

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа информационных технологий и робототехники  
Направление подготовки 09.04.02 Информационные системы и технологии  
Отделение информационных технологий

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
<b>Разработка программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет</b>

УДК 004.415:004.738.5:651.4/.9

Студент

Группа	ФИО	Подпись	Дата
8ИМ6А	Колесова Олеся Константиновна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Буркатовская Юлия Борисовна	к.ф.-м.н.		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Старикова Екатерина Васильевна	к.ф.н		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД ИШНКБ	Король Ирина Степановна	к.х.н		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Информационные системы и технологии	Марков Николай Григорьевич	д.т.н		

## ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)
<b>Общепрофессиональные компетенции</b>	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
<b>Профессиональные компетенции</b>	
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.
<b>Общекультурные компетенции</b>	
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.

Код результатов	Результат обучения (выпускник должен быть готов)
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа информационных технологий и робототехники  
 Направление, специальность 09.04.02 Информационные системы и технологии  
 Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП

\_\_\_\_\_  
 (Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

Магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ИМ6А	Колесова Олеся Константиновна

Тема работы:

Разработка программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

04.06.2018г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Задача по разработке программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	1) Изучение предметной области; 2) Исследование аналогичных систем; 3) Разработка технического задания на создание системы; 4) Проектирование пользовательского интерфейса; 5) Проектирование архитектуры системы; 6) Проектирование базы данных; 7) Реализация системы; 8) Тестирование системы.
<b>Перечень графического материала</b>	Диаграмма развертывания, физическая модель базы данных, диаграммы деятельности, эскизы пользовательского интерфейса, разработанный пользовательский интерфейс
<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент ОСГН ШБИП Старикова Е.В.
Социальная ответственность	Доцент ОКД ИШНКБ Король И.С.

Раздел на иностранном языке	Доцент ОИЯ ШБИП Комиссарова О.В.
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
Объект и методы исследования (Object and methods of research)	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	29.01.2018
---	------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Буркатовская Ю.Б.	к.ф.-м.н.		29.01.2018

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ6А	Колесова О.К.		29.01.2018

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа информационных технологий и робототехники  
 Направление подготовки 09.04.02 Информационные системы и технологии  
 Уровень образования: магистр  
 Отделение информационных технологий  
 Период выполнения: осенний / весенний семестр 2017/2018 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
 выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

04.06.2018г.

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
05.02.2018 г	Постановка задачи и анализ предметной области	5
12.02.2018 г	Анализ существующих аналогов	5
26.02.2018 г	Составление и утверждение технического задания	15
05.03.2018 г	Подбор и изучение материалов по тематике	10
19.03.2018 г	Проектирование системы	15
30.04.2018 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
21.05.2018 г.	Реализация системы	20
30.05.2018 г.	Социальная ответственность	10
31.05.2018 г.	Обязательное приложение на иностранном языке	5
04.06.2018 г.	Оформление пояснительной записки	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Буркатовская Ю.Б.	к.ф.-м.н.		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Информационные системы и технологии	Марков Н.Г.	д.т.н		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ИМ6А	Колесова Олеся Константиновна

<b>Инженерная школа</b>	Информационных технологий и робототехники	<b>Отделение</b>	Информационных технологий
<b>Уровень образования</b>	Магистратура	<b>Направление/специальность</b>	09.04.02 Информационные системы и технологии

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений и кредитования	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Оценочная карта конкурентных технических решений
2. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоёмкости и построение календарного графика, формирование бюджета
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определение ресурсоэффективности проекта

**Перечень графического материала:**

1. Оценочная карта конкурентных технических решений
2. Матрица SWOT
3. Иерархическая структура работ
4. Календарный план проекта
5. Бюджет проекта
6. Определение ресурсоэффективности проекта

<b>Дата выдачи задания для раздела по линейному графику</b>	12.03.2018
---	------------

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН ШБИП	Старикова Е. В.	к.ф.н		12.03.2018

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ6А	Колесова О.К.		12.03.2018

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8ИМБА		Колесова Олеся Константиновна	
Инженерная школа	Информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	Магистратура	Направление/специальность	09.04.02 Информационные системы и технологии

### Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программный комплекс электронного документооборота. Методика – разработка программного комплекса электронного документооборота. Рабочая зона – кабинет. Область применения – электронный документооборот.
--	--

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<b>1. Производственная безопасность</b> 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	Анализ вредных и опасных факторов, которые может создать объект исследования Анализ вредных и опасных факторов, которые могут возникнуть на производстве при внедрении объекта исследования Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных факторов
<b>2. Экологическая безопасность</b>	Анализ влияния объекта исследования на окружающую среду Анализ влияния процесса эксплуатации объекта на окружающую среду Обоснование мероприятий по защите окружающей среды
<b>3. Безопасность в чрезвычайных ситуациях</b>	Анализ вероятных ЧС, которые может инициировать объект исследований Анализ вероятных ЧС, которые могут возникнуть на производстве при внедрении объекта исследований Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС
<b>4. Правовые и организационные вопросы обеспечения безопасности</b>	Специальные правовые нормы трудового законодательства Организационные мероприятия при компоновке рабочей зоны

Дата выдачи задания для раздела по линейному графику	12.03.2018
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОКД ИШНКБ	Король И.С.	к.х.н		12.03.2018

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМБА	Колесова О.К.		12.03.2018

## РЕФЕРАТ

Выпускная квалификационная работа 127 с., 50 рис., 26 табл., 29 источников, 5 прил.

Ключевые слова: программный комплекс, электронный документооборот, веб-система, СУБД PostgreSQL, язык программирования Python, веб-фреймворк Django, двухэтапная аутентификация, безопасность.

Объектом исследования является задача по разработке программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет.

Цель работы – разработка программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет.

В процессе работы проводились анализ предметной области, исследование существующих аналогов, проектирование и разработка системы. Были изучены двухэтапная аутентификация, СУБД PostgreSQL, язык программирования Python, веб-фреймворк Django.

Степень внедрения: в настоящее время система находится на этапе тестирования и доработки в компании ЭлеСи.

Экономическая эффективность/значимость работы : увеличение отклика заказчик-интегратор в плане передачи необходимой информации, отслеживание жизненного цикла документов.

В будущем планируется осуществление сопровождения системы: взаимодействие с пользователями для консультирования по вопросам эксплуатации системы, сбору требований по модификациям и доработкам системы, а также реализация соответствующих модификаций и доработок.

## ОПРЕДЕЛЕНИЯ

Авторизация – проверка и определение полномочий на выполнение некоторых действий в соответствии с ранее выполненной аутентификацией.

Аутентификация – проверка соответствия субъекта и того, за кого он пытается себя выдать, с помощью некой уникальной информации.

Веб-сервер – сервер, который принимает запросы от пользователя и выдает им ответы.

Многофакторная аутентификация – расширенная аутентификация, метод контроля доступа к компьютеру, в котором пользователю для получения доступа к информации необходимо предъявить более одного «доказательства механизма аутентификации».

Проприетарное ПО – программное обеспечение, являющееся частной собственностью авторов или правообладателей.

Рендеринг – визуализация web-страниц.

Скачивание – сохранение файла на жесткий диск пользовательского устройства.

Статический файл – файл, код которого не выполняется сервером приложения.

Токен (также аппаратный токен, USB-ключ, криптографический токен) – компактное устройство, предназначенное для обеспечения информационной безопасности пользователя, также используется для идентификации его владельца, безопасного удалённого доступа к информационным ресурсам и т. д. Как правило, это физическое устройство, используемое для упрощения аутентификации. Также этот термин может относиться и к программным токенам, которые выдаются пользователю после успешной авторизации и являются ключом для доступа к службам.

Фреймворк – программная платформа, определяющая структуру программной системы, облегчающая разработку и объединение разных компонентов большого программного проекта.

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTTP/HTTPS – протокол прикладного уровня, с помощью которого происходит передача данных.

IIS – проприетарный набор серверов для нескольких служб Интернета от компании Microsoft.

MVC – архитектурное типовое решение для организации приложения.

ORM – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

SCADA – программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления.

URL – единообразный локатор (определитель местонахождения) ресурса.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – База данных;

ОС – Операционная система;

ПО – программное обеспечение;

СУБД – система управления базами данных;

CSS – Cascading Style Sheets, каскадные таблицы стилей;

E-mail – адрес электронной почты;

HTTP/HTTPS – HyperText Transfer Protocol/ HyperText Transfer Protocol  
Secure;

IIS – Internet Information Services;

MVC – Model, View, Controller;

ORM – Object-Relational Mapping, объектно-реляционное отображение;

SCADA – Supervisory Control And Data Acquisition, диспетчерское  
управление и сбор данных;

URL – Uniform Resource Locator.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	16
1 АНАЛИТИЧЕСКИЙ ОБЗОР .....	17
1.1 Обзор существующих систем.....	17
1.1.1 Seafile.....	17
1.1.2 BitTorrent Sync .....	18
1.1.3 NextCloud .....	19
1.1.4 Pydio.....	20
1.1.5 FreeNAS.....	20
1.1.6 ProjectSend.....	21
1.1.7 OpenFiler.....	21
1.2 Сравнительный анализ существующих систем.....	22
1.3 Безопасность данных.....	23
2 ОБЪЕКТ И МЕТОДЫ ИССЛЕДОВАНИЯ .....	26
2.1 Технические требования.....	26
2.1.1 Назначение и цели системы .....	26
2.1.2 Классы пользователей.....	26
2.1.3 Требования к системе .....	26
2.2 Выбор программных средств разработки .....	34
2.2.1 Выбор языка программирования и фреймворка.....	34
2.2.2 Выбор веб-сервера.....	36
2.2.3 Выбор СУБД.....	37
3 ПРОЕКТИРОВАНИЕ .....	39
3.1 Проектирование базы данных.....	39
3.2 Проектирование архитектуры системы.....	42
3.3 Проектирование интерфейса .....	43
4 РЕАЛИЗАЦИЯ .....	44
4.1 Серверная часть .....	44
4.2 Клиентская часть .....	52
5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ .....	57

5.1	Потенциальные потребители результатов исследования.....	57
5.1.1	Анализ конкурентных технических решений .....	58
5.1.2	Технология QuaD .....	60
5.1.3	SWOT-анализ.....	61
5.2	Планирование управления научно-техническим проектом.....	62
5.2.1	Иерархическая структура работ проекта .....	62
5.2.2	Контрольные события проекта .....	62
5.2.3	План проекта.....	63
5.3	Бюджет разработки.....	65
5.3.1	Расчёт материальных затрат.....	65
5.3.2	Основная заработная плата исполнителей темы.....	66
5.3.3	Дополнительная заработная плата исполнителей темы.....	68
5.3.4	Отчисления во внебюджетные фонды .....	68
5.3.5	Накладные расходы.....	69
5.3.6	Формирование бюджета затрат исследовательского проекта.....	69
5.4	Организационная структура проекта.....	70
5.5	Матрица ответственности.....	71
5.6	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	72
6	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	75
6.1	Производственная безопасность.....	76
6.1.1	Анализ вредных и опасных факторов .....	76
6.2	Экологическая безопасность .....	84
6.3	Безопасность в чрезвычайных ситуациях .....	84
6.4	Правовые и организационные вопросы обеспечения безопасности.....	85
6.4.1	Специальные правовые нормы трудового законодательства.....	85
6.4.2	Организационные мероприятия при компоновке рабочей зоны.....	85
	ЗАКЛЮЧЕНИЕ .....	86
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	87
	ПРИЛОЖЕНИЕ А .....	90
	ПРИЛОЖЕНИЕ Б.....	103

ПРИЛОЖЕНИЕ В .....	109
ПРИЛОЖЕНИЕ Г .....	110
ПРИЛОЖЕНИЕ Д .....	116

## ВВЕДЕНИЕ

Документы – это основные информационные ресурсы компании, работа с которыми требует правильной организации. Документы обеспечивают информационную поддержку принятия решений на всех уровнях и сопровождают ведение всех процессов.

При организации работы с документами компании через глобальную сеть интернет, важнейшим требованием является обеспечение информационной безопасности, которую можно обеспечить, используя двухэтапную аутентификацию.

Двухэтапная аутентификация – это метод идентификации пользователя в каком-либо сервисе (как правило, в интернете) при помощи запроса аутентификационных данных двух разных типов, что обеспечивает двухслойную, а значит, более эффективную защиту аккаунта от несанкционированного проникновения. На практике это обычно выглядит так: первый рубеж – это логин и пароль, второй – специальный код, приходящий по SMS или электронной почте. [1]

Целью данной работы является разработка программного комплекса электронного документооборота с повышенным уровнем безопасности в сети Интернет.

Для достижения цели были поставлены следующие задачи:

- 1) Изучение предметной области;
- 2) Исследование аналогичных систем;
- 3) Разработка технического задания на создание системы;
- 4) Проектирование пользовательского интерфейса;
- 5) Проектирование архитектуры системы;
- 6) Проектирование базы данных;
- 7) Реализация и тестирование системы.

# 1 АНАЛИТИЧЕСКИЙ ОБЗОР

## 1.1 Обзор существующих систем

Были найдены и изучены различные системы электронного документооборота, из которых было выбрано 7 наиболее известных и близких по функционалу для более детального рассмотрения.

К ним относятся следующие системы:

- 1) Seafile;
- 2) BitTorrent Sync;
- 3) NextCloud;
- 4) Pydio;
- 5) FreeNAS;
- 6) ProjectSend;
- 7) OpenFiler.

Далее приведены краткие описания каждой.

### 1.1.1 Seafile

Seafile – это облачный сервис для хранения корпоративных файлов с шифрованием на клиентской стороне.

Seafile создан китайскими разработчиками и распространяется в исходных кодах. Он позиционируется как средство синхронизации файлов и совместной работы для команд. Seafile использует центральное хранилище, к которому подключаются клиенты. Серверная часть существует в двух редакциях: Open Source и Professional. Поскольку Seafile позиционируется не только как сервис файловой синхронизации, но и как средство совместной работы, в нем предусмотрены такие инструменты, как встроенная вики, ведение списков задач, общий доступ к файлам через веб, онлайн-просмотр файлов с дискуссиями, управление учетными записями и группами, поддержка LDAP, обмен сообщениями.

Seafile основан на модифицированной под задачи файловой синхронизации модели Git. Основным понятием в Seafile является библиотека,

которая соответствует группе каталогов. В отличие от Git, файлы разделяются на блоки для более эффективной передачи по сети и хранения. Можно не только давать права пользователям и группам на синхронизацию библиотек, но и открывать общий доступ через веб как к отдельным файлам, так и к каталогам с правами только на чтение или и на чтение, и на запись. В качестве сервера баз данных Seafile может использовать SQLite, MySQL, PostgreSQL, веб-серверы Apache и nginx. Воспользоваться Seafile можно и без установки своего сервера – облачный сервис Seacloud, построенный на основе Seafile, в бесплатном тарифном плане предоставляет 1 Гб бесплатного дискового пространства и 5 Гб включенного трафика. Для оценки возможностей, предоставляемых Seafile, можно ознакомиться с демо-версией.

### **1.1.2 BitTorrent Sync**

В BitTorrent Sync используется подход, принципиально отличный от других систем. Синхронизация построена на основе децентрализованного peer-to-peer протокола. Если файл доступен сразу на нескольких устройствах, они могут передавать его одновременно, достигая при этом максимально возможной скорости. Не требуется указывать никаких адресов сервера – устройства с одним и тем же кодом найдут друг друга автоматически. Для этого используется несколько механизмов: поиск в локальной сети с помощью широковещательных пакетов, пиры могут обмениваться друг с другом информацией о других известных им пирах, пир может быть задан статически указанием адреса и порта, может быть использована DHT либо BitTorrent трекер-сервер, который пиры уведомляют о своей доступности и который может быть ими использован для проксирования трафика при невозможности установить прямое соединение.

При передаче файлы шифруются и не сохраняются на каких-либо устройствах, кроме тех, что были авторизованы пользователем. Для взаимной аутентификации устройств используется SRP. Сама компания BitTorrent хотя и имеет доступ к статистике сервиса, но заявляет, что никакие данные пользователей ей принципиально не могут быть доступны.

Поскольку центрального хранилища в BTSync нет, все участники равны, и, если две группы участников некоторое время выйдут из синхронизации, потом будет сложно разобраться в том, какая из версий основная. Синхронизация через HTTP/HTTPS не поддерживается, поэтому далеко не всегда он сможет пройти через сетевые экраны, и в современной защищенной корпоративной среде ему приходится туго. Нет возможности дать общий доступ к отдельному файлу/каталогу через веб. Администрирование большого количества каталогов и устройств затруднено. Невозможно дать доступ для синхронизации к каталогу, находящемуся внутри уже синхронизируемого каталога.[3]

### **1.1.3 NextCloud**

Nextcloud - это набор клиент-серверных программ для создания и использования служб хостинга файлов. Он функционально похож на Dropbox, хотя Nextcloud является открытым исходным кодом, позволяя всем устанавливать и управлять им на частном сервере.

В отличие от проприетарных служб, таких как Dropbox, открытая архитектура позволяет добавлять дополнительные функции к серверу в виде приложений и позволяет пользователю полностью контролировать свои данные.

Оригинальный разработчик ownCloud Фрэнк Карличек разветвил ownCloud и создал Nextcloud, который продолжает активно развиваться Фрэнком и другими членами оригинальной команды ownCloud.

Файлы Nextcloud хранятся в обычных структурах каталогов и могут быть доступны через WebDAV, если это необходимо. Пользовательские файлы зашифровываются во время транзита и могут быть зашифрованы в покое. Пользователи Nextcloud могут управлять календарями (CalDAV), контактами (CardDAV), запланированными задачами и потоковыми медиа (Ampache) с платформы.

С точки зрения администрирования Nextcloud разрешает администрирование пользователей и групп (через OpenID или LDAP). Контент

может использоваться совместно, определяя грамотные разрешения на чтение и запись между пользователями и / или группами. Кроме того, пользователи Nextcloud могут создавать общедоступные URL-адреса при совместном использовании файлов. Также доступна регистрация действий, связанных с файлами, а также запрет доступа на основе правил доступа к файлам.[4]

#### **1.1.4Pydio**

Pydio (ранее известный как AjaXplorer) – бесплатный сервис, позволяющий создать мастер-хранилище файлов (файл-хостинг) в собственном облаке, при этом отличительной чертой Pydio является простота интерфейса. Некоторые пользователи называют его опенсорсной альтернативой dropbox и box.com для предприятия. Pydio управляется с помощью веб-интерфейса и различных приложений и, следовательно, не привязан к конкретной операционной системе. Он способен превратить файловый или специально выделенный сервер в удобное средство обмена информацией между сотрудниками и внешними лицами.

Доступ к файлам осуществляется как с любых компьютеров, так и со смартфонов/планшетов. Помимо обмена файлами или папками Pydio позволяет создавать публичные минисайты для публикации списков документов, поддерживает управление правами и ролями, многоуровневую настройку синхронизации, перетаскивание и модальные окна, встроенный WebDAV доступ к хранилищам и др.

#### **1.1.5FreeNAS**

FreeNAS, пожалуй, одно из наиболее продвинутых решений Open Source для создания NAS-решения, которое способно удовлетворить все нужды предприятий. По сути, это операционная система, она поддерживает протоколы iSCSI и Rsync, при этом может выступать как в качестве сервера, так и в качестве клиента. FreeNAS можно установить на жесткий диск или USB-флэшку. При этом система занимает весь объем носителя, на который устанавливается, независимо от емкости, а все сетевые ресурсы для хранения информации размещаются на других жестких дисках.

FreeNAS имеет смысл использовать, прежде всего, для организации универсального файлохранилища.

### **1.1.6 ProjectSend**

ProjectSend также позволяет хранить предприятиям файлы и документы, но его особенность в том, что он предоставляет доступ к ним сторонним клиентам. ProjectSend упирает на то, что во времена, когда облачные службы не гарантируют сохранность данных, лучше получать доступ к файлам минуя эти службы. Вы можете не только отправлять документы и файлы для клиентов при помощи ProjectSend, но также и ваши клиенты могут загружать ваши файлы. На выбор можно создавать определенные группы клиентов. Загруженные файлы можно находить по описанию или названию при помощи встроенного поиска. ProjectSend также имеет систему шаблонов электронной почты, что упрощает работу с уведомлениями.

### **1.1.7 OpenFiler**

OpenFiler – это решение для создания внутреннего сетевого хранилища, которое может превратить любой x86-совместимый компьютер в устройство, по функциональности похожее на систему NetApp, но базирующееся на открытом коде. Установка выделенного файл-сервера не всегда оправдана. Такой подход потребует значительных финансовых затрат на покупку ПО и оборудования, которые могут стать обузой для небольших организаций. Именно им рекомендуется использование OpenFiler.

Особенностью OpenFiler является работа с внутренними (NAS) и внешними (SAN) дисковыми массивами как с единым фреймворком. В качестве файловой системы можно выбрать любую из поддерживаемых Linux-ядром – именно поэтому OpenFiler можно полностью настроить под конкретные задачи. Реализована синхронная и асинхронная блочная репликация данных при помощи rsync. OpenFiler разрешает развертывать хранилище на базе как SAN, так и NAS, причем как одно, так и другое сможет управляться при помощи простого веб-интерфейса.

Настройка и запуск OpenFiler может быть осуществлена в кратчайшие сроки, для этого не требуется специальных знаний. Дистрибутив имеет встроенную поддержку создания снимков – можно указывать для их создания временные промежутки и размер. Сервис имеет поддержку контроллера домена Windows, а также встроенную поддержку Samba.[5]

## 1.2 Сравнительный анализ существующих систем

После рассмотрения существующих систем в отдельности был проведен их сравнительный анализ.

Ключевыми характеристиками являлись:

- Наличие двухэтапной аутентификации;
- Возможность установки на свой сервер;
- Бесплатность;
- Открытый исходный код;
- Русскоязычный интерфейс.

В таблице 1.2.1 приведен сравнительный анализ перечисленных выше систем. [6]

Таблица 1.2.1 – Сравнительный анализ существующих систем

Название	Наличие русскоязычного интерфейса	Наличие двухэтапной аутентификации	Открытый исходный код	Возможность установки на свой сервер
Seafiler	-	-	+	+
BitTorrent Sync	+	-	-	-
Nextcloud	+	+	+	+
Pydio	+	-	+	+
FreeNAS	-	-	+	+
ProjectSend	-	-	+	+
OpenFiler	-	-	+	+

В результате сравнения систем можно сделать несколько выводов:

Из таблицы 1.2.1 видно, что большинством из перечисленных характеристик обладает система Nextcloud. В остальных не реализовано большее количество требований. Однако практически все системы при предоставлении необходимого функционала являются платными.

Каждый рассмотренный программный продукт, на сегодняшний день, реализует либо только часть из необходимых функций, либо содержит слишком много лишнего функционала. У большинства продуктов отсутствует двухфакторная аутентификация. Кроме того, в некоторых приложениях отсутствует возможность установки на свой сервер.

В связи с этим было решено разработать свою систему, которая будет реализовывать все необходимые функции и удовлетворять всем требованиям.

### **1.3 Безопасность данных**

При организации работы с документами компании через глобальную сеть интернет, важнейшим требованием является обеспечение информационной безопасности, которую можно обеспечить, используя двухэтапную или двухфакторную аутентификацию.

Двухфакторная или двухэтапная аутентификация – это метод идентификации пользователя в каком-либо сервисе (как правило, в Интернете) при помощи запроса аутентификационных данных двух разных типов, что обеспечивает двухслойную, а значит, более эффективную защиту аккаунта от несанкционированного проникновения. На практике это обычно выглядит так: первый рубеж – это логин и пароль, второй – специальный код, приходящий по SMS или электронной почте. Реже второй «слой» защиты запрашивает специальный USB-ключ или биометрические данные пользователя.

Двухэтапную аутентификацию часто называют двухфакторной, не делая особой разницы между понятиями. Двухфакторная аутентификация может быть двухэтапной, но обратное верно не всегда. Граница между этими понятиями очень тонкая, поэтому их часто и не различают.

Двухэтапная аутентификация (Two-Step Verification, 2SV). Вход выполняется в два этапа – например, сначала вы вводите пароль к учетной записи, а потом код из SMS.

Двухфакторная аутентификация (Two Factor Authentication, 2FA). Вход тоже может выполняться в два этапа, но они должны отличаться факторами. Например, сначала вводится пароль, а потом одноразовый пароль, сгенерированный аппаратным токеном. [7]

Проверку пароля, совмещенную с проверкой одноразового кода, присланного на электронную почту нельзя назвать двухфакторной аутентификацией, если почта сама защищена только паролем. Так как такая проверка подтвердит только знание пользователем двух паролей: от приложения/сервиса/сайта и от почты. Другими словами, это однотипные факторы. А двухфакторная аутентификация должна состоять из факторов разных типов. В противопоставление такому примеру - использование смарт карты с обязательным вводом PIN-кода вполне может считаться двухфакторной аутентификацией: первый фактор — владение смарт картой, а второй — знание PIN кода.

Двухэтапная защита не панацея от угона аккаунта, но достаточно надежный барьер, серьезно усложняющий злоумышленникам доступ к чужим данным и в какой-то степени нивелирующий недостатки классической парольной защиты. Ведь у паролей, на которых основано подавляющее большинство авторизационных механизмов в Интернете, есть неизбежные недостатки, которые фактически являются продолжением достоинств: короткие и простые пароли легко запомнить, но также легко подобрать, а длинные и сложные трудно взломать, но и запомнить непросто. По этой причине многие люди используют довольно тривиальные пароли, причем сразу во многих местах. Второй этап в подобных случаях оказывается крайне полезен, поскольку, даже если пароль был скомпрометирован, злоумышленнику придется или раздобыть мобильник жертвы, или угнать ее почтовый ящик.

Этот метод удобен еще и тем, что способен предупреждать хозяина аккаунта о попытке взлома: если на ваш телефон или почту вдруг приходит сообщение с одноразовым кодом при том, что вы никаких попыток логина не предпринимали, значит, вас пытаются взломать – самое время менять оказавшийся ненадежным пароль.

## 2 ОБЪЕКТ И МЕТОДЫ ИССЛЕДОВАНИЯ

### 2.1 Технические требования

#### 2.1.1 Назначение и цели системы

Назначение системы:

Организация личного онлайн кабинета, предназначенного для автоматизации процесса документооборота, как внутри компании ЭлеСи, так и с интеграторами системы SCADA Infinity.

Целями создания системы являются:

- увеличение отклика заказчик-интегратор в плане передачи необходимой информации;
- отслеживание жизненного цикла документов.

#### 2.1.2 Классы пользователей

Доступ к системе должны иметь только авторизованные пользователи, которые разделяются по правам доступа в соответствии с пунктом 2.1.3.3 данного документа.

#### 2.1.3 Требования к системе

##### 2.1.3.1 Группы требований

В таблице 2.1.1 приведены группы требований разрабатываемой системы.

Таблица 2.1.1 – Группы требований

Символ	Группа требований
<b>F</b>	Функциональные требования
<b>I</b>	Требования к интерфейсу пользователя
<b>AR</b>	Требования к правам доступа
<b>TS</b>	Требования к техническому обеспечению
<b>SR</b>	Требования к программному обеспечению

### 2.1.3.2 Функциональные требования

В таблице 2.1.2 приведены функциональные требования разрабатываемой системы.

Таблица 2.1.2 – Функциональные требования

Код требования	Требования	Примечания
<b>F.01</b>	<b>Общие требования</b>	
F.01.01	Должна быть реализована двухэтапная авторизация пользователей	
F.01.02	Должен быть многопользовательский режим работы	
F.01.03	Должно быть реализовано разграничение по правам доступа	
F.01.04	Должна быть возможность работать с документами	добавление документа (со своего компьютера в систему); скачивание документа (в выбранный каталог на компьютер из системы).
F.01.05	Должно быть реализовано оповещение о событиях	
F.01.06	Должно быть реализованы комментарии	
<b>F.02</b>	<b>Требования к документам</b>	
F.02.01	Документы должны находиться в карте документа, которая содержит описание документа и комментарии	
F.02.02	При создании карты документа пользователем должны формироваться следующие данные: – Название; – Комментарий; – Путь до файла.	
F.02.03	При просмотре карты документа, системой должны предоставляться данные: – Формат; – Ссылка URL; – Дата создания карты документа.	
<b>F.03</b>	<b>Требования к оповещению о событиях</b>	
F.03.01	Должны быть реализованы оповещения по e-mail об изменениях в доступных категориях и картах документов в соответствии с правами доступа	

F.03.02	После создания категории или карты документа, всем пользователям, имеющим к ним доступ, должно приходить оповещение на e-mail	Вид оповещения: «В системе «Имя системы» для вас создана новая категория/новая карта документа «Имя категории»/«Имя карты документа»».
F.03.03	После редактирования категории или карты документа, всем пользователям, имеющим к ним доступ, должно приходить оповещение на e-mail	Вид оповещения: «В системе «Имя системы» изменена/изменен категория/карта документа «Имя категории»/«Имя карты документа»».
F.03.04	После создания пользователем нового комментария, всем пользователям, имеющим к нему доступ, должно приходить оповещение на e-mail	Вид оповещения: «В системе «Имя системы» в карте документа «Имя карты документа» новый комментарий».
<b>F.04</b>	<b>Требования к истории карты документа</b>	
F.04.01	В истории карты документа должны отображаться сообщения от сервера, которые создаются в случае редактирования карты документа, добавления или скачивания документа	Вид сообщения: ««Имя пользователя» создал/изменил/скачал документ «Название документа»»
<b>F.05</b>	<b>Требования к комментариям</b>	
F.05.01	Комментарии должны быть реализованы в картах документов	
F.05.02	Ввод комментариев должен осуществляться в специальной области, после нажатия на кнопку «Отправить» в окне с комментариями должна появиться запись	Вид записи: ««Имя пользователя»: «Текст сообщения»»
<b>F.06</b>	<b>Требования к удалению</b>	
F.06.01	Должна быть реализована возможность удаления только карт документов, при наличии соответствующих прав доступа	
F.06.02	Физического удаления карт документов не происходит, они должны оставаться в базе и быть доступны администратору	

### 2.1.3.3 Требования к правам доступа

В таблице 2.1.3 приведены требования к правам доступа разрабатываемой системы.

Таблица 2.1.3 – Требования к правам доступа

Код требования	Требования	Примечания
<b>AR.01</b>	<b>Общие требования</b>	
AR.01.01	Доступ к системе должны иметь только внесенные в базу пользователи	
AR.01.02	Администратор системы должен обладать полным доступом ко всем функциям системы и управлять правами доступа других пользователей	
<b>AR.02</b>	<b>Требования к авторизации</b>	
AR.02.01	Должна быть реализована двухэтапная авторизация пользователей	
AR.02.02	После успешного ввода пользователем логина и пароля, ему на почту должен прийти сгенерированный системой код доступа	
AR.02.03	Код для авторизации должен генерироваться на 5 минут	
AR.02.04	При неверном вводе пользователем данных для авторизации три раза, возможность входа должна заблокироваться для данного пользователя на 15 минут	
AR.02.05	При неверном вводе пользователем данных для авторизации девять раз, возможность входа должна заблокироваться для данного пользователя на 24 часа	
<b>AR.03</b>	<b>Требования к правам доступа к группам пользователей</b>	
AR.03.01	Создавать группы пользователей должен иметь право только администратор	Должна быть возможность ввода названия, и выбора прав доступа
AR.03.02	Редактировать группы пользователей должен иметь право только администратор	Должна быть возможность редактирования названия и прав доступа

<b>Код требования</b>	<b>Требования</b>	<b>Примечания</b>
<b>AR.04</b>	<b>Требования к правам доступа к пользователям</b>	
AR.04.01	Создавать пользователей должен иметь право только администратор	Должна быть возможность ввода полей: –Имя; –E-mail; –Пароль. Должна быть возможность выбора: –Группа пользователей.
AR.04.02	Редактировать пользователей должен иметь право только администратор	
AR.04.03	Пользователи должны иметь доступ к редактированию только своего пароля	
<b>AR.05</b>	<b>Требования к правам доступа к категориям</b>	
AR.05.01	Должна быть реализовано правило доступа создания категорий	Должна быть возможность ввода полей: –Название; –Комментарий. Должна быть возможность выбора: –Группы пользователей, которые имеют доступ
AR.05.02	Должно быть реализовано правило доступа редактирования всех доступных категорий	
AR.05.03	Должно быть реализовано правило доступа редактирования только своих, созданных, категорий	
<b>AR.06</b>	<b>Требования к правам доступа к документам</b>	
AR.06.01	Должна быть реализовано правило доступа создания карт документов	Должна быть возможность ввода полей: –Название; –Комментарий. Должна быть возможность загрузки документа

<b>Код требования</b>	<b>Требования</b>	<b>Примечания</b>
AR.06.02	Должно быть реализовано правило доступа редактирования всех доступных карт документов	
AR.06.03	Должно быть реализовано правило доступа редактирования только своих, созданных, карт документов	
AR.06.04	Должно быть реализовано правило доступа скачивания всех доступных документов	
AR.06.05	Должно быть реализовано правило доступа скачивания только своих, созданных, документов	
AR.06.06	Должно быть реализовано правило доступа удаления только своих, созданных, карт документов	
<b>AR.07</b>	<b>Требования к правам доступа к истории документов</b>	
AR.07.01	Администратору должна быть доступна история всех документов	
AR.07.02	Пользователям может быть доступна история только своих документов	Информация: – Имя пользователя; – Операция; – Название документа; – Дата.
<b>AR.08</b>	<b>Требования к правам доступа к комментариям к карте документа</b>	
AR.08.01	Должно быть реализовано правило доступа отправки комментариев	
AR.08.02	Должно быть реализовано правило доступа только просмотра комментариев	

### 2.1.3.4 Требования к интерфейсу пользователя

В таблице 2.1.4 приведены требования к интерфейсу разрабатываемой системы.

Таблица 2.1.4 – Требования к интерфейсу

<b>Код требования</b>	<b>Требования</b>	<b>Примечания</b>
<b>I.01</b>	<b>Общие требования</b>	
I.01.01	Язык интерфейса системы – русский	
I.01.02	Не должно быть кнопок без имени или не помеченных специальной иконкой	

<b>Код требования</b>	<b>Требования</b>	<b>Примечания</b>
I.01.03	Категории и карты документов должны быть представлены в виде дерева структуры, вложенность категорий в котором не ограничена	
<b>I.02</b>	<b>Требования к интерфейсам</b>	
I.02.01	Интерфейс начальной страницы должен выглядеть согласно рисунку Б.1 в приложении Б Требования к интерфейсу пользователя	Пользователь должен ввести свой логин и пароль и нажать кнопку «Войти»
I.02.02	Интерфейс страницы ввода кода должен выглядеть согласно рисунку Б.2 в приложении Б Требования к интерфейсу пользователя	Пользователь должен ввести код, который пришел ему на почту и нажать кнопку «Войти»
I.02.03	Окно с ошибками должно выглядеть согласно рисунку Б.3 в приложении Б Требования к интерфейсу пользователя	
I.02.04	Окно с сообщением о неработоспособности кода должно выглядеть согласно рисунку Б.4 в приложении Б Требования к интерфейсу пользователя	
I.02.05	Интерфейс меню пользователя должен выглядеть согласно рисунку Б.5 в приложении Б Требования к интерфейсу пользователя	
I.02.06	Интерфейс страницы смены пароля должен выглядеть согласно рисунку Б.6 в приложении Б Требования к интерфейсу пользователя	
I.02.07	Интерфейс страницы просмотра категории должен выглядеть согласно рисунку Б.7 в приложении Б Требования к интерфейсу пользователя	
I.02.08	Интерфейс страницы просмотра карты документа должен выглядеть согласно рисунку Б.8 в приложении Б Требования к интерфейсу пользователя	
I.02.9	Интерфейс страницы создания категории должен выглядеть согласно рисунку Б.9 в приложении Б Требования к интерфейсу пользователя	
I.02.10	Интерфейс страницы создания карты документа должен выглядеть согласно рисунку Б.10 в приложении Б Требования к интерфейсу пользователя	Для прикрепления документа необходимо нажать кнопку «Выберите файл», выбрать нужный файл и нажать кнопку «Открыть»

Код требования	Требования	Примечания
I.02.11	Интерфейс страницы редактирования категории должен выглядеть согласно рисунку Б.11 в приложении Б Требования к интерфейсу пользователя	
I.02.12	Интерфейс страницы редактирования карты документа должен выглядеть согласно рисунку Б.12 в приложении Б Требования к интерфейсу пользователя	

### 2.1.3.5 Требования к техническому обеспечению

В таблице 2.1.5 приведены требования к техническому обеспечению разрабатываемой системы.

Таблица 2.1.5 – Требования к техническому обеспечению

Код требования	Требования	Примечания
TS.01	Требования к серверной части: <ul style="list-style-type: none"> <li>– Windows Server 2012 R2;</li> <li>– частота процессора – от 2 ГГц;</li> <li>– объем оперативной памяти – от 8 Гбайт;</li> <li>– PostgreSQL, версия 10.1 и выше;</li> <li>– Python, версия 3.6.4 и выше;</li> <li>– объем свободного дискового пространства на системном диске 1 Тбайт.</li> </ul>	

### 2.1.3.6 Требования к программному обеспечению

В таблице 2.1.6 приведены требования к программному обеспечению разрабатываемой системы.

Таблица 2.1.6 – Требования к программному обеспечению

Код требования	Требования	Примечания
SR.01	<b>Требования к клиентской части</b>	
SR.01.01	Должна функционировать под управлением следующих ОС: <ul style="list-style-type: none"> <li>– Windows версии не ниже 7;</li> <li>– Linux с версией ядра не ниже 3.2;</li> <li>– Android версии не ниже 4.4;</li> <li>– Mac OS;</li> <li>– Apple iOS версии не ниже 8.x.</li> </ul>	
SR.01.02	Не должна требовать дополнительных программных установок на стороне клиента	

Код требования	Требования	Примечания
SR.01.03	Должна функционировать в следующих браузерах с версиями не ниже указанных: <ul style="list-style-type: none"> <li>– Safari, версия 11 и выше;</li> <li>– Google Chrome, версия 62 и выше;</li> <li>– Internet Explorer, версия 11 и выше;</li> <li>– Microsoft Edge, версия 40 и выше;</li> <li>– Mozilla Firefox, версия 57 и выше;</li> <li>– Opera, версия 49 и выше;</li> <li>– iOS Safari, версия 8.4 и выше;</li> <li>– Android Browser, версия 4.4 и выше</li> </ul>	
<b>SR.02</b>	<b>Требования к серверной части</b>	
SR.02.01	Должна функционировать под управлением ОС Windows версии не ниже 7, на веб-сервере Internet Information Services версия 10 и выше, с установлением защищенного соединения по протоколу HTTPS	

## 2.2 Выбор программных средств разработки

Выбор программных средств разработки является основным этапом при разработке. Каждое программное средство предназначено для решения определенных задач. В данном случае необходимо выполнить следующее:

- 1) Выбор языка программирования и фреймворка;
- 2) Выбор веб-сервера;
- 3) Выбор СУБД.

### 2.2.1 Выбор языка программирования и фреймворка

Для разработки веб-приложения можно использовать различные языки программирования и фреймворки.

Обычно реализация веб-приложения состоит из двух частей – серверной и клиентской. Клиентская часть обычно реализуется с помощью средств HTML, CSS, JavaScript. Серверная часть реализуется с помощью различных языков программирования – Java, C#, PHP, Python, Ruby и т.д.

Наиболее популярными являются:

- PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений;

- Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода;

- Ruby – динамический, рефлексивный, интерпретируемый высокоуровневый язык программирования. [8]

Для каждого из описанных языков существуют фреймворки для построения веб-приложений: Laravel/Symfony/Yii2 для PHP, Django/Flask для Python, Ruby on Rails для Ruby. Данные фреймворки предоставляют набор готовых функций и API для взаимодействия с ними, а также диктуют правила построения архитектуры приложения.

С помощью Django и Python можно создать веб приложение, работающее как с файлами, так и с HTTP-запросами. Язык является интерпретируемым и свободно распространяемым. При этом данный язык поддерживает компиляцию в byte-код, что позволяет увеличить скорость выполнения.

Достоинства:

- простота разработки – исходный код имеет наглядную структуру, а благодаря отсутствию необходимости описывать переменные перед их инициализацией сохраняется семантика;

- кроссплатформенность интерпретатора – данная возможность позволяет разрабатывать и разворачивать веб-приложения на рабочей станции или сервере с любой операционной системой;

- большое количество готовых библиотек от сторонних разработчиков, позволяющих ускорить процесс разработки приложений;

- поскольку каждый Python-проект желательно запускать в виртуальном окружении (создаваемым приложением virtualenv), отсутствует конфликт версий библиотек и приложений, а код является изолированным на уровне операционной системы.

Фреймворк Django реализует шаблон MVT (Model, View, Template – интерпретацию шаблона MVC, где в роли контроллера выступает View, а в

роли представления - Template) и предоставляет богатый API для реализации как компонентов, необходимых для работы (моделей, представлений и шаблонов), так и собственных библиотек функций и классов.

В состав данного фреймворка входит большое количество разнообразных модулей, позволяющих создавать современные веб-приложения.

Достоинства решения на Python/Django:

- бесплатное распространение;
- кроссплатформенность;
- расширяемость;
- высокая производительность по сравнению с другими интерпретируемыми языками программирования;
- абстрагирование от конкретной СУБД;
- высокая скорость разработки.

Недостатки данного решения:

- непривычный синтаксис;
- необходимость конфигурации каждого узла системы.

### **2.2.2 Выбор веб-сервера**

Для работы с Python/Django могут быть использованы следующие веб-серверы:

- Apache с модулем mod\_python;
- Nginx с через приложение uWSGI;
- Gunicorn с Nginx в качестве reverse-Proxy;
- Microsoft IIS.

Был выбран Microsoft IIS, так как это является одним из требований к системе.

### 2.2.3 Выбор СУБД

Django поддерживает четыре системы управления базами данных:

- PostgreSQL;
- SQLite 3;
- MySQL;
- Oracle.

Используемая СУБД должна отвечать таким требованиям, как быстрота, простота, эффективность и надежность. По умолчанию в настройках Django указано использование SQLite. Django изначально проектировалась и разрабатывалась для работы с СУБД PostgreSQL.

В таблице 2.2.1 приведен сравнительный анализ данных СУБД. [10]

Таблица 2.2.1 – Сводная таблица характеристик СУБД

	PostgreSQL	MySQL	Oracle	SQLite
Стоимость	бесплатно	бесплатно	платно	бесплатно
Безопасность	Высокая	Высокая	Высокая	Низкая
Простота использования	Средняя	Высокая	Средняя	Высокая
Простота использования с Django Framework	Высокая	Средняя	Средняя	Высокая
Ограничения на работу с Django ORM	Нет	Есть	Есть	Есть
Выводы	+	-	-	-

PostgreSQL – свободная объектно-реляционная система управления базами данных, основанная на языке SQL – языке структурированных запросов, являющемся стандартом для разработки реляционных БД. Отличительной особенностью реляционных баз данных являются отношения. Внутренние механизмы оптимизации отношений позволяют добиться высокой скорости обработки. Также отношения позволяют связывать данные из нескольких таблиц при выполнении запросов.

К преимуществам данной СУБД относятся:

- высокопроизводительные и надежные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования;

- наследование;
- легкая расширяемость;
- многоверсионность – одновременная модификация базы данных несколькими пользователями с помощью механизма Multiversion Concurrency Control, благодаря чему отпадает нужда в блокировании чтения [12];
- возможность создания пользовательских типов данных.

PostgreSQL – наиболее подходящее решение для создания вебприложений и веб-сервисов любой сложности с помощью фреймворка Django, которые ориентированы на получение, хранение и обработку информации. Данное решение позволяет создавать динамически генерируемые HTML представления, то есть включающими информацию, изменяемую в реальном времени.

Так же в состав Django входит библиотека Django ORM, предоставляющая интерфейс для работы с СУБД. С помощью данной библиотеки, реализующей шаблон проектирования «Active Record», возможно создание запросов к БД в виде использования методов классов, унаследованных от класса Model. Также данная библиотека оптимизирует текст запроса и выполняет его только в момент обращения к данным, что позволяет не производить промежуточных запросов и тем самым снизить нагрузку на СУБД.

## 3 ПРОЕКТИРОВАНИЕ

### 3.1 Проектирование базы данных

Проектирование структуры БД является одной из основных и очень трудозатратных задач при разработке ПО, работающего с данными.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, называемыми моделями, и по ней генерируется схема базы данных. ORM позволяет работать с базой данных, без непосредственного использования SQL. [14]

А с помощью миграций происходит преобразование базы данных в соответствии с моделями.

Было создано 14 моделей:

UserInfo – для отображения дополнительной информации о пользователе (телефон, примечание);

Access – для отображения списка уровней доступа;

AccessHistory – для отображения истории доступа;

SignIn – для хранения ключей доступа;

UserBan – для блокировки пользователей по логину;

IPBan – для блокировки пользователей по ip-адресу;

SessionBan – для блокировки пользователей по сессии;

Action – для отображения списка действий над категориями и карточками документов;

Categories – для отображения списка категорий;

CategoriesHistory – для отображения истории категорий;

CategoriesUserGroups – для отображения связывания категорий и групп пользователей;

DocumentCards – для отображения списка карточек документов;

DocumentsHistory – для отображения истории карточек документов;

Posts – для отображения сообщений.

Каждая модель представлена классом, унаследованным от `django.db.models.Model`. Каждая модель содержит несколько атрибутов, каждый из которых отображает поле в таблице базы данных.

Каждое поле представлено экземпляром класса `Field`, например, `CharField` для текстовых полей и `DateTimeField` для полей даты и времени. Это указывает Django какие типы данных хранят эти поля.

Через класс `Meta` определяются дополнительные настройки для модели. Например, название таблицы в базе данных для этой модели (`db_table`), читабельное название модели (`verbose_name`), права доступа для этой модели (`permissions`).

Модель также может иметь методы. Минимально в каждой модели определяется стандартный метод класса для Python `__str__` (), чтобы вернуть удобочитаемую строку для каждого объекта. Эта строка используется для представления отдельных записей на сайте администрирования. Часто это возвращает поле названия или имени из модели.

Для примера, модель `DocumentCards` выглядит следующим образом:

```
def directory_path(instance, filename):
    category = Categories.objects.get(category_id=instance.category_id)
    parent_id = category.parent_id
    while parent_id is not None:
        category = Categories.objects.get(category_id=parent_id)
        parent_id = category.parent_id
    return '{0}/{1}'.format(category.name, filename)

class DocumentCards(models.Model):
    document_card_id = models.BigAutoField(primary_key=True, verbose_name='Id')
    name = models.TextField()
    comment = models.TextField(blank=True, null=True)
    file = models.FileField(upload_to=directory_path)
    category = models.ForeignKey(Categories, models.DO_NOTHING, related_name='docs')
    user = models.ForeignKey(User, models.DO_NOTHING)
    delete = models.BooleanField("Статус удаления", default=False)

class Meta:
    managed = False
    db_table = 'document_cards'
    verbose_name = 'Карта документов'
    verbose_name_plural = "Карты документов"
```

```

default_permissions = ()
permissions = (
    ("view_document_card", "Просмотр карт документов"),
    ("add_document", "Добавление карт документов"),
    ("change_own_document", "Редактирование своих карт документов"),
    ("change_all_document", "Редактирование всех карт документов"),
    ("download_own_document", "Скачивание своих документов"),
    ("download_all_document", "Скачивание всех документов"),
    ("delete_document", "Удаление своих карт документов"),
)

```

```

def __str__(self):
    return self.name

```

Модель содержит два текстовых поля, это название и комментарий для карточки документа, так же используется поле `document_card_id` для задания первичного ключа. Поля `category` и `user` являются внешними ключами на таблицы `Categories` и `User`, для отображения категории, в которой создана данная карточка, а также пользователя, который ее создал. Поле `delete` используется для пометки удалена ли карточка. Поле `file` используется для загрузки файла.

Атрибут `upload_to` позволяет указать каталог и название файла при его сохранении. В данном случае используется функция `directory_path`, которая будет вызвана для получения пути к загруженному файлу, включая имя файла. Вызываемый объект должен принимать два обязательных аргумента, и возвращать путь в стиле Unix (с прямыми слэшами), который будет передан в систему хранения файлов. Два аргумента это `instance` – объект, для которого сохраняется текущий файл, и `filename` – оригинальное имя файла.

После создания моделей используются миграции и для сохранения изменений моделей (и структуры базы данных) – это просто файлы на диске.

В приложении В приведена полученная схема базы данных. Таблицы `auth_user`, `auth_group`, `auth_user_groups`, `auth_group_permissions`, `auth_permission`, `auth_user_user_permissions`, `django_content_type`, `django_admin_log`, `admin_tools_menu_bookmark` и `admin_tools_dashboard_preferences` созданы автоматически Django и отвечают за аутентификацию пользователей и за работу панели администратора.

### 3.2 Проектирование архитектуры системы

Один из основных принципов фреймворка Django – DRY (don't repeat yourself). Веб-системы на Django строятся из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из заметных архитектурных отличий этого фреймворка от некоторых других. Также, в отличие от многих других фреймворков, обработчики URL в Django конфигурируются явно (при помощи регулярных выражений), а не автоматически задаются из структуры контроллеров.

Архитектура Django похожа на «Модель-Представление-Контроллер» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление (View), а презентационная логика Представления реализуется в Django уровнем Шаблонов (Templates). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).

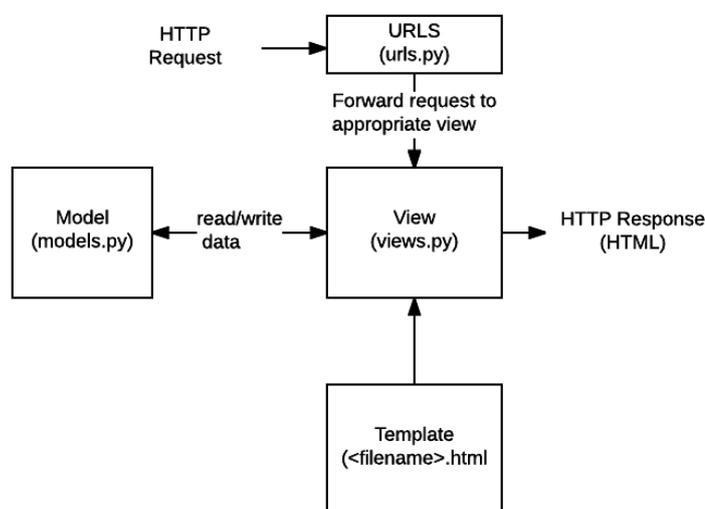


Рисунок 3.2.1 – Взаимодействие MTV

Первоначально разработка Django велась для обеспечения более удобной работы с новостными ресурсами, что достаточно сильно отразилось на архитектуре: фреймворк предоставляет ряд средств, которые помогают в быстрой разработке веб-сайтов информационного характера. Например, разработчику не требуется создавать контроллеры и страницы для административной части сайта, в Django есть встроенное приложение для

управления содержимым, которое можно включить в любой сайт, сделанный на Django, и которое может управлять сразу несколькими сайтами на одном сервере. [16]

Разрабатываемая система использует клиент-серверную технологию. В качестве клиента выступает браузер пользователя, в качестве сервера – совокупность программно-аппаратных средств, необходимых для работы системы (веб-сервер Microsoft IIS, СУБД PostgreSQL и веб-фреймворк Django на языке программирования Python).

Диаграмма развертывания представлена на рисунке 3.2.2.

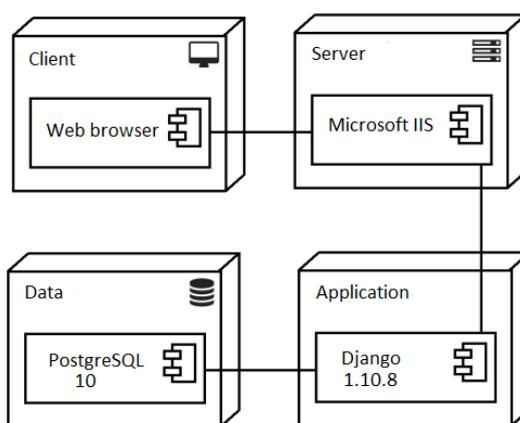


Рисунок 3.2.2 – Диаграмма развертывания

Диаграмма развёртывания описывает физическое развёртывание компонентов на узлах и указывает, какие аппаратные компоненты существуют и какие программные компоненты работают на каждом узле.

Диаграмма развертывания позволяет:

- определить и описать взаимосвязь компонентов системы по её физическим узлам;
- указать физическую связь между узлами системы на стадии её применения.

### 3.3 Проектирование интерфейса

Проектирование интерфейса производилось на основе требований к системе, на этапе составления технического задания. Эскизы интерфейса представлены в приложении Б.

## 4 РЕАЛИЗАЦИЯ

### 4.1 Серверная часть

Серверная логика веб-системы написана на языке программирования Python версии 3.5. За основу был взят веб-фреймворк Django версии 1.10.8.

Для удобной работы с веб-фреймворком Django была использована интегрированная среда разработки для языка программирования Python – PyCharm. PyCharm дает возможность производить анализ кода, а также дает удобный, современный графический отладчик, инструмент для запуска тестовых сценариев и что немаловажно поддерживает веб-разработку на веб-фреймворке Django.

На рисунке 4.1.1 представлена структура проекта.

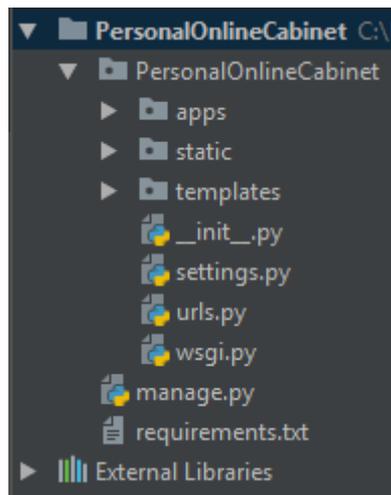


Рисунок 4.1.1 – Структура проекта

Изначально при создании проекта создаются несколько стандартных файлов:

- Внешний каталог PersonalOnlineCabinet – это просто контейнер для проекта.
- manage.py – скрипт, который позволяет взаимодействовать с проектом Django.
- Внутренний каталог PersonalOnlineCabinet – это пакет Python проекта.

- `__init__.py` – пустой файл, который указывает Python, что текущий каталог является пакетом Python.

- `settings.py` – настройки/конфигурация проекта.

- `urls.py` – конфигурация URL-ов для проекта Django. Это своего рода «содержание» всех Django-сайтов. Обработчики URL в Django настраиваются явно при помощи регулярных выражений.

- `wsgi.py` – точка входа проекта для WSGI-совместимых веб-серверов. Основной платформой для развертывания Django является WSGI, это фактически стандарт для веб-серверов и приложений на Python.

`requirements.txt` – содержит список команд для `pip`, который устанавливает необходимые версии зависимых пакетов.

`External Libraries` – это список подключенных библиотек.

Папки `static` и `templates` отвечают за внешний вид системы.

Папка `apps` содержит все приложения.

Проект построенный на веб-фреймворке Django состоит из одного или нескольких так называемых приложений. Каждое подобное приложение создается отчуждаемыми и подключаемыми, что позволяет использовать данные `application` в нескольких проектах. Это основная особенность архитектуры веб-фреймворка Django от его конкурентов.

В результате разработки было создано три приложения: `authentication` для реализации авторизации, `categories` для реализации работы с категориями и `document_cards` для реализации работы с картами документов.

На рисунке 4.1.2 приведена структура приложения authentication, остальные приложения состоят из аналогичных компонентов.

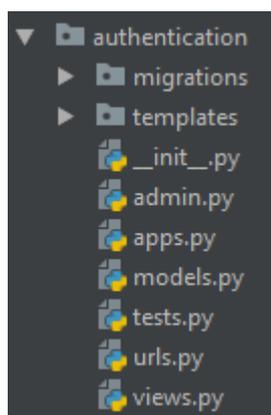


Рисунок 4.1.2 – Структура приложения authentication

Приложение содержит следующие стандартные компоненты:

- migrations – каталог с файлами с миграциями;

Django использует миграции для переноса изменений в моделях (добавление поля, удаление модели и т.д.) на структуру базы данных.

Следует рассматривать миграции, как систему контроля версий для базы данных. Команда makemigrations отвечает за сохранение состояния моделей в файле миграции – аналог коммита, а migrate отвечает за их применение к базе данных.

- templates – каталог с шаблонами для данного приложения;
- \_\_init\_\_.py – пустой файл, который указывает Python, что текущий каталог является пакетом Python;
- admin.py – здесь хранятся настройки стандартной административной панели;
- apps.py – для настройки некоторых атрибутов приложения;
- models.py – файл, в котором хранятся модели приложения;
- test.py – файл для юнит-тестов;
- urls.py – конфигурация URL-ов для проекта Django;
- views.py – файл для хранения представлений.[17]

Представление – это “тип” страниц вашего приложения, которое является функцией для обработки запроса и использует шаблон для генерации страницы.[18]

Представления являются основой серверной части системы и отвечают за:

- любую бизнес-логику;
- контекстный словарь с данными для шаблона;
- рендеринг шаблона с контекстом;
- объект отклика, который пропускает отрендеренные результаты обратно в среду разработки.

Приложение authentication включает пять представлений:

- login – отвечает за аутентификацию пользователя и генерацию проверочного кода активации;
- token – отвечает за второй этап аутентификации, проверку введенного кода, и авторизацию пользователя на сайте;
- logout – отвечает за выход пользователя из системы;
- change\_password – отвечает за смену пароля пользователем;
- change\_password\_done – отвечает за вывод страницы об успешной смене пароля.

На рисунке 4.1.3 представлена диаграмма деятельности для представления login, на рисунке 4.1.4 представлена диаграмма деятельности для представления token.

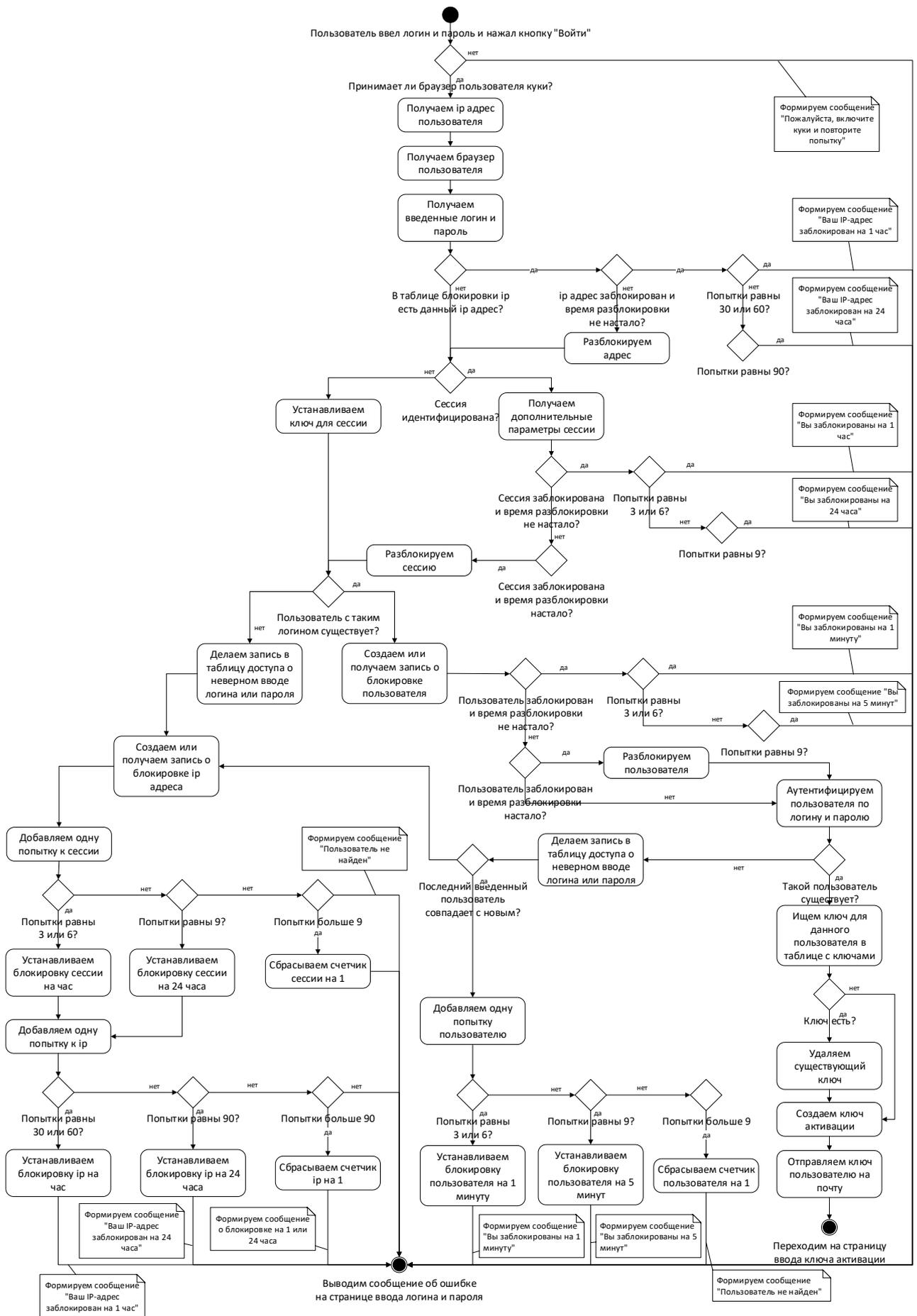


Рисунок 4.1.3 – Диаграмма деятельности для представления login



Ниже приведено представление `create_category`.

```
def create_category(request, category_id):
    if not request.user.is_authenticated():
        return redirect('/')
    else:
        if request.method == 'POST':
            name = request.POST.get('name', '')
            comment = request.POST.get('comment', '')
            category = Categories.objects.create(name=name, parent_id=category_id,
comment=comment)
            category.save()
            groups = request.POST.getlist('groups')
            for groups in groups:
                group = Group.objects.get(name=groups)
                categorygroups =
CategoriesUserGroups.objects.create(category_id=category.category_id, user_group_id=group.id)
                categorygroups.save()
            user = auth.get_user(request).id
            _id = category.category_id
            history = CategoriesHistory.objects.create(category_id=_id, user_id=user, action_id=1,
date=datetime.now(), name=name, comment=comment)
            history.save()
            for user in User.objects.all():
                if user.has_perm('categories.view_categories', category):
                    email = user.email
                    send_mail('Оповещение из «Личный кабинет ЭлеСи»', 'В системе «Личный
кабинет» создана новая категория' + ' «' + category.name + '». Ссылка:
http://217.18.157.5:8001/categories/%s' % category_id, 'kolok@elesy.ru', [email])
                    return HttpResponseRedirect('/categories/%s' % category_id)
            else:
                return render(request, "create_category.html",
                    {
                        'category': Categories.objects.filter(category_id=category_id),
                        'categories': Categories.objects.all(),
                        'user': auth.get_user(request),
                    })
```

В данном представлении сперва проверяется аутентифицирован ли пользователь, после чего получаются данные из запроса и создаются записи в таблицах `Categories` (добавляется новая категория), `CategoriesUserGroups` (добавляются выбранные группы для данной категории) и `CategoriesHistory` (добавляется запись о создании категории).

Приложение `document_cards` включает пять представлений:

- `document_card` – отвечает за вывод информации о карте документа;
- `create_document` – отвечает за создание карты документа;
- `edit_document_card` – отвечает за редактирование карты документа;

- delete\_document\_card – отвечает за удаление карты документа;
- addpost – отвечает за создание комментария.

Ниже приведено представление create\_category.

```
def edit_document_card(request, document_card_id):
    if not request.user.is_authenticated():
        return redirect('/')
    else:
        instance = get_object_or_404(DocumentCards, document_card_id=document_card_id)
        if request.method == "POST":
            form = DocumentForm(request.POST, request.FILES, instance=instance)
            if form.is_valid():
                instance = form.save()
                instance.name = request.POST.get('name', '')
                instance.comment = request.POST.get('comment', '')
                instance.save()
                user = auth.get_user(request).id
                _id = instance.document_card_id
                history = DocumentsHistory.objects.create(document_card_id=_id, user_id=user,
action_id=2, date=datetime.now(), name=instance.name, comment=instance.comment,
file=basename(instance.file.name))
                history.save()
                category = Categories.objects.get(category_id= instance.category_id)
                for user in User.objects.all():
                    if user.has_perm('document_cards.view_document_cards', category):
                        email = user.email
                        send_mail('Оповещение из «Личный кабинет ЭлеСи», 'В системе «Личный
кабинет» изменен документ' + ' «' + instance.name + '». Ссылка:
http://217.18.157.5:8001/document_cards/%s' % document_card_id, 'kolok@elesy.ru', [email])
                        return HttpResponseRedirect('/document_cards/%s' % document_card_id)
            else:
                form = DocumentForm(instance=instance)
                return render(request, "edit_document_card.html",
                    {
                        'form': form,
                        'categories': Categories.objects.all(),
                        'user': auth.get_user(request),
                        'date_create': DocumentsHistory.objects.filter(action_id=1),
                        'this': instance,
                    })
```

В данном представлении сперва проверяется аутентифицирован ли пользователь, после чего получается существующий экземпляр модели и данные из запроса и изменения сохраняются. После чего создается запись в таблице DocumentsHistory о изменении карты документа.

## 4.2 Клиентская часть

Клиентом разрабатываемой системы является браузер. Клиентская часть веб-системы написана с использованием HTML, CSS и мультипарадигменным языком программирования JavaScript. В качестве основного фреймворка для реализации интерфейса представлений был использован Bootstrap 3, который предоставляет разработчикам коллекцию инструментов для создания веб-страниц.

За клиентскую часть отвечают шаблоны, которые предоставляют удобный для дизайнера синтаксис для управления отображением информации пользователю.

Шаблоны содержат статический HTML и динамические данные, рендеринг которых описан специальным синтаксисом.

Django предоставляет бэкенд для собственной системы шаблонов, которая называется – язык шаблонов Django (Django template language, DTL).

Django предоставляет стандартный API для загрузки и рендеринга шаблонов, независимо от используемого бэкенда. Загрузка включает в себя поиск шаблона по названию и предварительную обработку, обычно выполняется загрузка шаблона в память. Рендеринг означает передачу данных контекста в шаблон и возвращение строки с результатом.

На рисунке 4.2.1 представлена структура общих папок static, media, templates, отвечающих за внешний вид системы.

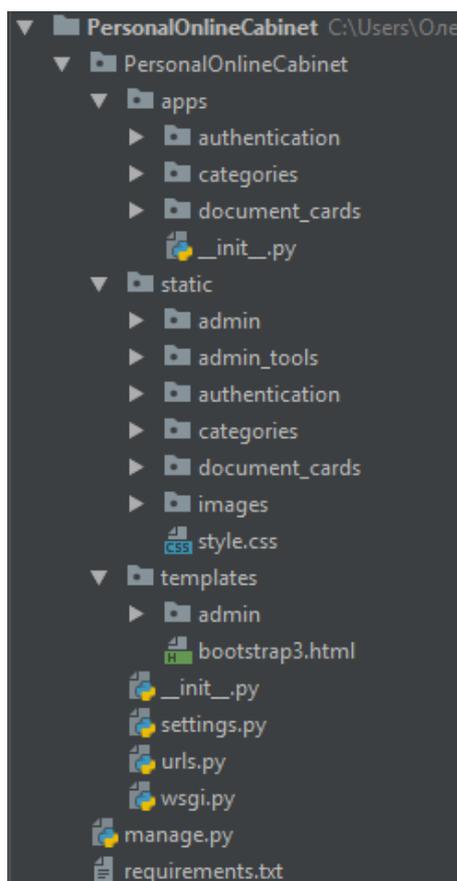


Рисунок 4.2.1 – Структура папок static, media, templates

Папка static содержит STATIC файлы, это файлы, прописанные в коде или шаблонах.

Файлы MEDIA, которые появляются в результате работы приложения находятся в папке media на сервере.

В static находятся папки для каждого приложения, содержащие CSS и Javascript файлы.

В общей папке templates содержатся шаблон для переопределения шаблона панели администратора, а также базовый шаблон bootstrap3.html задающий структуру и основные стили для всей системы.

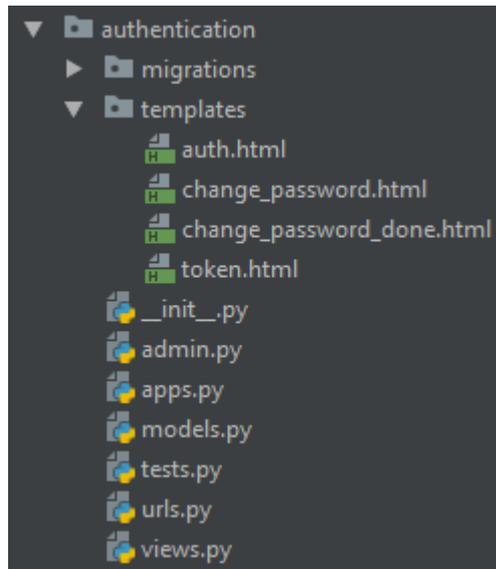


Рисунок 4.2.2 – Шаблоны приложения authentication

Приложение authentication содержит шаблоны auth и token, отвечающие за формирование страниц аутентификации и ввода кода подтверждения соответственно, а так же change\_password, для отображения страницы смены пароля пользователем и change\_password\_done, для отображения страницы с информацией об успешной смене пароля.

Шаблон auth.html представлен ниже:

```
{% extends 'bootstrap3.html' %}
{% load bootstrap3 %}
{% load staticfiles %}
{% block auth %}
  <link href={% static "authentication/style.css" %} rel="stylesheet">
  <form id="form_login" method="post" class="form-signin" role="form">
    {% csrf_token %}
    
    <input type="text" id="inputName" name="username" class="form-control"
placeholder="Учетная запись" required autofocus>
    <input type="password" id="password" name="password" class="form-control"
placeholder="Пароль" data-toggle="tooltip" data-placement="right" data-trigger="manual" data-
title="Включен Caps Lock" required>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Войти</button>
    <p class="mt-5 text-muted text-center">&copy; 2018</p>
    {% if messages %}
      <div class="modal fade" id="myModal" role="dialog">
        <div class="modal-dialog">
          <div class="modal-content">
            <div class="modal-header">
              <button type="button" class="close" data-dismiss="modal">&times;</button>
              <h4 class="modal-title">Ошибка</h4>
            </div>
            <div class="modal-body">

```

```

        {% for message in messages % }
            {{ message }}
        {% endfor % }
    </div>
</div>
</div>
</div>
    {% endif % }
</form>
<script type="text/javascript" src="{% static 'authentication/jquery.js' %}">
{% endblock % }

```

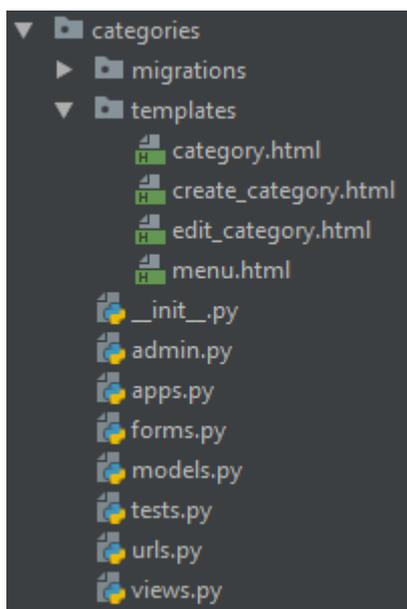


Рисунок 4.2.3 – Шаблоны приложения categories

Приложение categories содержит следующие шаблоны:

- menu – отображение главной страницы со списком родительских категорий и деревом объектов;
- create\_category – отображение страницы создания категории;
- edit\_category – отображение страницы редактирования категории;
- category – отображение страницы с содержимым каждой категории.

Шаблон menu.html представлен в приложении Г.

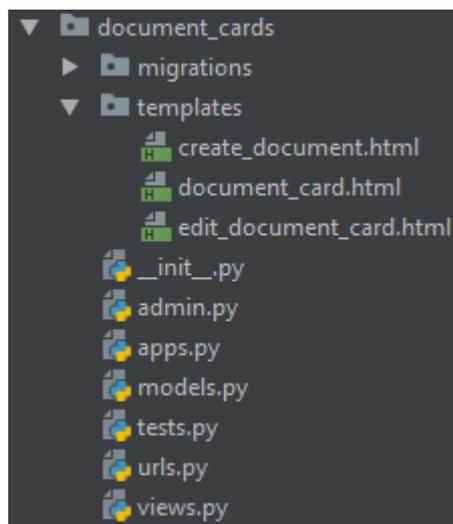


Рисунок 4.2.4 – Шаблоны приложения `document_cards`

Приложение `document_cards` содержит следующие шаблоны:

- `document_card` – отображение страницы просмотра карты документа;
- `create_document` – отображение страницы создания карты документа;
- `edit_document_card` – отображение страницы редактирования карты документа.

Шаблон `document_card.html` представлен в приложении Г.

Одна из сильных сторон Django – это автоматический интерфейс администратора. Он использует мета-данные модели, чтобы предоставить многофункциональный, готовый к использованию интерфейс для работы с содержимым сайта. Для настройки интерфейса администратора было установлено приложение `Django-admin-tools`, модифицирующее работу стандартной админки. Так же весь проект был настроен на русский язык, в файле `settings.py`, в моделях и в файле `admin.py` были созданы специальные классы, для настройки.

Результаты разработки в виде интерфейса пользователя представлены в приложении Д.

## **5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ**

Целью данного раздела является определение оценки коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения, а также планирование и формирование бюджета научных исследований, определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.

Достижение цели обеспечивается решением задач:

- разработка общей экономической идеи проекта, формирование концепции проекта;
- организация работ по научно-исследовательскому проекту;
- определение возможных альтернатив проведения научных исследований;
- планирование научно-исследовательских работ;
- оценки коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения;
- определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.

В данной диссертационной работе разработан программный комплекс электронного документооборота в рамках глобальной сети интернет с повышенным уровнем безопасности.

### **5.1 Потенциальные потребители результатов исследования**

Разработанная в рамках данной магистерской диссертации система может использоваться различными компаниями, в которых важно отслеживать жизненный цикл документов, а также поддерживать безопасность при работе с документами.

Для анализа потребителей результатов исследования был рассмотрен целевой рынок и проведено его сегментирование. На основе выявленных

данных была построена карта сегментирования, она представлена в таблице 5.1.1.

Таблица 5.1.1 – Карта сегментирования рынка услуг по разработке

		Вид приложения		
		Веб-приложение	Мобильное приложение	Настольное приложение
Размер компании	Крупные			
	Средние			
	Мелкие			

■ – области, где имеются продукты со схожим функционалом различных компаний. Однако данные продукты либо отвечают конкретным требованиям использующей их компании, либо универсальны и не отвечают требованиям, выдвигаемым к разрабатываемой системе. Абсолютных аналогов данного продукта нет.

Исходя из вышеприведенных данных, можно сделать выводы, определяющие результаты сегментирования рынка:

Основным сегментом рынка выбрана область веб-приложений;

Сегменты, на которые необходимо ориентироваться: средние и крупные компании;

Сегменты рынка, которые привлекательны для развития разработок в будущем: все виды компаний, при учете закладывания функционала для тонкой настройки конечной системы.

### **5.1.1 Анализ конкурентных технических решений**

Разрабатываемый комплекс является уникальной разработкой. Однако существует много различных систем документооборота.

Для оценки конкурентоспособности были выбраны две подобные системы – ownCloud (K1) и Microsoft SharePoint (K2), был проведен анализ конкурентных технических решений с использованием оценочных карт, результаты которого представлены в таблице 5.1.2. Позиция разработки и

конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_i, \quad (1)$$

где  $K$  – конкурентоспособность научной разработки или конкурента;

$B_i$  – вес показателя (в долях единицы);

$B_i$  – балл  $i$ -го показателя.

Таблица 5.1.2 – Оценочная карта для сравнения конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$B_{\phi}$	$B_{k1}$	$B_{k2}$	$K_{\phi}$	$K_{k1}$	$K_{k2}$
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1.Повышение производительности труда пользователя	0,1	5	5	4	0,5	0,5	0,4
2. Удобство в эксплуатации	0,07	5	5	4	0,35	0,35	0,28
3. Помехоустойчивость	0,04	4	5	4	0,16	0,2	0,16
5. Надежность	0,09	5	5	5	0,45	0,45	0,45
4. Энергоэкономичность	0,05	4	4	4	0,2	0,2	0,2
6. Низкий уровень шума	0,04	4	4	4	0,16	0,16	0,16
7. Безопасность	0,15	5	5	3	0,75	0,75	0,45
8. Потребность в ресурсах памяти	0,06	5	3	4	0,3	0,18	0,24
9. Функциональная мощность (предоставляемые возможности)	0,07	5	4	3	0,35	0,28	0,21
10. Качество интеллектуального интерфейса	0,06	5	5	4	0,3	0,3	0,24
<b>Экономические критерии оценки эффективности</b>							
1.Конкурентоспособность продукта	0,1	5	5	4	0,5	0,5	0,4
2. Низкая стоимость реализации проекта	0,09	5	3	3	0,45	0,27	0,27
3. Короткие сроки внедрения	0,04	5	4	4	0,2	0,16	0,16
4. Трудоемкость	0,04	5	4	4	0,2	0,16	0,16
<b>Итого</b>	<b>1</b>				<b>4,87</b>	<b>4,46</b>	<b>3,78</b>

На основании представленного выше анализа можно сделать вывод, что разработанная в данной диссертационной работе система является

конкурентоспособной.

### 5.1.2 Технология QuaD

Для измерения характеристик, описывающих качество новой разработки и ее перспективность на рынке и позволяющих принимать решение целесообразности вложения денежных средств в проект, была применена технология QuaD, результат представлен в таблице 5.1.3.

Таблица 5.1.3 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
1. Энергоэффективность	0,04	70	100	0,7	0,028
2. Помехоустойчивость	0,04	90	100	0,9	0,036
3. Надежность	0,05	90	100	0,9	0,045
4. Унифицированность	0,07	70	100	0,7	0,049
5. Уровень материалоемкости разработки	0,08	90	100	0,9	0,072
6. Уровень шума	0,05	90	100	0,9	0,045
7. Безопасность	0,07	90	100	0,9	0,063
8. Потребность в ресурсах памяти	0,03	60	100	0,6	0,018
9. Функциональная мощность	0,07	90	100	0,9	0,063
10. Простота эксплуатации	0,04	90	100	0,9	0,036
11. Качество интеллектуального интерфейса	0,03	80	100	0,8	0,024
12. Ремонтопригодность	0,03	80	100	0,8	0,024
<b>Показатели оценки коммерческого потенциала разработки</b>					
13. Конкурентоспособность продукта	0,07	80	100	0,8	0,056
14. Уровень проникновения на рынок	0,04	70	100	0,7	0,028
15. Перспективность рынка	0,07	90	100	0,9	0,063
16. Цена	0,05	80	100	0,8	0,04
17. Послепродажное обслуживание	0,03	70	100	0,7	0,021
18. Финансовая эффективность научной разработки	0,07	80	100	0,8	0,056
19. Срок выхода на рынок	0,03	50	100	0,5	0,015
20. Наличие сертификации разработки	0,04	50	100	0,5	0,02

<b>Итого</b>	<b>1</b>				<b>0,802</b>
--------------	----------	--	--	--	--------------

На основании полученных результатов можно сделать вывод о том, что разрабатываемая имитационная модель является перспективной.

### 5.1.3 SWOT-анализ

Для исследования внешней и внутренней среды разработки был проведен SWOT-анализ, результат которого представлен в таблице 5.1.4.

Таблица 5.1.4 – SWOT-анализ

	Сильные стороны проекта: С1. Разработка с учетом специфики. С2. Актуальность. С3. Удобство и простота эксплуатации. С4. Простота поддержки и доработки.	Слабые стороны проекта: Сл1. Малое количество специалистов Сл2. Ограничение распространение продукта Сл3. Для некоторых компаний ограниченный функционал Сл4. Простой, не яркий интерфейс
Возможности: В1. Отсутствие подобных систем у других компаний В2. Повышенный уровень безопасности системы В3. Возможность создания новых рабочих мест	Направления развития: 1. Продажа лицензий на использование другим компаниям.	Сдерживающие факторы: 1. Малое количество потенциальных потребителей.
Угрозы: У1. Появление и развитие конкурентов. У2. Прекращение поддержки и финансирования. У3. Кардинальное изменение требований заказчика.	Угрозы развития: 1. Появление конкурентов может свести на нет все преимущества. 2. Прекращение поддержки, может лишить проект возможности его усовершенствования и развития. 3. Изменение требований к продукту потребуют больших изменений.	Уязвимости: 1. Появление конкурентов по функционалу и цене могут привести к краху проекта. 2. При уменьшении финансирования, стоимость поддержки проекта может сделать проект не рентабельным. 3. Кардинальное изменение системы может оказаться слишком дорогим.

В процессе проведения SWOT-анализа были определены слабые и сильные стороны разработки, угрозы и возможности конкуренции в данной области. На основании выявленных параметров были определены мероприятия, позволяющие противостоять угрозам и улучшить текущее состояние разработки.

## 5.2 Планирование управления научно-техническим проектом

### 5.2.1 Иерархическая структура работ проекта

Иерархическая структура работ (ИСР) – детализация укрупненной структуры работ. В процессе создания ИСР структурируется и определяется содержание всего проекта.

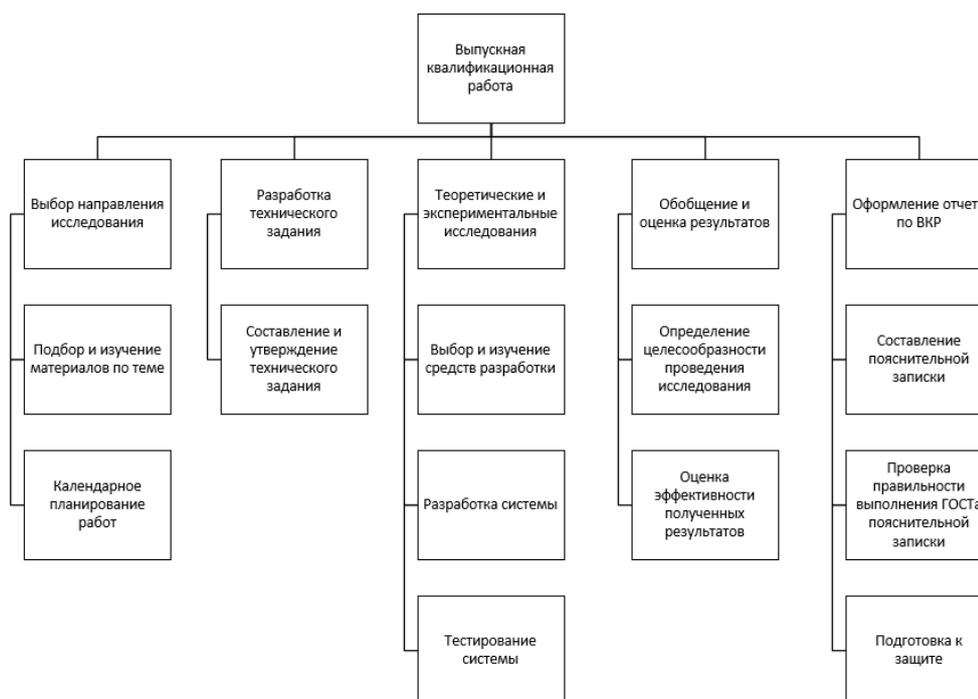


Рисунок 5.2.1 – Иерархическая структура работ

### 5.2.2 Контрольные события проекта

Ключевые события исследовательского проекта, их даты и результаты приведены в таблице 5.2.1.

Таблица 5.2.1 – Контрольные события проекта

№	Контрольное событие	Дата	Результат (подтверждающий документ)
1	Постановка задачи и анализ предметной области	04.02.2018	Задание на разработку системы
2	Анализ существующих аналогов	11.02.2018	Отчёт
3	Составление и утверждение технического задания	25.02.2018	
4	Подбор и изучение материалов по тематике	04.03.2018	Отчёт
5	Календарное планирование работ	06.03.2018	План работ
6	Проектирование системы	18.03.2018	Отчёт
7	Реализация системы	20.05.2018	Система
8	Тестирование и исправление системы	20.05.2018	Отчёт

№	Контрольное событие	Дата	Результат (подтверждающий документ)
9	Составление пояснительной записки	30.05.2018	Пояснительная записка
10	Проверка правильности выполнения ГОСТа пояснительной записки	3.06.2018	
11	Подготовка к защите	12.06.2018	Презентация, доклад

### 5.2.3 План проекта

В рамках планирования исследовательского проекта построен календарный план-график с помощью диаграммы Ганта. В данном случае работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения работ.

Линейный график представлен в таблице 5.2.2.

Таблица 5.2.2 – Календарный план проекта

Код	Название	Длительность, раб. дни	Дата начала работ	Дата окончания работ	Состав участников
1	Постановка задачи и анализ предметной области	5	29.01.2018	04.02.2018	Руководитель, инженер
2	Анализ существующих аналогов	5	05.02.2018	11.02.2018	Руководитель, инженер
3	Составление и утверждение технического задания	9	12.02.2018	25.02.2018	Руководитель, инженер
4	Подбор и изучение материалов по тематике	5	26.02.2018	04.03.2018	Инженер
5	Календарное планирование работ	2	05.03.2018	06.03.2018	Руководитель, инженер
6	Проектирование системы	6	07.03.2018	18.03.2018	Инженер
7	Реализация системы	42	19.03.2018	20.05.2018	Инженер
8	Тестирование и исправление системы	32	02.04.2018	20.05.2018	Инженер
9	Составление пояснительной записки	8	21.05.2018	30.05.2018	Инженер
10	Проверка правильности выполнения ГОСТа пояснительной записки	2	31.05.2018	3.06.2018	Руководитель, инженер
11	Подготовка к защите	6	4.06.2018	12.06.2018	Инженер

В таблице 5.2.3 представлен календарный план-график проведения научного исследования.

Таблица 5.2.3 – Календарный план-график проведения научного исследования

№ работ	Вид работ	Исполнители	Т <sub>к</sub> , кал.дн	Продолжительность выполнения работ, декады																
				Янв			Февраль			Март			Апрель			Май			Июнь	
				3	1	2	3	1	2	3	1	2	3	1	2	3	1	2		
1	Постановка задачи и анализ предметной области	Руководитель, инженер	7		▨	■														
2	Анализ существующих аналогов	Руководитель, инженер	7		▨	■														
3	Составление и утверждение технического задания	Руководитель, инженер	14			▨	■													
4	Подбор и изучение материалов по тематике	Инженер	7				■													
5	Календарное планирование работ	Руководитель, инженер	2				▨													
6	Проектирование системы	Инженер	12				■													
7	Реализация системы	Инженер	63						■	■	■	■	■	■	■	■	■	■		
8	Тестирование системы	Инженер	49								■	■	■	■	■	■	■	■		
9	Составление пояснительной записки	Инженер	10														■			
10	Проверка правильности выполнения ГОСТа пояснительной записки	Руководитель, инженер	4														▨			
11	Подготовка к защите	Инженер	9															■		

▨ – Руководитель      ■ – Инженер

### 5.3 Бюджет разработки

При планировании бюджета должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета используется следующая группировка затрат по статьям:

- материальные затраты;
- затраты на специальное оборудование для научных работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

#### 5.3.1 Расчёт материальных затрат

Расчет материальных затрат осуществляется по следующей формуле:

$$Z_M = (1 + k_T) \cdot \sum_{i=1}^m C_i \cdot N_{\text{расх}i}, \quad (2)$$

где  $m$  – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{\text{расх}i}$  – количество материальных ресурсов  $i$ -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м<sup>2</sup> и т.д.);

$C_i$  – цена приобретения единицы  $i$ -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м<sup>2</sup> и т.д.);

$k_T$  – коэффициент, учитывающий транспортно-заготовительные расходы, принимаются в пределах 15 – 25 % от стоимости материалов.

Основными затратами в данной работе являются затраты на электроэнергию и приобретение канцелярских товаров. Результаты расчётов по затратам на материалы приведены в таблице 5.3.1.

Затраты на электроэнергию рассчитываются по формуле:

$$C = C_{\text{эл}} \cdot P \cdot F_{\text{об}} = 5,8 \cdot 0,5 \cdot 960 = 2784, \quad (3)$$

где  $C_{\text{эл}}$  – тариф на электроэнергию (5,8 руб. за 1 кВт·ч);

$P$  – мощность оборудования, кВт;

$F_{об}$  – время использования оборудования, ч.

Затраты на электроэнергию составили 2784 рубля.

Таблица 5.3.1 – Материальные затраты

Наименование	Марка, размер	Количество	Цена за единицу, руб.	Сумма, руб.
Электронергия	–	480 кВт·ч	5,8	2784
Бумага	SmartList	130	1,1	143
Печать на листе А4	–	130	1,6	208
Ручка	INKJOY 300, Paper Mate	1	50	50
Доступ в интернет	–	4 месяца	300	1200
Персональный компьютер				45000
Всего за материалы				4385
Транспортно-заготовительные расходы				0
Итого по статье $C_m$				49385

### 5.3.2 Основная заработная плата исполнителей темы

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату:

$$C_{зп} = Z_{осн} + Z_{доп}, \quad (4)$$

где  $Z_{осн}$  – основная заработная плата;

$Z_{доп}$  – дополнительная заработная плата.

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_{раб}, \quad (5)$$

где  $Z_{дн}$  – среднедневная заработная плата работника, руб.;

$T_{раб}$  – продолжительность выполняемых работ, раб. дн.

Среднедневная заработная плата рассчитывается по следующей формуле:

$$Z_{дн} = (Z_m \cdot M) / T_d, \quad (6)$$

где  $Z_m$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

- а) при отпуске в 56 раб. дн.  $M = 10,1$  месяца, 5-дневная неделя;
- б) при отпуске в 48 раб. дн.  $M = 10,4$  месяца, 6-дневная неделя.

$T_0$  – действительный годовой фонд рабочего времени

научно-технического персонала, раб. дн. (таблица 5.3.2).

Таблица 5.3.2 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней:		
– Выходные	59	118
– Праздничные	14	14
Отпуск	48	24
Действительный годовой фонд рабочего времени	250	205

Инженер-программист во время выполнения настоящей работы ежемесячно получает заработную плату в размере 9489 руб. (без учёта РК).

Основная заработная плата научного руководителя рассчитывается на основании отраслевой оплаты труда. Отраслевая система оплаты труда в ТПУ предполагает следующий состав заработной платы:

- оклад – определяется предприятием, в ТПУ оклады распределены в соответствии с занимаемыми должностями, например, ассистент, старший преподаватель, доцент, профессор;
- стимулирующие выплаты – устанавливаются руководителем подразделений за эффективный труд, выполнение дополнительных обязанностей и т.д.;
- иные выплаты – районный коэффициент.

Для расчёта месячного должностного оклада работника применяется формула:

$$Z_m = Z_{ок} \cdot k_p, \quad (7)$$

где  $Z_{ок}$  – заработная плата по окладу, руб.;

$k_p$  – районный коэффициент, равный 1,3 (для Томска).

Руководителем данной научно-исследовательской работы является сотрудник в должности доцента с окладом 27500 руб. (без учёта РК). Расчёт основной заработной платы приведён в таблице 5.3.3.

Таблица 5.3.3 – Расчёт основной заработной платы

Исполнители	$Z_{ок}$ , руб.	$k_p$	$Z_m$ , руб.	$Z_{дн}$ , руб.	$T_{раб}$ , дн.	$Z_{осн}$ , руб.
Руководитель	27500	1,3	35750	1487,2	8	11897,6
Инженер	9489	1,3	12335,7	607,75	122	74146,6
Итого $Z_{осн}$ :						86044,2

### 5.3.3 Дополнительная заработная плата исполнителей темы

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций.

Дополнительная заработная плата рассчитывается исходя из 10-15 % от основной заработной платы работников, непосредственно участвующих в выполнении темы:

$$Z_{доп} = k_{доп} \cdot Z_{осн}, \quad (8)$$

где  $Z_{доп}$  – дополнительная заработная плата, руб.;

$k_{доп}$  – коэффициент дополнительной заработной платы;

$Z_{осн}$  – основная заработная плата, руб.

Примем коэффициент дополнительной заработной платы для научного руководителя равным 12 %. Результаты расчёта основной и дополнительной заработной платы исполнителей научного исследования представлены в таблице 5.3.4.

Таблица 5.3.4 – Заработная плата исполнителей исследовательской работы

Заработная плата	Руководитель	Инженер
Основная зарплата, руб.	11897,6	74146,6
Дополнительная зарплата, руб.	1427,7	8897,6
Зарплата исполнителя, руб.	13325,3	83044,2
Итого по статье $C_{зн}$ , руб.:	96369,5	

### 5.3.4 Отчисления во внебюджетные фонды

Размер отчислений во внебюджетные фонды составляет 30 % от суммы затрат на оплату труда работников, непосредственно занятых выполнением исследовательской работы.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$C_{внеб} = k_{внеб} \cdot (Z_{осн} + Z_{доп}), \quad (9)$$

где  $k_{внеб}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и другие).

Величина отчислений во внебюджетные фонды составляет для научного руководителя и инженера соответственно:

$$C_{внеб} = 0,3 \cdot (11897,6 + 1427,7) = 3997,6 \text{ руб.}$$

$$C_{внеб} = 0,3 \cdot (74146,6 + 8897,6) = 24913,26 \text{ руб.}$$

### **5.3.5 Накладные расходы**

В эту статью включаются затраты на управление и хозяйственное обслуживание, которые могут быть отнесены непосредственно на конкретную тему. Кроме того, сюда относятся расходы по содержанию, эксплуатации и ремонту оборудования, производственного инструмента и инвентаря, зданий, сооружений и др.

Расчет накладных расходов ведется по следующей формуле:

$$C_{накл} = k_{накл} \cdot (Z_{осн} + Z_{доп}), \quad (10)$$

где  $k_{накл}$  – коэффициент накладных расходов.

Накладные расходы в ТПУ составляют 25-35 % от суммы основной и дополнительной зарплаты работников, участвующих в выполнении темы.

Примем  $k_{накл} = 30 \%$ .

Накладные расходы составляют:

$$C_{накл} = 0,3 \cdot (11897,6 + 1427,7 + 74146,6 + 8897,6) = 28910,85 \text{ руб.}$$

### **5.3.6 Формирование бюджета затрат исследовательского проекта**

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект по

каждому варианту исполнения приведен в таблице 5.3.5.

Таблица 5.3.5 – Расчёт бюджета затрат исследовательского проекта

Наименование статьи	Сумма, руб
1. Материальные затраты исследования	49385
3. Затраты по основной заработной плате исполнителей темы	86044,2
4. Затраты по дополнительной заработной плате исполнителей темы	10325,3
5. Отчисления во внебюджетные фонды	28910,85
6. Накладные расходы	28910,85
Бюджет затрат исследования	203576,2

#### 5.4 Организационная структура проекта

Организационная структура проекта представляет собой временное структурное образование, создаваемое для достижения поставленных целей и задач проекта и включающее в себя всех участников процесса выполнения работ на каждом этапе.

Данной исследовательской работе соответствует функциональная структура организации. То есть организация рабочего процесса выстроена иерархически: у каждого участника проекта есть непосредственный руководитель, сотрудники разделены по областям специализации, каждой группой руководит компетентный специалист (функциональный руководитель).

Организационная структура научного проекта представлена на рисунке 5.4.1.



Рисунок 5.4.1 – Организационная структура научного проекта

## 5.5 Матрица ответственности

Степень ответственности каждого члена команды за принятые полномочия регламентируется матрицей ответственности. Матрица ответственности данного проекта представлена в таблице 5.5.1.

Таблица 5.5.1 – Матрица ответственности

Этапы проекта	Научный руководитель	Консультант раздела «Финансовый менеджмент»	Консультант раздела «Соответственность»	Консультант по языковому разделу	Инженер
Постановка задачи и анализ предметной области	О				И
Анализ существующих аналогов	О				И
Составление и утверждение технического задания	С				И
Подбор и изучение материалов по тематике	О				И
Календарное планирование работ	О				И
Проектирование системы					И
Реализация системы					И
Тестирование и исправление системы					И
Выполнение оценки ресурсоэффективности и ресурсосбережения		С			И
Выполнение раздела по социальной ответственности			С		И
Выполнение перевода части работы на английский язык				С	И
Составление пояснительной записки	С				И
Проверка правильности выполнения ГОСТа пояснительной записки	С				И
Подготовка к защите	О				И

Степень участия в проекте характеризуется следующим образом:

- ответственный (О) – лицо, отвечающее за реализацию этапа проекта и контролирующее его ход;
- исполнитель (И) – лицо, выполняющие работы в рамках этапа проекта.
- утверждающее лицо (У) – лицо, осуществляющее утверждение результатов этапа проекта (если этап предусматривает утверждение);

– согласующее лицо (С) – лицо, осуществляющее анализ результатов проекта и участвующее в принятии решения о соответствии результатов этапа требованиям.

## **5.6 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования**

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки определяется:

$$I_{финр}^{исп.i} = \frac{\Phi_{pi}}{\Phi_{max}}, \quad (11)$$

где  $I_{финр}^{исп.i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{max}$  - максимальная стоимость исполнения научно-исследовательского проекта.

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в размах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в размах (значение меньше единицы, но больше нуля).

Так как разработка имеет одно исполнение, то

$$I_{финр}^p = \frac{\Phi_p}{\Phi_{\max}} = \frac{203576,2}{203576,2} = 1;$$

Для аналогов соответственно:

$$I_{фина1}^{a1} = \frac{\Phi_{a1}}{\Phi_{\max}} = \frac{256506,012}{203576,2} = 1,26; \quad I_{фина1}^{a2} = \frac{\Phi_{a1}}{\Phi_{\max}} = \frac{303328,538}{203576,2} = 1,49;$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (12)$$

где  $I_{pi}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки;

$a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;

$b_i^a, b_i^p$  – бальная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

$n$  – число параметров сравнения.

Расчёт интегрального показателя ресурсоэффективности представлен ниже.

Таблица 5.6.1 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии \ ПО	Весовой коэффициент параметра	Текущий проект	Аналог 1	Аналог 2
1. Способствует росту производительности труда пользователя	0,35	5	4	4
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,25	4	4	4
3. Функциональная мощность	0,25	4	3	4
4. Материалоемкость	0,15	4	4	4
ИТОГО	1	4,35	3,75	4

$$I_{\text{тп}} = 5 \cdot 0,35 + 4 \cdot 0,25 + 4 \cdot 0,25 + 4 \cdot 0,15 = 4,35;$$

$$\text{Аналог 1} = 4 \cdot 0,35 + 4 \cdot 0,25 + 3 \cdot 0,25 + 4 \cdot 0,15 = 3,75;$$

$$\text{Аналог 2} = 4 \cdot 0,35 + 4 \cdot 0,25 + 4 \cdot 0,25 + 4 \cdot 0,15 = 4.$$

Интегральный показатель эффективности вариантов исполнения

разработки ( $I_{финр}^p$ ) и аналога ( $I_{финаi}^{ai}$ ) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{финр}^p = \frac{I_m^p}{I_{финр}^p}; I_{финаi}^{ai} = \frac{I_m^{ai}}{I_{финаi}^{ai}}; \quad (13)$$

В результате:

$$I_{финр}^p = \frac{I_m^p}{I_{финр}^p} = 4,35; I_{фина1}^{a1} = \frac{I_m^{a1}}{I_{фина1}^{a1}} = 2,97; I_{фина2}^{a2} = \frac{I_m^{a2}}{I_{фина2}^{a2}} = 2,68;$$

Сравнение интегрального показателя эффективности текущего проекта и аналогов позволит определить сравнительную эффективность проекта.

Сравнительная эффективность проекта:

$$\mathcal{E}_{cp} = \frac{I_{финр}^p}{I_{финаi}^{ai}} \quad (14)$$

Результат вычисления сравнительной эффективности проекта и сравнительная эффективность анализа представлены в таблице 5.6.2.

Таблица 5.6.2 – Сравнительная эффективность разработки

№	Показатели	Аналог 1	Аналог 2	Разработка
1	Интегральный финансовый показатель разработки	1,26	1,49	1
2	Интегральный показатель ресурсоэффективности разработки	3,75	4	4,35
3	Интегральный показатель эффективности	2,97	2,68	4,35
4	Сравнительная эффективность вариантов исполнения	1,46	1,62	1

Таким образом, основываясь на определении ресурсосберегающей, финансовой, бюджетной, социальной и экономической эффективности исследования, проведя необходимый сравнительный анализ, можно сделать вывод о превосходстве выполненной разработки над аналогами.

В ходе данного раздела была оценена коммерческая ценность разработки, определены перспективность и успешность проекта, с позиции ресурсоэффективности и ресурсосбережения, а также ресурсная, финансовая, бюджетная, социальная и экономическая эффективности исследования.

## 6 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Обеспечение производственной и экологической безопасности является необходимым условием реализации любых проектов, в том числе конструкторских и исследовательских. Обеспечение безопасности, в общем, предполагает создание безопасных и благоприятных рабочих условий для всех лиц, задействованных в работах, предусмотренных проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление возможных причин потенциальных несчастных случаев, производственных травм, профессиональных заболеваний, аварий и пожаров. Дальнейшими этапами являются разработка мероприятий по устранению выявленных причин и их реализация. Потенциальные причины и риски, а также конкретный набор мероприятий по их устранению, определяются спецификой выполняемых работ и априорными условиями труда (в частности, видом и состоянием рабочих мест исполнителей).

В данной работе реализуется программный комплекс электронного документооборота. Основным исполнителем работы является программист. Объектом для исследования работы является задача на разработку программного комплекса электронного документооборота.

На рабочем месте возможно возникновение вредных факторов, таких как: недостаточная освещенность рабочего места, повышение уровня шума, повышенная либо пониженная температура воздуха.

Также на данном рабочем месте могут проявляться опасные факторы среды, например, поражение электрическим током. Возможной чрезвычайной ситуацией на рабочем месте является возникновение пожара.

## **6.1 Производственная безопасность**

### **6.1.1 Анализ вредных и опасных факторов**

Классификация опасных и вредных факторов дана в основополагающем стандарте ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация».

#### **6.1.1.1 Вредное проявление факторов**

##### **Микроклимат рабочей зоны**

Обеспечение в помещениях нормальных метеорологических условий, оказывающих существенное влияние на тепловое самочувствие человека, является одним из необходимых благоприятных условий труда. Микроклимат в производственных помещениях, зависит от особенностей технологического процесса, а также внешних условий (категории работ, периода года, условий вентиляции и отопления).

Микроклимат характеризуется:

- температура,  $t$ , °С;
- относительная влажность,  $\varphi$ , %;
- скорость движения воздуха на рабочем месте,  $V$ , м/с;
- барометрическое давление,  $p$ , мм рт. ст. (не нормируется).

Повышенная влажность затрудняет теплоотдачу организма путем испарений при высокой температуре воздуха и способствует перегреву, а при низкой температуре, наоборот, усиливает теплоотдачу, способствуя переохлаждению. Оптимальными параметрами микроклимата являются те, которые при длительном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма без напряжения реакций терморегуляции, что создает ощущение теплового комфорта и служит предпосылкой для высокой работоспособности.

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительным физическим напряжением.

Интенсивность энерготрат организма для данной категории работ составляет до 120 ккал/ч (до 139 Вт).

На рабочих местах производственных помещений, согласно СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений», должны быть установлены оптимальные и допустимые микроклиматические условия для категории работ Ia. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (значительные и незначительные тепловыделения).

Оптимальные значения представлены в таблице 6.1.1. Оптимальные микроклиматические условия обеспечивают локальное и общее ощущение теплового комфорта при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

Таблица 6.1.1 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Оптимальные				
холодный	Легкая Ia	22-24	40-60	0,1
теплый	Легкая Ia	23-25	40-60	0,1

В таблице 6.1.2 приведены допустимые величины показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia.

Допустимые микроклиматические условия не вызывают повреждений или нарушений состояния здоровья, но могут приводить к возникновению общих и локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности.

Таблица 6.1.2 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Допустимые				
холодный	Легкая 1а	20-25	15-75	0,1
теплый	Легкая 1а	21-28	15-75	0,1 до 0,2

Параметры микроклимата в помещении, где находится рабочее место, регулируются системой центрального отопления и приточно-вытяжной вентиляцией.

К мероприятиям по оздоровлению воздушной среды в производственном помещении относятся: правильная организация вентиляции и кондиционирования воздуха, отопление помещений. Вентиляция может осуществляться естественным и механическим путём

### **Производственное освещение**

Недостаточность уровня освещения приводит к напряжению зрения, снижению его остроты, ослабляет внимание, приводит к наступлению преждевременной утомленности и головным болям. К дезориентации рабочего может привести неправильное направление света на рабочем месте, которое может создавать резкие тени, блики. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Весьма важен правильный расчет освещенности, иначе все эти вредные факторы могут привести к профзаболеваниям или несчастному случаю.

Необходимо создать систему освещения рабочего места для комфортной работы, отвечающую нормам, указанным в соответствии со СП 52.13330.2011. Согласно вышеуказанному документу, освещённость рабочего места должна быть равна 200 лк, так как работа программиста за компьютером относится к 3 категории зрительной работы.

В кабинете используется искусственное и естественное освещение. В качестве искусственного освещения используется система общего освещения.

## **Производственный шум**

Одним из вредных производственных факторов является шум. Под длительным воздействием на человека шума появляется усталость, снижается острота зрения, слуха, повышается кровяное давление, понижается внимание, ухудшается речевая коммуникация. Сильный продолжительный шум может привести к функциональным изменениям сердечнососудистой и нервной систем.

На рассматриваемом рабочем месте источниками шума являются технические средства – компьютеры, МФУ и кондиционер. Они издают незначительный шум, поэтому не влияют на работу. Уровень шума не должен превышать значений, установленных СН 2.2.4/2.1.8.562–96 и составлять не более 50 дБА.

Разработка шумобезопасной техники, применение средств и методов коллективной защиты, в том числе строительно-кустических, применение средств индивидуальной защиты должны обеспечивать защиту от вредного воздействия шума на организм человека.

Средства и методы коллективной защиты в зависимости от способа реализации подразделяются на строительно-акустические, архитектурно-планировочные и организационно-технические и включают в себя:

- применение звукоизоляции;
- изменение направленности излучения шума;
- акустическую обработку помещений.

В случае если другими способами обеспечить допустимый уровень шума на рабочем месте не удастся, применяются средства индивидуальной защиты.

Шумовая обстановка в рассматриваемой аудитории по субъективным ощущениям соответствует нормам.

## Электромагнитное излучение

Основным источником электромагнитных полей и излучений в рассматриваемом помещении являются компьютеры.

Правильный выбор визуальных параметров дисплея и светотехнических условий рабочего места являются важным условием безопасности пользователя перед экраном.

К зрительному утомлению, головным болям, значительной физиологической и психологической нагрузке, к ухудшению зрения приводит неправильный выбор освещенности экрана и яркости, контрастности знаков, цветов знаков и фона, при наличии бликов на экране, дрожания и мелькания изображения.

Требования по безопасности эксплуатации в России определены ГОСТ Р 50948-96, ГОСТ Р 50949-96 и СанПин 2.2.2/2.4.1340-03. Требования этих стандартов обязательны для всех мониторов.

В таблице 6.1.3 представлены временные допустимые уровни электромагнитных полей монитора на рабочем месте.

Таблица 6.1.3 – Временные допустимые уровни электромагнитных полей мониторов

Параметры	ВДУ ЭМП
Напряженность электрического поля диапазон 5Гц-2кГц (сверхнизкие)	25В/м
Напряженность электрического поля диапазон 2Гц-400кГц (низкие)	2,5В/м
Плотность магнитного потока диапазон 5Гц-2кГц (сверхнизкие)	250нТл
Плотность магнитного потока диапазон 2кГц-400кГц (низкие)	25нТл
Электростатический потенциал	500 В

Даже при соблюдении всех требований и стандартов к монитору ничто не сможет уберечь пользователя от ухудшения здоровья (в первую очередь зрения).

Источником ЭМИ в рассматриваемом помещении является монитор компьютера.

В данном случае используются только предупредительные меры защиты, такие как:

- размещение монитора таким образом, чтобы задняя его панель (область наибольшего излучения) была обращена от пользователя и окружающих его людей;
- достаточная освещенность рабочего места. Наиболее подходящим осветителем в данном случае является небольшая люминесцентная лампа;
- кратковременные перерывы в процессе работы.

### **6.1.1.2 Опасное проявление факторов**

К опасным факторам производственной среды можно отнести:

- электробезопасность;
- пожаробезопасность.

#### **Электробезопасность**

Одним из выявленным опасных факторов является поражение электрическим током, так как напряжение считается безопасным при  $U < 42$  В, а вычислительная техника питается от сети 220 В 50 Гц. Ток является опасным, так как 20 – 100 Гц – ток наиболее опасен. Поэтому результатом воздействия на организм человека электрического тока могут быть электрические травмы, электрические удары, и даже смерть [ГОСТ Р 12.1.009-2009].

Виды электротравм: местные электротравмы. К ним относятся: электрический ожог, электрические знаки, металлизация кожи, механические повреждения. Особую опасность представляют электрические травмы, которые выглядят в виде ожогов.

Электрический ожог возникает на том месте тела человека, в котором контакт происходит с токоведущей частью электроустановки. Электроожоги сопровождаются кровотечениями, омертвением отдельных участков тела.

В результате механического повреждения могут разорваться кровеносные сосуды, нервные ткани, а также случаются вывихи суставов и даже переломы костей. Такие повреждения могут возникнуть в результате сокращений мышц под действием тока, который проходит через тело человека.

Чтобы защититься от поражения током, необходимо:

- обеспечить недоступность токоведущих частей от случайных прикосновений;
- электрическое разделение цепи;
- устранять опасности поражения при проявлении напряжения на разных частях;

При работе с компьютером прикосновения к его элементам могут возникнуть токи статического электричества. Пыль на экране ухудшает видимость, а при подвижности воздуха может попасть на кожу лица и в легкие, что вызывает заболевание кожи и дыхательных путей.

Есть специальные шнуры питания с заземлением и экраны для снятия статического электричества, это поможет защититься от статического электричества, а также необходимо проводить регулярную влажную уборку рабочего помещения.

Мониторы являются источниками интенсивных электромагнитных полей. Электромагнитные поля могут вызывать изменения в клетках. Длительное воздействие низких частот ЭВМ вызывает нарушения сердечнососудистой и центральной нервной системы, небольшие изменения в составе крови. Возможно возникновение катаракты глаз, злокачественных опухолей при интенсивном длительном воздействии.

Степень воздействия зависит от продолжительности работы и индивидуальных особенностей организма.

Для снижения уровня воздействия, необходимо:

- экранирование экрана монитора;
- соблюдать оптимально расстояние от экрана;
- рационально размещать оборудование (если имеется несколько компьютеров, то расстояние между боковыми и задними стенками компьютеров должно быть 1,22 м);
- организовывать перерывы 10-15 минут через каждые 45-60 минут работы [СанПиН 2.2.2/2.4.1340-03].

## **Пожаробезопасность**

Пожаробезопасность – состояние объекта, характеризующее возможность предотвращения возникновения и развития пожара, а также воздействия на людей и имущество опасных факторов пожара. Пожарная безопасность объекта должна обеспечиваться системами предотвращения пожара и противопожарной защиты, в том числе организационно-техническими мероприятиями. Регулирование пожаробезопасности производится СНиП 21-01-97.

Методы противодействия пожару делятся на:

- профилактические;
- защитные.

Существуют следующие профилактические действия.

- Изолирование электропроводки для избегания возникновения короткого замыкания, способного привести к пожару.
- Изолирование розеток, расположенных в санузлах и на внешних стенах, от влаги.
- Устанавливание устройств защитного отключения и автоматических предохранителей.
- Теплоизоляция газовых и электрических плит от деревянной мебели.

Тушение пожара производится огнетушителями различного наполнения, песком и другими негорючими материалами, мешающими огню распространяться и гореть.

Для самозвакуации людей из горящих зданий применяется лебёдка, закреплённая с внешней стороны окна, по которой проживающие на высоких этажах люди могут спуститься на землю.

## **6.2 Экологическая безопасность**

Основные факторы, оказывающие негативные действия на экологию связаны с производством и эксплуатацией компьютерной техники. При рассмотрении влияния ПК на атмосферу, гидросферу и литосферу выявлены особо вредные выбросы: ртутные лампы, согласно ГОСТ Р 51768-2001. В случае выхода из строя компьютеров, они списываются и отправляются на специальный склад, который при необходимости принимает меры по утилизации списанной техники и комплектующих. В настоящее время в Томской области утилизацией занимаются две компании: городской полигон и ООО НПП «Экотом». Утилизацией опасных бытовых отходов занимаются компании: ООО «Торем», ООО «СибМеталлГрупп».

## **6.3 Безопасность в чрезвычайных ситуациях**

Чрезвычайная ситуация: Обстановка на определенной территории или акватории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которые могут повлечь или повлекли за собой человеческие жертвы, ущерб здоровью людей или окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей. ГОСТ Р 22.0.01-94 «Безопасность в чрезвычайных ситуациях».

Возможными чрезвычайными ситуациями могут быть:

- техногенные: взрывы, пожары, обрушение помещений, аварии на системах жизнеобеспечения;
- природные: наводнения, ураганы, бури, природные пожары;
- экологические: разрушение озонового слоя, кислотные дожди;
- биологические: эпидемии, пандемии;
- антропогенные: война, терроризм.

Меры предупреждения пожара рассмотрены в пункте 6.1.1.2 Пожаробезопасность.

## **6.4 Правовые и организационные вопросы обеспечения безопасности**

### **6.4.1 Специальные правовые нормы трудового законодательства**

Продолжительность рабочего дня не должна превышать 40 часов в неделю. Возможно, сокращение рабочего времени. Для работников, возраст которых меньше 16 лет – не более 24 часа в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II группы. Также рабочее время зависит от условий труда: для работников, работающих на рабочих местах с вредными условиями для жизни – не больше 36 часов в неделю.

Согласно СанПиН 2.2.2/2.4.1340-03, норма площади рабочего места с персональным компьютером составляет 4,5 м<sup>2</sup>. Можно сделать вывод, что рабочее помещение соответствует санитарным нормам.

Создание данного программного обеспечения и его эксплуатация не подразумевают физического труда, что негативно влияет на здоровье работника

### **6.4.2 Организационные мероприятия при компоновке рабочей зоны**

Рабочее место должно обеспечивать возможность удобного выполнения работ, учитывать размеры рабочей зоны, а также необходимость передвижения в ней работающего.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место должно соответствовать требованиям ГОСТ 12.2.032-78 «ССБТ. Общие эргономические требования».

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Не рекомендуется располагать компьютеры рядом друг с другом в целях уменьшения действия переменного электрического поля.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения работы было реализовано веб-приложение для электронного документооборота с повышенным уровнем безопасности в сети Интернет.

Таким образом, поставленные задачи работы успешно выполнены, цель достигнута. В дальнейшем планируется осуществление сопровождения системы: взаимодействие с пользователями для консультирования по вопросам эксплуатации системы, сбору требований по модификациям и доработкам системы, а так же реализация соответствующих модификаций и доработок.

В ходе данной работы были выполнены следующие работы:

- 1) Изучена предметная область;
- 2) Исследованы аналогичные системы;
- 3) Разработано техническое задание на создание системы;
- 4) Спроектирован пользовательский интерфейс;
- 5) Спроектирована архитектура системы;
- 6) Спроектирована база данных;
- 7) Реализован функционал системы.

На данный момент система находится в стадии эксплуатационного тестирования.

В пояснительной записке рассмотрены вопросы технико-экономического обоснования проекта, и определен наилучший вариант исполнения.

Отдельная глава посвящена вопросам безопасности жизнедеятельности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Двухфакторная аутентификация: что это и зачем оно нужно [Электронный ресурс]/ Kaspersky. Режим доступа: [https://www.kaspersky.ru/blog/what\\_is\\_two\\_factor\\_authentication/4272/](https://www.kaspersky.ru/blog/what_is_two_factor_authentication/4272/). Дата обращения: 03.03.18г.
2. Применение двухэтапной аутентификации для повышения безопасности пользователей информационных систем / Палютина Г. Н. // Фундаментальные и прикладные научные исследования. – 2015. – 91-95 с. Дата обращения: 05.03.18г.
3. Выбираем решение для персонального файлохранилища [Электронный ресурс]/ хакер.ru. Режим доступа: <https://hacker.ru/2014/09/04/personal-cloud-storage-review/>. Дата обращения: 07.03.18г.
4. Nextcloud [Электронный ресурс]/ Национальная библиотека им. Н. Э. Баумана. Режим доступа: <https://ru.bmstu.wiki/Nextcloud>. Дата обращения: 15.12.17г.
5. Пять недорогих Open Source-сервисов хранения для предприятий [Электронный ресурс]/ itWeek. Режим доступа: <https://www.itweek.ru/foss/article/detail.php?ID=177233>. Дата обращения: 04.12.17г.
6. Поиск облачных сервисов для бизнеса и личной продуктивности, сравнение характеристик и цен [Электронный ресурс]/ Режим доступа: <https://startpack.ru/>. Дата обращения: 17.12.17г.
7. А вы защищаете свои аккаунты двухфакторной или двухэтапной аутентификацией? [Электронный ресурс]/ Блог Вадима Стеркина. Режим доступа: <http://www.outsidethebox.ms/18372/>. Дата обращения: 14.03.18г.
8. PHP vs Python vs Ruby для разработки веб-приложений: подробное сравнение [Электронный ресурс]/ Режим доступа: <https://8d9.ru/php-vs-python->

vs-ruby-dlya-razrabotki-veb-prilozhenij-podrobnoe-sravnenie. Дата обращения: 15.12.17г.

9. Programming python / Lutz M. – O'Reilly, 1996. – Т. 8. Дата обращения: 12.12.17г.

10. SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных [Электронный ресурс]/ Режим доступа: <http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/>. Дата обращения: 23.12.17г.

11. Beginning Databases with PostgreSQL / Matthew N., Stones R. – Apress, 2005. Дата обращения: 12.03.18г.

12. PostgreSQL [Электронный ресурс]/ Википедия – свободная энциклопедия. Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL>. Дата обращения: 15.02.18г.

13. PostgreSQL: introduction and concepts / Momjian B. – New York : Addison-Wesley, 2001. – Т. 192. Дата обращения: 17.02.18г.

14. Django – фреймворк на Python [Электронный ресурс]/ Web creator. Режим доступа: <https://web-creator.ru/articles/django>. Дата обращения: 19.03.18г.

15. Python web development with Django / Forcier J., Bissex P., Chun W. J. – Addison-Wesley Professional, 2008. Дата обращения: 25.03.18г.

16. Базы данных в Python [Электронный ресурс]/ Сообщество программистов Python. Режим доступа: <https://python-scripts.com/database>. Дата обращения: 14.12.17г.

17. Часть 4. Python/Django и Hello World. Структура проекта Django и создание приложений [Электронный ресурс]/ Blog WebSofter, Режим доступа: <http://blog.websofter.ru/chast-4-pythondjango-i-hello-world-structura-django-i-sozdanie-priloghenuy/> Дата обращения: 12.02.18г.

18. Создаём своё первое приложение с Django, часть 1 [Электронный ресурс]/ Документация Django 1.9, Режим доступа: <https://djbook.ru/rel1.9/intro/tutorial01.html>. Дата обращения: 15.02.18г.

19. ГОСТ 12.0.003-74. ССБТ. Опасные и вредные производственные факторы. Классификация. Дата обращения: 15.04.18г.
20. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. Дата обращения: 17.04.18г.
21. СП 52.13330.2011. Естественное и искусственное освещение. Дата обращения: 12.04.18г.
22. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки. Дата обращения: 21.04.18г.
23. ГОСТ Р 50948-96. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. Дата обращения: 13.04.18г.
24. СанПин 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Дата обращения: 19.04.18г.
25. ГОСТ Р 12.1.009-2009. Система стандартов безопасности труда (ССБТ). Электробезопасность. Термины и определения. Дата обращения: 22.04.18г.
26. СНиП 21-01-97. Пожарная безопасность зданий и сооружений. Дата обращения: 23.04.18г.
27. ГОСТ Р 51768-2001. Ресурсосбережение. Обращение с отходами. Методика определения ртути в ртутьсодержащих отходах. Общие требования. Дата обращения: 25.04.18г.
28. ГОСТ Р 22.0.01-94. Безопасность в ЧС. Основные положения. Дата обращения: 26.04.18г.
29. ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования. Дата обращения: 28.04.18г.

# ПРИЛОЖЕНИЕ А

(обязательное)

## Chapter 2 Object and methods of research

Студент:

Группа	ФИО	Подпись	Дата
8ИМ6А	Колесова Олеся Константиновна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Буркатовская Ю.Б.	к.ф.-м.н.		

Консультант – лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ ШБИП	Комиссарова О.В.	к.ф.н.		

Консультант ОИТ ИШИТР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Мирошниченко Е.А.	к.т.н.		

## 1 Technical requirements

### 1.1 Purpose and objectives of the system

Purpose of the system

Organization of a personal online cabinet designed is automate the process of document circulation, both within the company EleSi, and with the integrators of the SCADA Infinity system.

The objectives of the system are:

- increasing the response of the customer-integrator in terms of the transfer of necessary information;
- tracking the life cycle of documents.

### 1.2 User classes

Access to the system should only be authorized users who are divided by access rights in accordance with paragraph 1.3.3 of this document.

### 1.3 System Requirements

#### 1.3.1 Requirements groups

Table A.1 lists the requirements for the system being developed.

Table A.1 - Requirements groups

Symbol	Requirements group
<b>F</b>	Functional requirements
<b>I</b>	User interface requirements
<b>AR</b>	Access rights requirements
<b>TS</b>	Requirements for technical support
<b>SR</b>	Software Requirements

#### 1.3.2 Functional requirements

Table A.2 shows the functional requirements of the system being developed.

Table A.2 - Functional requirements

Requirement code	Requirements	Notes
<b>F.01</b>	<b>General requirements</b>	
F.01.01	There should be two-step authorization of users	
F.01.02	It should be multi-user mode	

<b>Requirement code</b>	<b>Requirements</b>	<b>Notes</b>
F.01.03	It should be implemented the access rights	
F.01.04	It should be possible to work with documents	Adding a document (from your computer to the system); downloading the document (to the selected directory on the computer from the system, as one, and several documents at once).
F.01.05	An event notification should be implemented	
F.01.06	Comments should be implemented	
<b>F.02</b>	<b>Requirements for documents</b>	
F.02.01	The documents must be in the document card, which contains the document description and comments	
F.02.02	When creating a document card by the user, the following data should be generated: – Name; – Comment; – Path to file.	
F.02.03	When viewing a document card, the system should provide the following data: – Format; – URL link; – Date of creation of the document card.	
<b>F.03</b>	<b>Requirements for notifications about events</b>	
F.03.01	E-mail notifications should be implemented about changes in available categories and cards of documents in accordance with the access rights	
F.03.02	After creating a category or a document card, all users who have access to them should receive an e-mail notification	Alert type: "In the system "System name" a new category/ new document has been created for you "Category name"/ "Document card name"".

<b>Requirement code</b>	<b>Requirements</b>	<b>Notes</b>
F.03.03	After editing a category or a document card, all users who have access to them should receive an alert via e-mail	Alert type: "In the system "System name" was changed / changed the category/ document card "Name of category"/ "Name of document card"".
F.03.04	After the user creates a new comment, all users who have access to it should receive an e-mail notification	Alert type: "In system "System name" in the document card "Name of document card" added a new comment".
<b>F.04</b>	<b>Document history requirements</b>	
F.04.01	In the history of the document card, messages from the server should be displayed, which are created in case of editing a document card, adding or downloading a document	Type of message: "User name" add/changed/downloaded document "Document name""
<b>F.05</b>	<b>Comment requirements</b>	
F.05.01	Comments should be implemented in document cards	
F.05.02	Entering comments should be carried out in a special area, after clicking on the "Send" button, a commentary should appear in the comment box	Type of record: ""User name": "Message text""
<b>F.06</b>	<b>Deletion requirements</b>	
F.06.01	It should be possible to delete only document cards, if there are appropriate access rights	
F.06.02	Physical deletion of document cards does not occur, they must remain in the database and be accessible to the administrator	

### 1.3.3 Access requirements

Table A.3 provides requirements for the access rights of the system being developed.

Table A.3 - Access rights requirements

Requirement code	Requirements	Notes
<b>AR.01</b>	<b>General requirements</b>	
AR.01.01	Access to the system must only have been made to the user database	
AR.01.02	The system administrator should have full access to all functions and rights management of other users	
<b>AR.02</b>	<b>Authorization requirements</b>	
AR.02.01	There must be two-step authorization of users	
AR.02.02	After the user has successfully entered the login and password, he must receive the generated access code	
AR.02.03	The authorization code must be started for 5 minutes	
AR.02.04	If the data for authorization are incorrectly entered three times, the entry possibility must be blocked for this user for 15 minutes	
AR.02.05	If the user incorrectly enters the authorization data nine times, the login must be blocked for that user for 24 hours	
<b>AR.03</b>	<b>Requirements for access rights to user groups</b>	
AR.03.01	Creating a user group must be allowed only by the administrator	It should be possible to enter a name, and select the access rights
AR.03.02	Edit user groups should only be authorized by the administrator	It should be possible to edit the name and access rights
<b>AR.04</b>	<b>Requirements for user access rights</b>	
AR.04.01	Only administrators should be allowed to create users	It should be possible to enter the fields: –Name; –E-mail; –Password. There should be a choice: –User group.
AR.04.02	Edit users should only be authorized by the administrator	

<b>Requirement code</b>	<b>Requirements</b>	<b>Notes</b>
AR.04.03	Users should have access to edit only their password	
<b>AR.05</b>	<b>Requirements for access rights to categories</b>	
AR.05.01	A category access rule should be implemented	It should be possible to enter the fields: –Name; –Comment. There should be a choice: –Groups of users who have access
AR.05.02	The rule for accessing the editing of all available categories should be implemented	
AR.05.03	The rule of access to edit only their own, created, categories should be implemented	
<b>AR.06</b>	<b>Requirements for access rights to documents</b>	
AR.06.01	The rule of access for creating document cards should be implemented	It should be possible to enter the fields: – Name; – Comment. It should be possible to download a document from your computer
AR.06.02	The access rule for editing all available document cards should be implemented	
AR.06.03	The rule of access to edit only their own, created, document cards should be implemented	
AR.06.04	The download access rule for all available documents should be implemented	
AR.06.05	The rule of access to download only their own, created documents should be implemented	
AR.06.06	The rule of access to delete only their own, created, document cards should be implemented	
<b>AR.07</b>	<b>Requirements for document access rights</b>	
AR.07.01	The administrator should be able to see the history of all documents	
AR.07.02	Users can only see the history of their documents	Information: – User name; – Operation; – Title of the document; – The download date.

Requirement code	Requirements	Notes
<b>AR.08</b>	<b>Requirements for access rights to comments to a document card</b>	
AR.08.01	A comment access rule should be implemented	
AR.08.02	The access rule for viewing comments only should be implemented	

### 1.3.4 User interface requirements

Table A.4 provides the interface requirements for the system being developed.

Table A.4 - Interface Requirements

Requirement code	Requirements	Notes
<b>I.01</b>	<b>General requirements</b>	
I.01.01	System interface language - Russian	
I.01.02	There should be no buttons without a name or not marked with a special icon	
I.01.03	Categories and document cards should be presented in the form of a structure tree, nesting of categories in which is not limited	
<b>I.02</b>	<b>Interface requirements</b>	
I.02.01	The front-page interface should look like figure B.1 in appendix B User interface requirements	The user must enter his login in the form of an e-mail and password and click the "Login" button
I.02.02	The interface for the code entry page should look like Figure B.2 in Appendix B User interface requirements	The user must enter the code that came to his mail and click the "Login" button
I.02.03	The window with errors should look like in figure B.3 in appendix B User interface requirements	
I.02.04	The window with a message about the inoperability of the code should look like figure B.4 in appendix B User interface requirements	
I.02.05	The user menu interface should look like figure B.5 in appendix B User interface requirements	
I.02.06	The interface for the password change page should look like figure B.6 in appendix B User interface requirements	
I.02.07	The interface of the category view page should look like figure B.7 in appendix B User interface requirements	

<b>Requirement code</b>	<b>Requirements</b>	<b>Notes</b>
I.02.08	The interface of the document map view page should look like figure B.8 in appendix B User interface requirements	
I.02.9	The interface of the category creation page should look like figure B.10 in appendix B User interface requirements	
I.02.10	The interface for creating the document map should look like figure B.11 in appendix B User interface requirements	To attach a document, you must click the "Attach document" button, select the desired file and click the "Open" button
I.02.11	The interface of the category editing page should look like figure B.12 in appendix B User interface requirements	
I.02.12	The interface of the document map editing page should look like figure B.13 in appendix B User interface requirements	

### **1.3.5 Requirements for technical support**

Table A.5 shows the requirements for the technical support of the system being developed.

Table A.5 - Requirements for technical support

<b>Requirement code</b>	<b>Requirements</b>	<b>Notes</b>
TS.01	Server requirements: – Windows Server 2012 R2; – processor frequency - from 2 GHz; – the amount of RAM - from 8 GB; – PostgreSQL, version 10.1 and higher; – Python, version 3.6.4 and higher; – The amount of free disk space on the system disk is 1 TB.	

### 1.3.6 Software requirements

Table A.6 shows the software requirements for the system being developed.

Table A.6 - Software requirements

Requirement code	Requirements	Notes
<b>SR.01</b>	<b>Requirements to the client part</b>	
SR.01.01	Should function under the following operating systems: – Windows version is not lower than 7; – Linux with kernel version 3.2 or higher; – Android version 4.4 or higher; – Mac OS; – Apple iOS version not lower than 8.x.	
SR.01.02	Should not require additional software settings on the client side	
SR.01.03	Should function in the following browsers with versions not lower than specified: – Safari, version 11 and higher; – Google Chrome, version 62 and above; – Internet Explorer, version 11 and higher; – Microsoft Edge, version 40 and higher; – Mozilla Firefox, version 57 and higher; – Opera, version 49 and higher; – iOS Safari, version 8.4 and higher; – Android Browser, version 4.4 and higher.	
<b>SR.02</b>	<b>Server part requirements</b>	
SR.02.01	Should function under Windows version 7 or higher, on the Internet Information Services Web server version 10 and higher, with the establishment of a secure connection over the HTTPS protocol	

## 2 Selection of software development tools

The choice of software development tools is the main stage in the development. Each software tool is designed to solve specific tasks. In this case, you need to do the following:

- 1) Choose the programming language and framework;
- 2) Select a web server;
- 3) Selecting a DBMS.

## 2.1 Choice of programming language and framework

To develop a Web application, you can use various programming languages and frameworks.

Typically, the implementation of a web application consists of two parts - server and client. The client part is usually implemented using HTML, CSS, JavaScript. The server part is implemented using various programming languages - Java, C #, PHP, Python, Ruby, etc.

The most popular are:

- PHP – general-purpose scripting language used extensively for developing Web applications;
- Python – high-level general-purpose programming language designed to improve developer productivity and code readability;
- Ruby – dynamic, reflective, interpreted high-level programming language.

For each of the languages described, there are frameworks for building web applications: Laravel / Symphony / Yii2 for PHP, Django / Flask for Python, Ruby on Rails for Ruby. These frameworks provide a set of ready-made functions and APIs for interacting with them, as well as dictate the rules for building the application architecture.

With Django and Python, you can create a web application that works with both files and HTTP requests. The language is interpreted and freely distributed. In this case, this language supports compilation in byte-code, which allows you to increase the speed of execution.

Pluses:

- simplicity of development - the source code has a visual structure, and due to the absence of the need to describe the variables before their initialization, the semantics are preserved;
- cross-platform interpreter - this feature allows you to develop and deploy Web applications on a workstation or server with any operating system;

- a large number of ready-made libraries from third-party developers, allowing to accelerate the development of applications;
- since each Python project is desirable to run in a virtual environment (created by the virtualenv application), there is no conflict between the versions of libraries and applications, and the code is isolated at the level of the operating system.

The Django framework implements the MVT template (Model, View, Template - the interpretation of the MVC template, where View acts as the controller, and the Template as the view) and provides a rich API to implement both the components needed for the work (models, views and templates), and their own libraries of functions and classes.

This framework includes a large number of various modules that allow you to create modern web applications.

Advantages of the solution in Python / Django:

- free distribution;
- cross-platform;
- extensibility;
- high performance in comparison with other interpreted programming languages;
- abstracting from a specific database;
- high speed development.

Disadvantages of this solution:

- unusual syntax;
- the need to configure each node of the system.

## **2.2 Selecting a Web server**

The following web servers can be used to work with Python / Django:

- Apache with module mod\_python;
- Nginx with via uWSGI application;
- Gunicorn with Nginx as reverse-proxy;
- Microsoft IIS.

Microsoft IIS was chosen, as this is one of the requirements for the system.

### 2.3 Selecting the DBMS

Django supports four database management systems:

- PostgreSQL;
- SQLite 3;
- MySQL;
- Oracle.

The DBMS used should meet such requirements as speed, simplicity, efficiency and reliability. By default, the Django settings specify the use of SQLite. Django was originally designed and developed to work with the PostgreSQL DBMS.

Table A.7 provides a comparative analysis of DBMS data.

Table A.7 - Summary table of DBMS characteristics

	<b>PostgreSQL</b>	<b>MySQL</b>	<b>Oracle</b>	<b>SQLite</b>
Cost	Is free	Is free	Fee-paying	Is free
Security	High	High	High	Low
Easy to use	Medium	High	Medium	High
Easy to use with the Django Framework	High	Medium	Medium	High
Restrictions on working with Django ORM	No	Yes	Yes	Yes
Conclusions	+	-	-	-

PostgreSQL is a free object-relational database management system based on the SQL language of Structured Query Language, which is the standard for the development of relational databases. A distinctive feature of relational databases is the relationship. Internal mechanisms of relationship optimization allow achieving high processing speed. Also, relationships allow you to link data from several tables when you run queries.

The advantages of this DBMS include:

- high-performance and reliable transaction and replication mechanisms;
- extensible system of built-in programming languages;
- inheritance;
- easy extensibility;

- multiversion - simultaneous modification of the database by several users using the Multiversion Concurrency Control mechanism, which eliminates the need to block reading;

- the ability to create custom data types.

PostgreSQL is the most suitable solution for creating web applications and web services of any complexity with the help of the Django framework, which are focused on receiving, storing and processing information. This solution allows you to create dynamically generated HTML representations, that is, including information that is modified in real time.

Django also includes the Django ORM library, which provides an interface to work with DBMS. Using this library, which implements the "Active Record" design template, it is possible to create database queries in the form of using class methods inherited from the Model class. This library also optimizes the text of the request and executes it only at the moment of access to the data, which makes it possible not to make intermediate queries and thereby reduce the load on the DBMS.

## ПРИЛОЖЕНИЕ Б

### Требования к интерфейсу пользователя

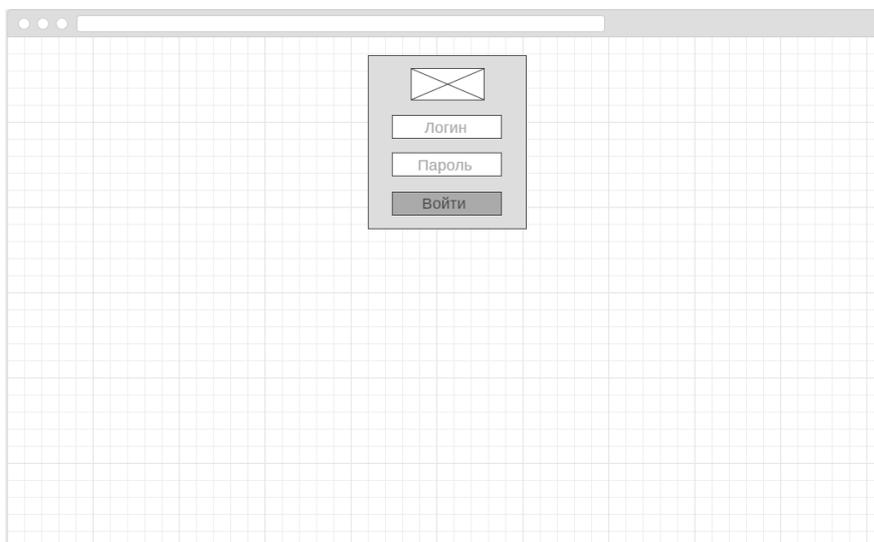


Рисунок Б.1 – Эскиз начальной страницы

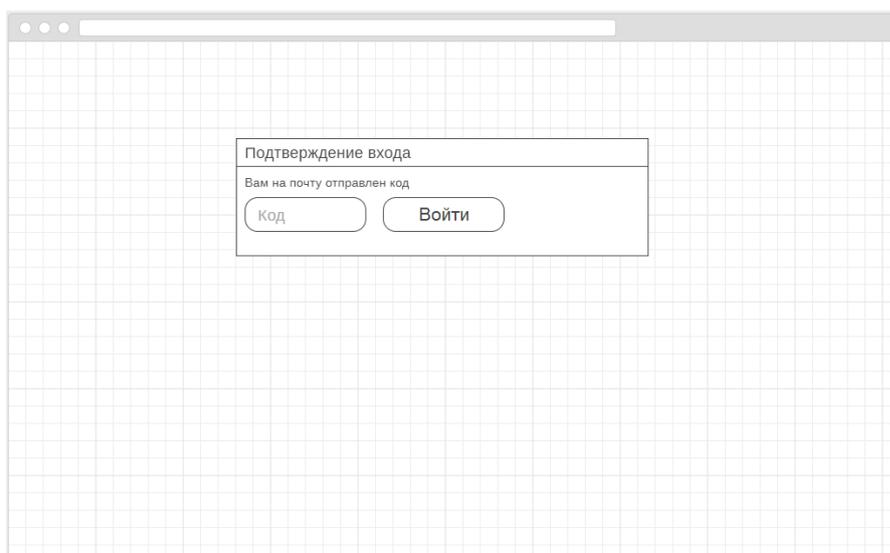


Рисунок Б.2 – Эскиз страницы ввода кода

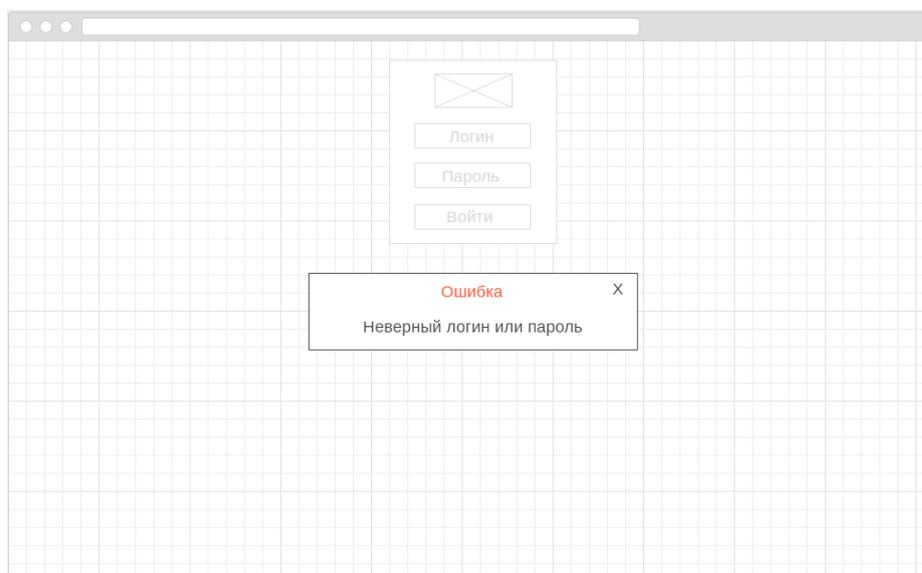


Рисунок Б.3 – Эскиз окна с ошибкой

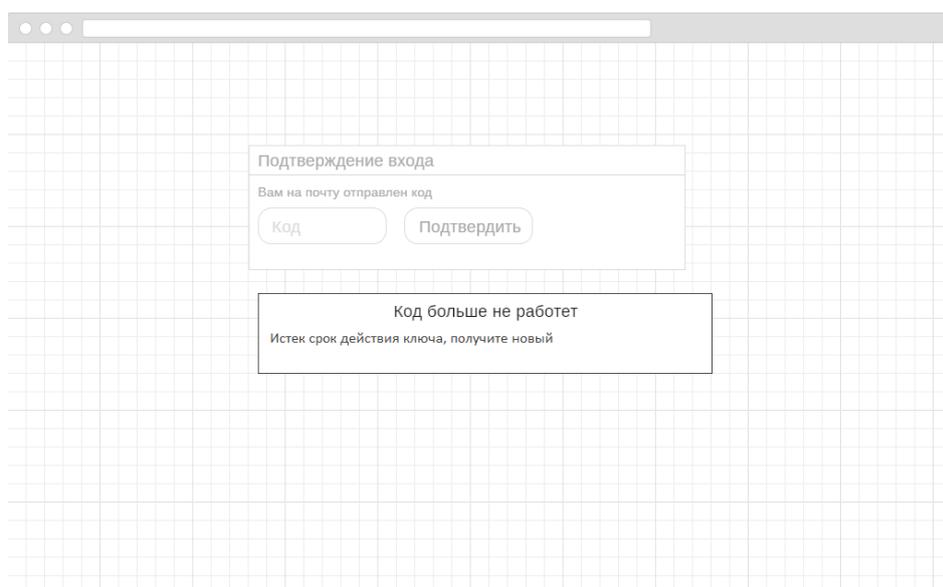


Рисунок Б.4 – Окно с сообщением о неработоспособности кода

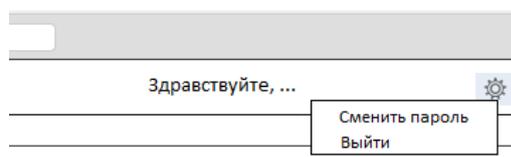


Рисунок Б.5 – Эскиз меню пользователя

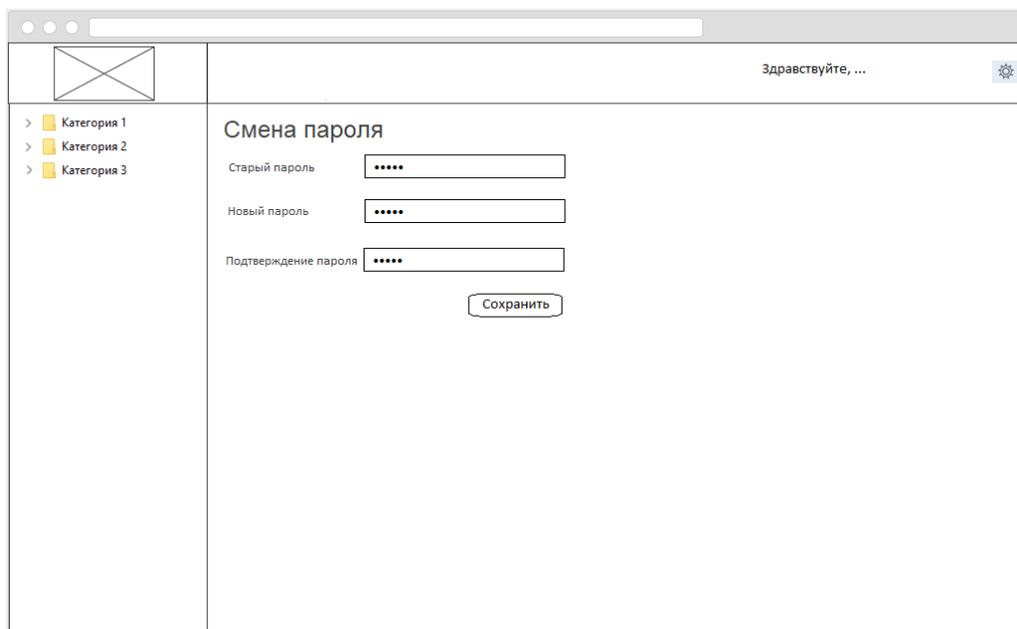


Рисунок Б.6 – Эскиз страницы смены пароля

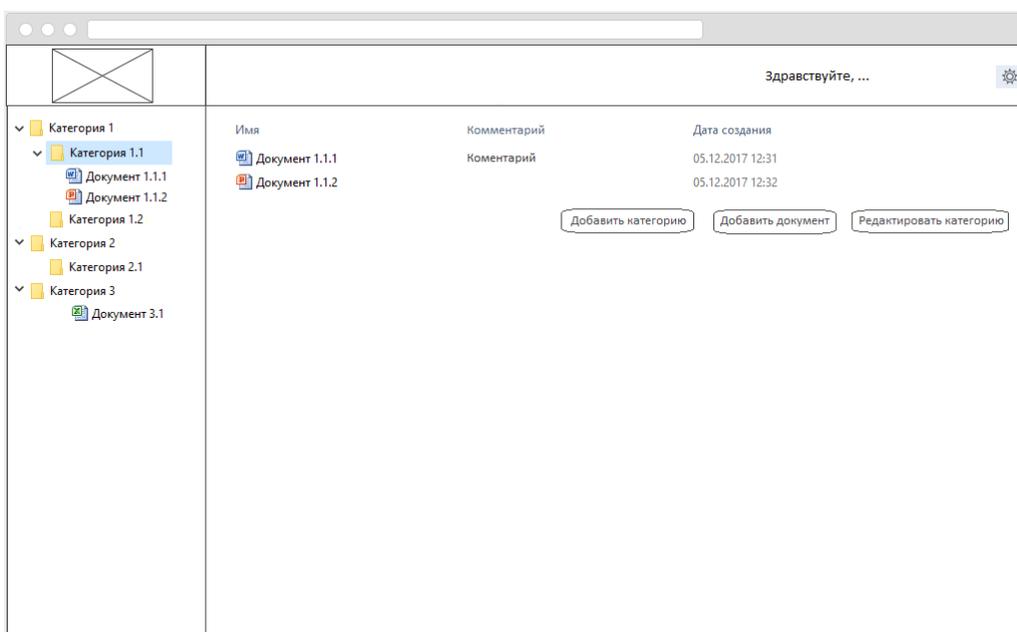


Рисунок Б.7 – Эскиз страницы просмотра категории

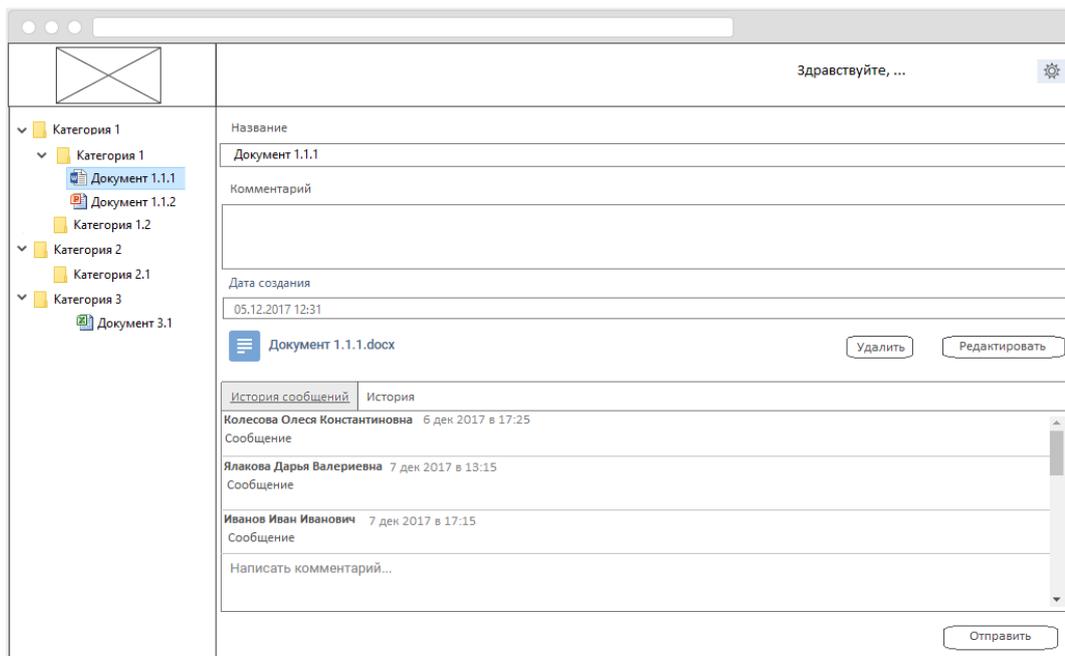


Рисунок Б.8 – Эскиз страницы просмотра карты документа

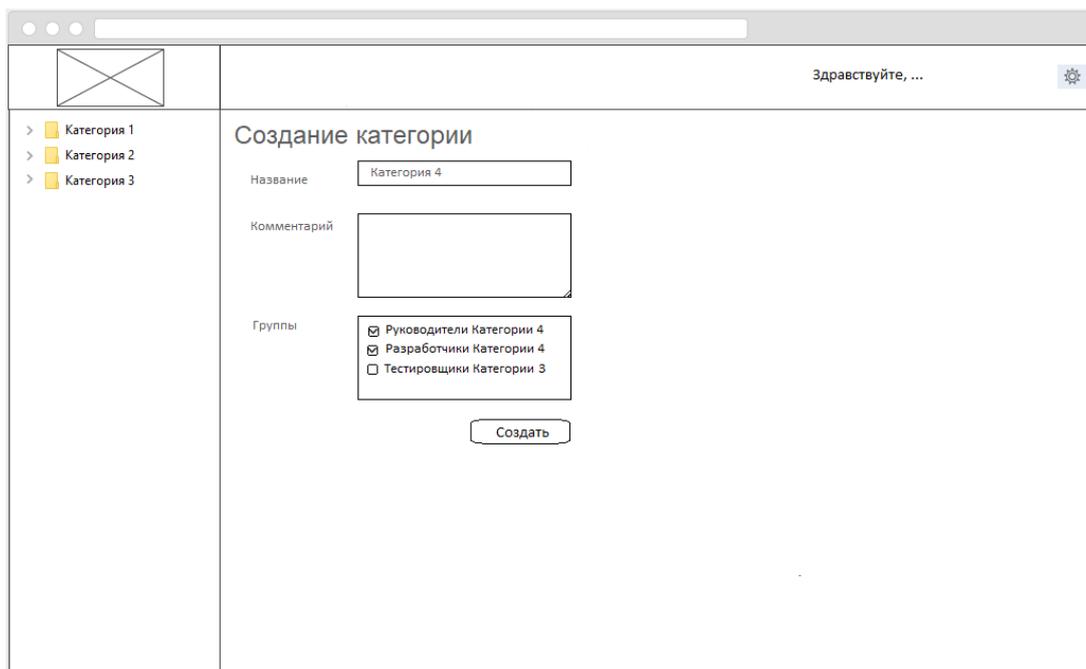


Рисунок Б.9 – Эскиз страницы создания категории

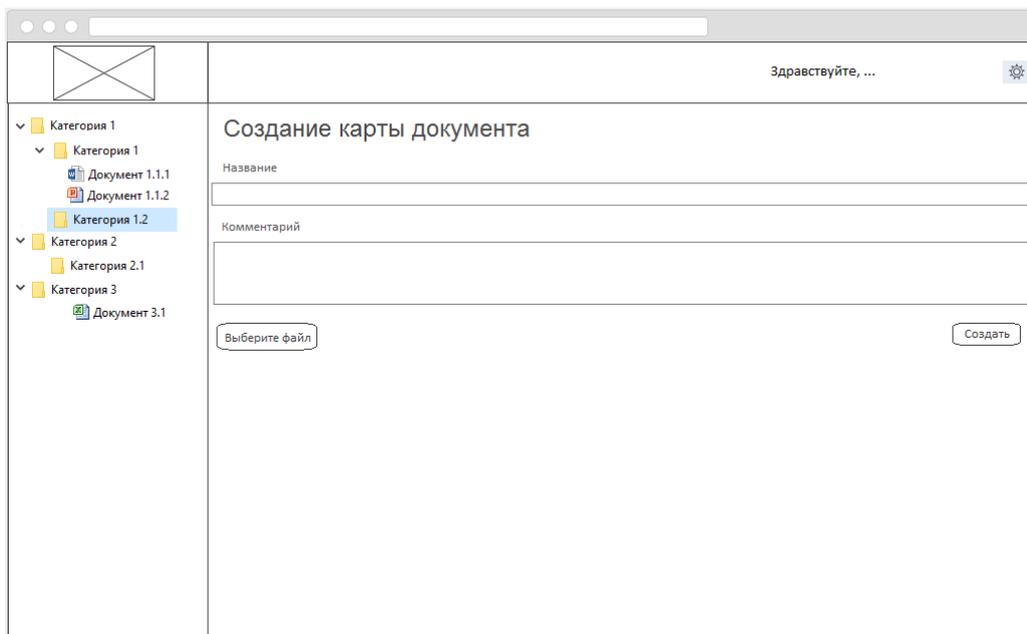


Рисунок Б.10 – Эскиз страницы создания карты документа

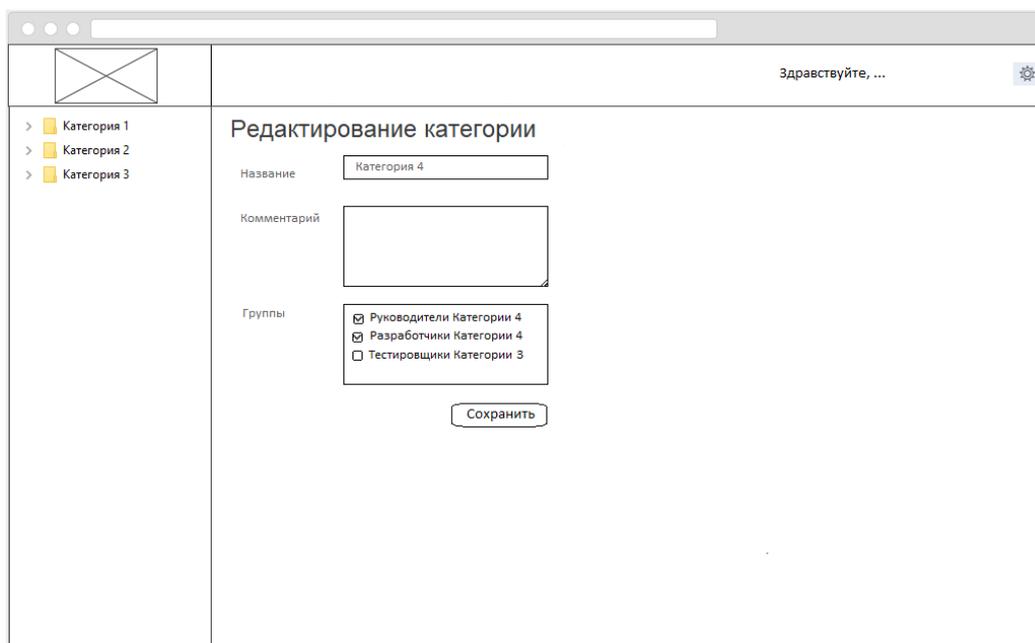


Рисунок Б.11 – Эскиз страницы редактирования категории

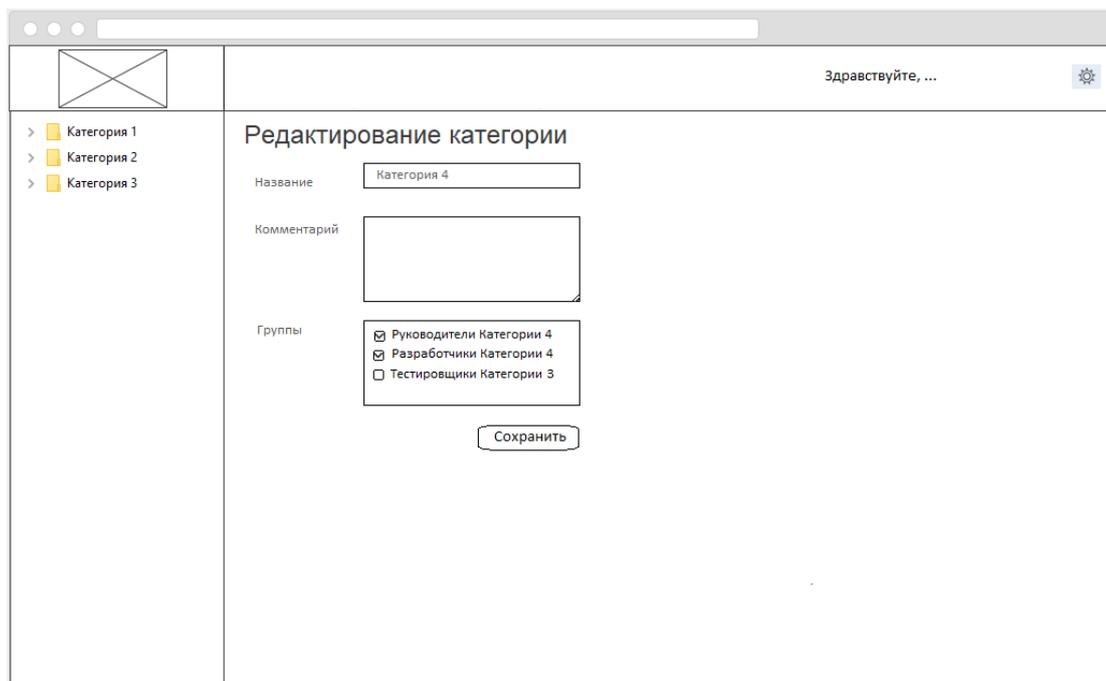
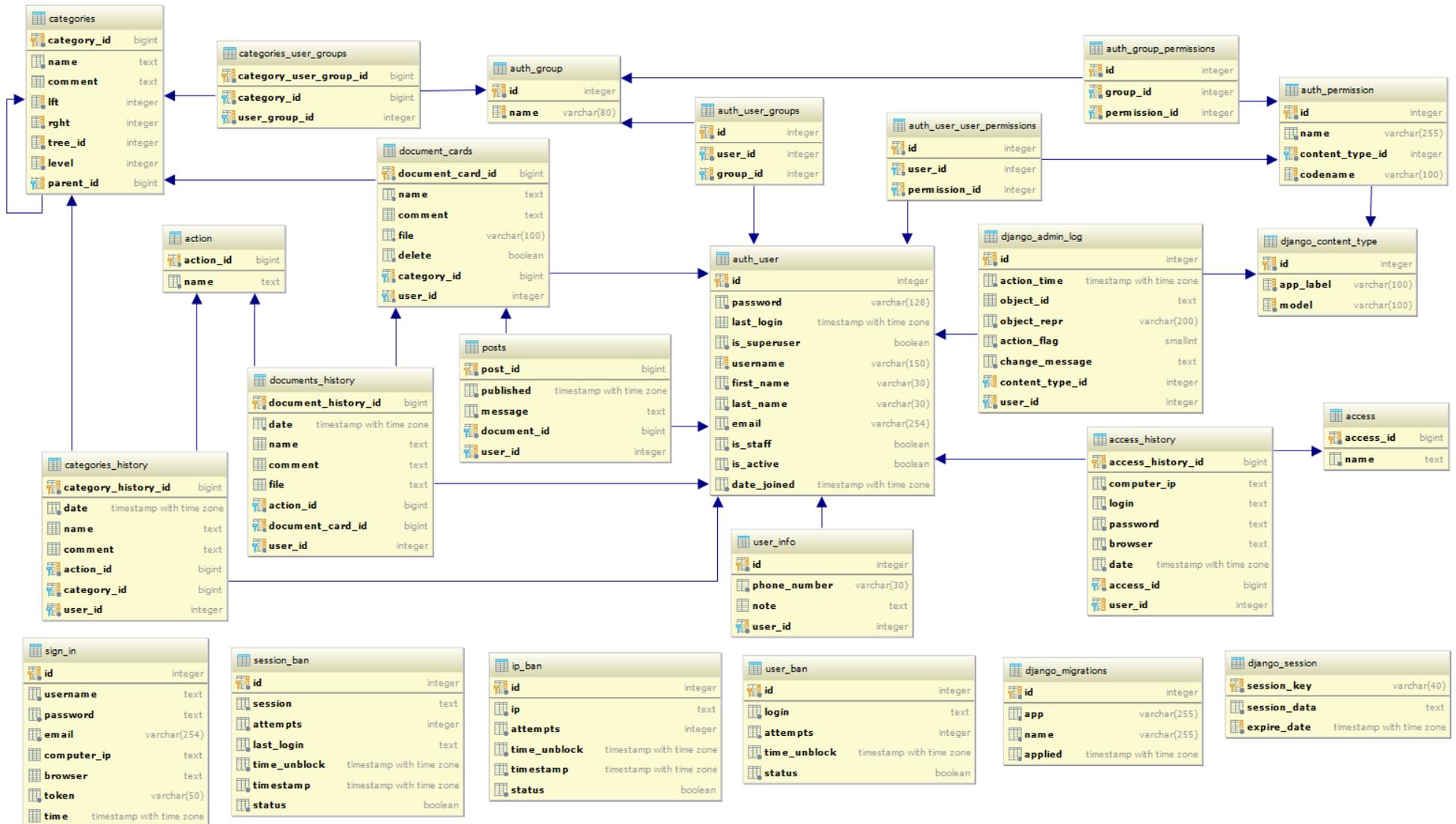


Рисунок Б.12 – Эскиз страницы редактирования карты документа

# ПРИЛОЖЕНИЕ В

## Схема базы данных





```

        <li><span><a href="/document_cards/{ { docs.document_card_id
}}"><i class="fa fa-file-o"></i>{ { docs.name }}</a></span></li>
        { % endif % }
        { % endfor % }
        { % endif % }
    </ul>
    { % elif node.is_leaf_node % }
    { % for docs in node.docs.all % }
    <ul>
        { % for docs in node.docs.all % }
        { % if docs.delete == False % }
        <li><span><a href="/document_cards/{ { docs.document_card_id
}}"><i class="fa fa-file-o"></i>{ { docs.name }}</a></span></li>
        { % endif % }
        { % endfor % }
    </ul>
    { % endfor % }
    { % endif % }
</li>
{ % endrecursetree % }
</ul>
</div>
</div>
<div class="col-xs-12 col-sm-9 col-sm-offset-3 col-md-10 col-md-offset-2 main">
    { % block info % }
    <div class="table-responsive .bs-docs-sidebar ">
        <table class="table nav">
            <thead>
                <tr>
                    <th>Название</th>
                    <th>Комментарий</th>
                    <th>Дата создания</th>
                </tr>
            </thead>
            <tbody>
                { % for parent in parents % }
                <tr>
                    <td>
                        <div class="bs-docs-sidebar hidden-print">
                            <ul class="nav bs-docs-sidenav">
                                <li><a href="/categories/{ { parent.category_id } }"><span
class="glyphicon glyphicon-folder-close"></span> { { parent.name }}</a></li>
                            </ul>
                        </div>
                    </td>
                    <td>
                        <div class="bs-docs-sidebar hidden-print">
                            <ul class="nav bs-docs-sidenav">
                                <li><a>{ { parent.comment }}</a></li>
                            </ul>
                        </div>
                    </td>
                </tr>
            </tbody>
        </table>

```

```

<td>
  <div class="bs-docs-sidebar hidden-print">
    <ul class="nav bs-docs-sidenav">
      {% for d in date %}
        {% if d.category_id == parent.category_id %}
          <li><a>{{ d.date }}</a></li>
        {% endif %}
      {% endfor %}
    </ul>
  </div>
</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
{% endblock %}
</div>
</div>
<script type="text/javascript" src="{% static 'categories/jquery.js' %}">
{% endblock %}

```

### Шаблон document\_card.html:

```

{% extends 'menu.html' %}
{% load bootstrap3 %}
{% bootstrap_pagination comment %}
{% load staticfiles %}
{% load mppt_tags %}
{% block info %}
<link href="{% static 'document_cards/style.css' %}" rel="stylesheet">
<h1 class="page-header">{{ this.name }}</h1>
{% if this.comment %}
  <div class="form-group">
    <h4 class="page-header1">Комментарий</h4>
    {{ this.comment }}
  </div>
{% endif %}
<div class="form-group">
  <h4 class="page-header1">Ссылка</h4>
  {% if ext == '.doc' or ext == '.docx' %}
    <i class="fa fa-file-word-o"></i>
  {% elif ext == '.xls' or ext == '.xlsx' %}
    <i class="fa fa-file-excel-o"></i>
  {% elif ext == '.ppt' or ext == '.pptx' %}
    <i class="fa fa-file-powerpoint-o"></i>
  {% elif ext == '.mp3' or ext == '.flac' or ext == '.ape' or ext == '.ogg' or ext == '.wav' or ext ==
'.ac3' or ext == '.wma' or ext == '.m4a' or ext == '.aac' %}
    <i class="fa fa-file-audio-o"></i>
  {% elif ext == '.bmp' or ext == '.jpg' or ext == '.jpeg' or ext == '.png' or ext == '.gif' or ext ==
'.tiff' or ext == '.ico' or ext == '.raw' %}

```

```

        <i class="fa fa-file-image-o"></i>
        {% elif ext == '.avi' or ext == '.wmw' or ext == '.mkv' or ext == '.3gp' or ext == '.flv' or ext ==
'.mpeg' or ext == '.mp4' or ext == '.mov' or ext == '.vob' %}
        <i class="fa fa-file-video-o"></i>
        {% elif ext == '.rar' or ext == '.zip' or ext == '.7z' or ext == '.tar' or ext == '.gzip' or ext == '.gz'
or ext == '.jar' %}
        <i class="fa fa-file-archive-o"></i>
        {% elif ext == '.pdf' %}
        <i class="fa fa-file-pdf-o"></i>
        {% elif ext == '.exe' %}
        <i class="fa fa-file-code-o"></i>
        {% else %}
        <i class="fa fa-file"></i>
        {% endif %}
        <a href="{{ this.file.url }}" class="href">{{ doc_name }}</a>
</div>
<div class="form-group">
    <h4 class="page-header1">Дата создания</h4>
    {% for date in date %}
        {% if date.document_card_id == this.document_card_id %}
            {{ date.date }}
        {% endif %}
    {% endfor %}
</div>
<div class="text-right mrg-top-30">
    <a class="btn btn-default" href="/document_cards/{{ this.document_card_id
}}/edit_document_card" role="button">Редактировать</a>
    <a class="btn btn-default" href="/document_cards/{{ this.document_card_id
}}/delete_document_card" role="button">Удалить</a>
</div>
<div class="form-group form-comment" role="tabpanel" >
    <ul class="nav nav-tabs" id="myTab" role="tablist">
        <li class="nav-item active">
            <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab"
aria-controls="home" aria-expanded="true">Комментарии ({{ posts.count }})</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab" aria-
controls="profile">История</a>
        </li>
    </ul>
    <div class="tab-content" id="myTabContent">
        <div class="tab-pane fade in active" role="tabpanel" id="home" aria-labelledby="home-
tab">
            {% if posts.count == 0%}
                Комментариев нет
                <hr>
            {% else %}
                <div class="comments">
                    <ul class="media-list">
                        {% for comment in comment %}
                            <li class="media">

```

```

<div class="media-body">
  <div class="media-heading">
    {% for user in users %}
      {% if comment.user_id == user.id %}
        <div class="author">{{ user.last_name }} {{ user.first_name
}}</div>
          {% endif %}
        {% endfor %}
      <div class="metadata">
        <span class="date">{{ comment.published }}</span>
      </div>
    </div>
    <div class="media-text text-justify">{{ comment.message }}</div>
    <hr>
  </div>
</li>
{% endfor %}
</ul>
</div>
<div class="pagination">
  <span class="step-links">
    {% if comment.has_previous %}
      <a href="?page={{ comment.previous_page_number }}"><span
class="glyphicon glyphicon-chevron-left"></span></a>
    {% endif %}
    <span class="current">
      Страница {{ comment.number }} из {{ comment.paginator.num_pages }}
    </span>
    {% if comment.has_next %}
      <a href="?page={{ comment.next_page_number }}"><span class="glyphicon
glyphicon-chevron-right"></span></a>
    {% endif %}
  </span>
</div>
{% endif %}
<div class="comment">
  <h4 class="title-comments">Добавить комментарий</h4>
  <form class="form_comment" action="/main/document_card/{{
this.document_card_id }}/addpost" method="post">
    {% csrf_token %}
    <div class="form-group">
      <textarea name="message" class="form-control" placeholder="Текст
комментария" rows="4"></textarea>
    </div>
    <button class="btn btn-default" type="submit">Отправить</button>
  </form>
</div>
</div>
<div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-tab">
  <div class="comments">
    <ul class="media-list">
      {% for history in history %}

```

```

<li class="media">
  <div class="media-body">
    <div class="media-heading">
      {% for user in users %}
        {% if history.user_id == user.id %}
          <div class="author">{{ user.last_name }} {{ user.first_name }}</div>
        {% endif %}
      {% endfor %}
    <div class="metadata">
      <span class="date">{{ history.date }}</span>
    </div>
  </div>
  <div class="media-text text-justify">
    {% if history.action_id == 1 %}
      Пользователь создал документ: Название: {{ history.name }},
      Комментарий: {{ history.comment }}, Файл: {{ history.file }}
    {% elif history.action_id == 2 %}
      Пользователь отредактировал документ: Название: {{ history.name
    }}, Комментарий: {{ history.comment }}, Файл: {{ history.file }}
    {% elif history.action_id == 4 %}
      Пользователь скачал документ {{ history.file }}
    {% endif %}
  </div>
  <hr>
</div>
</li>
{% endfor %}
</ul>
</div>
</div>
</div>
</div>
<script type="text/javascript" src="{% static 'document_cards/jquery.js' %}">
{% endblock %}

```

## ПРИЛОЖЕНИЕ Д

### Разработанный интерфейс

На рисунках Д.1 – Д.16 представлены интерфейсы всех страниц системы в браузере Google Chrome.

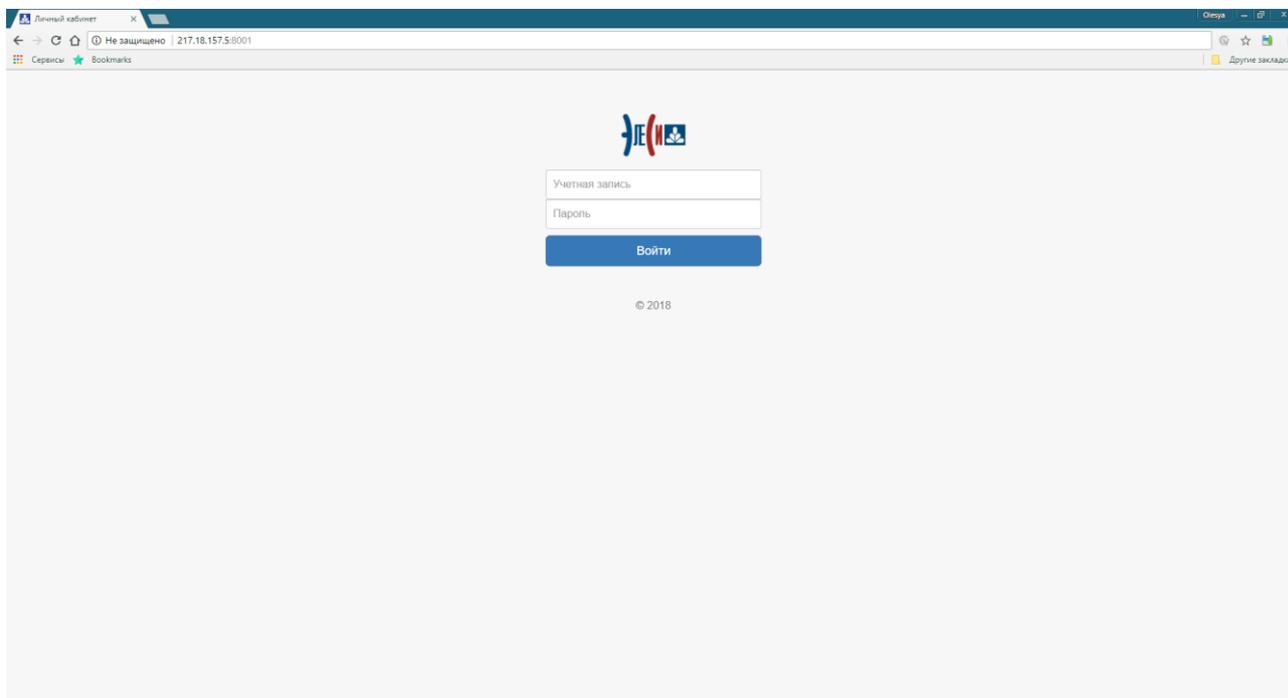


Рисунок Д.1 – Страница ввода логина и пароля

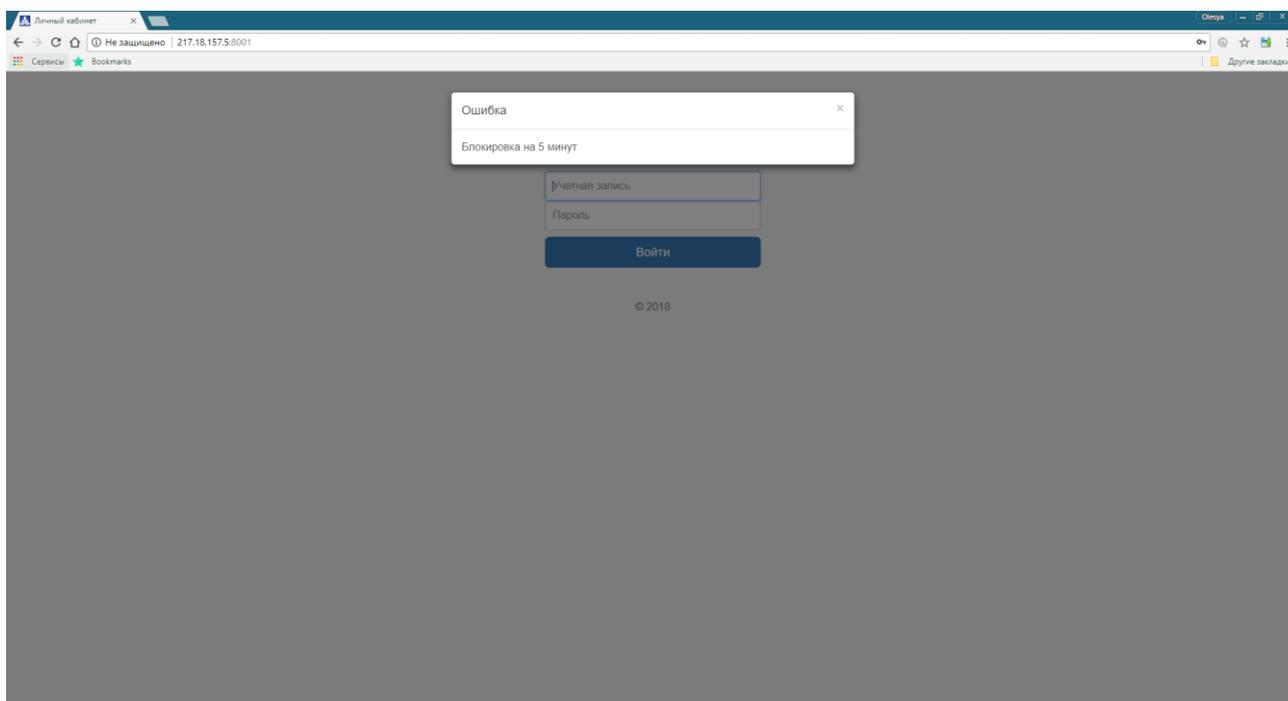


Рисунок Д.2 – Страница срабатывания блокировки при неверном вводе логина или пароля

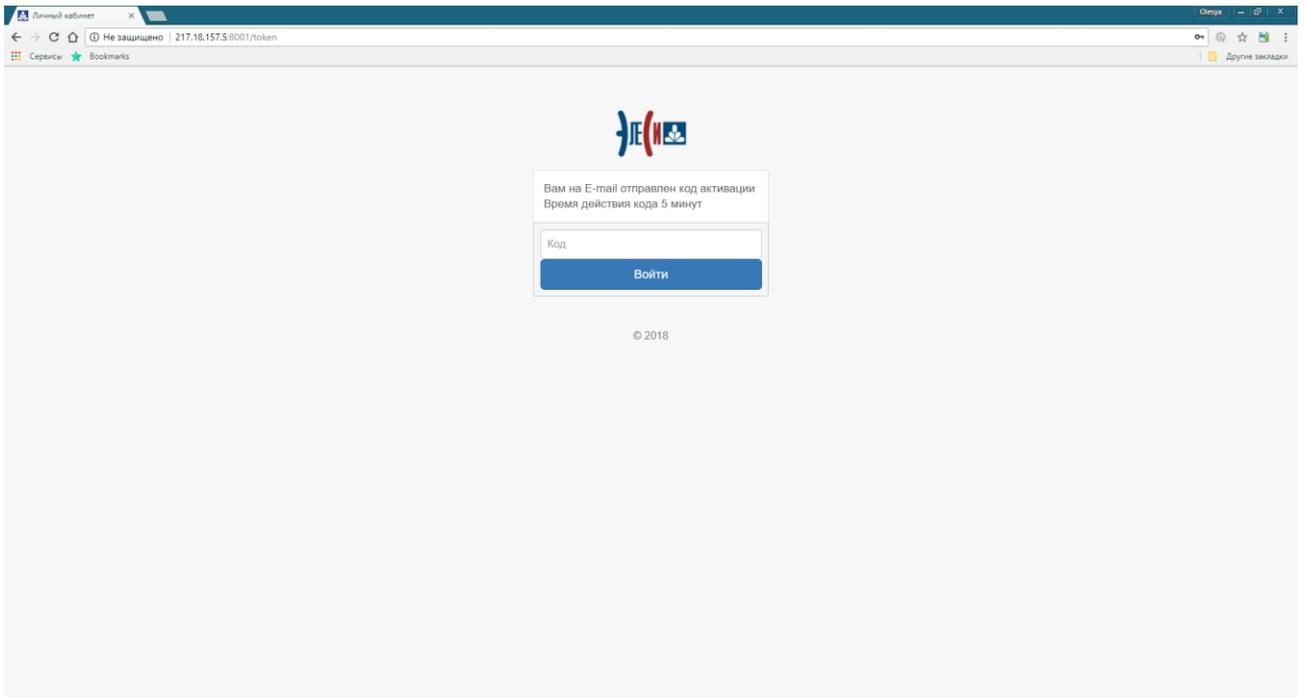


Рисунок Д.3 – Страница ввода кода активации

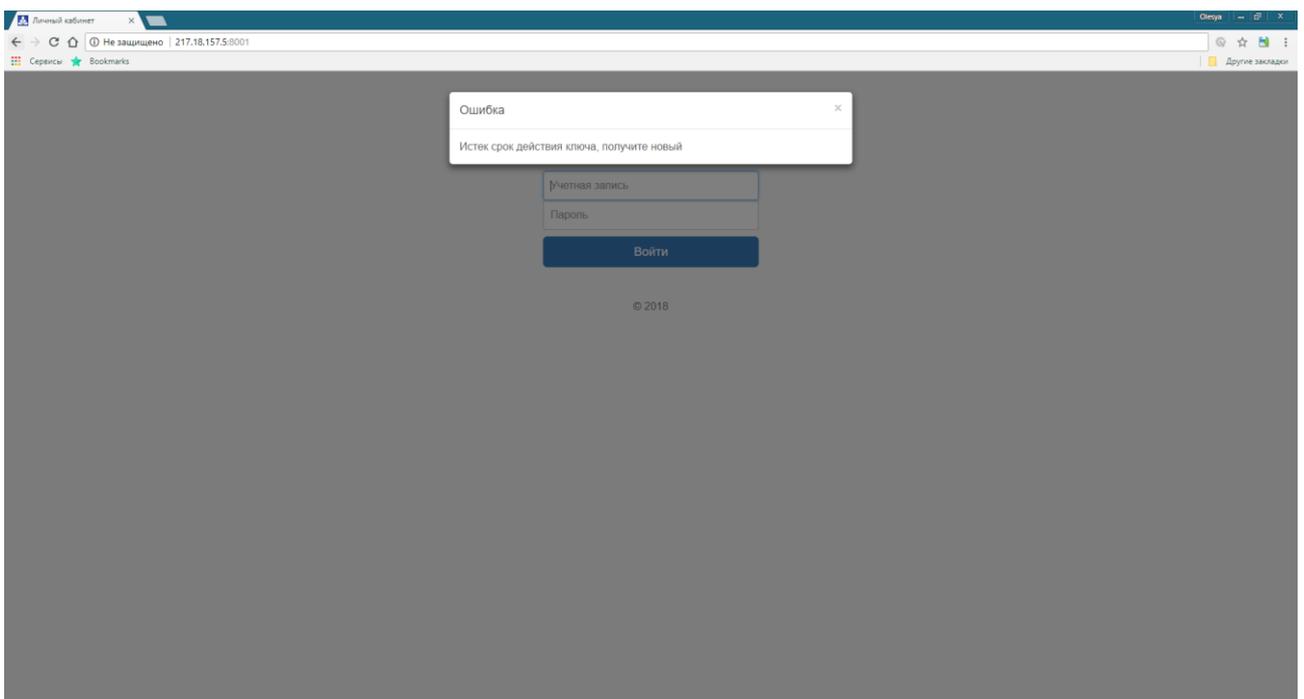


Рисунок Д.4 – Страница ошибки ввода кода активации

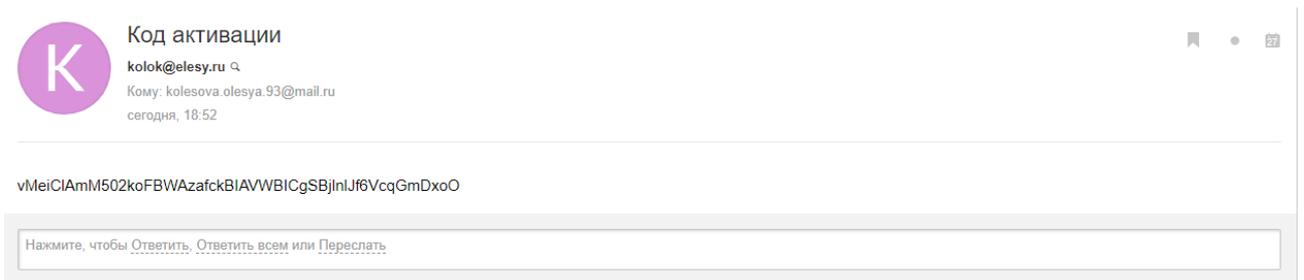


Рисунок Д.5 – Сообщение на почте с кодом активации

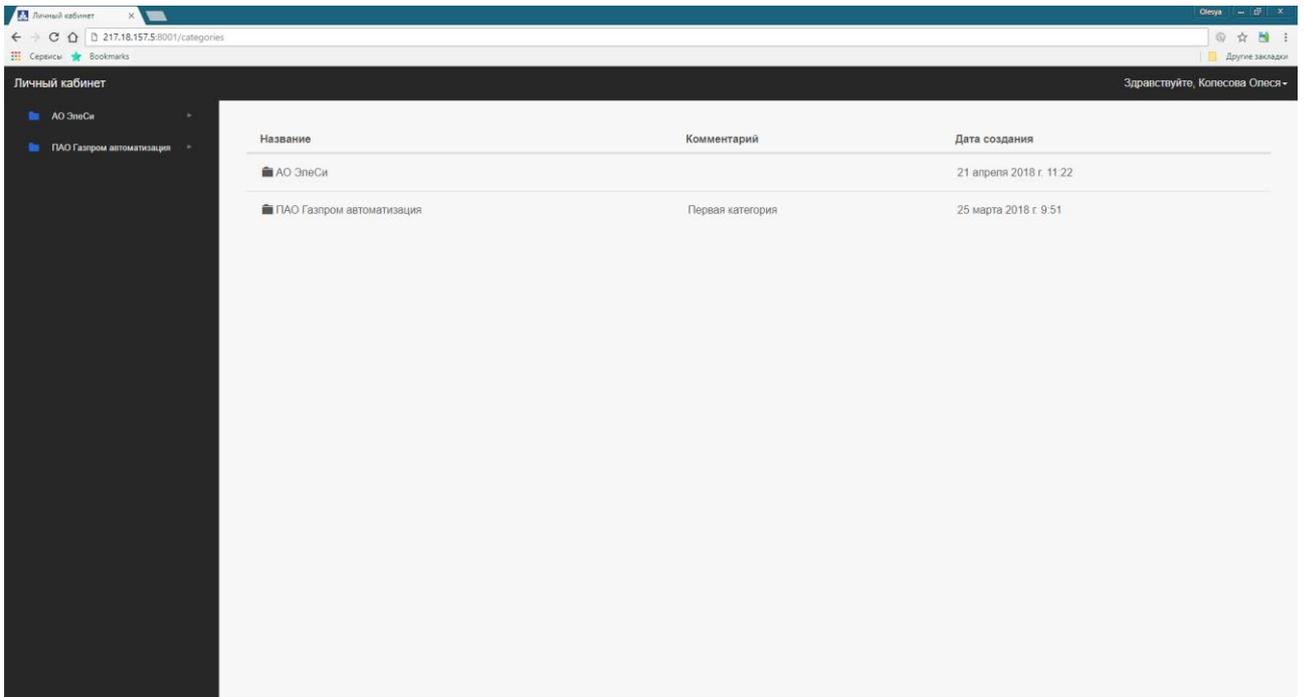


Рисунок Д.6 – Главная страница с родительскими категориями

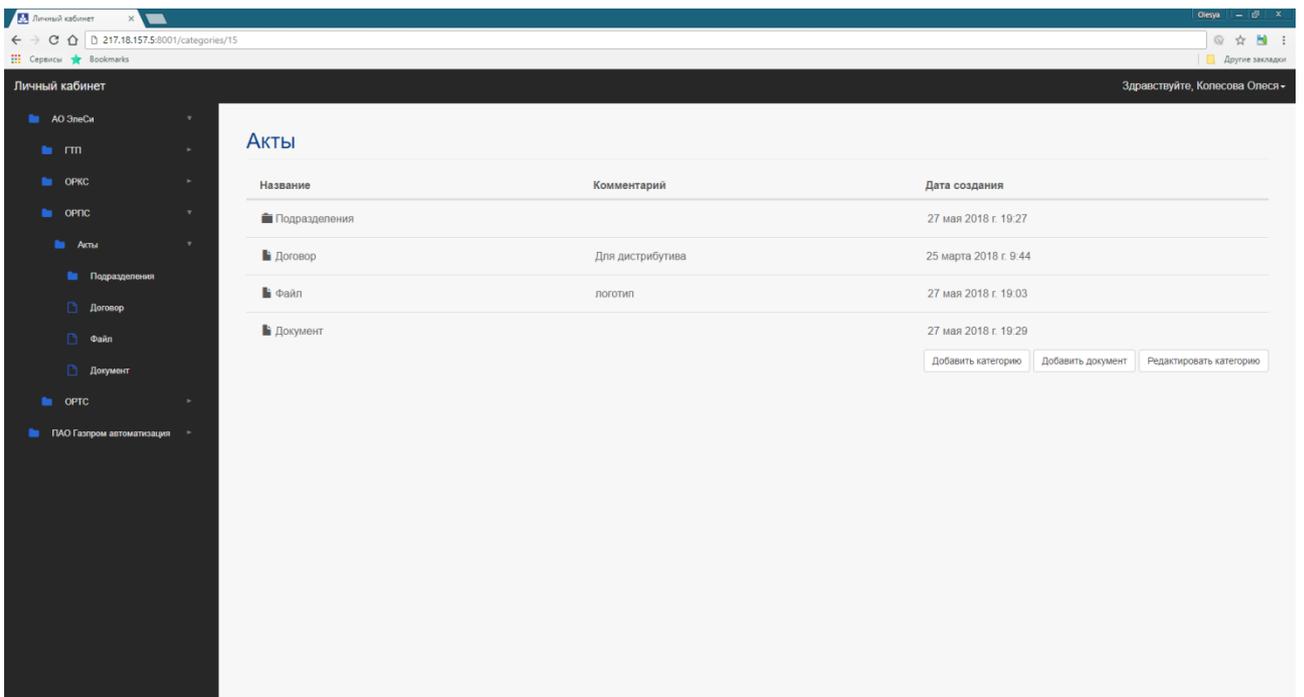


Рисунок Д.7 – Страница просмотра содержимого категории

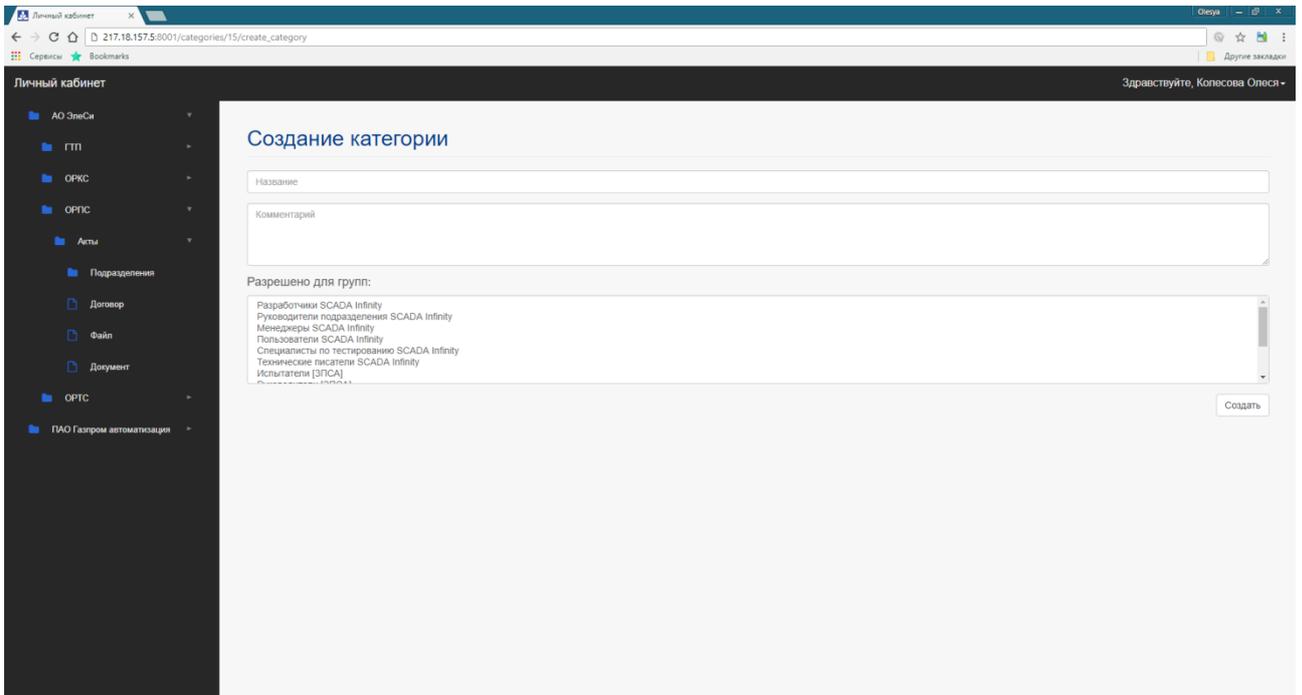


Рисунок Д.8 – Страница создания категории

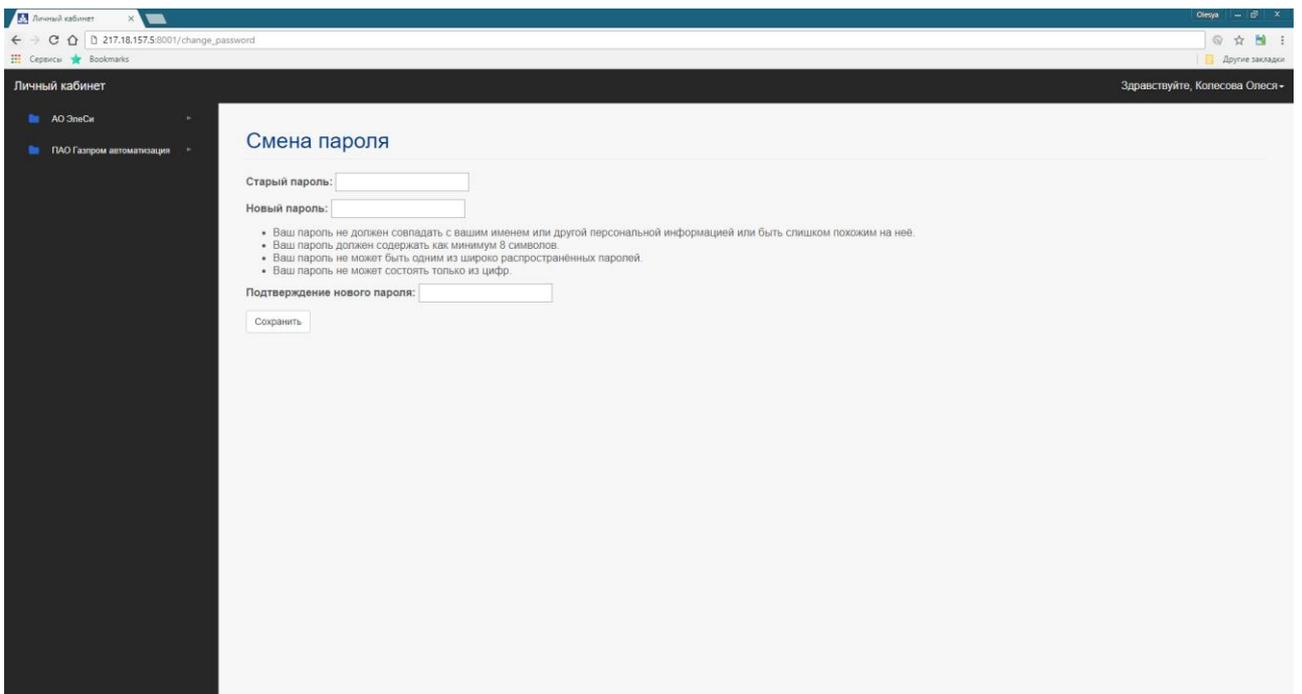


Рисунок Д.9 – Страница смены пароля

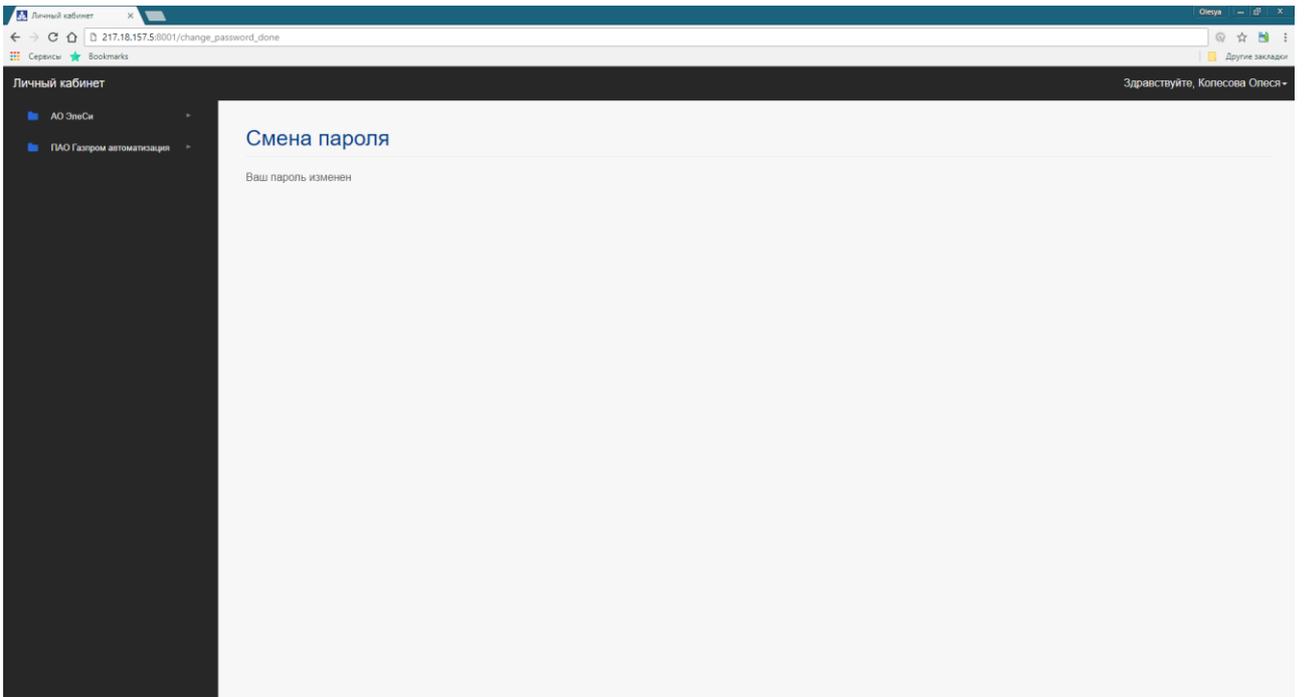


Рисунок Д.10 – Страница подтверждения смены пароля

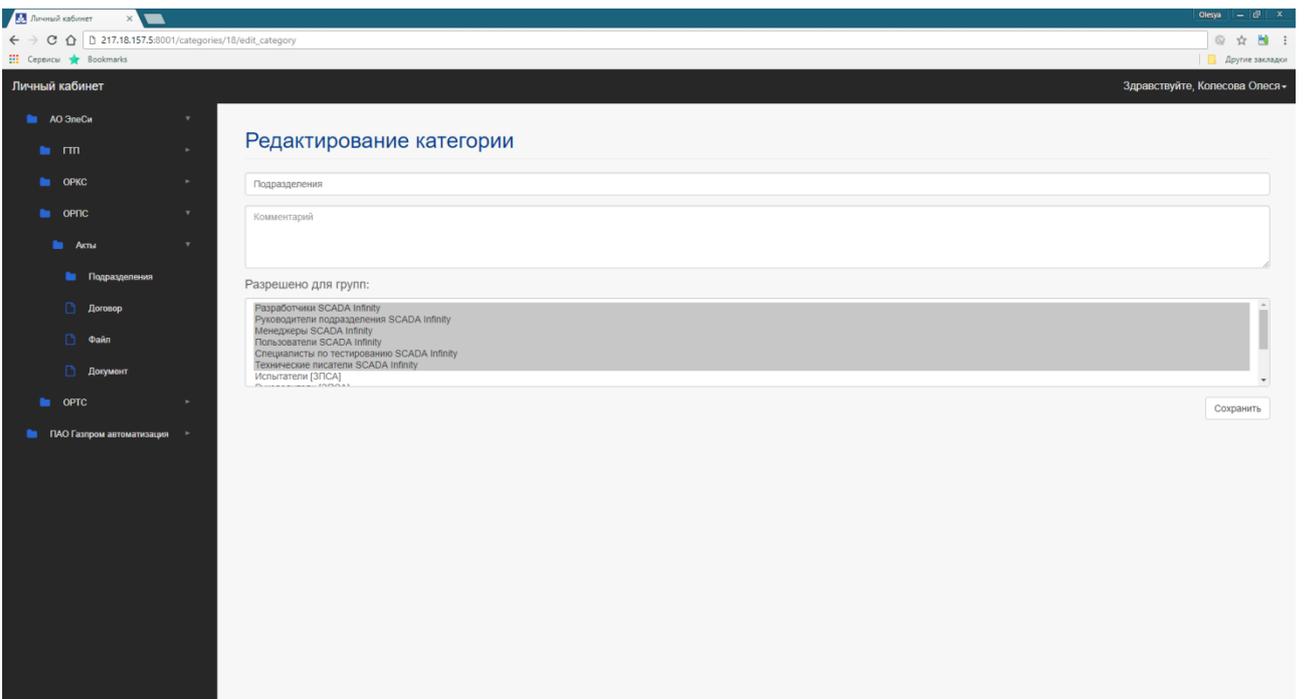


Рисунок Д.11 – Страница редактирования категории

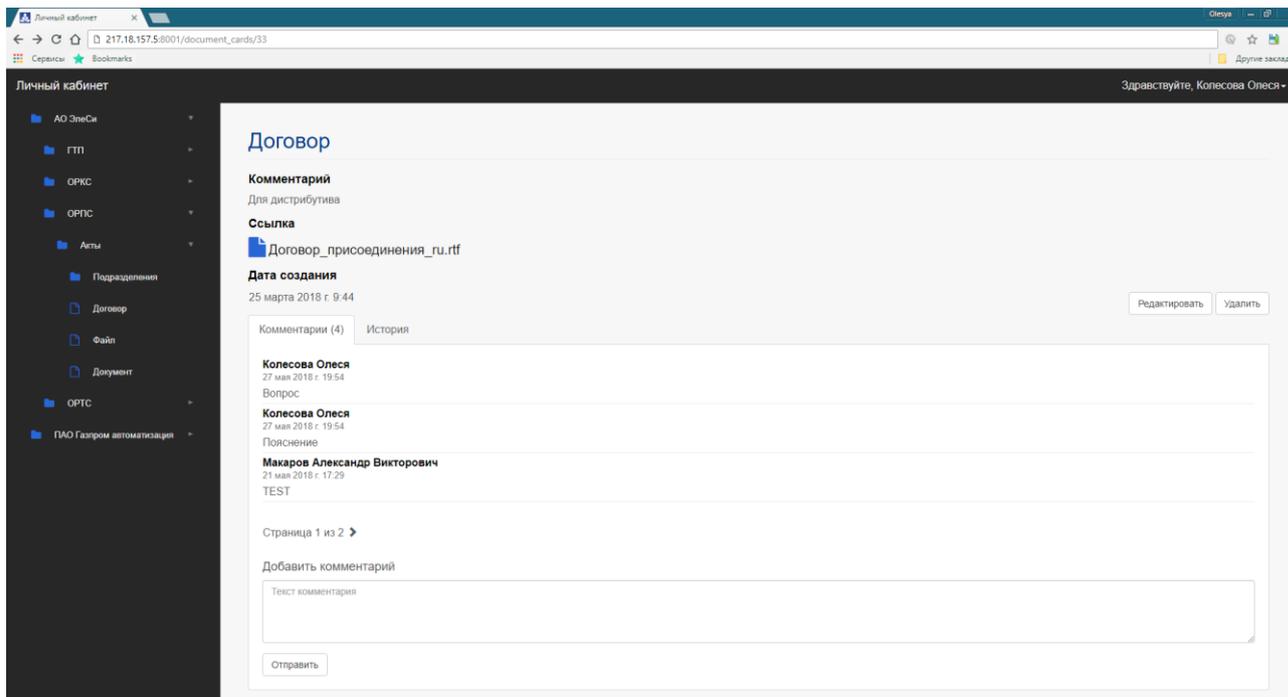


Рисунок Д.12 – Страница просмотра карты документа

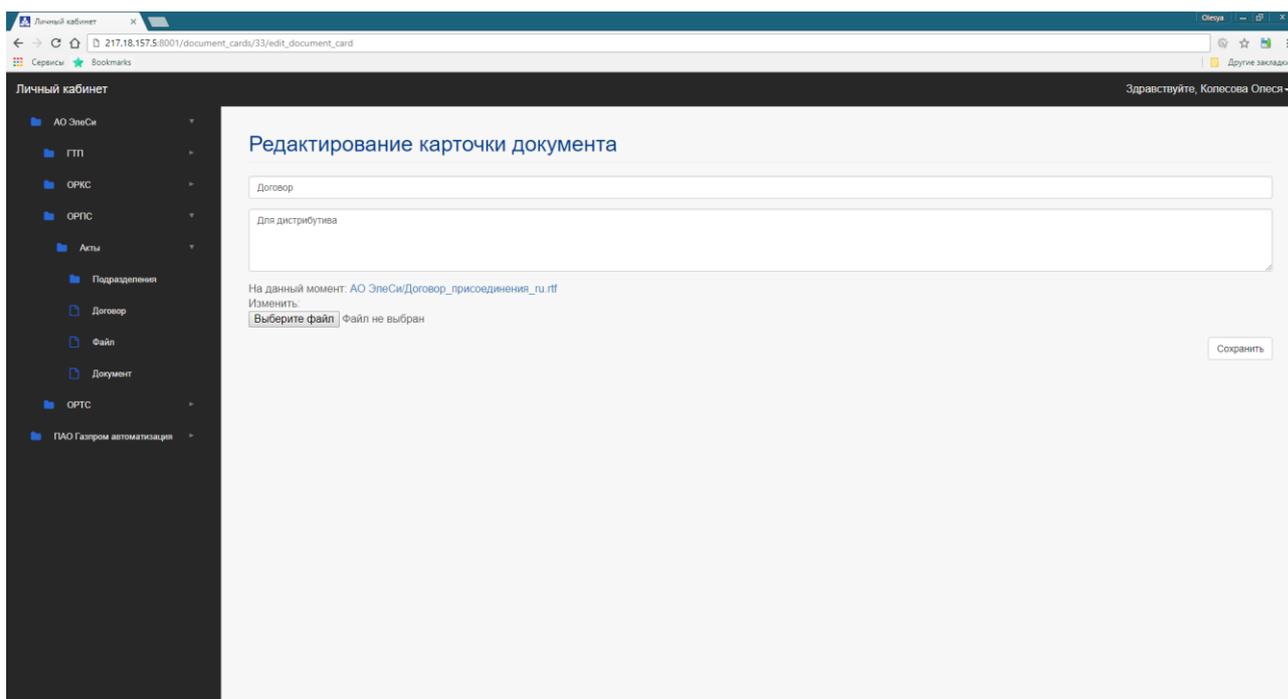


Рисунок Д.13 – Страница редактирования карты документа

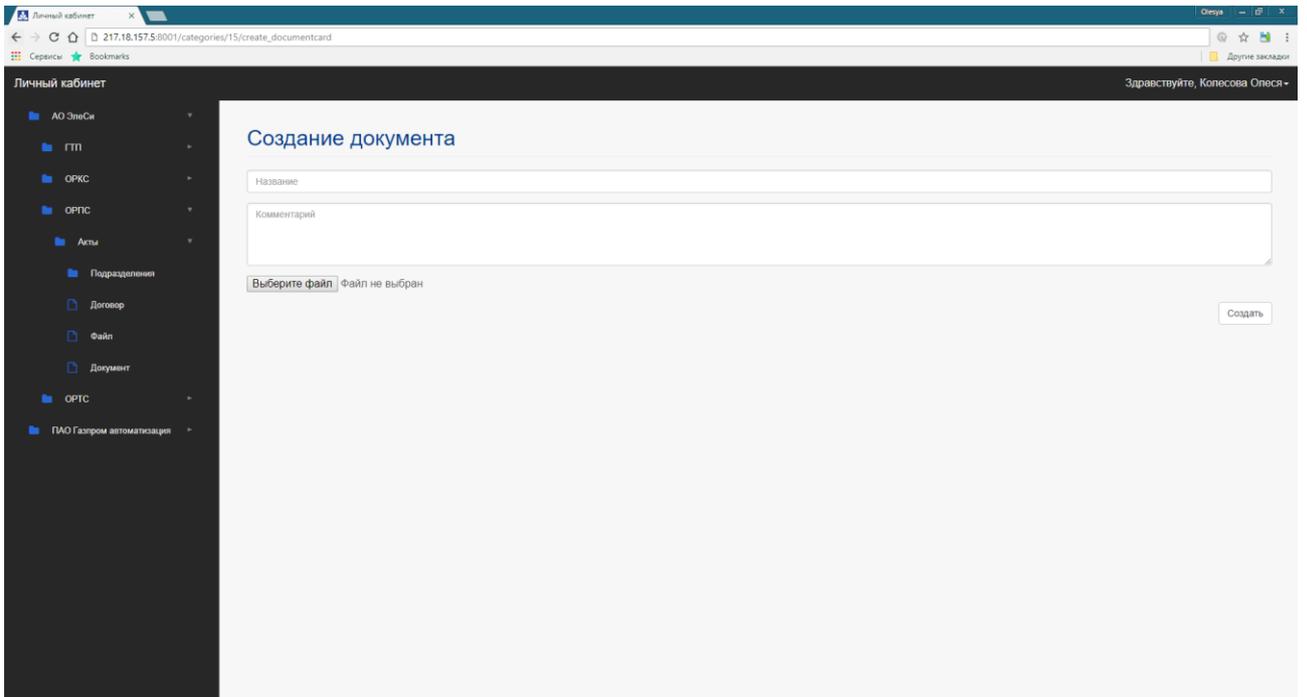


Рисунок Д.14 – Страница создания карты документа

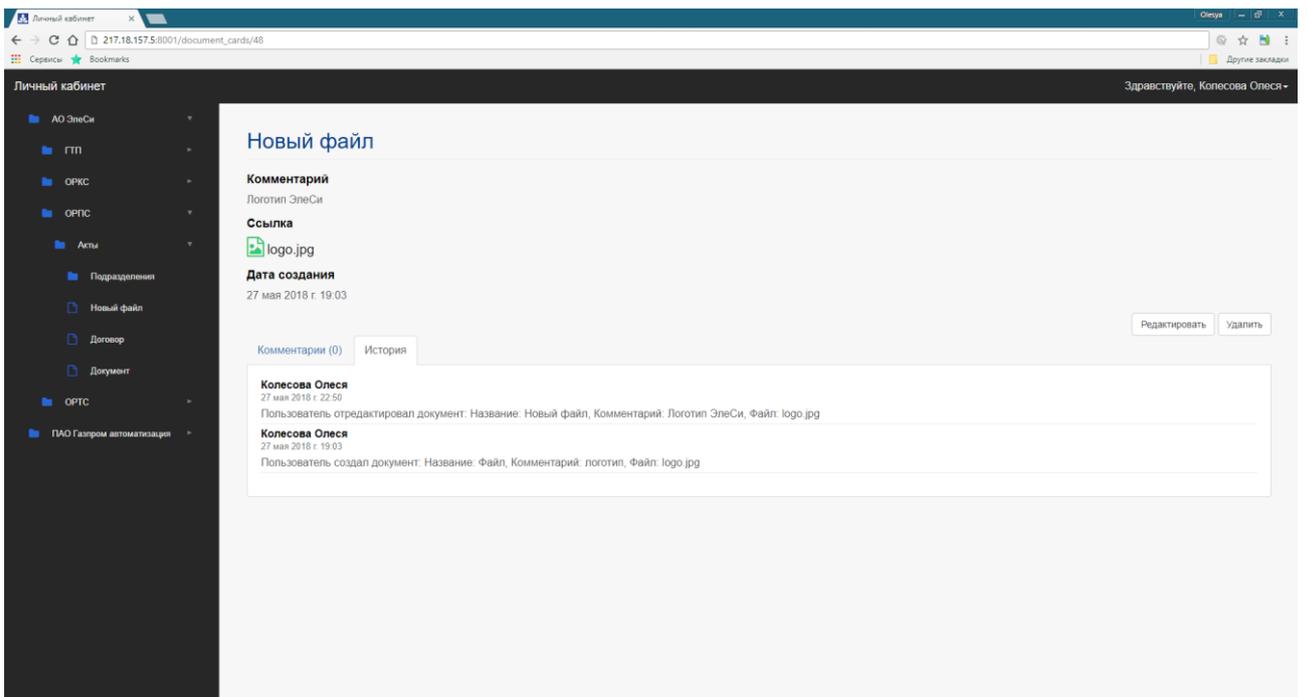


Рисунок Д.15 – Вкладка просмотра истории документа

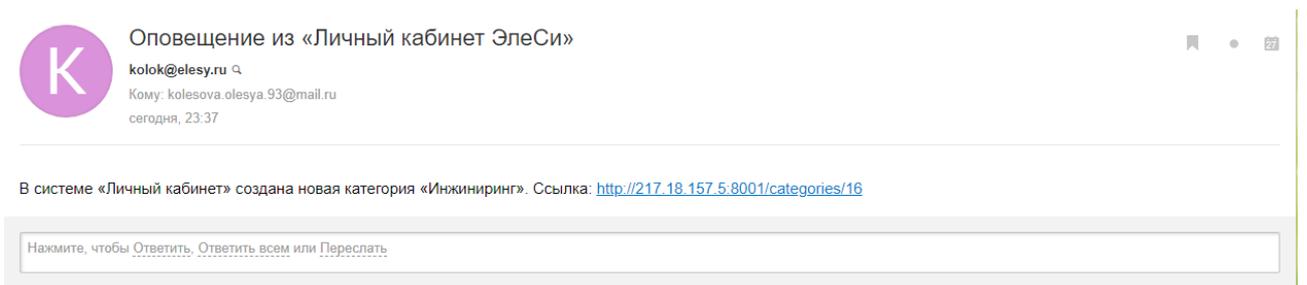


Рисунок Д.16 – Сообщение с оповещением на почте

На рисунках Д.17 – Д.20 представлены примеры отображения страниц с телефона под управлением Android 7.1.2.

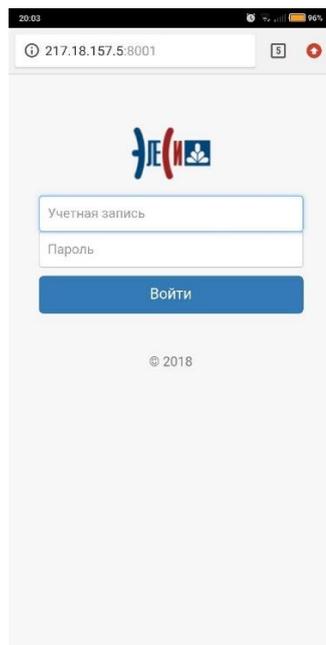


Рисунок Д.17 – Страница ввода логина и пароля, открытая на телефоне

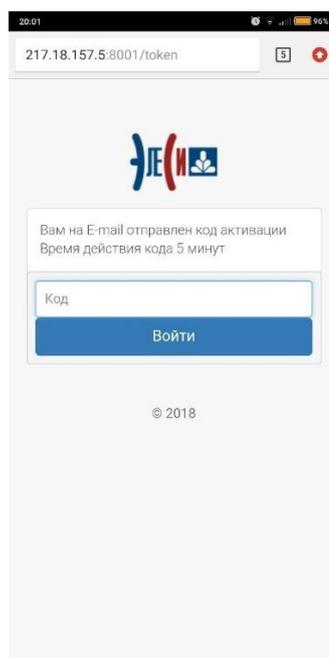


Рисунок Д.18 – Страница ввода кода активации, открытая на телефоне

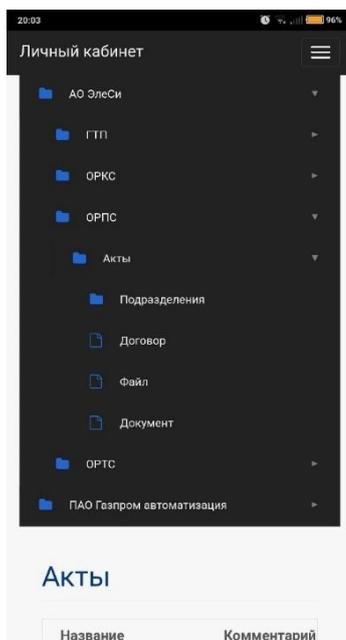


Рисунок Д.19 – Отображение дерева объектов на телефоне

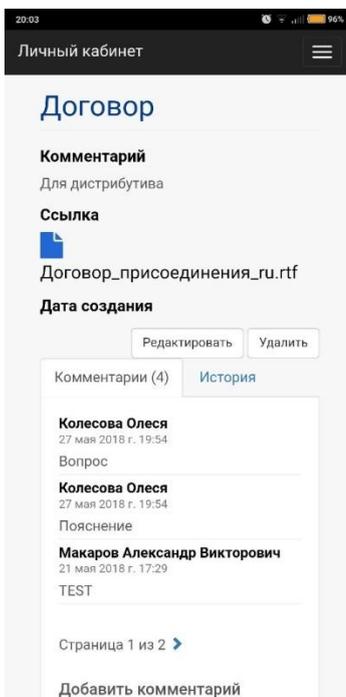


Рисунок Д.20 – Страница просмотра карты документа, открытая на телефоне

На рисунках Д.21 – Д.23 представлены примеры отображения страницы просмотра карты документа на разных браузерах.

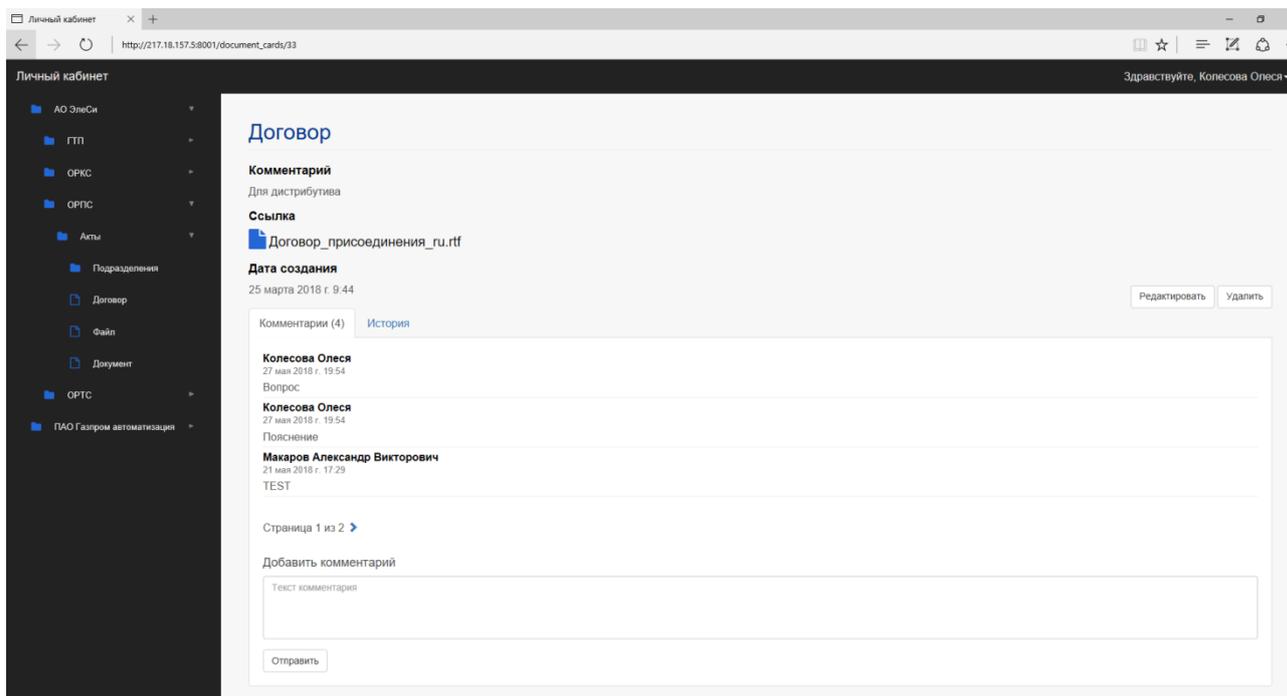


Рисунок Д.21 – Страница просмотра карты документа в браузере Microsoft Edge

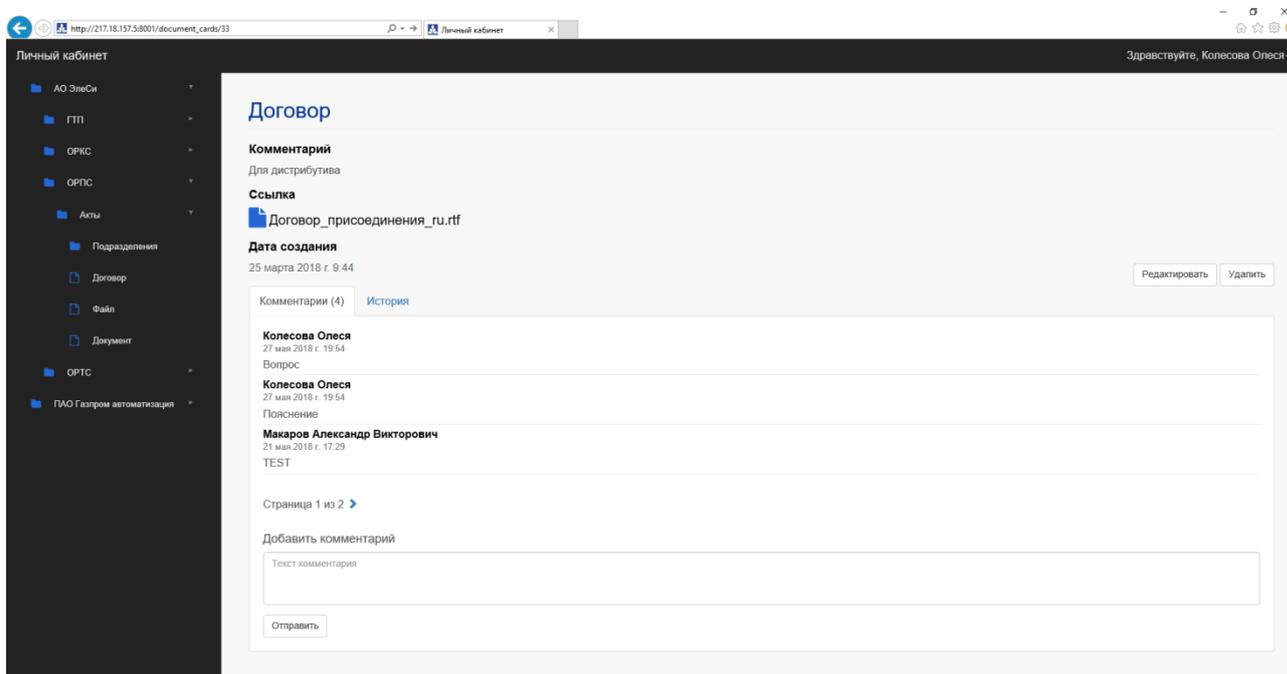


Рисунок Д.22 – Страница просмотра карты документа в браузере Internet Explorer

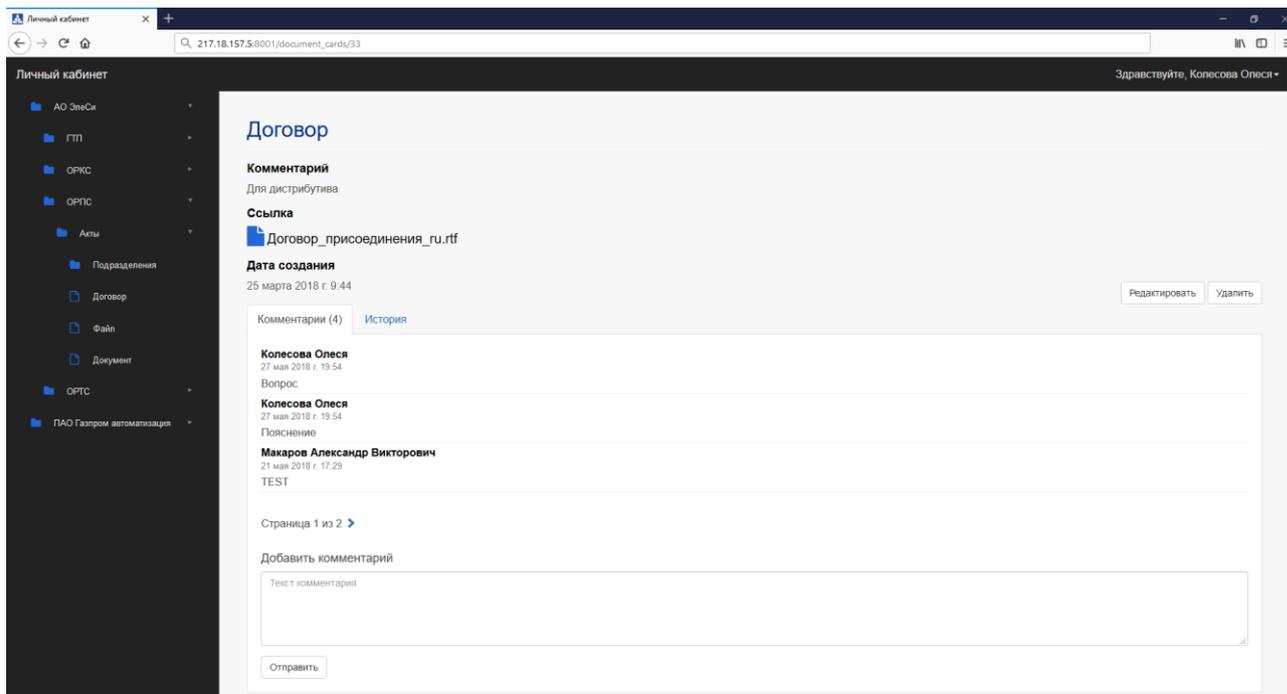


Рисунок Д.23 – Страница просмотра карты документа в браузере Firefox Quantum

пользователь	IP-адрес компьютера	введенный логин	введенный пароль или код	браузер	дата	уровень доступа
olesya	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	28 мая 2018 г. 7:39	доступ разрешен
-	79.136.161.40	olesya		Mozilla/5.0 (Linux; Android 7.1.2; Redmi 5 Plus Build/N2G47H) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.137 Mobile Safari/537.36	27 мая 2018 г. 20:00	доступ запрещен, неверный логин или пароль
olesya	79.136.161.40	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	27 мая 2018 г. 18:57	доступ разрешен
-	79.136.161.40	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	27 мая 2018 г. 18:49	доступ запрещен, неверный логин или пароль
olesya	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	25 мая 2018 г. 8:22	доступ разрешен
Makarov	10.14.0.82	Makarov		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	24 мая 2018 г. 13:53	доступ разрешен
olesya	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0	24 мая 2018 г. 10:10	доступ разрешен
olesya	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	24 мая 2018 г. 9:43	доступ разрешен
-	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	24 мая 2018 г. 9:38	доступ запрещен, неверный логин или пароль
olesya	79.136.161.40	olesya	pbkof2_sha25653000586lk6oLcXK3sz7m0G7MM0dLzNhf1tjx69jckJ0DgCU05G-VLMc-	Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	23 мая 2018 г. 21:39	доступ запрещен, неверный код
olesya	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	22 мая 2018 г. 10:41	доступ разрешен
-	10.14.0.62	olesya		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	22 мая 2018 г. 10:33	доступ запрещен, неверный логин или пароль
Makarov	10.14.0.82	Makarov		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	21 мая 2018 г. 17:27	доступ разрешен
-	10.14.0.82	Makarov		Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36	21 мая 2018 г. 17:08	доступ запрещен, неверный логин или пароль

Рисунок Д.24 – Страница просмотра истории доступа из панели администратора

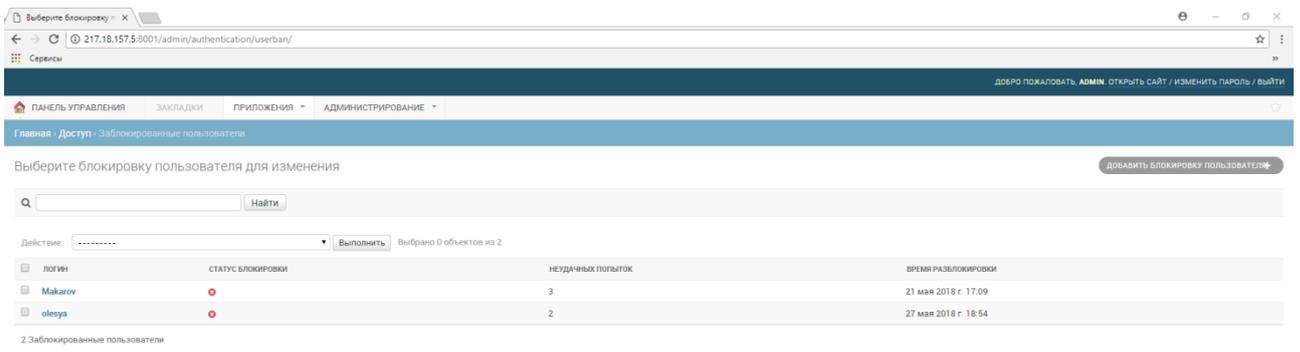


Рисунок Д.25 – Страница просмотра заблокированных пользователей из панели администратора

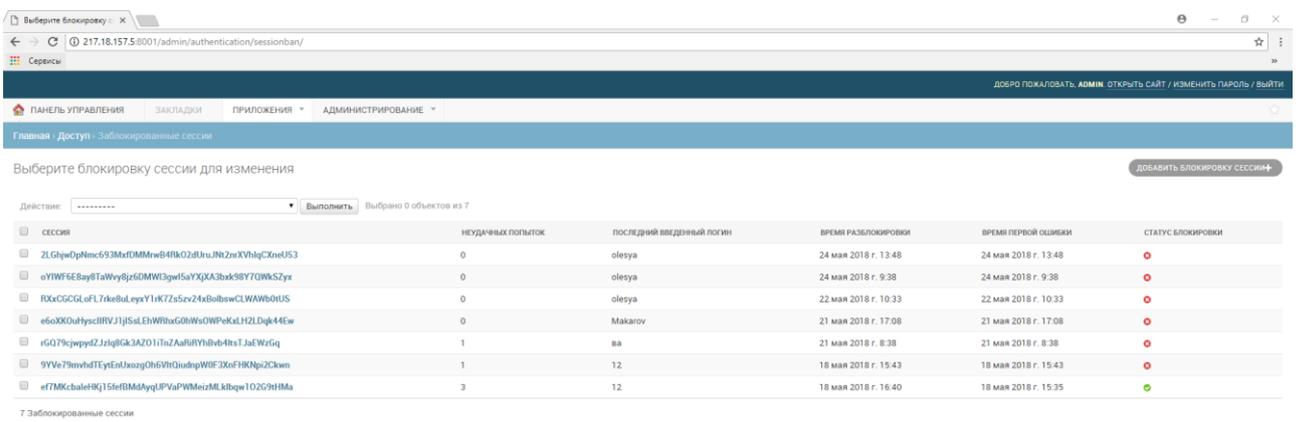


Рисунок Д.26 – Страница просмотра заблокированных сессий из панели администратора