение изменения заявки на ремонт
<i>ConfirmEditApplications.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о конкретной заявке на ремонт
	<i>confirm()</i>	Функция выполняет подтверждение изменения заявки на ремонт
<i>ConfirmDelete.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о конкретной технике или заявке на ремонт
	<i>confirm()</i>	Функция выполняет подтверждение удаления
<i>Report.jsx</i>	<i>setParams()</i>	Функция выполняет запрос на сервер о выбранных параметрах

Название модуля	Название функции	Примечание
	<i>changeParams()</i>	Функция выполняет изменение параметров отчета
	<i>setReport()</i>	Функция выполняет отправку на сервер данных для формирования отчета
<i>Setting.jsx</i>	<i>showTab()</i>	Функция выполняет отображение выбранной секции настроек
<i>AddType.jsx</i>	<i>setType()</i>	Функция выполняет отправку данных на сервер о созданном типе техники и справочных данных
<i>AddView.jsx</i>	<i>getType()</i>	Функция выполняет запрос данных на сервер о типах техники
	<i>setView()</i>	Функция выполняет отправку данных на сервер о созданном виде техники
<i>EditType.jsx</i>	<i>getType()</i>	Функция выполняет запрос данных на сервер о типах техники
	<i>setType()</i>	Функция выполняет отправку данных на сервер о созданном типе техники и справочных данных
<i>Obespechenost.jsx</i>	<i>getUnits()</i>	Функция выполняет запрос данных на сервер о единицах техники
	<i>showForm()</i>	Функция выполняет отображение формы добавления конкретной единицы техники
<i>AddTechnic.jsx</i>	<i>getData(id)</i>	Функция выполняет запрос данных на сервер о полях конкретного типа
	<i>setData()</i>	Функция отправки данных о добавленной технике
<i>Nalichie.jsx</i>	<i>getTechnics()</i>	Функция выполняет запрос данных на сервер о технике подразделения
	<i>editTechnics()</i>	Функция выполняет запрос данных на сервер о изменение техники подразделения

Название модуля	Название функции	Примечание
	<i>addModalFields(id)</i>	Функция выполняет заполнение массива конкретной техники
	<i>showTechnicForm(id)</i>	Функция выполняет отображение информации о конкретной технике
	<i>showEditTechnicForm(id)</i>	Функция выполняет отображение формы изменения конкретной техники
	<i>showDeleteForm(id)</i>	Функция выполняет отображение формы удаления техники
<i>EditTechnic.jsx</i>	<i>setData(id)</i>	Функция отправки данных о измененной технике
<i>DeleteTechnic.jsx</i>	<i>setData(id)</i>	Функция отправки данных о удаленной технике
<i>Documents.jsx</i>	<i>getData()</i>	Функция запроса данных о сконфигурированных документах
	<i>downloadDocument()</i>	Функция выгрузки документов
<i>Chat.jsx</i>	<i>getMessage()</i>	Функция запроса данных с сервера о сообщениях пользователя
	<i>sendMessage()</i>	Функция отправки сообщений пользователя на сервер
<i>Applications.jsx</i>	<i>getApplications()</i>	Функция выполняет запрос данных на сервер о заявках на ремонт подразделения
	<i>editApplications(id)</i>	Функция выполняет отправку данных на сервер о изменении заявки на ремонт
	<i>addApplications(id)</i>	Функция выполняет отправку данных на сервер о добавлении заявки на ремонт
	<i>showApplicationForm()</i>	Функция выполняет отображение информации о конкретной заявке на ремонт

Название модуля	Название функции	Примечание
	<i>showAddApplicationForm()</i>	Функция выполняет отображение формы добавления заявки на ремонт
	<i>showEditApplicationForm()</i>	Функция выполняет отображение формы изменения конкретной заявки на ремонт
<i>AddApplication.jsx</i>	<i>setData(id)</i>	Функция отправки данных о добавленной заявке на ремонт
<i>EditApplication.jsx</i>	<i>setData(id)</i>	Функция отправки данных о измененной заявке на ремонт
<i>DeleteApplication.jsx</i>	<i>setData(id)</i>	Функция отправки данных о удаленной заявке на ремонт
<i>AddUser.jsx</i>	<i>setUser()</i>	Функция отправки данных о созданном пользователе

Страница входа в программу представлена на рисунке 1.25.

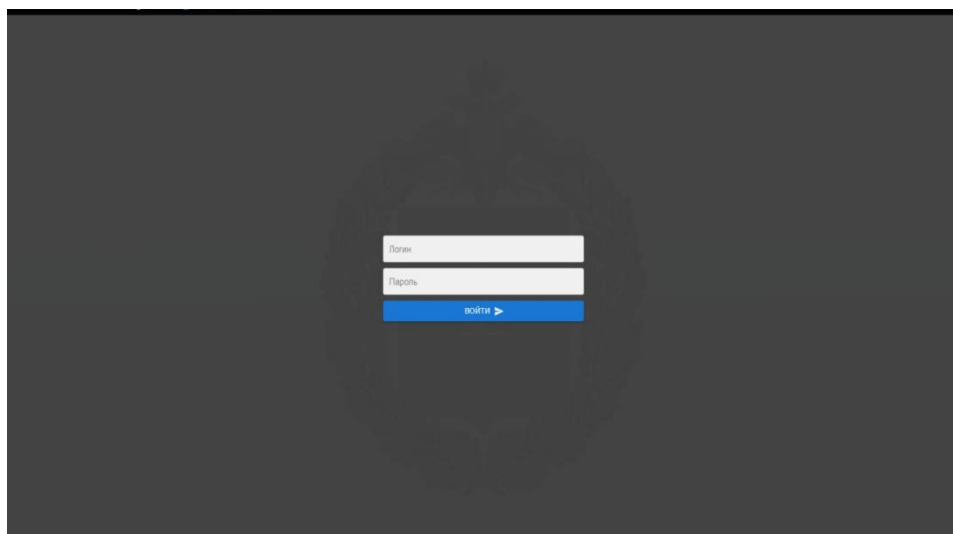


Рисунок 1.25 – Форма входа в программу

На странице входа в приложение размещены следующие элементы:

- поле для ввода логина, тип «text»;
- поле для ввода пароля, тип «password»;
- кнопка для входа в программу, тип «submit».

Главная страница пользователя представлена на рисунке 1.26.

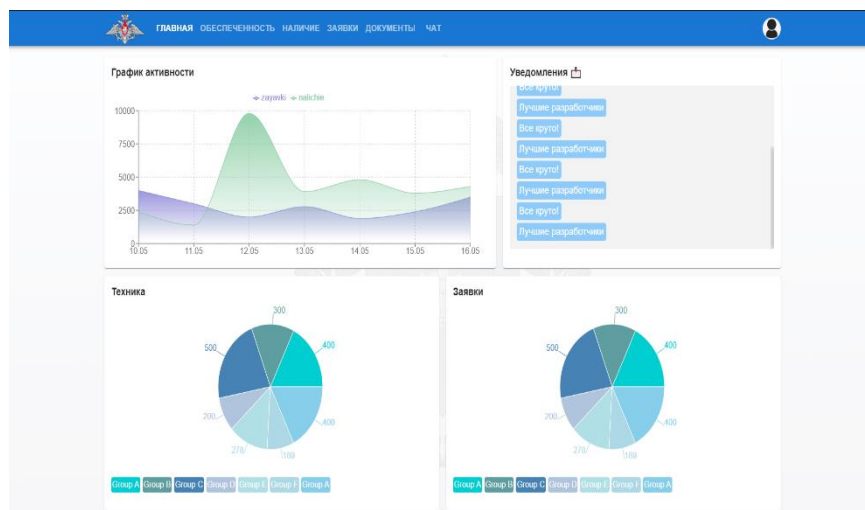


Рисунок 1.26 – Главная страница пользователя

На главной странице пользователя размещены следующие элементы:

- график активности пользователя;
- последние уведомления;
- статистика техники подразделения;
- статистика заявок на ремонт подразделения.

Графики страницы реализованы на основе компонентов библиотеки Recharts, а именно:

- *Line*;
- *Legend*;
- *AreaChart*;
- *Area*;
- *Tooltip*;
- *CartesianGrid*;
- *XAxis*;
- *YAxis*;
- *PieChart*;
- *Pie*;
- *Cell*.

Главная страница администратора представлена на рисунке 1.27.

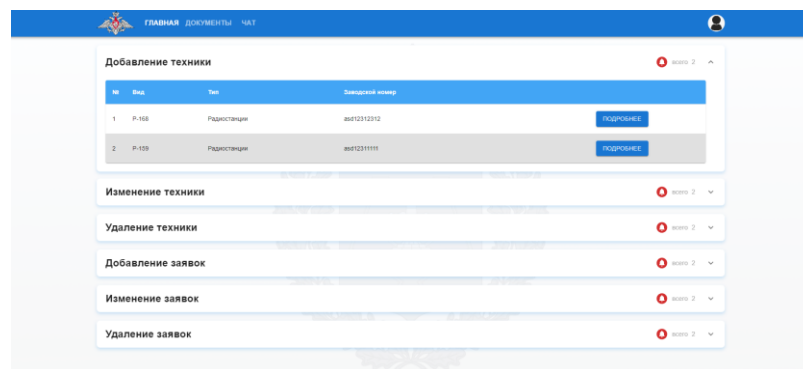


Рисунок 1.27 – Главная страница администратора

На главной странице администратора размещены следующие элементы:

- уведомления о подтверждении добавления техники;
- уведомления о подтверждении изменения техники;
- уведомления о подтверждении удаления техники;
- уведомления о подтверждении добавления заявки на ремонт;
- уведомления о подтверждении изменения заявки на ремонт;
- уведомления о подтверждении удаления заявки на ремонт.

При выборе конкретного типа уведомлений раскрывается таблица, которая содержит краткое описание и кнопку подробнее. После нажатия на кнопку подробнее отобразится форма, поля которой формируются в зависимости от данных с сервера. В данной форме отображается вся подробная информация и кнопка принятия или отклонения с пояснением причины.

Страница обеспеченности представлена на рисунке 1.28.

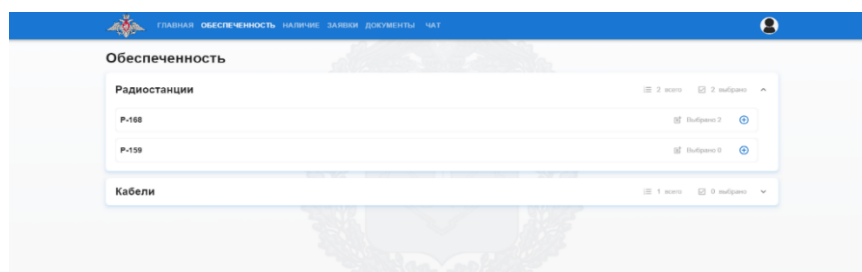


Рисунок 1.28 – Страница обеспеченности

На странице обеспеченности размещены следующие элементы:

- все типы техники;
- все виды техники;
- количество выбранной техники каждого вида и типа.

При выборе конкретного типа раскрывается список видов. Секция вида содержит:

- название вида;
- количество ранее добавленных единиц техники данного вида;
- кнопка добавления;

После нажатия на кнопку добавления открывается форма, поля которой формируются в зависимости от данных с сервера. Далее при условии заполнения всех полей доступна кнопка отправить. В результате добавления система оповещает пользователя сообщением.

Страница наличия представлена на рисунке 1.29.

№	Вид	Тип	Заказный номер	Статус	
1	☐	Радиостанция	№012121212	Принята	<button>изменить</button>
2	☐	Радиостанция	№012121111	Принята	<button>изменить</button>

Рисунок 1.29 – Страница наличия

На странице наличия размещена таблица со всеми единицами техники. Пользователь может нажать на кнопку изменить определенной техники, после чего откроется форма с ранее введенными данными. На форме также находится поле с причиной изменения.

Страница заявок представлена на рисунке 1.30.

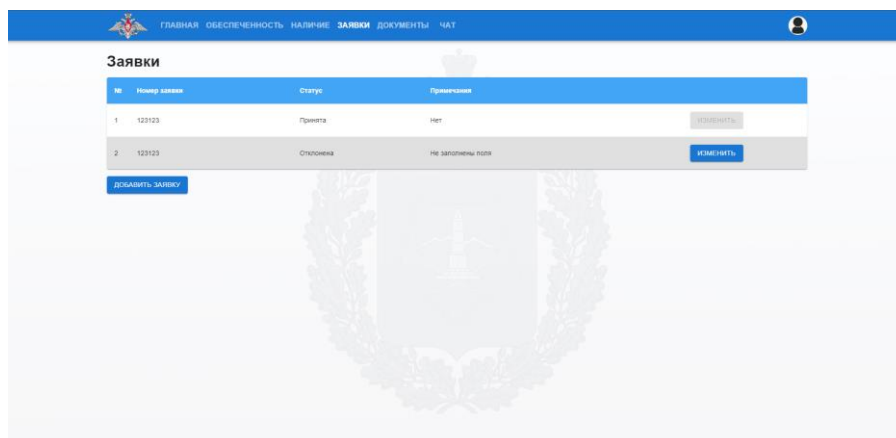


Рисунок 1.30 – Страница заявок

На странице заявок размещена таблица со всеми заявками на ремонт. Пользователь может нажать на кнопку изменить определенной заявки на ремонт, после чего откроется форма с ранее введенными данными. На форме также находится поле с причиной изменения. Также на странице размещена кнопка добавления заявки, при нажатии на которую открывается форма, примерный вид которой представлен на рисунке 1.31. На форме заявки кроме полей размещены кнопки добавления нескольких единиц техники.

Рисунок 1.31 – Форма заявки

Страница конструктора отчетов представлена на рисунке 1.32.

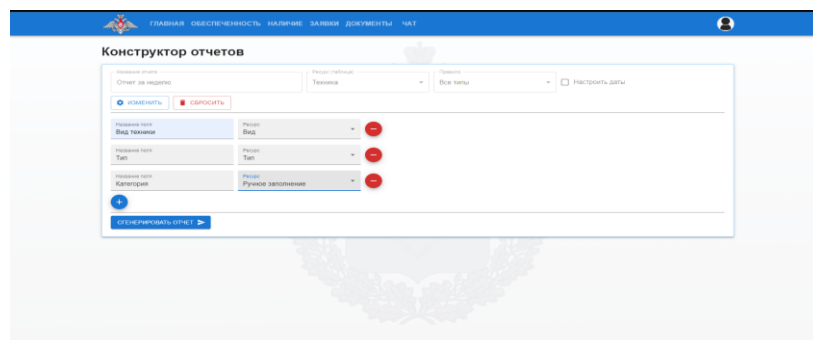


Рисунок 1.32 – Страница конструктора отчетов

Страница конструктора отчетов разделена на две части. Первая часть содержит:

- название отчета;
- ресурс(таблица);
- правило формирования (все типы, конкретный тип, конкретный тип и вид);
- выбор диапазона дат;
- кнопка подтвердить или изменить данные;
- кнопка сброса настроек.

Вторая часть содержит:

- пользовательские поля с выбором ресурса;
- кнопка добавления пользовательских полей;
- кнопка генерации отчета.

Страница документов представлена на рисунке 1.33.

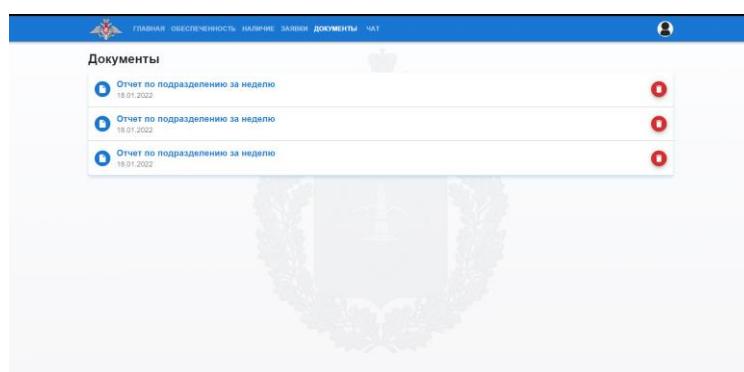


Рисунок 1.33 – Страница документов

Страница документов содержит список ранее сгенерированных отчетов, в котором каждая секция включает:

- название отчета;
- дату формирования отчета;
- кнопку загрузки документа;
- кнопку удаления документа;

Страница документов представлена на рисунке 1.34.

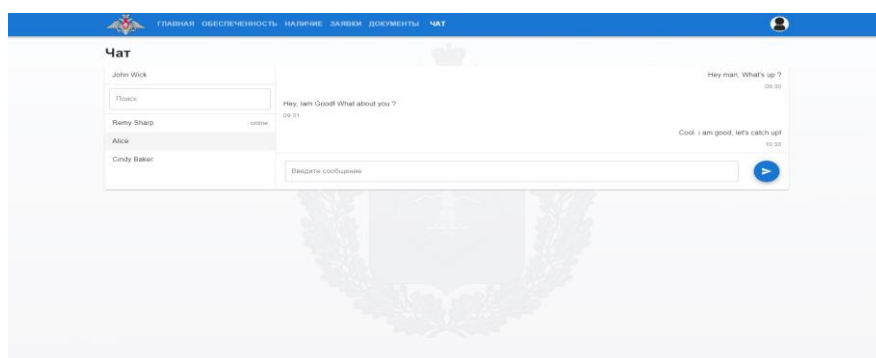


Рисунок 1.34 – Страница чата

Страница чата содержит список всех доступных администраторов для пользователя. После выбора пользователя отображается экран, который включает в себя всю переписку. В самом низу переписки расположено поле ввода сообщений и кнопка отправки.

Страница настройки данных представлена на рисунке 1.35.

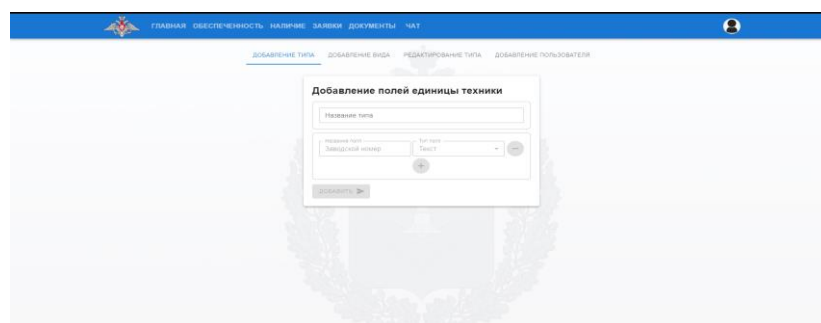


Рисунок 1.35 – Страница настройки данных

Страница настройки данных разделена на 4 части:

- добавление типов;

- добавление видов;
- редактирование типов;
- добавление пользователей.

Секция добавления типов включает в себя поля:

- название типа;
- заводской номер (по умолчанию);
- пользовательские поля с выбором типа заполнения;
- кнопку добавления.

Секция добавления видов включает в себя поля:

- список типов;
- название вида;
- кнопку добавления.

Секция редактирования типов включает в себя поля:

- список типов;
- список ранее добавленных пользователем полей с типами данных
- пользовательские поля с выбором типа заполнения;
- кнопку редактирования.

Секция добавления пользователей включает в себя поля:

- список пользователей;
- логин пользователя;
- пароль пользователя;
- кнопку добавления.

Страница справочной информации представлена на рисунке 1.36.

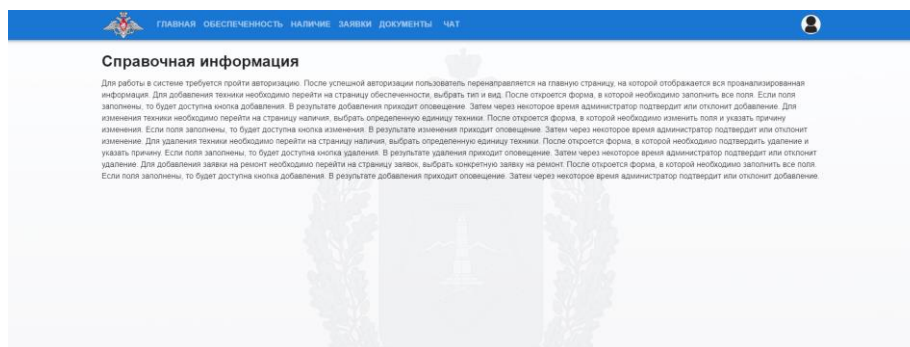


Рисунок 1.36 – Страница настройки данных

1.5.4 Используемые технические средства

Технические характеристики аппаратного обеспечения, на котором разрабатывалось и тестировалось программное обеспечение, представлены ниже:

- процессор *Intel* с тактовой частотой 2048 МГц;
- объем оперативной памяти – 1024 Мбайт;
- объем свободного пространства на жестком диске не менее 600 Мбайт;
- операционная система *Microsoft Windows 10*;
- прикладное программное обеспечение для просмотра страниц.

1.6 Программа и методика испытаний

1.6.1 Объект испытания

Объектом испытания является клиентская часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа написанная на языке программирования *JavaScript* в среде разработки *Microsoft Visual Code* с использованием библиотеки *React*.

1.6.2 Цель испытаний

Испытания проводятся с целью проверки правильного функционирования клиентской части, а также полноты удовлетворения требованиям технического задания.

1.6.3 Требования к клиентской части системы

Для проверки работоспособности клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа необходимо программное обеспечение в виде среды разработки *Microsoft Visual Code* и тестового сервера *Open Server*. Взаимодействие системы с пользователем осуществляется в интерактивном режиме, по средствам клиентского приложения с графическим интерфейсом. Клиентская часть системы функционирует под управлением ОС MCBC 12.1, MCBC 12.4, *Astra Linux SE*, *Windows XP* и выше.

1.6.4 Требования к программной документации

На испытание программной реализации клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа предоставляются следующие документы:

- техническое задание;
- описание проектирования приложения;
- описание программы;
- программу и методику испытаний;
- описание применения;
- текст программы.

1.6.5 Средства и порядок испытаний

Для проведения испытания клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа необходимо использовать технические средства с такими же характеристиками, на котором разрабатывалась данная подсистема. При тестировании использовался браузер *Google Chrome* в двух вкладках, одна из которых в режиме «инкогнито».

Порядок испытаний программной реализации клиентской части автоматизированной системы заключается в следующем:

- необходимо запустить тестовый сервер *Open Server*, на котором размещена клиентская часть и тестовая база данных;
- войти в систему под ролью администратора;
- добавить тип техники;
- добавить виды техники;
- войти в систему под ролью пользователя;
- под ролью пользователя добавить единицу техники;
- под ролью администратора подтвердить добавление единицы техники;
- под ролью пользователя изменить единицу техники;
- под ролью администратора подтвердить изменение единицы техники;
- под ролью пользователя добавить заявку на ремонт;
- под ролью администратора подтвердить добавление заявки на ремонт;
- под ролью пользователя изменить заявку на ремонт;
- под ролью администратора подтвердить изменение заявки на ремонт;
- под ролью пользователя сгенерировать отчет;
- выгрузить отчет;
- под ролью пользователя удалить заявку на ремонт;
- под ролью администратора подтвердить удаление заявки на ремонт;
- под ролью пользователя удалить единицу техники;
- под ролью администратора подтвердить удаление единицы техники;
- под ролью пользователя отправить сообщение администратору;
- под ролью администратора отправить сообщение пользователю;
- под ролью администратора изменить поля типа техники;
- под ролью администратора добавить нового пользователя;
- открыть страницу справочной информации.

Используя данный тест необходимо сделать общий вывод о корректности программной реализации клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа.

1.6.6 Методы испытания

Тестирование систем – важный этап производства ПО, направленный на детальное исследование программного кода и выявление ошибок в работе систем. Одна из главных целей тестирования – проверка соответствия работоспособности информационной системы в целом или ее отдельных модулей ожиданиям пользователя. Действия необходимые для тестирования клиентской части системы, а также ожидаемые и фактические результаты приведены в таблице 1.5. Результаты выполнения тестовых примеров приведены в приложении А.

Таблица 1.5 – Тестирование клиентской части

Название теста	Шаги теста	Ожидаемый результаты	Результат теста
Авторизация в системе	Запустить тестовый сервер. Открыть веб-приложение. Ввести данные для авторизации. Подтвердить ввод.	Отображение страницы авторизации. Если авторизация успешна, перенаправление на главную страницу в зависимости от роля пользователя.	Тест пройден. Результат представлен на рисунках А.1-А.3.
Добавление пользователя	Перейти на страницу настроек. Выбрать секцию «Добавление пользователя».	Отображение страницы настроек. Отображение формы	Тест пройден. Результат представлен на рисунках А.4 – А.5.

Название теста	Шаги теста	Ожидаемый результаты	Результат теста
	Ввести данные нового пользователя. Подтвердить ввод.	добавления пользователя. Отображение добавленного пользователя.	
Добавление типов техники	Перейти на страницу настроек. Выбрать секцию «Добавление типа». Ввести название типа, а также необходимые для заполнения поля Подтвердить ввод.	Отображение страницы настроек. Отображение формы добавления типа техники. Отображение добавленного типа техники на странице обеспеченности.	Тест пройден. Результат представлен на рисунке А.6 и на рисунке А.8.
Добавление видов техники	Перейти на страницу настроек. Выбрать секцию «Добавление вида». Ввести данные нового типа. Подтвердить ввод.	Отображение страницы настроек. Отображение формы добавления вида техники. Отображение добавленного вида на странице обеспеченности.	Тест пройден. Результат представлен на рисунках А.7 – А.8.
Добавление единицы техники	Перейти на страницу обеспеченности. Выбрать необходимый тип и вид техники. Ввести необходимые данные для заполнения. Подтвердить ввод.	Отображение страницы обеспеченности. Отображение формы добавления техники. Отображение увеличения счетчика техники на странице обеспеченности.	Тест пройден. Результат представлен на рисунках А.9. – А.10.

Название теста	Шаги теста	Ожидаемый результат	Результат теста
		Отображение добавленной техники на странице наличия.	
Подтверждение добавления техники администратором	Перейти на главную страницу. Выбрать секцию «Добавление техники». Выбрать добавленную пользователем технику. Проверить на правильность заполнения полей. Подтвердить добавление техники	Отображение главной страницы. Отображение уведомления о добавлении. Отображение формы для подтверждения. Отображение результата подтверждения на странице наличия пользователя.	Тест пройден. Результат представлен на рисунках А.11 – А.13.
Изменение единицы техники	Перейти на страницу наличия. Выбрать необходимую технику. Ввести измененные данные. Подтвердить ввод.	Отображение страницы наличия. Отображение формы изменения техники. Отображение измененной техники на странице наличия.	Тест пройден. Результат представлен на рисунках А.14 – А.15.
Подтверждение изменения техники администратором	Перейти на главную страницу. Выбрать секцию «Изменение техники». Выбрать измененную пользователем технику.	Отображение главной страницы. Отображение уведомления о изменении техники. Отображение формы для	Тест пройден. Результат представлен на рисунках А.16 – А.17.

Название теста	Шаги теста	Ожидаемый результаты	Результат теста
	Проверить на правильность заполнения полей. Подтвердить изменение техники	подтверждения. Отображение результата подтверждения на странице наличия пользователя.	
Добавление заявки на ремонт.	Перейти на страницу заявок. Выбрать «Добавить заявку». Ввести необходимые данные для заполнения. Подтвердить ввод.	Отображение страницы заявок. Отображение формы добавления заявки. Отображение добавленной заявки.	Тест пройден. Результат представлен на рисунках А.18 – А.20.
Подтверждение добавления заявки на ремонт администратором	Перейти на главную страницу. Выбрать секцию «Добавление заявок». Выбрать добавленную пользователем заявку. Проверить на правильность заполнения полей. Подтвердить добавление заявки.	Отображение главной страницы. Отображение уведомления о добавлении заявки. Отображение формы для подтверждения. Отображение результата подтверждения на странице заявок пользователя.	Тест пройден. Результат представлен на рисунках А.21 – А.23.
Изменение заявки на ремонт.	Перейти на страницу заявок. Выбрать кнопку изменения заявки Ввести измененные данные. Подтвердить ввод.	Отображение страницы заявок. Отображение формы изменения заявки. Отображение измененной заявки.	Тест пройден. Результат представлен на рисунках А.23 – А.24.

Название теста	Шаги теста	Ожидаемый результаты	Результат теста
Подтверждение изменения заявки на ремонт администратором	Перейти на главную страницу. Выбрать секцию «Изменение заявок». Выбрать измененную пользователем заявку. Проверить на правильность заполнения полей. Подтвердить изменение заявки.	Отображение главной страницы. Отображение уведомления о изменении заявки. Отображение формы для подтверждения. Отображение результата подтверждения на странице заявок пользователя.	Тест пройден. Результат представлен на рисунках А.25 – А.26.
Генерация отчета	Перейти на страницу конструктора отчетов. Заполнить данные для получения ресурса таблиц. Добавить необходимые пользователем поля. Нажать генерацию отчета. Перейти на страницу документов. Выбрать необходимый документ и нажать на загрузку.	Отображение страницы конструктора отчетов. Отображение полученных ресурсов таблиц. Отображение результата генерация отчета на странице документов. Отображение страницы документов. Выгруженный документ с сервера.	Тест пройден. Результат представлен на рисунках Б.27 – Б.29.

Название теста	Шаги теста	Ожидаемый результаты	Результат теста
Удаление заявки на ремонт	Перейти на страницу заявок. Выбрать кнопку удаления конкретной заявки Ввести причину удаления. Подтвердить удаление.	Отображение страницы заявок. Отображение формы удаления заявки. Удаления заявки из списка.	Тест пройден. Результат представлен на рисунке А.33.
Подтверждение удаления заявки на ремонт администратором	Перейти на главную страницу. Выбрать секцию «Удаление заявок». Выбрать удаленную пользователем заявку. Проверить на правильность заполнения полей. Подтвердить удаление заявки.	Отображение главной страницы. Отображение уведомления об удалении заявки. Отображение формы для подтверждения. Удаление заявки из списка на странице заявок пользователя.	Тест пройден. Результат представлен на рисунках А.34 – А.35.
Удаление единицы техники	Перейти на страницу наличия. Выбрать кнопку удаления конкретной техники. Ввести причину удаления. Подтвердить удаление.	Отображение страницы наличия. Отображение формы удаления техники. Удаления техники из списка.	Тест пройден. Результат представлен на рисунке А.30.
Подтверждение удаления единицы техники администратором	Перейти на главную страницу. Выбрать секцию «Удаление техники». Выбрать удаленную пользователем единицу техники..	Отображение главной страницы. Отображение уведомления об удалении техники. Отображение формы для	Тест пройден. Результат представлен на рисунках А.31 – А.32.

Название теста	Шаги теста	Ожидаемый результат	Результат теста
	Проверить на правильность заполнения полей. Подтвердить удаление заявки	подтверждения удаления техники из списка на странице наличия пользователя.	
Редактирование полей типов техники.	Перейти на страницу настроек. Выбрать секцию «Редактирование типа». Выбрать необходимый тип. Добавить новые или изменит существующие поля. Подтвердить ввод.	Отображение страницы настроек. Отображение формы изменения типа техники. Отображение измененного типа техники на странице обеспеченности.	Тест пройден. Результат представлен на рисунке А.39.
Удаление вида техники.	Перейти на страницу настроек. Выбрать секцию «Удаление данных». Выбрать необходимый для удаления вид. Подтвердить ввод.	Отображение страницы настроек. Отображение формы удаления вида техники. Удаление из списка на странице обеспеченности.	Тест пройден. Результат представлен на рисунке А.41.
Удаление типа техники.	Перейти на страницу настроек. Выбрать секцию «Удаление данных». Выбрать необходимый для удаления тип. Подтвердить ввод.	Отображение страницы настроек. Отображение формы удаления типа техники. Удаление из списка на странице обеспеченности.	Тест пройден. Результат представлен на рисунке А.40.

Название теста	Шаги теста	Ожидаемый результат	Результат теста
Обмен сообщениями с администратором	Войти под ролью пользователя. Перейти на страницу чата. Выбрать администратора. Заполнить поле сообщения. Подтвердить ввод.	Отображение страницы чата. Отображение списка все администраторов и содержание их диалогов. Отображение сообщения на странице чата у администратора.	Тест пройден. Результат представлен на рисунке А.36.
Обмен сообщениями с пользователем	Войти под ролью администратора. Перейти на страницу чата. Выбрать пользователя. Заполнить поле сообщения. Подтвердить ввод.	Отображение страницы чата. Отображение списка всех пользователей и содержание их диалогов. Отображение сообщения на странице чата у пользователя.	Тест пройден. Результат представлен на рисунке А.37.

Все пункты испытаний клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа выполнены в полном объеме. Следовательно, можно сделать вывод о том, что клиентская часть системы работает верно.

1.7 Описание применения

1.7.1 Руководство пользователя

Для работы в системе требуется пройти авторизацию (рисунок А.1).

После успешной авторизации пользователь перенаправляется на главную страницу, на которой отображается вся проанализированная информация (рисунок А.2).

Для добавления техники необходимо перейти на страницу обеспеченности (рисунок А.8), выбрать тип и вид. После откроется форма, в которой необходимо заполнить все поля. Если поля заполнены, то будет доступна кнопка добавления (рисунок А.9). В результате добавления приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит добавление.

Для изменения техники необходимо перейти на страницу наличия (рисунок А.13), выбрать определенную единицу техники. После откроется форма, в которой необходимо изменить поля и указать причину изменения (рисунок А.14). Если поля заполнены, то будет доступна кнопка изменения. В результате изменения приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит изменение.

Для удаления техники необходимо перейти на страницу наличия (рисунок А.13), выбрать определенную единицу техники. После откроется форма, в которой необходимо подтвердить удаление и указать причину (рисунок А.30). Если поля заполнены, то будет доступна кнопка удаления. В результате удаления приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит удаление.

Для добавления заявки на ремонт необходимо перейти на страницу заявок (рисунок А.18), выбрать конкретную заявку на ремонт. После откроется форма, в которой необходимо заполнить все поля (рисунок А.19). Если поля заполнены, то будет доступна кнопка добавления. В результате добавления приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит добавление.

Для изменения заявки на ремонт необходимо перейти на страницу заявок (рисунок А.23), выбрать конкретную заявку на ремонт. После откроется форма, в которой необходимо изменить поля и указать причину изменения (рисунок А.24). Если поля заполнены, то будет доступна кнопка изменения. В

результате изменения приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит изменение.

Для удаления заявки на ремонт необходимо перейти на страницу заявок (рисунок А.23), выбрать конкретную заявку на ремонт. После откроется форма, в которой необходимо подтвердить удаление и указать причину (рисунок А.33). Если поля заполнены, то будет доступна кнопка удаления. В результате удаления приходит оповещение. Затем через некоторое время администратор подтвердит или отклонит удаление.

Для формирования документа необходимо перейти на страницу конструктора отчетов, ввести все параметры и нажать кнопку генерации (рисунок А.27). Далее перейти на страницу документов и выбрать документ на скачивание (рисунок А.28). Если необходимо удалить документ, то в его секции находится кнопка удаления.

Для отправки сообщения администратору необходимо перейти на страницу чата. Далее выбрать администратора и отправить введенной сообщение (рисунок А.36). Кнопка отправки не будет доступна, пока сообщение не введено.

Для просмотра справочной информации необходимо перейти на страницу справочной информации (рисунок А.38).

Для выхода из системы необходимо выбрать вкладку меню выйти. Далее пользователь возвращается на страницу авторизации (рисунок А.1).

1.7.2 Руководство администратора

Для работы в системе требуется пройти авторизацию (рисунок А1).

После успешной авторизации администратор перенаправляется на главную страницу (рисунок А.3), на которой отображаются все уведомления.

Для подтверждения добавления техники необходимо выбрать секцию «Добавление техники». Затем выбрать конкретную единицу техники. После откроется форма, в которой указана вся информация (рисунок А.12). Если информация актуальна, необходимо нажать на кнопку подтверждения, иначе

указать причину отказа и нажать на кнопку отклонения. В результате добавления приходит оповещение.

Для подтверждения изменения техники необходимо выбрать секцию «Изменения техники». Затем выбрать конкретную единицу техники. После откроется форма, в которой указана вся информация (рисунок А.17). Если информация актуальна для изменения, необходимо нажать на кнопку подтверждения, иначе указать причину отказа и нажать на кнопку отклонения. В результате изменения приходит оповещение.

Для подтверждения удаления техники необходимо выбрать секцию «Удаление техники». Затем выбрать конкретную единицу техники. После откроется форма, в которой указана вся информация (рисунок А.32). Если информация актуальна для удаления, необходимо нажать на кнопку подтверждения, иначе указать причину отказа и нажать на кнопку отклонения. В результате удаления приходит оповещение.

Для подтверждения добавления заявок на ремонт необходимо выбрать секцию «Добавление заявки». Затем выбрать конкретную заявку на ремонт. После откроется форма, в которой указана вся информация (рисунок А.22). Если информация актуальна, необходимо нажать на кнопку подтверждения, иначе указать причину отказа и нажать на кнопку отклонения. В результате добавления приходит оповещение.

Для подтверждения изменения заявок на ремонт необходимо выбрать секцию «Изменение заявки». Затем выбрать конкретную заявку на ремонт. После откроется форма, в которой указана вся информация (рисунок А.26). Если информация актуальна для изменения, необходимо нажать на кнопку подтверждения, иначе указать причину отказа и нажать на кнопку отклонения. В результате изменения приходит оповещение.

Для подтверждения удаления заявок на ремонт необходимо выбрать секцию «Удаление заявки». Затем выбрать конкретную заявку на ремонт. После откроется форма, в которой указана вся информация (рисунок А.35).

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

Если информация актуальна для удаления, необходимо нажать на кнопку подтверждения, иначе указать причину отказа и нажать на кнопку отклонения. В результате удаления приходит оповещение.

Для добавления нового типа техники необходимо перейти на страницу настройки данных, выбрать секцию «Добавление типа» и ввести всю необходимую информацию (рисунок А.6). Если поля заполнены, то будет доступна кнопка добавления. В результате добавления приходит оповещение.

Для добавления нового вида техники необходимо перейти на страницу настройки данных, выбрать секцию «Добавление вида» и ввести всю необходимую информацию (рисунок А.7). Если поля заполнены, то будет доступна кнопка добавления. В результате добавления приходит оповещение.

Для изменения полей типа техники необходимо перейти на страницу настройки данных, выбрать секцию «Редактирование типа» и ввести всю необходимую информацию (рисунок А.39). Если поля заполнены, то будет доступна кнопка добавления. В результате добавления приходит оповещение.

Для добавления пользователей необходимо перейти на страницу настройки данных, выбрать секцию «Добавление пользователей» и ввести всю необходимую информацию (рисунок А.5). Если поля заполнены, то будет доступна кнопка добавления. В результате добавления приходит оповещение.

Для просмотра справочной информации необходимо перейти на страницу справочной информации (рисунок А.38).

Для выхода из системы необходимо выбрать вкладку меню выйти. Далее пользователь возвращается на страницу авторизации (рисунок А1).

2 Технико-экономическое обоснование разработки программы

2.1 Организация производства при разработке клиентской части АС

Для разработки программного продукта будут приобретены оборудование (вычислительная техника), лицензионное программное обеспечение и расходные материалы (дополнительные материалы). Данные по потребности в оборудовании и лицензионном программном обеспечении представлены соответственно в таблице 2.1 и 2.2

Таблица 2.1 – Потребность в оборудовании

Оборудование	Потребляемая мощность, Вт	Источник	Количество	Ориентировочная цена за единицу, рублей
Системный блок ASUS S300MA-310100082T, Intel Core i3 10100 3.6 ГГц, DDR4, Intel UHD Graphics 630	65	Магазин «Мвидео»	6	35 000
Клавиатура проводная Logitech K120 for Business (920-002522)	1,5	Магазин «Мвидео»	6	800
Монитор LG 24MK600M-B	17,5	Магазин «Мвидео»	6	12 500
Мышь проводная Logitech M90	1,25	Магазин «Мвидео»	6	450

Оборудование	Потребляемая мощность, Вт	Источник	Количество	Ориентировочная цена за единицу, рублей
Лазерный принтер <i>Pantum P2207</i>	370	Магазин «Мвидео»	1	6 000

Таблица 2.2 – Потребность в лицензионном программном обеспечении

Лицензионное программное обеспечение	Источник	Срок действия лицензии, дней	Цена, рублей
<i>Microsoft Windows 10</i>	https://www.microsoft.com/ru-ru/software-download/windows10	1 год	28 000
<i>Visual Studio Code</i>	https://code.visualstudio.com/	Бессрочная	-
<i>Open Server</i>	https://ospanel.io/	Бессрочная	-

Перечень дополнительных материалов, источники их приобретения, объем, и ориентировочные цены приведены в таблице 2.3.

Таблица 2.3 – Потребность в дополнительных материалах

Название	Источник	Количество	Ориентировочная цена за единицу, рублей
Рабочий стол	Магазин «Юнитекс»	6	10 000
Кресла	Магазин «Юнитекс»	6	8 000
Настольная лампа	Магазин «Юнитекс»	6	1 000
Канцтовары	Супермаркет «Лист»	12	200
Бумага	Супермаркет «Лист»	4	350

Разработка программного обеспечения должна осуществляться командой разработчиков, состав которой представлен в таблице 2.4.

Таблица 2.4 – Состав проектной команды

Специалист	Количество	Описание занятия
Руководитель проекта	1	Руководство и управление проектом
		Формирование команды исполнителей
		Управление ресурсами проекта
		Контроль за соблюдением требований заказчика
		Приемка и внедрение системы
Архитектор системы	1	Проектирование модели автоматизированной системы
Фронтенд программист	1	Написание и тестирование кода клиентской части
		Разработка программной документации
Бэкенд программист		Написание и тестирование кода серверной части
		Разработка программной документации
Тестировщик	1	Контроль выполнения тестов
		Подготовка спецификации на окружение
Веб-дизайнер	1	Разработка интерфейса
		Проектирование и планирование структуры сайта.
		Разработка навигации.

Данные по основной заработной плате разработчиков и их количественному составу приведены в таблице 2.5.

Таблица 2.5 – Состав проектной команды

Специалист	Количество	Сумма заработной платы, рублей
Руководитель проекта	1	80 000
Архитектор системы	1	30 000
Фронтенд программист	1	50 000
Бэкенд программист	1	60 000
Тестировщик	1	20 000
Веб-дизайнер	1	30 000

2.2 Календарное планирование разработки программного обеспечения

Календарное планирование представляет собой совокупность связанных между собой взаимными отношениями процедур, является основой для разработки бюджета проекта и осуществления контроля за реализацией проекта, который заключается в сравнении плановых и фактических показателей.

Основная цель календарного планирования состоит в построении модели реализации проекта, которая необходима для определения порядка выполнения работ по проекту и координации деятельности его участников.

На этапе календарного планирования определяются все необходимые параметры реализации проекта: наименования, порядок и продолжительность этапов выполнения проекта, потребность в трудовых ресурсах, сроки и объемы привлечения сторонних организаций и т.п.

Результатом календарного планирования является календарный план разработки, представленный в виде диаграммы Ганта. Деятельность по разработке календарных планов охватывает все этапы реализации проекта, начиная с разработки концепции проекта и заканчивая его внедрением в производство.

Построение диаграммы Ганта было осуществлено с использованием программного продукта MS Excel в соответствии с разработанным календарным планом, представленным на рисунке 2.1, где значение «Итого» позволяет сделать вывод о действительной реализации проекта.

Построенная диаграмма Ганта представлена на рисунке 2.2

Название этапа	Специалист	Даты работ		Длительность работы	Длительность этапа
		Начало работы	Окончание работы		
Анализ предметной области	Руководитель проекта	07.02.2022	13.02.2022	6	6
	Архитектор системы	07.02.2022	13.02.2022	6	
Экономическое обоснование разработки программного продукта	Руководитель проекта	14.02.2022	19.02.2022	5	5
	Архитектор системы	14.02.2022	18.02.2022	4	
Разработка требований к создаваемой системе	Руководитель проекта	20.02.2022	25.02.2022	5	5
	Архитектор системы	20.02.2022	25.02.2022	5	
Проектирование приложения	Руководитель проекта	26.02.2022	13.03.2022	10	10
	Архитектор системы	26.02.2022	08.03.2022	8	
	Фронтенд программист	26.02.2022	13.03.2022	10	
	Бэкенд программист	26.02.2022	13.03.2022	10	
	Веб-дизайнер	26.02.2022	08.03.2022	8	
Написание программного кода системы	Руководитель проекта	14.03.2022	03.04.2022	15	15
	Фронтенд программист	14.03.2022	03.04.2022	15	
	Бэкенд программист	14.03.2022	03.04.2022	15	
Промежуточное тестирование	Руководитель проекта	04.04.2022	11.04.2022	6	6
	Тестирующий	04.04.2022	11.04.2022	6	
Отладка программного кода системы	Руководитель проекта	12.04.2022	24.04.2022	9	9
	Фронтенд программист	12.04.2022	24.04.2022	9	
	Бэкенд программист	12.04.2022	24.04.2022	9	
Тестирование системы	Руководитель проекта	25.04.2022	10.05.2022	12	12
	Фронтенд программист	25.04.2022	10.05.2022	12	
	Бэкенд программист	25.04.2022	10.05.2022	12	
	Тестирующий	25.04.2022	10.05.2022	12	
Разработка документации	Руководитель проекта	11.05.2022	21.05.2022	8	8
	Фронтенд программист	11.05.2022	21.05.2022	8	
	Бэкенд программист	11.05.2022	21.05.2022	8	
Приемка и внедрение системы	Руководитель проекта	22.05.2022	01.06.2022	8	8
	Фронтенд программист	22.05.2022	01.06.2022	8	
	Бэкенд программист	22.05.2022	01.06.2022	8	
Итого:					84

Рисунок 2.1 – Календарный план

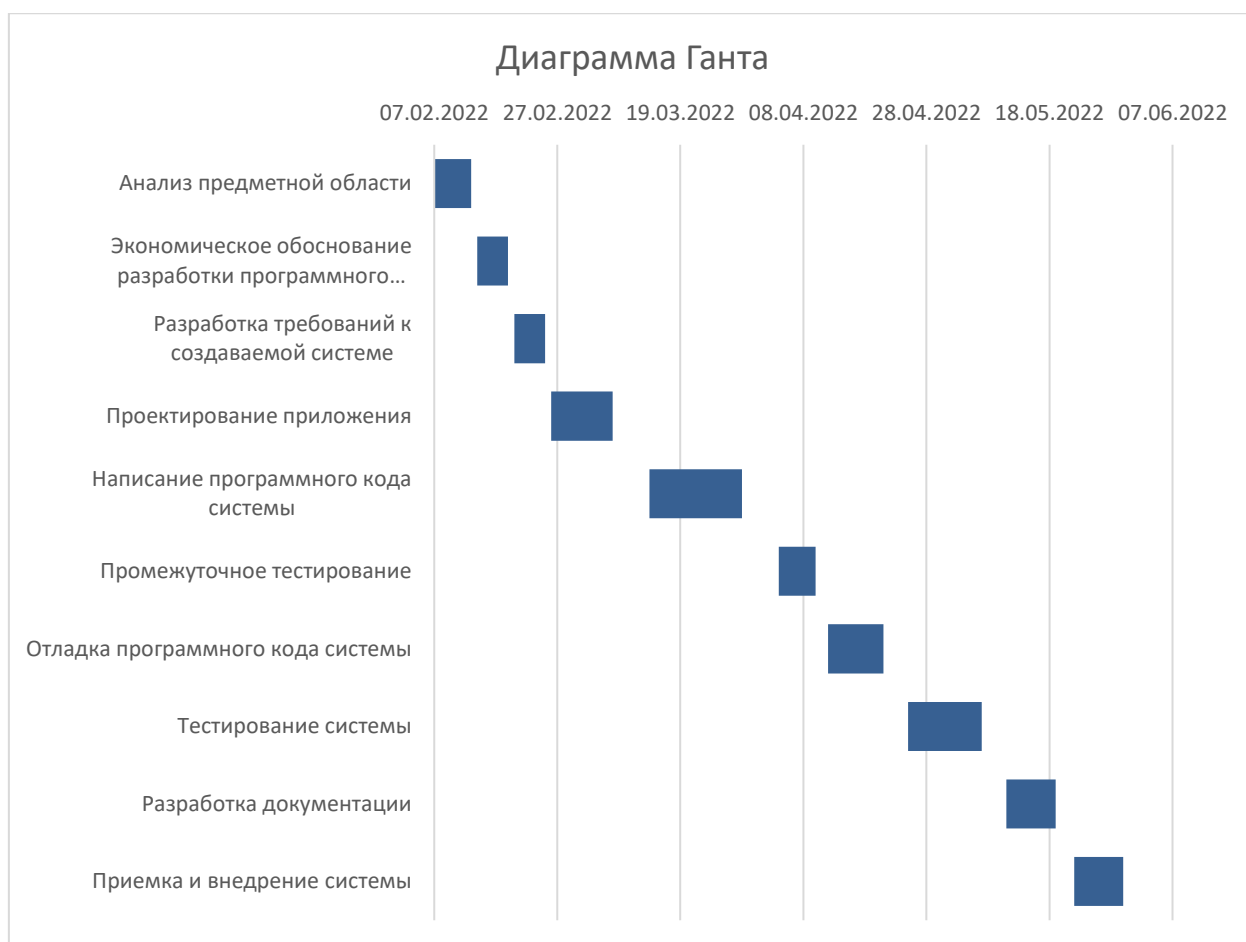


Рисунок 2.2 –Диаграмма Ганта

В соответствии с диаграммой Ганта длина критического пути разработки программы составила 84 рабочих дней, что соответствует минимальному количеству времени, необходимого для разработки автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа. С учетом выходных дней при 5 - дневной рабочей недели (21 день в месяц) по 8 часов в сутки длительность разработки составит 74 дня. Начало разработки 07.02.2022, окончание – 01.06.2022.

2.3 Расчет себестоимости программного обеспечения

Себестоимость разработки программного продукта включает следующие статьи затрат:

- затраты на приобретение оборудования;
- затраты на дополнительные материалы;
- затраты на приобретение лицензионного программного обеспечения;
- затраты на заработную плату;
- затраты на обслуживание.

Затраты на приобретение оборудования составят 229 300 рублей и представлены в таблице 2.1.

Затраты на программное обеспечение в период разработки составят 28 000 и представлены в таблице 2.2.

Затраты на дополнительные материалы в период разработки составят 117 800 рублей и представлены в таблице 2.3.

Затраты на заработную плату включают затраты на основную заработную плату разработчиков, а также отчисления на социальные нужды. Затраты на основную заработную плату разработчиков определяются исходя из установленной заработной платы, в соответствии со штатным расписанием, количества специалистов и их занятости в процессе разработки программного продукта (продолжительность работы в днях). С учетом выходных дней при 5-дневной рабочей неделе количество рабочих дней в месяце в среднем составляет 21 день. Затраты на основную заработную плату представлены в таблице 2.6.

Таблица 2.6 – Состав проектной команды

Специалист	Период занятости, дни	Сумма заработной платы, рублей	Заработная плата за период, рублей
Руководитель проекта	84	80 000	320 000
Архитектор системы	23	30 000	32 800
Фронтенд программист	62	50 000	147 600
Бэкенд программист	62	60 000	177 100
Тестировщик	18	20 000	17 100
Веб-дизайнер	8	50 000	19 100

Таким образом, затраты на основную заработную плату составят 713 700 рублей.

Отчисления на социальные нужды - это обязательные расходы каждого частного предпринимателя и организации. Отчисления на социальные нужды включают в себя взносы на все предусмотренные законодательством обязательные виды страхования и исчисляются в процентном соотношении от затрат на оплату труда работников.

Отчисления на социальные нужды производятся согласно Федеральному закону № 117-ФЗ (ред. от 30.04.2021) (с изменениями и дополнениями, вступившими в силу с 30.04.2021) и составляют 30% от затрат на заработную плату исполнителей.

Законом установлены следующие тарифы страховых взносов:

- пенсионный фонд Российской Федерации - 22 %;
- фонд социального страхования Российской Федерации - 2,9 %;
- федеральный фонд обязательного медицинского страхования – 5,1 %.

Данные по суммам отчислений на социальные нужды приведены соответственно в таблице 2.7.

Таблица 2.7 – Отчисления на социальные нужды

Специалист	Заработная плата за период, рублей	Пенсионный фонд РФ, рублей	Федеральный фонд обязательного медицинского страхования, рублей	Фонд социального страхования РФ, рублей	Общая сумма отчислений
Руководитель проекта	320 000	70 400	16 300	9 300	96 000
Архитектор системы	32 800	7 200	1 700	1 000	9 900
Фронтенд программист	147 600	32 400	7 500	4 300	44 200
Бэкенд программист	177 100	38 900	9 000	5 100	14 100
Тестировщик	17 100	3 700	900	500	5 100
Веб-дизайнер	19 100	4 200	1 000	600	5 800

Таким образом, общая сумма отчислений на социальные нужды разработчиков составит 175 100 рублей, а затраты на заработную плату разработчиков составят 713 700 рублей.

Затраты на обслуживание включают следующие статьи:

- амортизация оборудования, составляющая 20% от стоимости оборудования;
- обслуживание оборудования, составляющее 15% от амортизационных отчислений;
- затраты на электроэнергию, включающие затраты на электроэнергию персональных компьютеров, оргтехники и бытовых приборов;
- аренда офиса и расходы на его содержание;
- накладные расходы.

Амортизация оборудования составляет 20% от стоимости оборудования. Было приобретено 6 системных блоков по 35 000 рублей, 6 клавиатур по 800 рублей, 6 мониторов по 12 500 рублей, 6 компьютерных мышей по 450 рублей и 1 принтер стоимостью 6 000 рублей. Таким образом затраты на амортизацию оборудования в месяц составят:

$$((6 * 35\,000 + 6 * 800 + 6 * 12\,500 + 6 * 450 + 7\,000) * 0,2) / 12 =$$

$$= 4\,887 \text{ руб.}$$

Обслуживание оборудования составляет 15% от амортизационных отчислений. Таким образом затраты на обслуживание оборудования составят:

$$4\,887 * 0,15 = 733 \text{ руб.}$$

Предположительно в день по 8 часов будут работать 6 системных блоков, 6 мониторов, 6 клавиатур, 6 компьютерных мышей, 27 дней по 2 часа в сутки – принтер. Итого:

$$(65 * 8 * 84 * 6) + (1,5 * 8 * 784 * 6) + (17,5 * 8 * 84 * 6) + (1,25 * 8 * 84 * 6) + (370 * 2 * 27) = 414,11 \text{ кВт}$$

Таким образом, за 74 дня разработки программного обеспечения будет использовано 414,11 кВт электроэнергии на системные блоки, мониторы, клавиатуры, компьютерные мыши и принтер, а также 0,45 кВт на прочие приборы. Итог

$$414,11 + 0,45 = 414,56 \text{ кВт}$$

Накладные расходы составляют 60% от заработной платы. Таким образом затраты на накладные расходы составят:

$$713\,700 * 0,6 = 428\,220$$

Результаты расчётов на обслуживание приведены в таблице 2.8.

Таблица 2.8 – Затраты на обслуживание

Статья затрат	Показатель отчислений в месяц	Сумма отчислений в месяц, рублей	Итоговая сумма отчислений за период разработки, рублей
Амортизация оборудования	20,00%	4 887	19 548
Обслуживание оборудования	15,00%	733	2 932

Статья затрат	Показатель отчислений в месяц	Сумма отчислений в месяц, рублей	Итоговая сумма отчислений за период разработки, рублей
Затраты на электроэнергию	3,51 за 1 кВт/ч	1 455	4 526
Затраты на содержание помещения	-	25 000	100 000
Накладные расходы	60,00%	107 055	428 220
Итого:			555 226

Себестоимость разработки определяется как сумма затрат, включающая все статьи затрат на реализацию проекта, представленные в таблице 2.9.

Таблица 2.9 – Затраты на разработку проекта

Статья затрат	Сумма отчислений, руб.
Приобретение оборудования	229 300
Приобретение лицензионного программного обеспечения	28 000
Приобретение дополнительных материалов	117 800
Затраты на заработную плату разработчиков	713 700
Отчисления на социальные нужды	175 100
Затраты на обслуживание	555 226
Итого:	1 819 126

Таким образом, сумма затрат на разработку проекта составит 1 819 126 рублей и будет соответствовать себестоимости автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа.

2.4 Расчет экономической эффективности

Для расчета экономической эффективности от внедрения автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа необходимо проанализировать

существующие на рынке системы с аналогичным функционалом, основными из которых являются «1С Предприятие» и «Парус - Предприятие». Количество пользователей системы: 60-70.

Стоимость клиентской лицензии «1С Предприятие» на 100 рабочих мест будет составлять 360 000 руб. Стоимость внедрения системы представлена на рисунке 2.3.

№	Показатель	Проекты 5-49 АРМ (ср. 16 АРМ)		Проекты от 50 АРМ (ср. 127 АРМ)	
		Москва и СПб	Регионы РФ	Москва и СПб	Регионы РФ
1	Трудозатраты на 1 АРМ	29 чел./час		39 чел./час	
2	Ставка по работам (чел./час), руб.	2 130	1 440	2 200	1 460
3	Расчетная стоимость 1 АРМ (ср. трудозатраты * ср. ставку), руб.	61 770	41 760	85 800	56 940

Рисунок 2.3 – Трудозатраты на внедрение «1С Предприятие»

Таким образом, ориентировочная стоимость внедрения «1С Предприятие» - 4 350 000 руб.

Стоимость лицензии «Парус - Предприятие» по требуемым параметрам (Модуля без ограничения количества лицензий) с учетом внедрения будет составлять 2 015 000 руб.

Исходя из вышеперечисленных данных, можно сделать вывод, что разработка автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа экономически целесообразна.

3 Безопасность жизнедеятельности

3.1 Введение

Безопасность жизнедеятельности - это создание таких условий существования человека, общества и государства, которые обеспечивают состояние защищенности и отсутствия недопустимого риска нанесения вреда. В процессе трудовой деятельности на человеческий организм могут воздействовать опасные и вредные производственные факторы. Рабочее место программиста представляет собой совокупность факторов окружающей среды, в том числе и вредных, которыми являются:

- повышенный уровень шума, источниками которого служат вентиляционные устройства ЭВМ, устройства ввода-вывода, агрегаты кондиционирования и вентилирования воздуха, другие электрические приборы;

- высокие уровни электростатического и электромагнитного излучения, источниками которого являются видеотерминалы.

Кроме того, очень важен благоприятный микроклимат, чистота воздуха, освещенность в помещении, влажность, наличие пыли и др.

Параметры при работе с дисплейными терминалами и персональными компьютерами регламентирует СанПиН 2.2.2/2.4. 1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»; общие рекомендации по электробезопасности, а также особенности воздействия других факторов даны в «Типовой инструкции по охране труда при работе на персональном компьютере» (ТОИ Р-45-084-01).

3.2 Требования к параметрам воздушной среды

Микроклиматические условия на рабочих местах в помещениях с вычислительной техникой должны соответствовать требованиям, указанным в таблице 3.1.

Таблица 3.1 – Микроклимат помещений

Период года	Температура воздуха, °С	Скорость движения воздуха, м/с	Относительная влажность воздуха, %
Теплый	23-25	0,1-0,2	40-60
Холодный	22-24	до 0,1	40-60

На рабочем месте пользователя должны обеспечиваться оптимальные параметры микроклимата, которые установлены в СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания». На работах, производимых сидя и не требующих физического напряжения, температура воздуха должна быть в холодный период года от 22 до 24 °С, теплый период года - от 23 до 25 °С. Относительная влажность воздуха на постоянных рабочих местах должна составлять 40-60%, скорость движения воздуха должна быть 0,1 м/с. Для повышения влажности воздуха в помещениях следует применять увлажнители воздуха [6].

3.3 Требования к освещению рабочих мест

Согласно СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания». освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна

быть более 300 лк. Более подробная характеристика освещения в рабочих помещениях представлена в таблице 3.2.

Таблица 3.2 – Характеристика освещения в рабочих помещениях

Показатель	Нормируемое значение
Коэффициент естественной освещенности (КЕО) в зонах с устойчивым снежным покровом	Не ниже 1,2%
Коэффициент естественной освещенности (КЕО) на остальной территории	Не ниже 1,5%
Освещенность в зоне расположения рабочего документа для пользователей персональных компьютеров	300 – 500 ЛК
Наименьшая искусственная освещенность рабочих поверхностей в зоне обслуживания копировально-множительной техники	300 лк
Освещенность экрана монитора при местном освещении	Не более 300 лк
Яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения	Не более 200 кд/ кв.м.
Яркость бликов на экране монитора	Не более 40 кд/кв.м
Яркость потолка, при применении системы отраженного освещения	Не более 200 кд/кв.м.
Показатель ослепленности для источников общего искусственного освещения в производственных помещениях, предназначенных для размещения рабочих мест пользователей персональных компьютеров	Не более 20
Показатель ослепленности на рабочих местах пользователей копировально-множительной техники	Не более 40
Показатель дискомфорта в административно-общественных помещениях	Не более 40
Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях	Не более 200 кд/кв.м

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ при этом соотношение яркости между рабочими

поверхностями не должно превышать 3:1-5:1, а между рабочими поверхностями и поверхностями стен и оборудования - 10:1.

Коэффициент пульсации не должен превышать 5%.

Для исключения бликов отражений в экране светильников общего освещения рабочий стол с ПК следует размещать между рядами светильников. При этом светильники должны быть расположены параллельно горизонтальной линии взгляда работающего. Для уменьшения бликов рекомендуется использовать приэкранный защитный фильтр для видеомониторов/6/.

При рядом размещении рабочих столов не допускается расположение экранов дисплеев навстречу друг другу из-за их взаимного отражения, в противном случае между столами следует устанавливать перегородки.

3.4 Требования к уровням шума и вибрации на рабочих местах

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки».

Печатающее оборудование, являющееся источником шума, следует устанавливать на звукопоглощающей поверхности автономного рабочего места пользователя. Если уровни шума от печатающего оборудования превышают нормируемые, оно должно быть расположено вне помещения с ПК. Помещения для выполнения основной работы с ПК не должны быть расположены рядом (смежно) с производственными помещениями с повышенным уровнем шума (мастерские, производственные цеха и т. п.).

При выполнении основной работы на мониторах и ПЭВМ (диспетчерские, операторские, залы вычислительной техники и т. д.), где

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		88

работают инженерно-технические работники, уровень шума не должен превышать 60 дБА, в помещениях операторов ЭВМ (без дисплеев) - 65 дБА, на рабочих местах в помещениях, где размещаются шумные агрегаты вычислительных машин - 75 дБА.

Поскольку каждый ПК оснащен системами охлаждения, которые, в свою очередь, содержат источники вибрации; то ПК можно отнести к источнику общей вибрации 3 категории (электрические машины). При выполнении работ с использованием ПК в производственных помещениях уровень вибрации не должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип "в") в соответствии с действующими санитарно-эпидемиологическими нормативами, указанными в СН 2.2.4/2.1.8.566-96 «Производственная вибрация, вибрация в помещениях жилых и общественных зданий».

3.5 Расчет освещенности рабочего места

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения. Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

- по спектральному составу света они близки к дневному, естественному свету;
- обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);
- обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);
- более длительный срок службы [7].

Для расчета используются следующие исходные данные: комната имеет площадь 40 квадратных метров, длину - 8 метров, ширину - 5 метров, высоту - 3 м. Воспользуемся методом светового потока. Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = (E * K * S * Z) / n, \quad (3.1)$$

где F - рассчитываемый световой поток, Лм;

E - нормированная минимальная освещенность, Лк (согласно нормативам освещенности офисных и жилых объектов по СНиП, для офисного помещения с компьютерами минимальная освещенность равна 300Лк);

S - площадь освещаемого помещения (в нашем случае $S = 20\text{м}^2$);

Z - отношение средней освещенности к минимальной (обычно принимается равным 1,1...1,2, пусть $Z = 1,1$);

K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае $K = 1,5$);

n - коэффициент использования светового потока, (отражается в отношении светового потока к совокупному потоку всех ламп и обуславливается характеристиками светильника, размерами помещения и коэффициентами отражения от стен (РС) и потолка (РП)).

В помещениях, где находится компьютер, величины коэффициента отражения должны быть в следующих пределах: для потолка: 60...70%, для стен: 40...50%, для пола: около 30%. Для других поверхностей и рабочей мебели: 30...40%.

Рассчитаем освещенность исходя из следующих значений коэффициентов: $R_C = 40\%$, $R_{П} = 65\%$. Для этого сначала вычислим индекс

помещения I, определяемый геометрическую характеристику помещения по формуле:

$$I = S / (h * (A + B)), \quad (3.2)$$

где S - площадь помещения, $S = 40 \text{ м}^2$;

h - расчетная высота подвеса, $h = 3,2 \text{ м}$;

A - ширина помещения, $A = 5 \text{ м}$;

B - длина помещения, $B = 8 \text{ м}$.

Подставив значения получаем:

$$I = 40 / (3,2 * (5 + 8)) = 0,96$$

Зная индекс помещения I , по таблице расчета использования светового потока, приведенной на рисунке 3.1 находим коэффициент $n = 0,46$.

Светильник	«Глубоко-излучатель»			«Универсаль» без затенителя			«Люцетта»			ВЗГ 200 с отражателем			ОД			ПВЛ		
$\rho_{\text{п}}, \%$	30	50	70	30	50	70	30	50	70	30	50	70	30	50	70	30	50	70
$\rho_{\text{с}}, \%$	10	30	50	10	30	50	10	30	50	10	30	50	10	30	50	10	30	50
i	Коэффициент использования $\eta \cdot 100$																	
0,5	19	21	25	21	24	28	14	16	22	12	14	17	23	26	31	14	16	19
0,6	24	27	31	27	30	34	19	21	27	16	18	21	30	33	37	18	20	22
0,7	29	31	34	32	35	38	23	24	30	19	21	24	35	38	42	21	23	25
0,8	32	34	37	35	38	41	25	26	33	21	24	26	39	41	45	23	25	27
0,9	34	36	39	38	40	44	27	29	35	23	25	28	42	44	48	25	27	29
1,0	36	38	40	40	42	45	29	31	37	25	27	29	44	46	49	26	28	30
1,1	37	39	41	42	44	46	30	32	38	26	27	30	46	48	51	27	29	31
1,25	39	41	43	44	46	48	31	34	41	27	29	31	48	50	53	29	30	32
1,5	41	43	46	46	48	51	34	37	44	29	30	39	50	52	56	30	31	34
1,75	43	44	48	48	50	53	36	39	46	30	32	34	52	55	58	31	33	35
2,0	44	46	49	50	52	55	38	41	48	32	33	35	55	57	60	33	34	36
2,25	46	48	51	52	54	56	40	43	50	33	35	37	57	59	62	34	35	37
3,0	49	51	53	55	57	60	44	47	54	35	37	39	60	62	66	36	37	40
3,5	50	52	54	56	58	61	45	49	57	36	38	40	61	64	67	37	38	40
4,0	51	52	55	57	59	62	46	50	59	37	39	41	63	65	68	38	39	41
5,0	52	54	57	58	60	63	48	52	61	38	40	42	64	66	70	38	40	42

Рисунок 3.1 – Таблица расчета использования светового потока

Подставим все значения в формулу для определения светового потока F :

$$F = (300 \cdot 1,5 \cdot 40 \cdot 1,1) / 0,46 = 43\,043 \text{ Лм}$$

Для установки выберем люминесцентные лампы типа ЛБ-80 со световым потоком 5250 люкс. Чтобы получить количество ламп для установки, разделим полученный необходимый световой поток на значение светового потока одной лампы, и получим необходимое количество ламп $N = 8$. В качестве осветительных приборов будут использоваться светильники типа ОД, в котором содержится 2 лампы.

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		92

Заключение

В ходе выполнения дипломного проекта была изучена предметная область, в результате чего было сформулировано техническое задание на разработку клиентской части автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа.

Был произведен выбор инструментов для разработки клиентской части автоматизированной системы, выполнено инфологическое проектирование, в результате которого были разработаны диаграммы: вариантов использования, последовательности, деятельности, классов и развертывания автоматизированной системы.

Было осуществлено проектирование клиентской части системы, в результате которого были разработаны структуры страниц, а также карта экранных форм.

Была разработана клиентская часть, которая позволяет: отображать всю доступную информацию; делать запросы на добавление, изменение и удаление информации на сервер; обеспечивать обмен сообщениями между пользователями; авторизацию в системе; настраивать параметры для ввода данных в систему.

В разделе безопасности жизнедеятельности были выявлены основные вредные факторы воздействия компьютера на человека, описаны требования к офисным помещениям для минимизации воздействия этих факторов, а также произведен расчет освещенности рабочего места в помещении.

В экономическом разделе был произведен расчет трудоемкости и затрат на разработку автоматизированной системы.

Таким образом, задание на дипломное проектирование выполнено в полном объеме.

					ПГУ 1.090501.07.001 ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Техническое обеспечение связи и автоматизации: учеб. пособие / А. В. Чихачев, С. А. Миняев, В. Л. Назаров, О. Н. Немов, А. Н. Мурсаев, В. Б. Попов, Р. В. Еременко, С. В. Зинкин. Пенза: Изд-во ПГУ, 2018. 330 с.
2. Приказ Министра обороны РФ от 28.12.2013 N 969 "Об утверждении Руководства по содержанию вооружения и военной техники общевойскового назначения, военно-технического имущества в Вооруженных Силах Российской Федерации".
3. Фирма 1С: [Электронный ресурс] // Фирма 1С. Официальный сайт. URL: <https://1c.ru/rus/firm1c/firm1c.htm> (Дата обращения: 25.03.2022).
4. Фирма Парус: [Электронный ресурс] // Фирма Парус. Официальный сайт. URL: <https://parus.com/pp/> (Дата обращения: 25.03.2022).
5. Указ президента Российской Федерации «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года» от 7 мая 2018 года.
6. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания», утвержденные Постановлением Главного государственного санитарного врача РФ от 28.01.2021 № 2.
7. Справочная книга для проектирования электрического освещения. / Под ред. Г.Б. Кнорринга. – Л.: Энергия, 1976.

Приложение А

(обязательное)

Клиентская часть автоматизированной системы учета техники связи и
формирования отчетных документов в войсках связи военного округа

Результаты выполнения тестовых примеров

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		95

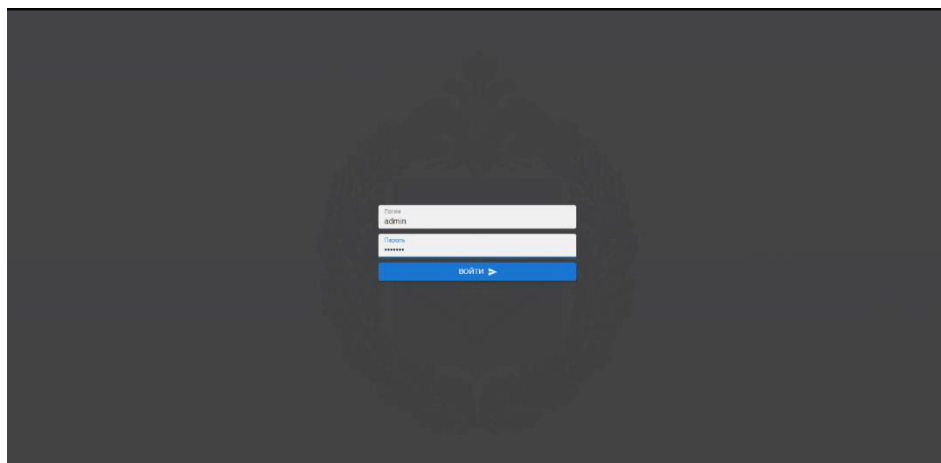


Рисунок А.1 – Авторизация в системе

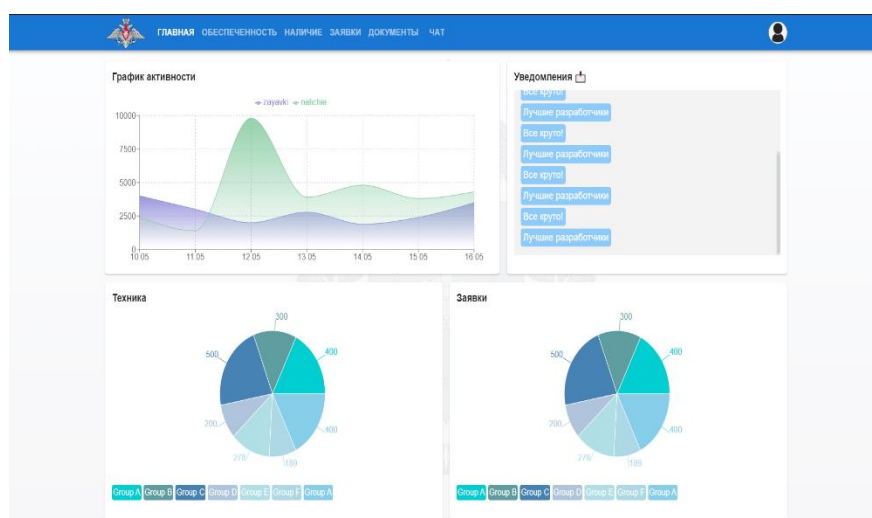


Рисунок А.2 – Главная страница пользователя

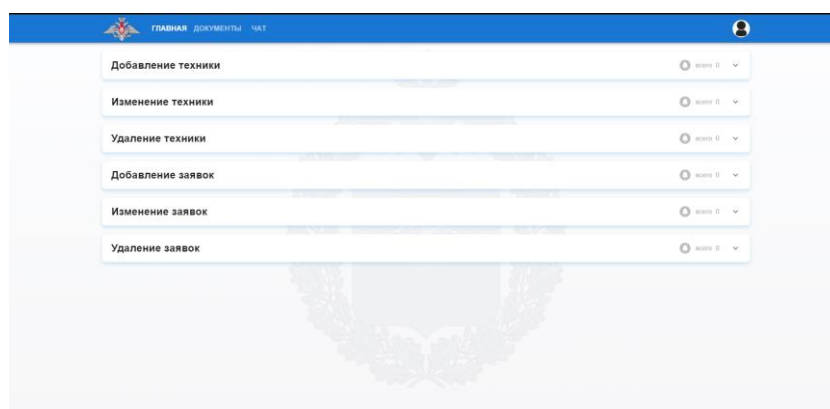


Рисунок А.3 – Главная страница администратора

Рисунок А.4 – Секция добавления пользователя с данными

Рисунок А.5 – Список добавленных пользователей

Рисунок А.6 – Добавление типа техники

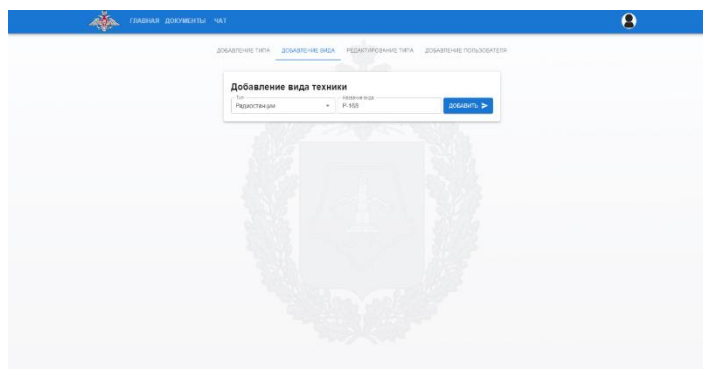


Рисунок А.7 – Добавление вида техники

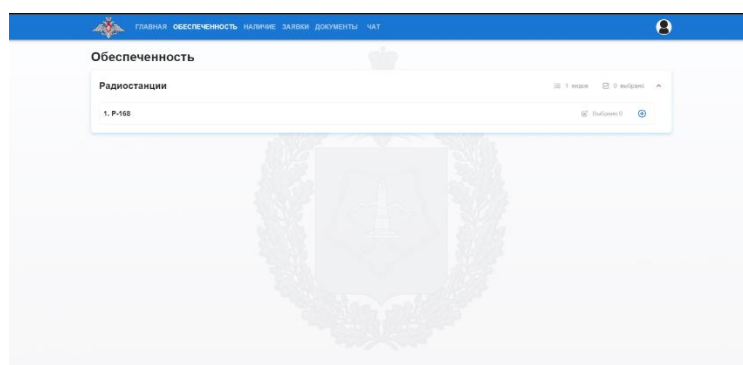


Рисунок А.8 – Страница обеспеченности

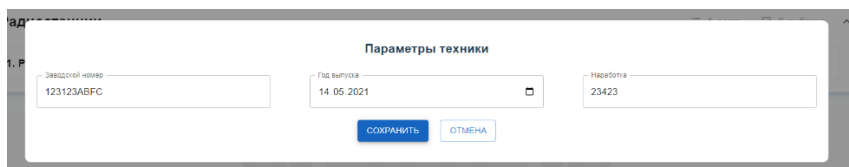


Рисунок А.9 – Форма добавления единицы техники

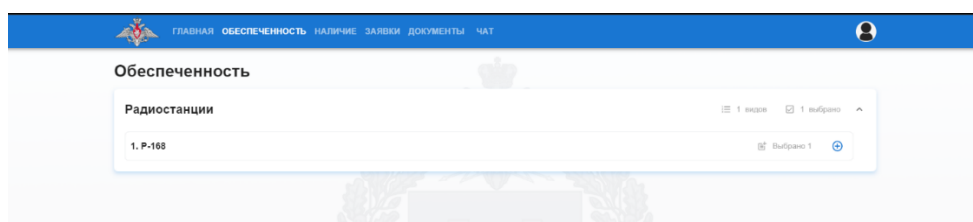


Рисунок А.10 – Увеличения счетчиков типов и видов техники

Добавление техники

№	Вид	Тип	Заводской номер
1	Р-168	Радиостанция	123123ABFC

Изм. 0

Удаление техники

Добавление заявок

Изм. 0

Удаление заявок

Рисунок А.11 – Главная страница администратора с уведомлением о добавлении техники

Подтверждение добавления техники

В/ч 12223 Радиостанции Р-168

Заводской номер 123123ABFC

Год выпуска 14.05.2021

23423

ПОДТВЕРДИТЬ

ОТКЛОНИТЬ

Рисунок А.12 – Форма подтверждения добавления техники

Наличие

№	Вид	Тип	Заводской номер	Статус
1	Р-168	Радиостанция	123123ABFC	Принята

Принять

Рисунок А.13 – Страница наличия с добавленной техникой

Изменение параметров техники

Причина изменения:
Изменилась наработка техники связи

Заводской номер: 123123ABFC Год выпуска: 14.05.2021 Нароботка: 24444

СОХРАНИТЬ **ОТМЕНА**

Рисунок А.14 – Форма изменения параметров техники

Наличие

№	Вид	Тип	Заводской номер	Статус
1	P-168	Радиостанции	123123ABFC	В эксплуатации

Рисунок А.15 – Страница наличия с изменением статуса техники

Добавление техники 0

Изменение техники 1

Удаление техники 0

Добавление заявок 0

Изменение заявок 0

Удаление заявок 0

Рисунок А.16 – Главная страница администратора с уведомлением об изменении техники

Изменение параметров техники

В/ч 12223 Радиостанции Р-168

Причина изменения:
Изменилась наработка техники связи

Новые параметры

Заводской номер: 123123ABFC Год выпуска: 14.05.2021 Нароботка: 24444

Старые параметры

Заводской номер: 123123ABFC Год выпуска: 14.05.2021 Нароботка: 23423

ПОДТВЕРДИТЬ

Причина отклонения:

ОТКЛОНИТЬ

Рисунок А.17 – Форма подтверждения изменения параметров техники

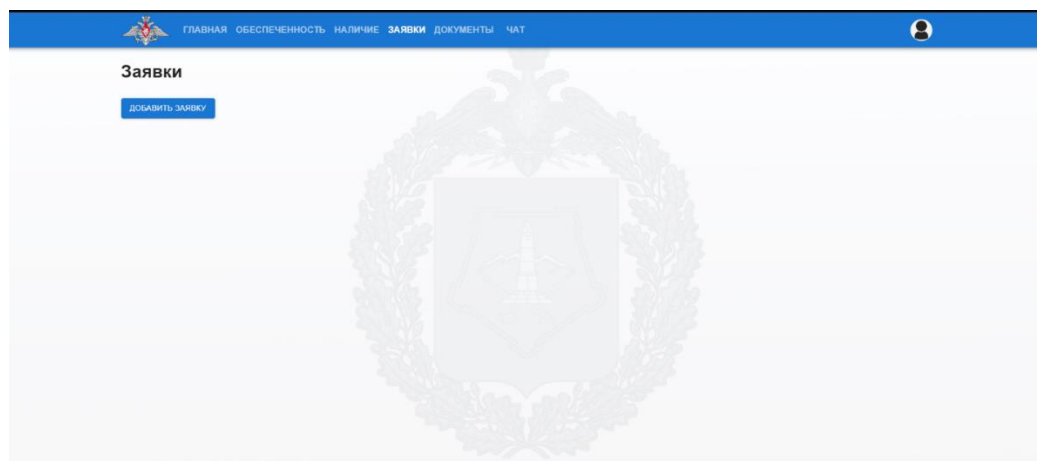


Рисунок А.18 – Страница заявок

Рисунок А.19 –Форма добавления заявки на ремонт

№	Номер заявки	Статус	Примечания
1	12321351	В рассмотрении	Нет

Рисунок А.20 –Страница заявок с добавленной заявкой

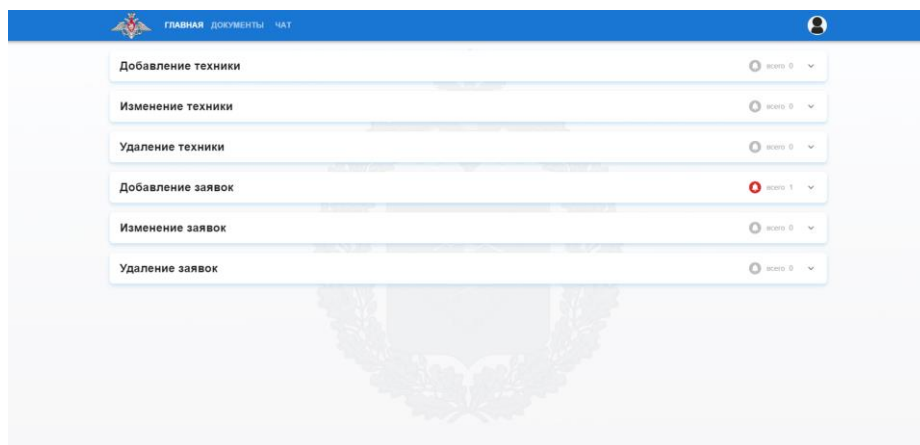


Рисунок А.21 – Главная страница администратора с уведомлением о добавлении заявки

Подтверждение добавления

В/ч 12223

Номер заявки
12321351

Техника
Радиостанция Р-168 Заводской номер 123123ABFC

Место дислокации
Пенза

Проявление неисправности
Сломалось реле

Название неисправного блока
Реле 123

Номер неисправного блока
AS-123213

Габариты
10x10

Количество мест
1

Контактная информация
Петров Иван Иванович

Примечание
Нет

ПОДТВЕРДИТЬ

Причина отклонения

ОТКЛОНИТЬ

Рисунок А.22 –Форма подтверждения добавления заявки на ремонт

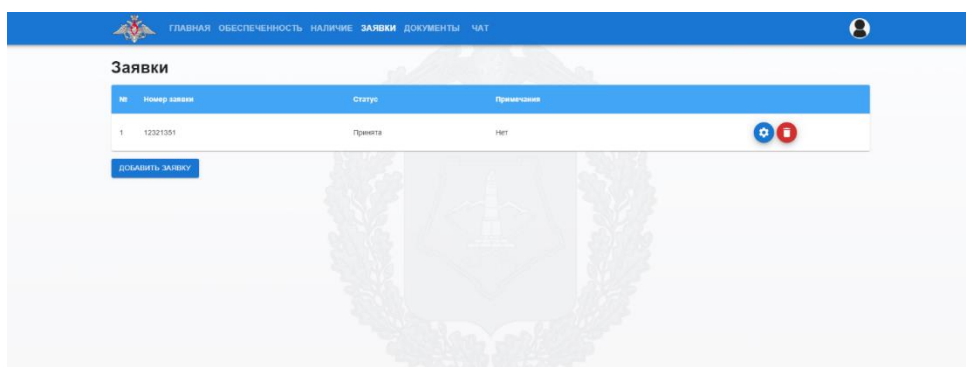


Рисунок А.23 –Страница с подтвержденной заявкой

Изменение заявки

Причина отклонения:
Неправильно указаны габариты

Номер заявки:
12321351

Техника:
Радиостанция Р-168 Заводской номер 123123ABFC

Место дислокации: Пенза

Проявление неисправности: Сломалось реле

Название неисправного блока: Реле 123

Номер неисправного блока: AS-123213

Габариты: 10x15

Количество мест: 1

Контактная информация: Петров Иван Иванович

Приложение: Нет

Рисунок А.24 –Форма изменения заявки на ремонт

ГЛАВНАЯ
ДОКУМЕНТЫ
ЧАТ

Добавление техники	всего 0
Изменение техники	всего 0
Удаление техники	всего 0
Добавление заявок	всего 0
Изменение заявок	всего 1
Удаление заявок	всего 0

Рисунок А.25 – Главная страница администратора с уведомлением об изменении заявки

Подтверждение изменения

Причина отклонения:
Неправильно указаны габариты

В/ч 12223

Номер заявки:
12321351

Габариты: 10x15

Количество мест: 1

Контактная информация: Петров Иван Иванович

Приложение: Нет

Старые параметры

Техника:
Радиостанция Р-168 Заводской номер 123123ABFC

Место дислокации: Пенза

Проявление неисправности: Сломалось реле

Название неисправного блока: Реле 123

Номер неисправного блока: AS-123213

Габариты: 10x10

Количество мест: 1

Контактная информация: Петров Иван Иванович

Приложение: Нет

Причина отклонения:

Рисунок А.26 –Форма подтверждения изменения заявки на ремонт

Рисунок А.27 – Конфигурация отчета

Рисунок А.28 – Доступные для выгрузки документы

Тестовый отчет.xlsx						
	A	B	C	D	E	F
1	№	Тип	Вид	Заводской номер	На гарантии	
2	1	Радиостанции	P-168	123123ABFC	Да	
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

Рисунок А.29 – Сформированный отчет

Удаление техники

Причина удаления
Техника списана, приказ №1112

Рисунок А.30 –Форма удаления техники

Добавление техники 0 заявок

Изменение техники 0 заявок

Удаление техники 1 заявки

Добавление заявок 0 заявок

Изменение заявок 0 заявок

Удаление заявок 0 заявок

Рисунок А.31 – Главная страница администратора с уведомлением об удалении техники

Подтверждение удаления

В/ч 12223 Радиостанции Р-168

Заводской номер: 123123ABFC Год выпуска: 14.05.2021 Нарядовый: 24444

Причина удаления
Техника списана, приказ №1112

Причина отклонения

Рисунок А.32 –Форма подтверждения удаления техники

Удаление заявки

Причина удаления
Заявка не актуальна

Рисунок А.33 –Форма удаления заявки на ремонт

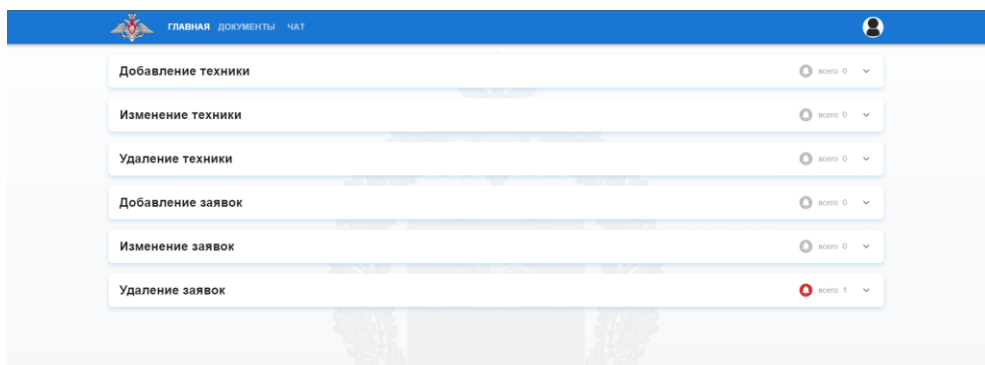


Рисунок А.34 – Главная страница администратора с уведомлением об удалении заявки

Подтверждение удаления

Причина удаления:
Заявка не актуальна

В/ч 12223
Номер заявки:
12321351

ПОДТВЕРДИТЬ

Причина отклонения:

ОТКЛОНИТЬ

Рисунок А.35 –Форма подтверждения удаления заявки на ремонт

Чат

12223

Поиск:

Администратор онлайн

Введите сообщение

Здравствуйте, не получается добавить технику. 09:30

Рисунок А.36 –Отправка сообщения администратору от пользователя

Чат

Администратор

Поиск:

12223 онлайн

Введите сообщение

Здравствуйте, не получается добавить технику. 09:30

Здравствуйте, расскажите о проблеме подробнее или перейдите на страницу справочной информации. 10:30

Рисунок А.37 –Отправка сообщения пользователя от администратора

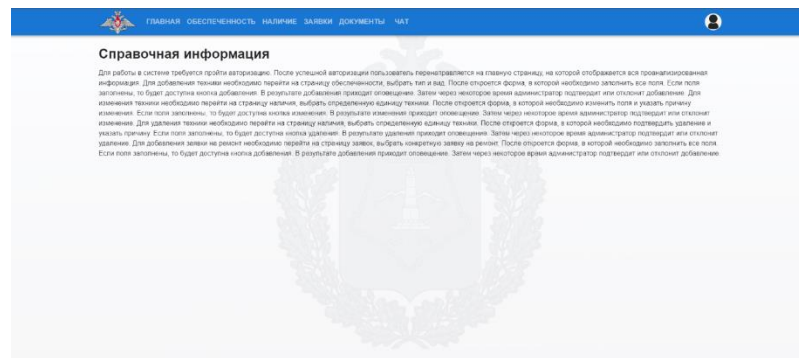


Рисунок А.38 – Страница справочной информации

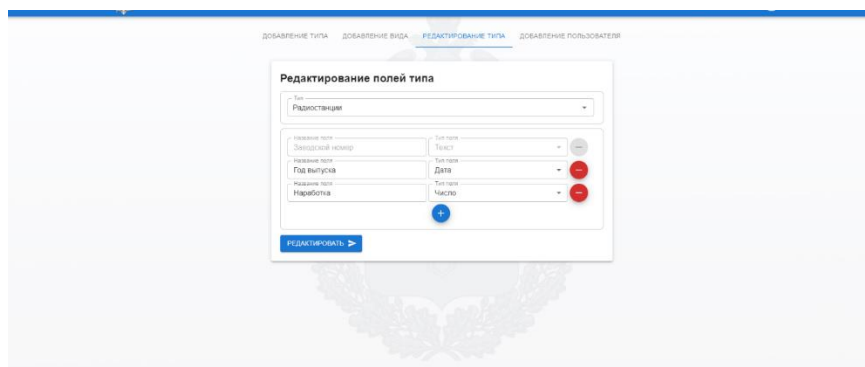


Рисунок А.39 – Редактирование полей типа техники

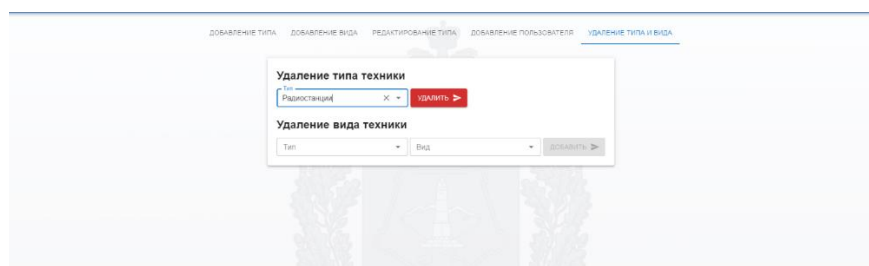


Рисунок А.40 – Удаление типа техники

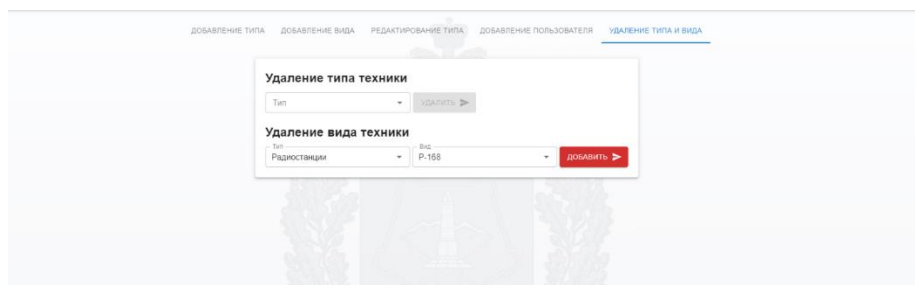


Рисунок А.41 – Удаление вида техники

Приложение Б

(обязательное)

Клиентская часть автоматизированной системы учета техники связи и
формирования отчетных документов в войсках связи военного округа

Листинг клиентской части

					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		108

```

function App() {
  const pages = [
    { name: 'Главная', url: '/', role: 1 },
    { name: 'Обеспеченность', url: '/obespechenost', role: 1 },
    { name: 'Наличие', url: '/nalichie', role: 1 },
    { name: 'Заявки', url: '/zayavki', role: 1 },
    { name: 'Документы', url: '/document', role: 1 },
    { name: 'Чат', url: '/chat', role: 1 },
  ];

  const pagesAdmin = [
    { name: 'Главная', url: '/', role: 1 },
    { name: 'Документы', url: '/document', role: 1 },
    { name: 'Чат', url: '/chat', role: 1 },
  ];

  const settings = ['Профиль', 'Выйти'];

  const [token, setToken] = React.useState(localStorage.getItem('data-laravel') || '');

  const checkToken = (component) => {
    if (token) {
      return component;
    } else {
      return <Navigate to="/login" />;
    }
  };

  return (
    <AppContext.Provider value={{ setToken }}>
      <Header pages={pages} settings={settings} />
      <Container maxWidth="xl" className="bg">
        <Routes>
          <Route path="/" element={checkToken(<HomeAdmin />)}></Route>
          <Route path="/nalichie" element={checkToken(<Nalichie />)}></Route>
        </Routes>
      </Container>
    </AppContext.Provider>
  );
}

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		109

```

    <Route path="/obespechenost" element={checkToken(<Obespechenost />)}></Route>

    <Route path="/zayavki" element={checkToken(<Zayavki />)}></Route>

    <Route path="/add/view" element={checkToken(<ViewTechnikAdd />)}></Route>

    <Route path="/add/edenica" element={checkToken(<EdenicaTechnikAdd />)}></Route>

    <Route path="/add/type" element={checkToken(<TypeTechnikAdd />)}></Route>

    <Route path="/login" element={<Login />}></Route>

    <Route path="/setting" element={checkToken(<ViewTechnikAdd />)}></Route>

    <Route path="/report" element={checkToken(<Report />)}></Route>

    <Route path="/document" element={checkToken(<Documents />)}></Route>

    <Route path="/chat" element={checkToken(<Chat />)}></Route>

    <Route path="/support" element={checkToken(<Support />)}></Route>

    <Route path="/test" element={<Test />}></Route>

  </Routes>

</Container>

</AppContext.Provider>

);
}

const Header = ({ pages, settings }) => {

  const { setToken } = React.useContext(AppContext);

  const navigate = useNavigate();

  let { pathname } = useLocation();

  const showHeader = pathname === '/login';

  const [anchorElNav, setAnchorElNav] = React.useState(null);

  const [anchorElUser, setAnchorElUser] = React.useState(null);

  const handleOpenNavMenu = (event) => {

    setAnchorElNav(event.currentTarget);

  };

  const handleOpenUserMenu = (event) => {

    setAnchorElUser(event.currentTarget);

  };

```

```

const handleCloseNavMenu = () => {
  setAnchorElNav(null);
};

const handleCloseUserMenu = () => {
  setAnchorElUser(null);
};

const logout = () => {
  handleCloseUserMenu();
  localStorage.removeItem('data-laravel');
  axios.get('http://diplom/logout');
  setToken("");
};

return (
  <>
    {!showHeader && (
      <AppBar position="static">
        <Container maxWidth="xl">
          <Toolbar disableGutters>
            <Typography
              variant="h6"
              noWrap
              component="div"
              sx={{ mr: 2, display: { xs: 'none', md: 'flex' } }}
            >
              
            </Typography>

```

					ИГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		111

```

<Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>

  <IconButton

    className="ic-btn"

    size="large"

    aria-label="account of current user"

    aria-controls="menu-appbar"

    aria-haspopup="true"

    onClick={handleOpenNavMenu}

    color="inherit"

  >

    <MenuIcon />

  </IconButton>

  <Menu

    id="menu-appbar"

    anchorEl={anchorElNav}

    anchorOrigin={{

      vertical: 'bottom',

      horizontal: 'left',

    }}

    keepMounted

    transformOrigin={{

      vertical: 'top',

      horizontal: 'left',

    }}

    open={Boolean(anchorElNav)}

    onClose={handleCloseNavMenu}

    sx={{

      display: { xs: 'block', md: 'none' },

    }}

  >

    {pages.map((page) => (

      <MenuItem key={page.name} onClick={handleCloseNavMenu}>

```

					ИГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		112

```

        <Typography textAlign="center">{page.name}</Typography>

      </MenuItem>

    )}

  </Menu>

</Box>

<Typography
  variant="h6"
  noWrap
  component="div"
  sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}
>

</Typography>

<Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' } }}>

  {pages.map((page) => (

    <Link
      to={page.url}
      key={page.name}
      style={{ textDecoration: 'none' }}
    >

      <Button
        className={
          pathname === page.url
            ? 'active--button'
            : 'default--button'
        }
        sx={{
          color: '#fff',
          display: 'block',

```

```

        fontSize: '16px',

        fontWeight: '600',

    }}

    >

        {page.name}

    </Button>

</Link>

)}}

</Box>

<Box sx={{ flexGrow: 0 }}>

    <Tooltip title="Открыть настройки">

        <IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>

            <Avatar alt="Пользователь" src="user.png" />

        </IconButton>

    </Tooltip>

    <Menu

        sx={{ mt: '45px' }}

        id="menu-appbar"

        anchorEl={anchorElUser}

        anchorOrigin={{

            vertical: 'top',

            horizontal: 'right',

        }}

        keepMounted

        transformOrigin={{

            vertical: 'top',

            horizontal: 'right',

        }}

        open={Boolean(anchorElUser)}

        onClose={handleCloseUserMenu}

    >

        <MenuItem

```

```

onClick={() => {
    navigate('/report');
    handleCloseUserMenu();
}}
>

<Typography textAlign="center">
    Конструктор отчетов
</Typography>
</MenuItem>
<MenuItem
    onClick={() => {
        navigate('/setting');
        handleCloseUserMenu();
    }}
>
    <Typography textAlign="center">Настройка данных</Typography>
</MenuItem>
<MenuItem
    onClick={() => {
        navigate('/support');
        handleCloseUserMenu();
    }}
>
    <Typography textAlign="center">Справочная информация
</Typography>
</MenuItem>
<MenuItem onClick={logout}>
    <Typography textAlign="center">Выйти</Typography>
</MenuItem>
</Menu>
</Box>
</Toolbar>

```

```

        </Container>

    </AppBar>

  )}

</>

);

};

export default Header;

import React from 'react';

import { styled } from '@mui/material/styles';

import TextField from '@mui/material/TextField';

import Button from '@mui/material/Button';

import InputLabel from '@mui/material/InputLabel';

import MenuItem from '@mui/material/MenuItem';

import FormHelperText from '@mui/material/FormHelperText';

import FormControl from '@mui/material/FormControl';

import Select from '@mui/material/Select';

import SendIcon from '@mui/icons-material/Send';

import LoadingButton from '@mui/lab/LoadingButton';

import IconButton from '@mui/material/IconButton';

import AddIcon from '@mui/icons-material/Add';

import Fab from '@mui/material/Fab';

import RemoveIcon from '@mui/icons-material/Remove';

import axios from 'axios';

import Autocomplete from '@mui/material/Autocomplete';

const EditType = ({ setOpen, setMessage }) => {

  const getType = async () => {

    const obj = { action: 'get_all_type' };

    console.log('Отправка данных:', obj);

    const res = await axios.post('http://diplom/add_table', obj);

    console.log('Ответ:', res.data);

    setTypeSelect(res.data);
  }

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		116

```

};

const getCells = async (table_id) => {
  const obj = { name_table: findNameTable(Number(table_id)), action: 'get_current_cells' };
  console.log('Отправка данных:', obj);
  try {
    const res = await axios.post('http://diplom/add_table', obj);
    console.log('Ответ:', res.data);
    setNewfields(res.data);
  } catch (error) {
    console.log('Ответ:', error);
  }
};

```

```

React.useEffect(() => {
  getType();
}, []);

```

```

const [typeSelect, setTypeSelect] = React.useState([]);

```

```

const setEdenica = async () => {
  const obj = {
    action: 'edit_type',
    name_table: findNameTable(Number(field)),
    cells: newfields,
  };
  console.log('Отправка данных:', obj);
  try {
    const res = await axios.post('http://diplom/add_table', obj);
    console.log('Ответ:', res.data);
    setLoading(false);
    if (res.data) {
      setOpen(true);
    }
  }
};

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		117

```

        setMessage(true);

        setField("");

        setNewfields([]);

    } else {

        setOpen(true);

        setMessage(false);

    }

} catch (error) {

    console.log(error);

}

};

const [field, setField] = React.useState("");

const [newfields, setNewfields] = React.useState([]);

const handleChange = (event) => {

    setField(event.target.value);

    if (event.target.value) {

        getCells(event.target.value);

    } else {

        setNewfields([]);

    }

};

const changeNewInput = (event) => {

    if (event.target.name.includes('select')) {

        let index = Number(event.target.name.split('select-new-')[1]);

        let tempArray = [...newfields];

        tempArray[index].baseType = event.target.value;

        setNewfields(tempArray);

    } else {

        let index = Number(event.target.name.split('text-new-')[1]);

```

					<p><i>ПГУ 1.090501.07.001 ПЗ</i></p>	Лист
						118
Изм.	Лист	№ докум.	Подпись	Дата		

```

    let tempArray = [...newfields];

    tempArray[index].name = event.target.value;

    setNewfields(tempArray);

  }
};

const findNameTable = (id) => {
  let obj = typeSelect.find((item) => item.id == id);
  return obj.table_name;
};

const handleAddClick = (event) => {
  setNewfields([
    ...newfields,
    {
      name: "",
      baseType: "",
    },
  ]);
};

const handleRemoveClick = (value) => {
  setNewfields(
    newfields.filter((elem, index) => {
      return index !== value;
    })
  );
};

const [loading, setLoading] = React.useState(false);

const handleClick = () => {
  setLoading(true);

```

					ИГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		119

```

    setEdenica();

};

const validation = () => {

    let isValid = !!field;

    if (newfields.length) {

        newfields.forEach((element) => {

            if (element.name !== 'view ид') {

                isValid = isValid && !!element.name && !!element.baseType;

            }

        });

    } else {

        isValid = false;

    }

    console.log(isValid);

    return isValid;

};

```

```

const typeChange = (event, newValue) => {

    console.log(newValue);

    if (newValue) {

        setField(newValue.id);

        getCells(newValue.id);

    } else {

        setNewfields([]);

    }

};

```

```

return (

    <div>

        <h2>Редактирование полей типа</h2>

        <div

            style={{

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		120

```

padding: 15,
border: '1px rgb(187 187 187 / 87%) solid',
borderRadius: 8,
width: '100%',
display: 'flex',
}}
>
{false && (
  <FormControl sx={{ width: '100%' }} size="small">
    <InputLabel id="label-1">Тип техники</InputLabel>
    <Select
      labelId="label-1"
      id="select-1"
      value={field}
      label="Тип техники"
      name="type"
      onChange={handleChange}
    >
      <MenuItem value="">
        <em>Не выбрано</em>
      </MenuItem>
      {typeSelect.map((value, index) => (
        <MenuItem key={index} value={value.id}>
          {value.name}
        </MenuItem>
      ))}
    </Select>
  </FormControl>
)}
<Autocomplete
  size="small"
  disablePortal
  id="select-view"

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		121

```

options={typeSelect}

onChange={typeChange}

value={field ? typeSelect.find((el) => el.id == field) : null}

getOptionLabel={ (option) => option.name}

sx={{ width: '100%', marginRight: '5px', background: '#fff' }}

renderInput= {(params) => <TextField name="type" {...params} label="Тип" />}

/>

</div>

<div

  className={field == '' ? 'block-disabled' : ''}

  style={{

    padding: 15,

    border: '1px rgb(187 187 187 / 87%) solid',

    borderRadius: 8,

    display: 'flex',

    flexDirection: 'column',

    width: 'fit-content',

    marginTop: 10,

    width: '100%',

  }}

>

  {newfields.map(

    (elem, index) =>

      elem.name !== 'виюв ид' && (

        <div

          key={index}

          style={{ display: 'flex', marginBottom: '5px', marginTop: '5px' }}

          >

            <TextField

              disabled={elem.name === 'Заводской номер'}

              id={'text-new-' + index}

              name={'text-new-' + index}

              label="Название поля"

```

```

variant="outlined"

size="small"

sx={{ width: '45%', marginRight: '5px' }}

value={elem.name}

onChange={changeNewInput}

/>

<FormControl

  disabled={elem.name === 'Заводской номер'}

  sx={{ width: '45%', marginRight: '5px' }}

  size="small"

>

  <InputLabel id={`label-new-${index}`}>Тип поля</InputLabel>

  <Select

    labelId={`label-new-${index}`}

    id={`select-new-${index}`}

    name={`select-new-${index}`}

    label="Тип поля"

    onChange={changeNewInput}

    value={elem.baseType}

  >

    <MenuItem value="">

      <em>Не выбрано</em>

    </MenuItem>

    <MenuItem value={`number`} >Число</MenuItem>

    <MenuItem value={`text`} >Текст</MenuItem>

    <MenuItem value={`date`} >Дата</MenuItem>

  </Select>

</FormControl>

<Fab

  disabled={elem.name === 'Заводской номер'}

  color="error"

  aria-label="add"

  size="small"

```

```

        onClick={() => {
            handleRemoveClick(index);
        }}
    >

    <RemoveIcon />

</Fab>

</div>

),
)}

<div style={{ display: 'flex', justifyContent: 'center' }}>

    <Fab

        disabled={field == ""}

        color="primary"

        aria-label="add"

        size="small"

        onClick={handleAddClick}

        sx={{ marginRight: '3px' }}

    >

        <AddIcon />

    </Fab>

</div>

</div>

<LoadingButton

    onClick={handleClick}

    disabled={!validation()}

    endIcon={<SendIcon />}

    loading={loading}

    loadingPosition="end"

    variant="contained"

    sx={{ marginTop: 1.5 }}

    >

        Редактировать

</LoadingButton>

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		124

```

    </div>

);
};

const handleRemoveClick = (value) => {
    setNewfields(
        newfields.filter((elem, index) => {
            return index !== value;
        }),
    );
};

const [loading, setLoading] = React.useState(false);

const handleClick = () => {
    setLoading(true);
    setEdenica();
};

const validation = () => {
    let isValid = !!field;
    if (newfields.length) {
        newfields.forEach((element) => {
            isValid = isValid && !!element.name && !!element.baseType;
        });
    } else {
        isValid = false;
    }
    return isValid;
};

return (

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		125

```

<div>

  <h2>Добавление полей единицы техники</h2>

  <div

    style={{

      padding: 15,

      border: '1px rgb(187 187 187 / 87%) solid',

      borderRadius: 8,

      width: '100%',

      display: 'flex',

    }}

  >

    <TextField

      id="type"

      name="type"

      label="Название типа"

      variant="outlined"

      size="small"

      sx={{ width: '100%' }}

      value={field}

      onChange={handleChange}

    />

  </div>

  <div

    className={field == " " ? 'block-disabled' : ""}

    style={{

      padding: 15,

      border: '1px rgb(187 187 187 / 87%) solid',

      borderRadius: 8,

      display: 'flex',

      flexDirection: 'column',

      width: 'fit-content',

      marginTop: 10,

      width: '100%',

```

```

    }}
  >
  {newfields.map((elem, index) => (
    <div
      key={index}
      style={{ display: 'flex', marginBottom: '5px', marginTop: '5px' }}
    >
      <TextField
        disabled={index === 0}
        id={'text-new-' + index}
        name={'text-new-' + index}
        label="Название поля"
        variant="outlined"
        size="small"
        sx={{ width: '45%', marginRight: '5px', background: '#fff !important' }}
        value={elem.name}
        onChange={changeNewInput}
      />
      <FormControl
        disabled={index === 0}
        sx={{ width: '45%', marginRight: '5px' }}
        Добавить
      </LoadingButton>
    </div>
  )
);
};

export default AddEdenica;

import React from 'react';
import { styled } from '@mui/material/styles';
import TextField from '@mui/material/TextField';
import Button from '@mui/material/Button';
import InputLabel from '@mui/material/InputLabel';

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		127

```

import MenuItem from '@mui/material/MenuItem';

import FormHelperText from '@mui/material/FormHelperText';

import FormControl from '@mui/material/FormControl';

import Select from '@mui/material/Select';

import SendIcon from '@mui/icons-material/Send';

import LoadingButton from '@mui/lab/LoadingButton';

import IconButton from '@mui/material/IconButton';

import AddIcon from '@mui/icons-material/Add';

import Fab from '@mui/material/Fab';

import { Paper } from '@mui/material';

import axios from 'axios';

const AddType = ({ setOpen, setMessage }) => {

  const [field, setField] = React.useState("");

  const handleChange = (event) => {

    setField(event.target.value);

  };

  const setType = async (value) => {

    const res = await axios.post('http://diplom/add_table', { ...value, action: 'set_type' });

    console.log(res.data);

    setLoading(false);

    if (res.data) {

      setOpen(true);

      setMessage(true);

      setField("");

    } else {

      setMessage(false);

    }

  };

  const [loading, setLoading] = React.useState(false);

  function handleClick() {

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		128

```

        setLoading(true);

        setType({});
    }

    return (
        <div>
            <h2>Добавление типа техники</h2>
            <div style={{ display: 'flex' }}>
                <TextField
                    id="outlined-basic"
                    label="Название типа"
                    variant="outlined"
                    size="small"
                    sx={{ height: '100% !important', marginRight: '5px' }}
                    onChange={handleChange}
                    value={field}
                />
                <LoadingButton
                    onClick={handleClick}
                    endIcon={<SendIcon />}
                    loading={loading}
                    loadingPosition="end"
                    variant="contained"
                    disabled={field == " " ? true : false}
                >
                    Добавить
                </LoadingButton>
            </div>
        </div>
    );
};

export default AddType;

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		129

```

import React from 'react';

import TextField from '@mui/material/TextField';

import InputLabel from '@mui/material/InputLabel';

import MenuItem from '@mui/material/MenuItem';

import FormControl from '@mui/material/FormControl';

import Select from '@mui/material/Select';

import SendIcon from '@mui/icons-material/Send';

import LoadingButton from '@mui/lab/LoadingButton';

import axios from 'axios';

import Autocomplete from '@mui/material/Autocomplete';

```

```

const AddView = ({ setOpen, setMessage }) => {

  const obj = { action: 'get_all_type' };

  React.useEffect(() => {

    console.log('Отправка данных:', obj);

    axios.post('http://diplom/add_table', obj).then((res) => {

      console.log('Ответ:', res.data);

      setmasSelect(res.data);

    });

  }, []);

```

```

const setType = async (value) => {

  console.log('Отправка данных:', { ...value, action: 'set_view' });

  const res = await axios.post('http://diplom/add_table', { ...value, action: 'set_view' });

  console.log('Ответ:', res.data);

  setLoading(false);

  if (res.data) {

    setOpen(true);

    setMessage(true);

    setFields({

      view: "",

      type: "",

    });

  }

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		130

```

    } else {

        setOpen(true);

        setMessage(false);

    }

};

const [masSelect, setmasSelect] = React.useState([]);

const [fields, setFields] = React.useState({

    view: "",

    type: "",

});

const handleChange = (event) => {

    setFields({

        ...fields,

, index) => (

        <MenuItem key={index} value={value.id}>

            {value.name}

        </MenuItem>

        )))

    </Select>

    </FormControl>

    })

    <Autocomplete

        size="small"

        disablePortal

        id="select-view"

        options={masSelect}

        onChange={typeChange}

        value={fields.type ? masSelect.find((el) => el.id == fields.type) : null}

        getOptionLabel={(option) => option.name}

```

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		131

```

        sx={{ width: '40%', marginRight: '5px', background: '#fff' }}

        renderInput={ (params) => <TextField name="type" {...params} label="Тип" /> }
    />

    <TextField

        id="text"

        label="Название вида"

        variant="outlined"

        size="small"

        name="view"

        onChange={handleChange}

        value={!fields.type ? fields.view : ""}

        disabled={!fields.type}

        sx={{ width: '40%', height: '100% !important', marginRight: '5px' }}

    />

    <LoadingButton

        onClick={handleClick}

        endIcon={ <SendIcon /> }

        loading={loading}

        loadingPosition="end"

        variant="contained"

        disabled={!isValid}

    >

        Добавить

    </LoadingButton>

</div>

</div>

);

};

```

```

        if (field.name !== 'view id') {

            return (

                <TextField

                    key={field.name}

```

					ПГУ 1.090501.07.001 ПЗ	Лист
						132
Изм.	Лист	№ докум.	Подпись	Дата		

```

        name={ field.name}

        label={ field.name}

        type={field.baseType}

        onChange={handleChange}

        value={ field.value}

    />

    );

}

}}

</form>

<div className={styles.ModalButtons}>

    <Button onClick={sendFields} variant="contained">

        Сохранить

    </Button>

    <Button onClick={handleClickClose} variant="outlined">

        Отмена

    </Button>

</div>

</div>

</div>

</ThemeProvider>

);

};

```

Приложение В

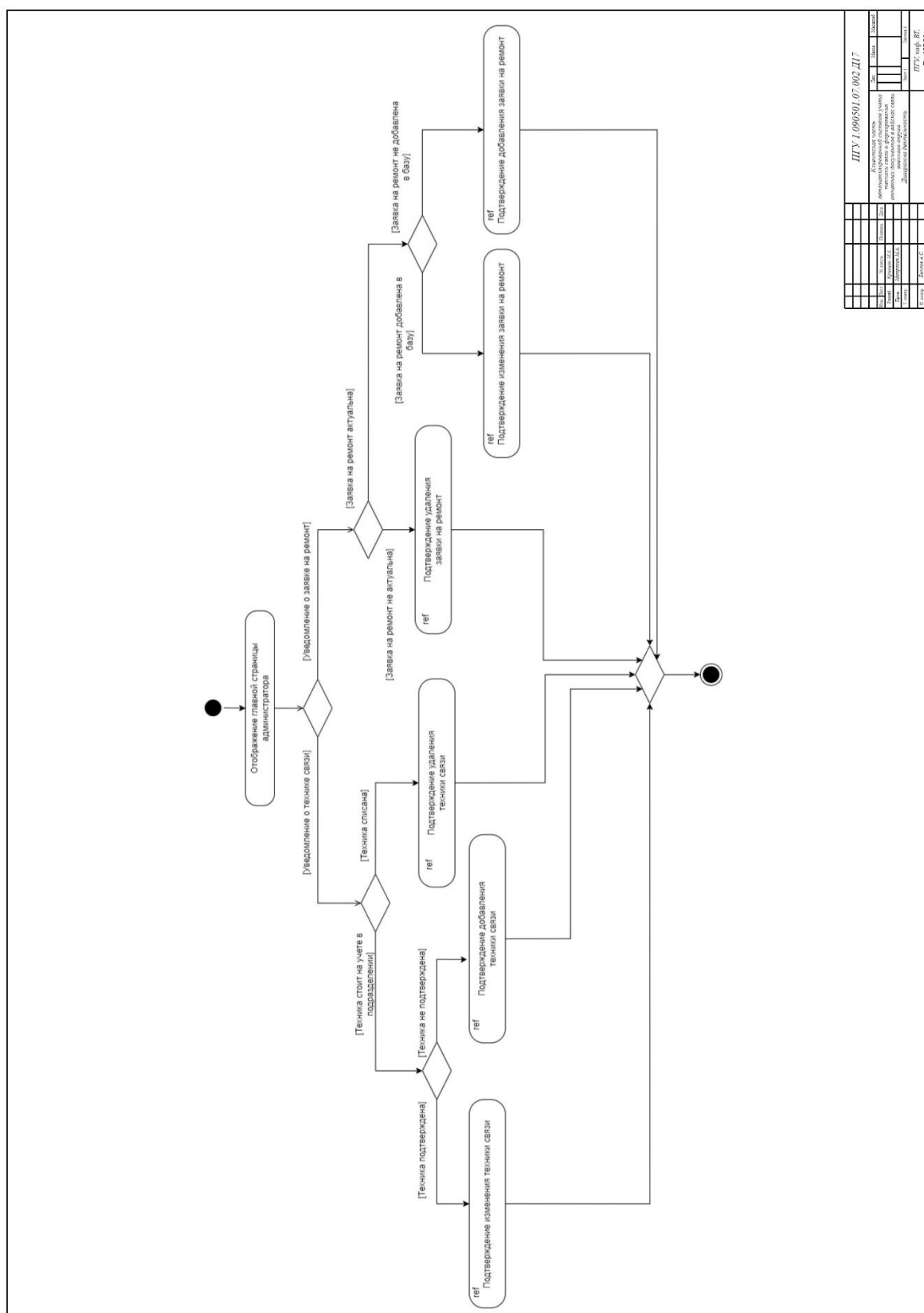
(обязательное)

Клиентская часть автоматизированной системы учета техники связи и
формирования отчетных документов в войсках связи военного округа

UML - диаграммы

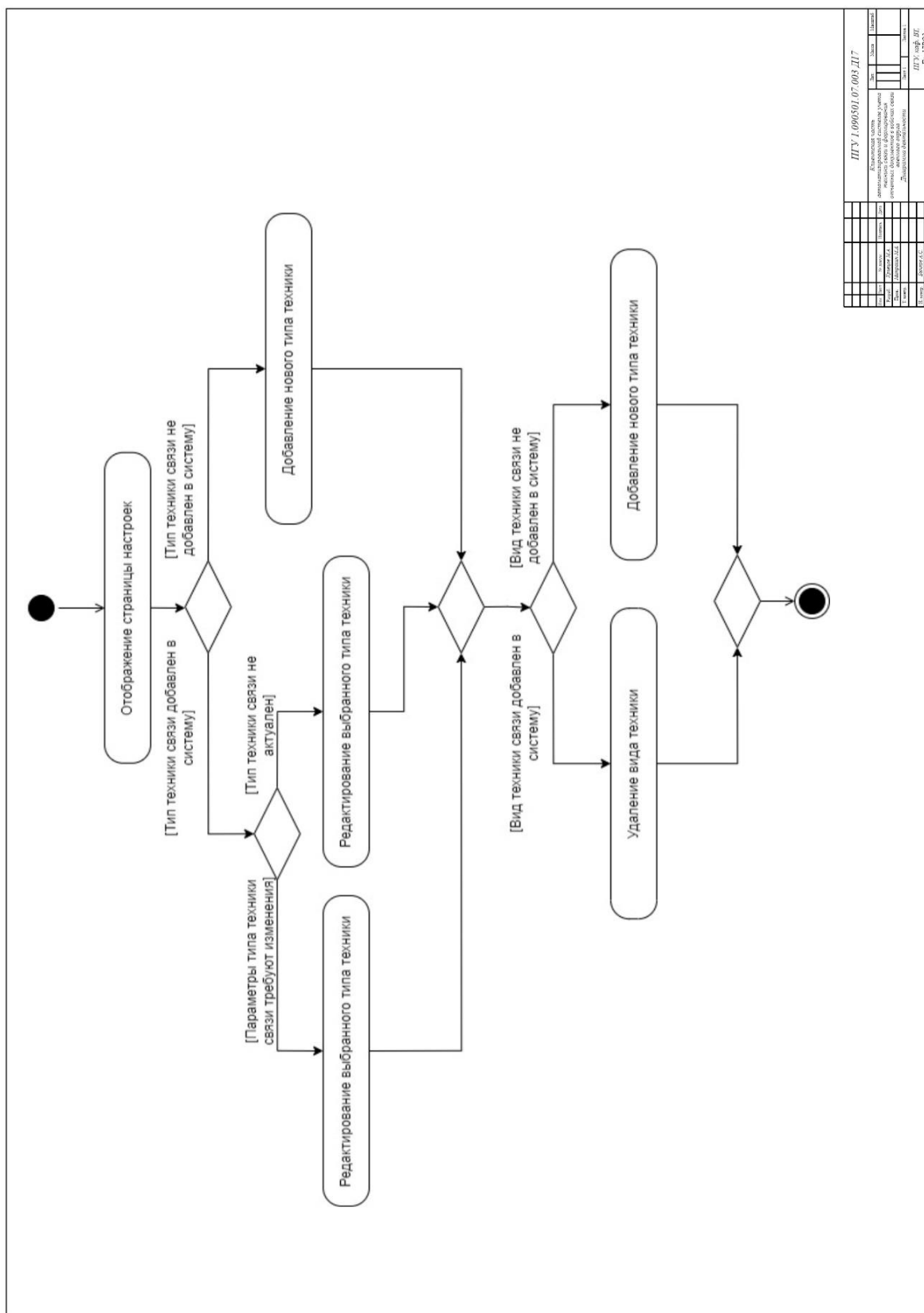
					<i>ПГУ 1.090501.07.001 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		134

Приложение В.1 – Диаграмма деятельности подтверждения действий пользователя



ПУУ 1.090501.07.002 Д17									
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.
И.О.П.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.	Д.И.И.

Приложение В.2 – Диаграмма деятельности настройки параметров для ввода пользователем данных



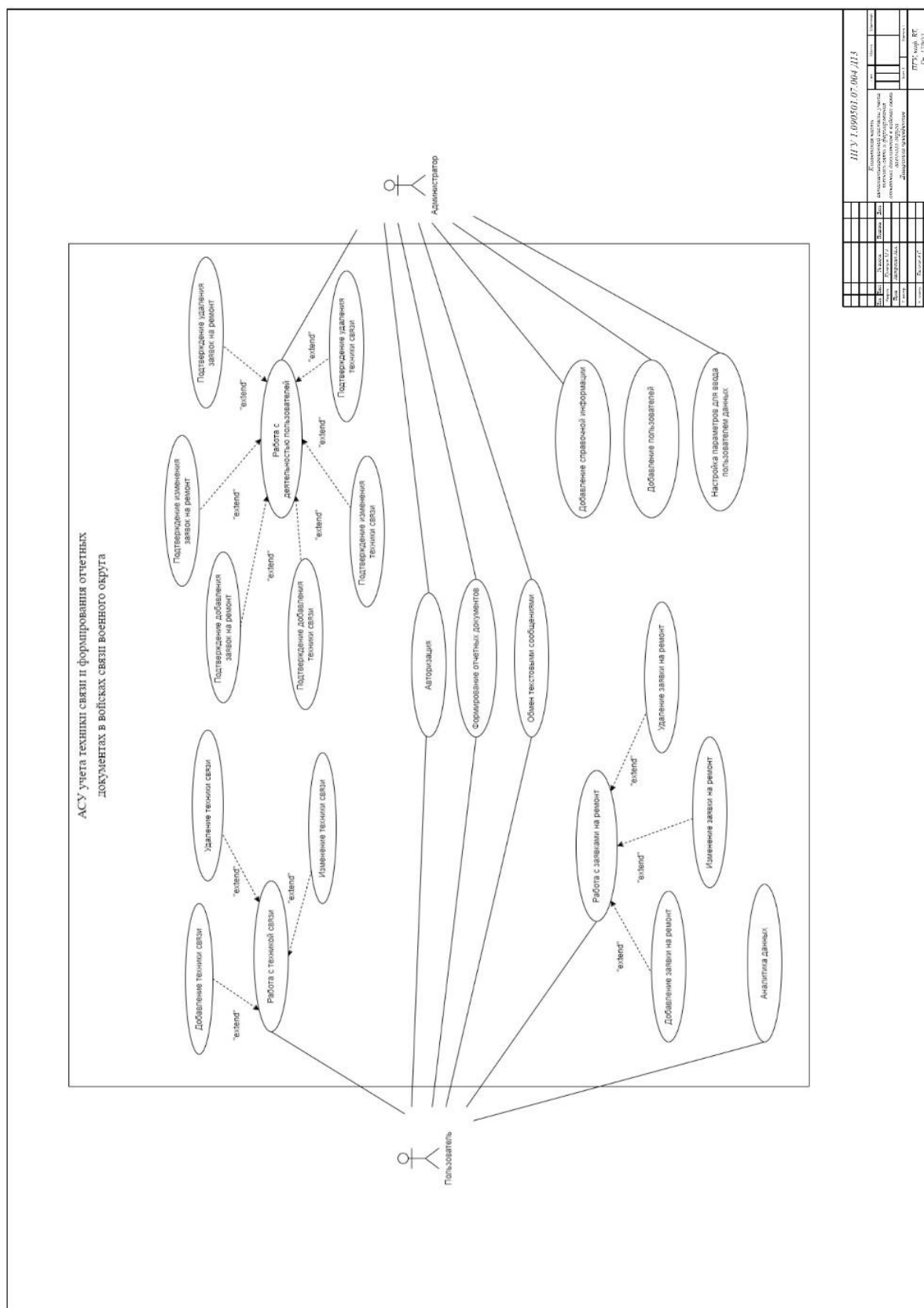
ПГУ 1.090501.07.003 ПЗ									
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель	Исполнитель

Изм.	Лист	№ докум.	Подпись	Дата

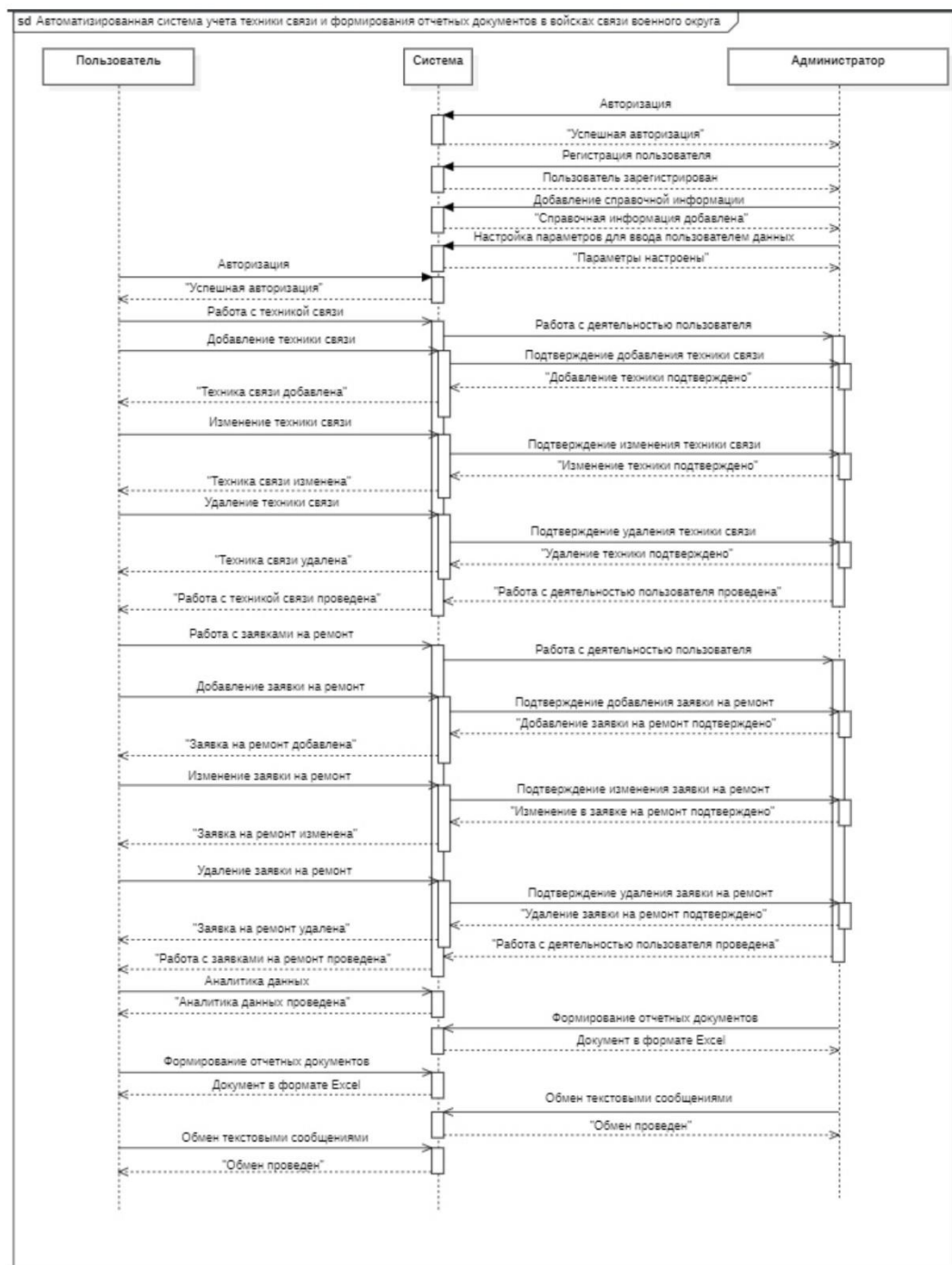
ПГУ 1.090501.07.001 ПЗ

Лист
136

Приложение В.3 – Диаграмма прецедентов

[illegible]

Приложение В.4 – Диаграмма последовательностей



ПУ 1.090501.07.005 Д14			
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность
Имя	Фамилия	Инициалы	Должность

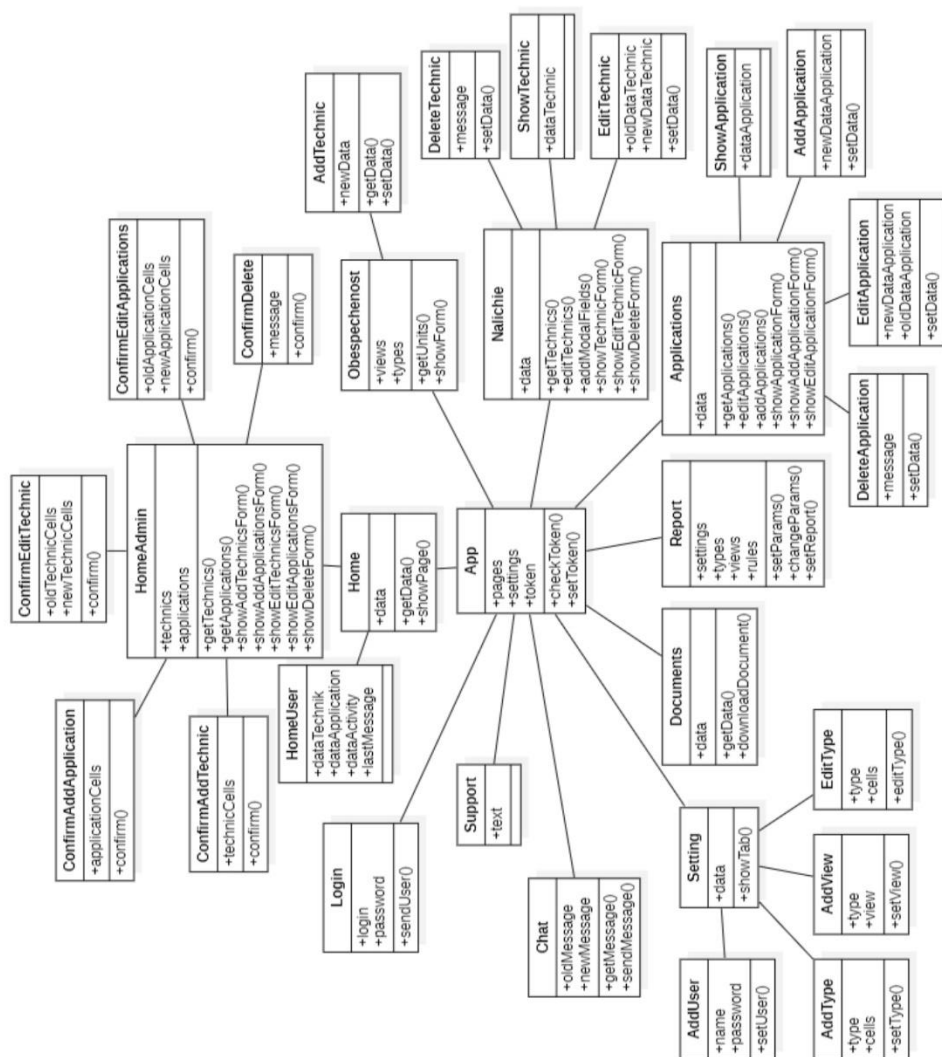
Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ПУ 1.090501.07.001 ПЗ

Лист

138

Приложение В.5 – Диаграмма классов

[illegible]

Приложение Г

(обязательное)

Клиентская часть автоматизированной системы учета техники связи и
формирования отчетных документов в войсках связи военного округа

Презентация

					ПГУ 1.090501.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		140

КЛИЕНТСКАЯ ЧАСТЬ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ТЕХНИКИ СВЯЗИ И ФОРМИРОВАНИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ В ВОЙСКАХ СВЯЗИ ВОЕННОГО ОКРУГА

Выполнил: студент гр.17ВО2 Кривцов М.А.

Руководитель: д.т.н., профессор, зав. каф. ВТ Митрохин М.А.

1

Цель работы и объекты

Цель работы – разработать клиентскую часть автоматизированной системы учета техники связи и формирования отчетных документов в войсках связи военного округа.

Объект работы – клиентская часть автоматизированной системы, выполняющая следующие функции:

- отображать всю доступную информацию;
- делать запросы на добавление, изменение и удаление информации на сервер;
- обеспечивать обмен сообщениями между пользователями;
- авторизацию в системе;
- настраивать параметры для ввода данных в систему.

2

Рекомендации к разработке

**Исключению военного учебного центра при
Пензенском государственном университете**
Пензенский
Александрович
Удальцов, Андрей Александрович

Согласно Вам, что средства учебного центра (УЦ) при проведении
обязательной стажировки на ЗИ ППС и АСУ Центрального военного округа
требовались, разработкой программного обеспечения для автоматизированного
учета техники связи и формирования отчетных документов в войсковой
военной службе.

Необходимость разработки данного программного продукта
выступила вследствие объема задач, решаемых органами военного
обеспечения войсками и АСУ.

Данная работа была выполнена в соответствии с требованиями
к работе для военных служб войсковой.

Наличие из этого, целью определения качества работы военного
назначения, дальнейшей разработки данной системы в рамках текущей
квалификационной работы.

Приложение:
1. Список курсов военного учебного центра при Пензенском
государственном университете, программа обязательной стажировки на
ЗИ ППС и АСУ ЦВО.

Протокол исполнительный обязанности
начальника штаба Центрального военного округа
генерал-майор
№ 08 2021 г.

С. Удальцов
Г. Тучков

Приложение №3

СПИСОК
курсов военного учебного центра при Пензенском государственном
университете, программа обязательной стажировки на
ЗИ ППС и АСУ Центрального военного округа

№	Дата	Фамилия, Имя, Отчество	Учебный
1.	04.05.1999	ДРЯГУН Виталий Владимирович	УЦОИ
2.	31.12.1999	ГУРВН Арина Олеговна	УЦОИ
3.	11.01.2000	КРЮКОВ Михаил Андреевич	УЦОИ
4.	22.08.1999	ТРОСЧУКОВ Владимир Александрович	УЦОИ

Протокол исполнительный обязанности
начальника штаба Центрального военного округа
генерал-майор
№ 08 2021 г.

Г. Тучков

3

Анализ предметной области



Развитие войск связи –
один из самых высоких
приоритетов в программе
обновления армии. По
мнению экспертов, по своей
эффективности применение
более современных средств
управления и связи
сопоставимо со
значительным увеличением
количества боевых средств
или с повышением их
боевых возможностей

На сегодняшний день, в процессе развития войск связи в
Вооруженных Силах Российской Федерации, актуален вопрос об
автоматизации учета вооружения и военной техники, и других
материальных ценностей.

4

Сравнение аналогов

Название	Плюсы	Минусы
1С Предприятие	<ul style="list-style-type: none"> - возможность изменения конфигурации для решения своих задач; - информационно-технологическое сопровождение; - ежемесячные обновления; - российская разработка; - возможность формирования отчетов. 	<ul style="list-style-type: none"> - для доработки конфигурации нужен обученный специалист; - невозможно работать с графикой; - работает на операционных системах семейства Microsoft Windows; - отсутствуют обновления языка; - отсутствует чат.
МойСклад	<ul style="list-style-type: none"> - возможность разграничения доступа; - шифрование данных; - российская разработка; - не требует установки на ПК; - простой в использовании. 	<ul style="list-style-type: none"> - для работы требуется интернет; - подходит для малых предприятий; - плохая техническая поддержка; - отсутствует конструктор отчетов; - отсутствует чат.
Okdesk	<ul style="list-style-type: none"> - для настройки не требуется специалист; - работает с большим количеством операционных систем; - удобный интерфейс; - присутствует конструктор отчетов; - хорошая техническая поддержка. 	<ul style="list-style-type: none"> - для работы требуется интернет; - сложности с поиском в базе нужной позиции; - отсутствует возможность разграничения прав пользователей; - отсутствует чат.
АИС Парус	<ul style="list-style-type: none"> - используется в государственном секторе; - доступна интеграция с продуктами фирм: 1С, Аскон, Адам, Интертех и прочими; - высокий уровень технической поддержки. 	<ul style="list-style-type: none"> - только разработчики имеют право проводить модификацию базовых модулей и приспосабливать их к специфике конкретного предприятия; - стоимость этой программы определяется стоимостью одного рабочего; - отсутствие возможности работы в веб-среде; - отсутствует чат.

5

Архитектура REST API



6

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.07.001 ПЗ

Лист

143

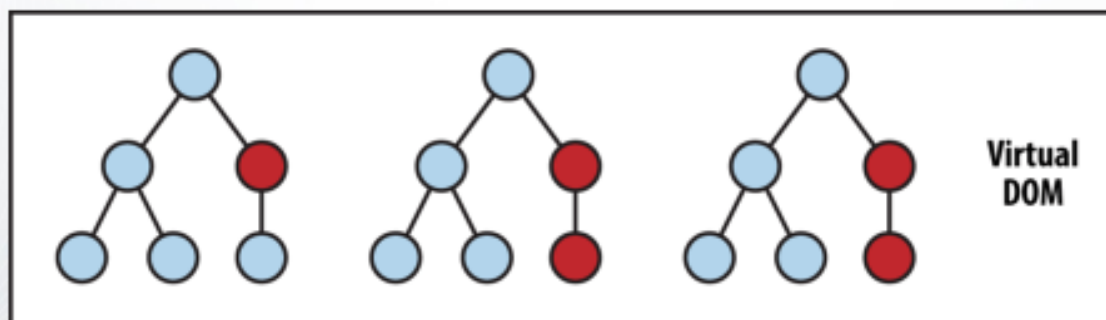
Выбор средств разработки



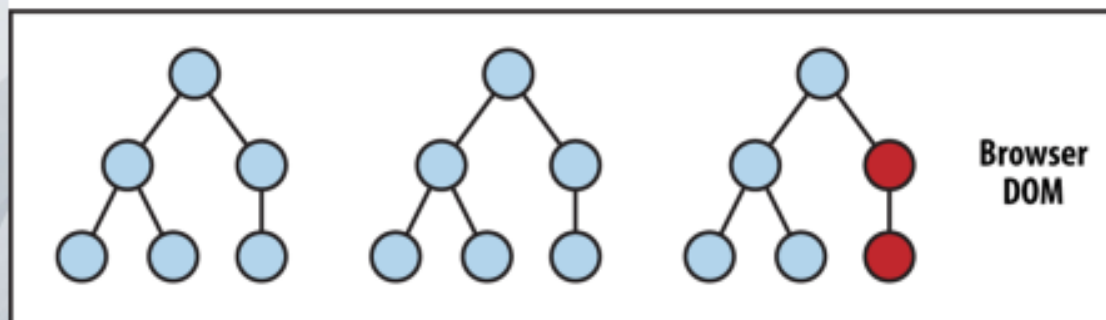
Recharts

7

Virtual DOM



State Change → Compute Diff → Re-render



8

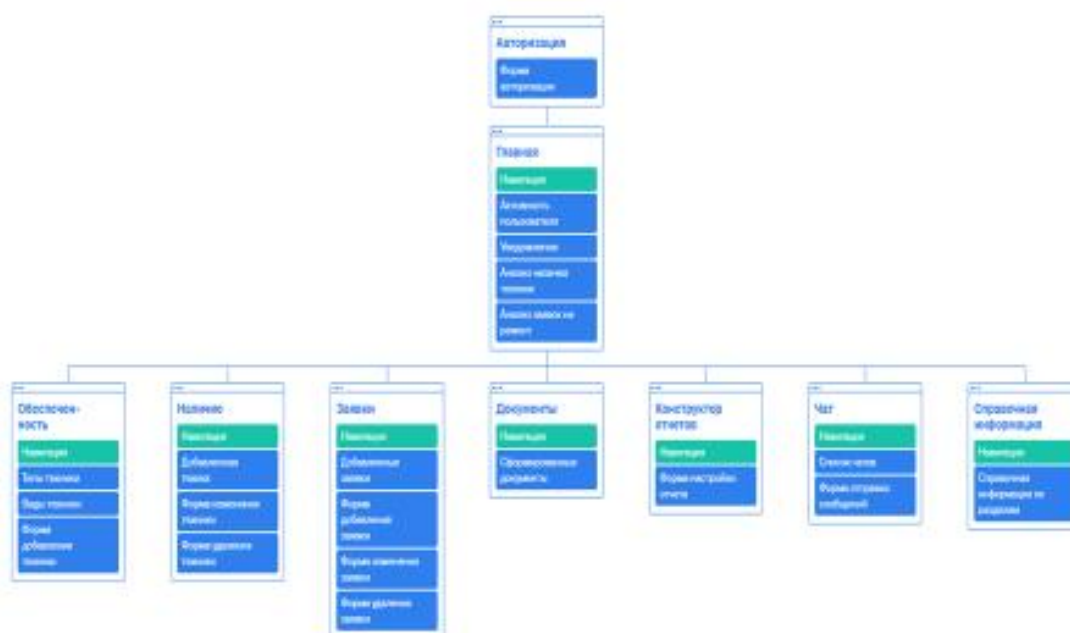
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.07.001 ПЗ

Лист

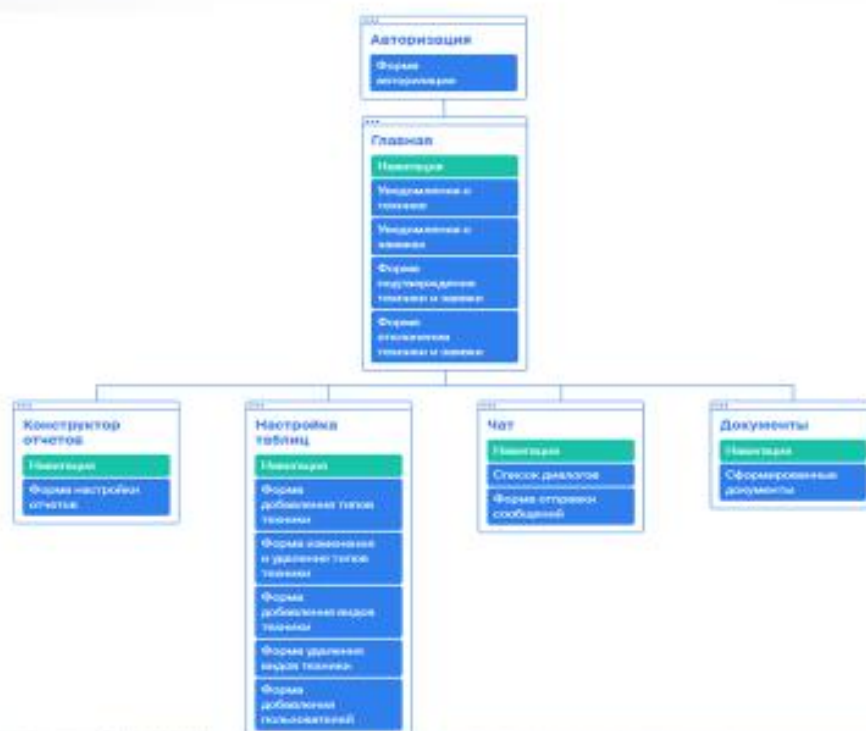
144

Структура страниц пользователя



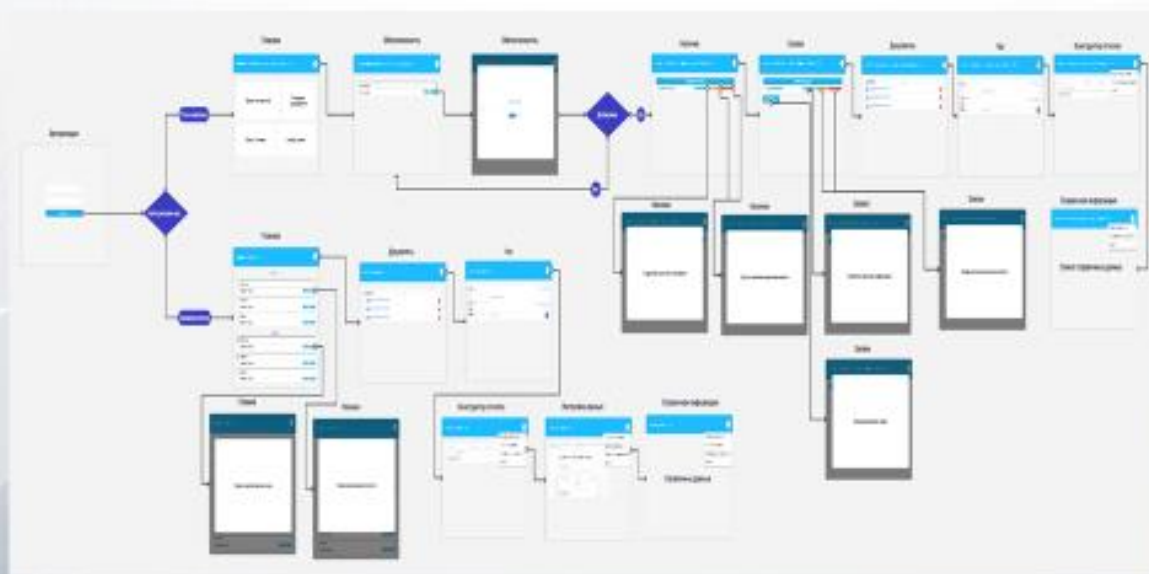
11

Структура страниц администратора



12

Карта экранных форм



13

Структура клиентской части



14

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.07.001 ПЗ

Лист

147

Разработанные экранные формы пользователя



«Авторизация»



«Главная»



«Обеспеченность»



«Наличие»



«Заявки»



«Конструктор отчетов»



«Документы»



«Чат»



«Справочная информация»

15

Разработанные экранные формы администратора



«Авторизация»



«Главная»



«Конструктор отчетов»



«Документы»



«Настройка данных»



«Чат»



«Справочная информация»

16

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090501.07.001 ПЗ

Лист

148