

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
 Направление подготовки 09.04.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ) информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы

Модуль автоматизированного администрирования проектными задачами сотрудников проектной компании

УДК 004.658:005.8:005.962.13

Студент

Группа	ФИО	Подпись	Дата
8ВМ71	Реннит Андрей Андреевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Фадеев Александр Сергеевич	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Меньшикова Е. В.	к.ф.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Алексеев Н. А.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Кочегурова Е.А.	к.т.н.		

Томск – 2019 г.

Планируемые результаты обучения

Код ре- зульта та	Результат обучения (выпускник должен быть го- тов)
	Общепрофессиональные компетенции
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
	Профессиональные компетенции
P5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
	Общекультурные компетенции
P8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.

P10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
 Направление подготовки 09.04.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ) информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации <small>(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)</small>

Студенту:

Группа	ФИО
8BM71	Реннит Андрей Андреевич

Тема работы:

Модуль автоматизированного администрирования проектными задачами сотрудников проектной компании	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i>	
Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i>	<ul style="list-style-type: none"> – Актуальность разработки системы – Исследование – Проектирование – Результаты работы – Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; – Социальная ответственность; Заключение по работе.

Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	Презентация в формате *.pptx
Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Меньшикова Екатерина Валентиновна
Социальная ответственность	Алексеев Николай Архипович
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Инструменты реализации программного обеспечения	
Заключение	

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

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Фадеев Александр Сергеевич	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ71	Реннит Андрей Андреевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕ-
СУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ВМ71	Реннит Андрей Андреевич

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	1. Оклад доцента – 33664Р 2. Оклад инженера – 12663Р 3. Стоимость одного квт\ч – 5,8Р 4. Стоимость оборудования - 34600Р
2. Нормы и нормативы расходования ресурсов	5. 6-часовой рабочий день 6. 6-ти дневная рабочая неделя
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	7. Единый социальный налог – 27,1%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	8. Потенциальные потребители результатов исследования 9. SWOT-анализ
2. Разработка устава научно-технического проекта	10. Цели и результат проекта 11. Организационная структура проекта 12. Ограничения и допущения проекта
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	13. План проекта 14. Бюджет научного исследования 15. Реестр рисков проекта

Перечень графического материала

1. Матрица SWOT 2. Диаграмма Ганта 3. Потенциальные риски

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Меньшикова Е.В.	к.ф.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ71	Реннит Андрей Андреевич		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8ВМ71	Реннит Андрей Андреевич

Школа	ИШИТР	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Магистратура	Направление/специальность	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:	
Характеристика объекта исследования (вещество, материал, рабочая зона) и области его применения	Модуль автоматизированного администрирования проектными задачами сотрудников проектной компании
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности	1. Специальные правовые нормы трудового законодательства; 2. Организационные мероприятия при компоновке рабочей зоны.
2. Производственная безопасность 2.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения 2.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	Вредные факторы: 1. Отклонения показателей микроклимата; 2. Недостаточная освещенность рабочей зоны; Опасные факторы: 1. Электрический ток 2. Опасность возникновения пожара
3. Экологическая безопасность	Источники выбросов в атмосферу; Образование сточных вод и отходов. Мероприятия по снижению вредного воздействия на ОС
4. Безопасность в чрезвычайных ситуациях	Вероятные ЧС при разработке и эксплуатации проектируемого решения и меры по их предупреждению

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ШБИП	ООД Алексеев Н.А.	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ71	Реннит Андрей Андреевич		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
 Направление подготовки 09.04.01 «Информатика и вычислительная техника»
 Уровень образования магистратура
 Отделение школы (НОЦ) информационных технологий
 Период выполнения _____ (осенний / весенний семестр 2018/2019 учебного года)

Форма представления работы:

магистерская диссертация

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

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы: _____

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.09.2018	Получение задания	0
12.02.2019	Исследование темы, выбор методов и средств разработки	30
15.05.2019	Разработка системы видеомониторинга и идентификации обучающегося в дистанционном обучении	30
05.06.2019	Оформление пояснительной записки	20
05.06.2019	Социальная ответственность	10
05.06.2019	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Фадеев А.С.	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кочегурова Е.А.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа содержит 93 страницы, 14 рисунков, 28 таблиц, 32 источников, 4 приложений.

Ключевые слова: планирование; администрирование; проектные работы; распределение задач; информационная система.

Объектом исследования является алгоритм сетевого планирования. Цель работы - разработка информационного модуля для повышения эффективности планирования работ проектной компании. Данный модуль должен связать выполнение различных работ и процессов во времени, получив прогноз общей продолжительности реализации всего проекта, а также распределения задач по сотрудникам компании наиболее эффективным образом, основываясь на трудоспособности сотрудников, а также прогнозирования времени их выполнения.

В процессе исследования будет проведен анализ популярных алгоритмов сетевого планирования и прогнозирования, а также описан используемый язык программирования, шаблон проектирования и среда разработки. В результате исследования была спроектирована и разработана система планирования проектных работ.

Целевой аудиторией являются проектные организации.

В рамках развития проекта в будущем планируется:

- оптимизация исходного кода для повышения производительности;
- расширение функциональных возможностей;
- реализация алгоритмов прогнозирования проектных работ;
- распределение задач по сотрудникам проектной компании на основе анализа их трудоспособности по заданным параметрам.

Определения, обозначения, сокращения

ГОСТ – Межгосударственный стандарт.

НТИ – научно-техническое исследование.

ОС – операционная система.

ПК – персональный компьютер.

ПЭВМ – персональная электронно-вычислительная машина.

СанПиН – Санитарные нормы и правила.

СМИ – средства массовой информации.

ФСС – Фонд Социального Страхования.

ФФОМС – Федеральный фонд обязательного медицинского страхования.

JavaScript – прототипно-ориентированный сценарный язык программирования.

PHP - скриптовый язык программирования общего назначения

HTTP - протокол передачи гипертекста

TCP/IP - сетевая модель передачи данных, представленных в цифровом виде

Nginx – почтовый прокси-сервер

MSSQL - система управления реляционными базами данных

MVC – Model-View-Controller — схема использования нескольких шаблонов проектирования.

SocketIO - JavaScript-библиотека для веб-приложений и обмена данными в реальном времени

RabbitMQ - программный брокер сообщений

SWOT-анализ (Strengths, Weaknesses, Opportunities, и Threats – метод стратегического планирования, заключающийся в выявлении факторов внутренней и внешней среды организации.

Оглавление

Введение.....	14
1. Анализ предметной области.....	15
1.1 Сетевое планирование	15
1.1.1 Метод критического пути	16
1.1.2 Диаграмма Ганта	17
1.1.3 Метод оценки и пересмотра планов (PERT)	17
1.1.4 Метод графической оценки и анализа (GERT)	18
1.1.5 Метод статистических испытаний (метод Монте-Карло)	19
2. Аналитический обзор аналогичных систем	19
3. Архитектура разрабатываемого приложения.....	20
3.1 Функциональные требования	20
3.2 Варианты использования системы	20
3.3 Архитектура программной системы	21
4. Инструменты реализации программного обеспечения	23
4.1 Программно-аппаратная часть приложения.....	23
4.1.1 Язык программирования	23
4.1.2 Symfony	24
4.1.3 Среда разработки	25
4.1.4 Шаблон проектирования	25
4.2 Клиентская сторона пользовательского интерфейса	26
4.2.1 Язык программирования	26
4.2.2 Nuxt.js и Vue.js.....	27
4.2.3 Реализации пользовательского интерфейса программного продукта....	27
4.2.3.1 Sigma.js	28
4.2.3.2 Arbor.js.....	28
4.2.3.3 Dracula.js	28
4.2.3.4 D3.JS	28
4.2.3.5 Cytoscape.js	29
4.2 Вывод по главе	29
5. Разработка модуля планирования работ	30
5.1 Методика вычисления критического пути	30
5.2 Реализация механизма быстрого создания задач.....	31

5.3	Подготовка входных данных	34
5.4	Работа с сервером.....	36
6.	Результат	38
7.	Финансовый менеджмент.....	41
7.1	Предпроектный анализ	41
7.3.1	Оценка готовности проекта к коммерциализации.....	47
7.4	Коммерциализация результатов научно-технического исследования	49
7.5	Инициация проекта.....	50
7.5.1	Ограничения и допущения проекта	50
7.5.2	Организационная структура проекта	51
7.6	Планирование управления научно-техническим проектом	51
7.6.1	План проекта.....	51
7.6.2	Бюджет научного исследования.....	53
7.6.3	Специальное оборудование для научных (экспериментальных) работ .	53
7.6.4	Расчет основной заработной платы исполнителей проекта	54
7.6.5	Дополнительная заработная плата научно-производственного персонала	56
7.6.6	Отчисления на социальные нужды	56
7.7.7	Накладные расходы	57
7.7.8	Расчет затрат на электричество	57
7.7.9	Формирование бюджета затрат проекта.....	57
7.8	Реестр рисков проекта	58
8	Социальная ответственность	60
8.1	Производственная безопасность	61
8.1.1	Вредные производственные факторы	62
8.1.1.1	Отклонение показателей микроклимата в помещении	62
8.1.1.2	Недостаточная освещенность рабочей зоны	64
8.1.2	Опасные производственные факторы	66
8.1.2.1	Опасность поражения электрическим током	66
8.1.2.2	Пожарвзрывобезопасность.....	68
8.2	Экологическая безопасность.....	69
8.2.1	Анализ воздействия продукта на окружающую среду	69
8.2.2	Решения по обеспечению экологической безопасности.....	69
8.3	Безопасность в чрезвычайных ситуациях.....	70
8.3.1	Перечень возможных ЧС при разработке и эксплуатации проектируемого решения	70

8.3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.....	71
6.4 Правовые и организационные вопросы обеспечения безопасности	73
8.4.1 Организационные мероприятия при компоновке рабочей зоны	74
8.4.2 Специфика влияния продукта на рабочий процесс.....	75
Заключение	77
Conclusion.....	78
Список источников	79
ПРИЛОЖЕНИЕ А	83
ПРИЛОЖЕНИЕ Б.....	84
ПРИЛОЖЕНИЕ В	85
ПРИЛОЖЕНИЕ Г	86

Введение

В современной экономике существуют организации, работы которых осуществляются на основе проекта. Проектное управление заключается в следующем: выбираются основные характеристики проектируемого оборудования или определяются конечные цели, а уже потом предпринимается попытка достичь их в соответствии с некоторыми временными и денежными ограничениями [1].

С учетом ограниченности ресурсов, высокую важность приобретает необходимость максимизации их эффективного использования за счет высокой оптимизации планирования и возможности маневрировать ресурсами [1]. Необходимость планирования проектных работ и строгого соблюдения плана также объясняется жесткими требованиями, предъявляемыми к рентабельности капиталовложений.

Для планирования проектных работ существует множество методов и средств, однако, их реализация зачастую не автоматизирована. В основном проектные предприятия используют программные приложения для визуализации работ, а также для контроля рабочего времени сотрудников. В таком случае предприятие оперирует только фактическими данными проекта. Следовательно, планирование проектных работ имеет высокую вероятность быть малоэффективным и неперспективным.

При проектировании объектов гражданского и промышленного назначения существует ряд особенностей для планирования:

- 1) Четкое разделение труда – работники из разных отделов никак не связаны между собой
- 2) Связанность проектных работ – работа одного раздела не может начаться пока другой не закончит свою.
- 3) Частые корректировки работ в связи с требованиями заказчика
- 4) Множественные дополнительные работы по объекту

Все вышеперечисленное непосредственно влияет на срок выполнения проектных работ, что усложняет процесс планирование, а также делает его более ценным для предприятия.

Целью данной магистерской диссертации является разработка информационного модуля для повышения эффективности планирования работ проектной компании. Данный модуль должен связать выполнение различных работ и процессов во времени, получив прогноз общей продолжительности реализации всего проекта, а также распределения задач по сотрудникам компании наиболее эффективным образом, основываясь на трудоспособности сотрудников, а также прогнозирования времени их выполнения.

Для реализации программного модуля планирования и прогнозирования проектных работ были сформулированы следующие задачи:

- 1) Провести анализ популярных алгоритмов сетевого планирования и прогнозирования.
- 2) Проанализировать системы планирования проектных работ.
- 3) Сформулировать основные функциональные требования разработки.
- 4) Выбрать методы и средства разработки.
- 5) Спроектировать архитектуру работы системы.
- 6) Разработать первую версию инструмента планирования проектных работ.
- 7) Провести тестирование разработанного инструмента на наличие ошибок.
- 8) Исправить ошибки, найденные на этапе тестирования.
- 9) Оценить полученные результаты работы программного приложения.

1. Анализ предметной области

Объектом исследования являются способы и средства планирования проектных работ проектной компании. Предметом исследования является сетевое планирование (сетевой анализ) и его методы.

1.1 Сетевое планирование

Сетевое планирование — это метод управления, используемых для анализа, контроля и мониторинга бизнес-процессов. Данный метод основывается на использовании математического аппарата теории графов и системного подхода для отображения и алгоритмизации комплексов взаимосвязанных работ В отличие от структуры

разбивки работ, сетевой план учитывает хронологический порядок действий, этапов и задач, их продолжительности и зависимостей и визуализирует их графически или в виде таблицы [2].

Методы сетевого анализа используются в управлении проектом, где элементы являются ключевыми действиями проекта во взаимосвязи во времени. Методы сетевого анализа фокусируются на расчете или оптимизации критического пути между элементами. К основным методам анализа сети относятся:

- Метод критического пути (МКП)
- Диаграмма Ганта
- Метод оценки и пересмотра планов (ПЕРТ, PERT)
- Метод графической оценки и анализа (GERT)
- Метод статистических испытаний (метод Монте-Карло)

1.1.1 Метод критического пути

Метод критического пути (МКП) является одним из основных детерминированных методов анализа сети. Его целью является определение продолжительности проекта на основе критической длины пути, которая представляет собой последовательность взаимозависимых действий. МКП позволяет облегчить эффективную временную координацию отдельных последовательных действий в рамках проекта [3].

Критический путь определяется как максимально длинный путь от начальной точки до конечной точки графика. У каждого проекта есть хотя бы один критический путь. Каждый критический путь состоит из списка видов деятельности, на которых руководитель проекта должен сосредоточиться больше всего, если он хочет обеспечить своевременное завершение проекта. Дата завершения последней задачи на критическом пути также является датой завершения проекта. Для критических задач, которые должны быть применены, общий запас времени и, следовательно, запас свободного времени равен нулю, то есть, что задержка начала этой задачи или увеличение ее продолжительности повлияет на конечную дату проекта. Критический путь отражается в графике и управлении проектом на всех этапах жизненного цикла проекта [3].

Метод МКП может быть использован в качестве инструмента в основном для оценки продолжительности проекта. Он используется для линейных проектов, где

продолжительность может быть оценена с высокой степенью точности, например, в строительной отрасли. Продолжительность деятельности по проекту обычно известна из прошлого опыта и знаний из данных о прошлых проектах. Длительности не определены статистически.

1.1.2 Диаграмма Ганта

Диаграмма Ганта — это графическое представление запланированной последовательности действий во времени, который используется для управления проектом. Автор - Генри Лоуренс Гант. Диаграмма Ганта отображается в столбцах (по горизонтали) за период времени, в который она планируется. В зависимости от продолжительности запланированного проекта, он отображается как период в соответствующих деталях (годы, месяцы, недели, дни). Затем в строках (по вертикали) отображаются частичные действия (иногда называемые задачами) - шаги, действия или подпроекты в порядке, соответствующем их логической последовательности в запланированном проекте. Продолжительность действия тогда связана с периодом времени [4].

Диаграмма Ганта обычно используется для планирования деятельности в рамках проекта. На практике используется простая форма диаграммы Ганта для графического представления действий в рамках проекта во времени. Это можно реализовать с помощью таблицы в простых офисных приложениях. Более сложная форма диаграммы Ганта представляет собой отображение различных преемственностей (емкости, материала, технологии и т. д.) Между различными видами деятельности. Этот метод планирования деятельности происходит от метода МКП.

1.1.3 Метод оценки и пересмотра планов (PERT)

Метод PERT является одним из стандартных методов сетевого анализа. Метод PERT является обобщением метода критического пути (МКП). PERT используется для управления сложными действиями, имеющими стохастический характер. Здесь продолжительность каждого подэтапа понимается как случайная величина, имеющая распределение вероятностей. Опытным путем было обнаружено, что на практике бета-распределение описывает его наилучшим образом, поскольку оно лучше отражает изменчивость условий эксплуатации [5].

Целью моделей PERT является такое расположение мероприятий, которое обеспечивало бы соблюдение сроков выполнения проекта с достаточно высокой вероятностью. Основное отличие от СРМ состоит в том, что продолжительность действия точно не известна, но она дается только с определенной вероятностью. Эта длительность не постоянная, а случайная величина с определенным распределением вероятностей. Из-за характера проблем, рассматриваемых в управлении проектами, бета-версия была выбрана для классических практик. Это распределение очень близко к нормальному распределению, оно непрерывное, одиночное, слегка асимметричное, но в отличие от нормального оно ограничено с обеих сторон [5].

Этот метод может использоваться, как и другие методы анализа сети, для оценки продолжительности проекта. Он используется в качестве альтернативы МКП. PERT — это метод, в основном используемый в управлении проектами и в области логистики и транспорта.

1.1.4 Метод графической оценки и анализа (GERT)

Методика графической оценки и анализа, широко известная как GERT, представляет собой метод сетевого анализа, используемый в управлении проектами, который позволяет проводить вероятностную обработку как сетевой логики, так и оценки продолжительности деятельности. Эта методика была впервые описана в 1966 году доктором Аланом Б. Прицкером из Университета Пердью и У. В. Хаппом [6].

По сравнению с другими методами, GERT редко используется в сложных системах. Тем не менее, подход GERT учитывает большинство ограничений, связанных с техникой PERT / МКП. GERT-сети позволяют более адекватно задавать сложные процессы строительного производства в тех случаях, когда затруднительно или невозможно (по объективным причинам) однозначно определить какие именно работы и в какой последовательности должны быть выполнены для достижения намеченного результата (т.е. существует многовариантность реализации проекта) [6].

Следует отметить, что "ручной" расчет GERT-сетей, моделирующих реальные процессы, чрезвычайно сложен, однако программное обеспечение для вычисления сетевых моделей такого типа в настоящее время, к сожалению, не распространено.

1.1.5 Метод статистических испытаний (метод Монте-Карло)

Методы Монте-Карло, или эксперименты Монте-Карло, представляют собой широкий класс вычислительных алгоритмов, которые полагаются на многократную случайную выборку для получения численных результатов. Основная концепция заключается в использовании случайности для решения проблем, которые могут быть в принципе детерминированными. Они часто используются в физических и математических задачах и наиболее полезны, когда трудно или невозможно использовать другие подходы. Методы Монте-Карло в основном используются в трех классах задач: оптимизация, численное интегрирование и генерация розыгрышей из распределения вероятностей [7].

2. Аналитический обзор аналогичных систем

В ходе работы был выполнен аналитический обзор различных систем планирования с использованием сетевого графика. Были рассмотрены одни из самых популярных программных решений: приложение «MsProject», приложение «Omniplan», веб приложение «Graph Online» и веб приложение «Graph editor».

В приложении А представлена таблица сравнения данных программных продуктов. Основными недостатками программных продуктов «Graph Online» и «Graph editor» являются невозможность работать с базой данных. Следовательно, данные решения подходят только для визуализации сетевого графика, а не для взаимодействия с реальными задачами проекта. Такие крупные программные решения как «MsProject» и «Omniplan» имеют достаточный функционал и возможность работы с ранее созданными задачами проекта, но являются достаточно нетривиальными в использовании и требуют большого опыта использования или больших временных затрат для удобной работы. Также

«Omniplan» доступен только на операционных системах, разработанных компанией «Apple» (iOS и Mac OS), а «MsProject» хоть и имеет возможность работы online, но является достаточно требовательной к ресурсам системы.

Поэтому разрабатываемый модуль администрирования проектных задач должен обладать следующими ключевыми отличительными особенностями:

- Работа с базой данных
- Простота в использовании

- Кроссплатформенность

3. Архитектура разрабатываемого приложения

Модуль автоматизированного администрирования проектными задачами сотрудников проектной компании предназначен для более удобного и эффективного планирования проектных работ компании. Качественное планирование проектных работ не только сокращает трудозатраты и время выполнения проекта, но и увеличивает рентабельность проекта и организации в целом, тем самым обеспечивает высокую конкурентоспособность. Одной из целей использования системы – удобное и эффективное планирование.

3.1 Функциональные требования

Функциональные требования определяют функциональные возможности системы, такие как какие именно услуги система должна предоставлять, как она должна реагировать на конкретные исходные данные и как система должна вести себя в определенных ситуациях. Функциональные требования зависят от типа программного обеспечения, от типа системы, в которой используется это программное обеспечение, а также от ожиданий пользователя [8].

Проектируемый модуль планирования должен предоставлять следующие функциональные возможности:

- Создание вершин
- Создание рёбер
- Фильтр графика по шифру раздела
- Просмотр критического пути проекта
- Автоматическое создания «начального события» графика
- Отмена предыдущего действия
- Группировка множества последовательных вершин
- Полноэкранный режим
- Выравнивание элементов графика относительно «начального события»

3.2 Варианты использования системы

Первым этапом проектирования автоматизированного модуля администрирования проектных задач была Use Case диаграмма. Диаграмма вариантов использова-

ния состоит из актеров, для которых система производит действие и собственно действия Use Case, которое описывает то, что актер хочет получить от системы. Актер обозначается стилизованным человечком, а действие – эллипсом с надписью, обозначающей действие в системе [9]. Диаграмма вариантов использования для данного программного продукта изображена в приложении Б.

Данная диаграмма вариантов использования отражает отношение между актёрами и прецедентами. Актерами являются пользователи системы. Пользователи могут управлять задачами и связями между ними, просматривать дополнительную информацию о задаче, просматривать критический путь проекта, а также управлять текущим графиком сетевого плана.

3.3 Архитектура программной системы

Диаграммы развёртывания (Deployment diagrams): используется для изображения связи программных компонентов, загружаемых вычислительной средой, связей между ними и моделирования других физических аспектов системы.

Программный продукт разделяется на серверный и клиентские узлы. Данные узлы связаны ассоциацией «один ко многим», потому что один сервер обслуживает несколько клиентов. Протоколами связи между узлами являются HTTP и TCP/IP протоколы. Физический узел «Application Server» состоит из компонентов «Nginx», «RabbitMQ», «MSSQL», «PHP», «SocketIO». Каждый программный компонент сервера упакован в своей Docker контейнер. Все Docker контейнеры объединены между собой виртуальной сетью посредством Docker-compose. В узле «User Client» пользователь взаимодействует с компонентом «Web Client», находясь в среде «Web Browser». Взаимодействие узлов «User Client» и «Application Server» происходит через узел «Web Server», с помощью компонента «Nginx». Диаграмма развёртывания представлена в приложении В.

Диаграмма компонентов показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. Она позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними. [10]

Программное приложение разделяется на две части: клиентская и серверная части. Клиентская часть состоит из компонента «Web client». Данные компоненты связаны с «Application server» с помощью интерфейса HTTP. Согласно диаграмме компонентов, представленной на рисунке 1, «Application server» реализовывает в себе бизнес-логику программного продукта.

Серверный компонент имеет один порт, который связывается с Nginx через HTTP протокол. В зависимости от URL в заголовке HTTP запроса Nginx либо отдает HTML страницу для веб-клиента, либо если URL удовлетворяет API, передает запрос на PHP сервер. PHP сервер взаимодействует с MS SQL сервером, а так же с RabbitMQ по протоколу AMQP, также PHP взаимодействует с компонентом Socket.io по протоколу Web-socket.

Общий поток данных выглядит следующим образом: клиентские приложения отправляют запрос на API, PHP обрабатывает запрос, получает данные из базы данных, генерирует ответ и отправляет ответ на запрос. Затем PHP отправляет сообщение в Socket.io об изменениях каких-либо сущностей, данное сообщение приходит всем клиентским приложениям, которые установили соединение с Socket.io. Помимо этого, PHP помещает сообщение в очередь RabbitMQ, о необходимости вызова процедуры у любого из получателей сообщений. PHP считывает сообщение из очереди, выполняет какие либо действия и отправляет сообщение в Socket.io сервер. Компоненты приложения, а также их взаимодействие изображено на рисунке 1.

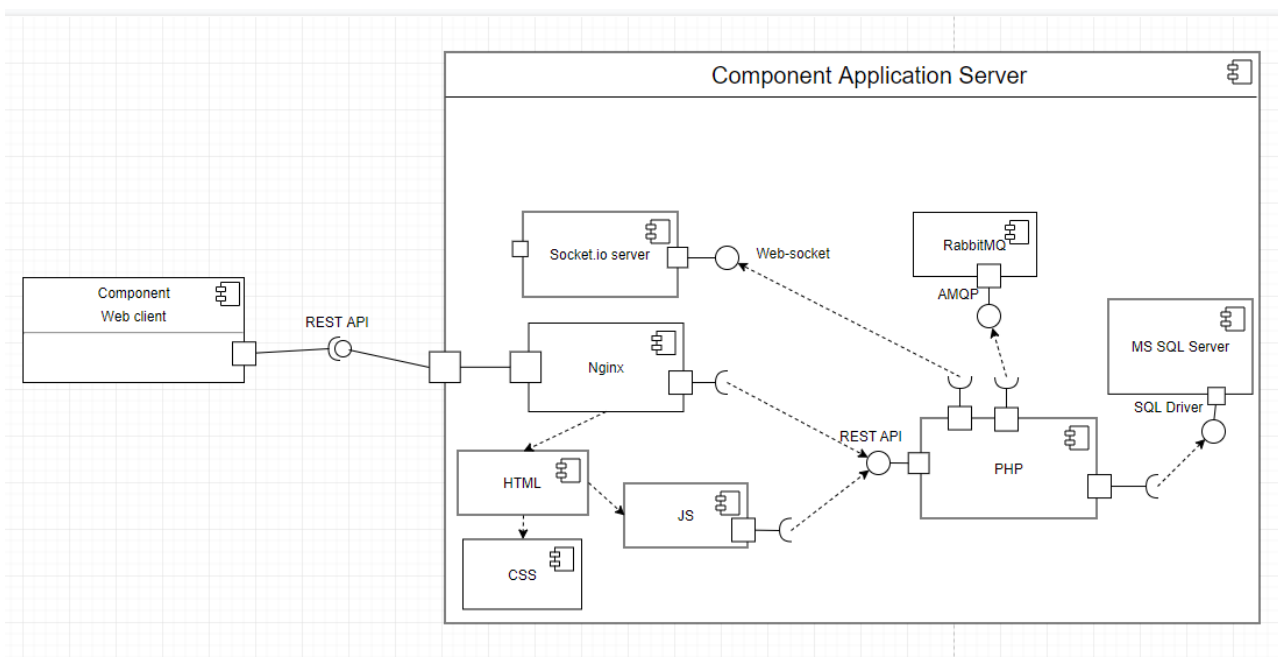


Рисунок 1 – Диаграмма компонентов серверной составляющей приложения

4. Инструменты реализации программного обеспечения

Согласно архитектуре разрабатываемого приложения необходимо разработать как клиентскую часть приложения, так и серверную.

4.1 Программно-аппаратная часть приложения

Программно-аппаратная часть – часть компьютерного приложения или кода программы, которое обеспечивает его функциональные возможности и к которым пользователь не имеет доступ. Большинство данных и операционный синтаксис хранятся и доступны в серверной части компьютерной системы. Программно-аппаратная часть также называется уровнем доступа к данным программного или аппаратного обеспечения и включает в себя любые функциональные возможности, к которым необходимо обращаться и осуществлять навигацию с помощью цифровых средств [11].

4.1.1 Язык программирования

Для разработки серверной части был выбран скриптовый язык PHP, интенсивно применяемый для разработки веб приложений.

PHP — это сценарный (скриптовый) серверный язык программирования общего назначения с открытым исходным кодом.

Сценарий — это набор инструкций по программированию, который интерпретируется во время выполнения. Язык сценариев — это язык, который интерпретирует сценарии во время выполнения. Скрипты обычно встраиваются в другие программные среды. Целью сценариев обычно является повышение производительности или выполнение рутинных задач для приложения. Серверные сценарии интерпретируются на сервере, в то время как клиентские сценарии интерпретируются клиентским приложением.

PHP — это серверный скрипт, который интерпретируется на сервере, а JavaScript - пример клиентского скрипта, который интерпретируется клиентским браузером. Как PHP, так и JavaScript могут быть встроены в HTML-страницы [12]. PHP обладает следующими преимуществами:

- Открытый исходным код.
- Низкий порог вхождения по сравнению с другими языками.
- Крупное сообщество программистов.
- Большинство серверов веб-хостинга поддерживают PHP по умолчанию, в отличие от других языков, таких как ASP, для которых требуется IIS. Это делает PHP экономически эффективным выбором.
- Регулярные обновления.
- PHP имеет встроенную поддержку для работы с MySQL. Также PHP может взаимодействовать с другими системами управления базами данных (Postgres, Oracle, Ms SQL Server, ODBC и т.д.)
- Кроссплатформенность — возможность использования в различных операционных системах, таких как Windows, Linux, Mac OS и т. д.

4.1.2 Symfony

В качестве фреймворка для разработки был выбран ведущий PHP-фреймворк для создания сайтов и веб-приложений Symfony.

Symfony — бесплатный PHP веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller — модель-представление-контроллер), который выпущен под лицензией MIT [13].

Основными преимуществами Symfony являются:

- Высокая гибкость.
- Простое тестирование.
- Долгосрочная поддержка и совместимость.
- Структурированная и понятная документация.

4.1.3 Среда разработки

Для создания программного продукта использована интегрированная среда разработки JetBrains PhpStorm.

PhpStorm – это интегрированная среда разработки на PHP с интеллектуальным редактором, которая глубоко понимает код, поддерживает PHP 5.3-7.3 для современных и классических проектов, обеспечивает лучшее в индустрии автодополнение кода, рефакторинг, предотвращение ошибок налету и поддерживает смешивание языков.

PhpStorm построен на IntelliJ IDEA, который написан на Java. Пользователи могут создавать свои собственные плагины.

4.1.4 Шаблон проектирования

Для проектирования приложения используется шаблон MVC (Model-View-Control).

MVC — это паттерн проектирования веб-приложений, который включает в себя несколько более мелких шаблонов. При использовании MVC модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента, благодаря чему модификация одного из них оказывает минимальное воздействие на остальные или не оказывает его вовсе. Основная цель применения MVC состоит в разделении данных и бизнес-логики от визуализации (внешнего вида). За счет такого разделения повышается возможность повторного использования программного кода и упрощается сопровождение (изменения внешнего вида, например, не отражаются на бизнес-логике) [14]. Основными компонентами данного шаблона проектирования являются:

- Модель. Модель является центральным компонентом данного шаблона проектирования. Она представляет собой динамическую структура данных приложения, независимых от пользовательского интерфейса. Также модель напрямую управляет данными, логикой и правилами приложения.

– Представление. Представление – это любое представление информации, такое как диаграмма или таблица. Возможны несколько представлений одной и той же информации

– Контроллер. Контроллер принимает входные данные и преобразует их в команды для модели или представления.

На рисунке 2 показана взаимосвязь между компонентами модель, представление и контроллер.

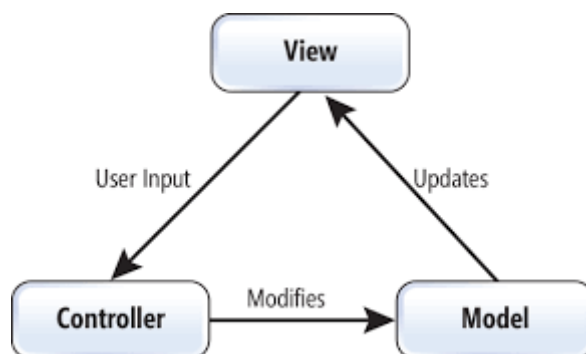


Рисунок 2 – Взаимосвязь компонентов шаблона MVC

Данный паттерн проектирования определяет взаимодействия между его компонентами:

– Модель отвечает за управление данными приложения. Модель получает пользовательский входные данные от контроллера.

– Представление обозначает представление модели в определенном формате.

Контроллер реагирует на входные данные пользователя и выполняет взаимодействия с объектами модели данных. Контроллер получает входные данные, при необходимости проверяет их и затем передает входные данные в модель.

4.2 Клиентская сторона пользовательского интерфейса

4.2.1 Язык программирования

Клиентская часть программного продукта была разработана на языке программирования JavaScript.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript. JavaScript обычно используется как встраиваемый язык

для программного доступа к объектам приложений [15]. JavaScript является клиентским языком сценариев, что означает, что исходный код обрабатывается веб-браузером клиента, а не веб-сервером. Функции JavaScript могут запускаться после загрузки веб-страницы без связи с сервером.

Подобно языкам сценариев на стороне сервера, таким как PHP и ASP, код JavaScript можно вставлять в любое место HTML-кода веб-страницы.

Основными преимуществами использования JavaScript являются:

- Универсальная поддержка. Все популярные современные веб-браузеры поддерживают JavaScript со встроенными интерпретаторами.
- Структурированность. JavaScript поддерживает большую часть синтаксиса структурированного программирования из C.
- Объекто-ориентированный подход. JavaScript почти полностью основан на объектах. В JavaScript объект — это ассоциативный массив, дополненный прототипом.

4.2.2 Nuxt.js и Vue.js

В качестве фреймворка для написания клиента модуля администрирования проектных работ была использована связка Nuxt.js и Vue.js.

Nuxt.js — это бесплатный веб-фреймворк с открытым исходным кодом, основанный на Vue.js, Node.js, Express.js, Webpack и Babel.js. Основной задачей этого фреймворка является рендеринг пользовательского интерфейса в условиях абстракции от клиент-серверной архитектуры [16].

Vue.js — это JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле [17].

4.2.3 Реализации пользовательского интерфейса программного продукта

Для визуализации автоматизированного модуля администрирования проектных задач было необходимо выбрать наиболее подходящую JavaScript библиотеку. В рамках работы над магистерской диссертацией были рассмотрены следующие решения.

4.1.1 Sigma.js

Sigmajs — это библиотека JavaScript, предназначенная для представления графов. Это облегчает публикацию сетей на веб-страницах и позволяет разработчикам интегрировать исследование сети в многофункциональные веб-приложения.

Данная библиотека предоставляет следующие функции:

- Поддержка действий мыши и касания.
- Обновлением и масштабированием при изменении размера контейнера.
- Рендеринг WebGL и Canvas.
- Перенастройкой графика и адаптацией размеров узлов и ребер к экрану.
- Кастомизация отображения узлов и ребер.
- Открытый API с поддержкой добавления новых функций и прослушиванием событий.

4.1.2 Arbor.js

Arbor.js – это библиотека визуализации графов, созданная с помощью веб-работников и jQuery. Вместо того, чтобы пытаться быть всеобъемлющим фреймворком, arbor предоставляет эффективный алгоритм принудительной компоновки плюс абстракции для организации графов и обработки обновления экрана.

- Работа с canvas, SVG или даже позиционированными элементами HTML.
- Кастомизация ребер и узлов.
- Открытый исходный код.

4.1.3 Dracula.js

Dracula.js представляет собой набор инструментов для отображения и размещения интерактивных связанных графов и сетей, а также различных связанных алгоритмов из области теории графов.

- Работа с canvas, SVG.
- Открытый исходный код.
- Кастомизация ребер и узлов.

4.1.4 D3.JS

D3.js — это библиотека JavaScript для управления документами, в основе которых лежат данные. D3 помогает претворить данные в жизнь, используя HTML, SVG и CSS. D3 позволяет привязывать произвольные данные к DOM, и затем применять

результаты манипуляций с ними к документу. Данная библиотека имеет обширный функционал, но для разработки инструмента планирования проектных работ могут быть использованы следующие функции:

- Работа с SVG.
- Открытый API, содержащий несколько сотен функций.

4.1.5 Cytoscape.js

Cytoscape.js - это библиотека теории графов с открытым исходным кодом, написанная на JS. Cytoscape.js предназначена для анализа и визуализации графиков. Cytoscape.js имеет следующий функционал:

- Большой набор тестов, которые можно запустить в браузере или терминале.
- Поддерживает селекторы для краткой фильтрации и запросов к графам.
- Использует таблицы стилей для отделения представления от данных без учета рендеринга.
- Абстрагированные и унифицированные сенсорные события поверх знакомой модели событий
- Встроенная поддержка стандартных жестов как на рабочем столе, так и на сенсорном для удобства.
- Поддерживает теорию множеств.
- Включает алгоритмы теории графов, от BFS до PageRank.
- Кастомизация узлов, ребер, а также их анимации.

4.2 Вывод по главе

Исходя из вышеописанного, большинство существующих библиотек предназначены только для визуализации графов, а не для взаимодействия с ними. Основная проблема заключается в том, что большую часть функционала необходимо разработать самостоятельно. Следовательно, на основе анализа существующих javascript библиотек была выбрана Cytoscape.js. Так как данная библиотека имеет уже реализованные механизмы создания, позиционирования элементов, а также их анимации. Помимо этого, выбранная библиотека имеет реализованные алгоритмы теории графов, что упрощает задачу реализации программного продукта.

5. Разработка модуля планирования работ

В процессе разработки возникли трудности с реализацией некоторых функциональных требований.

5.1 Методика вычисления критического пути

Последовательность операций, требующая наибольшего времени для своего завершения, определяет наименьшее время, за которое может быть выполнен проект. Это время называют длительностью выполнения проекта. Указанная последовательность операций, определяющая длительность проекта называется критическим путем. Критический путь всегда начинается с самого первого события графика и проходит через весь график, заканчиваясь последним событием [17].

Метод вычисления критического пути разработан на основе алгоритма Беллмана — Форда и др. Данный алгоритм используется для поиска кратчайшего пути во взвешенном графе. Алгоритм находит кратчайшие пути от одной вершины графа до всех остальных. В отличие от алгоритма Дейкстры, алгоритм Беллмана — Форда допускает рёбра с отрицательным весом.

Алгоритм маршрутизации RIP (алгоритм Беллмана — Форда) был впервые разработан в 1969 году, как основной для сети ARPANET [18].

Вышеописанный алгоритм был использован для нахождения последовательности задач, которая требует наибольшее времени реализации. Так как алгоритм имеет возможность работы с отрицательным весом рёбер, все веса были инвертированы (умножены на -1) для получения необходимого результата.

Основной проблемой при реализации была техническая возможность строить граф, содержащий циклы задач, что невозможно для вычисления критического пути, а также бессмысленно с точки зрения планирования проектных работ предприятия.

Для устранения этой проблемы было принято решение искать возможный цикл при создании рёбер между вершинами графика сетевого планирования. При создании рёбра, которое приведет к появлению цикла пользователь будет уведомлён об обнаружении цикла, а данное ребро будет удалено (Рисунок № 3,4).

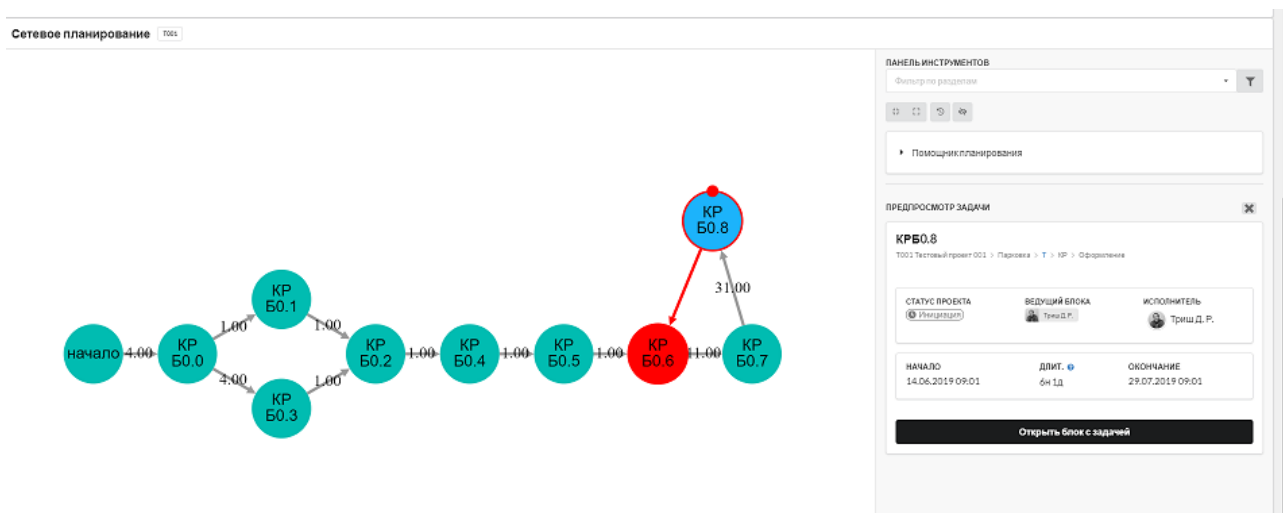


Рисунок 3 – Попытка создания цикла проектных задач

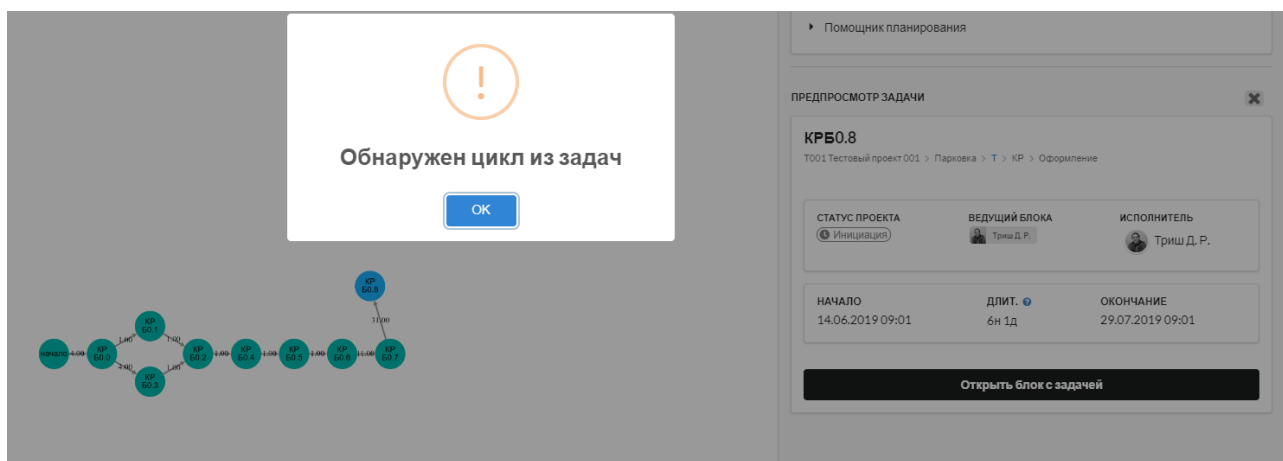


Рисунок 4 – Уведомление пользователя об обнаружения цикла

5.2 Реализация механизма быстрого создания задач

Одним из функциональных требований разрабатываемого приложения было разработать механизм быстрого создания задач для экономии времени при планировании проектных работ. Данный функционал подразумевает создание задачи связанной с текущей, трудозатраты которой равняются одному рабочему дню (8 часам). Время начала и окончания задачи должны быть равными времени начала и окончания текущей задаче, т.е. от которой производилось «быстрое создание». Исполнитель задачи и раздел в котором она находится должны соответствовать текущей задаче.

При разработке данного функционала была обнаружена проблема сохранения рёбер и узлов. В случае множественного создания задач, используя данный функционал клиент отправлял запрос на сервер для сохранения каждой вершины и ребра.

При такой реализации зачастую происходила ошибка из-за того, что запрос на сохранения ребра поступал ранее, чем запрос на сохранения вершины, а так как обязательным параметром для создания ребра является целевая вершина, то создание такого ребра невозможно.

Для решения данной проблемы были созданы два массива для сохранения вершин и ребер. Отправления запроса от клиента на сервер происходит последовательно. В первую очередь отправляется запрос о сохранении всех вершин, которые присутствуют в массиве. После того как массив вершин станет пустым, клиент отправляет запрос о сохранении всех рёбер. Использование вышеописанной реализации позволило избежать возникновение ошибок, а также контролировать запросы клиента к серверу.

Для сохранения были использованы функции `save ()` и `saveLink()`. При создании большого количества задач, используя функцию «быстрое создание» созданные рёбра и вершины добавляются в массивы `linksQueue()` и `taskQueue()` соответственно. Сначала происходит рекурсивное сохранение всех элементы массива `taskQueue()`. На каждой итерации рекурсии реализовано удаление элемента из массива `taskQueue()` в случае получения сообщение об его успешном сохранении от сервера. После того как количество элементов массива `taskQueue()` станет равным нулю, вызывается функция `saveLink()`, которая в свою очередь отправляет запрос на сервер для сохранения всех рёбер. Программная реализация вышеописанных функций представлена на рисунке 5 и 6.


```

saveTasks () {
  this.recursiveSaving = true

  let t = this.taskQueue[0]
  if (!t) {
    this.recursiveSaving = false
    this.saveLinks()
    return
  }
  this.$store.dispatch('business/task/create', t)
  .then(value => {
    if (value) {
      let index = this.taskQueue.findIndex(q => q.id === t.id)
      this.taskQueue.splice(index, 1)
      if (this.taskQueue[0]) this.taskQueue[0].start_at = value.end_at
      eventBus.$emit('gantt_task_create', value)
    } else {
      eventBus.$emit('netplan_remove', t.id)
    }
    this.saveTasks()
  })
},

```

Рисунок 5 – Программная реализация сохранения множества вершин

```

saveLinks () {
  this.recursiveSaving = true

  let l = this.linkQueue[0]
  if (!l) {
    this.recursiveSaving = false
    return
  }
  this.$store.dispatch('business/task/createLink', l).then(value => {
    if (!value) {
      eventBus.$emit('netplan_remove', l.id)
    } else {
      let index = this.linkQueue.findIndex(q => q.id === l.id)
      this.linkQueue.splice(index, 1)
      let type = myCytoscape.getElementById(l.id).data().type
      if (type === 'dashed') myCytoscape.getElementById(l.id).data()
      myCytoscape.getElementById(l.id).data().weight =
        type === '' ? this.$store.getters['business/task/byId'](l.target).labor_all.toFixed(2) : ''
      myCytoscape.style().update()
      eventBus.$emit('gantt_link_add', l)
    }
    this.saveLinks()
  })
},

```

Рисунок 6 – Программная реализация сохранения множества рёбер

Так как для реализации данного функционала необходимо сохранять новую вершину с разницей в один рабочий день от текущей, было определено требование

учитывать не только выходные дни, а также праздники производственного календаря компании.

Для реализации данного требования был доработан серверный функционал. После получения данных о вершине сервер проверяет, чтобы время и окончания задачи находились в рабочем диапазоне. В случае если время начала или окончания выходят за рабочий диапазон, то сервер устанавливает время начала или окончания равным времени начала или окончания рабочего дня соответственно. Далее сервер устанавливает дату начала задачи на следующий рабочий день от переданной в соответствии с выходными днями и праздниками компании. После обработки сервер сохраняет задачу в базе данных и отправляет ответ клиенту.

5.3 Подготовка входных данных

Согласно функциональным требованиям, каждая задача, которая отображается на сетевом графике в качестве вершины, должна называться определенным образом (Рисунок 7).



Рисунок 7 – Отображение задачи на сетевом графике

Помимо отображения задач, необходимо визуализировать связи между ними. Связи могут быть двух типов: связь задач в одном разделе и связь задач в разных разделах. Также библиотека cytoscape.js требует определенной структуры входных данных. Для приведения входных данных к необходимому формату был написан javascript код (рисунок 8).

```

function getTasks (entity) {
  let type = getStore(entity.entity_type, true)
  if (entity.entity_type === 4) section = entity
  let parentName = getStore(entity.entity_type - 1)
  let children = store.business[type].list.filter(obj => obj[parentName + '_id'] === entity.id)
  children.forEach(e => {
    if (e.entity_type !== 6) {
      getTasks(e)
      return
    }
    if (e.entity_type === 6) {
      if (section && sections.find(s => s.id === section.id) === undefined) sections.push(section)
      let cardPos = store.business.card.list.find(c => c.id === e.card_id).position
      let nameAdd = '\n' + '5${cardPos}.${e.position ? e.position : 0}'
      tasks.push(
        { data: { id: e.id, name: section.code + nameAdd } }
      )
      links.forEach(l => {
        if (l.source === e.id || l.target === e.id) {
          if (edges.find(e => e.data.source === l.source && e.data.target === l.target) === undefined) {
            let sourceId = l.source ? l.source : 'start'
            let type = getLineType(l.source, l.target)
            edges.push(
              { data: { id: l.id,
                source: sourceId,
                target: l.target,
                weight: store.business.task.list.find(t => t.id === l.target).labor_all.toFixed(2),
                type: type } }
            )
          }
        }
      })
    }
  })
}

function getLineType (sourceId, targetId) {
  if (!store.business.task.list.find(t => t.id === sourceId) || !store.business.task.list.find(t => t.id === targetId)) return ''
  let sourceSection = store.business.card.list.find(c => c.id === store.business.task.list.find(t => t.id === sourceId).card_id).section_id
  let targetSection = store.business.card.list.find(c => c.id === store.business.task.list.find(t => t.id === targetId).card_id).section_id
  return sourceSection === targetSection ? '' : 'dashed'
}

```

Рисунок 8 – Реализация приведения данных к определенному формату

Приведенный выше код вызывается каждый раз при переходе пользователем на страницу разработанного модуля. В связи с этим пользователю необходимо дождаться пока клиент обработает все данные, приведет их к необходимому формату и отрисует сетевой график. В процессе обработки данных пользователь не может взаимодействовать с другими элементами веб сайта, т.к. клиент приложения «занят» необходимыми вычислениями. Для оптимизации такого варианта использования в процессе реализации был применен механизм Web Workers.

Web Workers - это механизм, который позволяет скрипту выполняться в фоновом потоке, который отделен от основного потока веб-приложения. Преимущество заключается в том, ресурсоёмкие вычисления могут выполняться в отдельном потоке, позволяя запустить основной (обычно пользовательский) поток без блокировки и замедления [19].

Web Worker-ы предоставляют простое средство для запуска скриптов в фоновом потоке. Поток Worker'a может выполнять задачи без вмешательства в пользовательский интерфейс [19].

В результате вышеописанный код был вынесен в отдельный файл `networkPlanning.js` в качестве Web worker-a. Такая реализация позволила «освободить» пользовательский интерфейс от нагрузки, а также улучшить опыт взаимодействия с интерфейсом реализованного продукта.

5.4 Работа с сервером

Так как архитектура разработанного приложения преимущественно клиент-серверная, то предполагаются частые запросы на сервер. В процессе тестирования была обнаружена следующая уязвимость: при большом количестве запросов на сохранение данных в момент времени сервер в связи с большой загрузкой перестаёт отвечать. Для разрешения данной проблемы было принято решение использовать очередь сообщений

Очередь сообщений – программный компонент разработки программного обеспечения, который используется для межпроцессорного или межпоточкового взаимодействия в одном и том же процессе. Для обмена сообщений используется очередь.

Очереди сообщений предоставляют асинхронный протокол передачи данных, означая, что отправитель и получатель сообщения не обязаны взаимодействовать с очередью сообщений одновременно. Размещённые в очереди сообщения хранятся до тех пор, пока получатель не получит их [20].

В качестве «брокера сообщений» было использовано популярное программное решение RabbitMQ.

В конечном итоге использование данного программного решение устранило проблему, связанную с нагрузкой сервера, а также привнесло следующие плюсы текущей реализации:

- Слабое связывание — независимость процессов друг от друга
- Избыточность — экономное использование ресурсов памяти.
- Масштабируемость — очереди сообщений позволяют распределить процессы обработки информации.

- Эластичность — очереди сообщений используются в качестве буфера для данных в условиях большой нагрузки, уменьшая нагрузку на систему обработки информации и не допуская ее отказа.
- Отказоустойчивость — в случае отказа системы, сообщения попадут на обработку, когда система восстановится.
- Гарантированная доставка — сообщение будет получено и обработано в любом случае.
- Анализ потоков данных — очереди сообщений позволяют выявлять узкие места в потоках данных приложения.

На рисунке 9 изображена диаграмма последовательности вызовов при создании задачи.

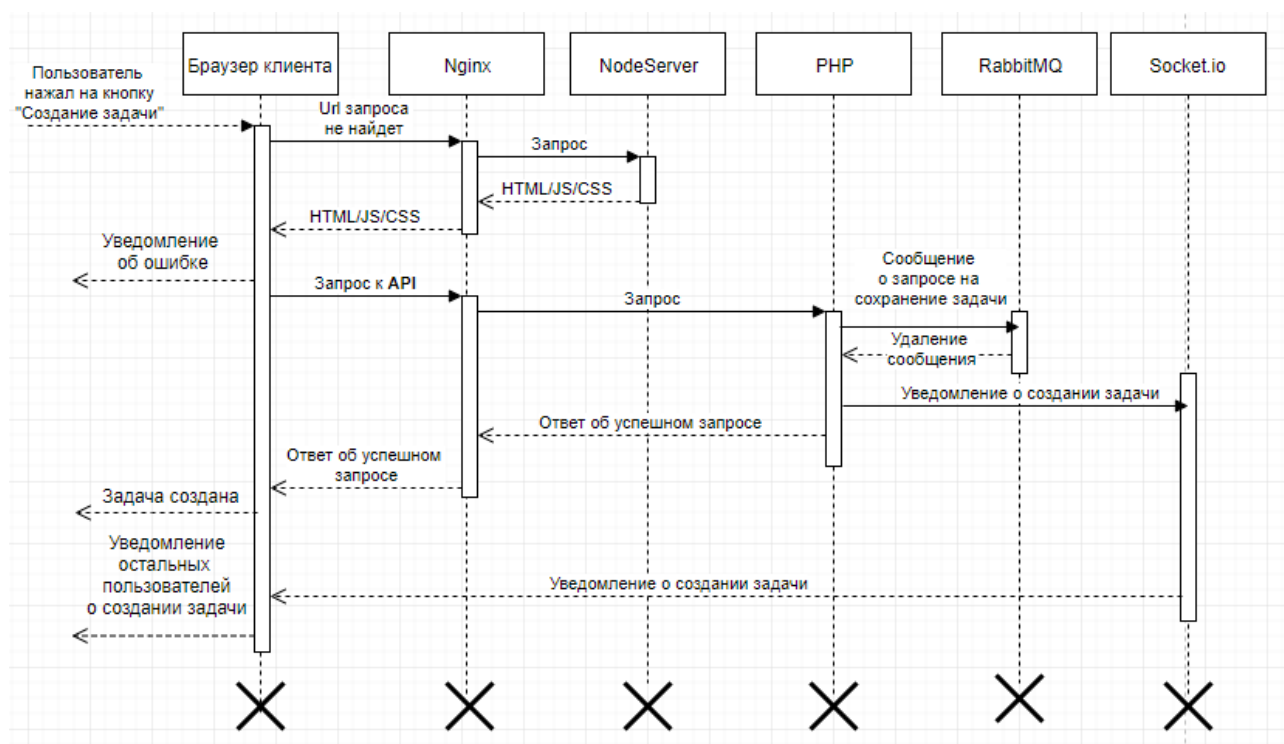


Рисунок 9 – Последовательность вызовов при создании задачи

После нажатия пользователем кнопки «Создать задачу», браузер исполняет JavaScript код выполняет запрос на сервер Nginx. В случае, когда URL запроса не найдет, Nginx отправляет запрос на сервер Node. Node сервер возвращает по цепочке файлы HTML, CSS, JS. В таком случае пользователь получает уведомление об ошибке. При успешной валидации URL, сервер Nginx проксирует запрос в PHP приложение, которое отправляет сообщение о необходимости сохранить задачу в очередь

RabbitMQ. Далее получатель сообщения на сервере PHP выполняет создание задачи, после которого отправляет данные в Socket.io сервер, который в свою очередь по предустановленному соединению отправляет его на веб клиент пользователя.

6. Результат

В результате была разработан инструмент администрирования проектных работ проектной компании. Данный инструмент обладает следующим функционалом:

- Создание, удаление, обновление узлов.
- Создание, удаление, обновление ребер.
- Фильтрация графика по разделам (Рисунок 14).
- Сворачивания последовательных ребер (Рисунок 13).
- Позиционирования графа относительно точки начала.
- Отмены последнего действия.
- Отображение критического пути и его анимации (Рисунок 11).
- Подсветка несоединенных ребер.
- Предупреждения о создании цикла задач.
- Отображение карточки задачи.
- Быстрое создание задачи от текущей.

На рисунке 10 изображен пользовательский интерфейс работы приложения. Контекстное меню вершины изображено на рисунке 12.

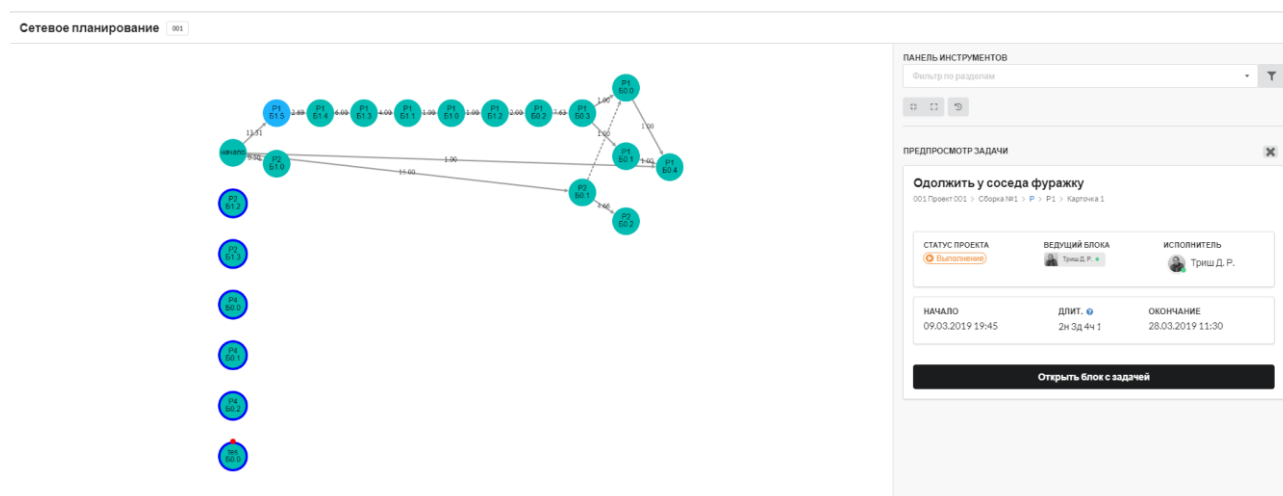


Рисунок 10 – Интерфейс работы приложения

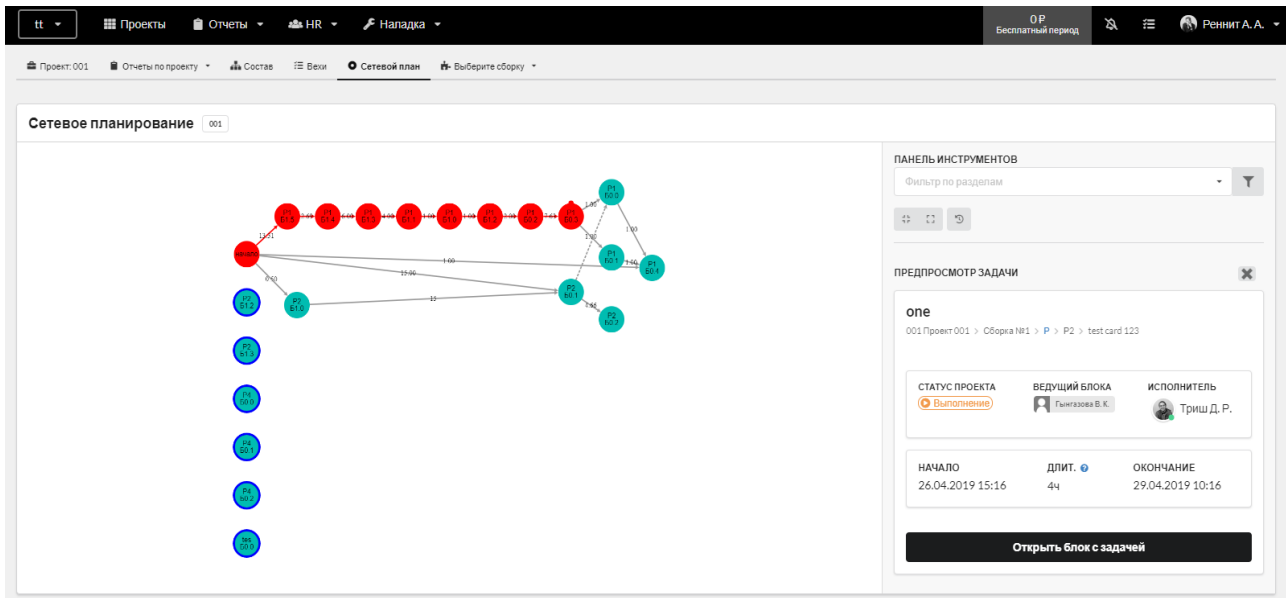


Рисунок 11 – Отображение критического пути проекта

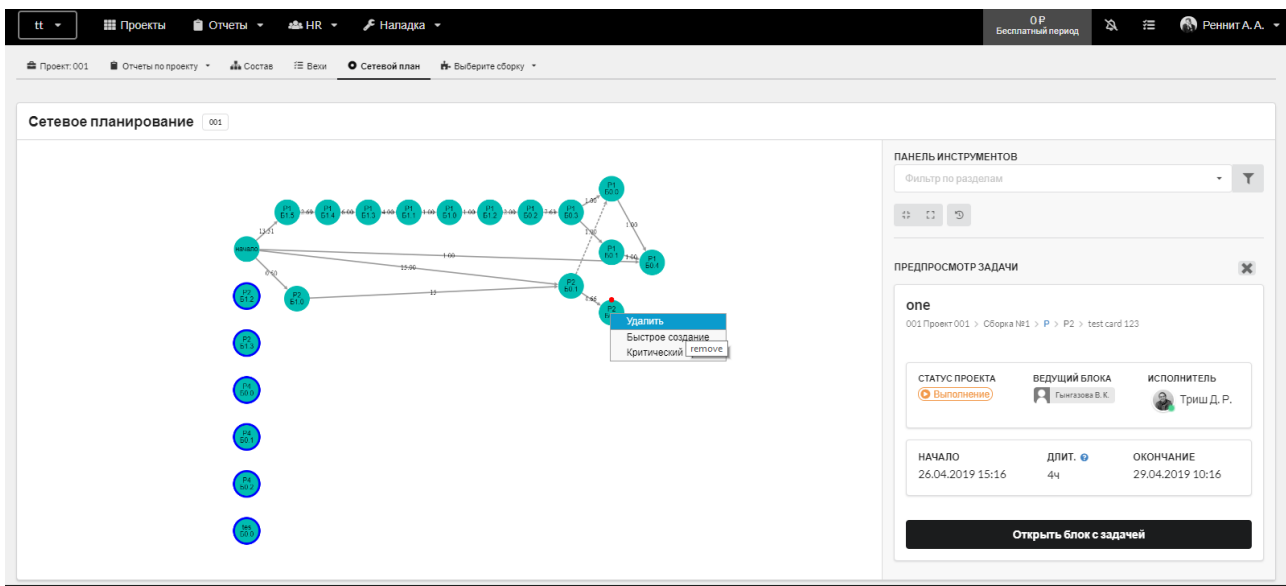


Рисунок 12 – Демонстрация контекстного меню узла

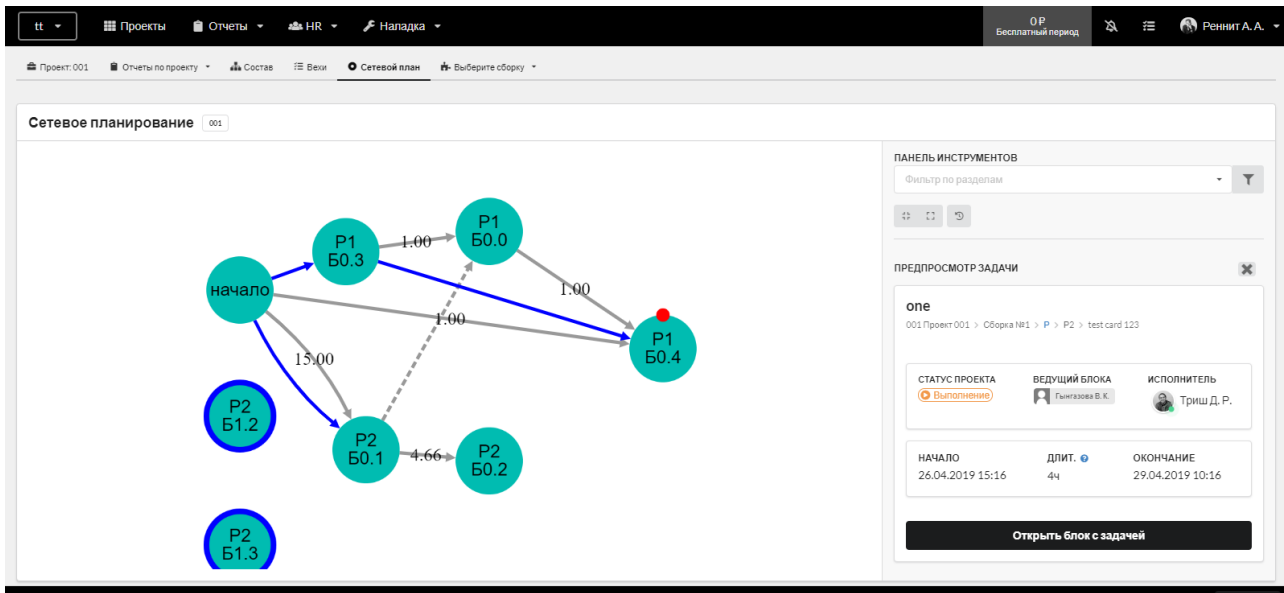


Рисунок 13 – Сворачивание последовательных ребер

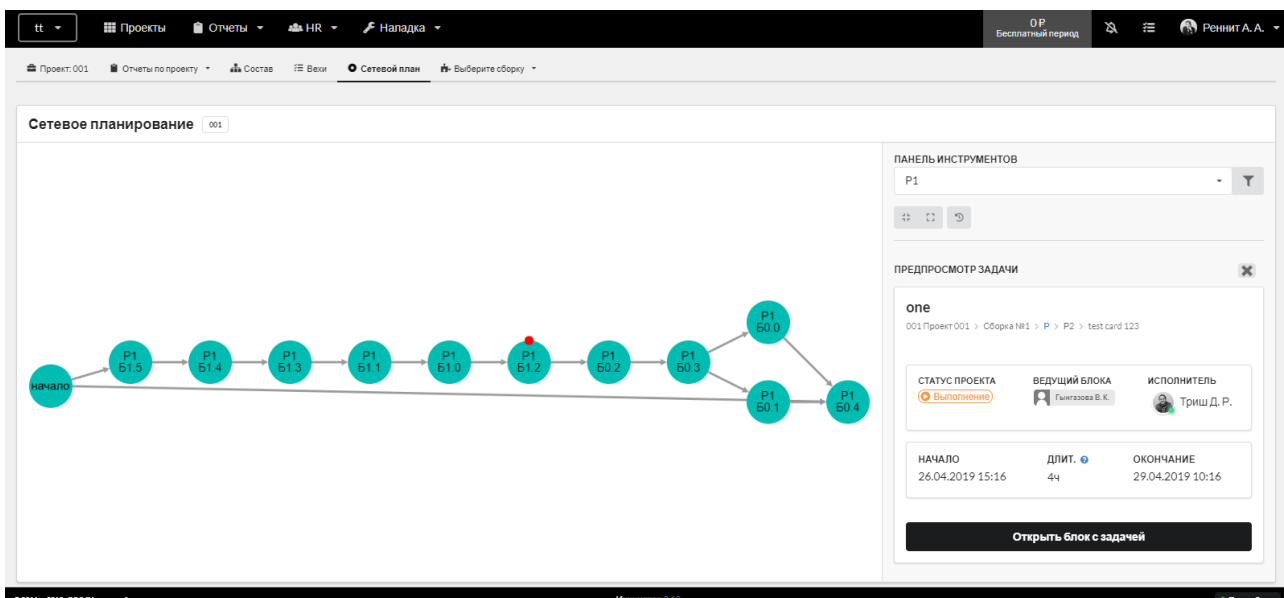


Рисунок 14 – Фильтрация задач по разделу проекта

7. Финансовый менеджмент

7.1 Предпроектный анализ

7.1.1 Потенциальные потребители результатов исследования

Модуль автоматизированного администрирования проектных задач предназначен для проектных организаций, которым необходимо оптимизировать проектные работы. Поэтому целевой аудиторией будут рассматриваться коммерческие и некоммерческие проектные организации.

Целевым рынком разработки является рынок систем для планирования проектных работ предприятия. Один из критериев сегментации – вид потребителей, основанный на целевой аудитории. Вторым по значимости может стать размер организации, так как размер предприятия имеет связь не только с финансовыми возможностями, но и с преследуемыми целями и уже существующей проблемой организации проектных работ. К примеру, организации малого масштаба имеют ограниченные финансовые возможности, но также обладают высокой заинтересованностью в результатах работы сервиса, так как ограниченный ресурс не позволяет им разработать свой программный продукт. Данный критерий основан на поведенческом принципе сегментирования, который предполагает разделение потребителей на группы в зависимости от характера использования программного продукта.

В таблице 1 представлена карта сегментирования рынка на основе наиболее значимых критериев.

Таблица 1 – Карта сегментирования рынка систем для анализа и управления сетевым трафиком

		Виды потребителей	
		Коммерческие организации	Некоммерческие организации
Размер компании	Крупные		
	Средние		
	Малые		

Как видно из карты сегментирования, перспективным направлением является ориентация на компании, прежде всего, средних и крупных масштабов, заинтересованных в автоматизированном администрировании проектных работ. Следует отметить, что рассматриваемый в данной работе проект отвечает критериям описанного сегмента рынка.

7.2 Анализ конкурентных технических решений

Анализ существующих конкурентных решений позволяет определить сильные и слабые стороны созданного продукта, обозначить направление его развития, а значит, не только оценить, но и повысить конкурентоспособность продукта на рынке.

Наиболее популярными аналогами представленной разработки являются такие сервисы как:

- « Graph_editor » (K1);
- «Graph Online» (K2);

Для того чтобы провести более детальный анализ сравнительной эффективности аналогов, была составлена оценочная карта для сравнения конкурентных разработок. Данная карта представлена в виде таблицы 2.

Таблица 2 – Оценочная карта для сравнения конкурентных программных решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Бф	Бк1	Бк2	Кф	Кк1	Кк2
Технические критерии оценки ресурсоэффективности							
Функциональные возможности	0,2	5	2	2	1	0,4	0,4
Скорость работы	0,15	4	3	5	0,6	0,15	0,75
Удобство использования интерфейса	0,2	4	5	4	0,8	1	0,8
Потребность в ресурсах памяти	0,1	5	2	2	0,4	0,5	0,5
Экономические критерии оценки ресурсоэффективности							
Конкурентоспособность продукта	0,1	5	3	4	0,5	0,3	0,4
Цена	0,15	5	4	2	0,75	0,6	0,3
Популярность на рынке	0,05	0	5	3	0	0,25	0,15
Поддержка продукта	0,05	5	1	5	0,25	0,05	0,25
Итого	1				4,3	3,25	3,55

Экспертная оценка основных технических и экономических характеристик конкурентных программных решений показывает, что разрабатываемый инструмент для администрирования проектных работ проектной компании является конкурентоспособной по сравнению с представленными аналогами.

Основными недостатками конкурентных программных продуктов являются функциональные возможности и удобство в эксплуатации.

Разрабатываемый программный продукт для планирования проектных работ предоставляет широкий спектр функциональных возможностей, имеет простой и удобный пользовательский интерфейс, а также надежен при использовании.

7.3 SWOT-анализ

SWOT-анализ позволяет выявить сильные и слабые стороны проекта, а также определить возможности и угрозы для реализации.

Выявленные достоинства, недостатки, а также возможности и угрозы представлены в виде матрицы SWOT, представленной в таблице 3.

Таблица 3 – Матрица SWOT-анализа

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Актуальность разработки.</p> <p>С2. Низкие требования к аппаратно-программному обеспечению.</p> <p>С3. Удобный интерфейс.</p> <p>С4. Обширные функциональные возможности.</p> <p>С5. Возможность последующей доработки.</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Для использования продукта необходим интернет.</p> <p>Сл2. Необходима постоянная поддержка продукта.</p> <p>Сл3. Необходимость интеграции в информационную систему</p> <p>Сл4. Длительная разработка.</p> <p>Сл5. Незнание программного продукта на целевом рынке.</p>
<p>Возможности:</p> <p>В1. Реализация новых</p>	<p>Направления развития:</p>	<p>Сдерживающие факторы:</p>

<p>функций.</p> <p>В2. Расширение списка используемых устройств.</p> <p>В3. Повышение скорости работы с большим количеством данных.</p> <p>В4. Рост потребностей клиентов.</p> <p>В5. Захват смежных сегментов целевого рынка.</p>	<ol style="list-style-type: none"> 1. В1В5С1С3С5 – добавление функций для автоматизации процесса планирования. 2. В2В5С1С2С4 – увеличение количества поддерживаемых устройств. 3. В3В4С1С3 – внедрение механизмов для ускорения работы. 4. В3В4С1С2 – реклама и продвижение продукта на целевом рынке. 	<ol style="list-style-type: none"> 1. В2Сл4 – отсутствие кроссплатформенности уменьшает спрос на продукт. 2. В4В5Сл2Сл4 – отсутствие людей, имеющих опыт разработки на различных платформах. 3. В3В5Сл5 – отсутствие опыта продвижения программного продукта на рынке.
<p>Угрозы:</p> <p>У1. Появление и развитие аналогичных систем.</p> <p>У2. Непопулярность продукта на рынке.</p> <p>У3. Сбои работы в различных средах функционирования продукта.</p>	<p>Угрозы развития:</p> <ol style="list-style-type: none"> 1. У2У3С1С3С5 – непопулярность продукта на рынке снизит мотивацию разработчика к развитию проекта. 2. У1С1 – развитие конкурентных систем может привести к снижению спроса на продукт. 3. У1У3С3С5 – отсутствие необходимой функциональности может снизить интерес пользователей к развитию продукта и желание участвовать в его разработке. <p>Постоянная поддержка продукта, разработка и внедрение нового функционала</p>	<p>Уязвимости:</p> <ol style="list-style-type: none"> 1. У3Сл1Сл5 – сбои работы программы при первом появлении на рынке могут способствовать провалу проекта. 2. У3У4Сл1Сл2Сл4 – нестабильность работы всей системы в целом. 3. У3Сл1 – система может не уведомить пользователя о критической ошибке, что ведет к потере данных. <p>Написание функциональных тестов и внедрение системы тестирования</p>

Для того, чтобы разобраться с различными комбинациями взаимосвязей областей матрицы SWOT, были построены интерактивные матрицы проекта, показывающие соответствия параметров SWOT-анализа.

Интерактивная матрица проекта полей «Сильные стороны и возможности» представлена в таблице 4.

Таблица 4 – Интерактивная матрица сильных сторон и возможностей проекта

Сильные стороны проекта						
Возможности проекта		C1	C2	C3	C4	C5
	B1	+	+	+	+	-
	B2	+	+	0	-	-
	B3	+	0	-	+	-
	B4	+	+	+	-	+
	B5	+	+	+	+	-

Интерактивная матрица проекта полей «Слабые стороны и возможности» представлена в таблице 5.

Таблица 5 – Интерактивная матрица слабых сторон и возможностей проекта

Слабые стороны проекта						
Возможности проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	B1	+	-	+	-	0
	B2	-	0	-	+	-
	B3	+	-	+	-	+
	B4	-	+	-	+	-
	B5	-	+	0	+	+

Интерактивная матрица проекта полей «Сильные стороны и угрозы» представлена в таблице 6.

Таблица 6 – Интерактивная матрица сильных сторон и угроз проекта

Сильные стороны проекта						
Угрозы проекта		C1	C2	C3	C4	C5
	У1	+	0	0	0	-
	У2	-	+	+	+	+
	У3	-	+	+	+	+

Интерактивная матрица проекта полей «Слабые стороны и угрозы» представлена в таблице 7.

Таблица 7 – Интерактивная матрица слабых сторон и угроз проекта

Слабые стороны проекта						
		Сл1	Сл2	Сл3	Сл4	Сл5
Угрозы проекта	У1	0	-	0	-	0
	У2	-	0	-	-	0
	У3	+	+	-	-	+

Результаты анализа интерактивных матриц проекта представлены в матрице SWOT-анализа в таблице 3.

7.3.1 Оценка готовности проекта к коммерциализации

Разработанный инструмент сетевого планирования в перспективе нацелен на широкую аудиторию пользователей. Однако для того, чтобы вывести разработанный продукт, требуется провести оценку готовности проекта к коммерциализации. В таблице 12 представлен бланк степени готовности разработанного сервиса.

Таблица 12 – Бланк оценки степени готовности научного проекта к коммерциализации

№ п / п	Наименование	Степень проработанности научного проекта	Уровень имеющихся знаний у разработчика
1 .	Определен имеющийся научно-технический задел	5	5
2 .	Определены перспективные направления коммерциализации научно-технического задела	5	5
3 .	Определены отрасли и технологии (товары, услуги) для предложения на рынке	5	5
4 .	Определена товарная форма научно-технического задела для представления на рынок	5	5
5 .	Определены авторы и осуществлена охрана их прав	3	4
6 .	Проведена оценка стоимости интеллектуальной собственности	4	4
7 .	Проведены маркетинговые исследования рынков сбыта	4	4
8 .	Разработан бизнес-план коммерциализации научной разработки	2	3
9 .	Определены пути продвижения научной разработки на рынок	3	2
10 .	Разработана стратегия (форма) реализации научной разработки	5	5
11 .	Проработаны вопросы международного сотрудничества и выхода на зарубежный рынок	3	3
12 .	Проработаны вопросы использования услуг инфраструктуры поддержки, получения льгот	4	3
13 .	Проработаны вопросы финансирования коммерциализации научной разработки	3	3
14 .	Имеется команда для коммерциализации научной разработки	3	2
15 .	Проработан механизм реализации научного проекта	5	5
	ИТОГО БАЛЛОВ	59	58

Как видно из таблицы 12, степень проработанности проекта составила 59 баллов, а уровень имеющихся знаний у разработчика составляет 58 балла. Из этого следует, что разработанный сервис представляет собой достаточно перспективный для коммерциализации продукт, а степень знаний разработчика позволяет осуществить эту коммерциализацию. Безусловно, высокая степень готовности продукта к коммерциализации позволит получить большой объем инвестиций и направить его на развитие сервиса (т.е. увеличение его производительности, расширение функционала). Однако для более эффективного продвижения продукта потребуются помощь специалистов по маркетинговым исследованиям и продвижению продукта, финансированию коммерциализации и юридическим аспектам реализации проекта.

7.4 Коммерциализация результатов научно-технического исследования

Рассматриваемый в данной работе проект в перспективе предполагает объединение с другими проектами по смежной тематике для увеличения функциональных возможностей сервиса. Помимо этого, как было отмечено ранее, сервис может быть со временем усовершенствован согласно предпочтениям пользователей и будущему течению развития решаемой задачи. Таким образом, разработка представленного сервиса представляет собой долгосрочный проект.

Исходя из отмеченных ранее аспектов, наиболее приемлемым методом коммерциализации сервиса может стать организация совместного предприятия. Совместное предприятие позволит объединить команду разработчиков сервиса, а также проводить автономное совершенствование продукта. Безусловно, помимо команды разработчиков в такое предприятие следует привлечь специалистов по маркетингу и правовым вопросам организации деятельности.

Очевидное достоинство организации такого предприятия состоит в том, что все его ресурсы направлены на развитие одного продукта, таким образом увеличивается эффективность развития и продвижения разрабатываемого сервиса.

7.5 Инициация проекта

В данном разделе представлен устав магистерской работы. В таблице 8 показаны заинтересованные стороны и их ожидания.

Таблица 8 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
Проектные компании, ООО «СибТехПроект»	Стабильно работающий веб-сервис для автоматизации администрирования проектных работ
НИ ТПУ	Увеличение числа научных публикаций, дипломов на научно-практических конференциях

В таблице 9 рассмотрены цели, ожидаемые результаты проекта, а также критерии достижения целей.

Таблица 9 – Цели и результаты проекта

Цели проекта:	Разработать инструмент оптимизации проектных работ путем сетевого планирования. А также реализовать его в виде веб приложения
Ожидаемые результаты проекта:	Разработанный инструмент сетевого планирования с высокой конкурентоспособностью и степенью готовности к коммерциализации
Критерии приемки результата проекта:	Стабильная работа разработанного сервиса
Требования к результату проекта:	Требование:
	Работоспособность при использовании некорректных входных данных
	Одинаковая скорость загрузки при любом наборе данных
	Удобный пользовательский интерфейс
	Привлекательный для потенциального пользователя функционал
	Разработанный функционал полностью соответствует проектным решениям

7.5.1 Ограничения и допущения проекта

При разработке инструмента для планирования проектных работ необходимо учитывать несколько ограничений. Они представлены в таблице 10.

Таблица 10 – Ограничения проекта

Фактор	Ограничения / допущения
3.1 Бюджет проекта	179860 рублей
3.1.1 Источник финансирования	НИ ТПУ
3.2 Сроки проекта	11.02.2019 – 01.06.2019
3.2.1 Дата утверждения плана управления проектом	11.02.2019
3.2.2 Дата завершения проекта	01.06.2019
3.3 Прочие ограничения и допущения	Время работы участников проекта не может превышать 6 часа в день

7.5.2 Организационная структура проекта

Рабочая группа, роли участников, их функции и трудозатраты каждого участника научно-исследовательского проекта представлены в таблице 11.

Таблица 11 – Рабочая группа проекта

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функции	Трудо-затраты, час.
1	Фадеев Александр Сергеевич, доцент	Руководитель проекта	Проверка корректности выполнения проекта	37
2	Реннит Андрей Андреевич, магистрант	Исполнитель по проекту	Проектирование и разработка	480
ИТОГО:				517

7.6 Планирование управления научно-техническим проектом

7.6.1 План проекта

В рамках планирования научного проекта построены календарный и линейный графики проекта. Линейный график представлен в таблице 12.

Таблица 12 – Линейный график проекта в рабочих днях

Код работы (из ИСР)	Название	Длительность, дни	Дата начала работ	Дата окончания работ	Состав участников
1	Обзор существующего ПО	5	11.02.19	15.02.19	Инженер (дипломник)
2	Проектирование архитектуры	20	16.02.19	12.03.19	Научный руководитель, Инженер
3	Разработка методики и алгоритма	17	13.03.19	06.04.19	Научный руководитель, Инженер
4	Разработка программы	17	07.04.19	30.04.19	Инженер
5	Написание тестов	7	01.05.19	11.05.19	Инженер
6	Тестирование продукта	2	12.05.19	14.05.19	Инженер
7	Оценка эффективности полученных результатов	5	15.05.19	21.05.19	Инженер
8	Составление пояснительной документации	7	22.05.19	30.05.19	Инженер

Для иллюстрации календарного плана проекта на временном промежутке используется диаграмма Ганта (таблица 13).

Таблица 13 – Диаграмма Ганта

№ работ	Вид работ	Исполнители	T к _к кад.дн.	Продолжительность выполнения работ													
				февр.		март			апрель			май			июнь		
				2	3	1	2	3	1	2	3	1	2	3	1	2	
1	Обзор существующего ПО	И	5	■													
2	Проектирование архитектуры	Р, И	25	▨	■												
3	Разработка методики и алгоритма	Р, И	25			▨	■										
4	Разработка программы	И	24						■	■							
5	Написание тестов	И	11									■	■				
6	Тестирование продукта	И	3											■			
7	Оценка эффективности полученных результатов	Р, И	7												■	■	
8	Составление пояснительной документации	И	9														■

▨ – руководитель ■ – инженер

7.6.2 Бюджет научного исследования

В данном разделе рассматриваются вопросы планирования бюджета НТИД.

7.6.3 Специальное оборудование для научных (экспериментальных) работ

Для выполнения работы необходим персональный компьютер и программный софт «PhpStrom». В таблице 14 представлены материальные затраты НТИ.

Таблица 14 – Расчет затрат по статье «Спецоборудование для научных работ»

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, тыс.руб.	Общая стоимость оборудования, тыс.руб.
1.	Компьютер	1	34600	34600
2.	Среда разработки PhpStorm	1	12876	12876
	Итого			47476

7.6.4 Расчет основной заработной платы исполнителей проекта

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы оплаты труда. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы (размер определяется Положением об оплате труда).

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

$$C_{зп} = Z_{осн} + Z_{доп},$$

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

$Z_{доп}$ – дополнительная заработная плата;

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} * T_p,$$

где T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

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

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m * M}{F_d},$$

где F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (таблица 15).

Z_m – месячный должностной оклад работника, руб.;

M – количество месяцев работы в течение года без отпусков:

при отпуске в 48 рабочих дней $M=10,4$ месяца, 6 – дневная неделя;

Таблица 15 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней	52	52
- выходные дни	14	14
- праздничные дни		
Потери рабочего времени		
- отпуск	48	48
- невыходы по болезни		
Действительный годовой фонд рабочего времени	251	251

Месячный должностной оклад работника:

$$Z_m = Z_b * (1 + k_{пр} + k_d) * k_p,$$

где Z_b – базовый оклад, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3;

k_d – коэффициент доплат и надбавок составляет примерно 0,2;

k_p – районный коэффициент, равный 1,3 г. Томск.

Расчёт основной заработной платы приведён в таблице 16.

Таблица 16 – Расчет основной заработной платы

Исполнители	Z_b , руб.	$k_{пр}$	k_d	k_p	Z_m , руб.	$Z_{дн}$, руб.	T_p , раб. дн.	$Z_{осн}$, руб.
Руководитель	33664	0,3	0,2	1,3	65644,8	2720	6,2	16864
Инженер	12663	-	-	1,3	16461,9	682,1	80	54568

Таким образом, затраты на основную заработную плату составили 77838,7 рублей.

7.6.5 Дополнительная заработная плата научно-производственного персонала

В данную статью включается сумма выплат, предусмотренных законодательством о труде, например, оплата очередных и дополнительных отпусков; оплата времени, связанного с выполнением государственных и общественных обязанностей; выплата вознаграждения за выслугу лет и т.п. Дополнительная заработная плата рассчитывается исходя из 10% от основной заработной платы, работников, непосредственно участвующих в выполнении темы:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}},$$

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

$k_{\text{доп}}$ – коэффициент дополнительной зарплаты;

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

В таблице 17 приведена форма расчёта основной и дополнительной заработной платы.

Таблица 17 - Заработная плата исполнителей НИИ

Заработная плата	Руководитель	Инженер
Основная заработная плата	16864	54568
Дополнительная зарплата	1686,4	5456,8
Зарплата исполнителя	18550,4	60024,8
Итого по статье $C_{\text{зп}}$	78575,2	

7.6.6 Отчисления на социальные нужды

Статья включает в себя отчисления во внебюджетные фонды

$$C_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}),$$

где $k_{\text{внеб}}$ - коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.) для текущих вычислений равен 27,1%.

Тогда сумма отчислений во внебюджетные фонды равна, руб.:

$$C_{\text{внеб}} = 0,271 * 78575,2 = 21293,88,$$

7.7.7 Накладные расходы

В эту статью включаются затраты на управление и хозяйственное обслуживание, которые могут быть отнесены непосредственно на конкретную тему. Кроме того, сюда относятся расходы по содержанию, эксплуатации и ремонту оборудования, производственного инструмента и инвентаря, зданий, сооружений и др. Накладные расходы составляют 30% от суммы основной и дополнительной заработной платы, работников, непосредственно участвующих в выполнении темы.

$$C_{\text{накл}} = k_{\text{накл}} * (З_{\text{осн}} + З_{\text{доп}}),$$

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

Тогда:

$$C_{\text{накл}} = 0,3 * 78575,2 = 23572,56$$

7.7.8 Расчет затрат на электричество

В данном пункте рассчитываются затраты на электричество за время работы над проектом. Вычисление производится по следующей формуле:

$$C_{\text{эл.об}} = P_{\text{об}} * t_{\text{об}} * Ц_{\text{э}},$$

где $P_{\text{об}}$ - мощность, потребляемая оборудованием, кВт, равно приблизительно 700 Вт;

$t_{\text{об}}$ – время работы оборудования, час;

$Ц_{\text{э}}$ - тариф на 1 кВт*час.

Для ТПУ тариф одного кВт*час равен 5,8 рублей.

Тогда расчет потребленной электроэнергии за время работы над проектом выглядит следующим образом:

$$C_{\text{накл}} = 0,6 * 517 * 5,8 = 1799,16$$

7.7.9 Формирование бюджета затрат проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формиро-

вании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку продукции. Группировка затрат по статья представлена в таблице 18.

Таблица 18 – Группировка затрат по статьям

Статьи						
Расходы на электро-энергию	Специальное оборудование для научных (экспериментальных) работ	Основная за-работная плата	Допол-ните льная за-работная плата	Расходы на социальные нужды	Наклад-ные рас-ходы	Итого плано-вая се-бестои-мость
1799,16	47476	78575,2	7143,2	21293,88	23572,56	179860

7.8 Реестр рисков проекта

Риски проекта включают в себя различные неопределенные события, которые могут возникнуть в проекте и вызвать негативные последствия. Риски представлены в таблице 19.

Таблица 19 – Реестр рисков

Риск	Потенциальное воздействие	Вероя тность наступ ления (1-5)	Влияние риска (1-5)	Уровень риска *	Способы смягчения риска	Условия наступления
Несобл юдение сроков проекта	Проект не будет закончен во-время	2	4	Средний	Составле-ние плана реализации проекта	Неверное планирова-ние времени
Недо-статок знаний не поз-волит создать про-дукт, отвеча-ющий	Продукт ненадлежащего качества	2	5	Средний	Чтение Литера-туры по специаль-ности	Недостаток знаний и навыков у разработ-чика

	требо- ваниям						
	Созда- ние про- дукта, не соот- вет- ствующе- го ожида- ниям заказ- чика	Не востребованнос- ть системы	1	2	Низкий	Сбор тре- бований и тщательно е проекти- рование си- стемы	Изменение требований к системе

8 Социальная ответственность

Целью выпускной квалификационной работы является создание для программного продукта “Инициатор” прототипа модуля, который мог бы автоматически распределять проектные задачи на сотрудников проектной компании по определенным критериям. Для разработки продукта проводились различные теоретические исследования, анализ и структуризация полученных данных, а также проектирование и программирование системы с использованием вычислительной техники.

Так как выполнение работы заключалось в разработке системы для прогнозирования и оптимизации работы проектной организации, то в качестве рабочего места будет рассмотрено рабочее место оператора персональной электронной вычислительной машины (ПЭВМ).

Использование средств вычислительной техники, накладывает целый ряд вредных факторов на человека, что впоследствии снижает производительность его труда и может привести к существенным проблемам со здоровьем сотрудника.

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

Первичным этапом в задаче обеспечения безопасности труда является выявление и анализ вредных и опасных факторов труда оператора ПЭВМ, возможных причин потенциальных аварий и пожаров, производственных травм, профессиональных заболеваний. Следующими этапами в задаче обеспечения безопасности труда являются разработка мероприятий по защите вредных и опасных факторов, оценка условий труда и микроклимата рабочей среды.

Поэтому данный раздел посвящен анализу вредных и опасных факторов производственной среды для операторов ПЭВМ, разработке программ по минимизации воздействия вредоносного и опасного влияния выявленных факторов, а также программ по снижению вредных воздействий на окружающую среду, экономии невозобновимых ресурсов и защите в чрезвычайных ситуациях.

8.1 Производственная безопасность

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке или эксплуатации проекта.

Производственные условия на рабочем месте характеризуются наличием различных опасных и вредных производственных факторов, оказывающих негативное влияние на работников.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме [23].

Вредные факторы характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом, вредные факторы проявляются при определенных условиях таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения.

В таблице 1 представлены возможные вредные и опасные факторы, возникающие при работе за ПЭВМ.

Таблица 1 – Вредные и опасные факторы, возникающие при работе за ПЭВМ

Наименование видов работ	Факторы	Нормативные документы
Вредные факторы		
Работа за ПЭВМ	Отклонение показателей микроклимата (температуры и влажности воздуха)	СанПиН 2.2.2/2.4.1340-03 [2] СанПиН 2.2.4.548-96 [3]
	Недостаточная освещенность рабочей зоны	СанПиН 2.2.2/2.4.1340-03
Опасные факторы		
Работа за ПЭВМ	Опасность поражения электрическим током	ГОСТ 12.1.038–82 [4]
	Пожаровзрывоопасность	ФЗ от 22.07.2008 №123-ФЗ [5]

8.1.1 Вредные производственные факторы

8.1.1.1 Отклонение показателей микроклимата в помещении

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

К параметрам, характеризующим микроклимат в производственных помещениях, относятся:

- Температура воздуха (t, °C);
- Температура поверхностей (t, °C);
- Относительная влажность воздуха (φ, %);
- Скорость движения воздуха (v, м/с);

- Интенсивность теплового облучения (I , Вт/м²).

В производственных помещениях для работы с ПЭВМ происходит постоянное выделение тепла самой вычислительной техникой, вспомогательными приборами и средствами освещения. Поскольку оператор расположен в непосредственной близости с источниками выделения тепла, то данный фактор является одним из важнейших вредных факторов производственной среды оператора ПЭВМ, а высокая температура воздуха способствует быстрому перегреву организма и быстрой утомляемости [29].

Влажность оказывает большое влияние на терморегуляцию организма. Так, например, высокие показатели относительной влажности (более 85 %) затрудняют терморегуляцию снижая возможность испарения пота, низкие показатели влажности (менее 20 %) вызывают пересыхание слизистых оболочек человека [30].

Работа программиста относится к категории Ia, которые производятся сидя и сопровождаются незначительным физическим напряжением. Интенсивность энерготрат организма для данной категории работ составляет до 120 ккал/ч (до 139 Вт).

Оптимальные значения показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia представлены в таблице 2. [26]

Таблица 2 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	22 – 24	60 – 40	0,1
Теплый	Ia	21 – 23	60 – 40	0,1

Оптимальные микроклиматические условия обеспечивают общее и локальное ощущение теплового комфорта при минимальном напряжении механизмов терморегуляции, не вызывают отклонений в состоянии здоровья, создают

предпосылки для высокого уровня работоспособности и являются предпочтительными на рабочих местах.

В таблице 3 приведены допустимые величины показателей микроклимата на рабочих местах производственных помещений согласно СанПиН 2.2.4.548-96 для категории работ Ia. [26]

Таблица 3 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia	20 – 25	15 – 75	0,1
Теплый	Ia	21 – 28	15 – 75	0,1 – 0,2

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

Согласно требованиям, СанПиН 2.2.2/2.4.1340–03, в кабинете поддерживается температура равная 19–20 С°, при относительной влажности в 55–58%. Для этого в помещении проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ [25].

8.1.1.2 Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени.

Недостаточный уровень освещенности в помещении приводит к снижению остроты зрения, головным болям, снижению концентрации внимания и, как следствие, к ухудшению производительности труда.

Причиной недостаточной освещенности являются недостаточность естественного освещения, недостаточность искусственного освещения, пониженная контрастность.

Помещение, в котором производится разработка программного сервиса, представляет собой помещение, длина и ширина которого составляет 6м и 3м соответственно. Высота помещения равна 2,5 м, а высота рабочей поверхности 0,78 м.

По нормам, установленным СанПиН 2.2.2/2.4.1340-03, минимальная освещённость на рабочих местах, оборудованных ПЭВМ на поверхности стола в зоне размещения рабочего документа должна быть 300 лк. Коэффициент Z (отношение средней освещенности к минимальной) примем равным $Z = 1.1$. Коэффициент запаса определяется по таблице [12] в зависимости от запылённости помещения, в нашем случае $K = 1.5$ (коэффициент запаса). Коэффициент использования, выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы. Он зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризующих коэффициентами отражения от стен (R_C) и потолка (R_P), значение коэффициентов R_C и R_P определим по таблице из СНиП 23-05-95, $R_C = 50\%$, $R_P = 70\%$. Для освещения выбираем светильник типа ОД с $\lambda = 1,4$.

Так как высота помещения равна всего 2.5м, приняв высоту свеса равную $h_c = 0$ м, получаем $h = 2,5 - 0,78 = 1.72$ м;

$$L = 1.4 * 1.72 = 2.408 \text{ м}; L/3 = 0.803\text{м}.$$

Размещаем светильники в 3 ряда. В каждом ряду можно установить по одному светильнику типа ШОД мощностью 40 Вт и длиной 1.228 м. Учитывая, что в каждом светильнике установлено две лампы, то общее количество ламп в помещении составляет $N = 6$.

Находим индекс освещения.

$$i = 18/1.72 * (6 + 3) = 1.16.$$

Согласно коэффициентам использования светового потока светильников с люминесцентными лампами, коэффициент использования светового потока $\eta = 0.41$.

$$\Phi = \frac{300 \cdot 18 \cdot 0.803 \cdot 1.1}{6 \cdot 0.41} = 1939 \text{ Лм}$$

Световой поток выбранной лампы равен 2850 лм. Согласно вышеописанным расчетом проведем проверку выполнения условия:

$$-10\% \leq \frac{\Phi_{\text{л.станд}} - \Phi_{\text{л.расч}}}{\Phi_{\text{л.станд}}} 100\% \leq +20\%$$

Следовательно,

$$-10\% \leq 3,2\% \leq +20\%$$

Рассчитаем электрическую мощность осветительной установки

$$P = 6 \cdot 40 = 240 \text{ Вт}$$

В итоге, для общего равномерного освещения рабочего места для разработки площадью 18м^2 необходимо установить 3 светильников ШОД по 2 лампы в каждом, мощностью лампы 40 Вт. Общая мощность осветительной установки составила 240 Вт.

8.1.2 Опасные производственные факторы

8.1.2.1 Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания.

Опасность поражения человека электрическим током оценивается величиной тока I (А), проходящего через его тело, или напряжением прикосновения U (В). Степень опасного воздействия на человека электрического тока зависит от рода и величины напряжения тока, частоты электрического тока, пути тока через тело человека, продолжительности его воздействия на организм человека, а также условий внешней среды.

Электрический ток, протекая через тело человека, производит термическое, механическое и световое воздействие – электролитическое разложение жидкости (в том числе и крови), судорожное сокращение мышц, разрыв тканей и поражение глаз.

Работа с ПЭВМ является опасной с точки зрения поражения током, так как практически во всех частях компьютера течет электрический ток. Поражение электрическим током при работе в ПЭВМ возможно при наличии оголенных участков на кабеле, нарушении изоляции распределительных устройств и от токоведущих частей компьютера в случае их пробоя и нарушении изоляции, при работе с ПЭВМ во влажной одежде и влажными руками.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединения с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

К мероприятиям по предотвращению возможности поражения электрическим током относятся:

- При производстве монтажных работ необходимо использовать только исправный инструмент, аттестованный службой КИПиА;
- С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- При включенном сетевом напряжении работы на задней панели должны быть запрещены;
- Все работы по устранению неисправностей должен производить квалифицированный персонал;
- Необходимо постоянно следить за исправностью электропроводки [25].

Согласно ГОСТ 12.1.038-82 на рабочем месте программиста допускаются уровни напряжений прикосновения и токов, представленные в таблице 4. [27]

Таблица 2 – Предельно допустимые напряжения прикосновения и токи

Род тока	Напряжение прикосновения, В	Ток, мА
	не более	
Переменный, 50 Гц	2,0	0,3
Постоянный	8,0	1,0

Значения напряжения прикосновения и токов приведены при продолжительности воздействия не более 10 минут в сутки.

8.1.2.2 Пожарвзрывобезопасность

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями.

Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий по предупреждению пожаров и взрывов.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Для устранения возможных причин возникновения пожаров необходимо проводить следующие мероприятия:

- **Организационные мероприятия:**
 - противопожарный инструктаж обслуживающего персонала;
 - обучение персонала техники безопасности;
 - разработка инструкций, плакатов, планов эвакуации.
- **Эксплуатационные мероприятия:**
 - соблюдение эксплуатационных норм оборудования;

- выбор и использование современных автоматических средств тушения пожаров.
- Технические мероприятия:
 - профилактический осмотр и ремонт оборудования;
 - соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления, вентиляции и освещения.

8.2 Экологическая безопасность

8.2.1 Анализ воздействия продукта на окружающую среду

Разработанный программный продукт, не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде.

Объект, на котором производилась разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер санитарно-защитной зоны которых равен 50 м.

Основными факторами, оказывающими негативные действия на экологию, являются факторы, связанные с производством и эксплуатацией компьютерной техники. В частности, отходы и выбросы, имеющие место на этапе производства компьютеров, а также отходы, связанные с неполной их утилизацией.

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

- локальное повышение электромагнитного и радиоактивного фона;
- неоправданное потребление электроэнергии (связано с использованием компьютера не на полную мощность в течение всего его время работы) и прочее.

8.2.2 Решения по обеспечению экологической безопасности

При разработке любых автоматизированных систем возникает необходимость утилизировать производственные отходы, в качестве которых в данном

случае выступают люминесцентные лампы, бумажные отходы (макулатура), а также неисправные детали персональных компьютеров, плат, контроллеров.

По истечении срока службы люминесцентных ламп, лампу требуется сдать на утилизацию в специализированное учреждение, имеющие специальную лицензию на данный вид деятельности. Бумажные отходы должны передаваться в соответствующие организации для дальнейшей переработки во вторичные бумажные изделия. Неисправные комплектующие персональных компьютеров и должны передаваться специализированным организациям по утилизации, имеющим лицензию на дальнейшую их переработку.

8.3 Безопасность в чрезвычайных ситуациях

8.3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения

Возможными чрезвычайными ситуациями могут быть:

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

Общие правила поведения при чрезвычайных ситуациях:

1) Не паниковать и не поддаваться панике. Призывать окружающих к спокойствию.

2) По возможности немедленно позвонить по телефону «01», сообщить что случилось, указать точный адрес места происшествия, назвать свою фамилию и номер своего телефона.

3) Включить устройства передачи звука (радио, телевизор), а также прослушать информацию, передаваемую через уличные громкоговорители и громкоговорящие устройства. В речевом сообщении будут озвучены основные рекомендации и правила поведения.

4) Выполнять рекомендации специалистов (сотрудников полиции, медицинских работников, пожарных, спасателей).

5) Не создавать условия, которые препятствуют и затрудняют действия сотрудников полиции, медицинских работников, спасателей, пожарных.

Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар.

Причинами возникновения данного вида ЧС могут являться:

- возникновением короткого замыкания в электропроводке;
- возгоранием устройств ПЭВМ из-за неисправности аппаратуры;
- возгоранием устройств искусственного освещения;
- возгоранием мебели по причине нарушения правил пожарной безопасности, а также неправильного использования дополнительных бытовых электроприборов и электроустановок.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [30].

8.3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.

Пожарная безопасность подразумевает надлежащее состояние объекта с исключением возможности возникновения очага возгорания (пожара) и его распространения в пространстве. Обеспечение пожарной безопасности — приоритетная задача для любого предприятия. Создание системы защиты регламентировано законом и нормативными документами различных ведомств.

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару,

а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легко воспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- Принять по возможности меры по эвакуации людей и материальных ценностей;
- Отключить от сети закрепленное за ним электрооборудование;
- Приступить к тушению пожара имеющимися средствами пожаротушения;
- Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран внутреннего противопожарного водопровода. Они предназначены для тушения

начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [30].

8.4 Правовые и организационные вопросы обеспечения безопасности

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на персональном компьютере (ПК) и перерывов, регламентированных с учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Вид трудовой деятельности на персональном компьютере в рамках данной работы соответствует группе В – творческая работа в режиме диалога с ПК, категория трудовой деятельности – I (до 2 часов непосредственной работы на ПК).

При 8-часовой рабочей смене и работе на ПК, соответствующей описанным выше критериям необходимо через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва устраивать регламентированные перерывы продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1-3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук,

а также массажа. Комплексы упражнений целесообразно менять через 2-3 недели.

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях. [32]

8.4.1 Организационные мероприятия при компоновке рабочей зоны

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с ПЭВМ. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Оно должно удовлетворять следующим требованиям:

- обеспечивать возможность удобного выполнения работ;

- учитывать физическую тяжесть работ;
- учитывать размеры рабочей зоны и необходимость передвижения в ней работающего;
- учитывать технологические особенности процесса выполнения работ.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям СанПин 2.2.2/2.4.1340-03.

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

8.4.2 Специфика влияния продукта на рабочий процесс

Разрабатываемый в ходе выполнения магистерской диссертации программный продукт используется для планирования и прогнозирования работ проектной компании. Данный продукт не влияет на организацию рабочей зоны,

но работа с ним позволит реорганизовать работу специалистов проектного предприятия, что в свою очередь повлияет на организацию рабочего процесса. Использование разработанного веб модуля позволит получить прогноз общей продолжительности реализации всего проекта, а также оптимизировать планирование проектных работ и сократить их продолжительность.

Заключение

В результате выполнения магистерской диссертации был спроектирован и разработан информационный модуль для эффективного сетевого планирования работ проектной компании.

Разработанный модуль является клиент-серверным приложением, клиентская часть которого написана на языке программирования JavaScript, а серверная на языке программирования PHP.

Программа отвечает всем заданным требованиям. Разработанное приложение обладает простым и удобным интерфейсом. Данная программа имеет возможность работать с ранее созданными проектными задачами. Подготовка входных данных для их визуализации происходит в фоновом потоке благодаря использованию механизма WebWorker. В данном приложении реализован алгоритм поиска критического пути, а также его подсветка и анимация. Также разработанное приложение имеет возможность быстрого создания задач с заданными параметрами. Серверная часть разработанного продукта способна выдержать достаточно большое количество запросов на создание, удаление или редактирование данных, используя «брокер сообщений» RabbitMQ.

В процессе исследования был проведен анализ популярных алгоритмов сетевого планирования и прогнозирования, а также описан используемый язык программирования, шаблон проектирования и среда разработки.

Разработанное приложение помогает планировать проектные работы проектной организации. Также приложение помогает контролировать сроки работ и их зависимость между собой.

В рамках развития проекта в будущем планируется:

- оптимизация исходного кода для повышения производительности;
- расширение функциональных возможностей;
- реализация алгоритмов прогнозирования проектных работ;
- распределение задач по сотрудникам проектной компании на основе анализа их трудоспособности по заданным параметрам.

Conclusion

As a result of the master's thesis, an information module for effective network planning of the project company's works was designed. The developed module is a client-server application, the client part of which is written in the programming language JavaScript, and the server part in the PHP programming language.

The program meets all specified requirements. The developed application has a simple and intuitive interface. This program has the ability to work with previously created project tasks. The preparation of input data for their visualization takes place in the background stream through the use of the WebWorker mechanism. This application implements a critical path search algorithm, as well as its highlighting and animation. Also, the developed application has the ability to quickly create tasks with specified parameters. The server part of the developed product is able to withstand a fairly large number of requests for creating, deleting or editing data using RabbitMQ "message broker".

In the course of the study, an analysis of popular network planning and forecasting algorithms was carried out, and a programming language, a design pattern and an environment design were described.

The developed application helps to plan the design work of the project organization. The application helps to control the timing of works and their dependence among themselves.

In the framework of the project development in the future it is planned:

- optimization of the source code to increase productivity;
- enhanced functionality;
- implementation of design work forecasting algorithms;
- distribution of tasks according to the employees of the project company based on the analysis of their working capacity according to specified parameters.

Список источников

1. Ахьюджа, Х. Н. Сетевые методы управления в проектировании и производстве [Текст] / Х.Н. Ахьюджа; перевод с англ. Б.С. Луныкова, В.М. Симонина, под ред. В.В. Калашникова. - М.: Мир, 1979. - 638 с.; Перевод изд.: Construction performance control by networks / H.N. Ahuja (New York etc., 1979); Дата обращения: 16.05.2019.
2. Сетевое планирование [Электронный ресурс] Wikipedia. URL: https://ru.wikipedia.org/wiki/Сетевое_планирование, свободный. Яз. Рус. Дата обращения: 14.05.2019.
3. Method CPM (Critical Path Method) [Электронный ресурс] Management mania URL: <https://managementmania.com/en/critical-path-method-cpm>, свободный. Яз. Англ. Дата обращения: 24.05.2019.
4. Gantt Chart [Электронный ресурс] Management mania URL: <https://managementmania.com/en/gantt-chart>, свободный. Яз. Рус. Дата обращения: 12.05.2019.
5. PERT Method (Program Evaluation and Review Technique) [Электронный ресурс] Management mania URL: <https://managementmania.com/en/pert-method>, свободный. Яз. Англ. Дата обращения: 23.05.2019.
6. Метод GERT [Электронный ресурс] Topknowledge URL: <http://topknowledge.ru/investmen/3187-metod-gert.html>, свободный. Яз. Англ. Дата обращения: 20.05.2019.
7. Моделирование методом Монте-Карло [Электронный ресурс] Sewiki URL: http://sewiki.ru/Моделирование_методом_Монте-Карло, свободный. Яз. Рус. Дата обращения: 10.05.2019.
8. Functional requirements [Электронный ресурс] / Tutorials point – simply easy learning. URL: https://www.tutorialspoint.com/software_testing_dictionary/functional_requirements.htm, свободный. Яз. Англ. Дата обращения: 06.05.2019.
9. Нотация и семантика языка UML [Электронный ресурс] / Национальный Открытый Университет «ИНТУИТ». URL:

<http://www.intuit.ru/studies/courses/32/32/info>, свободный. Яз. Рус. Дата обращения: 11.05.2019.

10. UML Deployment Diagrams [Электронный ресурс] / Uml-diagrams URL: <https://www.uml-diagrams.org/deployment-diagrams.html>, свободный. Яз. Англ. Дата обращения: 09.05.2019.

11. Программно-аппаратный комплекс [Электронный ресурс] / Wikipedia URL: https://ru.wikipedia.org/wiki/Программно-аппаратный_комплекс, свободный. Яз. Рус. Дата обращения: 17.05.2019.

12. Что такое PHP? [Электронный ресурс] / PHP Manual. URL: <http://php.net/manual/ru/intro-what-is.php>, свободный. Яз. Рус. Дата обращения: 26.05.2019.

13. What is Symfony [Электронный ресурс] / Symfony. URL: <https://symfony.com/what-is-symfony>, свободный. Яз. Англ. Дата обращения: 25.05.2019.

14. MVC [Электронный ресурс] / Web Creator. URL: <https://web-creator.ru/articles/mvc>, свободный. Яз. Рус. Дата обращения: 26.05.2019.

15. JavaScript [Электронный ресурс] / Wikipedia. URL: <https://ru.wikipedia.org/wiki/JavaScript>, свободный. Яз. Рус. Дата обращения: 25.05.2019.

16. Nuxt.js [Электронный ресурс] / Nuxt.js - The Vue.js Framework. URL: <https://nuxtjs.org>, свободный. Яз. Англ. Дата обращения: 24.05.2019.

17. Vue.js [Электронный ресурс] / Wikipedia. URL: <https://en.wikipedia.org/wiki/Vue.js>, свободный. Яз. Англ. Дата обращения: 24.05.2019

18. Bellman Ford Algorithm (Simple Implementation) [Электронный ресурс] / Geeksforgeeks URL: <https://www.geeksforgeeks.org/bellman-ford-algorithm-simple-implementation/>, свободный. Яз. Англ. Дата обращения: 24.05.2019.

19. Использование Web Workers [Электронный ресурс] / Mozilla URL: https://developer.mozilla.org/ru/docs/DOM/Using_web_workers, свободный. Яз. Рус. Дата обращения: 22.05.2019.

20. Очередь сообщений [Электронный ресурс] / Amazon URL: <https://aws.amazon.com/ru/message-queue>, свободный. Яз. Рус. Дата обращения: 20.05.2019.

21. Охрана труда [Электронный ресурс] / Безопасность жизнедеятельности. URL: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohrana-truda.html>, свободный. Яз. Рус. Дата обращения: 13.05.2019.

22. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Яз. Рус. Дата обращения: 13.05.2019.

23. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений. Яз. Рус. Дата обращения: 13.05.2019.

24. ГОСТ 12.1.038–82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. Яз. Рус. Дата обращения: 13.05.2019.

25. Федеральный закон от 22.07.2008 N 123-ФЗ (ред. От 13.07.2015) «Технический регламент о требованиях пожарной безопасности» [Электронный ресурс] /КонсультантПлюс. URL: http://www.consultant.ru/document/cons_doc_LAW_78699/, свободный. Яз. Рус. Дата обращения: 13.05.2019.

26. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп. – М. : Издательство «Альфа-Пресс», 2008. Яз. Рус. Дата обращения: 13.05.2019.

27. Назаренко О. Б. Безопасность жизнедеятельности: учебное пособие / О. Б. Назаренко, Ю. А. Амелькович; Томский политехнический университет. – 3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2013. Яз. Рус. Дата обращения: 13.05.2019.

28. ГОСТ Р 55090-2012 Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги. Яз. Рус. Дата обращения: 13.05.2019.

29. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности [Электронный ресурс] /

Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/1200032102>, свободный. Яз. Рус. Дата обращения: 13.05.2019.

30. ППБ 01–03. Правила пожарной безопасности в Российской Федерации. – М.: Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, 2003.

31. Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 03.07.2016) (с изм. и доп., вступ. в силу с 01.01.2019) [Электронный ресурс] / Консультант Плюс. URL: https://www.consultant.ru/document/cons_doc_LAW_34683/, свободный. Яз. Рус. Дата обращения: 13.05.2019.

32. СНиП 23-05-95. «Естественное и искусственное освещение». Яз. Рус. Дата обращения: 12.05.2019.

ПРИЛОЖЕНИЕ А

Таблица сравнения программных продуктов для сетевого планирования

Программный продукт	Тип распространения	Простота использования	Взаимодействие с базой данных	Возможность пользоваться online	Импорт данных	Алгоритмы теории графов
“Ms Project”	Платная	нет	да	да	да	частично
“Graph Editor”	Бесплатная	да	нет	да	нет	нет
“Omniplan”	Платная	нет	да	нет	да	частично
“Graph Online”	Бесплатная	да	нет	да	нет	да

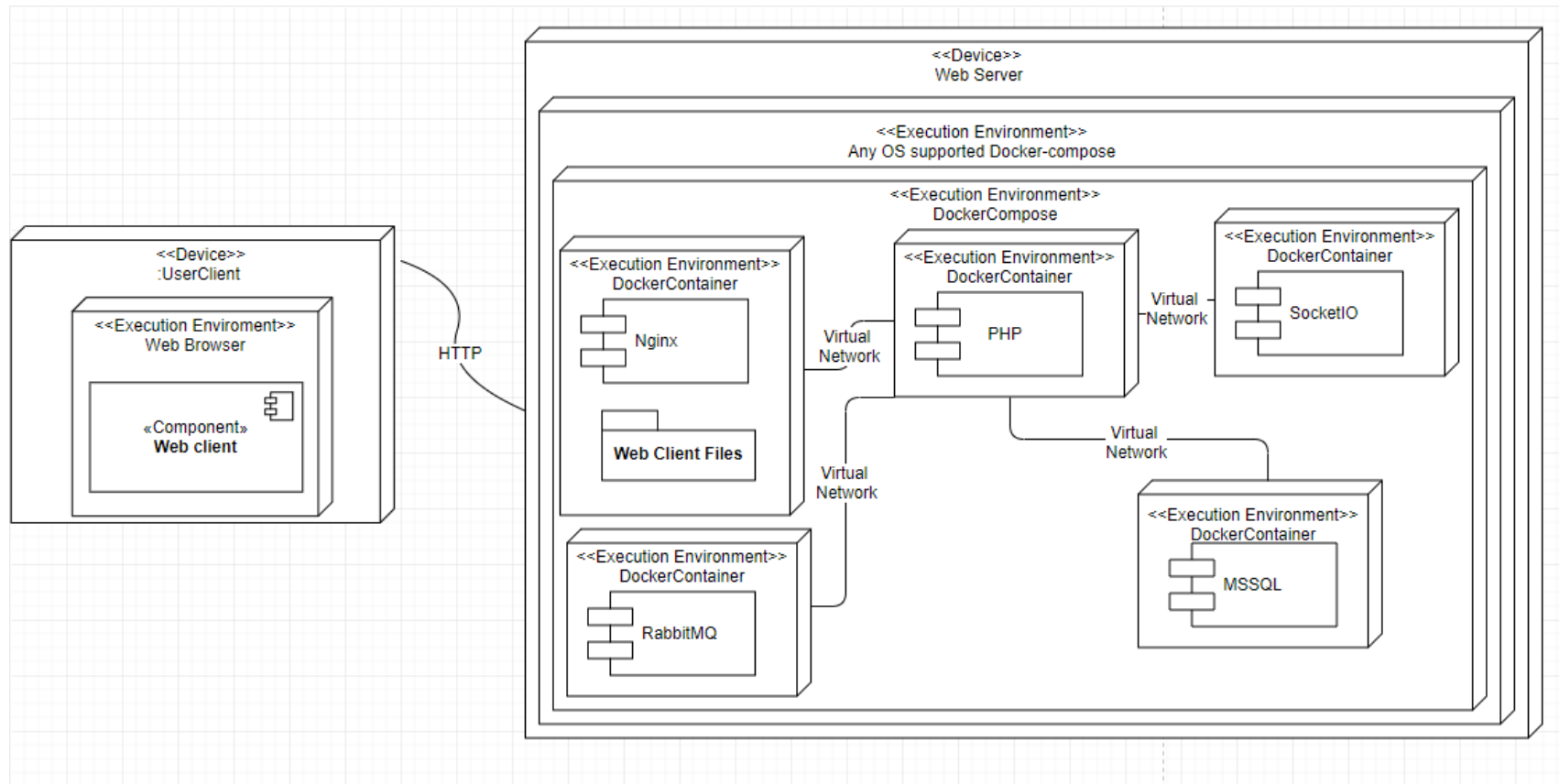
ПРИЛОЖЕНИЕ Б

Диаграмма вариантов использования



ПРИЛОЖЕНИЕ В

Диаграмма развертывания



Приложение Г

Part 4

Software implementation tools

Студент:

Группа	ФИО	Подпись	Дата
8BM71	Реннит Андрей Андреевич		

Консультант школы отделения (НОЦ) ИШИТР:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Фадеев Александр Сергеевич	К.Т.Н.		

Консультант – лингвист отделения (НОЦ) ШБИП:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Куркан Наталия Владимировна	-		

1. Software implementation tools

In accordance to the architecture of the developed application, it is necessary to develop both the client part and the server part.

1.1 Backend application part

The back end refers to parts of a computer application or a program's code that allows to operate and cannot be accessed by a user. Most data and operating syntax are stored and accessed in the back end of a computer system. Typically, the code is comprised of one or more programming languages. The back end is also called the data access layer of software or hardware and includes any functionality that needs to be accessed and navigated to by digital means [11].

1.1.1 Programming language

For the server-side development, the PHP scripting language, which is intensively used for developing web applications was chosen.

PHP is a server-side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

A script is a set of programming instructions that is interpreted at runtime. A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments.

The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application. Server-side scripts are interpreted on the server while client-side scripts are interpreted by the client application.

PHP is a server-side script that is interpreted on the server while JavaScript is an example of a client-side script that is interpreted by the client browser. Both PHP and JavaScript can be embedded into HTML pages [12]. PHP has the following advantages:

- PHP is open source and free.
- Short learning curve compared to other languages such as JSP, ASP etc.
- Large community document.
- Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost-effective choice.

- PHP is regular updated to keep abreast with the latest technology trends.
- Other benefit that you get with PHP is that it's a server-side scripting language; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.
- PHP has in built support for working hand in hand with MySQL. Also, PHP works with another database management systems (Postgres, Oracle, MS SQL Server, ODBC etc.).
- PHP is a cross platform. It means that you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

1.1.2 Symfony

Symfony was chosen as a framework for development. It is a leading PHP framework for sites and web applications creation.

Symfony is a PHP web application framework and a set of reusable PHP components/libraries. Symfony was published as free software on October 18, 2005 and released under the MIT license [13].

The main advantages of symfony are:

- High flexibility.
- Simple testing.
- Long-term support and compatibility.
- Structed and clear documentation.

1.1.3 Development environment

To create a software product IDE (integrated development environment) JetBrains PhpStorm was used. PhpStorm is a commercial, cross-platform IDE for PHP built on JetBrains' IntelliJ IDEA platform.

PhpStorm provides an editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactorings for PHP and JavaScript code. PhpStorm's code completion supports PHP 5.3 - 7.1, and 7.2 (modern and legacy projects), including generators, coroutines, the finally keyword, list in foreach, namespaces, closures, traits and short array syntax. It includes a full-fledged SQL editor with editable query results.

PhpStorm is built on IntelliJ IDEA, which is written in Java. Users can extend the IDE by installing plugins created for the IntelliJ Platform or write their own plugins.

1.1.4 Design Pattern

To design an application pattern MVC (Model-View-Control) was used.

Model–View–Controller (usually known as MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the way's information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for code reuse and parallel development.

Traditionally used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications. Popular programming languages like JavaScript, Python, Ruby, PHP, Java, and C# have MVC frameworks that are used in web application development straight out of the box [14].

The main components of this design pattern are:

- Model. The central component of the pattern. It is the application's dynamic data structure, independent of the user interface. It directly manages the data, logic and rules of the application.
- View. Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- Controller. Accepts input and converts it to commands for the model or view.

Figure 3 shows the relationship between the components model, view, and controller.

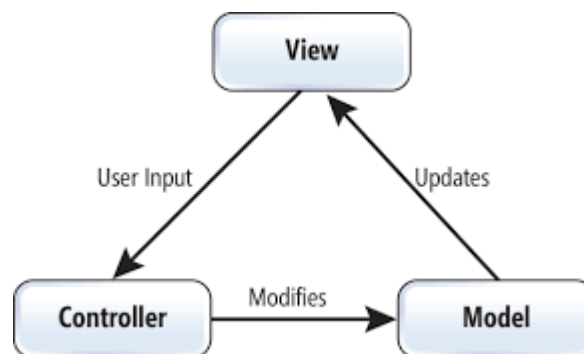


Figure 3 – Interconnection of MVC Template Components

In addition to dividing the application into these components, the model–view–controller design defines the interactions between them:

- The model is responsible for managing the data of the application. It receives user input from the controller.
- The view means presentation of the model in a particular format.
- The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

As with other software patterns, MVC expresses the "core of the solution" to a problem while allowing it to be adapted for each system.

1.2 Front end application part.

1.2.1 Programming language

The client part of the software product was developed in the JavaScript programming language.

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server [15].

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser. The main advantages of using JavaScript are:

- Universal support. All popular modern Web browsers support JavaScript with built-in interpreters.

– Imperative and structured. JavaScript supports much of the structured programming syntax from C programming language. ECMAScript 2015 added keywords `let` and `const` for block scoping, meaning JavaScript now has both function and block scoping. Like C, JavaScript makes a distinction between expressions and statements.

– Object-oriented. JavaScript is almost entirely object-based. In JavaScript, an object is an associative array, augmented with a prototype. A property may be added, rebound, or deleted at run-time. Most properties of an object (and any property that belongs to an object's prototype inheritance chain) can be enumerated using a loop.

1.2.2 Nuxt.js и Vue.js

Nuxt.js and Vue.js were used as a framework for writing the client part of the project administration module.

Nuxt.js is a free and open source web application framework based on Vue.js, Node.js, Express.js, Webpack and Babel.js. The framework is advertised as "meta-framework for universal applications". The term universal is used here with the meaning that the goal of the framework is to enable users to create web views in JavaScript utilizing familiar Vue.js single file component system, that can function both as in-browser single page application (SPA) views, as well as server-rendered web views which are then (after server rendering) "rehydrated" to full SPA functionality. Additionally, the framework enables users to have the content, or parts of it, fully pre-rendered on the server and served in the manner of static site generators [16].

Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications [17].

1.3 User interface implementing

To visualize an automated project task of the administration module, it was necessary to select the most appropriate JavaScript library. As part of the work on the master's thesis, the following decisions were considered.

1.3.1 Sigma.js

Sigma is a JavaScript library dedicated to graph drawing. It makes easy to publish networks on Web pages, and allows developers to integrate network exploration in rich Web applications.

This library provides the following functions:

- The default configuration of sigma deals with mouse and touch support.
 - Refreshing and rescaling when the container's size changes.
 - Rendering on WebGL and Canvas.
 - Recentering the graph and adapting the nodes and edges sizes to the screen.
 - Sigma provides a lot of different settings to make it easy to customize how to draw and interact with networks.
- The public API makes it possible to modify the data, move the camera, refresh the rendering, listen to events.

1.3.2 Arbor.js

Arbor is a graph visualization library built with web workers and jQuery. Rather than trying to be an all-encompassing framework, arbor provides an efficient, force-directed layout algorithm plus abstractions for graph organization and screen refresh handling.

- Canvas, SVG, or even positioned HTML elements usage
- Edges and nodes customization.
- Open source.

1.3.3 Dracula.js

Dracula.js is a set of tools to display and layout interactive connected graphs and networks, along with various related algorithms from the field of graph theory.

- Canvas and SVG usage.
- Open source.
- Edges and nodes customization.

1.3.4 D3.JS

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. This library has extensive functionality, but only the following functions can be used to develop a design planning tool:

- SVG usage.
- Open source.

- Huge number of functions.

1.3.5 Cytoscape.js

Cytoscape.js is an open-source graph theory (a.k.a. network) library written in JS. You can use Cytoscape.js for graph analysis and visualization. The library contains many useful functions in graph theory, such as:

- Has a large suite of tests that can be run in the browser or the terminal.
- Supports selectors for terse filtering and graph querying.
- Uses stylesheets to separate presentation from data in a rendering agnostic manner.
- Abstracted and unified touch events on top of a familiar event model.
- Built-in support for standard gestures on both desktop and touch.
- Supports set theory operations
- Includes graph theory algorithms, from BFS to PageRank.
- Nodes and edges customization.

1.4 Conclusion

According to the above mentioned, most of the existing libraries are intended only for visualizing graphs, and not for interacting with them. The main problem is that most of the functionality must be developed independently. Therefore, based on the analysis of existing JavaScript libraries, Cytoscape.js was selected. Since this library has already implemented mechanisms for creating, positioning elements, as well as their animation. In addition, the selected library has implemented graph theory algorithms, which simplifies the task of implementing a software product.