

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

Институт кибернетики
Направление подготовки 09.03.02 «Информационные системы и технологии»
Кафедра информационных систем и технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка веб-приложения для автоматического построения отчетов на основе системы компьютерной вёрстки TeX

УДК 004.738.1:378.267-027.48

Студент

Группа	ФИО	Подпись	Дата
8ИЗБ	Ветров Алексей Александрович		

Руководитель

Должность	ФИО	Учёная степень, звание	Подпись	Дата
Ассистент каф. ИСТ	Хаустов П.А.	-		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Учёная степень, звание	Подпись	Дата
Доцент каф. МЕН	Рахимов Т.Р.	К.Э.Н., ДОЦЕНТ		

По разделу «Социальная ответственность»

Должность	ФИО	Учёная степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Акулов П.А.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Учёная степень, звание	Подпись	Дата
ИСТ	Мальчуков А.Н.	К.Т.Н.		

Томск – 2017 г.

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ПО ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ
ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ 09.03.02 «ИНФОРМАЦИОННЫЕ
СИСТЕМЫ И ТЕХНОЛОГИИ», ИК ТПУ, ПРОФИЛЬ «ГЕОИНФОРМАЦИОННЫЕ
СИСТЕМЫ»**

Код результата	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественно научные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций.
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.

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

Институт кибернетики
Направление подготовки 09.03.02 «Информационные системы и технологии»
Кафедра информационных систем и технологий

УТВЕРЖДАЮ:
Зав. кафедрой ИСТ

_____ Мальчуков А.Н.
(подпись) (дата) (ФИО)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8ИЗБ	Ветров Алексей Александрович

Тема работы:

Разработка веб-приложения для автоматического построения отчётов на основе системы компьютерной вёрстки TeX

Утверждена приказом директора (номер, дата)	№ 664/с 03.02.2017
---	--------------------

Срок сдачи студентом выполненной работы	30.05.2017
---	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Общее описание требуемого функционала от программного продукта.
Перечень подлежащих исследованию, проектированию и разработке вопросов	<ul style="list-style-type: none">– анализ требований потенциальных пользователей к программному продукту;– анализ существующих продуктов для создания документов;– анализ средств для разработки веб-приложений;– создание формата описания документа;– разработка системы преобразования текстового представления документа в графический;– разработка графического и функционального интерфейса редактирования документа;– разработка веб-приложения для редактирования документов;– ресурсоэффективность и ресурсосбережение;– социальная ответственность.

Перечень графического материала	<ul style="list-style-type: none"> – Диаграмма классов модели документа; – Диаграмма классов серверной части; – Схема взаимодействия компонентов веб-приложения; – Графический интерфейс редактора документов;
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Рахимов Тимур Рустамович
Социальная ответственность	Акулов Петр Анатольевич

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	05.09.2016
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ИСТ	Хаустов Павел Александрович	-		05.09.2016

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗБ	Ветров Алексей Александрович		05.09.2016

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

Институт кибернетики

Направление подготовки 09.03.02 «Информационные системы и технологии»

Уровень образования бакалавриат

Кафедра информационных систем и технологий

Период выполнения осенний / весенний 2016/2017 учебного года

Форма представления работы:

Бакалаврская работа

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

Срок сдачи студентом выполненной работы:	30.05.2017
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
24.10.16	Аналитический обзор	15
26.12.16	Проектирование системы	30
06.03.17	Разработка системы вёрстки	10
27.03.17	Разработка серверной части	10
03.05.17	Разработка веб-приложения	15
22.05.17	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
22.05.17	Социальная ответственность	10

Составил преподаватель

Должность	ФИО	Учёная степень, звание	Подпись	Дата
Ассистент каф. ИСТ	Хаустов П.А.	-		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Учёная степень, звание	Подпись	Дата
ИСТ	Мальчуков А.Н.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа 88 с., 11 рис., 18 табл., 25 источников, 1 прил.

Ключевые слова: системы электронного документооборота, компьютерная вёрстка, веб-приложение, TeX, javascript, html, построение отчётов.

Объект исследования: система компьютерной верстки TeX и её использование для автоматизации построения отчётов.

Цель работы – разработка веб-приложения для работы с системой компьютерной вёрстки TeX для автоматизации построения отчётов по заранее заданным шаблонам.

Область применения: компьютерная вёрстка, электронный документооборот, стандартизация электронной документации.

Экономическая значимость работы: сокращается время и стоимость рабочих часов на подготовку отчётов, увеличивается производительность работы, сокращается расход бумаги.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

LaTeX – система компьютерной верстки, предназначенная для описания содержания и структуры документа, а также формирующая на основе выбранного шаблона оформленный и готовый для распечатки документ.

HTML5 (HyperText Markup Language, version 5) – язык для структурирования и представления содержимого всемирной паутины.

CSS3 (Cascading Style Sheets 3) – формальный язык разметки, используемый как средство описания и оформления внешнего вида веб-страниц.

JavaScript – прототипно-ориентированный сценарный язык программирования, встраиваемый в браузер для программного доступа к объектам приложений.

MVC – принцип разработки приложений, согласно которому оно должно иметь модель, представление и контролер, взаимодействующие друг с другом.

WYSIWYG – What You See Is What You Get – принцип работы приложений для редактирования документов, при котором пользователь видит и редактирует документ таким, каким он будет отправлен на печать.

Оглавление

Введение	10
1 Теоретический анализ	12
1.1 Обзор существующих решений	12
1.2 Анализ требований пользователей	13
1.3 Анализ инструментальных средств	15
2 Проектирование	18
2.1 Общая структура системы	18
2.2 Формат описания документов	19
2.3 Система хранения данных	26
2.4 Интерфейс обмена данными между клиентом и сервером	29
3 Реализация	35
3.1 Система вёрстки документов	35
3.2 Серверная часть	39
3.3 Клиентская часть	40
4 Результаты работы	45
5 Финансовый менеджмент, ресурсоэффективность и энергосбережение	48
5.1 Введение	48
5.2 Оценка коммерческого потенциала	49
5.3 Определение возможных альтернатив проведения научных исследований	54
5.4 Планирование научно-исследовательских работ	55
5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	63
6 Социальная ответственность	68
6.1 Профессиональная и социальная безопасность	69
6.2 Экологическая безопасность	73

6.3	Безопасность в чрезвычайных ситуациях	75
6.4	Правовые и организационные вопросы обеспечения безопасности	78
	Заключение	82
	Список публикаций студента	83
	Список использованных источников.....	84
	Приложение А CD диск.....	88

ВВЕДЕНИЕ

Система для автоматического построения отчётов является актуальной для учащихся высших учебных заведений. В ходе обучения и выполнения научной работы приходится создавать большое количество документов, среди которых отчёты о выполнении лабораторных работ и индивидуальных заданий, пояснительные записки к курсовым и дипломным работам. Каждый из таких документов должен быть выполнен с учётом определённых стандартов. Эти стандарты состоят из десятков страниц, определяющих большое количество правил по оформлению документов, такие как шрифт, поля, отступы, интервалы, правила оформления заголовков и так далее. Соблюдения этих правил делает документы единообразными и приятными для просмотра, но на то, чтобы поддерживать их в документе, требуется немало усилий. В итоге, на оформление отчётности может уйти больше времени, чем на саму работу.

Цель работы – автоматизирование процесса по созданию и оформлению отчётности.

Объектом исследования являются системы компьютерной вёрстки документов и их применение в автоматизированных системах. В качестве предмета исследования были выбраны системы, основанные на языке TeX, так как этот язык позволяет задавать разметку документов и правила преобразования текстовой и графической информации в файл, пригодный для печати.

Работа с созданием документов на языке TeX требует от пользователя специфических знаний, касаемых синтаксиса языка и используемых команд. Поэтому основной задачей работы является создание интерфейса для работы с документами по принципу WYSIWIG, что позволит сразу видеть результат редактирования документа и снизит количество необходимых знаний для пользователей

Ключевой особенностью данной системы является поддержка стандар-

тов по оформлению документов. Встроенные в систему шаблоны позволяют абстрагироваться от внешнего вида документов и работать с содержанием и структурой документа. Система шаблонов позволяет обобщить принцип работы с разными типами документов, имеющими схожую структуру, но различный внешний вид. Работая с одним и тем же структурным макетом, с помощью шаблонов можно создать любую организацию данных из этого макета в итоговом документе.

Также система должна поддерживать создание собственных шаблонов, позволяющих подстроить документ под собственные нужды.

Результатом работы должна являться работающая система с шаблонами, предназначенными для использования студентами Томского политехнического университета.

1. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

1.1 Обзор существующих решений

Для создания документов многие используют пакет приложений Microsoft Office, в состав которого входит текстовый процессор Microsoft Word. Это приложение поддерживает подход WYSIWIG, что позволяет работать с ним даже начинающим пользователям. Приложение поддерживает шаблоны, стили оформления, вставку изображений, создание таблиц, графический ввод формул и так далее.

Полноценное использование WYSIWIG не позволяет разделить содержание и внешний вид документа. В любой момент пользователь может случайно изменить стиль определённого элемента, нарушив правила оформления документа. Microsoft Word может предупреждать только о простейших типографических ошибках, таких как орфографические и пунктуационные. Эти правила встроены в Microsoft Word, и пользователь не имеет возможность задавать собственные правила, определяемые стандартами.

Приложения Microsoft Office не имеют поддержки версионности документов. Приложение хранит только историю изменений с момента открытия документа до момента закрытия. После закрытия, документ остаётся в том виде, в котором он был сохранён, отсутствует возможность возврата предыдущих версий документа без использования стороннего программного обеспечения.

Также недостатком является высокая стоимость продуктов, входящих в состав Microsoft Office. А свободные для использования аналоги, такие как LibreOffice не имеют поддержки редактирования документов через веб-браузер.

Другой известной системой для работы с документами является комплекс веб-приложений Google Docs. Эти приложения работают полностью че-

рез веб-браузер, используют принцип WYSIWIG и имеют схожие возможности по оформлению документов, что и продукты Microsoft Office. Ключевыми особенностями Google Docs является хранение документов на сервере Google, поддержка одновременного редактирования документов несколькими пользователями и версионность. Недостатком такого web-приложения является присутствие в нём лишь простейшего редактора формул, поддерживающего только буквы греческого алфавита, математические символы, знаки отношений, математические операции и стрелки. В формулу нельзя вставить более сложные математические объекты, такие как системы уравнений или матрицы. Это делает нецелесообразным использование Google Docs для оформления отчётов о выполнении учебных и научных работ.

1.2 Анализ требований пользователей

Так как у разрабатываемого приложения существуют конкуренты, которые занимают большую долю на рынке текстовых процессоров, необходимо создать конкурентоспособные преимущества. Для этого необходимо проанализировать требования тех пользователей, на которых будет ориентирован продукт.

Потенциальными пользователями приложения являются студенты высших учебных заведений, которые учатся на технических специальностях. Основным вариантом использования приложения является написание отчётов по лабораторным работам, выполняемым в ходе обучения.

Так как руководитель выполняемой работы является преподавателем технической дисциплины «Введение в информационные технологии», он может определить минимальный функционал, требуемый для написания отчётов к лабораторным работам по этой дисциплине.

Основными элементами отчётов являются:

- титульный лист;

- заголовки;
- абзацы;
- формулы;
- таблицы;
- рисунки.

Таким образом, приложения обязано давать возможность создавать и редактировать элементы таких типов.

Для повышения производительности труда пользователей необходимо определить, как именно составляются отчёты, какие этапы требуют больше всего затрат усилий и времени, и что можно упростить или автоматизировать.

Основными этапами выполнения лабораторных работ являются [1]:

- а) изучение методических указаний к лабораторной работе;
- б) выполнение лабораторной работы с документированием хода работы;
- в) подготовка документа для отчёта, оформление титульного листа;
- г) заполнение цели работы и постановки задания;
- д) заполнение хода работы в отчёте;
- е) заполнение результата работы и выводов.

Для всех студентов, выполняющих одну и ту же лабораторную работу, пункты в и г будут одни и те же, а пункты б и д можно выполнять одновременно. Если преподаватель или студент создаст общий шаблон для лабораторной работы, то другим студентам не нужно будет делать то же самое повторно, тем самым увеличивается продуктивность работы. Также шаблон даст возможность задать единообразное оформление отчётов для всех студентов в соответствии с университетскими стандартами.

Таким образом, приложение должно давать возможность задавать уникальное содержимое отчётов каждому студенту, выполняющему лабораторную работу, на основе готовых шаблонов, без необходимости повторно создавать общие для всех работ элементы. Способ задания содержимого не должен требовать от студента оформления его внешнего вида. Форматирование текста долж-

но быть отдельно от содержимого, содержаться в шаблоне и быть стандартным для всех документов.

Последним шагом был выполнен сбор дополнительных требований: того, что пользователи хотели бы видеть в приложении. Было собрано много различных предложений, из которых были выбраны наиболее востребованные и реализуемые в разумные сроки, а именно:

- история изменений;
- автоматическое сохранение документов;
- общий доступ к документам.

1.3 Анализ инструментальных средств

Приложение может иметь собственную систему вёрстки, а может использовать уже существующую. Написание собственной системы требует знания типографики, а также большое количество времени и усилий. Но в этом нет необходимости, так как уже существуют готовые системы, основанные на языке TeX.

Язык TeX был изобретён Дональдом Кнудом для более качественной вёрстки своих книг. С тех пор, TeX приобрёл большую популярность в академической среде за счёт открытости исходного кода, кроссплатформенности и расширяемого функционала.

Документы на языке TeX представляют собой код, который исполняется системой вёрстки для получения документа, пригодная для печати. Такой подход отличается от принципа WYSIWIG, так как пользователь не влияет на внешний вид текста непосредственно. Вёрстка на основе кода позволяет использовать TeX в составе других систем, преобразующих документы из собственного формата в TeX.

Язык TeX постоянно развивается. В настоящее время существуют следующие расширения TeX:

- LaTeX – набор макросов и пакетов, расширяющих синтаксические и функциональные возможности TeX [2];
- XeTeX – система вёрстки, поддерживающая OpenType шрифты и кодировку Unicode. Имеет те же возможности, что и LaTeX, а также совместим с большинством его пакетов [3].
- ConTeXT – схожая с LaTeX система вёрстки, имеющая свой синтаксис и набор пакетов.

Так как в стандартах зачастую указаны OpenType шрифты, а также в современных веб-приложениях используется кодировка Unicode, наиболее подходящим расширением является XeTeX.

Система вёрстки XeTeX может быть запущена на многих популярных операционных системах. Это позволяет сделать приложение также кроссплатформенным, при условии использования кроссплатформенного языка программирования. Таким языком является C#: приложение на этом языке может быть запущено на операционных системах Windows, Linux и Mac.

Другим вариантом языка для кроссплатформенного приложения являются JavaScript, а именно диалекта ECMAScript 5. Такое приложение может быть запущено во многих современных браузерах [4].

Но система вёрстки XeTeX не поддерживает JavaScript, а также требует большое количество памяти на жёстком диске пользователей. Поэтому она может быть вынесена на отдельный сервер на языке C#, который будет использоваться веб-приложением на языке JavaScript.

Сервер может быть написан с помощью стандартных функций языка C#, а может быть написан с использованием библиотеки ServiceStack. Она поддерживается любой операционной системой и позволяют создавать сервисы для обработки входящих HTTP-запросов. Эта библиотека является одной из самых быстрых для C# [5].

Веб-приложение на JavaScript также может быть написано с использованием библиотек. Наиболее популярными библиотеками на 2017 год являются

ся [6]:

- AngularJS;
- Angular 2;
- ReactJS;
- Vue.js;
- Ember.js;
- Meteor.js.

Для реализации веб-приложения был выбрана платформа Angular 2, так как она поддерживает парадигму Модель-Представление-Контроллер, позволяющую разделить логику приложения от модели данных и графического интерфейса. Такой подход повышает независимость отдельных участков кода от других, что позволяет тестировать их отдельно от всей системы.

Также Angular 2 поддерживает строго типизированный язык TypeScript, являющийся диалектом JavaScript. Строгая типизация позволяет исключить ошибки несоответствия типов на этапе написания кода, а также повышает читаемость кода.

2. ПРОЕКТИРОВАНИЕ

2.1 Общая структура системы

Система должна состоять из двух частей: клиентской части, работающей в браузере пользователя, и серверной части, работающей на отдельном сервере. Общая структура системы представлена на схеме 1.

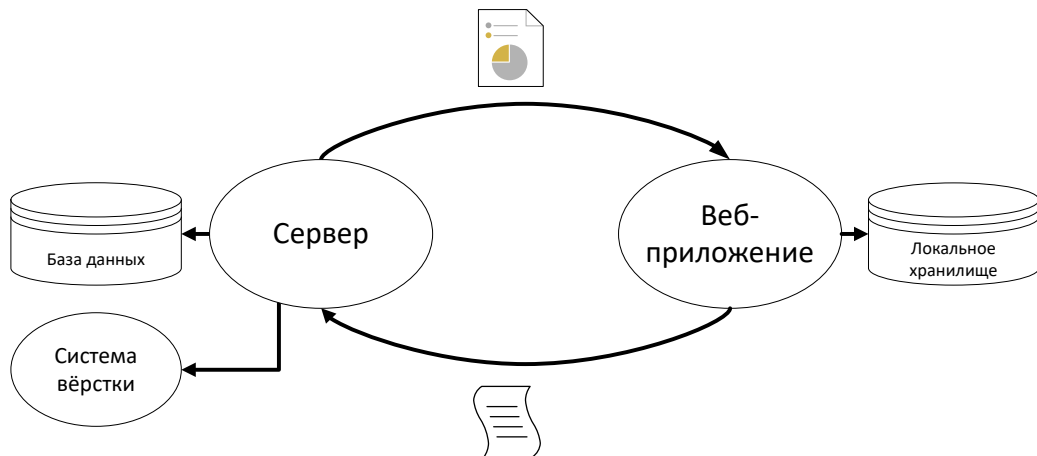


Рисунок 1 – Общая структура системы

Основное взаимодействие пользователя с системой происходит через клиентскую часть. В ней должен быть реализован графический интерфейс редактора документов, позволяющий создавать новые и изменять существующие документы.

Клиентская часть также должна предоставлять личный кабинет каждому пользователю для управления собственными документами, которые были сохранены на сервере.

Пользовательские документы должны также храниться в локальном хранилище браузера для того, чтобы пользователь мог работать с документом даже при временном отсутствии соединения с сервером.

Серверная часть должна преобразовывать документы, поступающие из клиентской части, из текстового вида в графический с помощью системы вёрстки. Также сервер должен иметь базу данных для хранения информации о пользователях и пользовательские документы.

2.2 Формат описания документов

Для описания документов был выбран формат JavaScript Object Notation – JSON. Этот формат по умолчанию поддерживается интерпретатором языка JavaScript, встроенного в веб-браузер, и является более компактным по сравнению с форматом XML [7].

Объекты в JSON содержат в себе множество пар «ключ-значение», которые описывают свойства объекта. Значением может являться строка, число, массив или другой объект. Объекты не имеют фиксированного типа и могут иметь произвольную структуру. Но это неудобно при обработке таких документов типизированными языками программирования C# и TypeScript.

Для решения этой проблемы необходимо создать классы для каждого объекта, к которым будут преобразовываться документы во время операции десериализации. Эта операция сопоставляет каждое поле объекта класса паре «ключ-значение» объекта в формате JSON.

В итоге текстовое представление объекта преобразуется в объект фиксированного типа, который содержит заранее известную структуру. Эта структура описана с помощью диаграммы классов, представленной на рисунке 2.

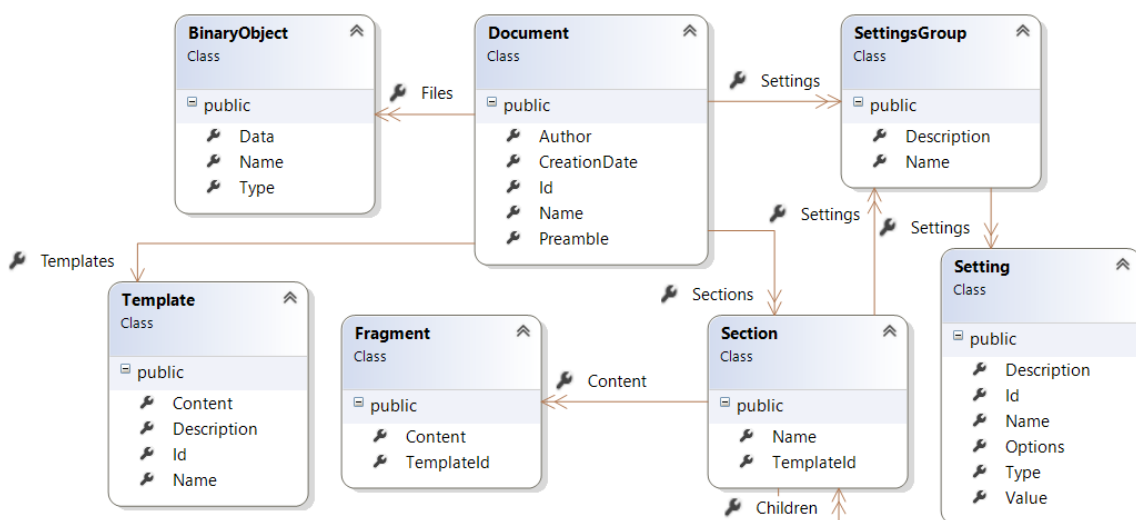


Рисунок 2 – Структура документа

Весь документ описывается классом Document. Объект этого класса со-

держит в себе:

- Название документа (Name): строка, которая содержит название документа.
- Уникальный идентификатор (Id): строка, которая является уникальной среди всех документов на сервере и служит для поиска документа в базе данных.
- Дата создания (CreationDate): строка, содержащая дату создания документа в формате ISO 8601 [8].
- Автор (Author): строка, содержащая идентификатор автора документа в базе данных сервера.
- Преамбула (Preamble): строка, в которой описывается служебная информация для системы вёрстки TeX. Она описывает расширения языка, используемые в документе, размеры страницы, поля, шрифты, стили, макросы и другие команды, которые будут записаны в самом начале документа на языке TeX.
- Параметры (Settings): массив объектов типа SettingsGroup, который хранит параметры, общие для всего документа (см. подраздел 2.2.5).
- Двоичные файлы (Files): массив объектов типа BinaryObject, которые содержат двоичные файлы, поставляемые вместе с документом, например, рисунки (см. подраздел 2.2.6).
- Разделы (Sections): массив объектов типа Section, которые задают иерархическую структуру документа (см. подраздел 2.2.1).
- Шаблоны (Templates): массив объектов типа Template, которые содержат описание фрагментов документа на языке TeX (см. подраздел 2.2.3).

2.2.1 Структура содержимого документа

Содержимое документа представлено в виде иерархического дерева. Корнем дерева является сам документ, а внутри него может находиться произвольное количество разделов.

Каждый раздел может иметь подразделы, они в свою очередь также могут иметь подразделы и так далее. В конце концов, в листьях дерева будут непосредственно фрагменты документа (см. подраздел 2.2.2).

Разделы задаются классом `Section`. Объекты этого класса включают в себя следующие поля:

- Название (`Name`): строка, содержащая название раздела.
- Шаблон (`TemplateId`): строка, содержащая идентификатор шаблона, по которому будет строиться раздел (см. подраздел 2.2.3).
- Содержимое (`Content`): массив объектов типа `Fragment`, описывающий фрагменты раздела (см. подраздел 2.2.2).
- Подразделы (`Children`): массив объектов типа `Section`, содержащий подразделы этого раздела. Подразделы размещаются непосредственно после содержимого.
- Параметры (`Settings`): массив объектов типа `SettingsGroup`, который хранит параметры текущего раздела (см. подраздел 2.2.5).

2.2.2 Фрагменты документа

Фрагменты документа описывают смысловое содержимое документа. Они задаются объектом, который может иметь произвольные поля и вставляется в документ по правилам, заданным с помощью шаблона или системы вёрстки. Таким объектом может быть текст, рисунок, таблица, листинг кода, формула и т.д.

Фрагменты документа описываются объектами класса `Fragment`. Они содержат в себе:

- Шаблон (`TemplateId`): строка, задающая шаблон фрагмента, согласно которому он будет отформатирован.
- Содержимое (`Content`): строка, задающая содержимое фрагмента в виде объекта в формате JSON.

2.2.3 Описание шаблонов

Шаблон описывает то, как части документа будут интерпретированы системой вёрстки и преобразованы в графический формат. Так как для вёрстки документов используется LaTeX, то и шаблоны должны быть написаны на языке TeX.

Для того, чтобы в шаблон можно было включать элементы документа, необходимо расширить возможности языка TeX для вставки в него макросов, которые будут заменены содержимым документа.

Все макросы включаются в шаблон в формате `%%macros_id%%`. На месте `macros_id` указывается название макроса, которое может состоять из латинских букв, цифр, знаков подчёркивания, дефиса двоеточия. Некоторые макросы создаются автоматически на основе содержимого частей документа, а часть задаётся в параметрах документа или раздела (см. подраздел 2.2.5).

Автоматические макросы бывают двух типов:

- Поле (`%%field:...%%`): вместо этого макроса в документ будет вставлено содержимое заданного поля из JSON объекта, описывающего содержимого фрагмента. Например, если объект задаёт текст и представлен как `{"text": "Привет, мир!"}`, то макрос `%%field:text%%` будет преобразован в `Привет, мир!`.

- Обработанное содержимое (`%%typeset:...%%`): вместо этого макроса в документ будет вставлен результат работы системы вёрстки содержимого фрагмента. В зависимости от макроса, система может по разному обрабатывать фрагмент. Все способы обработки перечислены в разделе 3.1.4.

Шаблон задаётся объектом класса `Template`, который содержит следующие поля:

- Идентификатор (`Id`): строка, содержащая идентификатор шаблона, уникальный для всего документа.

- Название (Name): строка, содержащая название шаблона.
- Описание (Description): строка, содержащая описание шаблона.
- Тип редактора (EditorType): строка, определяющая редактор, который необходимо использовать для изменения фрагментов, построенных на этом шаблоне (см. подраздел 3.3.2).
- Содержимое (Content): строка, содержащая сам шаблон на языке TeX с использованием макросов.

2.2.4 Формат описания таблиц

Таблицы представляют собой двумерный массив ячеек, внутри которых может быть некоторое содержимое. В массиве можно выделить ряды (первое измерение массива), и столбцы (второе измерение массива). Таблицы задаются объектами класса `Tabular`, записанными в формате JSON. Диаграмма классов таблиц представлена на рисунке 3.

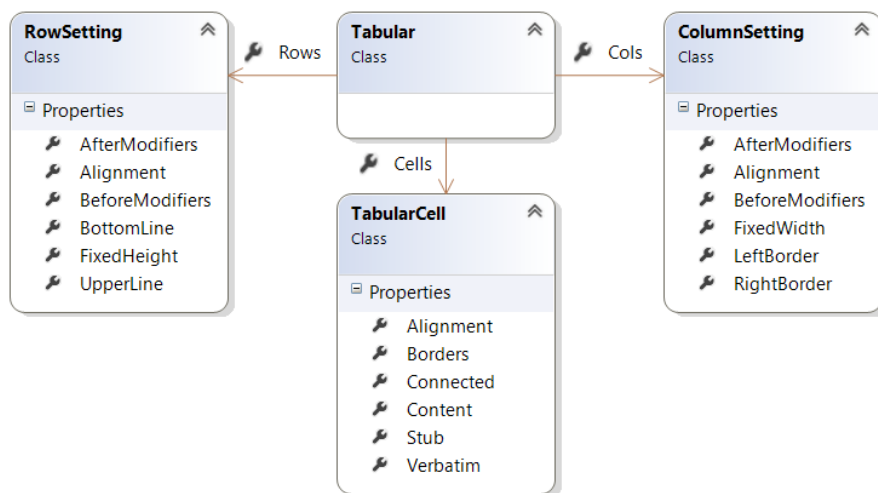


Рисунок 3 – Диаграмма класса `Tabular`

Ряды задаются объектами класса `RowSettings`. Они могут иметь либо фиксированную высоту (`FixedHeight`), либо высоту, равную самой высокой ячейке ряда (при отсутствии `FixedHeight`). Внутри рядов может быть три вида выравнивания (`Alignment`): сверху (`t`), снизу (`b`) и по середине (`m`). Также ряд может иметь описание разделяющей линии сверху (`UpperLine`) и снизу

(BottomLine). Если описание линии отсутствует, то должно быть использовано описание соответствующей линии из ряда сверху или снизу.

Столбцы задаются объектами класса ColumnSettings. Они могут иметь либо фиксированную широту (FixedWidth), либо широту, равную самой широкой ячейке столбца (при отсутствии FixedWidth). Внутри столбца может быть четыре вида выравнивания (Alignment): слева (l), справа (r), по середине (c), или по ширине (p). Также ряд может иметь описание разделяющей линии справа (LeftBorder) и слева (RightBorder). Если описание линии отсутствует, то должно быть использовано описание соответствующей линии из столбца справа или слева.

Также ко всем ячейкам столбца или строки могут быть применены модификаторы, которые добавляются либо до содержимого ячейки (BeforeModifiers), либо после (AfterModifiers).

Ячейки таблиц могут иметь либо содержимое (Content), разрешённое языком TeX к использованию в окружении «tabular», либо произвольный текст (при наличии поля «Verbatim» равного «true»). Ячейки могут переопределять выравнивание в строке (VerticalAlignment), либо в столбце (HorizontalAlignment). Ячейки могут переопределять границы ячейки с помощью массива логических значений Borders, в котором указывается наличие границы в следующем порядке: сверху, слева, снизу и справа.

Ячейка может быть связана с соседними ячейками с помощью массива Connected, в котором указывается наличие связи в следующем порядке: с ячейкой сверху, слева, снизу и справа. Если ячейки связаны, то между ними отсутствует границы, а текст будет записан во всей области связности. Область связности обязана быть прямоугольником. Также внутри области связности должна быть только одна ячейка, содержащая текст, а остальные должны иметь параметр «Stub» равный «true» и не иметь содержимого.

2.2.5 Параметры документа

Помимо текстового содержимого, в документе могут содержаться дополнительные параметры, которые хранят информацию для системы вёрстки и могут быть изменены пользователем документа.

Эти параметры могут быть использованы в виде макросов и включены в шаблоны документа. Каждый параметр создаёт макрос вида `%%param: . . .%%`, где вместо многоточия указывается идентификатор параметра.

Параметры могут быть как общими для всего документа (глобальными), так и принадлежащими определённому разделу (локальными). Локальный параметр может перезаписать глобальный или другой, у которого совпадает идентификатор и который задан выше в иерархическом дереве.

Параметры объединяются в группы для удобства при их просмотре и редактировании. Группы параметров задаются объектами класса `SettingsGroup`, который содержит название группы (`Name`) и текстовое описание (`Description`), а также массив настроек `Settings`, содержащий сами настройки в виде объектов класса `Setting`.

Класс `Setting` содержит следующие поля:

- Идентификатор (`Id`): строка, содержащая идентификатор параметра, уникальный в пределах раздела или глобальных параметров.
- Название (`Name`): строка, содержащая название настройки.
- Описание (`Description`): строка, содержащая текстовое описание настройки.
- Тип (`Type`): строка, содержащая тип настройки.
- Варианты значений (`Options`): массив строк, в которых указаны возможные варианты значения параметра.
- Значение (`Value`): строка, содержащая значение параметра. Значением может быть либо один из вариантов из массива `Options`, либо произвольная

строка, если массив Options пустой.

2.2.6 Класс BinaryObject

Вместе с документом системе вёрстки могут понадобиться дополнительные файлы, например, растровые изображения, которые необходимо вставить в документ. Любой двоичный файл можно записать в текстовом виде с помощью кодировки Base64 с указанием MIME типа [9].

Двоичные файлы описываются с помощью класса BinaryObject, содержащего следующие поля:

- Имя файла (Name): строка, содержащая имя файла.
- Тип файла (Type): строка, содержащая тип файла в формате MIME (например, image/jpeg).
- Данные (Data): строка, содержащая содержимое файла, закодированное в форме Base64.

2.3 Система хранения данных

В серверной части необходимо хранить пользователей, их документы, историю изменений документов, а также образцы, общие для всех пользователей. Для этого используется база данных, построенная согласно модели, изображённой на рисунке 4.

Пользователи хранятся в таблице Users. Эта таблица имеет следующие поля:

- Идентификатор (Id: Int32): уникальный идентификатор пользователя.
- Почтовый ящик (Email: String): электронная почта, используемая для связи с пользователем и для входа в систему.
- Имя (Name: String): имя и фамилия пользователя.
- Пароль (Password: String): SHA-1 хеш пароля пользователя.

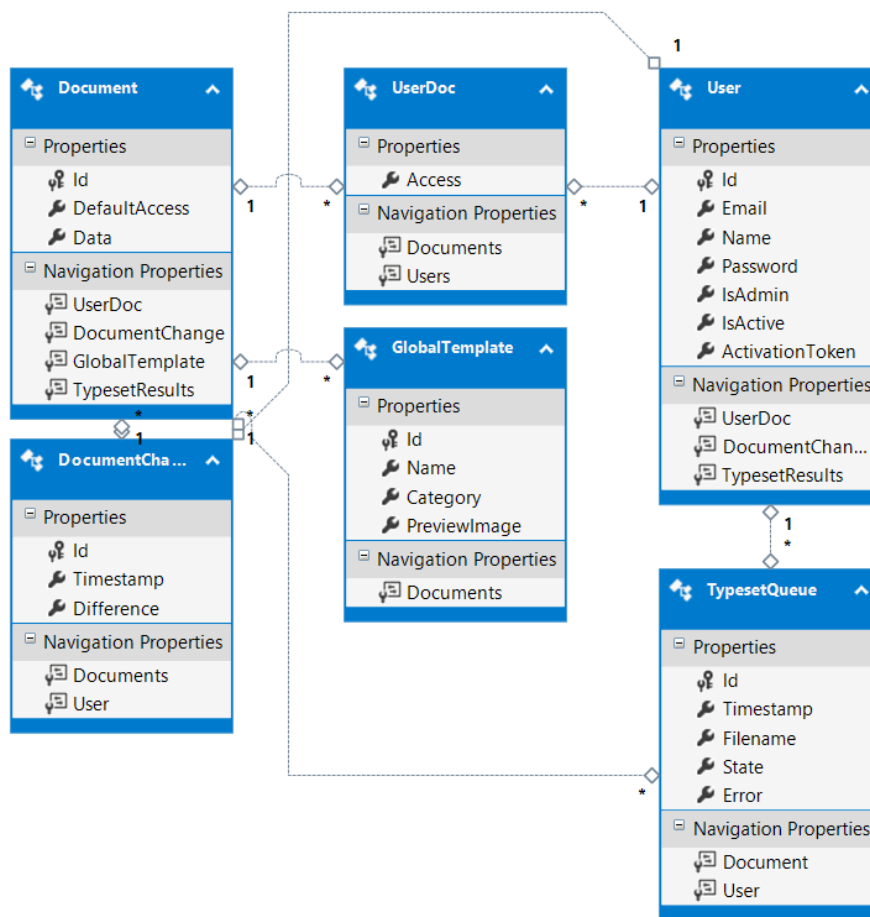


Рисунок 4 – Модель базы данных

- Права администратора (IsAdmin:Boolean): равен true, если пользователь имеет права администратора.
- Статус активации (IsActive:Boolean): равен true, если пользователь подтвердил свою почту и может войти в учётную запись.
- Ключ активации (ActivationToken:String): случайная строка, необходимая для подтверждения почты пользователя.

Документы хранятся в таблице Documents. Эта таблица имеет следующие поля:

- Идентификатор (Id:Int32): уникальный идентификатор документа.
- Уровень доступа по умолчанию (DefaultAccess:Int16): уровень доступа для пользователей, которые не связаны с этим документом (включая неавторизованных).
- Данные (Data:String): содержимое документа в формате JSON (см. раз-

дел 2.2).

Каждый пользователь может иметь несколько документов, а также каждый документ может быть отредактирован несколькими пользователями. Для реализации связи «много ко многим» была создана таблица UserDoc. В ней также хранится уровень доступа пользователя к документу:

- доступ к просмотру;
- доступ к редактированию;
- доступ к управлению (удаления, добавления пользователей).

Некоторые документы могут стать образцами для новых документов и быть размещены на главной странице приложения. Для этого была создана таблица GlobalTemplate, в которой хранится название образца (Name:String), группа(Category:String), к которой он относится, изображение для каталога образцов в формате png (PreviewImage:Binary) и ссылка на документ-образец.

Также документы могут иметь историю изменений. Каждое изменение записывается в таблицу DocumentChanges, в которой указывается время изменения (Timestamp) в формате Unix Time, пользователь, совершивший изменение, и разницу между файлами (Difference).

Если пользователь запрашивает документ, пригодный для печати, то его документ отправляется в очередь вёрстки. Она хранится в таблице TypesetQueue, которая содержит следующие поля:

- Идентификатор (Id:Int32): уникальный идентификатор в очереди.
- Время отправки (Timestamp:Int32): время отправки документа на печать в формате Unix time.
- Рабочая область (Workspace:String): название папки, в которой происходит вёрстка.
- Состояние (State:String): один из вариантов состояния документа в очереди.
- Ошибки (Error:String): ошибки, возникшие при выполнении вёрстки.

Состояние документа в очереди может быть следующим:

- ожидание («waiting»);
- обработка («progress»);
- ошибка («error»);
- успех («success»).

2.4 Интерфейс обмена данными между клиентом и сервером

Сервер должен предоставлять API – программный интерфейс приложения, для взаимодействия с клиентской частью. Интерфейс реализован с помощью HTTP запросов, которые перечислены в таблице 1. Параметры, которые необходимо передать с каждым запросом перечислены в таблице 2, а ответ сервера на запросы описан в таблице 3.

Таблица 1 – HTTP запросы

URL	Метод	Описание
/auth	POST	Аутентификация пользователя в системе.
/register	POST	Регистрации нового пользователя в системе.
/activate	POST	Активация пользователя.
/templates	GET	Получить список шаблонов.
/newdoc	POST	Создать новый документ.
/fetchdoc	GET	Получить существующий документ.
/pushdoc	POST	Обновить существующий документ.
/removedoc	POST	Удалить существующий документ.
/setaccess	POST	Обновить права доступа пользователя к документу.
/setglobal	POST	Добавить документ в шаблоны для всех пользователей. Требуется права администратора.

Продолжение таблицы 1

URL	Метод	Описание
/dochistory	GET	Список изменений документа.
/typeset	POST	Отправить документ на вёрстку.
/typesetqueue	GET	Получить информацию о выполненных над документом операций вёрстки.

Таблица 2 – Передаваемые данные HTTP запросов

URL	Метод	Передаваемые параметры	Значение
/auth	POST	email	Почтовый ящик
		password	Пароль
/register	POST	email	Почтовый ящик регистрируемого пользователя
		name	Псевдоним пользователя
		password	Пароль
/activate	POST	id	Идентификатор пользователя
		token	Токен активации
/templates	GET	category	Категория шаблонов
/newdoc	POST	id	Идентификатор шаблона, из которого необходимо создать документ. Необязательный параметр.

Продолжение таблицы 2

URL	Метод	Передаваемые параметры	Значение
/fetchdoc	POST	id	Идентификатор документа
		timestamp	Время изменения, в которое необходимо получить документ. Необязательный параметр.
/pushdoc	POST	id	Идентификатор документа
		data	Документ в формате JSON
/removedoc	POST	id	Идентификатор документа
/setaccess	POST	id	Идентификатор документа
		email	Почта пользователя
		access	Уровень доступа (0 – если необходимо удалить доступ)
/setglobal	POST	id	Идентификатор документа
		name	Название шаблона
		group	Группа шаблона
		preview	Изображения для предпросмотра в формате PNG
/dochistory	GET	id	Идентификатор документа
/typeset	POST	id	Идентификатор документа
/typesetqueue	GET	id	Идентификатор документа

Таблица 3 – Получаемые данные HTTP запросов

URL	Получаемые данные	Значение
/auth	id	Идентификатор аутентифицированного пользователя
	is_admin	Информация о статусе пользователя
	HTTP cookies	В кукисах запроса будет информация для авторизации пользователя при дальнейших запросах
	HTTP статус	200 в случае успешной авторизации, 401 в случае неправильных авторизационных данных.
/register	HTTP статус	200 в случае успешной регистрации, 406 если уже существует пользователь с регистрируемым именем пользователя.
/acivate	HTTP статус	200 в случае успешной активации, 401 в случае неправильных активационных данных.
/templates	HTTP ответ	JSON объект, содержащий список шаблонов.
	HTTP статус	200 в случае успешного получения данных, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.

Продолжение таблицы 3

URL	Получаемые данные	Значение
/newdoc	HTTP статус	200 в случае успешного создания документа, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.
/fetchdoc	HTTP ответ	JSON объект, содержащий документ.
	HTTP статус	200 в случае успешного получения данных, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.
/pushdoc	HTTP статус	200 в случае успешного обновления документа, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.
/removedoc	HTTP статус	200 в случае успешного удаления документа, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.
/setaccess	HTTP статус	200 в случае успешного обновления прав, 400 в случае неправильных данных, 401 в случае неавторизированного доступа.

Продолжение таблицы 3

URL	Получаемые данные	Значение
/setglobal	HTTP статус	200 в случае успешного создания шаблона, 400 в случае неправильных данных, 401 в случае неавторизованного доступа.
/dochistory	HTTP ответ	JSON объект, содержащий список изменения документа.
	HTTP статус	200 в случае успешного получения данных, 400 в случае неправильных данных, 401 в случае неавторизованного доступа.
/typeset	HTTP статус	200 в случае успешной отправки документа на печать, 400 в случае неправильных данных, 401 в случае неавторизованного доступа.
/typesetqueue	HTTP ответ	JSON объект, содержащий список отправленных на вёрстку документов.
	HTTP статус	200 в случае успешного получения данных, 400 в случае неправильных данных, 401 в случае неавторизованного доступа.

3. РЕАЛИЗАЦИЯ

3.1 Система вёрстки документов

Для преобразования документа из текстового формата, описанного в разделе 2.2, в документ, пригодный для печати, необходима система вёрстки. В качестве инструмента для вёрстки используется система XeTeX, которая может верстать документы, написанные на языке TeX.

Процесс работы системы вёрстки можно разделить на следующие этапы:

- создание рабочей области;
- запись в рабочую область файлов, прикреплённых к документу;
- сборка документа на языке TeX и запись его в рабочую область;
- запуск XeTeX в рабочей области;
- обработка сообщений XeTeX, повторный запуск, если необходимо;
- возврат готового документа или ошибок.

3.1.1 Рабочая область

Для работы системы вёрстки при обработке одного документа создаётся отдельная папка для того, чтобы файлы этого документа не пересекались с другими документами и файлами. Эта папка называется рабочей областью. Процесс вёрстки организовывается так, что всё необходимое создаётся только в этой папке, причём доступ к другим папкам отсутствует, что повышает безопасность работы системы.

При создании папки система назначает ей случайное название так, чтобы не пересекаться с уже существующими папками. После окончания работы системы папка удаляется.

3.1.2 Создание файлов

Для работы системы вёрстки, в рабочей области создаётся файл документа с расширением «.tex», и случайным названием, чтобы оно не пересекалось с двоичными файлами, прикрепленными к документу. Этот файл будет использован для записи документа на языке TeX.

Затем в папке создаются остальные файлы, прикрепленные к документу. Этим файлам назначается имя, указанное в информации о файле, затем в него записывается расшифрованное из формата Base64 содержимое файла.

3.1.3 Сборка документа

Документ в формате JSON представляет собой иерархичное дерево, а формате TeX – это один файл. Поэтому система вёрстки должна произвести рекурсивный обход дерева с помощью алгоритма поиска в глубину, которая возвращает результат сборки каждой части в порядке обхода. Также система сборки в ходе обхода поддерживает список параметров раздела для того, чтобы в них отражались как параметры обрабатываемого раздела непосредственно, так и параметры из объектов выше в иерархии (наследованные параметры). Результат сборки документа (корневого узла дерева) записывается в файл TeX.

Сборка фрагмента осуществляется с помощью шаблона, назначенного этому фрагменту. Если шаблон отсутствует, система должна остановить процесс вёрстки и вернуть пользователю синтаксическую ошибку.

Шаблон просматривается в поисках макросов, которые необходимо обработать системе вёрстки. Для поиска и выделения макросов используется конечный автомат (рис. 5).

Он выделяет из макроса префикс, определяющий тип макроса, и его название. В зависимости от префикса, система вёрстки выполняет следующие

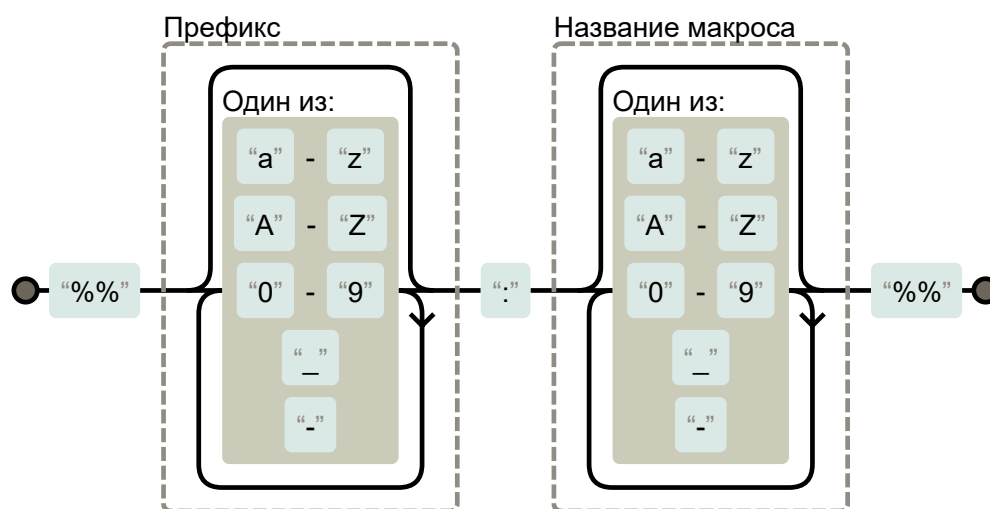


Рисунок 5 – Конечный автомат для поиска макросов

процедуры:

- если префикс равен «param», то макрос ищется в параметрах обрабатываемого раздела, а также в наследованных параметрах;
- если префикс равен «field», то макрос ищется в полях обрабатываемого фрагмента;
- если префикс равен «typeset», то макрос собирается непосредственно в момент вёрстки с помощью оператора, задаваемого названием макроса.

3.1.4 Операторы сборки фрагмента

Система вёрстки должна предоставлять следующие операторы:

- запись фрагментов («typeset:content»);
- запись подразделов («typeset:children»);
- вёрстка таблицы («typeset:tabular»).

Оператор «typeset:content» может быть использован только в шаблоне раздела для того, чтобы включить результат сборки фрагментов раздела в шаблон. Фрагменты склеиваются в том же порядке, что и в массиве фрагментов, и разделяются между собой одним переводом строки.

Оператор «typeset:children» также используется только в шаблоне разде-

ла для того, чтобы включить результат сборки подразделов раздела в шаблон. Подразделы склеиваются в том же порядке, что и в массиве подразделов, и разделяются одним переводом строки.

Оператор «`typeset:tabular`» может быть использован только в шаблоне фрагмента для того, чтобы сверстать таблицу, заданную полем «`tabular`» объекта фрагмента. Формат этого поля описан в разделе 2.2.4.

3.1.5 Запуск системы вёрстки XeTeX

После того, как рабочая область будет подготовлена для вёрстки документа, необходимо запустить сам процесс вёрстки. Для этого выполняется команда:

```
xelatex.exe -interaction=nonstopmode %.tex > texlog.txt
```

которая запускает XeTeX в режиме вёрстки без остановки при ошибках и отправляет информацию о проведённой вёрстке в файл `texlog.txt`. Затем этот файл анализируется с помощью программы `texloganalyser` для поиска ошибок и предупреждений.

Так как алгоритм вёрстки XeTeX является однократным, может потребоваться повторный запуск процесса вёрстки для документов, требующих несколько проходов. Количество операций повторного запуска определяется на основании отладочной информации XeTeX.

Если в отладочном выводе отсутствуют ошибки, система сохраняет полученный документ в папку с готовыми документами и возвращает успешный статус выполнения (код завершения приложения 0). Иначе система возвращает код завершения приложения, отличный от нуля. При любом исходе, система сохраняет отладочную информацию в специальную папку.

3.2 Серверная часть

3.2.1 Обработка запросов

Для обработки запросов пользователей был написан сервис на платформе ServiceStack для языка программирования C#.

Эта платформа занимается обработкой входящих запросов, анализом параметров запроса и переадресацией его соответствующему сервису. Сервис, в свою очередь, анализирует URL запроса и HTTP-метод и вызывает функцию обработки, соответствующую запросу.

Все запросы перечислены в разделе 2.4.

3.2.2 Обработка очереди

Для обработки очереди вёрстки в отдельном потоке выполнения приложения сервера выполняется цикл, который делает следующее:

- Запрашивает в базе данных документы, которые находятся в статусе «waiting». Количество документов, обрабатываемых за одну итерацию цикла зависит от настроек сервера.
- Запускает параллельно несколько экземпляров системы вёрстки и передаёт им выбранные документы.
- Устанавливает статус «process» у обрабатываемых документов.
- Ожидает завершения выполнения процесса вёрстки.
- В зависимости от возвращаемого статуса, устанавливает статус «success» или «error» у обработанных документов.
- Передаёт отладочную информацию в базу данных.

3.2.3 База данных

Взаимодействие с базой данных реализовано с помощью ADO.NET Entity Framework. Entity Framework направлен на то, чтобы давать приложениям возможность чтения и изменения данных, представленных в виде сущностей и связей в концептуальной модели, описанной в разделе 2.3.

Entity Framework использует данные в модели и файлах сопоставления для преобразования запросов объектов к типам сущностей, представленным в концептуальной модели, в запросы, зависящие от источника данных. Результаты запросов преобразуются в объекты, которыми управляют Entity Framework. Запросы осуществляются с помощью LINQ API [10].

3.3 Клиентская часть

3.3.1 Компоненты веб-приложения

Веб-приложение работает непосредственно с моделью документа, предоставляет интерфейс и функционал для работы с ней. Приложение разбито на компоненты, так называемые «чёрные ящики», имеющие интерфейс для взаимодействия друг с другом, но скрывающие внутреннюю реализацию. Таким образом, компоненты могут добавляться, изменяться и тестироваться без необходимости менять всю архитектуру приложения. Общая схема клиентской части представлена на рисунке 6.

Основным модулем для работы с документом является модуль управления документом (DocumentManageModule). Он содержит в себе сервисы, необходимые для работы с документом, а также связывает вместе компоненты редактора.

При открытии ссылки на документ, пользователь перенаправляется

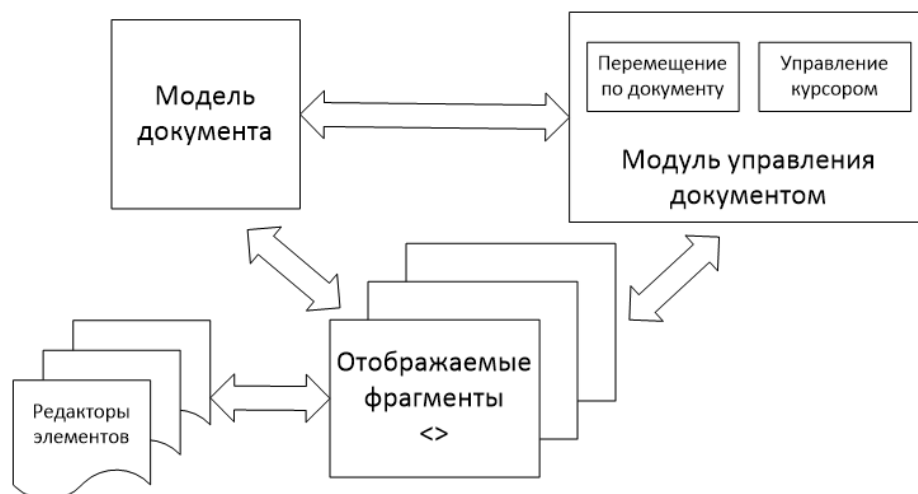


Рисунок 6 – Конфигурации компонентов Angular

на страницу редактирования документа, которая обслуживается компонентом `DocumentEditorComponent`. Интерфейс этой страницы представлен на рисунке 9.

Слева на этой странице находится дерево документа, которое обслуживается компонентом `DocumentTreeComponent`. Справа находятся фрагменты документа, которые находятся в выбранной части документа. Эти фрагменты отрисовываются с помощью соответствующих компонентов.

3.3.2 Редакторы фрагментов

Веб-приложение позволяет создавать и редактировать следующие типы фрагментов:

- параграф (paragraph);
- рисунок (picture);
- формула (equation);
- таблица (table);
- список (list).

Добавление фрагментов осуществляется переносом с помощью мыши соответствующей пиктограммы (рисунок 7) внутрь документа.

Каждый фрагмент имеет свой интерфейс для взаимодействия с пользо-

2. Работа с файлом hosts

Для сопоставления доменных имён помимо DNS-сервера используется файл hosts. В него было добавлено соответствие между именем WS1.host2 и его IP-адресом, затем был очищен кэш распознавателя DNS.

```
C:\Documents and Settings\MS 1>ipconfig /flushdns
Настройка протокола IP для Windows
Успешно сброшен кэш распознавателя DNS.
```

Рисунок 7 – Очистка кэша DNS



Рисунок 7 – Редакторы фрагментов

вателем. Параграф позволяет задавать выравнивание (по ширине, по левому краю, по правому краю), кегль шрифта и начертание (полужирный, курсив) для выделенного фрагмента текста.

Рисунок позволяет загружать изображения в форматах JPEG, PNG и BMP, а также устанавливать подпись к рисунку. Рисунки нумеруются автоматически в зависимости от положения в тексте.

Редактор формул поддерживает ввод формул как с помощью команд LaTeX, так и с помощью экранной клавиатуры. Приложение поддерживает ввод различных математических операторов и знаков и позволяет вводить многоуровневые формулы. Более подробно возможности и реализация редактора формул описана в предыдущей публикации студента [11].

Редактор таблиц позволяет создавать двумерный массив ячеек, внутри которых может быть произвольное содержимое. Пользователь может добавлять в таблицу столбцы, строки, изменять высоту и ширину строк и столбцов, склеивать соседние ячейки, задавать выравнивание внутри ячейки и стиль границ таблицы.

Редактор списков позволяет добавлять и удалять элементы списка, задавать уровень элемента и стиль нумерации.

3.3.3 Вспомогательные сервисы

Для синхронизации модели документа с хранилищем (локальным и удалённым) был создан сервис `DocumentProviderService`. При любом изменении содержимого документа, редактор уведомляет этот сервис. После этого модель документа сохраняется в локальном хранилище. Если пользователь случайно закроет браузер или отключит питание компьютера, при возврате к редактированию документа сервис восстановит версию документа из локального хранилища. Если пользователь прекращает редактирование документа или нажимает кнопку «Сохранить», то сервис отправляет документ на сервер.

Для перемещения курсора между фрагментами был реализован сервис `DocumentNavigationService`, который поддерживает текущее положение курсора и перехватывает нажатие курсорных клавиш для того, чтобы переместить курсор в новое место. Также этот сервис поддерживает начальное и конечное положение курсора при выделении текста, и возвращает указатель на выделенный участок текста или фрагмент.

3.3.4 Параметры и автоматические страницы

У каждого документа есть глобальные параметры, задающие значения макросов, используемых в шаблоне документа (см. раздел 2.2.3). Эти параметры группируются по блокам, каждый блок отвечает за тот или иной аспект документа. Внутри блока находятся настройки, каждая может принимать либо произвольное значение (`Type = "text"`), либо значение из списка заданного в документе (`Type = "option"`). Параметры управляются компонентом `DocumentPreferencesComponent`.

По такому же принципу организуются и автоматические страницы: каждый раздел может иметь ассоциированные параметры, которые отобра-

жаются выше содержимого. А если шаблон раздела не содержит макрос `%%typeset:content%%`, то внутри него не может быть содержимого и он становится автоматической страницей, изображённой на рисунке ??.

3.3.5 Подготовка документа к печати

При переходе в раздел «Сборка документа», документ отправляется на сервер в очередь вёрстки. Вместо редактора включается компонент `DocumentPublishComponent`, который выполняет запрос к серверу на получение очереди вёрстки документа. Этот запрос имеет HTTP-параметр `Keep-Alive`, который позволяет веб-приложению ожидать окончания выполнения сборки документа. После окончания сборки, клиентская часть предоставляет возможность просмотреть или скачать документ.

3.3.6 Пользовательские профили

Для использования веб-приложения необходимо быть зарегистрированным пользователем системы. Для управления пользователями был написан модуль `ProfileModule`, с помощью которого можно зарегистрировать нового пользователя, войти в систему, изменить данные существующего пользователя, просмотреть собственные документы и историю их изменения.

4. РЕЗУЛЬТАТЫ РАБОТЫ

Разработанная система состоит из нескольких частей, взаимодействующих друг с другом для создания и обработки пользовательских документов.

Серверная часть занимается обработкой запросов от пользователей, обработкой очереди вёрстки и взаимодействием с базой данных и написана на языке C# на платформе ServiceStack, что позволяет запускать сервер на любой операционной системе, поддерживающей .NET Framework 4.5.

Для преобразования документа из текстового формата в документ, пригодный для печати, была разработано приложение, которое преобразует документ в формат TeX и использует систему компьютерной вёрстки XeTeX для преобразования документа в формат PDF.

Для хранения данных сервер использует базу данных PostgreSQL, так как она поддерживается многими операционными системами, например, Windows, Linux и MacOS [12].

Для реализации клиентской части был выбран MVC фреймворк Angular, который формирует структуру приложения. Для оформления интерактивных элементов были созданы HTML шаблоны, использующие библиотеку Google Material, которые позволяют оформить компоненты по стандартам оформления от Google. На рисунках 8-10 изображены элементы интерфейса веб-приложения.

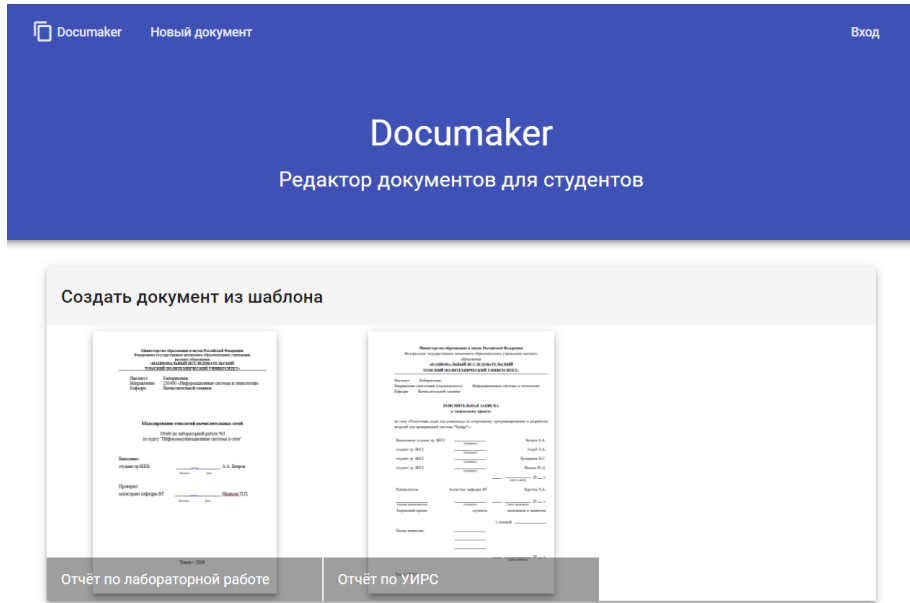


Рисунок 8 – Главная страница

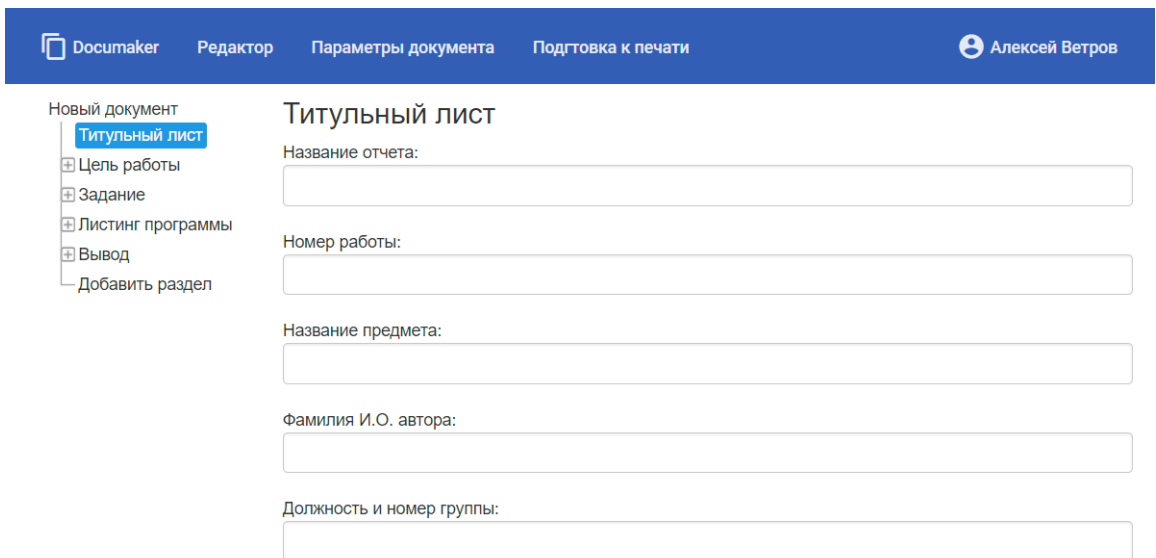


Рисунок 9 – Редактор автоматической страницы

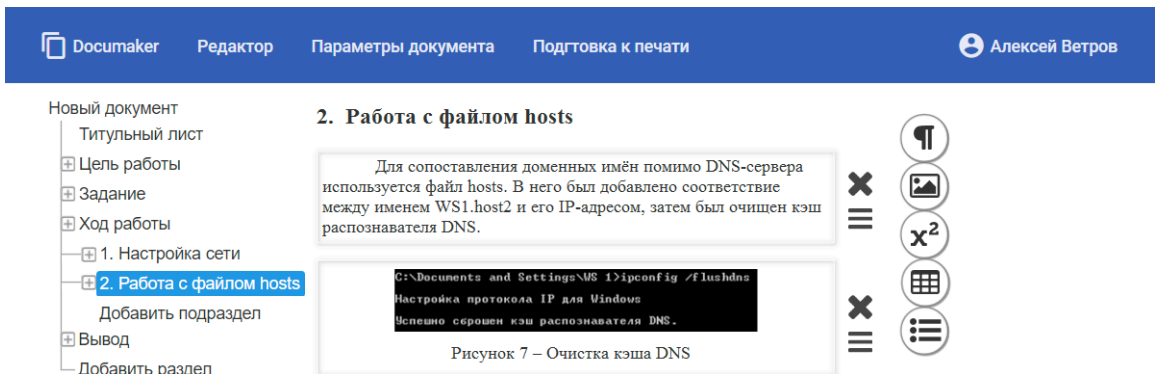


Рисунок 10 – Редактор документа

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И
ЭНЕРГОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИЗБ	Ветров Алексей Александрович

Институт	Кибернетики	Кафедра	Вычислительной техники
Уровень образования	Бакалавр	Направление / специальность	09.03.02 «Информационные системы и технологии»

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

<i>1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>
<i>2. Нормы и нормативы расходования ресурсов</i>	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<i>1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения.</i>	<i>5. Проведение SWOT-анализа.</i>
<i>2. Планирование и научно-исследовательских работ.</i>	<i>6. Определение структуры работ.</i>
<i>3. Определение потенциальных потребителей.</i>	<i>7. Разработка календарного плана-графика и бюджета работ.</i>
<i>4. Анализ конкурентных технических решений.</i>	<i>8. Анализ ресурсной, финансовой и экономической эффективности НИ</i>

Перечень графического материала:

- 1. Оценка конкурентоспособности технических решений*
- 2. Матрица SWOT*
- 3. Альтернативы проведения НИ*
- 4. График проведения и бюджет НИ*
- 5. Оценка ресурсной, финансовой и экономической эффективности НИ*

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. МЕН	Рахимов Тимур Рустамович	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗБ	Ветров Алексей Александрович		

5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И ЭНЕРГОСБЕРЕЖЕНИЕ

5.1 Введение

Несмотря на то, что сейчас активно развивается направление электронного документооборота, во многих организациях всё ещё используется бумажная отчётность. Вся эта документация требует соблюдения многостраничных стандартов, и возникает потребность в том, чтобы упростить и автоматизировать эту работу.

На данный момент существует множество различных текстовых редакторов с большим количеством возможностей. Они позволяют создавать документы с любым графическим и текстовым наполнением, задавать любое оформление текста и многое другое.

Но принципы, по которым создаётся документ в таких редакторах, не позволяют пользователю абстрагироваться от оформления текста, и заниматься только содержанием. Помимо смыслового наполнения, требуется немало времени и использованной бумаги для того, чтобы добиться типографского качества вёрстки документа.

Цель работы с разделом «Финансовый менеджмент и ресурсоэффективность» состоит в том, чтобы разработать приложение, наиболее эффективно использующее ресурсы пользователей системы, при этом затратив наименьшее количество средств на разработку.

Для этого необходимо оценить конкурентные решения с точки зрения эффективности, найти те характеристики, которые можно сделать лучше для повышения конкурентноспособности разрабатываемого приложения.

Также необходимо оценить требования пользователей к продукту, чтобы повысить производительность их труда.

После этого необходимо определить наиболее оптимальный способ выполнения работы, а именно, выработать альтернативные варианты и оценить их с точки зрения эффективности.

Таким образом, при оптимизации проекта достигается его наибольшая финансовая и ресурсоэффективность, что является приоритетной задачей разработки.

5.2 Оценка коммерческого потенциала

5.2.1 Потенциальные потребители результатов исследования

Целевым рынком разрабатываемой системы является рынок систем для работы с текстами. На этом рынке представлены следующие группы программ:

- Текстовые редакторы
- Текстовые процессоры
- Издательские системы

Текстовые редакторы – это программы для создания, просмотра и изменения текстовых данных. Текст, с которым работает текстовый редактор, не предназначен для печатных документов, так как он представлен в виде последовательности символов. Текстовые редакторы используются для редактирования текстов программ.

В свою очередь, текстовые процессоры поддерживают форматирование текста и подготовку документа к печати. Также они позволяют вставлять изображения, таблицы и другие графические элементы.

Издательские системы – это специализированное программное обеспечение для создания макетов страниц печатных изданий, таких как книги, газеты, журналы. Позволяют более точно контролировать типографические параметры текста для повышения его качества и читаемости.

Основными пользователями систем для работы с текстами являются

все пользователи ПК, так как текст является неотъемлемой частью информации, обрабатываемой с помощью компьютера. Это подтверждается тем, что простейший базовый редактор текста «Блокнот» входит в состав наиболее используемой на персональных компьютерах операционной системы Microsoft Windows.

Некоторые области деятельности требуют больше возможностей для работы с текстом:

- Документооборот и делопроизводство
- Разработка ПО
- Научная деятельность
- Издательства

На основании областей деятельности и категорий ПО можно построить карту сегментирования рынка (таблица 4).

Таблица 4 – Карта сегментирования рынка обработки текстов

		Категории ПО		
		Текстовые редакторы	Текстовые процессоры	Издательские системы
Области деятельности	Документооборот и делопроизводство			
	Разработка ПО			
	Научная деятельность			
	Издательства			

Microsoft Word		Visual Studio Code		LaTeX		Блокнот	
----------------	--	--------------------	--	-------	--	---------	--

Разрабатываемое приложение будет ориентироваться на пользователей, занимающихся написанием научных статей и, возможно, планирующих опубликовать результаты своей деятельности. На карте сегментирования можно заметить, что в этой области существует конкуренция, следовательно, разрабаты-

ваемое приложение должно иметь преимущества перед ними.

5.2.2 Анализ конкурентных решений

На данный момент основными конкурентами разрабатываемой системы являются:

- а) Microsoft Word
- б) Google Docs
- в) LaTeX

Для определения конкурентоспособности приложения необходимо проанализировать его достоинства и недостатки по отношению к конкурентам. Были выделены основные критерии оценки, по которым можно выполнить сравнение технических решений, и на основе этих критериев была построена оценочная карта (таблица 5).

По итогам анализа конкурентных решений были выявлены преимущества разрабатываемого приложения, выделяющие его на фоне других. Следовательно, на этих преимуществах стоит сфокусировать внимание при дальнейшем развитии приложения.

Таблица 5 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы				Конкурентоспособность			
		B_{ϕ}	$B_{\kappa 1}$	$B_{\kappa 2}$	$B_{\kappa 3}$	K_{ϕ}	$K_{\kappa 1}$	$K_{\kappa 2}$	$K_{\kappa 3}$
Технические критерии оценки ресурсоэффективности									
1. Повышение производительности труда пользователя	0.1	5	4	4	3	0.5	0.4	0.4	0.3
2. Удобство в эксплуатации	0.07	4	3	5	2	0.28	0.21	0.35	0.14
3. Совместимость с устройствами	0.05	5	3	5	5	0.25	0.15	0.25	0.25
4. Устойчивость к перебоям электропитания	0.07	5	4	5	2	0.35	0.28	0.35	0.14
5. Потребность в ресурсах памяти	0.03	4	3	5	4	0.12	0.09	0.15	0.12
6. Помощь при оформлении документа	0.08	5	4	4	5	0.4	0.32	0.32	0.4
7. Функциональные возможности	0.1	5	4	3	5	0.5	0.4	0.3	0.5
8. Скорость работы	0.08	4	4	5	4	0.32	0.32	0.4	0.32
9. Качество интеллектуального интерфейса	0.08	5	4	5	2	0.4	0.32	0.4	0.16
Экономические критерии оценки эффективности									
1. Конкурентоспособность продукта	0.07	5	5	4	3	0.35	0.35	0.28	0.21
2. Уровень проникновения на рынок	0.04	2	5	4	3	0.08	0.2	0.16	0.12
3. Цена	0.09	5	2	5	5	0.45	0.18	0.45	0.45
4. Послепродажное сопровождение	0.08	4	5	3	2	0.32	0.4	0.24	0.16
5. Законченность продукта	0.06	4	5	4	3	0.24	0.3	0.24	0.18
Итого						4.56	3.92	4.29	3.45

5.2.3 SWOT-анализ

Для исследования внешней и внутренней среды проекта был проведён SWOT-анализ. Были перечислены сильные, слабые стороны проекта, возможности и угрозы внешней среды. В таблице 6 приведены результаты анализа в виде матрицы с рекомендациями к дальнейшему развитию проекта.

Таблица 6 – Матрица SWOT

	<p>Сильные стороны: С1. Встроенные стандарты оформления документов С2. Работа в веб-браузере С3. Большой набор готовых элементов С4. Приспособленность к научной работе С5. Поддержка истории изменений</p>	<p>Слабые стороны: Сл1. Работа только при наличии Интернета Сл2. Сложность задания собственных правил оформления Сл3. Сложность изменения поведения готовых элементов Сл4. Невозможность создания произвольных документов</p>
<p>Возможности: В1. Увеличение популярности веб-приложений В2. Повышение стоимости конкурирующих продуктов В3. Повышение требований к оформлению документов В4. Появление интереса к автоматизации построения отчётов</p>	<p>Для того, чтобы повысить интерес к продукту необходимо проводить обучающие семинары среди студентов, рассказывающие, какие возможности присутствуют в приложении и как их использовать.</p>	<p>Часть возможностей системы можно сделать доступной без интернета за счёт сохранения информации на компьютере пользователя. В приложении можно добавить инструкции «для профессионалов» с видеоуроками создания собственных элементов и правил оформления.</p>
<p>Угрозы: У1: Переход на систему электронного документооборота У2: Появление схожего функционала в конкурирующих продуктах У3: Появление новых стандартов оформления документов У4: Проявление неучтённых видов содержимого документов</p>	<p>Так как формат документов не зависит от внешнего вида, система может стать частью электронного документооборота. Узконаправленность системы может дать более точное соответствие ожиданиям пользователя.</p>	<p>Система должна иметь обратную связь с пользователями для своевременного выявления новых требований и проблем с текущим функционалом. Можно проводить массовые опросы для определения удовлетворённости пользователей работой системы.</p>

5.3 Определение возможных альтернатив проведения научных исследований

Для оценки ресурсоэффективности разрабатываемого приложения необходимо определить и проанализировать возможные варианты создания проекта. Наиболее оптимальным способом поиска альтернатив является синтез вариантов из особенностей строения (морфологии) проекта. Основные характеристики системы и возможные варианты значения этих характеристик были выделены в морфологическую матрицу системы (таблица 7).

Таблица 7 – Морфологическая матрица системы

	1	2	3
А. Тип приложения	Веб-приложение	Настольное приложение	Мобильное приложение
Б. Платформа серверной части	ASP.NET	ServiceStack	NodeJS
В. Платформа клиентской части	Angular	ReactJS	JQuery
Г. Хранение документов	На стороне клиента	На стороне сервера	Дублирование на клиенте и на сервере
Д. Вёрстка документов	На стороне клиента	На стороне сервера	
Е. Система вёрстки	LaTeX	Microsoft Word	Собственное решение
Ж. Тип редактора документов	WYSIWIG	Текст и структура	Комбинированное решение
З. Формат хранения документов	JSON	XML	Собственный формат
И. Формат описания стандартов	Расширение языка разметки	Преамбула TeX	Файл стилей

Путём комбинации различных параметров морфологии проекта были

определены три наиболее оптимальных варианта исполнения:

- А1Б2В1Г3Д2Е1Ж3З1И1
- А2Г1Д1Е2Ж1З2И3
- А1Б1В2Г1Д1Е1Ж2З3И2

5.4 Планирование научно-исследовательских работ

5.4.1 Структура работ в рамках научного исследования

Процесс проведения работ в рамках научного исследования имеет определённую структуру и может быть разбит на этапы, которые включают в себя список работ и их исполнителей. Перечень этапов, работ и распределение исполнителей приведены в таблице 8.

Таблица 8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№	Содержание работ	Должность исполнителя
Разработка ТЗ	1	Составление и утверждение технического задания	Руководитель
Выбор направления исследований	2	Подбор и изучение материалов по теме	Инженер
	3	Изучение существующих продуктов	Инженер
	4	Выбор платформы разработки	Инженер, руководитель
	5	Календарное планирование работ по теме	Инженер, руководитель
Теоретические исследования	6	Анализ требований пользователей к продукту	Инженер, руководитель

Продолжение таблицы 8

Основные этапы	№	Содержание работ	Должность исполнителя
	7	Составление требуемого от системы функционала	Инженер
Проектирование	8	Составление общей структуры системы	Инженер
	9	Создание формата описания документа	Инженер
	10	Разработка системы вёрстки документов	Инженер
	11	Разработка графического интерфейса редактора	Инженер
Реализация ПО	12	Реализация серверной части	Инженер
	13	Реализация клиентской части	Инженер
	14	Интеграционное тестирование	Инженер, руководитель

5.4.2 Определение трудоёмкости выполнения работ

Таблица 9 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях T_{pi}			Длительность работ в календарных днях T_{ki}		
	t_{min} , чел-дни			t_{max} , чел-дни			$t_{ожж}$, чел-дни											
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Составление ТЗ	1	1	1	2	2	2	1.4	1.4	1.4	1	1	1	1.4	1.4	1.4	2	2	2

Продолжение таблицы 9

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях T_{pi}			Длительность работ в календарных днях T_{ki}		
	t_{min} , чел-дни			t_{max} , чел-дни			$t_{ожж}$, чел-дни											
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Изучение материалов	2	2	2	5	5	5	3.2	3.2	3.2	1	1	1	3.2	3.2	3.2	5	5	5
Изучение аналогов	1	1	1	3	3	3	1.8	1.8	1.8	1	1	1	1.8	1.8	1.8	3	3	3
Выбор платформы разработки	2	2	2	3	3	3	2.4	2.4	2.4	2	2	2	1.2	1.2	1.2	2	2	2
Календарное планирование	1	1	1	2	2	2	1.4	1.4	1.4	2	2	2	0.7	0.7	0.7	1	1	1
Анализ требований к продукту	1	1	1	3	3	3	1.8	1.8	1.8	2	2	2	0.9	0.9	0.9	1	1	1
Составление функционала	3	3	3	5	5	5	3.8	3.8	3.8	1	1	1	3.8	3.8	3.8	6	6	6
Составление структуры системы	2	1	2	4	3	5	2.8	1.8	3.2	1	1	1	2.8	1.8	3.2	4	3	5
Создание формата документа	4	2	15	10	7	20	6.4	4.0	17	1	1	1	6.4	4	17	10	6	26
Разработка системы вёрстки	5	10	15	10	20	30	7.0	14	21	1	1	1	7	14	21	11	21	32
Разработка интерфейса	7	5	3	12	10	8	9.0	7.0	5.0	1	1	1	9	7	5	14	11	8
Реализация серверной части	3	0	2	7	0	5	4.6	0.0	3.2	1	1	1	4.6	0	3.2	7	0	5
Реализация клиентской части	10	20	7	20	30	15	14	24	10	1	1	1	14	24	10	21	36	15
Интеграционное тестирование	5	5	10	10	10	15	7.0	7.0	12	2	2	2	3.5	3.5	6	5	5	9



Рисунок 11 – Календарный план-график проведения НИОКР по теме

5.4.3 Основная заработная плата исполнителей

Одной из главных статей бюджета при разработке программного обеспечения является основная заработная плата исполнителей. Объем расходов на заработную плату определяется исходя из оклада, определяемого предприятием разработчика, и количества дней, затраченных на выполнение работ. Расхо-

ды определяются по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p \quad (1)$$

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

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн. (таблица 9);

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

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_{\partial}} \quad (2)$$

где Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года;

F_{∂} – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (табл. 10).

Таблица 10 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней – выходные дни – праздничные дни	118	118
Потери рабочего времени – отпуск – невыходы по болезни	24	24
Действительный годовой фонд рабочего времени	223	223

Расчёт основной заработной платы исполнителей приведён в таблице 11, а также в таблице 12, где описаны затраты на основную заработную плату по каждому этапу выполняемой работы.

Таблица 11 – Расчёт основной заработной платы исполнителей

Исполнители	Z_m , руб	F_o , дни	M , мес	Z_o , руб	T_p , раб. дн.			$Z_{осн}$, руб		
					Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель	20000	223	11.2	1004	7.7	7.7	10.2	15496	15496	20491
Дипломник	40000	223	11.2	2009	58.9	65.9	77.2	59164	66195	77546

Таблица 12 – Расчёт затрат на выплату основной заработной платы

№ эта-па (таб. 8)	Исполнители по категориям			Трудоёмкость, чел.дн.			Заработная плата, приходящаяся на один чел.дн., тыс. руб.			Всего заработная плата по тарифу (окладам), тыс. руб.		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
1	Р	Р	Р	1.4	1.4	1.4	2.01	2.01	2.01	2.8	2.8	2.8
2	Д	Д	Д	3.2	3.2	3.2	1.00	1.00	1.00	3.2	3.2	3.2
3	Д	Д	Д	1.8	1.8	1.8	1.00	1.00	1.00	1.8	1.8	1.8
4	Р,Д	Р,Д	Р,Д	2.4	2.4	2.4	1.51	1.51	1.51	3.6	3.6	3.6
5	Р,Д	Р,Д	Р,Д	1.4	1.4	1.4	1.51	1.51	1.51	2.1	2.1	2.1
6	Р,Д	Р,Д	Р,Д	1.8	1.8	1.8	1.51	1.51	1.51	2.7	2.7	2.7
7	Д	Д	Д	3.8	3.8	3.8	1.00	1.00	1.00	3.8	3.8	3.8
8	Д	Д	Д	2.8	1.8	3.2	1.00	1.00	1.00	2.8	1.8	3.2
9	Д	Д	Д	6.4	4	17	1.00	1.00	1.00	6.4	4.0	17.1
10	Д	Д	Д	7	14	21	1.00	1.00	1.00	7.0	14.1	21.1
11	Д	Д	Д	9	7	5	1.00	1.00	1.00	9.0	7.0	5.0
12	Д	Д	Д	4.6	0	3.2	1.00	1.00	1.00	4.6	0.0	3.2
13	Д	Д	Д	14	24	10.2	1.00	1.00	1.00	14.1	24.1	10.2
14	Р,Д	Р,Д	Р,Д	7	7	12	1.51	1.51	1.51	10.5	10.5	18.1
Итого:										74.6	81.7	98.0

5.4.4 Дополнительная заработная плата исполнителей

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 (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Дополнительная заработная плата вычисляется по следующей формуле:

$$Z_{дон} = k_{дон} * Z_{осн} \quad (3)$$

где $k_{дон}$ — коэффициент дополнительной заработной платы, который принимается равным 0.15.

Дополнительная заработная плата исполнителей перечислена в таблице 13.

Таблица 13 – Дополнительная заработная плата исполнителей

Исполнитель	Исп. 1	Исп. 2	Исп. 3
Руководитель	2320	2320	3074
Исполнитель	8875	9929	11632

5.4.5 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов учитываются обязательные отчисления во внебюджетные фонды от затрат на оплату труда работников.

Величина отчислений вычисляется следующим образом:

$$Z_{внеб} = k_{внеб} \cdot (Z_{осн} + Z_{дон}) \quad (4)$$

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

Для учреждений осуществляющих образовательную и научную деятельность коэффициент отчислений полагается 0.271. Отчисления во внебюджетные фонды представлены в таблице 14.

Таблица 14 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель проекта	15469	15469	20491	2320	2320	3074
Студент-дипломник	59164	66196	77546	8875	9929	11632
Коэффициент отчислений во внебюджетные фонды	0.271					
Итого						
Исполнение 1	23259					
Исполнение 2	25451					
Исполнение 3	30553					

5.4.6 Формирование бюджета затрат научно-исследовательского проекта

По итогам расчётов, выполненных в предыдущих разделах, можно составить полный бюджет затрат на реализацию каждого из вариантов исполнения проекта (таблица 15).

Таблица 15 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.			Примечание
	Исп. 1	Исп. 2	Исп. 3	
1. Затраты по основной заработной плате исполнителей темы	74633	81665	98038	Пункт 5.4.3
2. Затраты по дополнительной заработной плате исполнителей темы	11195	12249	14706	Пункт 5.4.4
3. Отчисления во внебюджетные фонды	23259	25451	30553	Пункт 5.4.5
4. Накладные расходы	17454	19098	22928	16% от суммы ст. 1-3
5. Бюджет затрат НИИ	126541	138463	166225	Сумма ст. 1-4

5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Оценка целесообразности вариантов исполнения проекта выполняется с помощью интегрального показателя эффективности научного исследования, вычисляемой на основе финансовой эффективности и ресурсоэффективности.

Для нахождения интегрального финансового показателя используется вариант исполнения с наибольшей бюджетной стоимостью, а затем на него делаются все остальные варианты. Таким образом определяется, во сколько раз эффективнее каждый вариант, чем самый дорогой. Расчет приведен в таблице 16.

Таблица 16 – Расчет интегрального финансового показателя

Бюджет затрат	126541	138463	166225
Интегральный финансовый показатель	0.76	0.83	1.00

Для нахождения интегрального показателя ресурсоэффективности используется средневзвешенная оценка вариантов по пятибальной шкале. Эта оценка строится на основе важных характеристик ресурсоэффективности, каждой из которых присвоен определённый вес в зависимости от значимости ха-

рактеристики. Расчёт приведён в таблице 17.

Таблица 17 – Сравнительная оценка характеристик вариантов исполнения

	Весовой коэффициент параметра	Исп. 1	Исп. 2	Исп. 3
1. Повышение производительности труда пользователя	0.19	5	5	4
2. Удобство в эксплуатации	0.18	4	5	4
3. Совместимость с устройствами	0.09	5	2	4
4. Устойчивость к перебоям электропитания	0.11	5	4	5
5. Потребность в ресурсах памяти	0.13	5	3	4
6. Функциональные возможности	0.16	4	5	5
7. Скорость работы	0.14	4	4	5
ИТОГО	1	4.52	4.22	4.41

Общий интегральный показатель варианта исполнения вычисляется по формуле:

$$I_{исп} = \frac{I_{рес}}{I_{фин}} \quad (5)$$

где $I_{рес}$ – интегральный показатель ресурсоэффективности разработки;

$I_{фин}$ – интегральный финансовый показатель.

По общему показателю можно вычислить сравнительную эффективность вариантов использования по отношению к одному из них. Таким образом, можно определить, насколько дороже (эффективность больше единицы), или дешевле (эффективность меньше единицы) тот или иной вариант.

Все вышеперечисленные характеристики для каждого варианта исполнения представлены в сводной таблице 18.

Таблица 18 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0.76	0.83	1.00
2	Интегральный показатель ресурсоэффективности разработки	4.52	4.22	4.41
3	Интегральный показатель эффективности	5.94	5.07	4.41
4	Сравнительная эффективность вариантов исполнения	1.00	1.17	1.35

По итогам выполнения работы по оценке коммерческого потенциала, возможных альтернатив проведения научного исследования, определения ресурсной и финансовой эффективности исследования был найден наиболее подходящий вариант исполнения работы (вариант 1):

- Тип приложения: веб-приложение
- Платформа серверной части: ServiceStack
- Платформа клиентской части: Angular
- Хранение документов на стороне клиента и сервера
- Вёрстка документов на стороне сервера
- Система вёрстки: LaTeX
- Тип редактора документов: комбинированное решение
- Формат хранения документов: JSON
- Формат описания стандартов: расширение языка разметки

Этот вариант наиболее выгодный для разработки, так как требует наименьшее количество финансов для реализации. Также этот вариант наиболее привлекателен для будущих пользователей, так как в большинстве характеристик ресурсоэффективности он лучше, чем другие варианты.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИЗБ	Ветров Алексей Александрович

Институт	Кибернетики	Кафедра	Вычислительной техники
Уровень образования	Бакалавр	Направление / специальность	09.03.02 «Информационные системы и технологии»

Исходные данные к разделу «Социальная ответственность»:

<p><i>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</i></p>	<p><i>Объектом дипломной работы является веб-приложение, обеспечивающее автоматическую генерацию отчётов.</i></p> <p><i>Область применения: в сети Internet.</i></p> <p><i>Сфера применения: образовательные учреждения.</i></p>
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p><i>1. Производственная безопасность</i></p> <p><i>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</i></p> <ul style="list-style-type: none"> – физико-химическая природа вредности, её связь с разрабатываемой темой; – действие фактора на организм человека; – приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ); – предлагаемые средства защиты (сначала коллективной защиты, затем – индивидуальные защитные средства). <p><i>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</i></p> <ul style="list-style-type: none"> – термические опасности (источники, средства защиты); – электробезопасность (в т.ч. статическое электричество, молниезащита – источники, средства защиты); – пожаровзрывобезопасность (причины, профилактические мероприятия, первичные средства пожаротушения). 	<p><i>1.1 Производственная безопасность на стадии разработки и реализации системы:</i></p> <p><i>Вредные факторы:</i></p> <ul style="list-style-type: none"> – Монотонность работы – Нарушение параметров микроклимата – Недостаточность рабочего освещения <p><i>Опасные факторы:</i></p> <ul style="list-style-type: none"> – Поражение электрическим током – Возгорание
<p><i>2. Экологическая безопасность:</i></p> <ul style="list-style-type: none"> – защита селитебной зоны – анализ воздействия объекта на атмосферу (выбросы); – анализ воздействия объекта на гидросферу (сбросы); – анализ воздействия объекта на литосферу (отходы); – разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды. 	<p><i>2.1 Влияние объекта исследования на окружающую среду:</i></p> <ul style="list-style-type: none"> – чрезмерное потребление электроэнергии, картриджей, бумаги; <p><i>2.2 Мероприятия по защите окружающей среды:</i></p> <ul style="list-style-type: none"> – экономия электроэнергии; – использование бумаги с двух сторон; – экономия бумаги на этапе использования системы.

<p>3. Безопасность в чрезвычайных ситуациях:</p> <ul style="list-style-type: none"> – перечень возможных ЧС при разработке и эксплуатации проектируемого решения; – выбор наиболее типичной ЧС; – разработка превентивных мер по предупреждению ЧС; – разработка действий в результате возникшей ЧС и мер по ликвидации её последствий. 	<p>3.1 Возможные чрезвычайные ситуации:</p> <ul style="list-style-type: none"> – пожар; – взрыв. <p>3.2 Необходимые мероприятия по предотвращению возгорания.</p>
<p>4. Правовые и организационные вопросы обеспечения безопасности:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>4.1. В процессе разработки и использования требуется ПК, в соответствии с этим рассмотрены правовые и организационные нормы для работы с ПК:</p> <ul style="list-style-type: none"> – Организация рабочих мест. – Оформление трудовых отношений. – Рабочее время и время отдыха работников. – Оплата и нормирование труда. – Ответственность за нарушение трудового законодательства и иных актов, содержащих нормы трудового права. <p>4.2. Специфика влияния проекта:</p> <ul style="list-style-type: none"> – Сокращается использование бумаги – Уменьшается время на подготовку документа – Упрощение обмена документами с руководителем

Дата выдачи задания для раздела по линейному графику	06.02.2017
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов Пётр Анатольевич	–		06.02.2017

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗБ	Ветров Алексей Александрович		06.02.2017

6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Введение. В связи с развитием научных технологий, значимую роль играет безопасное исполнение сотрудниками своих профессиональных обязанностей. Для этого была создана наука о безопасности труда и жизнедеятельности человека. В основу данной науки легли мероприятия служащие для обеспечения безопасности и гигиены труда, промышленной безопасности, охраны окружающей среды и ресурсосбережению [13].

Цель написания данного раздела является принятие проектных решений, исключающих несчастные случаи в производстве, защиту здоровья работников и снижение вредных воздействий на окружающую среду, экономное расходование невозобновимых природных ресурсов.

Для этого были поставлены следующие задачи:

- а) Обнаружение и изучение вредных и опасных факторов, которые могут возникнуть на рабочем месте;
- б) Анализ влияния процесса разработки web-приложения на окружающую среду;
- в) Анализ вероятных ЧС;
- г) Правовые и организационные вопросы обеспечения безопасности.

Основное внимание в данной главе будет уделено рассмотрению следующих вопросов:

- а) определение оптимальных условий труда для более продуктивного рабочего процесса;
- б) расчет необходимого освещения в производственном помещении;
- в) расчет уровня шума.

6.1 Профессиональная и социальная безопасность

В данном разделе анализируются вредные и опасные факторы, которые могут возникать при разработке или эксплуатации проектируемого решения [14]. Так как разработка ведётся с применением персонального компьютера, она может спровоцировать следующие вредные и опасные факторы:

Вредные факторы:

- Монотонность работы;
- Влажность воздуха.

Опасные факторы:

- Недостаточная освещенность рабочего места;
- Статическое электричество;
- Высокое напряжение электрической сети.

Для предотвращения возникновения данных факторов необходимо рассмотреть меры защиты от данных факторов. В противном случае их воздействие может нанести вред работнику – от утомляемости и снижения работоспособности до травм и серьёзных проблем со здоровьем.

6.1.1 Монотонность работы

Монотонный режим работы заключается в некоторых видов работ, при которых человек долгое время выполняет однообразные элементарные действия либо имеет место быть предельная концентрация внимания на какой либо деятельности. Работа программиста очень сильно подвержена этому неблагоприятному фактору, так как при написании кода от программиста требуется очень большая концентрация и повышенное внимание. Этот фактор относится к психофизическому типу.

Для борьбы с монотонностью можно использовать следующие методы:

- усложнение рабочих операций, выполняемых действий, объединение их в комплексы;
- увеличение темпа работы или подачи информации (сигналов);
- расчленение общего задания на отдельные части для того, чтобы появились промежуточные (поэтапные) цели;
- прерываться на 5-минутный отдых.

6.1.2 Повышенная или пониженная влажность воздуха

Повышенная или пониженная влажность воздуха также играет значимую роль в процессе работы. Повышенная влажность воздуха создаёт неблагоприятные метеорологические условия – происходит нарушение терморегуляции и перегревание организма, уменьшается испарение пота, а следовательно, уменьшается и отдача тепла организмом, что резко ухудшает состояние и работоспособность человека. Низкая относительная влажность воздуха способствует испарению пота, в результате чего происходит быстрая отдача тепла организмом. Понижение относительной влажности воздуха до 20 % вызывает неприятное ощущение сухости слизистых оболочек верхних дыхательных путей.

Источником отклонений влажности воздуха в помещении могут являться тепловыделяющие приборы (например, ЭВМ и орг. техника), либо охлаждающие устройства (такие как кондиционеры).

Для того, чтобы устранить повышенную, либо пониженную влажность, необходимо правильно (в зависимости от требований) настроить систему вентиляции в помещении.

В соответствии с нормами СанПиН 2.2.4.548-96 [15] воздух бывает

- сухим (на 55% насыщен водяными парами);
- умеренно сухим (56% - 70%);
- умеренно влажным (71% - 85%);
- очень влажным (86% и более).

6.1.3 Недостаточная освещённость рабочего места

Недостаточная освещённость рабочего места способна затруднить выполнение работы, вызывать утомление, увеличивается риск получения производственной травмы. Длительное нахождение в условиях недостаточной освещённости приводит к снижению интенсивности обмена веществ в организме, ослаблением его отзывчивости, а также отрицательно влияет на близорукость.

Освещённость помещения можно разделить на два типа: комбинированное и общее. Так как рабочий процесс будет находиться в офисном помещении с окнами, то логично выбрать систему общего освещения. В соответствии с СП 52.13330.2011 [16] была определена необходимая освещённость для рабочего помещения. Так как в процессе программирования рабочий процесс в основном основывается на экране ЭВМ, то значение освещённости мы возьмем в 300 лк.

6.1.4 Статическое электричество

Статическое электричество может образовываться при эксплуатации электроустановок высокого напряжения постоянного тока и электризации диэлектрических материалов.

Предельно допустимый уровень напряженности электростатических полей (ЕПРЕД) устанавливается равным 60 кВ/м в течение 1 ч.

При напряженности электростатических полей менее 20 кВ/м время пребывания в электростатических полях не регламентируется [17].

Основными средствами против накопления статики в нашем случае являются:

- отвод зарядов путем заземления оборудования и коммуникаций;
- применение антистатических веществ.

6.1.5 Высокое напряжение электрической сети

Высокое напряжение электрической сети относится к опасным факторам на рабочем месте.

В процессе разработки информационной системы основные работы проводятся с помощью персональных компьютеров, которые могут быть источником поражения электрическим током.

Основные подверженные опасности процессы:

- риск возникновения короткого замыкания, в системном блоке либо периферийном оборудовании;
- непосредственное соприкосновение с оборудованием и внешним окружением помещения, оказавшимся под напряжением;
- поражение электрическим током в результате повреждения изоляции.

Сама рабочая зона скомпонована таким образом, что бы исключить возможность перетирания заизолированных проводов и контакта внешнего оборудования с возможным источником замыкания. Так же в электросхеме помещения имеются автоматы отключающие питание при возникновении короткого замыкания. В электрической схеме имеются коллективные способы электрозащиты такие как: изоляция токопроводящих частей, защитное заземление оборудования.

Так как работы по созданию информационной системы проходят в помещении с нормальными условиями, где влажность не превышает 50%, температура воздуха не более 27 градусов Цельсия, мебель не токопроводящая, и имеются изолированные полы с антистатическим покрытием.

Помещение, в котором проходят работы, относится к категории помещений с повышенной опасностью (ПУЭ [18]), так как в условиях производства используется значительное количество электроустановок.

В помещениях с повышенной опасностью для питания ЭВМ, светиль-

ников (в частности, светильников с люминесцентными лампами) должно применяться напряжение не выше 220 В переменного или постоянного тока. Защитными мерами являются зануление металлических корпусов светильников. Питание электроприемников должно выполняться от сети 380/220 В с системой заземления TN-S или TN-C-S. В зданиях следует применять кабели с медными жилами, если их расчетное сечение не превышает 16 мм² [18].

Защитные меры электробезопасности зданий должны выполняться в соответствии с требованиями ПУЭ. Во всех помещениях необходимо присоединять открытые проводящие части ЭВМ, светильников общего освещения к нулевому защитному проводнику. Применение подвесных светильников, не оснащенных зажимами для присоединения к нулевому защитному проводнику, не допускается. В групповой сети, питающей штепсельные розетки для переносных электрических приборов, должна быть обеспечена дополнительная защита при случайном непреднамеренном прикосновении к опасным токоведущим частям посредством устройства защитного отключения. В качестве дифференциальных автоматических выключателей использованы дифференциальные выключатели, не имеющие источника питания, зависящего от напряжения защищаемой сети [18]. В здании используется безопасная система заземления TN-S (нулевой рабочий и нулевой защитный проводники работают отдельно по всей системе) на ТП-6 кВ через РУ -0,4 кВ, соединенной со зданием пятижильным проводником трехфазной сети.

6.2 Экологическая безопасность

Работы по реализации проекта не оказывают значительного влияния на окружающую среду, так как в процессе работы не используются вредные химические соединения в концентрациях превышающих допустимые.

Основные устройства, используемые в работе, это ЭВМ и принтер, что ведет к чрезмерному потреблению энергии и различных расходных материалов:

картриджи, бумага.

Электропотребление в процессе работы происходит средствами из неисчерпаемых запасов воды (ГЭС). Воздействие гидроэлектростанций на экосистему обусловлено следующим: это нерегулируемые спуски воды из водохранилища ГЭС на несколько дней (обычно на 10-20 дней). В результате этих нерегулируемых, бесконтрольных спусков воды в реку, в пойме реки и окружающем районе могут произойти неблагоприятные экологические изменения, такие как затопление, что бес сомнений влияет на биосферу и гидросферу.

Чтобы решить проблему чрезмерного использования разных расходных материалов (картриджей, бумаги), для предварительных проверок можно использовать бумагу с двух сторон, также можно использовать экологическую бумагу (повторно переработанная). Необходимо бережнее использовать природные ресурсы, так как производство обычной бумаги оказывает негативное влияние на экологию, а экологическая бумага не используется повсеместно. При производстве обычной бумаги в воду и воздух вбрасываются токсичные вещества, такие как метанол, толуол, диоксид хлора и прочее. При производстве экологической бумаги требуется меньшее количество химических веществ. В тоже время, необходимо отметить, что при её производстве может появляться большее количество шлама. Также следует отметить, что для производства бумаги используется древесина. Лес относится к возобновляемым ресурсам, но необходимо снизить темп его вырубки. В противном случае человечеству может грозить экологическая катастрофа.

На этапе эксплуатации системы она будет способствовать экономии бумаги. Во-первых, в реализованной системе будет существовать возможность обмена отчётами без их печати. Во-вторых, вследствие использования шаблонов, содержащих всё необходимое форматирование, титульные листы, а также некоторую долю содержания отчёта, будет сокращаться количество вносимых ошибок. Это приведёт к снижению количества случаев, когда студенту необходимо заново печатать работу и тратить бумажные ресурсы.

6.3 Безопасность в чрезвычайных ситуациях

Согласно Федеральному закону «О защите населения и территорий от ЧС природного и техногенного характера» (1996 г.), в соответствии с государственным стандартом Российской Федерации ГОСТ Р 22.0.02-94 [19] «Безопасность в чрезвычайных ситуациях» были установлены термины для применения во всех видах документации и литературы по безопасности в чрезвычайных ситуациях.

Чрезвычайная ситуация (ЧС) – это обстановка на определенной территории, 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 среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Чрезвычайной ситуации предшествует возникновение источника чрезвычайной ситуации – опасного природного явления, аварии или опасного техногенного происшествия, широко распространенной инфекционной болезни людей, сельскохозяйственных растений и животных, а также применения современных средств поражения [20].

В этой главе будут рассмотрены возможные чрезвычайные ситуации техногенного характера, которые могут произойти в период разрабатываемого проекта, чтоб постараться их избежать. Будут рассмотрены угрозы пожаров, взрывов.

6.3.1 Анализ вероятных ЧС, которые могут возникнуть в период исследования

Вероятными ЧС, которые может инициировать объект исследований, являются взрыв или пожар. Основными источниками пожаров и взрывов на рабочем месте могут стать:

- электрический ток;
- удар молнии;
- статическое электричество. Основными причинами возникновения взрывов и пожара в помещениях с компьютерами являются:
 - наличие на рабочем месте и неграмотное использование бытовых приборов или их применение при наличии неисправности;
 - курение в местах, для этого не предназначенных;
 - применение нескольких электрических приборов одновременно, что ведет к перегрузке электросети;
 - оставление электроприборов без внимания и их перегрев; неправильное расположение электронагревателей – вблизи штор, покрывал;
 - неисправность розеток, электропроводки и выключателей, которые могут приводить к замыканию или пробое изоляции;
 - возгорание вследствие внешних воздействий (во время грозы возможно попадание молнии).

6.3.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС

Обеспечение безопасности в чрезвычайных ситуациях включает в себя:

- принятие и соблюдение правовых норм;
- выполнение эколого-защитных, отраслевых или ведомственных требо-

ваний и правил;

– также проведение комплекса организационных, экономических, эколого-защитных, санитарно-гигиенических, санитарно-эпидемиологических и специальных мероприятий, направленных на обеспечение защиты населения, объектов народного хозяйства и иного назначения, окружающей природной среды, от опасностей в чрезвычайных ситуациях [21].

Меры по устранению пожара, можно разделить на следующие категории:

- инструктаж персонала;
- создание инструкций и схем эвакуации;
- свободные пути эвакуации;
- доступ к средствам пожаротушения;
- поддержание необходимого уровня изоляций токопроводящих приборов;
- соблюдение мер технической безопасности при работах;
- автоматические стационарные системы тушения пожаров, первичные средства пожаротушения.

Основным средством пожаротушения являются порошковые огнетушители, расположенные согласно ГОСТ 12.4.009 [22].

Главными технически-организационными мероприятиями по улучшению постоянства функционирования объекта являются:

- эвакуация сотрудников;
- поддержка в состоянии готовности средств и сил для ликвидации чрезвычайных ситуаций [22].

6.4 Правовые и организационные вопросы обеспечения безопасности

6.4.1 Правовые нормы трудового законодательства

За состоянием безопасности труда установлены строгие государственные, ведомственный и общественный надзор и контроль. В основе нормативно-правовых актов в области безопасности жизнедеятельности лежат:

- Конституция РФ;
- Трудовой кодекс РФ;
- Кодекс "Об административных правонарушениях";
- Гражданский кодекс РФ;
- Федеральный закон "Об основах охраны труд в РФ";
- Основы законодательства об охране здоровья граждан;
- Закон РФ "О санитарно-эпидемиологическом благополучия населения".

Правовые нормы трудового законодательства [23]:

- организация рабочих мест;
- оформление трудовых отношений;
- рабочее время и время отдыха работников;
- оплата и нормирование труда;
- ответственность за нарушение трудового законодательства и иных актов, содержащих нормы трудового права.

Контроль за состоянием условий труда на предприятиях осуществляют специально созданные службы охраны труда совместно с комитетом профсоюзов. Контроль за состоянием условий труда заключается в проверке состояния производственных условий для работающих, выявлении отклонений от требований безопасности, законодательства о труде, стандартов, правил и норм охраны труда, постановлений, директивных документов, а также проверке выполне-

ния службами, подразделениями и отдельными группами своих обязанностей в области охраны труда. Этот контроль осуществляют должностные лица и специалисты, утвержденные приказом по административному подразделению. Ответственность за безопасность труда в целом по предприятию несут директор и главный инженер.

Чтобы спроектировать рабочую зону, отвечающую специальным правовым нормам трудового законодательства, необходимо использовать следующее:

- условия допуска лиц к работе;
- необходимость соблюдения правил внутреннего распорядка;
- запрещение курения и распития спиртных напитков;
- характеристика вредных и опасных факторов;
- нормы спецодежды;
- пожаробезопасность, взрывобезопасность;
- порядок уведомления администрации о неисправностях, об авариях, о травмах;
- способы оказания первой помощи пострадавшим;
- правила личной гигиены;
- порядок ответственности работающего.

6.4.2 Организационные мероприятия при компоновке рабочей зоны

При компоновки рабочей зоны необходимо учесть то, что основную часть работы инженер проводит перед монитором в неподвижном положении, следовательно, следует рассмотреть ряд вопросов, касающийся компоновки рабочего места с учётом эргономики.

Согласно ГОСТ 12.2.032-78 ССБТ [24] рабочее место для выполнения работ сидя организуют при легкой работе, не требующей свободного передви-

жения работающего, а также при работе средней тяжести в случаях, обусловленных особенностями технологического процесса. Конструкция рабочего места и взаимное расположение всех его элементов (сиденье, органы управления, средства отображения информации и т.д.) должны соответствовать антропометрическим, физиологическим и психологическим требованиям, а также характеру работы.

Высота рабочего места подбирается исходя из видов работ, которые будут выполняться на производстве, в данном случае работа автора будет связана с ЭВМ, следовательно, рабочее место будет настраиваться в пределах от 680-750 мм над уровнем пола. Конструкция регулируемого кресла должна обеспечивать оптимальную рабочую позу для конкретного роста.

Рабочая поверхность должна обеспечивать работу для дисплея, устройства ввода данных и дополнительного оборудования, материалов, а также для кистей рук и локтей пользователя. При работе в позах сидя и стоя также достаточно иметь пространственный зазор между телом пользователя и частями рабочей станции. Также пользователь должен иметь возможность наклонить или повернуть видеодисплей таким образом, чтобы сохранить ненапряженную рабочую позу независимо от высоты уровня глаз. Угол обзора не должен превышать 40 градусов по всей активной площади экрана.

В соответствии с ГОСТ Р ИСО 9241-4-2009 [25] при использовании стандартной съемной клавиатуры, следует учитывать тот факт, что конструкция клавиатуры может влиять на эффективность, результативность работы и удовлетворенность пользователя. Характеристики клавиатуры, которые влияют на эффективность работы, включают в себя: раскладки буквенных и числовых клавиш, лингвистические различия, физические характеристики отдельных клавиш и общую конфигурацию корпуса клавиатуры. Определенные характеристики клавиатуры могут влиять на осанку, принимаемую оператором, работающим с видеотерминалом. Высота основного ряда клавиатуры для оператора должна составлять от 30 до 35 мм. Угол наклона клавиатуры по отноше-

нию к оператору – от 50 до 120 и должен быть регулируемым. Также клавиатура должна быть легко установлена на рабочей поверхности, не скользить и не качаться.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы была разработана система для создания и оформления документов. Это приложение было спроектировано с учётом требований студентов Томского политехнического университета, обучающихся на технических специальностях и выполняющих лабораторные и курсовые работы во время обучения.

Для работы с приложением студентам необходимо иметь лишь современный браузер и подключение к сети Интернет. Также от пользователя не требуется дополнительных знаний об особенностях оформления печатных документов. Разработанный пользовательский интерфейс не перегружен элементами управления и интуитивно понятен. Это даёт возможность пользоваться приложением широкому кругу лиц, что упростит выполнение студентами отчётности и увеличит продуктивность обучения.

Узкая направленность является преимуществом при использовании системы для оформления отчётности, но в то же время она является недостатком при необходимости создания произвольных документов. Интерфейс пользователя фиксирует поведение стандартных элементов таким образом, чтобы пользователь не имел возможности нарушить встроенные в документ правила оформления. Но система не предоставляет возможности изменить эти правила через веб-приложение, для изменения шаблона необходимо редактировать его исходный код.

Разработанный формат описания документов позволяет задавать произвольные правила оформления, соответствующие синтаксису языка TeX. Таким образом, для создания новых шаблонов, необходим специалист, который знает формат описания документов и язык TeX. При наличии специалиста, система может быть внедрена в любом учреждении, в котором требуются документы, имеющие единые правила оформления.

СПИСОК ПУБЛИКАЦИЙ СТУДЕНТА

1. Ветров А. А., Хаустов П. А. Визуальный редактор математических формул для систем электронного документооборота [Электронный ресурс] // Молодежь и современные информационные технологии: сборник трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых. В 2 т., Томск, 7-11 ноября 2016. – Томск: ТПУ, 2017. – Т. 2. – С. 64-65. – Режим доступа: [http://portal.tpu.ru:7777/f_ic/files/science/activities/msit/msit2016/Sbornik_2016/Sbornik_MSIT_2016_\(Tom2\).pdf](http://portal.tpu.ru:7777/f_ic/files/science/activities/msit/msit2016/Sbornik_2016/Sbornik_MSIT_2016_(Tom2).pdf)
2. Белов А. С., Газизов А. Т., Ветров А. А. Расписание для студентов элитного образования [Электронный ресурс] // Ресурсоэффективным технологиям — энергию и энтузиазм молодых: сборник научных трудов V Всероссийской конференции студентов элитного технического образования, Томск, 25-27 Марта 2014. – Томск: ТПУ, 2014. – С. 192-193. – Режим доступа: <http://www.lib.tpu.ru/fulltext/c/2014/C08/C08.pdf>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Общие указания к выполнению лабораторных работ по курсу информатика [Текст]. – Томск : Издательство ТПУ, 2011. – 13 с. – Режим доступа: http://portal.tpu.ru/departments/kafedra/eafu/obrazovanie/informatika/Tab_met/ukazan_inf1.pdf.

2. LaTeX2e: An unofficial reference manual [Electronic resource]. – [S. 1.] : TUG, 2015. – oct. – Access mode: <http://tug.org/texinfohtml/latex2e.html>.

3. Robertson, Will. The X_YTEX reference guide [Text] / Will Robertson, Khaled Hosny. – [S. 1. : s. n.], 2013. – may. – 20 p. – Access mode: <http://mirror.unl.edu/ctan/info/xetexref/xetex-reference.pdf>.

4. Zaytsev, Juriy. ECMAScript 6 compatibility table [Electronic resource]. – [S. 1. : s. n.]. – Access mode: <https://kangax.github.io/compat-table/es6/>.

5. Collection of .NET Benchmarks ordered by most recent [Electronic resource]. – [S. 1.] : Liquidbit Ltd., 2011. – Access mode: <http://mono.servicestack.net/benchmarks/>.

6. Korotyа, Eugeniya. 5 Best JavaScript Frameworks in 2017 [Electronic resource]. – [S. 1.] : Hackernoon, 2017. – jan. – Access mode: <https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282>.

7. К.А., Канаев. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой [Текст] / Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. // Фундаментальные исследования. – 2015. – № 2-25. – С. 5569–5572. – Режим доступа: <https://www.>

fundamental-research.ru/ru/article/view?id=38464 (дата обращения: 30.05.2017).

8. ISO 8601:2004 Data elements and interchange formats – Information interchange – Representation of dates and times [Text]. – Geneva, Switzerland : International Organization for Standardization, 2004. – 33 p. – Access mode: <https://www.iso.org/standard/40874.html>.

9. Freed, Ned. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies [Text]. – RFC 2045. – 1996. – Nov. – Access mode: <https://rfc-editor.org/rfc/rfc2045.txt>.

10. Lerman, Julia. Programming Entity Framework [Text] / Julia Lerman ; Ed. by O'Reilly Media. – [S. l.] : O'Reilly Media, 2010. – 920 p. – ISBN: 978-0-596-80726-9.

11. А.А., Ветров. Визуальный редактор математических формул для систем электронного документооборота [Текст] / Ветров А.А., Хаустов П.А. // Молодежь и современные информационные технологии: сборник трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых. В 2 т., Томск, 7-11 ноября 2016. – Т. 2. – Томск : ТПУ, 2017. – С. 64–65. – Режим доступа: [http://portal.tpu.ru:7777/f_ic/files/science/activities/msit/msit2016/Sbornik_2016/Sbornik_MSIT_2016_\(Tom2\).pdf](http://portal.tpu.ru:7777/f_ic/files/science/activities/msit/msit2016/Sbornik_2016/Sbornik_MSIT_2016_(Tom2).pdf).

12. Стоунз, Ричард. PostgreSQL. Основы [Текст] / Ричард Стоунз, Нейл Мэттью. – СПб : Символ-Плюс, 2002. – 640 с. – ISBN: 5-93286-043-X, 1-861005-15-6.

13. IS CSR 26000:2011. Социальная ответственность организации. Требования [Текст].

14. Романенко, С.В. Методические указания по разработке раздела «Социальная ответственность» выпускной квалификационной работы магистра, специалиста и бакалавра всех направлений (специальностей) и форм обучения ТПУ [Текст] / С.В. Романенко, Ю.В. Анищенко. – Томск : Изд-во Томского политехнического универ-та, 2016. – 11 с.
15. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений. [Текст].
16. СП 52.13330.2011 Естественное и искусственное освещение. [Текст].
17. ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов [Текст]. – М. : Изд-во стандартов.
18. ПУЭ. Правила устройства электроустановок. Издание 7. [Текст].
19. ГОСТ Р 22.0.02-94 Безопасность в чрезвычайных ситуациях. Термины и определения основных понятий [Текст]. – М. : Изд-во стандартов.
20. Безопасность и защита человека в чрезвычайных ситуациях: Учебное пособие [Текст] / Святова Н.В., Мисбахов А.А., Кабыш Е.Г. [и др.]. – Казань : ТГГ-ПУ, 2011. – 132 с.
21. Безопасность жизнедеятельности: Учебн. [Текст] / Под ред. С. В. Белова. – М : Высшая школа, 1999. – 448 с.
22. ГОСТ 12.4.009-83. Система стандартов безопасности труда. Пожарная техника для защиты объектов. Основные виды. Размещение и обслуживание [Текст]. – М. : Изд-во стандартов.
23. Трудовой кодекс Российской Федерации (ТК РФ) [Текст].
24. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования [Текст]. – М. : Изд-во стандартов.

25. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре [Текст]. – М. : Изд-во стандартов.

ПРИЛОЖЕНИЕ А
CD ДИСК