

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

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.04.04 Программная инженерия
 ООП/ОПОП Разработка программно-информационных систем
 Отделение школы (НОЦ) Информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема работы
Разработка веб-приложения для управления личными и командными ресурсами

УДК 004.774:005.96

Обучающийся

Группа	ФИО	Подпись	Дата
8K93	Садиков Максим Андреевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова Вероника Валерьевна	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Гасанов Магеррам Али оглы	Д.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	К.Т.Н.		

Томск – 2023 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
ОПК(У)-1	Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК(У)-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК(У)-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК(У)-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

ОПК(У)-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой
ОПК(У)-8	Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий.
ПК(У)-1	Способен выполнять интеграцию программных модулей и компонент
ПК(У)-2	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения
ПК(У)-3	Способен создавать техническую документацию на продукцию в сфере информационных технологий, управлять технической информацией
ПК(У)-4	Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных
ПК(У)-5	Способен проводить, оценивать и следить за выполнением концептуального, функционального и логического проектирования систем малого и среднего масштаба и сложности

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

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 09.03.04 «Программная инженерия»
 ООП/ОПОП – Разработка программно-информационных систем
 Отделение школы (НОЦ) – Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП/ОПОП
 _____ Чердынцев Е.С.
 (Подпись) (Дата) (ФИО)

ЗАДАНИЕ **на выполнение выпускной квалификационной работы**

Обучающийся:

Группа	ФИО
8K93	Садиков Максим Андреевич

Тема работы:

Разработка веб-приложения для управления личными и командными ресурсами	
Утверждена приказом директора (дата, номер)	от 30.01.2023г. №30-96/с

Срок сдачи обучающимся выполненной работы:	14.06.2023 г.
--	---------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом проектирования в выпускной квалификационной работе является веб-приложения для управления личными и командными ресурсами
Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке	1. Анализ предметной области 2. Обзор архитектуры приложения 3. Разработка функционала по модернизации приложения 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение. 5. Социальная ответственность
Перечень графического материала	Презентация в формате *.pptx

Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна
Названия разделов, которые должны быть написаны на иностранном языке:	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	27.02.2023 г.
---	---------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова Вероника Валерьевна	к.т.н.		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8K93	Садиков Максим Андреевич		

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

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 09.03.04 «Программная инженерия»
 ООП/ОПОП – Разработка программно-информационных систем
 Отделение школы (НОЦ) – Отделение информационных технологий

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающийся:

Группа	ФИО
8K93	Садиков Максим Андреевич

Тема работы:

Разработка веб-приложения для управления личными и командными ресурсами

Срок сдачи обучающимся выполненной работы:	14.06.2023 г.
--	---------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	1. Анализ предметной области	20
	2. Обзор архитектуры приложения	25
	3. Разработка функционала по модернизации приложения	25
	4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.	15
	5. Социальная ответственность	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Соколова Вероника Валерьевна	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП/ОПОП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н.		

Обучающийся

Группа	ФИО	Подпись	Дата
8K93	Садиков Максим Андреевич		

РЕФЕРАТ

Выпускная квалификационная работа содержит 91 страницу (без учёта приложений), 25 рисунков, 17 таблиц, 2 приложения и 23 литературных источника.

Ключевые слова: система планирования, система управления проектами, planuway, spa, javascript, typescript, react, angularjs.

Объектом исследования является разработка веб-приложения для планирования и управления проектами.

Целью выпускной квалификационной работы является разработка новых функций для приложения «Planuway», а именно: создание функции копирования карточки как записи времени, разработка меню коллекций, модернизация функции системных сообщений, а также усовершенствование функции создания зависимых карточек.

В перечень задач входит анализ предметной области, проведение обзора архитектуры приложения, а также разработка функционала, модернизирующего систему.

В результате выполнения работы был проанализирован рынок систем управления проектами, рассмотрены конкурентные преимущества системы. Было внедрено несколько новых функций, которые улучшают пользовательский интерфейс, а также приносят новые пользовательские возможности для более эффективной работы с приложением.

Основные характеристики: язык программирования – JavaScript/TypeScript, фреймворк для разработки клиентской части – «AngularJS» и «React», библиотека графических компонентов – «MUI».

Степень внедрения: приложение развернуто на серверах Microsoft.

Область применения: рынок систем управления проектами в IT.

В первой главе рассматривается анализ предметной области на основе исследований рынка систем управления проектами.

Во второй главе рассмотрена архитектура клиентской части приложения.

В третьей главе приводится описание модернизированных функциональностей приложения.

Пятая глава анализирует результаты разработки, учитывая финансовый менеджмент, ресурсоэффективность и ресурсосбережение. Оценивается эффективность решения и производится расчет финансовых и временных затрат.

Шестая глава посвящена рассмотрению правовых и организационных вопросов по обеспечению безопасности на рабочем месте при разработке программного решения. Также оцениваются аспекты экологической безопасности производства и рассматриваются наиболее типичные чрезвычайные ситуации.

СОДЕРЖАНИЕ

Реферат	7
Введение.....	11
Определения, обозначения, сокращения.....	13
Обзор литературы	15
Глава 1. Исследование предметной области	16
1.1. Описание концепции приложения для управления проектами	16
1.2. Описание процессов управления проектами	17
1.2.1. Анализ процесса управления проектами	17
1.2.2. Диаграмма процесса A1 «Планирование и управление проектом» в нотации EPC	19
1.2.3. Диаграмма процесса A2 «Выполнение задачи»	20
1.3. Требования к системе планирования задач и управления проектами.....	21
1.4. Анализ конкурентных решений.....	23
1.5. Сравнение с конкурентными решениями	23
Глава 2. Обзор архитектуры приложения	26
2.1. Роли и функциональные возможности пользователей приложения	26
2.1.1. Функциональные возможности авторизованного пользователя приложения.....	27
2.2. Создание макетов приложения	29
2.3. Описание программно-технических средств реализации клиентской части приложения	29
2.4. Архитектура клиентской части приложения	31
Вывод по главе	34
Глава 3. Разработка модулей для модернизации системы	35
3.1. Разработка модуля приложения интеграции с Google календарём	35
3.2. Модернизация меню коллекций	38
3.3. Модернизация функционала системных оповещений.....	43
3.3. Реализация копирования записи времени из карточки.....	44
3.4. Создание зависимых карточек из компактного диалога карточки и редактирование зависимостей.....	46
Вывод по главе	46
Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	49
4.1. Введение.....	49
4.2. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	49
4.2.1. Потенциальные потребители результатов исследования	49
4.3. Анализ конкурентных технических решений	51
4.4. Оценка конкурентоспособности проекта.....	53
4.5. SWOT-анализ.....	55

ВВЕДЕНИЕ

Ранее для управления проектами применялся лишь небольшой набор инструментов: зачастую использовались диаграммы Ганта и таблицы Excel для отслеживания и управления расписанием проекта [1]. Также использовались бумажные дневники, чтобы фиксировать задачи и прогресс, а также для координации и коммуникации между разработчиками и менеджерами проектов. Кроме того, для управления проектами менеджеры могли использовать мнемоники и принципы, такие как концепция «SMART», подразумевающие установление целей, которые являются специфичными, измеримыми, достижимыми, релевантными и своевременными.

Однако, без специальных инструментов управления проектами, процесс оказывался менее эффективным и сложным в автоматизации.

Системы управления проектами значительно упрощают, систематизируют и автоматизируют работу менеджеров проектов и разработчиков, позволяя лучше контролировать прогресс выполнения проектных задач и предсказывать результаты. Большая часть процессов командного взаимодействия, связанных с отслеживанием статусов задач, времени на их выполнение, а также сбор статистики и многое другое в настоящее время автоматизированы благодаря использованию систем управления проектами. Это ускоряет процесс разработки и позволяет разработчикам сосредоточиться на написании кода, а менеджерам гораздо более эффективно контролировать процесс выполнения задач.

Современные системы управления проектами в основном представляют собой кроссплатформенные веб-приложения, что позволяет получать к ним доступ из любой точки мира [2]. Они имеют удобный интерфейс и интуитивно понятные инструменты для создания графиков, диаграмм и отчетов для отслеживания прогресса проекта. Системы управления проектами позволяют сократить время на планирование и увеличить эффективность работы команды. Многие из таких систем имеют интеграции с системами контроля версий (VCS), такие как «Git», и с системами автоматизации сборки, такие как «Apache Ant» и «Make», которые позволяют выполнять задачи, необходимые для разработки, тестирования и сборки приложения, с минимальным участием разработчиков.

Таким образом, исходя из перечисленных фактов, системы управления проектами стали неотъемлемой частью процесса разработки, а их появление на раннем этапе и внедрение в мир IT дало сильный прирост в эффективности разработки и перевернуло современное представление о процессе командной работы.

Целью выпускной квалификационной работы является разработка новых функций для приложения «Planuway», а именно: создание функции копирования карточки как записи времени, разработка меню коллекций, модернизация функции системных сообщений, а также усовершенствование функции создания зависимых карточек.

Задачи, которые необходимо выполнить для достижения поставленной цели:

- Анализ предметной области систем управления проектами.
- Обзор архитектуры приложения.
- Разработка функционала для модернизации системы.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

Определения

В данной работе использован следующий перечень определений:

Planyway – интегрированная система планирования задач и управления проектами, которая позволяет организовать работу в команде, создать расписание и контролировать выполнение задач.

JavaScript – скриптовый язык программирования для создания динамических и интерактивных веб-страниц на стороне клиента.

TypeScript – язык программирования, входящий в подмножество языков JavaScript, который расширяет функциональность JavaScript, добавляя возможности статической типизации на этапе компиляции.

SPA – веб-приложения, которые загружают одну HTML-страницу и динамически обновляют ее при взаимодействии с пользователем.

React – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

AngularJS – JavaScript фреймворк для разработки масштабируемых веб-приложений.

MobX – библиотека, state-менеджер которая делает управление состоянием простым и масштабируемым за счет прозрачного применения ООП реактивного программирования.

WebStorm – интегрированная среда разработки на JavaScript и связанных с ним технологиях.

Gulp – инструмент-менеджер задач, который используется для автоматизации повторяющихся задач, таких как компиляция исходного кода в CSS и JavaScript, автоматическая перезагрузка страницы в браузере при изменении исходного кода, оптимизация изображений и других файлов и т.д.

Webpack – сборщик модулей, который позволяет объединять несколько JavaScript-файлов в один, обрабатывать файлы изображений, фонтов, стилей и других файлов, а также генерировать бандлы кода для оптимизации загрузки страницы.

Babel – транспайлер, который преобразует код ES6+ в код, который может работать в устаревших браузерах или помогает использовать новейшие возможности JavaScript в современных браузерах.

WebSocket – протокол связи, который позволяет установить постоянное соединение между клиентом и сервером через один порт и обмениваться данными в режиме реального времени.

Bundle (бандл) – сгенерированный JavaScript-файл, содержащий все зависимости, необходимые для запуска веб-приложения.

Бэклог (от англ. backlog) – список задач или работ, представленных к выполнению в рамках проекта.

IIS (Internet Information Services) – веб-сервер для хостинга веб-сайтов и приложений на платформе Windows.

Обозначения и сокращения

В данной работе использован следующий перечень обозначений и сокращений:

UI – User Interface;

MUI – Material User Interface;

DOM – Document Object Model;

HTTP – Hypertext Transfer Protocol;

HTML – Hypertext Markup Language;

API – Application Programming Interface;

IoC – Inversion of Control;

DI – Dependency Injection;

MVVM – Model-View-ViewModel;

IT – Information Technology;

MVC – Model View Controller;

URL – Uniform Resource Locator;

SSL – Secure Sockets Layer;

IDE – Integrated Development Environment.

ОБЗОР ЛИТЕРАТУРЫ

Для анализа архитектуры приложения и разработки модулей по модернизации клиентской части системы была рассмотрена предметная область и средства программной реализации приложения.

В процессе анализа предметной области были исследован рынок приложений для планирования и управления проектами. Исследование о процессах управления проектами было описано в [3]. Основные требования к приложениям для управления проектами в сфере информационных технологий и их описание представлены в источнике [4].

Для модернизации функционала в клиентской части приложения использовались фреймворки для создания масштабируемых SPA-приложений с компонентной и MVVM архитектурой – «AngularJS» и «React». Особенности данных технологий описаны в источниках [8] и [9].

Также в ходе анализа архитектуры приложения использовались базовые средства для сборки проекта. Для целей сборки приложения использовался «Gulp» и «Webpack», особенности по работе с которыми описаны в [6].

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание концепции приложения для управления проектами

Приложение для управления проектами предназначены для упрощения организации и управления командной работой проекта. Для этих целей система управления проектами предоставляет ряд функций, таких как создание задач, управление их статусами, определение сроков задач, а также назначение приоритета и ответственных за задачи. Также одной из важных функций, которыми зачастую обладают приложения для управления проектами, это система оповещений, предназначенная для уведомления причастных к задаче членов команды об изменении её статуса или приближении дедлайна.

Системы управления проектами решают проблемы, связанные с отсутствием структурированности в организации управления и командного взаимодействия, что позволяет повысить эффективность работы и увеличить общее качество выполняемых работ, а также уменьшить вероятность срыва сроков.

Такие приложения подходят для менеджеров проектов при организации командной работы в коммерческой среде, также оно может быть полезно для бизнес-аналитиков для сбора данных, их анализа и принятия каких-либо решений на основе полученных результатов.

В последние годы значительно выросла популярность систем автоматизации управления проектами в IT-сфере. Эти системы предназначены для решения задач управления сложными и динамичными проектами разработки программного обеспечения. Они обеспечивают видимость статуса проекта, облегчают общение между членами команды и позволяют лучше планировать, отслеживать и сообщать о ходе проекта. С помощью этих систем IT-команды могут повысить свою производительность, снизить проектные риски и предоставлять более качественные программные продукты.

Таким образом, основные функции, которые включает приложения для управления проектами [3]:

- календарное планирование проекта на основе задач и их приоритетов;
- управление задачами (изменение их статусов и приоритетов);
- управление командными ресурсами;
- ведение отчётности по выполнению проектных задач;
- интеграции с другими сервисами, предоставляющими функции

календарного планирования

- оповещение участников проекта об изменении статусов задач.

Описанные функции позволяют контролировать командную работу на всех её этапах.

1.2. Описание процессов управления проектами

В результате анализа предметной области управления проектами были выявлены основные бизнес-процессы, необходимые для моделирования.

1.2.1. Анализ процесса управления проектами

Результатом процессов, связанных с управлением проектами, является отчёт о проделанной работе и о состоянии проекта. На рисунке 1 представляет собой детализацию контекстной диаграммы в нотации IDEF0. Данные для диаграммы основаны на основе процессах проектной деятельности, характерных для ИТ-сферы. Контекстная диаграмма показывает общий контекст и окружение исследуемой системы. Она отображает входные и выходные данные, средства и механизмы управления, участвующие в активности процесса или системы, и даёт общее представление о взаимосвязи этих элементов.

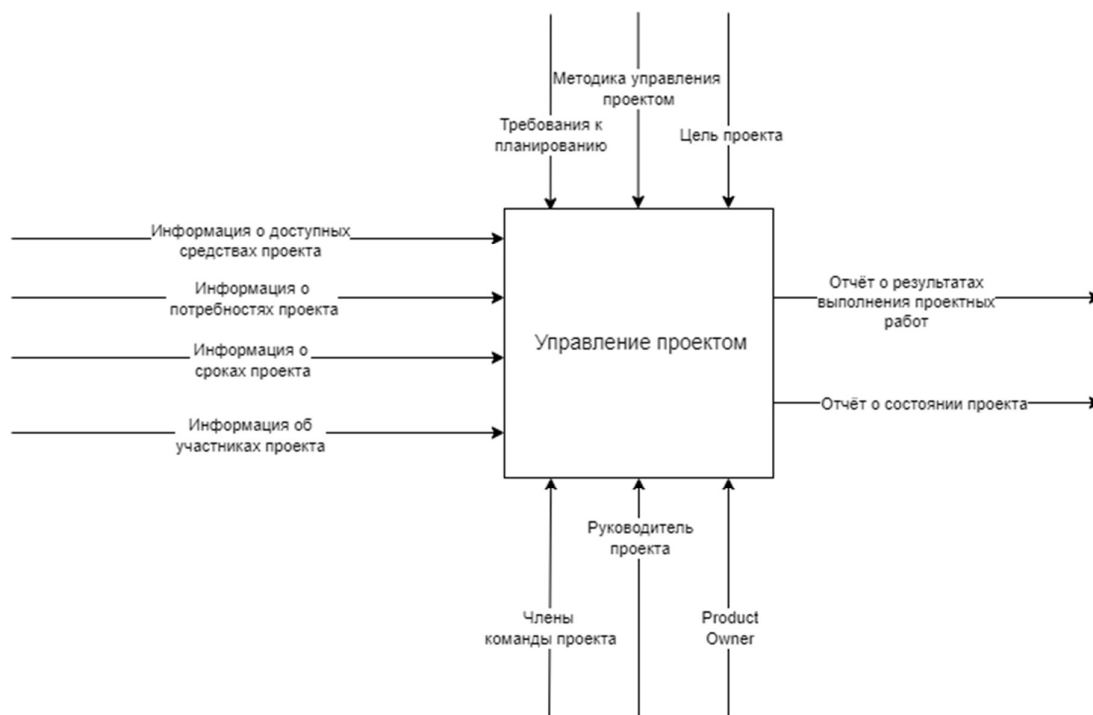


Рисунок 1.1 – Контекстная диаграмма процесса управления проектами

Контекстная диаграмма даёт общее понимание о процессе управления проектом. На вход системы поступает информация о сроках проекта, участниках, его потребностях и доступных средствах. Основные механизмы управления проектом – методики управления

проектом, а также его цель и требования к планированию. На выходе контекстной диаграммы отчёт о результатах проекта и отчёт о текущем состоянии проекта, что позволяет проводить ретроспективу на этапах выполнения проекта с частотой, регламентируемой методикой управления проектом, а также при завершении проекта.

Для более детального рассмотрения процесса управления проектом была спроектирована декомпозиция контекстной диаграммы (рис 1.2).

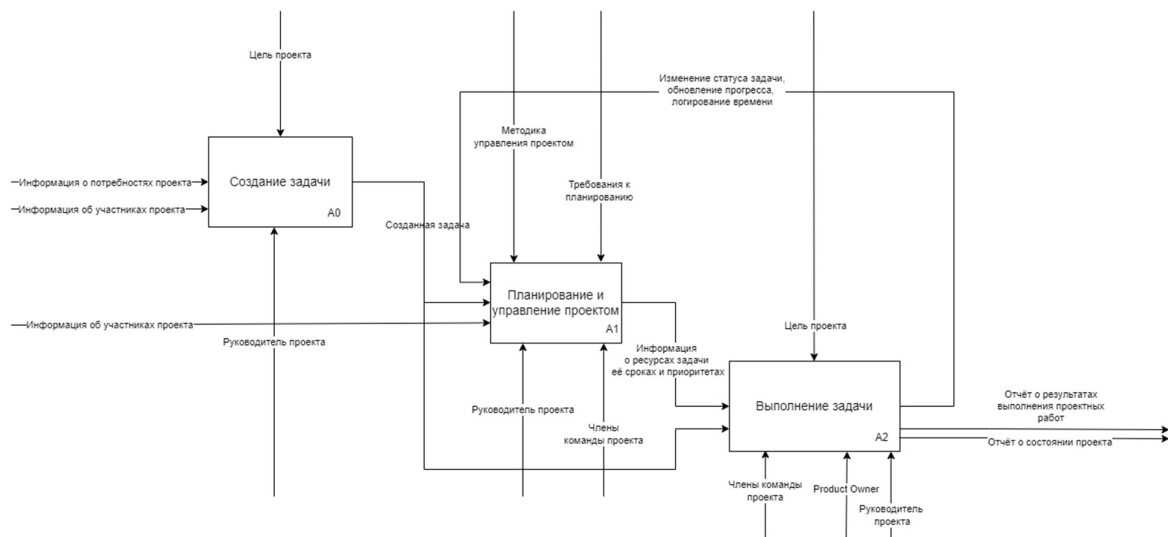


Рисунок 1.2 – Декомпозированная диаграмма IDEF0

Процесс управления проектом основан на базовых операциях:

- A0 «Создание задачи», включающий: создание описания задачи, оценки времени выполнения и сроков, определение исполнителей задачи и её приоритета, статуса задачи на основе информации о потребностях проекта и её создание (добавление в backlog задач).
- A1 «Планирование и управление проектом»: определение требований к задаче, распределение задачи, мониторинг выполнения задачи, отслеживание времени, потраченной на работы, связанные с ней.
- A2 «Выполнение задачи»: исполнение задачи непосредственно её участниками, обновление статусов задачи по мере её выполнения, оповещение о прогрессе выполнения менеджеров проекта и причастных к задаче членов команды.

Детальное рассмотрение процесса A0 «Создание задачи» отражено на рисунке 1.3.

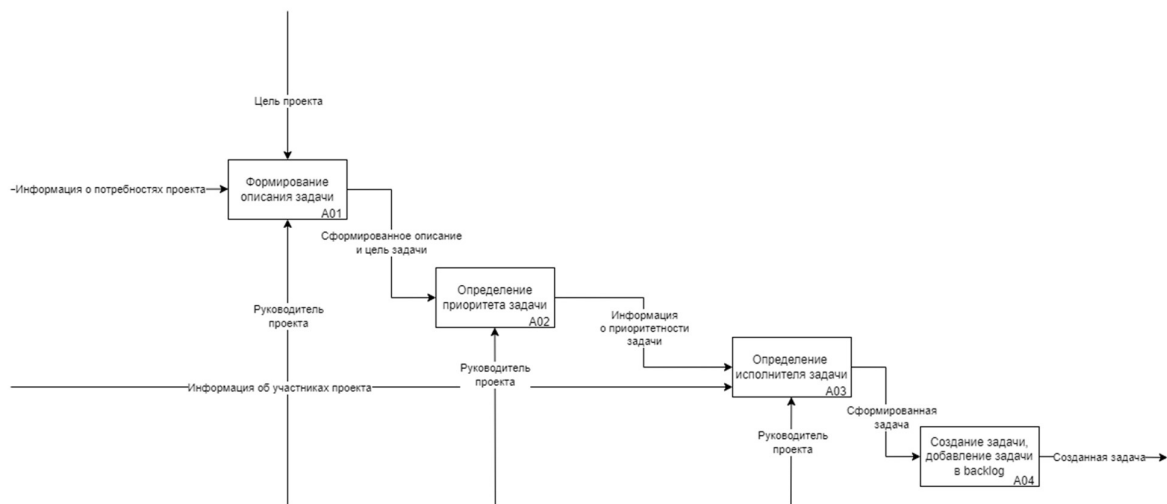


Рисунок 1.3 – Декомпозированная диаграмма процесса A0 «Создание задачи»

Основным исполнителем данного процесса является руководитель проекта. На основе информации о потребностях проекта в создании необходимой функциональности или исправлении текущей с сопутствующей аналитикой, проектный менеджер формирует описание задачи, отражающее её бизнес ценность или при её отсутствии раскрывает проблему, которую решает выполнение текущей задачи. Далее на основе принимая во внимание другие задачи проекта, менеджер определяет приоритет задачи, который отражает важность и срочность её выполнения относительно текущих задач, находящихся в работе. После выбора следует выбор исполнителя(ей), наиболее подходящих под условия, определенные в описании задачи. После выбора участника задача создаётся и отправляется в backlog.

1.2.2. Диаграмма процесса A1 «Планирование и управление проектом» в нотации EPC

Для рассмотрения процесса планирования и управления проектом была составлена диаграмма в нотации EPC (рис 1.4).

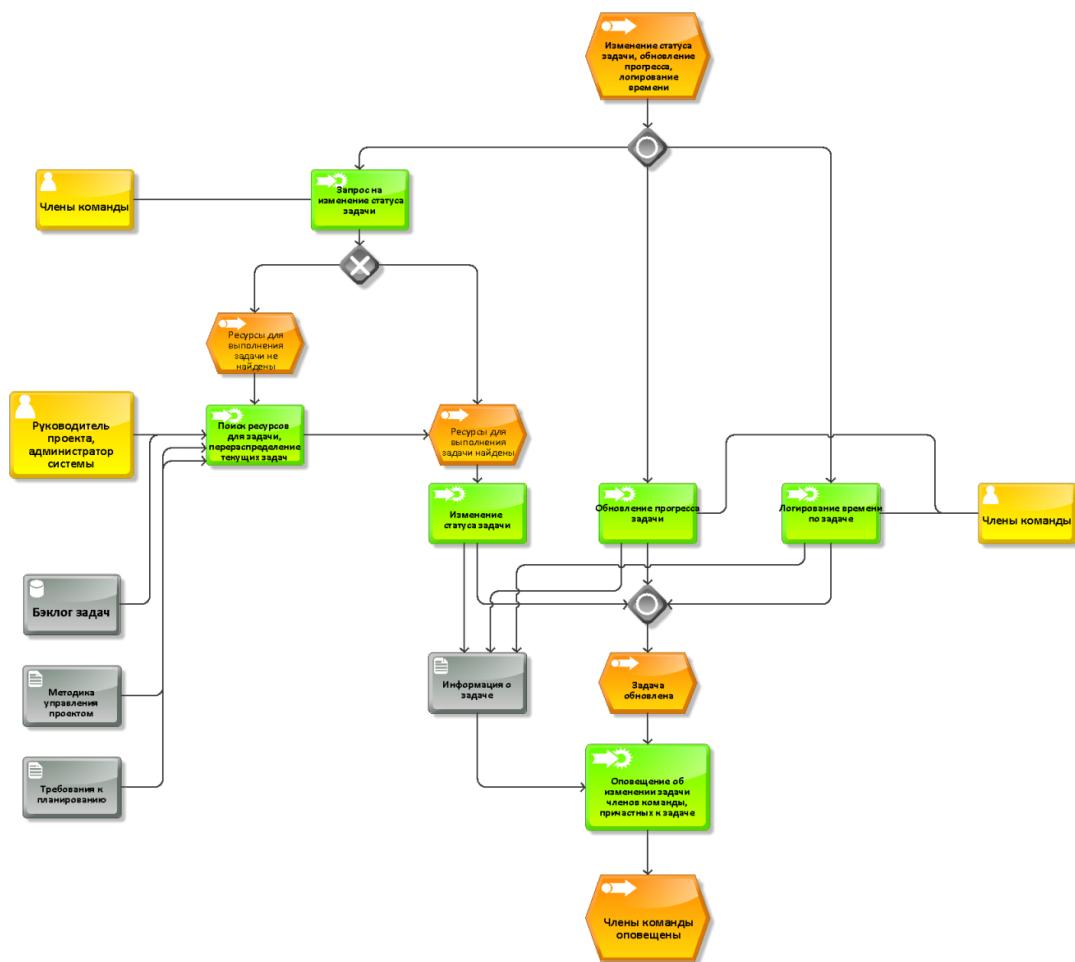


Рисунок 1.4 - Диаграмма процесса A1 «Планирование и управление проектом» в нотации EPC

В процессе планирования и управления проектом принимают непосредственное участие члены команды, которые отвечают за реализацию и выполнение командных работ, а также проектный руководитель, обязанностью которого является контроль за корректное с точки зрения принятым в команде методикам выполнение задач. Данная диаграмма намеренно упрощена рассмотрением только нескольких возможных действий, связанных с планированием задачи, однако на данном уровне декомпозиции можно определить, что основная роль проектного менеджера – валидация задач и подбор ресурсов на их выполнение, а для членов команды – своевременное изменение статусов задач.

1.2.3. Диаграмма процесса A2 «Выполнение задачи»

Для подробного рассмотрения процесса выполнения задачи была построена диаграмма в нотации DFD (рис 1.5).

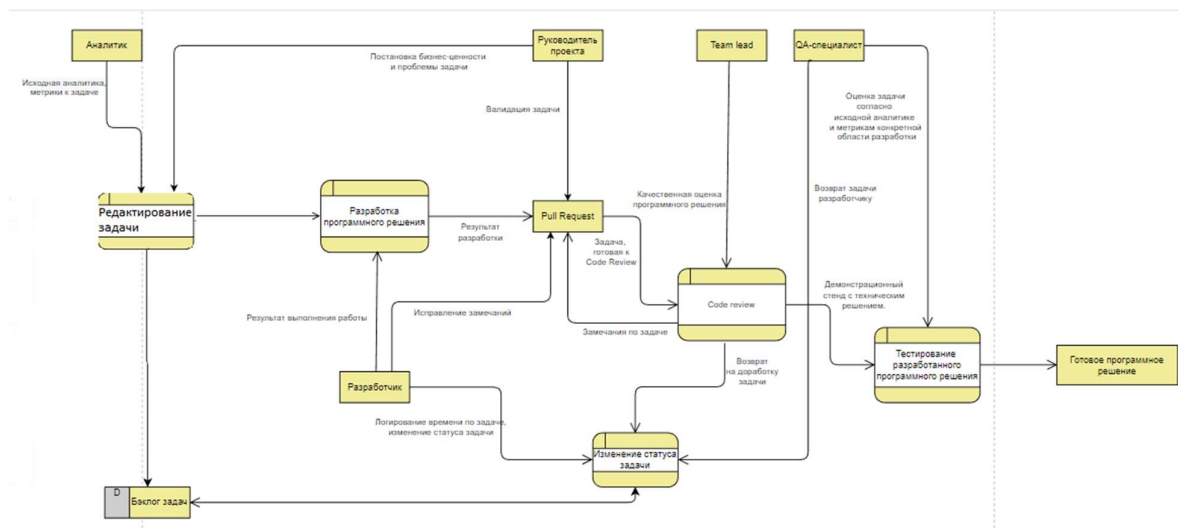


Рисунок 1.5 - Диаграмма процесса A2 «Выполнение задачи»

Данная диаграмма была спроектирована на основе процессов, происходящих при выполнении задачи в IT-сфере [4]. Она отражает переход данных между объектами системы.

Исходная задача поступает из бэклога и содержит необходимую информацию для её технической реализации разработчиком. По окончании выполнения задачи разработчик формирует «pull request», что соответствует формированию запроса на внедрение написанного кода в основную «development» ветку текущего проекта; на диаграмме отражен переход от сущности «Исходная задача» к сущности «pull request». После этого наступает процесс «code review», включающий оценку качества кода членами команды (cross review) или, как продемонстрировано на диаграмме, главным техническим специалистом в команде. После процесса «review» задача переходит на тестирование, по окончании которого выполнение задачи на стороне разработчиков можно считать завершённым.

Данная диаграмма для упрощения намеренно не учитывает некоторые из возможных вариаций валидации задачи на стороне разработки, однако она позволяет вычленить общее представление о процессах перехода данных при выполнении задачи в рамках IT-команды, которые отражают изменения статуса задачи в рамках разрабатываемой системы.

1.3. Требования к системе планирования задач и управления проектами

В результате анализа предметной области систем управления проектами были выявлены основные функциональные требования к системе:

1. Управление задачами. Система должна позволять создавать, назначать, просматривать и редактировать задачи. Каждая задача должна иметь определенный статус и приоритет. Также должна быть возможность добавления комментариев и прикрепления файлов к задачам (при создании описания или прикрепления аналитики к задаче).

2. Планирование проектов. Система должна позволять создавать план проекта, включая список задач, даты начала и окончания работ, ресурсы, необходимые для выполнения проекта и оценку времени выполнения каждой задачи. При изменении одной из задач, система должна автоматически пересчитывать даты окончания и суммарную оценку времени выполнения всего проекта.

3. Управление ресурсами. Система должна иметь возможность назначать исполнителей задач. Также во избежание конфликтов при назначении исполнителей задач, система должна отображать загрузку командных ресурсов и предупреждать о возможных конфликтах при их наличии.

4. Управление временем. Система должна позволять вести учет времени, затраченного на выполнение задач. Работа над задачей должна начинаться и заканчиваться путем изменения статуса задач посредством перетаскивания её в доске «Kanban» или при ручном изменении статуса. Система должна рассчитывать затраченное на задачу время и отображать его в меню задачи и при формировании отчёта.

5. Управление ролями. Система должна иметь возможность ограничивать возможности пользователей или членов команды в проекте в зависимости от их роли. Основные роли, которые должна предоставлять система: администратор – неограниченные возможности редактирования и создания задач; член команды – ограничения на удаления задач; наблюдатель – только чтение.

6. Отчетность. Система должна позволять создавать отчеты о выполнении проектов, состоянии задач, загрузке ресурсов и затраченном времени. Отчеты должны быть визуально понятными и легко читаемыми. Для этого отчёты должны содержать графики, визуально соответствующие действительности.

7. Система уведомлений. Система должна оповещать пользователей об изменениях статуса задач, напоминать о необходимости выполнения задач и предупреждать о возможных проблемах и задержках. Уведомления могут быть отправлены на электронную почту или мобильное устройство, а также посредством push-уведомлений.

8. Интеграция с другими сервисами. Система должна иметь возможность интегрироваться с другими системами со схожим функционалом, а также с корпоративными мессенджерами.

Таким образом, наличие описанных функциональных требований в одном приложении позволит эффективно планировать задачи, управлять проектами и осуществлять коммуникацию между членами команды в единой среде.

1.4. Анализ конкурентных решений

Для анализа конкурентных решений среди систем планирования ресурсов и управления проектами были избраны наиболее функциональные приложения, пользующиеся большой популярностью в сфере информационных технологий, так как данная отрасль включает основную целевую аудиторию приложения. Такими сервисами стали «Jira» и «Trello»:

«Jira» – это система от компании «Atlassian», предназначенная для управления задачами и проектами. «Jira» имеет широкий спектр функционала, такой как возможность создания задач, установка приоритетов, присвоение ответственности, проверка статуса задач и т.д. Однако «Jira» требует длительного изучения, что затрудняет ее использование на ранних этапах.

«Trello» - это простой и интуитивно понятный сервис для управления задачами и проектами. Он предоставляет возможность создавать карточки на «Kanban» доске и перемещать их между столбцами, что сопровождается изменением их статуса. «Trello» также позволяет добавлять комментарии, вложения и устанавливать даты выполнения задач. Недостатком «Trello» по сравнению с «Jira» и другими крупными системами является отсутствие продвинутого функционала, такого как «Gantt-диаграммы» и подробные отчеты.

1.5. Сравнение с конкурентными решениями

Сравнивая разрабатываемое приложение с «Jira», можно заметить, что Jira имеет более мощный функционал, который может быть необходим для выполнения более сложных проектов. Однако, система требует длительного изучения и обучения новых пользователей, что затрудняет ее использование на ранних этапах. С другой стороны, «Trello» предоставляет более простой интерфейс, но не имеет расширенного функционала, на который могут полагаться более продвинутые проекты. Разрабатываемое приложение комбинирует лучшие возможности обоих решений – простой интерфейс «Trello» и продвинутый функционал «Jira».

При более детальном сравнении можно выявить, что все рассмотренные системы наилучшим образом подходят для различных масштабов проектов:

«Trello», обладая наименьшим функционалом, наилучшим образом подходит для небольших проектов и малого бизнеса.

Разрабатываемое приложение обладает широким функционалом и оптимальным интерфейсом, что делает его удобным как для мелких проектов, так и для проектов средних масштабов и среднего бизнеса (1-250 человек).

«Jira» является наиболее масштабным проектом и обладает широким функционалом, в котором зачастую нет необходимости при малых масштабах команд.

В итоге разрабатываемое приложение выделяется на фоне других систем планирования ресурсов и управления проектами благодаря своему оптимизированному интерфейсу, удобным инструментам логирования времени и возможности создания диаграмм Ганта. Уникальные особенности системы, такие как интеграция с различными сервисами, такими как «Google Calendars», «Microsoft Teams», «Trello», создание диаграмм Ганта на основе текущих календарных задач в сочетании с простым и удобным интерфейсом делают разрабатываемую систему оптимальным выбором для бизнеса, которому требуется эффективное управление ресурсами и проектами.

Вывод по главе:

В разделе анализа предметной области была представлена общая информация о проектируемой системе в контексте рассматриваемой предметной области управления ресурсами и проектами.

Также был проведен анализ процессов, характерных для управления ресурсами и проектами, включающий моделирование данных процессов в диаграммах различных нотаций. На основе результатов анализа рассмотренных процессов которого были определены функциональные требования к системе.

При детальном анализе конкурентных решений, были выявлены их основные достоинства и недостатки, а также сформирован общий вывод о конкурентоспособности разрабатываемого приложения.

ГЛАВА 2. ОБЗОР АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ

2.1. Роли и функциональные возможности пользователей приложения

Роли в приложении – это механизм управления правами доступа пользователей к различным функциональным возможностям приложения. Этот механизм создан для обеспечения безопасности и контроля за доступом пользователей к конфиденциальной информации.

В рассмотренных ранее разделах были описана роль пользователя системы в качестве администратора и члена команды. Как было сказано ранее, администратор имеет права редактирования любых задач, имеет полномочия создания задач и их удаления. Также присутствуют роли наблюдателей, членов проекта, авторизованных и неавторизованных пользователей:

- Авторизованный пользователь. Может обладать несколькими ролями в зависимости от принадлежности к проекту. Общая роль для пользователей администратор, член проекта, наблюдатель.
- Неавторизованный пользователь. Роль с ограниченным функционалом использования приложения.
- Администратор. Имеет полный доступ к редактированию проекта.
- Член проекта. Может иметь доступ на редактирование или чтение проекта в зависимости от разрешений, выданных администратором проекта.
- Также в зависимости от подписки, доступ пользователя к функциональности приложения ограничен:
 - Пробный период. Доступны все функции приложения в течение 14 дней.
 - Бесплатная версия. Ограниченные возможности использования приложения, доступ к базовым функциям календаря.
 - «Business-версия». Доступны весь функционал, ограничено максимальное количество пользователей в команде (до 20 человек).
 - «Enterprise-версия». Доступны все функции приложения.

В дальнейшем в примерах будут рассматриваться функциональные возможности пользователя, владеющего «business» подпиской.

2.1.1. Функциональные возможности авторизованного пользователя приложения

Основные функциональные возможности, доступные авторизованному пользователю в зависимости от его роли:

1. Создание, редактирование и удаление задач в календаре.
2. Создание календарей и досок.
3. Создание коллекций.
4. Просмотр своих событий и задач в календаре.
5. Добавление комментариев и прикрепление файлов к задачам.
6. Назначение ответственных и делегирование задач другим пользователям в доске.
7. Определение приоритетов и дедлайнов для задач.
8. Создание шаблонов для повторяющихся задач и событий.
9. Интеграция с «Google Календарем» и «Outlook».
10. Просмотр «iCal-календарей».
11. Определение прав доступа и управление настройками доступа к своим задачам и событиям для других пользователей.

Роли, доступные для авторизованного пользователя, могут включать администратора, членов проекта и наблюдателей. Авторизованный пользователь может выполнять функции, которые ему разрешены в зависимости от роли, выданной администратором текущей доски, в которой находится пользователь.

Диаграмма вариантов использования авторизованного пользователя представлена на рисунке 2.1.

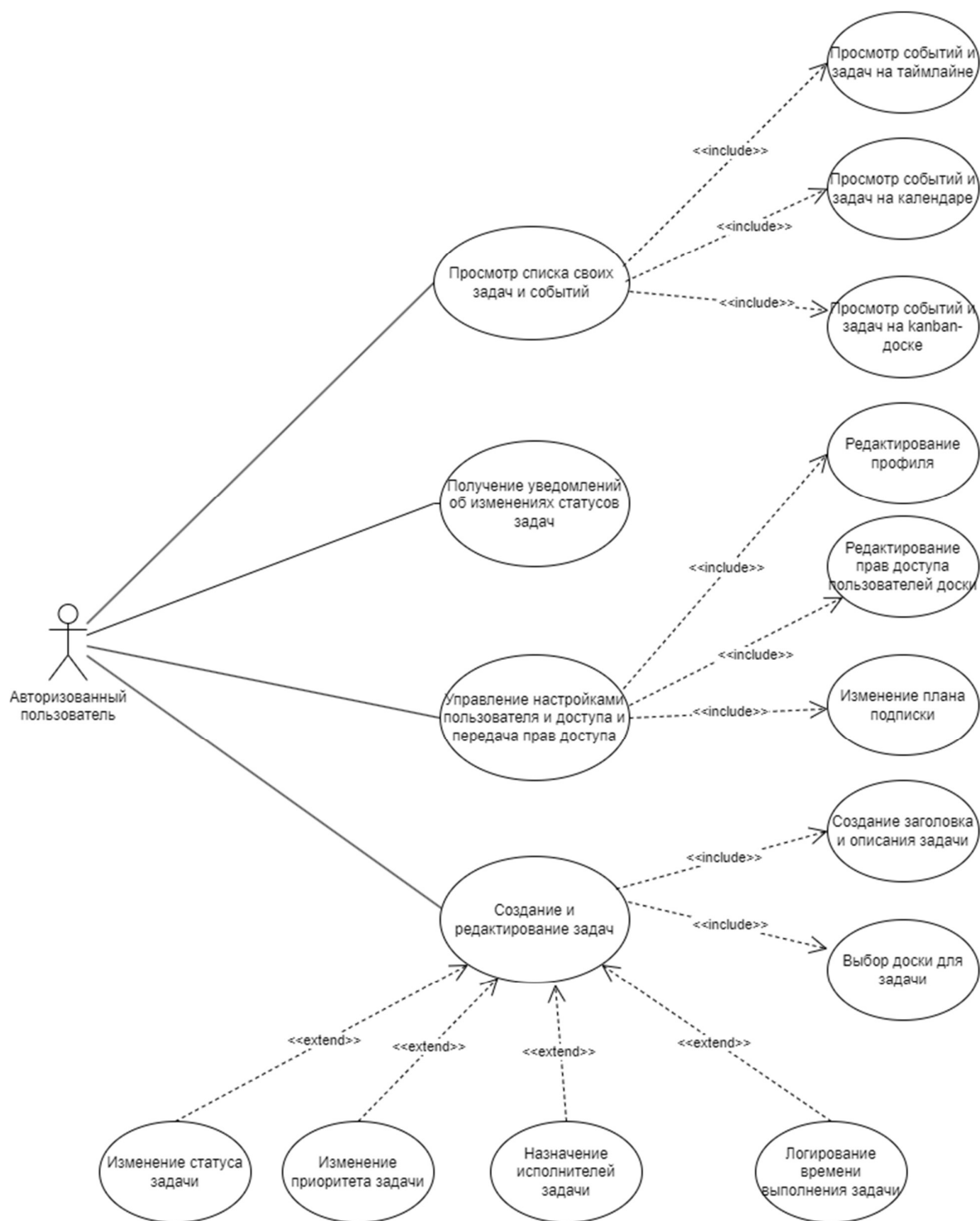


Рисунок 2.1 – Диаграмма вариантов использования авторизованного пользователя

2.2. Создание макетов приложения

В качестве инструмента для передачи дизайнерских решений разработчику использовался онлайн сервис «Figma», который на сегодняшний день является самым популярным среди дизайнеров и фронтенд-разработчиков.

«Figma» – это инструмент веб-дизайна, который позволяет создавать прототипы, макеты и интерфейсы, а также предоставляет возможности совместного редактирования макетов в режиме реального времени.

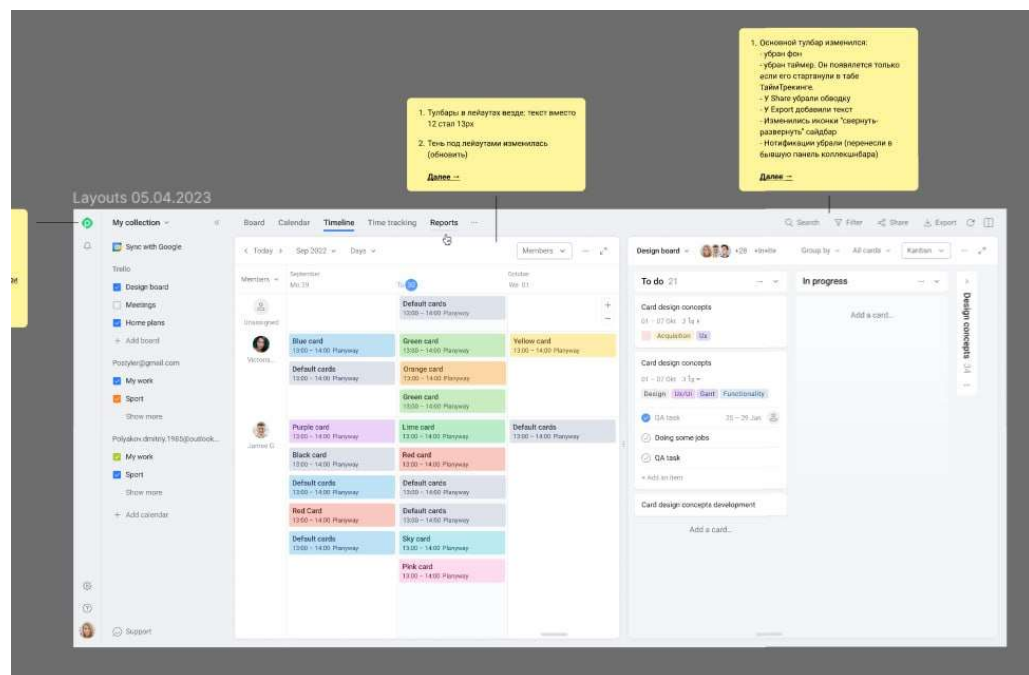


Рисунок 2.2 – Пример макета приложения

Данный инструмент позволяет в удобной форме переносить некоторые CSS свойства, такие как размер шрифта, цвет текста, фона из исходного макета в код приложения.

2.3. Описание программно-технических средств реализации клиентской части приложения

Клиентская часть приложения реализована посредством использования двух популярных JavaScript-фреймворков.

На текущий момент большая часть проекта написана с использованием фреймворка «AngularJS», который основан на сочетании архитектурных паттернов MVC и MVVM. В

свое время данный фреймворк обрёл большую популярность за счёт использования (Dependency Injection) или внедрение зависимостей – это одна из особенностей фреймворка «AngularJS», которая позволяет упростить создание приложений и облегчить тестирование кода.

На сегодняшний день самым популярным JS-фреймворком является библиотека «React». «React» – библиотека для создания пользовательских интерфейсов с открытым исходным кодом. Представляет собой набор инструментов для создания графических интерфейсов и содержит встроенные инструменты для реактивного программирования. Сама по себе библиотека достаточно легковесная, сразу включает «babel» и «webpack» для сборки модулей и транспиляции кода. «React-код» получается достаточно оптимизированным даже без необходимости временных затрат на улучшение производительности со стороны разработчика.

Философия «React» – скорость, а именно скорость разработки и скорость работы. За счёт оптимизированных алгоритмов рендеринга, использованием «virtual-DOM» – виртуальное дерево DOM-элементов и механизма согласования (reconciliation), «React» позволяет избежать лишних рендеров и тем самым оптимизировать производительность итогового приложения. Также среди положительных сторон «React-а» можно отметить возможность использования сторонних библиотек для роутинга приложения, state-менеджмента и работы с HTTP и т.д. Также в отличие от аналогичных JavaScript-фреймворков в основе «React» лежит компонентный подход, что делает его гибким и масштабируемым.

Согласно данным «NPM Trends», инструмента для визуализации количества загрузок JS-пакетов, «React» с сильным отрывом опережает по популярности остальные JS-фреймворки [5].

Downloads in past 2 Years ▾

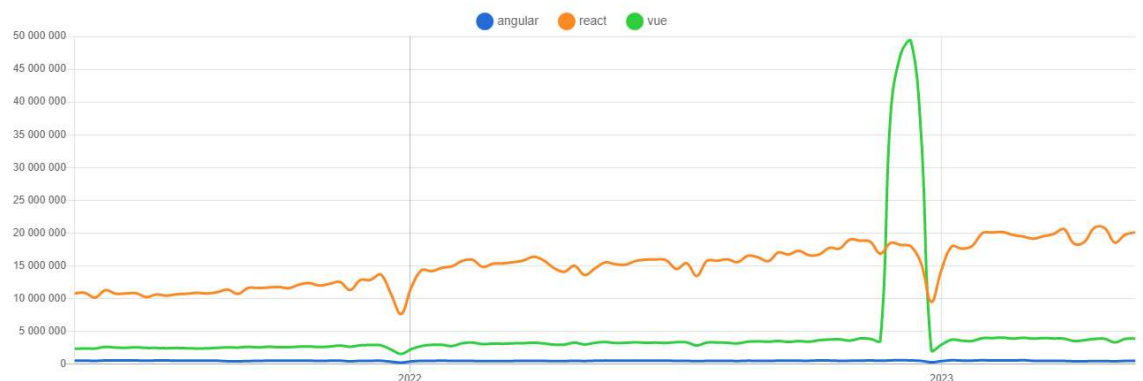


Рисунок 2.3 – Сравнение популярности JavaScript фреймворков

С целью перехода приложения на более современные технологии, приложение переписывается на «React». Для этого используется библиотека «react18-to-angularjs», которая позволяет встраивать «React-компоненты» в «AngularJS». Данный подход позволяет использовать внедрение фабричных angular зависимостей в «React-компоненты».

В качестве языка разработки на стороне «React» используется TypeScript, а на стороне «AngularJS» – JavaScript.

2.4. Архитектура клиентской части приложения

Упрощенно, архитектура приложения представлена на рисунке 8

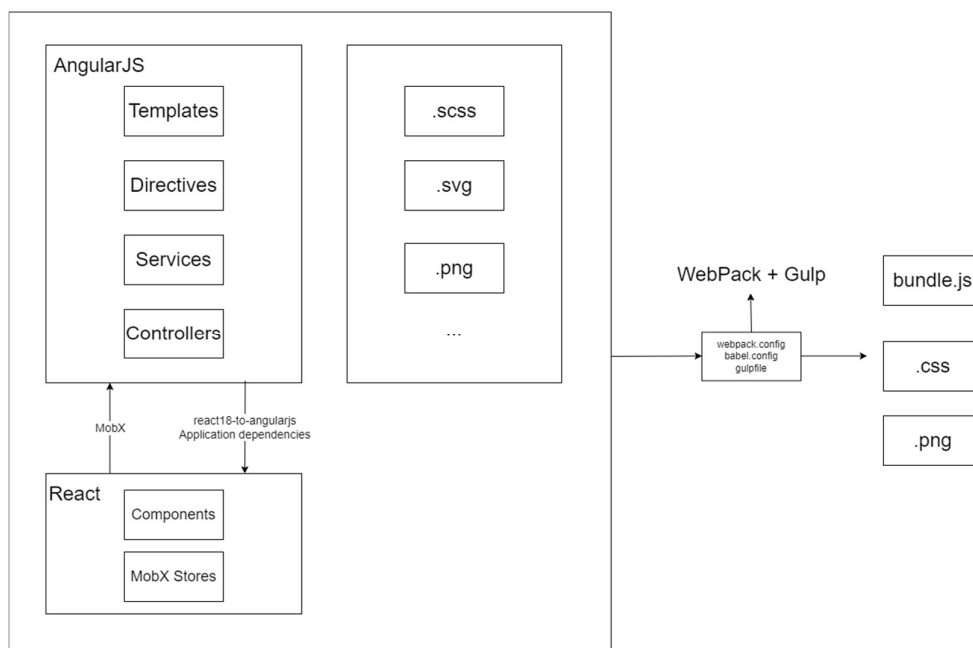


Рисунок 2.4 – Архитектура клиентской части приложения

Как было ранее упомянуто, в качестве фреймворков для разработки используются «React» и «AngularJS».

Архитектура «AngularJS» приложения [9] представлена на рисунке 1.10.

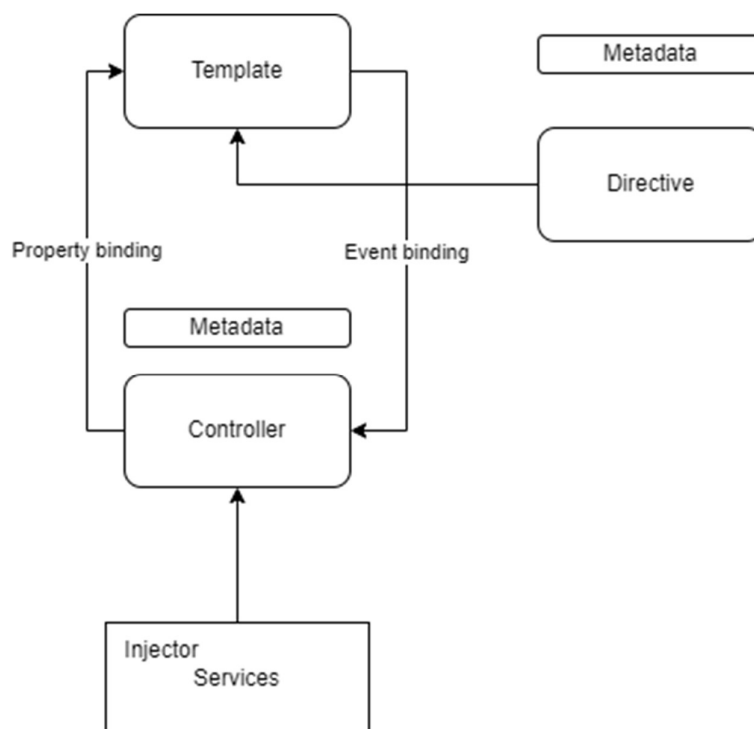


Рисунок 2.5 – Архитектура Angular JS приложения

Для сборки приложения используется связка двух сборщиков «Webpack» и «Gulp». Одной из главных функций «Webpack» является генерация сжатых бандлов, которые содержат только те компоненты, которые действительно используются в приложении. Это позволяет снизить размер файлов и ускорить загрузку страницы.

«Gulp» и «Webpack» позволяют обрабатывать задачи, которые необходимы для сборки приложения. Основные задачи, выполняемые сборщиками проекта [6]:

- Компиляция Sass в CSS.
- Компиляция TypeScript в JavaScript.
- Минификация CSS и JavaScript файлов.
- Объединение нескольких файлов JavaScript в один.
- Копирование файлов в нужные директории.
- Оптимизация изображений.
- Добавление префиксов в CSS.

В качестве веб-сервера для хостинга веб-приложения при разработке используется «IIS». Он позволяет настраивать «deploy» приложения, настраивать SSL-сертификаты для деплоя, а также настраивать перехват запросов, идущих из приложения и заменять целевой URL HTTP-запроса. Данные об адресе целевого запроса хранятся в переменных окружения.

Так как система содержит ряд функций, позволяющих осуществлять интеграцию с другими сервисами приложение имеет несколько сборок.

1. Standalone – основная сборка исходного приложения.
2. Powerup – расширение для Trello.
3. Chrome – расширение для браузера «Google Chrome».

Данные сборки отличаются переопределением некоторых сервисов и компонентов основного приложения. Для каждой из сборок существует своя конфигурация «Gulp» и «WebPack». Подобный подход позволяет уменьшить общий размер бандла приложения, что является ключевым фактором при разработке SPA-приложения, так как чем больше размер бандла, тем дольше происходит загрузка приложения.

Для графических компонентов приложения используется собственный «прт-пакет» – «planway UI», находящийся в приватном доступе. Данный пакет содержит набор графических компонентов, основанных на компонентах библиотеки «MUJ». Подход, подразумевающий использование графической библиотеки, значительно упрощает разработку, а также при правильном их семантическом использовании зачастую гарантируют корректность работы экранных считывающих устройств. Все компоненты кастомизированны под собственную стилистику и переиспользуются по всему приложению для сохранения консистентности дизайна.

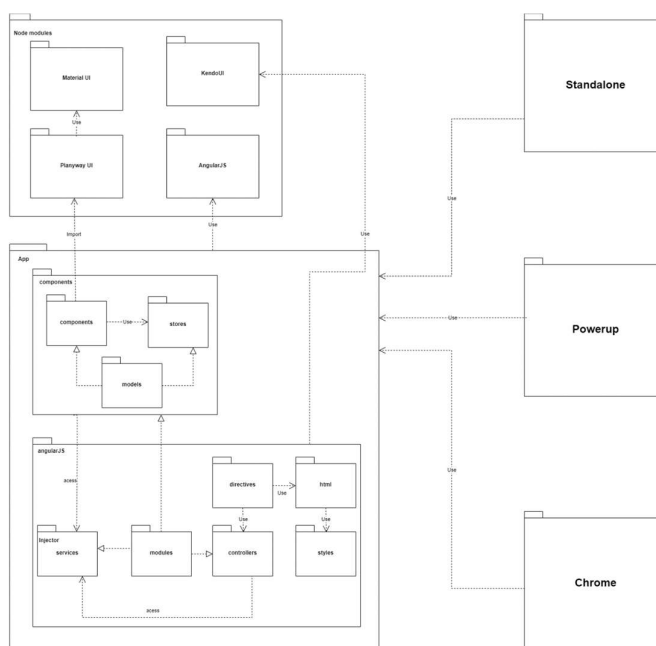


Рисунок 2.6 – Расширенная схема архитектуры приложения

Диаграмма пакетов, представленная на рисунке 2.6 описывает взаимодействие перечисленных модулей системы.

Вывод по главе:

Было приведено описание программно-технических средств, используемых при разработке приложения. В дополнении к описанию программно-технических средств была приведена архитектура клиентской части приложения.

ГЛАВА 3. РАЗРАБОТКА МОДУЛЕЙ ДЛЯ МОДЕРНИЗАЦИИ СИСТЕМЫ

В качестве инструмента разработки используется IDE (интегрированная среда разработки) «WebStorm». Данный инструмент предназначен для разработки на языке «JavaScript» и поддерживает встроенные инструменты форматирования кода, а также содержит удобный графический интерфейс для работы с «Git».

3.1. Разработка модуля приложения интеграции с Google календарём

Calendar API – это веб-сервис, предоставляемый Google для работы с календарем Google. Он позволяет пользователям создавать, изменять и просматривать события в календаре, а также получать уведомления о них.

Основные методы запросов API «Google Calendar» включают в себя [7]:

1. Методы для работы с календарями:

- `calendarList.list` – список всех календарей, которые доступны для пользователя;
- `calendars.get` – получение информации о конкретном календаре;
- `calendars.insert` – создание нового календаря
- Методы для работы с событиями в календаре:
 - `events.list` – список всех событий в календаре за определенный период
 - `events.get` – получение информации о конкретном событии
 - `events.insert` - создание нового события
 - `events.update` – изменение существующего события
 - `events.delete` – удаление события
- Методы для работы с уведомлениями:
 - `channels.watch` – установка веб-канала для получения уведомлений о событиях в календаре
 - `channels.stop` – отмена уведомлений для указанного канала

Таким образом, Google предоставляет набор API для синхронизации календарей с другими приложениями.

```

/**
 * Загружает список подключенных Google календарей
 *
 * @param {AccountId} accountId идентификатор аккаунта
 * @param {boolean} onlyChanged true, если требуется выполнить загрузку только изменившихся календарей, иначе - false.
 * По умолчанию - false
 *
 * @returns {Promise<(GoogleCalendar & { isDeleted: boolean })[]>}
 */
async _getGoogleCalendarsAsync(accountId, onlyChanged = false) :Promise<...> {
    const accountCache :GoogleAccountCache = this._cache.get(accountId);

    try {
        let {
            items: googleCalendars :Object[] ,
            nextSyncToken :string
        } = await this.listAsync(
            accountId,
            path: '/users/me/calendarList',
            description: 'Get Google calendars',
            GOOGLE_CALENDAR_FIELDS,
            options: {
                syncToken: onlyChanged ? accountCache.calendarsSyncToken : undefined
            }
        );

        accountCache.calendarsSyncToken = nextSyncToken;

        return googleCalendars
            .map(calendar :Object => {
                let isReadOnly :boolean = calendar.accessRole !== 'owner' && calendar.accessRole !== 'writer';

                return {
                    id: calendar.id,
                    shortId: md5(calendar.id),
                    accountId,
                    name: getGoogleCalendarName(calendar, accountCache.user),
                }
            })
    }
}

```

Рисунок 3.1 – Пример запроса на получение текущих календарей пользователя

Работа по синхронизации календарей организована с использованием протокола «WebSocket». Это означает, что при изменении данных в Google календаре, в разрабатываемом приложении эти изменения тоже будут видны.

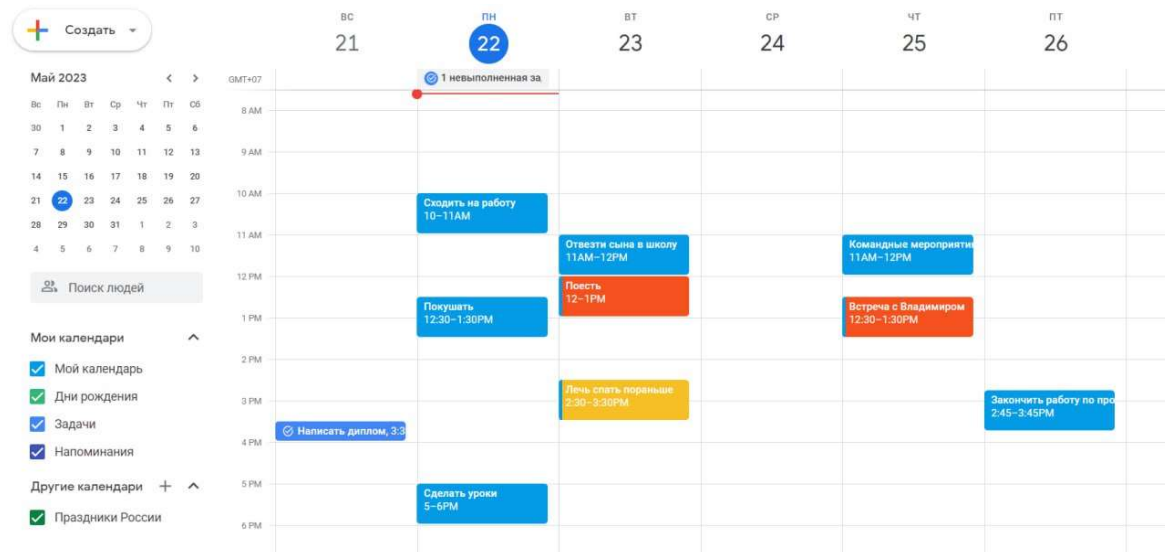


Рисунок 3.2 – Google Calendar

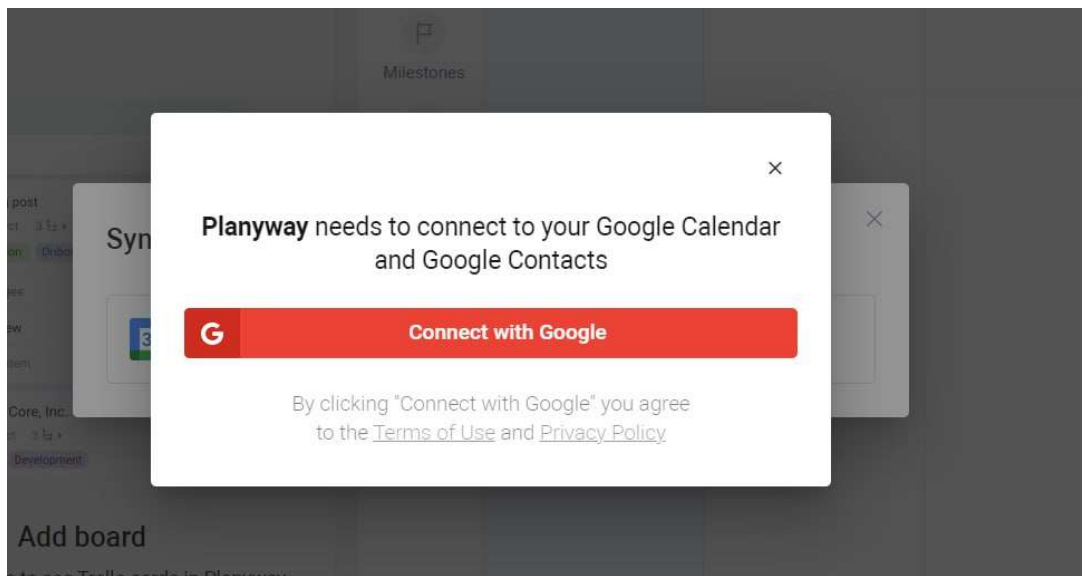


Рисунок 3.3 – Авторизация в Google-аккаунте

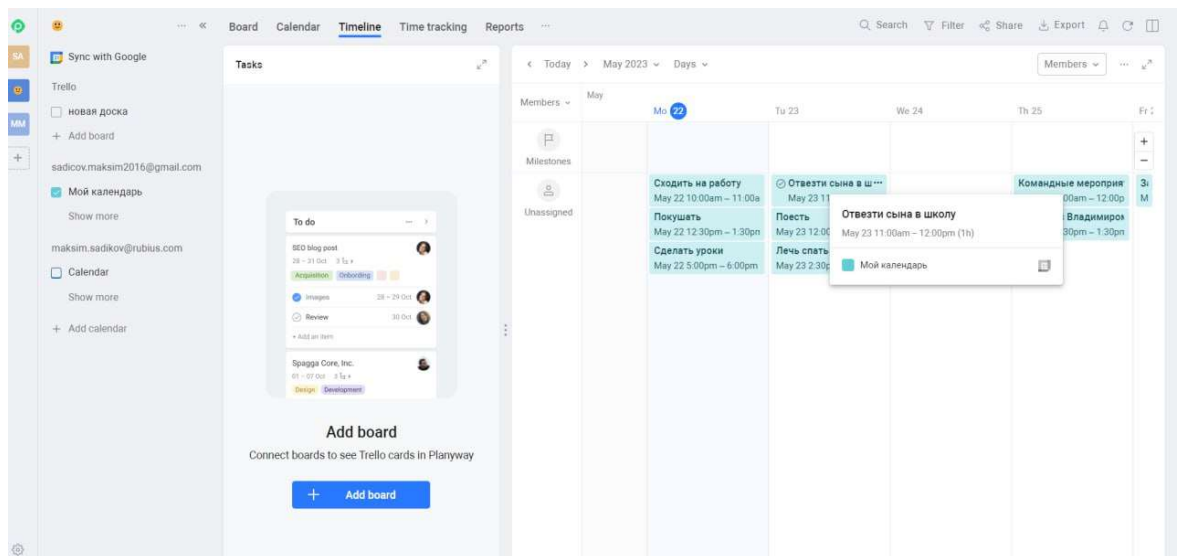


Рисунок 3.4 – Синхронизация календаря в разрабатываемом приложении

Для синхронизации приложения с календарями Google, используется описанный ранее «Google Calendar» API. Сначала пользователь должен предоставить приложению доступ к своему календарю через «OAuth-авторизацию». После этого есть возможность получить доступ к API «Google Calendar» и создавать, изменять или удалять события из календаря пользователя.

Когда пользователь создает новое событие в разрабатываемой системе, приложение отправляет соответствующий запрос к API «Google Calendar», чтобы создать новое событие в календаре пользователя. Если пользователь изменяет событие или удаляет его в «Planyway», происходит отправка соответствующего запроса API «Google Calendar», чтобы обновить или удалить это событие в календаре пользователя.

Одной из важных технических деталей реализации синхронизации является обработка конфликтов. В некоторых случаях может произойти ситуация, когда изменения в событии произошли и в разрабатываемом приложении, и в «Google Calendar» API, но они были выполнены на разных устройствах. В таком случае может возникнуть конфликт между изменениями. Чтобы предотвратить потерю данных, была реализована обработка конфликтов на серверной стороне.

Другая важная составляющая двусторонней синхронизации – это проверка на наличие интернет-соединения перед отправкой запроса к API «Google Calendar» и обработка ошибок соединения. Если соединение отсутствует, приложение отображает сообщение об ошибке и запоминает запросы для отправки в память браузера. Как только соединение восстанавливается, они отправляются в API «Google Calendar» в порядке очереди.

3.2. Модернизация меню коллекций

Коллекции в «Planuway» – это функциональность, которая помогает пользователю объединять несколько календарей в общее хранилище и управлять ими вместе. Коллекции могут содержать календари из различных сервисов, таких как «Google Calendar», «Outlook», и «Trello».

Среди основных функций коллекций в «Planuway» можно выделить:

- Возможность объединения календарей из разных сервисов в общем хранилище.
- Удобное управление календарем, включая быстрое добавление событий и задач.
- Возможность делиться коллекциями с другими пользователями.

Однако коллекции в предыдущем варианте занимали целую панель и с точки пользовательского интерфейса занимали большую часть левой половины боковой панели. Было принято решение переработать панель коллекции в виде выпадающего меню в верхней части боковой панели.

Макет меню коллекций представлен на рисунке 3.5.

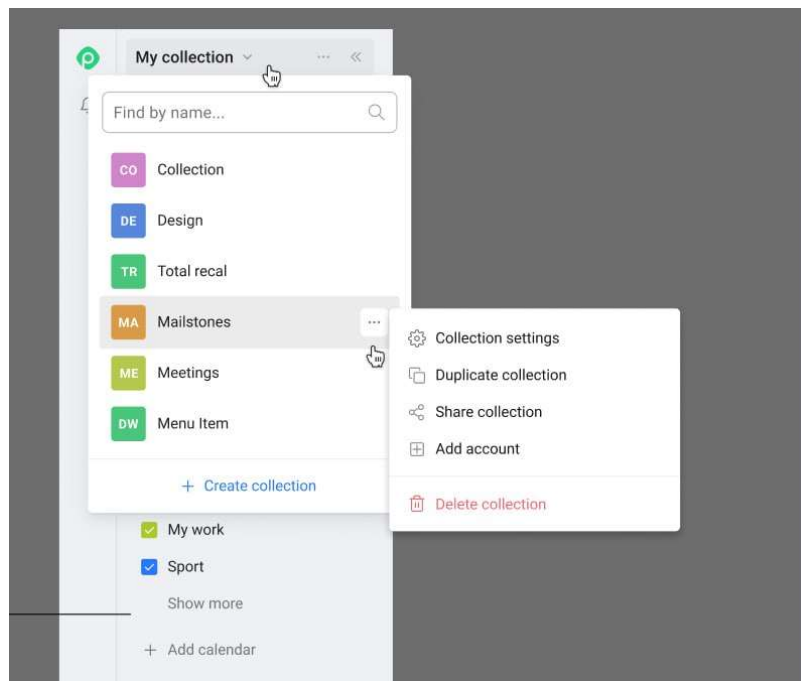


Рисунок 3.5 – Макет меню коллекций

Реализация данного модуля подразумевала следующий перечень работ: переписывание компонентов меню из боковой панели на «React», а также создание меню коллекций. Результатом первого было создание боковой панели с компонентами меню пользователя, настройками и т.д.

Результат переписывания данных элементов интерфейса, а также соответствующей бизнес-логики показал преимущества компонентной архитектуры «React-а» [8]. К примеру, компоненты меню, такие как «PwDebugMenu», «PwProfileMenu» и т.д., которые используются ниже в панели коллекций – «PwCollectionPanel» легко могут быть использованы в любой другой части приложения, так как они подразумевают использование изолированных от остальных частей системы компонентов и типов, кроме компонентов, вынесенных в глобальный scope или библиотечных программных решений.

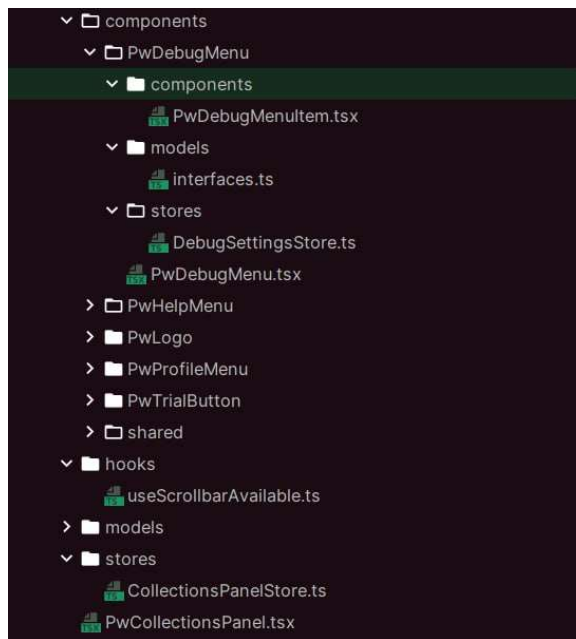


Рисунок 3.6 – Структура модуля CollectionPanel

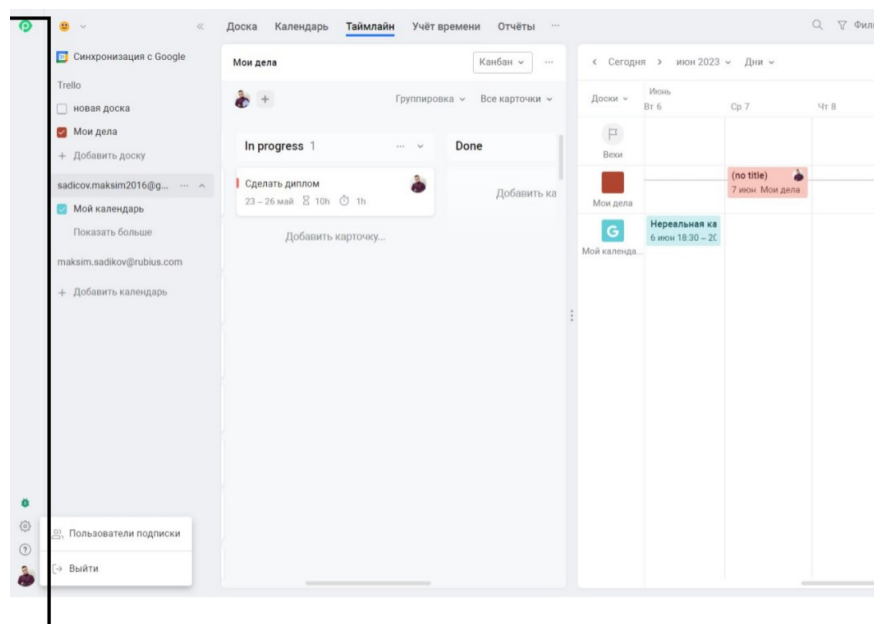


Рисунок 3.7 – Результат разработки

После создания панели коллекций выработанный архитектурный подход к созданию независимых частей приложения был использован при создании меню коллекций. Для реализации меню коллекций был разработан следующий перечень графических компонентов, которые вошли в UI-kit приложения:

«PwMenu» – компонент меню. Принимает в качестве входных параметров компонент кнопки, при нажатии на которую происходит открытие меню.

«PwDragndrop» – компонент для перетаскивания коллекций в меню. Реализован с использованием библиотек «react-dnd», «react-dnd-touch-backend», «react-dnd-touch-preview-backend».

Остальные компоненты можно описать, как компоненты, реализующие бизнес-логику приложения.

Данный модуль является встраиваемым в «AngularJS» компонент боковой панели с помощью библиотеки «react18-to-angularjs». Это подход позволяет использовать сервисы из IoC-контейнера из «AngularJS» напрямую в «React». Пример реализации внедрения зависимостей с помощью библиотеки «react-18-to-angularjs» можно увидеть в приложении А.

Далее зависимости внедряются в «mobx-стор», который создается при инициализации самого верхнеуровневого компонента, который и служит точкой входа для описываемого «React-модуля». После этого в верхнеуровневом компоненте создаётся React-контекст, содержащий в себе все «сторы», используемые в модуле.

«Mob-x» необходим для упрощения работы с реактивностью «React-модуля» и служат единственным источником информации о текущем состоянии описываемой части приложения.

«Сторы» содержат в себе большую часть бизнес-логики компонентов, а именно REST-запросы, взаимодействие с сервисами «AngularJS», что включает получение и обработку данных об изменении состояния в сервисах. Для последней цели поля, для которых необходимо отслеживание изменения состояния, также декорируются как observable поля непосредственно в AngularJS-сервисе. «Сторы» представляют собой TypeScript-классы, содержащие, как правило, поля, которые содержат состояние приложения и используются в компонентах, с модификатором «public», а также методы, инкапсулирующие бизнес-логику приложения.

Диаграмма классов разработанного модуля представлена в приложении Б.

Как результат разработки данного модуля, пользователю открывается новый функционал:

- Поиск коллекций с помощью меню коллекций.

- Открытие контекстного меню коллекции без необходимости перехода в неё.
- Просмотр полного названия коллекций в отличие от предыдущей версии.

Демонстрация разработанного функционала представлена на рисунках 3.8, 3.9.

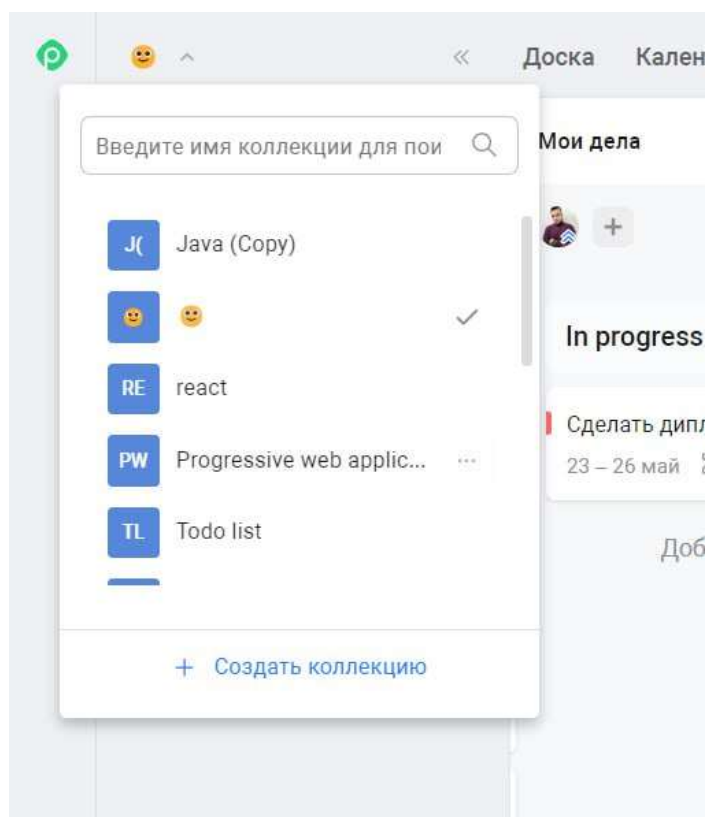


Рисунок 3.8 – Демонстрация функциональности меню коллекций

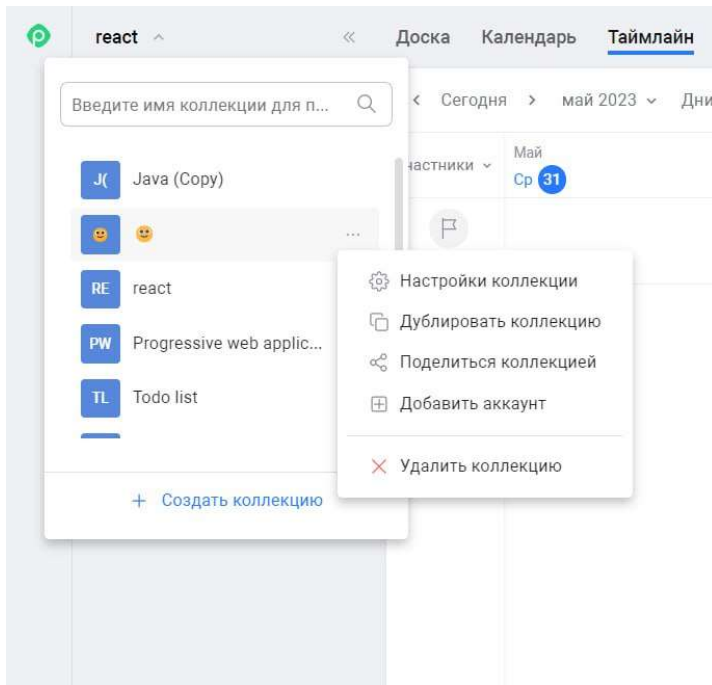


Рисунок 3.9 – Открытие контекстного меню коллекции

3.3. Модернизация функционала системных оповещений

Системные нотификации – это сообщения, которые появляются на экране устройства, чтобы уведомить пользователя о каком-то событии. Они нужны для того, чтобы пользователь мог быстро получать информацию о важных событиях, которые происходят в приложении. Системные нотификации также помогают улучшить опыт использования устройства и приложений, предоставляя удобный способ быстрого доступа к информации и действиям по ее обработке.

Всего в приложении существуют 4 вида системных сообщений:

1. Success – сообщения об успешном завершении операции.
2. Warning – предупреждение.
3. Error – ошибка.
4. Info – информационные сообщения общего назначения.

Также в приложении подразумеваются подвиды системных сообщений, например, сообщение с кнопкой перезагрузки при обновлении настроек системы. К общим требованиям также можно отнести возможность отображения нескольких нотификаций одновременно и возможность закрытия сообщения.

Данный функционал предполагал реализацию отдельного сервиса «React-приложения» для возможности управления системными нотификациями из любой точки приложения.

Сервис был выполнен в виде отдельного «singleton-сервиса», который содержал в себе «observable-стека», в котором временно хранились объекты нотификаций для отображения, а также методы по управлению стеком – pop и push, предполагающие основные операции со стеком – поместить элемент в стек и извлечь.

Также для реализации возможности отображения нескольких нотификаций и кастомизации компонентов нотификаций в зависимости от типа использовались библиотеки «notistack» и «MUI». Разработанный функционал изображен на рисунке 3.10.

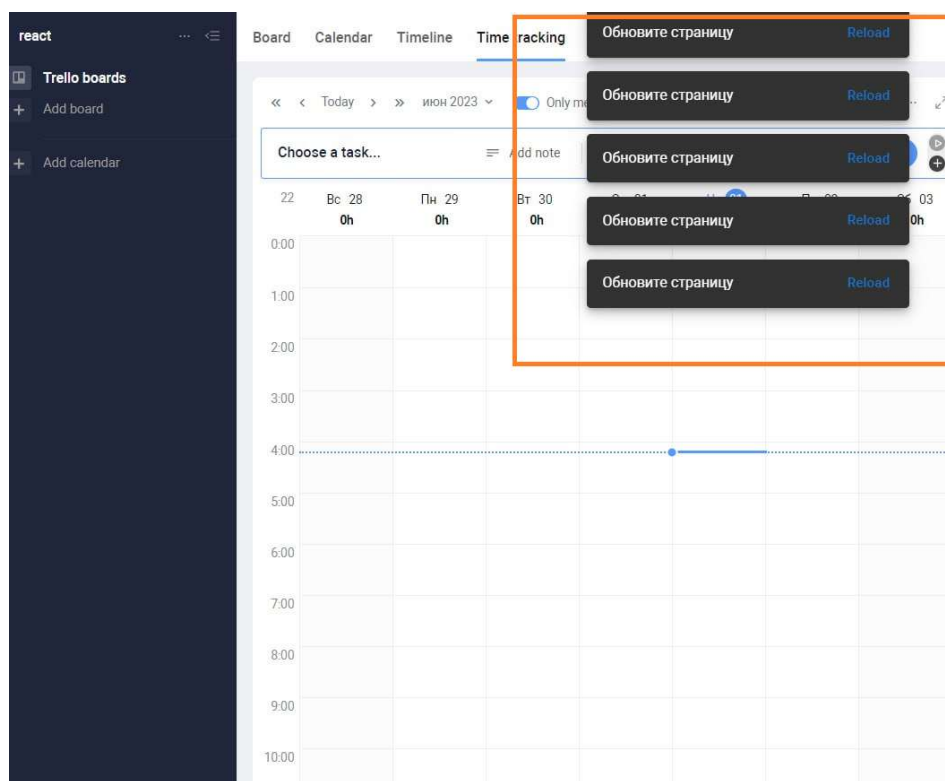


Рисунок 3.10 – Демонстрация функционала системных сообщений

3.3. Реализация копирования записи времени из карточки

«Таймтрекинг» в «Planuway» представляет собой функцию, которая позволяет отслеживать и записывать время, затраченное на выполнение задач и проектов.

Для карточек сторонних сервисов была реализована возможность создания записи времени исходя из параметров самой карточки, т.е. за основу берется дата и описание самое карточки и на их основе создаётся запись времени.

Создание записи времени происходит при выборе в контекстном меню карточки пункта «copy as timeentry».

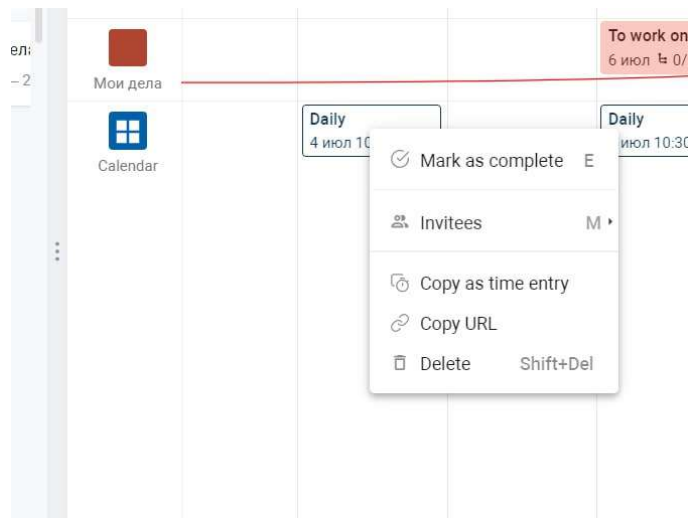


Рисунок 3.11 – Контекстное меню карточки с пунктом «Copy as timeentry»

Затем происходит открытие диалога редактирования записи времени.

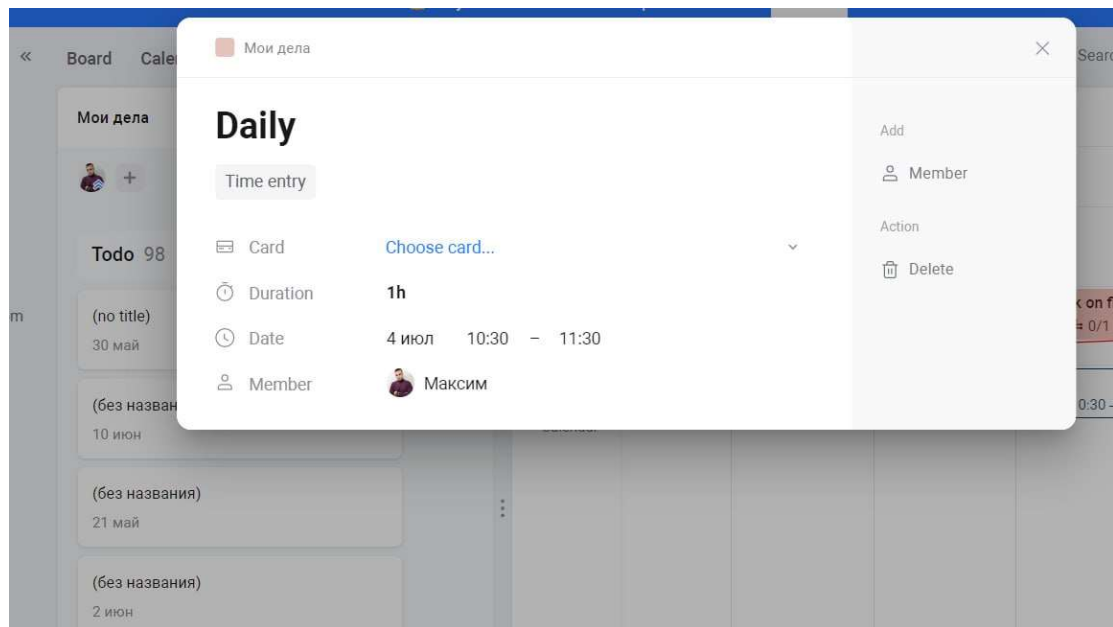


Рисунок 3.12 – Открытие меню создание записи времени с параметрами карточки

Реализация данного функционала включала реализацию метода, преобразующего карточку в запись времени в сервисе «CardsService». Копирование карточки происходит с различными параметрами в зависимости от настроек исходной карточки.

3.4. Создание зависимых карточек из компактного диалога карточки и редактирование зависимостей

«Dependency» в «Planuway» представляет собой функцию, которая позволяет связывать задачи в проекте, определяя зависимости между ними. Это дает возможность пользователям управлять порядком выполнения задач и обеспечивать, чтобы каждая задача была выполнена в правильном порядке, прежде чем начать следующую.

Был реализован функционал создания и редактирования «Dependency» в компактном диалоге карточки, что по предположению улучшит активацию функции создания зависимых карточек (аналог диаграммы Ганта).

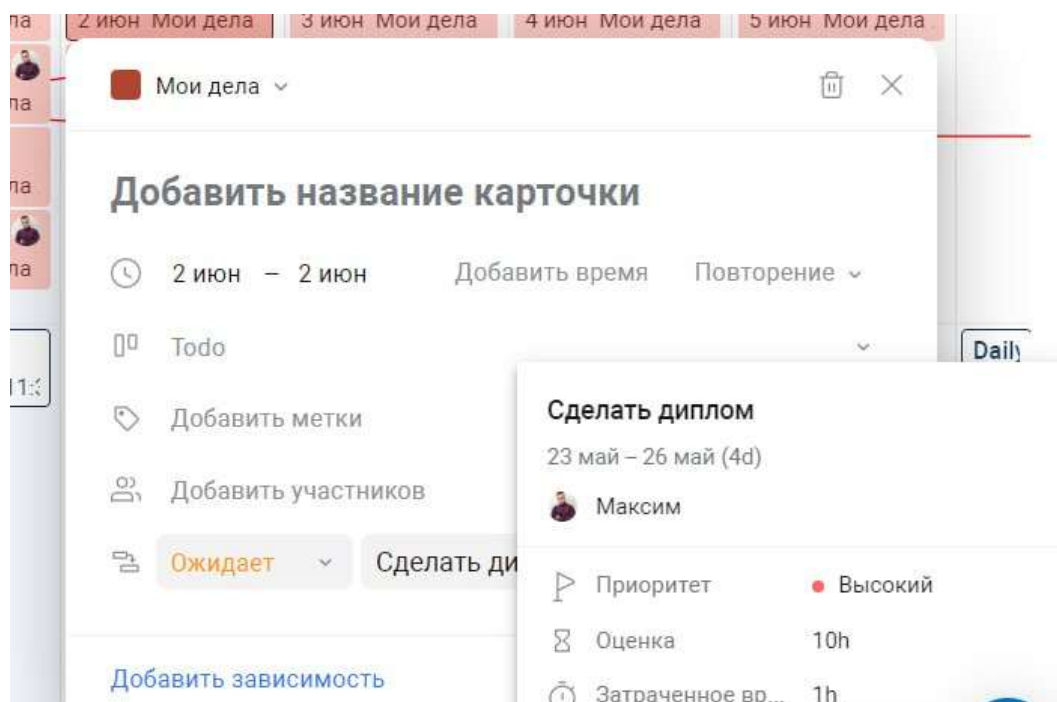


Рисунок 3.13 – Компактный диалог карточки с возможностью добавления зависимых событий

Вывод по главе:

В результате участия в разработке приложения «Planuway» было принято участие в модернизации функциональности синхронизации приложения с «Google Calendars».

Была реализована панель коллекций на базе современного фреймворка «React», что приблизило приложение к переходу на «React» и повысило модульность проекта.

Также было разработано меню коллекций, расширяющее пользовательские возможности по управлению коллекциями. Также меню коллекций позволило избавиться от загроможденности интерфейса и упростить с целью повышения комфортности пользовательского опыта.

Также были модернизированы существующие функции приложения, такие как таймтрекинг, «dependency» и переписана система уведомлений с использованием более современного фреймворка.

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

Студенту:

Группа	ФИО
8K93	Садиков Максим Андреевич

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	Программная инженерия

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Амортизационные затраты на спецоборудование – 22000 рублей; Затраты на основную и дополнительную з/п – 237 884 + 35 682 рублей; Затраты на отчисление во внебюджетные фонды – 82 070 рублей; Накладные расходы – 76 421 рубля
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого и инновационного потенциала НТИ	1. Описание потенциальных потребителей. 2. Анализ конкурентных технических решений. 3. SWOT-анализ.
2. Планирование и формирование бюджета научных исследований	1. Структура работ. 2. Определение трудоемкости. 3. Расчёт бюджета проекта
3. Определение ресурсной, финансовой, экономической эффективности	1. Интегральный финансовый показатель. 2. Интегральный показатель ресурсоэффективности. 3. Интегральный показатель эффективности

Перечень графического материала (с точным указанием обязательных чертежей):

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

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Гасанов М.А.	д-р экон. наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K93	Садиков Максим Андреевич		

ГЛАВА 4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

4.1. Введение

Финансовый менеджмент и экономический анализ необходимы для оценки финансовой жизнеспособности продукта, определения его цены, и определения влияния на финансовую выгоду разработки. Они помогают разработчикам продукта принимать обоснованные решения на основе финансовых данных и минимизировать риски на всех этапах разработки. Также одним из ключевых факторов, которые позволяют определить экономический анализ, является целесообразность разработки продукта, которая зависит от множества факторов. В числе факторов, определяющих целесообразность разработки рыночная потребность в продукте, уровень конкуренции, техническая реализуемость проекта.

Данный раздел направлен на проведение экономического анализа программного продукта для определения его успешности и перспективности, оценки эффективности и возможных рисков.

Выпускная квалификационная работа представляет собой разработку веб-приложения, которое позволяет планировать персональные задачи и командную работу. Функционал приложения включает в себя возможности планирование командных ресурсов и контроля времени работы над задачами, а также интеграцию с другими сервисами, имеющими схожий функционал. Приложение имеет интуитивно понятный интерфейс и позволяет создавать, отслеживать и управлять задачами и проектами разных сфер деятельности. Распределение задач между членами команды, установка приоритетов и контроль за процессом выполнением и инструменты для анализа, составления отчётов также входят в перечень функционала приложения. Данный продукт может быть использован в различных сферах бизнеса, в образовании и пригоден для персонального использования. Все задачи и проекты хранятся в облаке, что облегчает работу с ними на разных устройствах и в любом месте.

4.2. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

4.2.1. Потенциальные потребители результатов исследования

Информационная система для планирования и приоритизации потенциально имеет широкий круг потребителей. Система может применяться в личных целях и при работе в

команде или планировании бизнес-процессов и в зависимости от целей пользователей формируется набор необходимых инструментов посредством ограничения возможностей при соответствующем виде подписки.

При персональном использовании приложение предоставляет функционал для планирования задач, с возможностью создания напоминаний и инструментами, позволяющим строго контролировать выполнение повседневных задач и их статусы. Наличие такого функционала может быть при планировании событий и задач в личной жизни, в образовании при планировании расписания занятий и т.д. Практическая значимость для пользователей, использующих приложение в личных целях, также дополняется возможностью интеграции с наиболее популярными сервисами со схожим функционалом, т.е. при наличии уже готовых календарей с запланированными задачами в других сервисах есть возможность их переноса.

Потенциальные потребители, использующие приложение для бизнеса, получают возможности работы в команде, отслеживание затрат времени работы над задачами и другие инструменты, позволяющие автоматизировать командную работу и могут послужить единым источником истины при планировании различных видов работ, определении трудозатрат и выявлении выгодных стратегий организации бизнес-процессов за счёт интуитивно-понятного интерфейса. Возможные сферы применения продукта:

- планирование и управление задачами в IT-проектах и разработке;
- управление временем и задачами в логистике и управлении поставками;
- управление задачами и событиями в медицинских организациях;
- планирование задач и событий в правительственных организациях и некоммерческих организациях;
- управление событиями и задачами в сфере маркетинга и рекламы;
- организация учебных и учебных задач в образовательных учреждениях.

Основная же целевая аудитория с точки зрения платежеспособности и количества потенциальных клиентов – команды в IT-сфере. Система предоставляет полный объем для мониторинга задач и работы по различным методологиям, как по «Scrum», включающий требования к регулярным оценкам задач, проведению ретроспективы и различных строго регламентированных процессов, так и для более простой и гибкой – «Kanban», где от приложения требуется лишь функциональность перетаскивания карточек между досками для изменения их статуса и возможность просмотра бэклога задач.

Исходя из всего вышесказанного, можно сделать вывод о необычной гибкости приложения: его функционал подходит как для использования в личных целях, так и для использования в бизнесе, где все бизнес-процессы протекают стремительно и правильно организованный процесс планирования задач может сыграть ключевую роль в успешности компании.

4.3. Анализ конкурентных технических решений

Конкурентный анализ технических решений - это процесс изучения продуктов и технологий, которые используются конкурентами на рынке. Это важный инструмент, который помогает понять, насколько хорошо продукт или технология соответствует требованиям рынка и каков уровень конкуренции среди продуктов, выпускаемых конкурентами.

Кроме того, конкурентный анализ технических решений может помочь выявить сильные и слабые стороны продукта, а также оценить, какие технологии применяются конкурентами и каковы их возможности для развития. Это позволяет определить потенциальные проблемы и риски, связанные с рассматриваемым продуктом.

В России существует широкий рынок приложений для управления проектами и задачами, использующих карточки и доски для визуализации процессов. Стартапы, такие как «Trello», «Asana», «Jira» и «Wrike», активно привлекают пользователей своими удобными интерфейсами и мощными функциональными возможностями.

Карточки и доски стали важной частью процесса управления проектами и задачами в России, поскольку они позволяют визуализировать процессы и сделать их более понятными и доступными для всех участников. Конкуренция на рынке приложений для управления проектами и задачами в России довольно высока, что приводит к появлению новых инновационных продуктов и улучшению существующих. Среди пользователей приложений для управления проектами и задачами на основе карточек и досок в России преобладают молодые предприниматели, стартапы и компании, работающие в IT-секторе

Несколько известных конкурентных технических решений, сравнимых с «Planuway»:

- «Trello» – онлайн-сервис управления проектами и задачами использующий карточки и доски. Согласно данным на 2018 год, количество пользователей «Trello» составляет более 25 млн. человек.

- «Asana» – онлайн-сервис управления проектами и задачами со множеством инструментов для организации работы.
- «Monday.com» – онлайн-сервис управления проектами и задачами с графическими дашбордами и функциями мониторинга.
- «Slack» – платформа для коммуникации и сотрудничества, которая позволяет пользователям общаться и координировать работу через сообщения. Система является специализированным инструментом для командной работы, а её использование способствует увеличению производительности и эффективности команды.
- «Jira» – веб-приложение для управления проектами и задачами, часто используется в области разработки программного обеспечения. Аудитория Jira очень разнообразна и включает в себя как малые и средние компании, так и крупные корпорации. Jira стала популярным решением для различных индустрий, включая IT, финансы, государственный сектор, здравоохранение, образование и другие.

Наиболее популярным из представленных конкурентных решений является «Jira». Данный продукт находится на рынке более 20 лет и за свою историю обрёл массовую популярность в IT-индустрии в связи с наличием функциональных возможностей для интеграции с другими сервисами такими как «Git». Также значительный рост популярности «Jira» объясняется его гибкими возможностями управления задачами и проектами, настраиваемыми шаблонами проектов, большим количеством расширений и функциональных возможностей для сбора и визуализации статистических данных, что может быть полезно при проведении ретроспективы во время цикла разработки программного обеспечения.

Второе наиболее популярное решение среди предложенных – «Trello». В настоящее время «Trello» используется не только стартапами и малым бизнесом, но и крупными корпорациями. Он пользуется спросом во многих отраслях, включая IT, маркетинг, образование и другие. «Trello» помогает управлять проектами, отслеживать прогресс, планировать задачи и оценивать результаты своей команды. Одной из особенностей «Trello» является возможность интеграции с другими инструментами, такими как «Slack», «Google Drive» или «Dropbox».

В дальнейшем для оценки конкурентоспособности проекта будут использоваться именно «Jira» и «Trello», как наиболее успешные среди рынка приложений для управления проектами и задачами.

4.4. Оценка конкурентоспособности проекта

Оценочная карта конкурентных технических решений - это инструмент, который помогает проводить конкурентный анализ более структурированно и объективно. Её использование при экономическом анализе позволяет систематизировать информацию о конкурентах и их технических решениях, упрощает процесс сравнения разных технических решений, а также помогает идентифицировать преимущества и недостатки конкурентных решений.

Для оценки конкурентоспособности проекта составлена оценочная карта с выделенными наиболее важными аспектами продукта (табл. 4.1).

Таблица 4.1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _т	Б _j	Б _р	К _т	К _j	К _р
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Гибкость системы, возможность интеграции с другими сервисами	0.16	7	10	8	1,12	1,6	1,28
2. Удобство использования приложения	0.17	8	5	9	1,36	0,85	1,53
3. Техническая поддержка	0.13	7	10	10	0,91	1,3	1,3
4. Скорость работы приложение	0.15	10	8	8	1,5	1,2	1,2
5. Адаптивность, возможность работы приложения на различных устройствах	0.09	5	7	9	0,45	0,63	0,81
6. Функциональность приложения	0.2	6	10	9	1,2	2	1,8
Экономические критерии оценки эффективности							
7. Стоимость подписки	0.04	9	7	10	0,36	0,28	0,4
8. Экономические выгоды бизнеса от использования приложения	0.06	4	8	9	0,24	0,48	0,54
Итого	1	56	65	72	7,14	8,34	8,86

Где:

Б_р – Разработанное приложение

Б_j – Jira

Б_т – Trello

Анализ конкурентных технических решений определяется с помощью формулы (4.1):

$$K = \sum B_i \times B_i \quad (4.1)$$

Где:

K – конкурентоспособность вида;

B_i – вес критерия (в долях единицы);

B_i – балл i -го показателя.

Средневзвешенное значение показателя качества и перспективности научной разработки составляет 72, что говорит о перспективности научно-исследовательской работы выше среднего.

Исходя из остальных данных, полученных в результате анализа было выяснено, что функциональность приложения является наиболее значимым критерием ввиду непосредственного влияния на возможности и эффективность выполнения задач. Хорошо спроектированная и разработанная функциональность приложения позволяет максимально точно и эффективно решать задачи и принимать управленческие решения на основе доступных данных в рамках одного приложения. Также функциональность позволяет адаптироваться к различным случаям заданий, что помогает экономить время и ресурсы на разработку новых продуктов и решение задач, которые уже были решены в рамках системы. В общем, функциональность обеспечивает согласованность на всех уровнях и позволяет достигать высочайшего уровня производительности при работе с задачами и планировании работы команды. При наличии в одном приложении большого количества функционала и его средним качеством и у другого высоким его качеством, но небольшим количеством, предпочтение отдастся первому, так как возможности пользователя оказываются больше и пропадает нужда использовать другие узкоспециализированные решения.

Также исходя из анализа, «Trello» отстаёт по многим позициям, в отличие от «Jira». «Jira» имеет интеграцию с большим количеством сервисов, что делает её сильным конкурентом. Однако многие жалуются на сложность интерфейса. Интерфейс может быть запутанным из-за большого количества разделов, вкладок, кнопок и ссылок. Некоторые функциональности могут быть неочевидными и скрытыми.

Используя этот недостаток и превратив его в конкурентное преимущество, можно привлечь аудиторию за счёт использования большого набора функциональности приложения и при этом размещении его интуитивно понятным образом во избежание

перегруженности интерфейса. Также стоит сделать акцент на возможности интеграции приложения с другими сервисами, так как этот критерий является одним из главных преимуществ конкурента.

4.5. SWOT-анализ

SWOT – это методика, которая используется для оценки сильных и слабых сторон, возможностей и угроз, связанных с определенным бизнес-проектом или стратегическим решением. SWOT-анализ позволяет в разработке наиболее эффективных стратегий на рынке и ресурсы для их реализации, а также помогает оценить конкурентов и обнаружить их слабые стороны.

SWOT-анализ основывается на 4 категориях, первые буквы которых и составляют само название анализа: сильные стороны (Strength), слабые стороны (Weaknesses), возможности (Opportunities) и угрозы (Threats).

На первом этапе SWOT анализа в таблице 4.2 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации веб-приложения.

Таблица 4.2 - SWOT-анализ разрабатываемого продукта

	Сильные стороны научно-исследовательского проекта: С1. Удобный и интуитивно-понятный интерфейс, что позволяет пользователям быстро начать работу с продуктом; С2. Высокий уровень интеграции с другими инструментами, что упрощает взаимодействие с отслеживаемыми проектами; С3. Широкий функциональный набор, включающий в себя различные методы и технологии управления проектами. С4. Наличие инструментов сбора и визуализации данных по проектам.	Слабые стороны научно-исследовательского проекта: Сл1. Высокий ценник на подписку, что может отпугивать клиентов с небольшим бюджетом; Сл2. Ограниченные возможности в тонкой настройке шаблонов проектов. Сл3. Местами долгий отклик интерфейса на действия пользователя при наличии большого числа карточек, что может создавать дискомфорт для пользователей
--	---	--

	C5. Наличие удобного таймтрекинга.	
Возможности: B1. Развитие проекта с использованием возможностей машинного обучения и анализа данных; B2. Улучшение интеграции с другими инструментами, такими как Jira, Microsoft Teams, Trello и другие; B3. Модернизация системы уведомлений и расширение возможностей шаблонов проектов. B4. Интеграция с сервисом GitHub для привлечения большой аудитории разработчиков по аналогии со схожими сервисами.		
Угрозы: U1. Конкуренция со стороны других продуктов для управления проектами, таких как Jira, Trello и Asana, которые также обладают высоким уровнем функциональности; U2. Увеличение количества требований и ожиданий со стороны пользователей; U3. Снижение спроса на рынке продуктов для управления проектами, вызванное экономическими кризисами или другими факторамиSS.		

На втором этапе SWOT-анализа производится оценка внешней среды, чтобы выявить возможности и угрозы, которые могут повлиять на проект. Оценка соответствия сильных и слабых сторон проекта внешним условиям помогает понять необходимость стратегических изменений. Интерактивная матрица проекта используется для анализа

различных комбинаций взаимосвязей областей SWOT-матрицы и может быть использована в качестве основы для оценки стратегических выборов. Факторы оцениваются знаками «+» (сильное соответствие), «-» (слабое соответствие) и «0» (нет уверенности). Интерактивная матрица проекта представлена в таблицах 4.3, 4.4, 4.5, 4.6.

Анализ интерактивных таблиц представляется в форме записи сильно коррелирующих сильных сторон и возможностей или слабых сторон и возможностей.

Сильные стороны + возможности — какие сильные стороны помогут реализовать возможности.

Таблица 4.3 – Интерактивная матрица сильных сторон и возможностей

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

Коррелирующие сильные стороны и возможности:

- B2C1C2C4;
- B4C2C4;
- B2B3C1;
- B2B4C2;
- B1B2B4C4.

Наличие инструментов сбора и визуализации данных по проектам – наиболее сильная сторона продукта.

При интеграции с другими сервисами такие сильные стороны проекта, как удобный интерфейс и наличие функциональности для визуализации данных по проектам могут положительно сказаться на успешности предпринятой кампании и позволят привлечь пользователей из других сервисов.

Также для интеграции с «GitHub» необходимо развивать возможности проекта по сбору и аналитике данных, так как основная часть пользовательской аудитории данного

решения в основном принадлежит IT-сфере, где аналитика данных является важной частью бизнес-процессов.

Матрица корреляций слабых сторон и возможностей позволяет проанализировать, как слабые стороны могут помешать воспользоваться возможностями продукта.

Таблица 4.4 – Интерактивная матрица слабых сторон и возможностей

	Слабые стороны			
Возможности проекта		Сл1	Сл2	Сл3
	B1	-	-	-
	B2	-	-	-
	B3	-	+	-
	B4	+	+	+

Сильная корреляция:

- B4Сл1Сл2Сл3;
- B3B4Сл2.

Самая слабая сторона проекта – ограничение возможности в тонкой настройке шаблонов проекта. Отсутствие такой функции или её ограниченность может оттолкнуть новых пользователей при интеграции с другими сервисами, а также затормозить развитие существующих и новых инструментов продукта. Одна из возможных стратегий решения проблемы – развитие данной функциональности

Сильные стороны + угрозы - как сильные стороны помогут защититься от угроз.

Таблица 4.5 – Интерактивная матрица сильных сторон и угроз

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

Корреляции:

- У1У2У3С43;

– У1С1С2С3С4С5.

При детальном анализе очевидно, что угроза У1 – конкуренция со стороны других продуктов, успешно перекрывается сильными сторонами проекта. Наиболее сильная сторона – С3 (широкий функциональный набор), наиболее слабая – С1 (удобный интерфейс), так как она меньше всего защищает от текущих угроз.

Слабые стороны + угрозы проекта - какие слабые стороны повышают вероятность, что угрозы навредят бизнесу.

Таблица 4.6 – Интерактивная матрица слабых сторон и угроз

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

Корреляции:

У1У2Сл2Сл3

Долгий отклик интерфейса и ограниченные возможности проекта плохо сказываются на конкурирующей способности продукта по сравнению с другими сервисами, а также способны со временем отрицательно сказаться на привлечении новых клиентов с ростом требований к системам планирования и управления проектам.

В рамках третьего этапа была составлена итоговая матрица SWOT-анализа.

Таблица 4.7 – Итоговая матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта: С1. Удобный и интуитивно-понятный интерфейс, что позволяет пользователям быстро начать работу с продуктом; С2. Высокий уровень интеграции с другими инструментами, что	Слабые стороны научно-исследовательского проекта: Сл1. Высокий ценник на подписку, что может отпугивать клиентов с небольшим бюджетом; Сл2. Ограниченные возможности в тонкой настройке шаблонов проектов.
--	--	---

	упрощает взаимодействие с отслеживаемыми проектами; С3. Широкий функциональный набор, включающий в себя различные методы и технологии управления проектами. С4. Наличие инструментов сбора и визуализации данных по проектам. С5. Наличие удобного по сравнению с аналогами таймтрекинга.	Сл3. Местами долгий отклик интерфейса на действия пользователя при наличии большого числа карточек, что может создавать дискомфорт для пользователей
Возможности: В1. Разработка мобильного приложения, которое расширило бы область применения продукта; В2. Улучшение интеграции с другими инструментами, такими как Jira, Microsoft Teams, Trello и другие; В3. Модернизация системы уведомлений и расширение возможностей шаблонов проектов. В4. Интеграция с сервисом GitHub для привлечения большой аудитории разработчиков по аналогии со схожими сервисами.	Сильной стороной продукта является наличие инструментов для сбора и визуализации данных по проектам. При интеграции с другими сервисами удобный интерфейс и возможность анализировать данные по проектам могут повысить эффективность кампании и привлечь пользователей из других сервисов. Развитие возможностей проекта по сбору и анализу данных необходимо для интеграции с GitHub, так как IT-сфера является основной аудиторией данного решения, где анализ данных является важной частью бизнес-процессов.	Проект имеет слабое место в виде ограниченных возможностей для тонкой настройки шаблонов проектов. Если такая функция отсутствует или ее достаточно неудобно использовать, это может отпугнуть новых пользователей, желающих интегрировать его с другими сервисами, и замедлить развитие как уже существующих, так и новых функций продукта. Один из возможных способов устранения этой проблемы - разработка и улучшение соответствующей функциональности.
Угрозы: У1. Конкуренция со стороны других продуктов для управления проектами, таких как Jira, Trello и Asana, которые также	Риск конкуренции со стороны других продуктов несущественен благодаря преимуществам проекта. Основным преимуществом является широкий функциональный набор.	Долгое время отклика интерфейса и ограниченные возможности проекта снижают его конкурентную способность в сравнении с другими сервисами и могут с течением времени

обладают высоким уровнем функциональности; У2. Реструктуризация стратегических партнерских отношений с некоторыми производителями инструментов, что может привести к необходимости внедрения дополнительных функций или отказа от некоторых областей применения; У3. Снижение спроса на рынке продуктов для управления проектами, вызванное экономическими кризисами, повышением стоимости на входные материалы или другими факторами.	Добавление нового функционала и улучшение существующего позволит привлекать новых пользователей и удерживать старых за счёт регулярного развития инструментов продукта и отсутствия нужды в использовании других сервисов.	отрицательно повлиять на привлечение новых клиентов, у которых возрастают требования к системам планирования и управления проектами. Одной из стратегий исправления может быть оптимизация кодовой базы проекта и обновление технологий, используемых в разработке приложения.
---	---	---

Вывод по SWOT-анализу:

Были рассмотрены слабые и сильные стороны проекта и выявлены две основные стратегии развития проекта, при котором наиболее высока его конкурентоспособность и вероятна его успешность на рынке: развитие возможностей проекта по сбору и анализу данных с использованием искусственного интеллекта позволит повысить охват сфер применения продукта, а также уверенно закрепиться в отрасли ИТ; улучшение производительности продукта и регулярная техническая поддержка позволят приложению закрепить статус качественного технического решения и успешно конкурировать со схожими продуктами за счёт наличия широкого и надёжного инструментария для управления проектами.

4.6. Планирование работ по научно-техническому исследованию

4.6.1. Структура работ в рамках научного исследования

Проектирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках проектирования клиентской части веб-приложения;

- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения технического проекта.

Порядок составления этапов и работ от разработки технического задания и до оформления итогового отчета, распределение исполнителей по данным видам работ приведен в таблице 4.8.

Таблица 4.8 – Перечень этапов работ при проектировании

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Выбор и утверждение темы ВКР и разработка ТЗ	1	Составление календарного плана графика выполнения бакалаврской работы	Инженер-разработчик, Научный руководитель
	2	Разработка ТЗ	Инженер-разработчик
Теоретические исследования выбранного направления	3	Подбор и изучение литературы в исследуемой области	Научный руководитель
	4	Анализ предметной области	Инженер-разработчик
Проектирование функциональности приложения	5	Анализ конкурентных решений	Инженер-разработчик, Научный руководитель
	6	Выявление функциональных и нефункциональных требований системы	Инженер-разработчик
Моделирование системы	7	Построение диаграмм в нотациях IDEF0, EPC, BPMN, IDEF3, UML	Инженер-разработчик
	8	Оптимизация и корректировка бизнес-процессов	Инженер, Научный руководитель
	9	Утверждение оптимизированной модели системы, дополнение функциональных требований	Инженер-разработчик, Научный руководитель

Техническое проектирование приложения	10	Выбор технологического стека и проектирование архитектуры приложения	Инженер-разработчик
Программная-разработка приложения	11	Проектирование БД, написание REST-API с учётом выбранных технологий разработки	Инженер-разработчик
	12	Проектирование дизайна приложения	Инженер-разработчик
	13	Создание пользовательского интерфейса приложения, реализующего функциональные требования системы	Инженер-разработчик
	14	Тестирование приложения, исправление неработающего функционала	Инженер-разработчик
Заключительный этап	15	Проведение ретроспективы по выполненным работам	Научный руководитель, Инженер-разработчик
	16	Составление отчета о проделанной работе и оценка эффективности полученных результатов	Научный руководитель, Инженер-разработчик

4.6.2. Определение трудоемкости выполнения работ

Определение трудоёмкости выполнения работ позволяет оценить количество времени, трудовых и материальных ресурсов, необходимых для выполнения проекта.

Трудоемкость выполнения проекта оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от 60 множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{ожі} = \frac{3t_{ожі} + 2t_{ожі}}{5}, \quad (4.2)$$

Где:

$t_{ожі}$ – ожидаемая трудоемкость выполнения i -ой работы, человеко-дни;

t_{mini} – минимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни;

t_{maxi} – максимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни;

Исходя из величины ожидаемой трудоемкости, можно определить продолжительность каждой i -ой работы в рабочих днях T_{Pi} , при этом учитывается параллельность выполнения работ разными исполнителями. Данный расчёт позволяет определить величину заработной платы.

$$T_{Pi} = \frac{t_{ожи}}{Ч_i} \quad (4.3)$$

Где:

T_{Pi} – продолжительность одной работы, рабочие дни;

$t_{ожи}$ – ожидаемая трудоемкость выполнения одной работы, человеко-дни;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел;

Для перевода длительности каждого этапа из рабочих в календарные дни, необходимо воспользоваться формулой:

$$T_{ki} = T_{Pi} \times k \quad (4.4)$$

Где:

T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{Pi} – продолжительность выполнения i -й работы в рабочих днях;

k – календарный коэффициент.

Календарный коэффициент определяется по формуле:

$$k = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{праз}}} = \frac{365}{365 - 89 - 29} = 1,48 \quad (4.5)$$

Где:

$T_{\text{кал}}$ – общее количество календарных дней в году;

$T_{\text{вых}}$ – общее количество выходных дней в году;

$T_{\text{праз}}$ – общее количество праздничных дней в году;

Расчеты временных показателей проведения научного исследования представляются в таблице 4.9:

Таблица 4.9 – Временные показатели научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях T_{Pi}	Длительность работ в календарных днях T_{ki}
	t_{mini} , чел-дни		t_{maxi} , чел-дни		$t_{ожи}$, чел-дни			
	Научн.п ук.	Инжен ер	Научн.п ук.	Инжене р	Научн. рук.	Инжен ер		
Составление календарного плана графика выполнения бакалаврской работы	1	1	2	2	1,4	1,4	1,4	2,072
Разработка ТЗ	0	2	0	3	0	2,4	2,4	3,552
Подбор и изучение литературы в исследуемой области	1	3	2	5	1,4	3,8	3,8	5,624
Анализ предметной области	0	2	0	4	0	2,8	2,8	4,144
Анализ конкурентных решений	1	2	2	3	1,4	2,4	2,4	3,552
Выявление функциональных и нефункциональных требований	0	2	0	5	0	3,2	3,2	4,736
Построение диаграмм в нотациях IDEF0 и т.д.	0	3	0	5	0	3,8	3,8	5,624
Оптимизация и корректировка бизнес-процессов	2	2	3	3	2,4	2,4	2,4	3,552
Утверждение оптимизированной модели системы, дополнение функциональных требований	1	1	1	1	1	1	1	1,48

Выбор технологического стека и проектирование архитектуры приложения	0	2	0	3	0	2,4	2,4	3,552
Проектирование БД, написание REST-API	0	40	0	60	0	48	48	71,04
Проектирование дизайна приложения	0	10	0	18	0	13,2	13,2	19,536
Создание пользовательского интерфейса приложения, реализующего функциональные требования	0	40	0	60	0	48	48	71,04
Тестирование приложения, исправление неработающего функционала	0	20	0	25	0	22	22	32,56
Проведение ретроспективы по выполненным работам	1	1	2	2	1,4	1,4	1,4	2,072
Составление отчета о проделанной работе и оценка эффективности результатов	1	10	2	18	1,4	13,2	13,2	19,536
Итого:	8	141	14	217	10,4	171,4	171,4	253,672

Согласно данным, полученным в результате расчётов, разработка приложения займёт около 254 календарных дней. Наиболее длительные этапы – разработка клиентской и серверной части, а также проектирование дизайна приложения. Остальные этапы примерно одинаковы по длительности и включают аналитическую часть и моделирование. При этом большая часть нагрузки лежит на разработчике.

Согласно полученным данным можно построить диаграмму Ганта (рис 4.1).

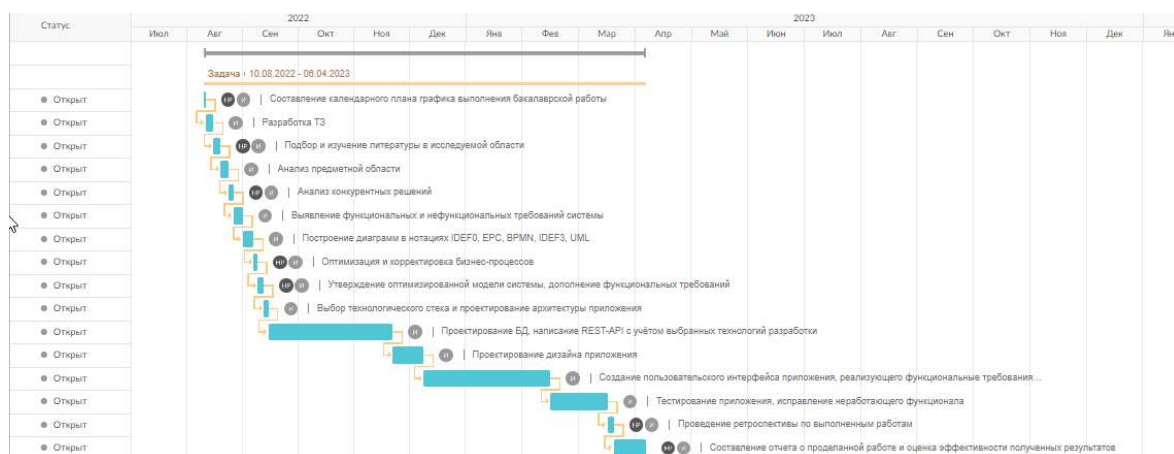


Рисунок 4.1 – Диаграмма Ганта

4.7. Бюджет научно-технического исследования

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

- материальные затраты ТП;
- затраты на оборудование;
- амортизационные отчисления;
- заработная плата исполнителей темы;
- отчисления во внебюджетные фонды;
- накладные расходы.

4.7.1. Расчёт материальных затрат научно-технического исследования

Материальных затрат нет ввиду того, что исследовательская работа проводилась в офисе места работы.

4.7.2. Расчет затрат на специальное оборудование для научных работ

При разработке программного продукта производилось использование программного обеспечения со свободной моделью распространения и программное обеспечение бесплатной версии, урезанного функционала которого было достаточно для проведения исследовательской работы. Поэтому амортизация программного обеспечения в рамках проведения исследовательской работы была равна 0.

Для написания исследовательской работы использовался ноутбук стоимостью около 80000 рублей и периферийные устройства на сумму около 20000 рублей. Общая стоимость технического оборудования составила около 1000000 рублей. Срок полезного использования офисных машин составляет от 2 до 3 лет, для расчётов был принят срок 3 года. Время написания научно-исследовательской работы составило 8 месяцев. Исходя из этих данных, можно вычислить амортизационные отчисления:

Норма амортизации: рассчитывается по формуле:

$$H_A = \frac{1}{n} \quad (4.6)$$

Где: n – срок полезного использования в количестве лет.

Амортизация оборудования рассчитывается по формуле:

$$A = \frac{H_A I}{12} \times t \quad (4.7)$$

Где:

I – итоговая сумма, тыс. руб.; t – время использования, мес.

$$H_A = \frac{1}{n} = \frac{1}{3} = 0,33$$

Общая сумма амортизационных отчислений (по формуле 4.7):

$$A = \frac{H_A I}{12} \times t = \frac{0,33 \times 1000000}{12} \times 8 = 22000 \text{ руб}$$

4.7.3. Основная заработная плата исполнителя темы

Основная заработная плата одного работника рассчитывается по следующей формуле:

$$З_{\text{осн}} = З_{\text{дн}} \times T_p \quad (4.8)$$

Где:

$Z_{\text{дн}}$ – среднедневная заработная плата, руб.;

T_p – продолжительность работ, выполняемых работником, раб. дней. (по таблице 9 для инженера: $T_{p2} = 171$ дней, для руководителя: $T_{p1} = 10$ дней).

Среднедневная заработная плата рассчитывается по формуле 4.9:

$$Z_{\text{дн}} = \frac{Z_m \times M}{F_d} \quad (4.9)$$

Где:

Z_m – месячный должностной оклад работника, руб.;

F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. Дней (в данном случае $F_d = 247$ дней);

M – количество месяцев работы без отпуска в течение года (при отпуске в 118 раб. дней, $M = 8,1$ месяц, 6-дневная рабочая неделя);

Должностной оклад работника за месяц определяется по формуле 4.10:

$$Z_m = Z_{mc} \times (1 + k_{\text{пр}} + k_d) \times k_p \quad (4.10)$$

Где:

Z_{mc} – заработная плата, согласно тарифной ставке, руб (для руководителя $Z_{mc1} = 30000$ руб, а для инженера $Z_{mc2} = 20000$ руб);

$k_{\text{пр}}$ – премиальный коэффициент, равен 0,3;

k_d – коэффициент доплат и надбавок, равен 0,2;

k_p – районный коэффициент, равен 1,3 (для г. Томск);

По формуле 4.10 определяется должностной оклад руководителя за месяц:

$$Z_{m1} = Z_{mc1} \times (1 + k_{\text{пр}} + k_d) \times k_p = 30000 \times (1 + 0,3 + 0,2) \times 1,3 = 58500 \text{ руб}$$

По формуле 4,10 определяется должностной оклад инженера за месяц:

$$Z_{m2} = Z_{mc2} \times (1 + k_{\text{пр}} + k_d) \times k_p = 20000 \times (1 + 0,3 + 0,2) \times 1,3 = 39000 \text{ руб}$$

Среднедневная заработная плата у руководителя рассчитывается по формуле 4.9:

$$З_{дн1} = \frac{З_м \times М}{F_д} = \frac{58500 \times 8,1}{247} = 1\,918,42 \text{ руб}$$

Среднедневная заработная плата у инженера рассчитывается по формуле 4.9:

$$З_{дн2} = \frac{З_м \times М}{F_д} = \frac{39000 \times 8,1}{247} = 1\,278,95 \text{ руб}$$

Основная заработная плата руководителя рассчитывается по формуле 4,8:

$$З_{осн1} = З_{дн1} \times Т_{р1} = 1\,918,42 \times 10 = 19\,184,21 \text{ руб}$$

Основная заработная плата инженера рассчитывается по формуле 4,8:

$$З_{осн2} = З_{дн2} \times Т_{р2} = 1\,278,95 \times 171 = 218\,700,45 \text{ руб}$$

Таким образом, затраты на общую основную заработную плату составляют:

$$З_{осн\text{ обще}} = З_{осн1} + З_{осн2} = 19\,184,21 + 218\,700,45 = 237\,884,66 \text{ руб}$$

Перечисленные информации представляются в таблице 4.10:

Таблица 4.10 – Расчеты основной заработной платы исполнителей

Исполнители НИ	$З_{мс}$, руб	$k_{пр}$	k_d	k_p	$З_м$, руб	$З_{дн}$, руб	T_p , дн	$З_{осн}$, руб
Руководитель	30000	0,3	0,2	1,3	58500	1918	10	19184
Инженер	20000	0,3	0,2	1,3	39000	1278	171	218700
Итого:								237884

4.7.4. Расчёт дополнительной заработной платы

Дополнительная заработная плата учитывает величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Дополнительная заработная плата определяется по формуле 4.11:

$$З_{доп} = k_{доп} \times З_{осн} \quad (4.11)$$

Где:

$Z_{\text{доп}}$ – дополнительная заработная плата;

$Z_{\text{осн}}$ – основная заработная плата;

$k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимаем равным 0,15);

По формуле 4.11 определяется дополнительная заработная плата для руководителя:

$$Z_{\text{доп1}} = k_{\text{доп}} \times Z_{\text{осн}} = 0,15 \times 19184 = 2\,877,6 \text{ руб}$$

По формуле 4.11 определяется дополнительная заработная плата для инженера:

$$Z_{\text{доп2}} = k_{\text{доп}} \times Z_{\text{осн}} = 0,15 \times 218700 = 32\,805 \text{ руб}$$

Таким образом, общая дополнительная заработная плата составляет:

$$Z_{\text{доп общ}} = Z_{\text{доп1}} + Z_{\text{доп2}} = 2\,877,6 + 32\,805 = 35\,682,6 \text{ руб}$$

4.7.5. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Отчисления во внебюджетные фонды определяются по формуле 4.12:

$$Z_{\text{внеб1}} = k_{\text{внеб}} \times (Z_{\text{осн1}} + Z_{\text{доп1}}) \quad (4.12)$$

Где:

$k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд ОМС и социальное страхование). Общая ставка взносов составляет в 2021 году –30% (ст. 425 НК РФ).

Отчисления во внебюджетные фонды для руководителя определяются по формуле 4.12:

$$Z_{\text{внеб1}} = k_{\text{внеб}} \times (Z_{\text{осн1}} + Z_{\text{доп1}}) = 0,3 \times (19184 + 2\,877,6) = 6\,618,48 \text{ руб}$$

Отчисления во внебюджетные фонды для инженера определяются по формуле 4.12:

$$З_{внеб2} = k_{внеб} \times (З_{осн2} + З_{доп2}) = 0,3 \times (218700 + 32\,805,14) = 75\,451,5 \text{ руб}$$

Таким образом, общие затраты на составляется отчисления во внебюджетные фонды:

$$З_{внеб\text{ общ}} = З_{внеб1} + З_{внеб2} = 6\,618,48 + 75\,451,5 = 82\,070 \text{ руб}$$

4.7.6. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и т.д.

Величина накладных расходов определяется по формуле 4.13:

$$З_{накл} = \left(\sum \text{статей} \right) k_{нр} \quad (4.13)$$

Где:

$k_{нр}$ – коэффициент, учитывающий накладные расходы. Величина коэффициента принимается равной 0,16.

$$З_{накл} = (22000 + 100000 + 237884 + 35\,682 + 82\,070) \times 0,16 = 76\,421,76$$

4.7.7. Бюджетная стоимость НИР

Группировка затрат по статьям представляется в таблице 4.11:

Таблица 4.11 – Группировка затрат по статьям.

Статьи							
1	2	3	4	5	6	7	8
Материалы, руб	Амортизация, руб	Основная заработная плата, руб	Дополнительная заработная плата, руб	Отчисления на социальные нужды, руб	Итого без накладных расходов, руб	Накладные расходы, руб	Стоимость бюджета, руб
100000	22000	237884	35 682	82 070	477636	76 421	554 057

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

Для определения эффективности исследования рассчитан интегральный показатель эффективности научного исследования путем определения интегральных показателей финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки рассчитывается по формуле 4.14:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{\text{pi}}}{\Phi_{\text{max}}} \quad (4.14)$$

Где:

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

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения из всех вариантов;

В результате расчетов интегральных финансовых показателей по трем вариантам разработки текущий проект с меньшим перевесом признан считается более приемлемым с точки зрения финансовой эффективности.

Интегральные показатели ресурсоэффективности всех вариантов определяются путем сравнительной оценки их характеристик, распределенных с учетом весового коэффициента каждого параметра (таблица 4.12).

Таблица 4.12 – Сравнительная оценка характеристик всех вариантов.

Объекты исследования Критерии	Весовой коэффициент параметра	Текущий проект	Исп.2	Исп.3
1. Гибкость системы, возможность интеграции с другими сервисами	0.16	1,28	1,12	1,6
2. Удобство использования приложения	0.17	1,53	1,36	0,85
3. Техническая поддержка	0.13	1,3	0,91	1,3
4. Скорость работы приложение	0.15	1,2	1,5	1,2
5. Адаптивность, возможность работы приложения на различных устройствах	0.09	0,81	0,45	0,63

6. Функциональность приложения	0.2	1,8	1,2	2
7. Стоимость подписки	0.04	0,4	0,36	0,28
8. Экономические выгоды бизнеса от использования приложения	0.06	0,54	0,24	0,48
ИТОГО	1	8,86	7,14	8,34

Интегральные показатели эффективности всех вариантов вычисляются на основании показателей ресурсоэффективности и интегральных финансовых показателей по формуле 4.15:

$$I_{эф.i} = \frac{I_{p}^{исп.i}}{I_{финр}^{исп.i}} \quad (4.15)$$

Таблица 4.13 – Сравнительные эффективности разработок.

№ п/п	Показатели	Текущий проект	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,94	0,83	0,93
2	Интегральный показатель ресурсоэффективности разработки	8,86	7,14	8,34
3	Интегральный показатель эффективности	4.48	3.78	4.50
4	Сравнительная эффективность вариантов исполнения	0.96	0,78	0,90

Вывод по главе:

В текущем разделе был проведен анализ результатов разработки с точки зрения финансового менеджмента, ресурсоэффективности и сбережения ресурсов. Была оценена эффективность решения, произведен расчет финансовых и временных затрат. Также был проведен SWOT-анализ, были выявлены сильные и слабые стороны продукта и определены основные стратегии развития продукта. Была выполнены мероприятия по планированию научно-исследовательских работ по проекту и построена диаграмма Ганта, отражающая запланированный график работ.

Бюджетная стоимость научного исследования составила 554057 рублей. Общая продолжительность выполнения – 254 календарных дня.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8K93		Садиков Максим Андреевич	
Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка веб-приложения для управления личными и командными ресурсами	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	<i>Объект исследования:</i> веб-приложения, написанные с использованием фреймворков языка JavaScript <i>Область применения:</i> телефоны, компьютеры, планшеты <i>Рабочая зона:</i> офис <i>Размеры помещения:</i> 80 кв. м. <i>Количество и наименование оборудования рабочей зоны:</i> Персональный компьютер (1 шт.), Периферийные устройства (1 шт.) <i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> разработка и тестирование программного обеспечения
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования. ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора. ГОСТ Р ИСО 9241-7-2009. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (VDT). Часть 7. Требования к дисплеям при наличии отражений МР 2.2.9.2311-07. 2.2.9. Состояние здоровья работающих в связи с состоянием производственной среды
2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов	Вредные факторы: 1. Отсутствие или недостаток необходимого естественного или искусственного освещения 2. Эмоциональные перегрузки 3. Статические физические перегрузки 4. Монотонность трудового процесса

	Опасные факторы: Факторы, связанные с электрическим током, возникающим под действием разницы потенциалов, под действие которого попадает разработчик Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: система естественного и искусственного освещения, соблюдение режима работы, регулярные перерывы в течение рабочего дня, наличие регулируемого по высоте и наклону спинки кресла, наличие устройств автоматического отключения и сигнализации
3. Экологическая безопасность при разработке проектного решения	Селитебная зона: разработка не оказывает никакого воздействия Атмосфера: разработка не оказывает никакого воздействия Гидросфера: разработка не оказывает никакого воздействия Литосфера: загрязнение почвы при неправильной утилизации неисправного оборудования, использованных батареек
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	Возможные ЧС: Внезапное обрушение здания, аварии на коммунальных системах жизнеобеспечения населения, пожар, угроза пандемии. Наиболее типичная ЧС: Возникновение пожара, под которым понимается вышедший из-под контроля процесс горения, обусловленный возгоранием вычислительной техники.
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K93	Садиков Максим Андреевич		

ГЛАВА 5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

5.1. Введение

Основная часть проекта заключается в разработке веб-приложения для планирования персональных задач и командной работы, в частности: планирование командных ресурсов, тайм трекинг, с возможностью интеграции с сервисами со схожим функционалом, к примеру, интеграция и синхронизация с другими календарями.

Приложение с описанным функционалом позволяет в удобном и интуитивно понятном формате создавать, отслеживать и управлять задачами и проектами. С помощью приложения можно легко распределять задачи, устанавливать приоритеты и следить за их выполнением. Оно подходит для использования в разных сферах деятельности, будь то бизнес, образование или личная жизнь. Все задачи и проекты хранятся в облаке, что позволяет с легкостью работать с ними на разных устройствах и в любом месте.

Основная пользовательская аудитория – прогрессивные стартапы, команды разработчиков или любые другие, работающие по методологиям Scrum или Kanban. Приложение содержит обширный инструментарий, позволяющий автоматизировать рабочий процесс согласно упомянутым методологиям.

Актуальность разработки приложения обуславливается ростом доли рынка IT в мировой экономике в последние годы, ведь основная часть команд, пользующая инструментами планирования, трекерами задач, приходится именно на IT. Бизнес-процессы в данной сфере протекают наиболее стремительно, особенно что касается заказной разработки, исходя из данного критерия, требования к организации планированию задач значительно повышаются и возможность автоматизации связанных с этим может сыграть ключевую роль в успешности компании.

Помещение для разработки представляет собой комнату размером 8*10 м. Рабочее место оборудовано персональным компьютером и периферийными устройствами.

5.2. Правовые и организационные вопросы обеспечения безопасности

Трудовой кодекс РФ [10] определяет основные правила взаимоотношений между работодателем и работником, включая вопросы оплаты труда, предоставление отпусков, обязанности по охране труда, нормирование рабочего дня, прохождение больничных и других.

В соответствии с классификацией тяжести труда, работа, которая предполагает взаимодействие работников с компьютерами, относится к категории 1б. В данную

категорию также входят работы, которые выполняются при нахождении в состоянии сидения, стоя или ходьбы и связаны с физическим напряжением.

По законодательству РФ нормальная продолжительность рабочей недели составляет 40 часов. Однако, продолжительность рабочего времени для некоторых категорий работников может быть сокращена (например, для инвалидов I или II группы, а также для работников, занятых на вредных производствах).

При этом виде работ при работе с плоским монитором необходимо чтобы площадь на одно постоянное рабочее место пользователя персонального компьютера должна быть не менее 4.5 квадратных метров.

Температура в помещении должна быть оптимальной и в пределах от 20 до 24 градусов по Цельсию, с влажностью воздуха от 40 до 60 процентов и скоростью движения воздушного потока от 0,2 до 0,3 метров в секунду.

Согласно стандарту безопасности труда, ГОСТ 12.2.032-78 [11], рабочий стол при работе в сидячем положении должен соответствовать современным требованиям эргономики и предоставлять удобное размещение оборудования с учетом его размеров и характера работы. Для этого может использоваться рабочий стол любой конструкции.

При проектировании оборудования и организации рабочего места следует также учитывать антропометрические показатели женщин (если работают только женщины) и мужчин (если работают только мужчины); если оборудование обслуживают женщины и мужчины — общие средние показатели женщин и мужчин.

Размещать рабочие столы следует таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам и естественный свет падал преимущественно слева.

Выполнение требований на данном рабочем месте отражено в таблице 5.1, согласно ГОСТ 12.2.032-78 и ГОСТ 21889-76. «Система «человек-машина». Кресло человека-оператора. Общие эргономические требования» [12].

Таблица 5.1 – Требования к рабочему месту

Требование	Требуемое значение	Значение параметров в помещении
Высота рабочей поверхности стола	Регулируемая высота (от 680 до 800мм)	Соответствует

	Нерегулируемая высота (725мм)	
Рабочий стул	Подъемноповоротный и регулируемый по высоте и углу наклона спинки	Соответствует
Расположение монитора от глаз пользователя	От 600 до 700мм	Соответствует
Освещение	Освещенность на поверхности стола в зоне размещения рабочего документа должна быть равна 300 — 500 лк	Соответствует

ГОСТом Р ИСО 9241-4-2009 «Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре» [13] регламентируется то, с чем будет взаимодействовать пользователь, а также размеры.

Рабочая поверхность буквенно-цифровых клавиш для нажатия должна быть по площади не менее 110 мм², ширина рабочей поверхности клавиши должна составлять от 12 до 15 мм. Указанные ограничения относятся к буквенно-цифровым и цифровым группам клавиш. Рабочая поверхность клавиш и кнопок не должна иметь закругленные кромки и на них не должно быть зазубрин и заусенцев.

Пользователи будут взаимодействовать с компьютером, в составе которого имеется видеодисплей, требования к которому регламентируются в ГОСТе Р ИСО 9241-7-2009. «Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (ВДТ). Часть 7. Требования к дисплеям при наличии отражений» [15].

Проектное расстояние наблюдения для выполнения обычных офисных задач должно быть не менее 400 мм. В некоторых случаях (например, при работе с ярлыками программируемых клавиш на сенсорных экранах) проектное расстояние наблюдения может быть уменьшено до 300 мм.

5.3. Производственная безопасность

Производственные факторы являются частным случаем факторов окружающей человека среды обитания и человеческой деятельности, связанных и (или) порождаемых производственной и трудовой деятельностью.

Опасные производственные факторы (ОПФ) - это факторы, которые могут привести к несчастным случаям на производстве. К ним относятся наличие опасных веществ на производстве, движущиеся механизмы и оборудование, повышенный шум и другие. Также могут быть опасными факторы при работе на высоте, в условиях повышенного или пониженного давления и температуры. ОПФ могут привести к травмам и заболеваниям работников, а также к ухудшению качества продукции и снижению производительности труда.

Вредные производственные факторы — это факторы, присутствующие в окружающей среде на производстве, которые могут привести к вредному воздействию на здоровье работников и причинить заболевания. Такие факторы могут включать в себя химические, биологические и физические агенты, а также факторы психологической нагрузки.

Условия труда, в которых проводится разработка, могут спровоцировать появление вредных и опасных факторов производства, указанных в таблице 5.2.

Таблица 5.2 – Возможные опасные и вредные факторы на рабочем месте инженера-разработчика согласно ГОСТ 12.0.003-2015 [16].

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отсутствие или недостаток необходимого естественного или искусственного освещения	СП 52. 13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*
Эмоциональные перегрузки	МР 2.2.9.2311-07. 2.2.9. Состояние здоровья работающих в связи с состоянием производственной среды
Статические физические перегрузки	Приказ Минтруда России от 29 октября 2021 г. № 774н «Об утверждении общих требований к организации безопасного рабочего места»

Монотонность трудового процесса	Р 2.2.2006-05 Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда
Факторы, связанные с электрическим током, возникающим под действием разницы потенциалов, под действие которого попадает разработчик	ГОСТ 12.1.038–82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов

5.4. Анализ опасных и вредных производственных факторов

5.4.1 Отсутствие или недостаток необходимого естественного или искусственного освещения

Недостаток освещения возникает вследствие отсутствия должного количества источников искусственного или естественного освещения в рабочей зоне. Недостаточная освещенность снижает работоспособность, значительно влияет на здоровье работников, а именно на их качество зрения, а также может вызывать дискомфорт и утомляемость.

В соответствии с требованиями СанПиН 1.2.3685-21 [17] к освещению рабочего места, на поверхности стола необходима освещенность 300-500 лк. Освещение не должно создавать бликов на поверхности экрана, а освещенность экрана не должна превышать 300 лк. Рекомендуется ограничивать неравномерность распределения яркости в поле зрения пользователя, а коэффициент пульсации не должен превышать 5%.

В СП 52.13330.2016 [18] зрительная работа сотрудника, работающего с ПК охарактеризована как работа разряда Б – высокой точности (наименьший эквивалентный размер объекта различения – 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). Для снижения влияния фактора недостаточной освещенности на рабочем месте необходимо, чтобы уровень естественного освещения и яркость экрана персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение и приводит к быстрой утомляемости, а также коэффициент естественного освещения должен быть равен 4,2% и 1,5% при верхнем или комбинированном освещении и при боковом освещении соответственно.

Также согласно СП 52.13330.2016 для искусственного освещения следует использовать энергоэффективные источники света, отдавая предпочтение при равной

мощности источникам света с наибольшими световой отдачей и сроком службы, с учетом требований к цветоразличению. Для общего и местного освещения помещений следует использовать источники света с цветовой температурой от 2400 до 6800 К. Интенсивность ультрафиолетового излучения в диапазоне длин волн 320-400 нм не должна превышать 0,03 Вт/м. Наличие в спектре излучения длин волн менее 320 нм не допускается. Не допускается применение для освещения ламп накаливания общего назначения мощностью 100 Вт и более.

Для того чтобы зрительно расширить пространство и наполнить его светом, можно убрать некоторые перегородки в помещении, которые не являются несущими конструкциями. Также вариантом решения недостаточной естественной освещенности помещения является использование светоотражающих поверхностей.

5.4.2. Эмоциональные перегрузки

Согласно МР 2.2.9.2311-07. 2.2.9 [16] эмоциональные перегрузки могут привести к возможным нарушениям в работе центральной нервной системы и возникновению сердечно-сосудистых заболеваний, а также к нарушениям сна, снижению работоспособности и концентрации.

Для снижения воздействия эмоциональных перегрузок необходимо соблюдать определенные правила: ограничивать время непрерывной работы, не более 4 часов при стандартном 8-часовом рабочем дне и делать перерывы каждый час на 5-10 минут и через каждые два часа - на 15 минут. Также необходимо соблюдать режим сна, вести здоровый образ жизни и уделять время спорту.

Также одним из факторов, которые могут поспособствовать снижению эмоционального стресса, это умение декомпозировать задачи для снижения чувства давления и повышение эффективности труда.

5.4.3. Статические физические перегрузки

Статические перегрузки вызываются длительным пребыванием человека в вынужденной рабочей позе или длительным статическим напряжениям отдельных групп мышц при выполнении работ.

Статические физические перегрузки возникают из-за неправильного оснащения рабочего места, например, стул на рабочем месте не позволяет регулировать высоту и

наклон спинки. Такого рода перегрузки отражаются, преимущественно, на опорно-двигательный аппарат и функциональные системы организма (сердечно-сосудистую, дыхательную и др.). Статические перегрузки могут приводить к таким заболеваниям, как: остеохондроз, сколиоз, варикозное расширение вен, а также общее снижение иммунитета.

Чтобы компенсировать этот фактор, необходимо руководствоваться приказом Минтруда России об организации безопасного рабочего места. Рабочее место должно быть оборудовано креслом с возможностью регулировки его высоты и наклона спинки.

5.4.4. Монотонность трудового процесса

Монотония – напряжение, вызванное однообразием выполняемых действий, невозможностью переключения внимания, повышенными требованиями, как к концентрации, так и к устойчивости внимания.

В процессе деятельности помимо состояния утомления возникает состояние монотонности, отрицательно действующее на психическое состояние и работоспособность человека. Состояние переживания монотонности вызывается действительным и кажущимся однообразием выполняемых на работе движений и действий.

Под влиянием переживания монотонности человек, не умеющий это психическое состояние сдерживать или устранять, становится вялым, безучастным к работе. Состояние монотонности также отрицательно действует на организм человека, приводя к преждевременному утомлению. Психофизиологические проявления состояния монотонии свидетельствуют о пониженной психофизиологической активности человека и заключаются в следующем:

- снижении уровня бодрствования;
- снижении тонуса симпатического отдела вегетативной нервной системы (урежение частоты пульса, снижение артериального давления, увеличение аритмии пульса и др.);
- снижении тонуса скелетной мускулатуры.
- Основными показателями монотонного труда являются:
- структурное однообразие, простота рабочих приемов (операций);
- непродолжительный временной цикл;
- высокая степень повторяемости;
- отсутствие творческих элементов;

- строго определенный регламент действий;
- принудительный ритм и темп;
- дефицит информации о ходе выполнения работы;

Работу за компьютером можно считать монотонной, так как многие из перечисленных факторов характерны для неё.

Для снижения психофизических факторов можно назначать короткие дополнительные перерывы для отдыха работника.

Также одним из вариантов снижения монотонности является внедрение посещения специальных помещений психологической разгрузки. К примеру, комната с настольными играми или спортивный зал. Наличие подобных развлечений и привлечение к ним внимания во время перерыва способно положительно сказываться на снижении утомляемости и общего повышения удовлетворенности трудом.

5.4.5. Факторы, связанные с электрическим током, возникающим под действием разницы потенциалов, под действие которого попадает разработчик

Поражение электрическим током является одним из самых опасных видов вредного воздействия из-за того, что электрический ток является невидимым для глаз человека. Действие электрического тока на организм человека носит своеобразный и разносторонний характер. Проходя через организм человека, электрический ток производит термическое, электролитическое и биологическое действие.

В результате воздействия электротока могут наблюдаться следующие последствия:

- остановка дыхания, сердцебиения, кровообращения;
- потеря сознания;
- ожоговые повреждения кожи и слизистых в местах входа и выхода электрического разряда («петля тока»);
- переломы костей;
- поражение нервной системы.

При работе с компьютером, при прикосновении к его составляющим, могут возникнуть токи статического напряжения.

В таблице 5.4 представлены предельно допустимые значения напряжения прикосновения и тока на рабочем месте разработчика, согласно ГОСТу 12.1.038-82 [19].

Таблица 5.4 – Предельно допустимые значения напряжения прикосновения и тока

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

В качестве мер защиты нужно использовать оградительные устройства, устройства автоматического отключения и сигнализации, изолирующие устройства и покрытия, устройства защитного заземления.

5.5. Экологическая безопасность при разработке проектного решения

Вредного влияния компьютера на окружающую среду во время работы в настоящее время не выявлено. Однако предполагаемый источник загрязнения в виде отходов при поломке компьютера и его атрибутах (в том числе клавиатура, видеодисплейный терминал, мышь, кабели и др.) подлежит утилизации 4 класса опасности. Эта утилизация подразумевает проведение процедур, при которых более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки.

Процедура утилизации должна быть проведена в соответствии с ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» [20].

Также источником загрязнения могут быть отработавшие щелочные, угольно-цинковые и цинковые батареи. Данный тип батареек содержит вредные для окружающей среды металлы и подлежит утилизации, в процессе которой опасные металлы извлекаются из батарей для вторичного производства изделий. Отработанные батарейки принадлежат 2 классу опасности.

Процедура утилизации батареек должна проводиться в соответствии с ГОСТ Р 55101-2012 «Обращение с отходами. Руководство по безопасному сбору, хранению и транспортированию гальванических элементов» [21].

5.6. Безопасность в чрезвычайных ситуациях

Чрезвычайная ситуация (ЧС) – обстановка на определенной территории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которая может повлечь за собой человеческие жертвы, а также ущерб здоровью человека или окружающей среде, значительные материальные потери и нарушение условий жизнедеятельности.

5.6.1. Пожар

Согласно СанПиН 1.2.3685-21, офисное помещение можно отнести к категории В (пожароопасное).

На основании Федерального закона от 22.07.2008 N 123-ФЗ (ред. от 30.04.2021) «Технический регламент о требованиях пожарной безопасности» [22] классом возможного пожара является класс А: горение твердых веществ и материалов.

К первичным средствам пожаротушения относятся все виды переносных и передвижных огнетушителей, оборудование пожарных кранов, ящики с порошковыми составами (песок, перлит и т.п.), а также огнестойкие ткани (асбестовое полотно, кошма, войлок и т.п.).

Первичные средства пожаротушения должны размещаться в легкодоступных местах и не должны быть помехой и препятствием при эвакуации персонала из помещений.

Согласно Постановлению Правительства РФ от 16.09.2020 N 1479 (ред. от 24.10.2022) «Об утверждении Правил противопожарного режима в Российской Федерации», при обнаружении пожара или признаков горения в помещении разработчик обязан:

- немедленно сообщить об этом по телефону в пожарную охрану с указанием наименования объекта защиты, адреса места его расположения, места возникновения пожара, а также фамилии сообщаемого информацию;
- принять меры по эвакуации людей, а при условии отсутствия угрозы жизни и здоровью людей меры по тушению пожара в начальной стадии.

Вывод по главе:

В соответствии с приказом Минтруда России от 15.12.2020 N 903н «Об утверждении Правил по охране труда при эксплуатации электроустановок» (Зарегистрировано в Минюсте России 30.12.2020 N 61957) рабочему помещению относится к категории помещений без повышенной опасности.

В отношении опасности поражения людей электрическим током согласно пункту 1.1.13 ПУЭ-7 рабочая зона является помещением без повышенной опасности. Согласно «Правилам по охране труда при эксплуатации электроустановок» [23] персонал должен иметь первую группу по электробезопасности.

Перечень работы, выполняемые в процессе выполнения ВКР, включая разработку программного обеспечения за персональным компьютером, можно отнести к категории тяжести труда Ib в связи с тем, что большая часть работ производится сидя и сопровождается небольшим физическим напряжением.

В связи с наличием горючих материалов и веществ в рабочем помещении, ему можно присвоить категорию В по взрывопожарной и пожарной безопасности согласно СП 12.13130.2009.

Объекту можно присвоить 4 категорию с точки зрения негативного воздействия на окружающую среду по причине отсутствия выбросов или факторов, негативно влияющих на окружающую среду непосредственно во время разработки.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы был проведен анализ предметной области систем управления проектами, рассмотрена архитектура клиентской части приложения «Planuway» и подходы используемые при разработке данного решения. Также были выполнены задачи, предоставляющие пользователям новым функционал и расширяющие систему, а также была переписана часть функционала системы на более современную платформу, что позволило улучшить модульность и масштабируемость текущей кодовой базы проекта.

В результате проведенного анализа предметной области системы управления проектами были обозначены основные проблемы, с которыми сталкиваются организации при реализации проектов. Были проанализированы существующие системы управления проектами и их особенности с точки зрения эффективности и функциональности, а также выявлена конкурентоспособность текущей системы.

Обзор архитектуры приложения позволил определить ключевые компоненты и модули клиентской части системы. Общее понимание архитектуры проекта помогло при разработке новых модулей приложения и модернизации старых.

В результате разработки функционала по модернизации приложения были созданы новые инструменты и функции для управления проектами. Система была интегрирована с современными инструментами и сервисами, а также был создан функционал, упрощающий взаимодействие пользователя с коллекциями и позволяющий использовать ранее существующие функции в более удобном представлении.

Также были рассмотрены вопросы ресурсоэффективности и ресурсосбережения, а также составлен календарный график выполнения работ по реализации научного исследования и рассчитана общая стоимость разработки.

В разделе «Социальная ответственность» были рассмотрены вопросы организации безопасности при разработке, проведен анализ производственной и экологической безопасности, а также рассмотрены наиболее типичные чрезвычайные ситуации и составлен перечень действий сотрудников при их возникновении.

В результате все задачи, поставленные в ходе выполнения выпускной квалификационной работы, были выполнены. Был освоен подход к разработке в команде «Planuway» и изучен новый стек технологий, а также на примере данного продукта рассмотрен архитектурный подход к реализации масштабируемых веб-приложений.

СПИСОК ЛИТЕРАТУРЫ

1. The Evolution of Project Management // ProjectSmart - URL: <https://www.projectsmart.co.uk/history-of-project-management/evolution-of-project-management.php> (дата обращения: 01.06.2023). – Текст: электронный.
2. Seymour, T «The History Of Project Management» / Seymour, T, Hussein, S. – Текст: электронный // International Journal of Management & Information Systems (IJMIS) : [сайт]. – 2014. – 11 сент. - URL: <https://clutejournals.com/index.php/IJMIS/article/view/8820> (дата обращения: 01.06.2023).
3. Сафонова А.А. Информационные системы управления проектами / А.А. Сафонова, О.Н. Кускачева – Текст: электронный // Формула менеджмента. – 2020. - №1. – С. 22-23. URL: <https://cyberleninka.ru/article/n/informatsionnye-sistemy-upravleniya-proektami/viewer> (дата обращения 01.06.2023). – Режим доступа: Научная электронная библиотека Киберленинка
4. Кагерманов Ш.Ш. Программное и организационное обеспечение гибкого управления предприятиями / Кагерманов Ш.Ш. – Текст: электронный // StudNet. 2020. №8. URL: <https://cyberleninka.ru/article/n/programmnoe-i-organizatsionnoe-obespechenie-gibkogo-upravleniya-predpriyatiyami> (дата обращения: 01.06.2023) – Режим доступа: Научная электронная библиотека Киберленинка
5. angular vs react vs vue // npm trends - URL: <https://npmtrends.com/angular-vs-react-vs-vue> (дата обращения: 01.06.2023). – Текст: электронный.
6. Webpack: The Missing Tutorial // GitHub – URL: <https://github.com/shekhargulati/52-technologies-in-2016/blob/master/36-webpack/README.md> (дата обращения: 01.06.2023). – Текст: электронный.
7. Google Calendar API overview – Текст: электронный // Google Workspace – URL: <https://developers.google.com/calendar/api/guides/overview?hl=en> (дата обращения: 01.06.2023). – Текст: электронный.
8. Компоненты и свойства // React – URL: <https://ru.react.js.org/docs/components-and-props.html> (дата обращения 01.06.2023). – Текст: электронный.
9. Developer Guide // AngularJS – URL: <https://docs.angularjs.org/guide#application-structure> (дата обращения 01.06.2023). – Текст: электронный.
10. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 19.12.2022) [Электронный ресурс] // Консультант : сайт. – Режим доступа:

http://www.consultant.ru/document/cons_doc_law_34683/ (дата обращения: 01.06.2023).

11. ГОСТ 12.2.032-78. Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200003913> (дата обращения: 01.06.2023).
12. ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200012832> (дата обращения: 01.06.2023).
13. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200076334> (дата обращения: 01.06.2023).
14. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200076558> (дата обращения: 01.06.2023).
15. ГОСТ Р ИСО 9241-7-2009. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (ВДТ). Часть 7. Требования к дисплеям при наличии отражений // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200064119> (дата обращения: 01.06.2023).
16. МР 2.2.9.2311-07. 2.2.9. Состояние здоровья работающих в связи с состоянием производственной среды // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200072234> (дата обращения: 01.06.2023).
17. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/573500115> (дата обращения: 01.06.2023).
18. СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 01.06.2023).
19. ГОСТ 12.1.038-82. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/5200313> (дата обращения: 01.06.2023).

20. ГОСТ Р 53692-2009. Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200081740> (дата обращения: 01.06.2023).
21. ГОСТ Р 55101-2012. Обращение с отходами. Руководство по безопасному сбору, хранению и транспортированию гальванических элементов // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/1200103906> (дата обращения: 01.06.2023).
22. ФЗ от 22.07.2008 N 123-ФЗ. Технический регламент о требованиях пожарной безопасности // [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/902111644> (дата обращения: 01.06.2023)
23. Правила по охране труда при использовании электроустановок// [Электронный ресурс] // Кодекс: сайт. – Режим доступа: <https://docs.cntd.ru/document/573264184> (дата обращения: 04.04.2022)

ПРИЛОЖЕНИЕ А

```
...
PwMainSidebarHeader.componentName = 'pwMainSidebarHeader';
//Внедрение зависимостей
PwMainSidebarHeader.componentInjections = [
    'collectionSrv',
    'telemetrySrv',
    'dialogSrv',
    'settingsSrv',
    '$translate',
    'planywayIntegrationSrv',
    'appCommonSrv',
    '$log',
    'notificationSrv',
    'platformSrv',
    '$rootScope'
];

export default observer(PwMainSidebarHeader);
```

Данный пример демонстрирует описание AngularJS зависимостей внутри React-компонента. Внутри компонента описанные зависимости используются для создания стора следующим образом:

```
...

//Описание типа props
type IPwMainSidebarHeaderProps = IMainSidebarHeaderInjections;

//Компонент, в котором используются AngularJS зависимости
const PwMainSidebarHeader = (props: IPwMainSidebarHeaderProps) => {
    const {
        collectionSrv: collectionService,
        telemetrySrv: telemetryService,
        dialogSrv: dialogService,
        settingsSrv: settingsService,
        $translate: translateService,
        planywayIntegrationSrv: planywayIntegrationService,
        appCommonSrv: appCommonService,
```

```

        $log: logService,
        notificationSrv: notificationService,
        platformSrv: platformService,
        $rootScope: rootScope
    } = props;
    //Создание Mobx-стора
    const { current: sidebarCollectionsMenu } = useRef(new
SearchCollectionsForm(collectionService,
    ...

```

Далее в Angular-шаблоне данный компонент используется следующим образом:

```

<!-- Header коллекции React -->
<pw-main-sidebar-header></pw-main-sidebar-header>

```

ПРИЛОЖЕНИЕ Б

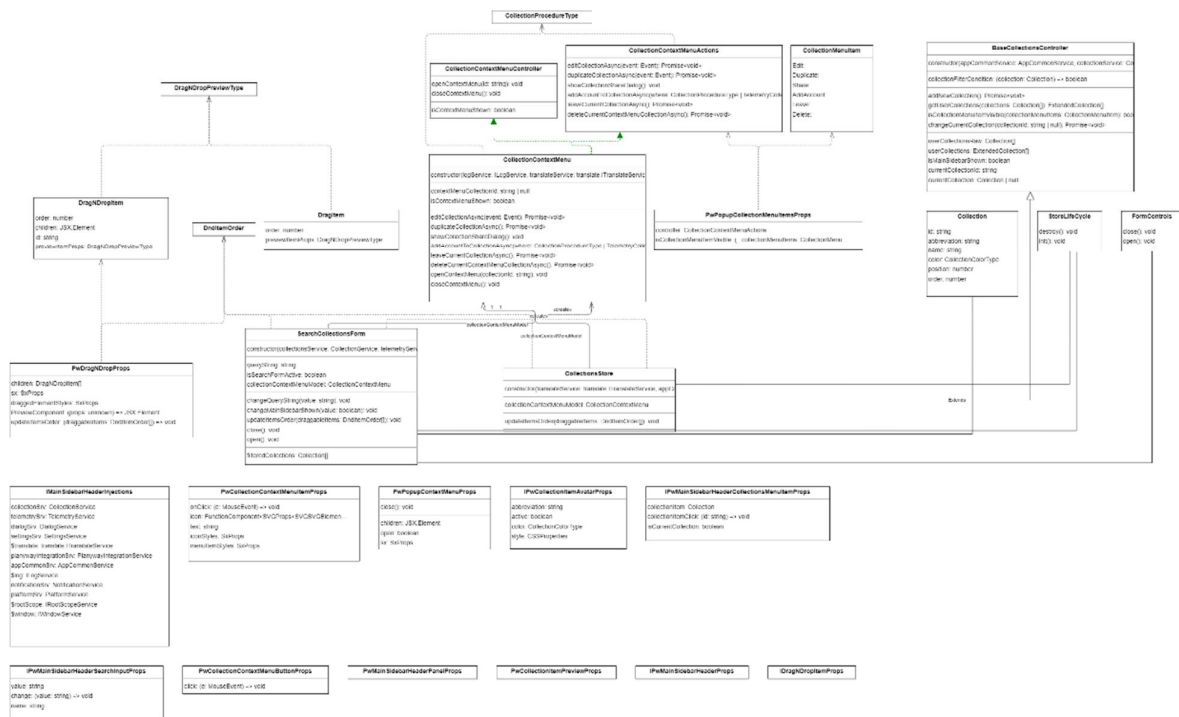


Рисунок 3.14 – Диаграмма классов разработанного функционала