

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра Прикладная математика и информатика  
(наименование)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки, специальности)

Разработка социальных и экономических информационных систем  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка веб-приложения «Платформа получения грантов»»

Обучающийся

А.З. Мехмедов

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.п.н., доцент, О.Ю. Копша

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

## Аннотация

Ключевые слова: сервис получения грантов, бизнес-процессы, веб-технологии, golang, postgresql, vuejs, javascript.

Работа выполнена в объёме 72 страницы, содержит 54 рисунка и 7 таблиц. Бакалаврская работа состоит из введения, трёх глав, заключения и списка использованной литературы.

Тема бакалаврской работы – «Разработка веб-приложения «Платформа получения грантов»».

Целью бакалаврской работы является разработка веб-приложения получения грантов.

Первая глава работы описывает функциональное моделирование проектируемого веб-приложения. В ней проводится анализ предметной области, выделяются интересующие нас бизнес-процессы, проводится их функциональная декомпозиция и формируются требования к веб-приложению.

Во второй главе проводится логическое проектирование веб-приложения, выстраивается её архитектура, разрабатывается диаграмма вариантов использования и логические схемы данных веб-приложения.

Третья глава бакалаврской работы посвящена практической реализации функций веб-приложения. В данной главе выполняется выбор средств реализации ИС, а затем на выбранной платформе реализуются предварительно спроектированные функциональные возможности. Далее проводится оценка экономической эффективности проекта.

В заключении подводятся общий итог работы, перечисляются выполненные задачи и мероприятия, а также описывается практическое значение бакалаврской работы.

## Оглавление

Глава 1 Теоретическая часть.....	6
1.1 Анализ предметной области ООО Цифродата .....	6
1.1 Изучение обеспечения аппаратными и программными средствами отдела онлайн сервисов .....	10
1.3 Анализ бизнес-процессов отдела онлайн сервисов и обоснование необходимости в разработке веб-приложения.....	11
1.4 Анализ существующих сервисов получения грантов и постановка задачи.....	16
Глава 2 Проектная часть.....	23
2.1 Разработка архитектуры веб-приложения.....	23
2.2 Разработка модели данных ИС .....	25
2.3 Разработка логической модели.....	28
2.4 Разработка серверного приложения.....	31
2.5 Разработка интерфейса пользователя .....	36
2.6 Разработка клиентского приложения.....	41
Глава 3 Анализ работы веб-приложения .....	50
3.1 Проведение контрольного примера использования.....	50
3.2 Расчёт экономической эффективности.....	58
Заключение .....	64
Список используемых источников.....	66
Приложение А Дамп базы данных .....	68

## Введение

Целевое финансирование – это выделение средств, предназначенных для осуществления проектов целевого назначения и поступившие в проект от других лиц или из бюджета. Спрос на целевое финансирование остается неизменно высоким, что в свою очередь порождает спрос на повышение эффективности взаимодействия между сторонами финансирования.

Грантодатели стремятся к максимальному распространению информации о грантах для выполнения ключевых показателей своей деятельности.

В свою очередь соискатели финансирования находятся в процессе поиска средств для реализации и распространения своих проектов.

Реализация проектов, востребованных на рынке, является сильной стороной ООО Цифродата, однако сфера получения грантов до недавнего времени была представлена в рамках уже существующих онлайн проектов компании.

Актуальность данной выпускной квалификационной работы заключается в острой необходимости разработки веб-сервиса получения грантов, реализация которого разгрузит отдел онлайн сервисов, позволит развивать как присутствие компании в области получения грантов, так и позволит перенаправить освободившиеся в результате внедрения ресурсы на другие проекты.

Объектом исследования является отдел онлайн сервисов ООО Цифродата, который занимается разработкой и сопровождением различных онлайн сервисов компании.

В качестве предмета исследования выступает веб-сервис «Платформа получения грантов», автоматизирующий бизнес-процессы отдела онлайн сервисов.

Для достижения указанной цели необходимо изучить текущую модель бизнес-процессов в отделе онлайн сервисов, по результатам изучения выявить процессы требующие автоматизации, предложить пути оптимизации, провести сравнительный анализ присутствующих на рынке решений, отобрать лучшие практики из существующих систем и применить их в разработке, определить список технологий для разработки собственного веб-приложения, разработать реализацию и внедрить разработанное веб-приложение в отдел онлайн сервисов, а так же по итогам проведенной работы оценить экономический эффект от разработки. [2]

Выпускная квалификационная работа включает в себя введение, три главы и заключение.

Первая глава описывает процесс исследования организационной структуры и проведение анализа бизнес-процессов ООО Цифродата, в том числе отдела онлайн сервисов, поставлена задача на разработку веб-сервиса получения грантов.

Вторая глава представляет проектную часть, в которой определен список технологий для разработки и выполнено проектирование системы.

Третья глава содержит экономическое обоснование спроектированной системы и описание контрольного примера использования системы на практике.

В заключении представлены итоги выполненной работы, определена практическая значимость разработанного веб-сервиса.

## **Глава 1 Теоретическая часть**

### **1.1 Анализ предметной области ООО Цифродата**

Предметом рассмотрения в данной работе является ООО Цифродата.

Полное название: общество с ограниченной ответственностью  
Цифродата.

Основной вид деятельности: Разработка компьютерного программного обеспечения.

Компания Цифродата оказывает услуги по разработке программного обеспечения и организацией мероприятий на протяжении более чем 10 лет.

Основным профилем компании созданием инфосистем и техническое, финансовое и организационное содействие проведению мероприятий.

На данный момент компания расширила спектр своей деятельности такими услугами как разработка комплексных веб-сервисов конференций, предоставления функционала собственной платформы для больших мероприятий, таких как мероприятия от Yandex и Сколково.

Помимо веб-сервисов компания предоставляет услуги по разработке мобильных приложений и агентское сопровождение мероприятий.

Небольшие команды в департаменте разработки управляются промежуточными руководителями.

Стратегия деятельности компании состоит в непрекращающемся совершенствовании подходов, методов и реализаций в проведении мероприятий и всестороннем обеспечении как с технической, так и с финансовой и организационной сторон.

Общая организационная структура представлена на рисунке 1.



Рисунок 1 – Организационная структура органов управления

Разберем подробнее структуру отдела онлайн сервисов, представленную на рисунке 2.



Рисунок 2 – Организационная структура отдела разработки

Руководитель отдела онлайн сервисов определяет приоритеты в разработке и поддержке онлайн сервисов и услуг.

В качестве услуг можно перечислить: техническую поддержку точек проведения мероприятий, оценку и планирование показателей результатов информационных рассылок, операционное управление работой технических ресурсов онлайн сервисов, помимо этого руководитель определяет приоритет поступающим клиентским заявкам, придавая особое внимание сообщениям о дефектах подотчетных онлайн сервисов.

Специалист поддержки принимает заявки от клиентов сервисов, определяет первичный поиск решения возникших проблем, заводит задачи на решение сложных задач в систему технических заявок и дефектов. Сообщает обратившимся в поддержку о результатах своих действий и интересующей клиентов информации. Фиксирует результаты своей работы в системе учета произведенных работ.

Ведущий специалист осуществляет тестирование новой версии сервисов перед поставкой в публичный доступ. Обеспечивает публикацию информации о грантах. Принимает входящие данные от корпоративных клиентов системы проведения мероприятий в различных форматах и преобразует их в структурированные объекты системы проведения мероприятий.

Главный специалист отвечает за создание информационных, партнерские и другие виды рассылок электронных писем среди пользователей сервисов, аналитику событий онлайн сервисов, проведение онлайн мероприятий. Взаимодействует с разработчиками сервисов в целях улучшения функционала онлайн сервисов, согласует окна обслуживания сервисов с потребителями. Услуги и обязанности отдела онлайн сервисов представлены на рисунке 3.





Рисунок 3 – Перечень онлайн сервисов

Как мы можем видеть из рисунка 3 – отдел онлайн сервисов обеспечивает функционирование не только сервиса мероприятий, массовых рассылок и поддержки, но и обеспечивает бесперебойную поставку нового функционала, контроль за состоянием онлайн сервисов и публикацию грантов.

Комплексное обслуживание пользователей онлайн сервисов позволяет своевременно реагировать на поступающие запросы о расширении функционала, а компетенции специалистов в онлайн услугах и высокая связность с отделом разработки позволяют качественно составлять заявки на исправлении неточностей выполнения бизнес-требований и дефектов сервисов. [15]

## 1.2 Изучение обеспечения аппаратными и программными средствами отдела онлайн сервисов

Проведем оценку текущей информационной составляющей в деятельности организации.

Сотрудники отдела используют персональные компьютеры, которые связаны вместе в единую локальную сеть для организации рабочих процессов.

Характеристики рабочих станций:

- операционная система Windows 10,
- компьютер с четырехъядерным процессором от 2 ГГц,
- оперативная память 8 Гб,
- веб-браузер Chrome,
- монитор,
- доступ в сеть интернет,
- клавиатура и мышь.

Таблица 1 – Анализ аппаратного и программного обеспечения в организации

Техническое и программное обеспечение	Требует обновления
Компьютер с процессором от 2 ГГц	нет
Оперативная память не менее 8 Гб	то же
Монитор	»
Принтер	»
Клавиатура и мышь	»
Веб камера	»
Операционная система Windows 10	»
Веб-браузер Google Chrome	»

Как мы можем видеть из таблицы 1, сотрудники обеспечены современным производительным персональным компьютером, обеспечивающим удобную работу с онлайн сервисами и клиентами.

### 1.3 Анализ бизнес-процессов отдела онлайн сервисов и обоснование необходимости в разработке веб-приложения

Используем стандарт IDEF0 для проведения анализа деятельности отдела онлайн сервисов. Стандарт IDEF0 предназначен для создания детальной и точной функциональной модели системы, которая полностью отображает ее структуру и функции. [7]

Эта модель включает в себя определение потоков информации и материальных объектов, которые связывают все функции системы между собой. Точка зрения: ведущий специалист отдела онлайн сервисов. Цель: анализ деятельности отдела онлайн сервисов в области публикации грантов.

Диаграмма представлена на рисунке 4.

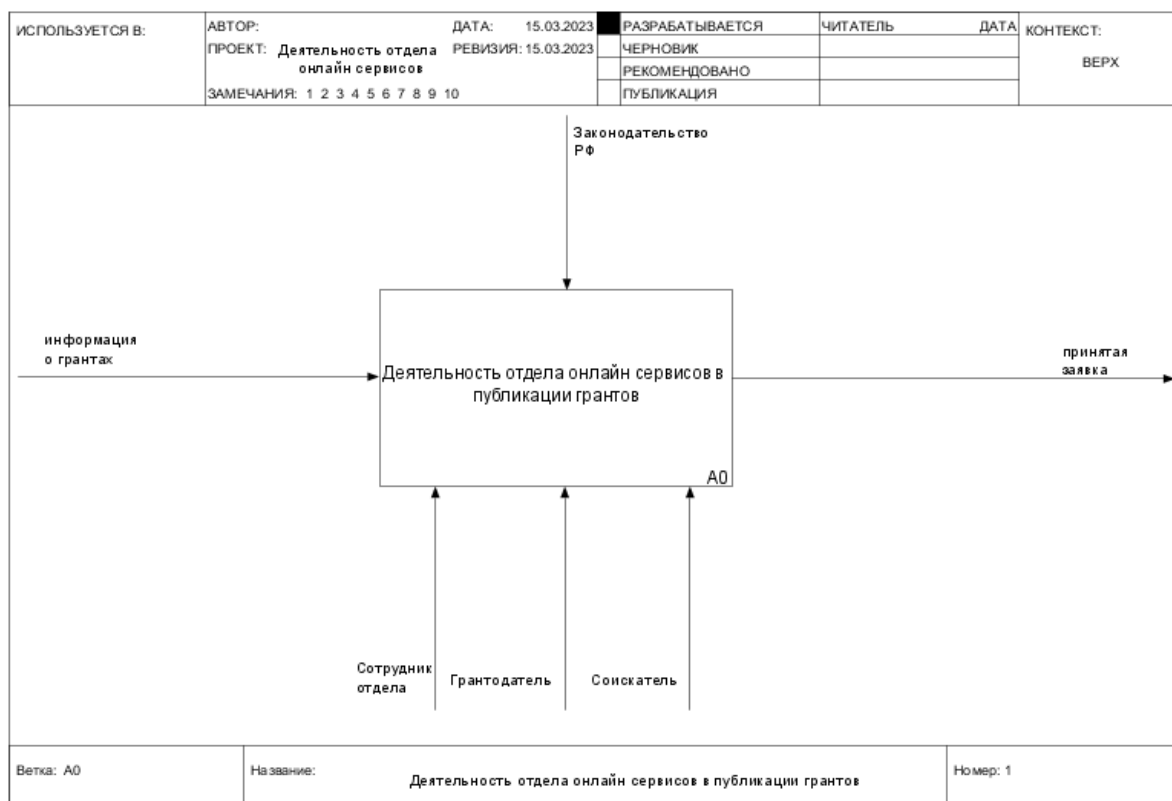


Рисунок 4 – Контекстная диаграмма деятельности отдела онлайн сервисов в публикации грантов («как есть»)

Проведем декомпозицию верхнего уровня процесса публикации гранта (рисунок 5).

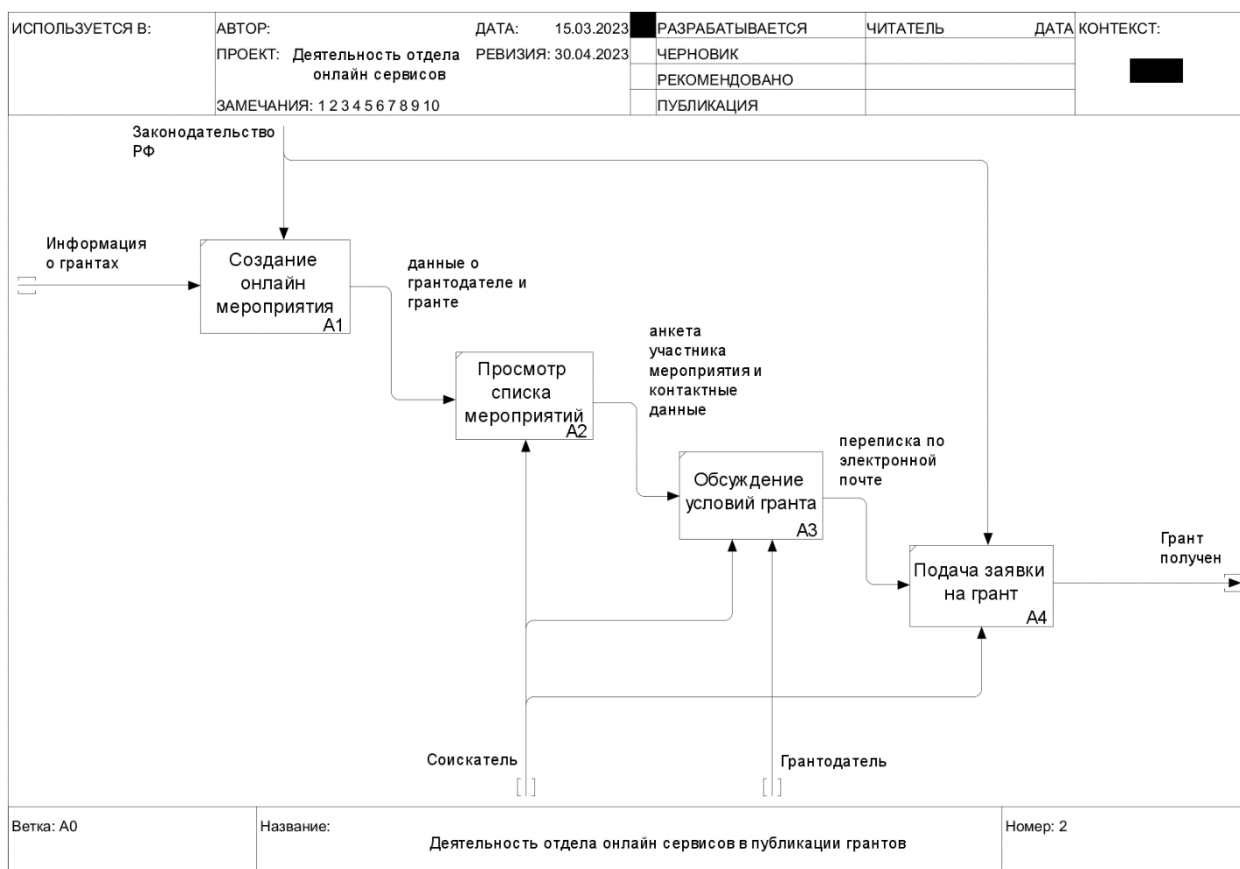


Рисунок 5 – Декомпозиция верхнего уровня деятельности отдела онлайн сервисов в публикации грантов («как есть»)

Проведя анализ контекстной диаграммы и диаграммы с декомпозицией верхнего уровня, можно сделать следующие выводы: персонал осуществляет публикацию информации о грантах через механику проведения онлайн мероприятий, предоставитель грантов получает информацию о соискантах в виде опросов привязанных к регистрации на мероприятия. [7]

Таким образом в текущем виде функциональность публикации грантов, поиска грантов и подачи заявок на получение гранта сильно

ограничена, как в сфере используемой в данный момент функциональности, так и в потенциале развития функциональности конфликтующей с механикой проведения онлайн мероприятий.

Исходя из данного заключения можно определить, что существующая система, выстроенная при работе ведущего специалиста в значительной степени, не реализует необходимый функционал в предметной области, подвержен ограничениям используемой функциональностью проведения мероприятий, в результате чего является неэффективной и требующей улучшений.

Контекстная диаграмма и диаграмма с декомпозицией верхнего уровня в полной мере описывают недостатки данных процессов и позволяют сделать однозначные выводы. В следствие чего декомпозиции нижнего уровня не требуется.

Всё рассмотренное ранее позволяет получить вывод о том, что для публикации грантов требуется специализированный сервис позволяющий искать гранты со специальными фильтрами, получать заявки от соискателей, и далее расширять функциональность, не подвергая изменениям другие существующие сервисы.

Для устранения недостатков в уже существующих бизнес-процессах необходимо провести моделирование контекстной диаграммы деятельности отдела онлайн-сервисов по типу «как должно быть» (Рисунок 6). Из полученной диаграммы мы видим, что необходимое покрытие функциональности поддержки грантов возьмет на себя сервис грантов.

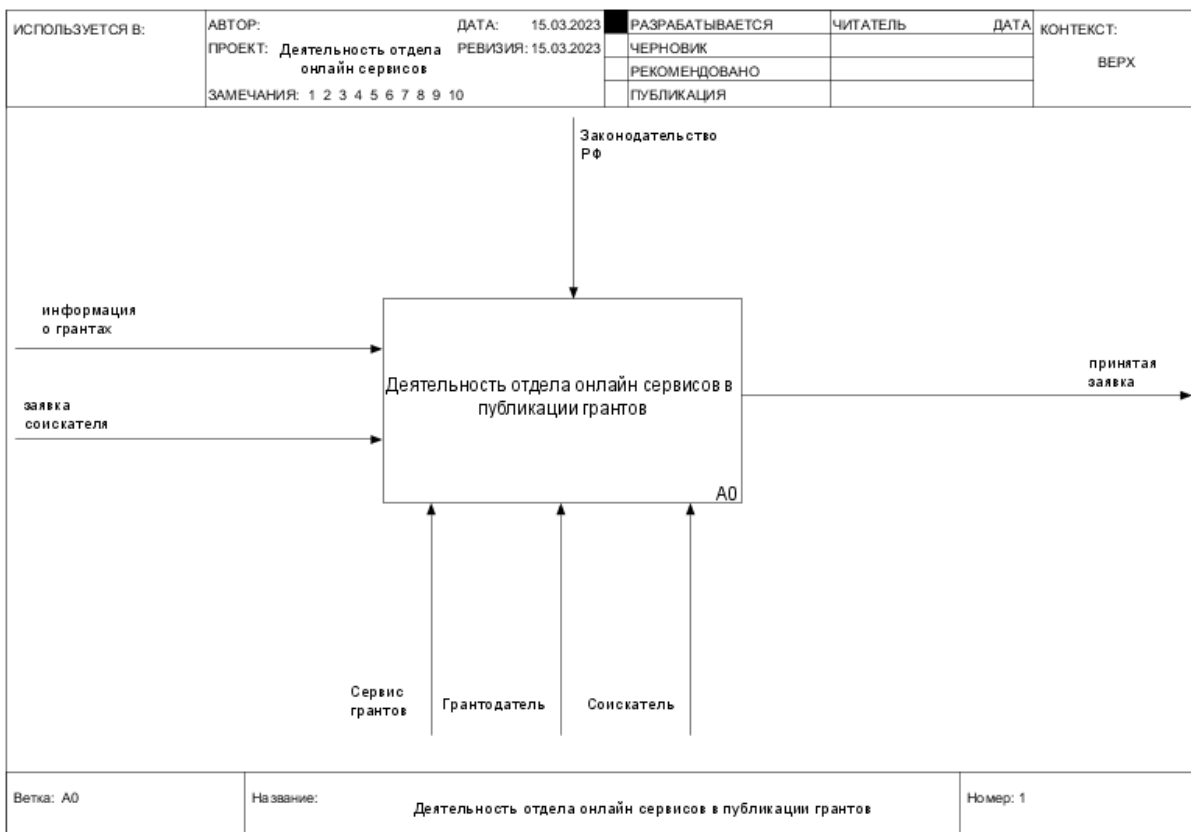


Рисунок 6 – Контекстная диаграмма деятельности отдела онлайн сервисов в публикации грантов («как должно быть»)

Исходя из рисунка 6 можно определить, что процесс публикации грантов должен быть оптимизирован в рамках участия сотрудников отдела, путем непосредственной коммуникации участников процесса через сервис грантов.

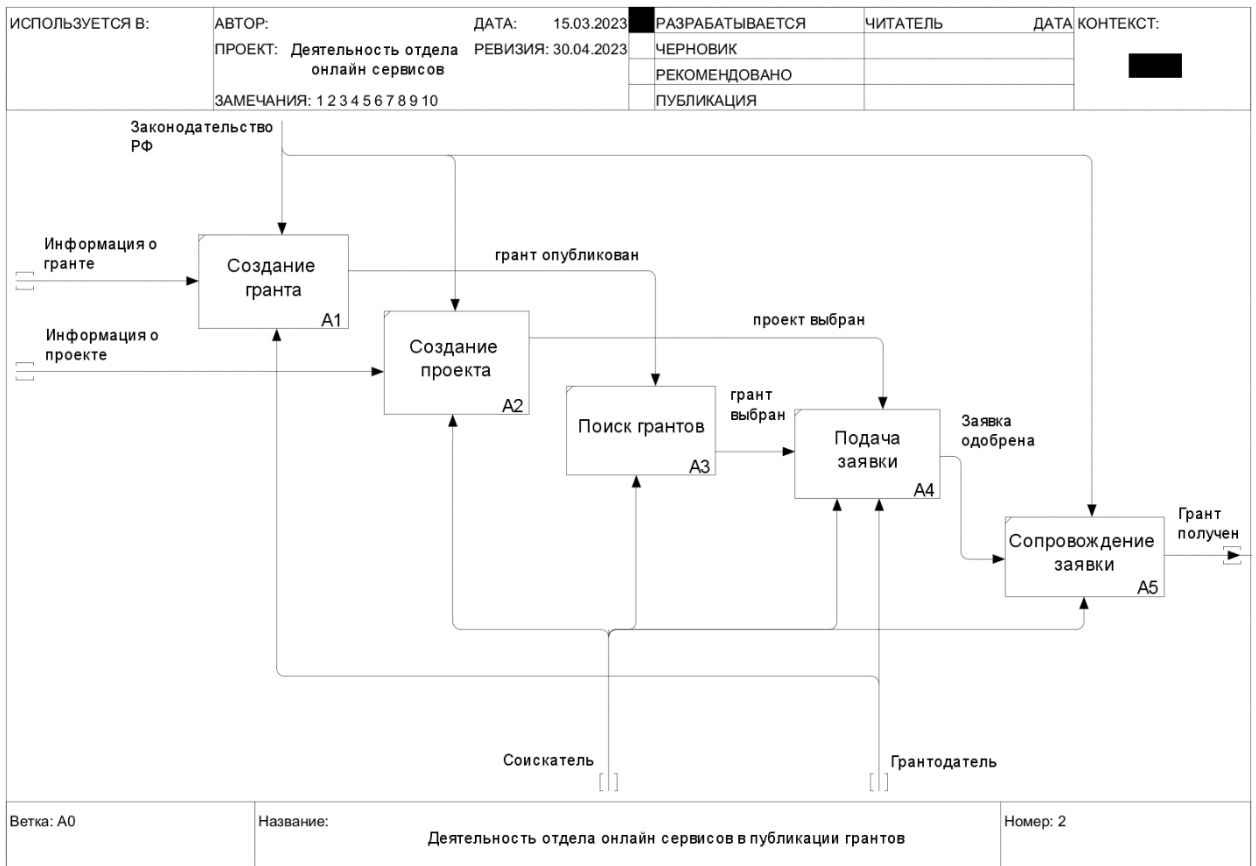


Рисунок 7 – Декомпозиция верхнего уровня деятельности отдела онлайн сервисов («как должно быть»)

В новой диаграмме деятельности отдела (рисунок 7), мы можем видеть сокращение участия сотрудников отдела в публикации и поддержке грантов, что позволяет оптимизировать нагрузку на отдел и сократить количество задействованного персонала.

## **1.4 Анализ существующих сервисов получения грантов и постановка задачи**

Проведение анализа существующих сервисов по получению грантов позволит выявить их преимущества и недостатки. Данный анализ позволит определить сильные стороны существующих сервисов, а также их слабые места, что в свою очередь позволит разработать наиболее эффективное и востребованное веб-приложение получения грантов.

Качественные исследования помогают нам понять предметную область, контекст и ограничения продукта – причем иным, более действенным способом, чем количественные исследования. [9]

Среди крупных систем поддержки грантов можно выделить такие сервисы как «Культура. Гранты России» [grants.culture.ru](http://grants.culture.ru), «Конференции.Новости.События» [www.rsci.ru/grants](http://www.rsci.ru/grants), а также «Минобрнауки России» [minobrnauki.gov.ru](http://minobrnauki.gov.ru).

Использование существующих систем не может полностью удовлетворить необходимость в функциональности плотно интегрированной в онлайн сервисы ООО Цифродата, однако, рассмотрев подробнее перечисленные выше сервисы мы можем определить ключевую функциональность присущую сервисам поддержки грантов.

«Культура. Гранты России» – данный ресурс представляет общедоступную базу грантов и конкурсов в области культуры и искусства.

В первую очередь стоит отметить широкую функциональность фильтрации грантов в разделе расширенного поиска, мы можем отфильтровать гранты по следующим особенностям:

- фразе в описании,
- направлении финансирования,
- периоде приема заявок,



- наименованию и типу грантодателя,
- финансовому источнику,
- региону,
- категории соискателей,
- сумме гранта,
- актуальности гранта.

Интерфейс сервиса «Культура. Гранты России» представлен на рисунке 8.

Культура. Гранты России

Общероссийская база конкурсов и грантов в области культуры и искусства

Вход / Регистрация

О проекте Гранты Грантодатели Грантрайтерам Новости

Поиск по сайту Регион: Все регионы Направление: Все варианты Расширенный поиск

Введите фразу

Тип конкурса: Все варианты, Грант, Конкурс, Премия, Субсидия

Направление: Все варианты, Архитектура, Библиотеки, Волонтеры, Детское творчество

Период приема заявок: 04.03.2023 - 04.03.2023

Наименование грантодателя: Все варианты, "МОСКОВСКОЕ АГЕНТСТВО РЕА. АДМИНИСТРАЦИЯ ГОРОДА СМ... АНО "Азбука Музыки" АНО "ГАЛА-РУС"

Финансовый источник: Все варианты, Федеральные средства, Региональные средства, Муниципальные средства, Частные средства

Ограничение по региону: Все регионы, Алтайский край, Амурская область, Архангельская область, Астраханская область

Тип грантодателя: Органы государственной власти, Органы исполнительной власти, Органы местного самоуправления, Федеральный орган исполнител, Юридические лица

Категория соискателей: Все варианты, Физические лица, Индивидуальные предпринима, Коммерческие организации, Некоммерческие организации

Сумма гранта:

Показать только открытые конкурсы  Требование по стороннему софинансированию проекта

Рисунок 8 – Внешний вид расширенного поиска сервиса «Культура. Гранты России»

В сервисе «Культура. Гранты России» общий список грантов представлен в виде названий грантов.

При просмотре можно увидеть поля позволяющие узнать подробную информацию о гранте, следует отметить возможность указания большого количества направлений, указания ссылок в описании гранта, прямые контактные данные организатора.

«Конференции.Новости.События» – данный web-сервис позволяет найти гранты по категории и ключевому слову, сервис имеет минимальный функционал. Помимо поиска грантов сервис предоставляет только возможность ознакомиться с описанием гранта, что уступает в функциональности текущей реализации в онлайн сервисах ООО Цифродата.

Интерфейс сервиса «Конференции.Новости.События» показан на рисунке 9.

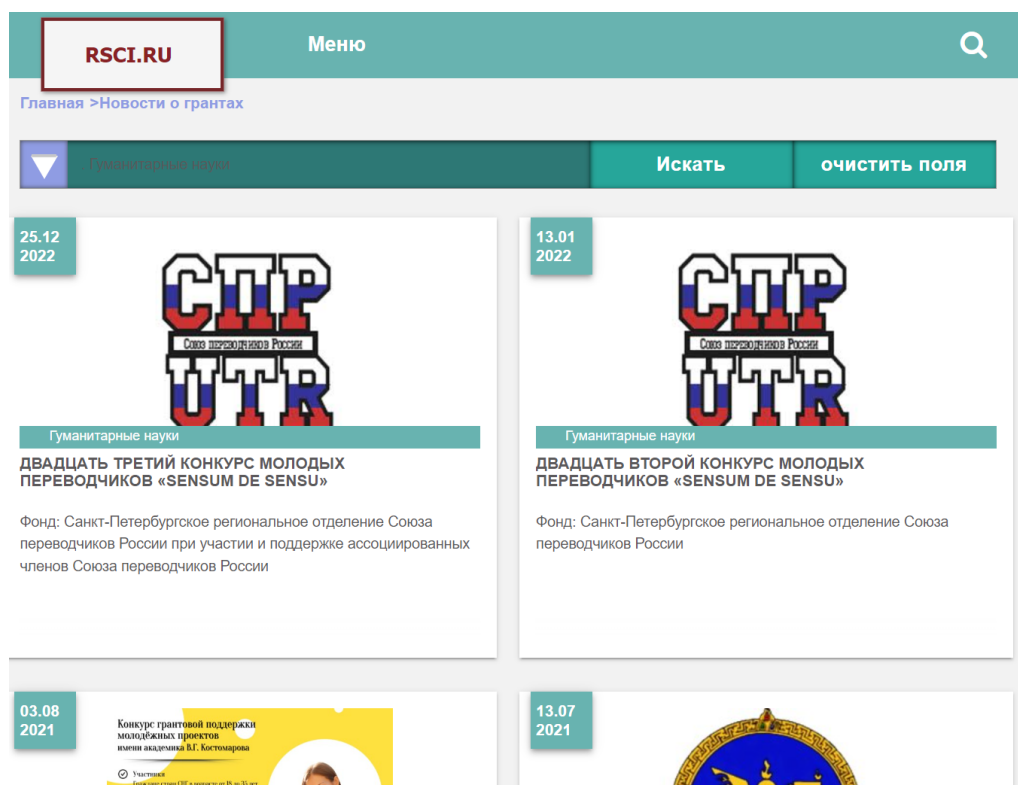


Рисунок 9 – Внешний вид сервиса Конференции.Новости.События

В целом сервис позволяет оценить упрощение пользовательского интерфейса, позволяющее легче воспринимать информацию.

Сайт грантов Минобрнауки России – сервис, предоставляемый министерством образования и науки Российской Федерации выполнен в строгом стиле и позволяет искать гранты по названию, актуальности и датам публикации. Функциональность сервиса ограничена предоставлением общего описания, ссылкам на конкурсную документацию и указанием ссылок на внешние ресурсы. Сервис является по большей части информационным, и не позволяет интерактивно подавать заявки на гранты, и сопровождать процесс выдачи гранта.

На рисунке 10 отображен внешний вид сервиса грантов Минобрнауки России.

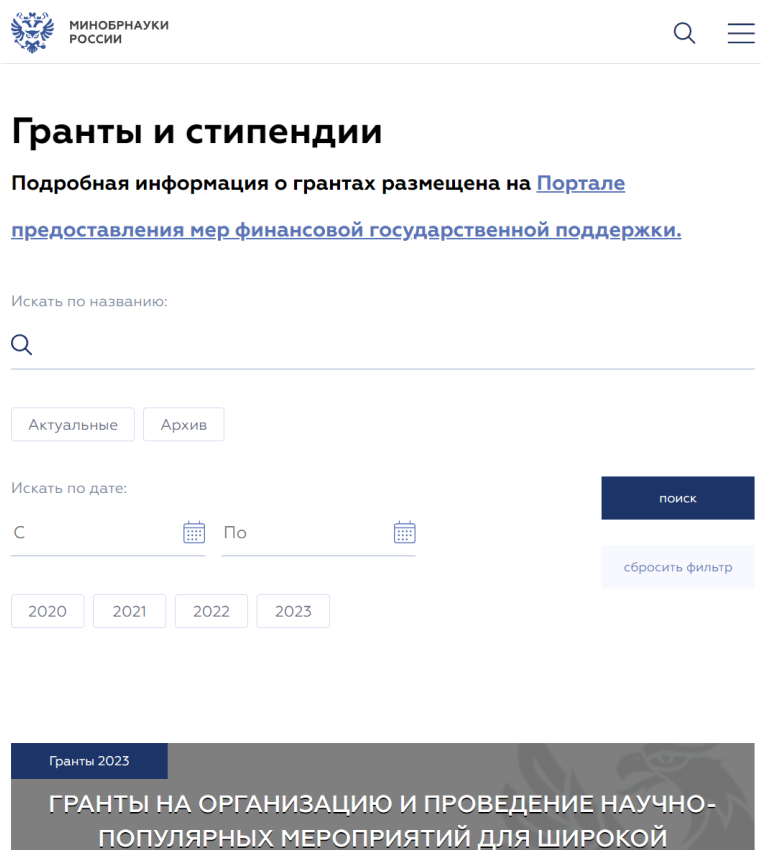


Рисунок 10 – Поиск грантов на сервисе «Минобрнауки России»

Составим сравнительную таблицу для рассмотренных веб-сервисов (Таблица 2), где цифрой обозначим проработанность конкретных разделов.

Таблица 2 – Сравнительный анализ веб-сервисов получения грантов

Критерий	Культура. Гранты России	Конференции. Новости. События	Гранты Минобрнауки России
Поиск грантов	5	2	2
Просмотр гранта	5	3	4
Указание контактных данных	3	3	3
Работа с файлами	1	1	1
Доступный интерфейс	2	3	5
Итого	16	12	16

Исходя из обзора трех существующих систем поддержки грантов, мы определили распространенную функциональность поиска, и полностью отсутствующую интерактивность подачи заявок и взаимодействия грантодателя и соискателя в существующих онлайн сервисах, что подтверждает необходимость реализации собственного сервиса поддержки грантов.

Рассматривая функциональность систем, мы можем видеть нереализованную потребность в расширении существующей функциональности; предоставлении не только поиска и выдачи информации о гранте, но также управление заявками и обмен сообщениями между участвующими сторонами, в том числе с прикрепленными файлами.

Особенности существующих решений говорят о популярности минималистичного дизайна в данной сфере, поэтому логично использовать данный подход по дизайну в качестве основы.

Разрабатываемое веб-приложение должно предоставлять пользователям следующий функционал:

- создание и внесение изменений в карточки грантов в том числе указание названия, описания, лимитов бюджетирования, периоде актуальности, категорий и регионов проектов, попадающих под цели финансирования;
- удаление созданных карточек грантов создателем,
- поиск опубликованных грантов с фильтрацией по категориям, регионам, бюджетам и датам проведения,
- подача соискателями заявок на участие в гранте,
- регистрация пользователей,
- аутентификация пользователей,
- просмотр и модерация заявок на гранты от соискателей грантодателем,
- управление информацией своей аккаунта пользователем,
- просмотр списка своих грантов грантодателем,
- ведение одобренной заявки со стороны как грантодателя, так и соискателя, в виде обмена сообщениями между грантодателем, с возможностью прикрепления файлов.

Для реализации описанного функционала в краткие сроки, для простоты последующей поддержки, а также в целях совместимости с современными трендами развития web-индустрии, список технических требований можно представить в виде следующих утверждений:

- разработанное приложение должен состоять из двух компонентов серверной части и клиентской части обменивающихся информацией посредством программного интерфейса REST с передачей данных путем http запросов;

- серверная часть приложения должна быть написана на простом в изучении, производительном, компилируемом языке `golang`, в целях экономии технических ресурсов;
- в целях экономии ресурса разработки и унификации необходимо применить фреймворк `goswagger` в качестве реализации `http` сервера, данный фреймворк позволяет быстро и эффективно разрабатывать веб-приложения, предоставляющие программный интерфейс описанный на языке `YAML`;
- для уровня абстракции базы данных необходимо использовать библиотеку `gorm`, для простоты хранения данных системы;
- клиентскую часть необходимо написать на фреймворке `Vuetify`, версии 3, предоставляющего набор готовых динамических компонентов графического интерфейса для веб-страниц, что позволяет оптимизировать расходы на проработку пользовательских интерфейсов.

#### Вывод по главе 1

Первая глава был выполнен анализ структуры организации ООО Цифродата, а также исследование бизнес-процессов отдела онлайн-сервисов.

После проведения анализа были выявлены недостатки в текущих бизнес-процессах, построены контекстные диаграммы `IDEFO` «как есть» и «как должно быть».

Завершающим этапом был выполнен сравнительный анализ существующих веб-сервисов получения грантов, что позволило определить сильные и слабые стороны существующих сервисов, что позволило выявить наиболее востребованную функциональность веб-приложений получения грантов.

## Глава 2 Проектная часть

### 2.1 Разработка архитектуры веб-приложения

Архитектура обеспечивает разделение ответственности, позволяя разрабатывать приложение взаимодействуя только с теми частями, которые необходимы для выполнения заданной функции. [14]

Для создания программной архитектуры сервиса поддержки грантов построим диаграмму компонентов. Диаграммы компонентов состоят из набора компонентов, интерфейсов и зависимостей, обобщений, ассоциаций и отношений реализации. Диаграммы компонентов UML имеют важную особенность, заключающуюся в повторно используемых компонентах, эта особенность, особенно актуальна для разработки сложных приложений. [12]

Пользователи подключается к сервису с помощью браузера (Chrome, Edge, Mozilla, Opera и др.). Программная архитектура веб-приложения представлена на рисунке 11.

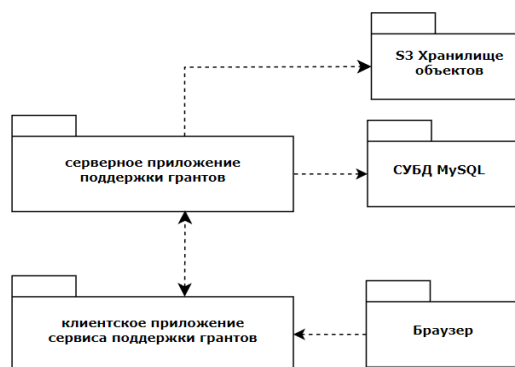


Рисунок 11 – Программная архитектура веб-приложения

Для реализации веб-приложения использована трехуровневая архитектура «клиент-сервер». [4]

Целью архитектуры клиент/сервер является максимально полное использование информационных ресурсов и совместная работа с различными устройствами. [11]

Для представления архитектуры веб-приложения используем диаграмму развертывания UML.

Для описания развертывания веб-приложения необходимо представить на диаграмме:

- аппаратные компоненты
- программные компоненты
- способы коммуникации всех компонентов диаграммы

Диаграмма развертывания веб-приложения представлена на рисунке 12.

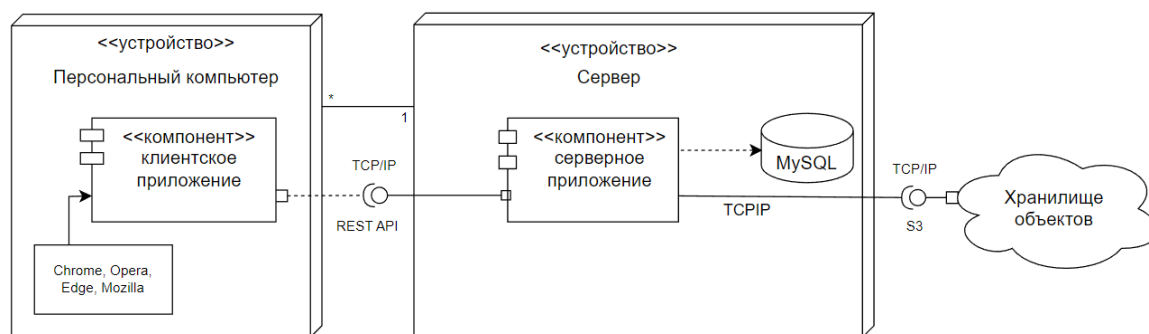


Рисунок 12 – Диаграмма развертывания веб-приложения сопровождения грантов

Для развертывания ИС необходим облачный хостинг, обеспечивающий следующие требования:

- управляемый хостингом оркестратор для развертывания контейнеров docker,



- обеспечение высокой доступности путем обеспечения трех различных зон обслуживания для параллельного развертывания контейнеров,
- управляемый хостингом сервис MySQL версии 5.7 или выше,
- хранилище объектов совместимое с протоколом S3.

В программной архитектуре принимаем во внимание, стек технологий: язык программирования golang, реляционная база данных MySQL, использование фреймворков VUE JS, go-swagger, gorm.

## 2.2 Разработка модели данных веб-приложения

Бизнес сущности, используемые в проекте, будут представлены таблицами базы данных. Сущности, в контексте технологического процесса работы проекта, могут включать в себя процессы, абстракции, объекты и действия. Каждая сущность должна содержать атрибуты, которые представляют физические параметры и характеристики, относящиеся к этой сущности. [6]

Выделим сущности, участвующие в процессе работы. (Таблица 3).

Таблица 3 – Сущности базы данных

Сущность	Назначение
Пользователи	Данные о пользователях
Проекты	Данные о проектах
Гранты	Данные о грантах
Заявки	Данные о заявках
Сообщения	Данные о сообщениях
Файлы	Данные о файлах пользователей
Категория	Данные о категориях
Регион	Данные о регионах
Этапы проектов	Данные об этапах проектов

Между сущностями установлены следующие связи:

- пользователи-проекты – «один ко многим»,
- пользователи-гранты – «один ко многим»,
- пользователи-файлы – «один ко многим»,
- пользователи-сообщения – «один ко многим»,
- проекты-заявки – «один ко многим»,
- гранты-заявки – «один ко многим»,
- заявки-сообщения – «один ко многим»,
- сообщения-файлы – «один ко многим»,
- проекты-категории – «многие ко многим»,
- проекты-регионы – «многие ко многим»,
- категории-гранты – «один ко многим»,
- регионы-гранты – «многие ко многим».

Для обеспечения целостности базы данных использованы первичные и внешние ключи.

Целостностью данных, хранимых в СУБД, называется их корректность и непротиворечивость. [3]

Сущностям необходимо назначить атрибуты, характеризующие каждую из них.

Результатом выполнения набора SQL команд (Приложение А) является база данных, структура которой представлена на рисунке 13.

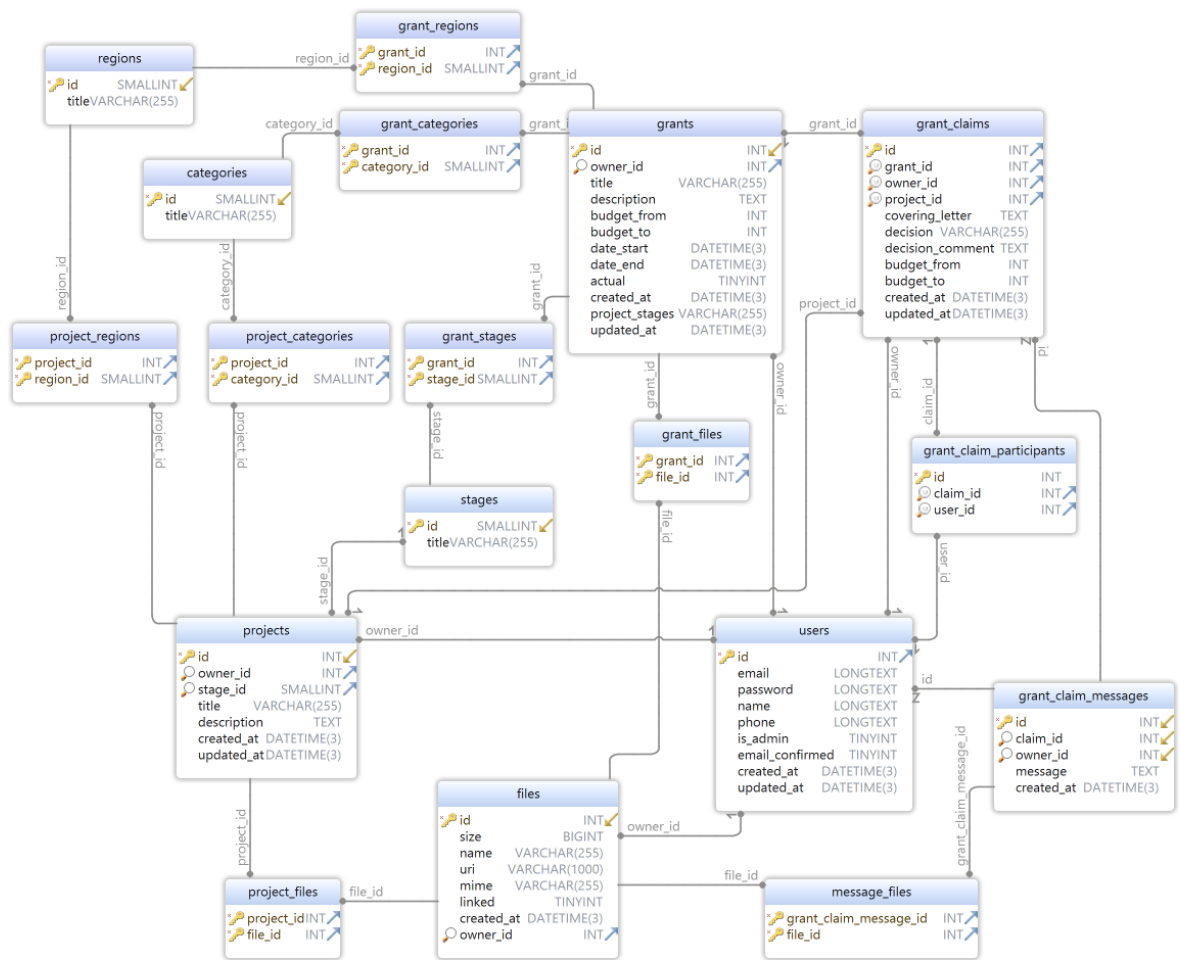


Рисунок 13 – Структура базы данных

База данных веб-приложения представлена в общей сложности 18 таблицами, функционирующих на основе подсистемы InnoDB которая в свою очередь обеспечивает механизм транзакций и внешних ключей в СУБД MySQL.

Данные таблиц соответствуют нормальным формам реляционных БД.

Для каждой таблицы соблюдается первая нормальная форма (1НФ). Это обеспечено путем хранения всех полей таблицы в простом (неделимом) виде, отсутствием повторяющихся полей и хранением в каждом поле только одного значения, которое не является списком или множеством значений.

Таблицы приведены ко второй нормальной форме (2НФ), что обеспечивается нахождением в 1НФ, обладанием простыми первичными ключами и зависимостью любого не ключевого поля от первичного ключа (составные ключи не допускаются). Третья нормальная форма соблюдена, обеспечением соответствия второй нормальной форме и отсутствием зависимости между не ключевыми полями.

### 2.3 Разработка логической модели

Требуется определить рабочие процессы и информационные потоки, которые необходимы для выполнения поставленных задач. События и потоки представлены в таблице (Таблица 4).

Таблица 4 – События и потоки проектируемой ИС

Действующее лицо	Событие	Результат
Неавторизованный пользователь	Авторизация	Вход в приложение
	Регистрация в сервисе	Создан новый пользователь
Авторизованный пользователь	Подтверждение адреса электронной почты	Пользователь может авторизоваться в сервисе
	Создание нового гранта	В сервис добавлен новый грант
	Корректировка данных гранта	Данные по сотруднику скорректированы
	Удаление гранта	Грант удален
	Просмотр списка своих грантов	Просмотр всех грантов, созданных пользователем
	Просмотр списка заявок, на грант созданный пользователем	Просмотр всех поступивших заявок на грант
	Поиск грантов в публичном каталоге	Просмотр грантов, созданных пользователями сервиса
	Фильтрация результатов поиска грантов в публичном каталоге	Просмотр отфильтрованных по различным критериям грантов

Продолжение таблицы 4

Авторизованный пользователь	Просмотр информации о гранте, найденном в публичном каталоге	Просмотр информации о гранте
	Решение по заявке на грант	Изменение статуса заявки на грант
	Создание нового проекта	В сервис добавлен новый проект
	Просмотр списка своих проектов	Просмотр проектов, созданных пользователем
	Удаление проекта	Проект удален
	Создание заявки на грант	Заявка на грант создана
	Загрузка файла	Файл загружен на объектное хранилище
	Отправка сообщения при поддержке заявки на грант	В сервис добавлено новое сообщение
Администратор	Просмотр списка пользователей	Просмотр списка пользователей
	Блокировка пользователя	Пользователь заблокирован

Для категоризации функционала веб-приложения, требуется создать диаграмму вариантов использования (прецедентов). Данная диаграмма представляет отображение основных сценариев использования веб-приложения и демонстрирует действия пользователя в различных ролях (Рисунок 14).

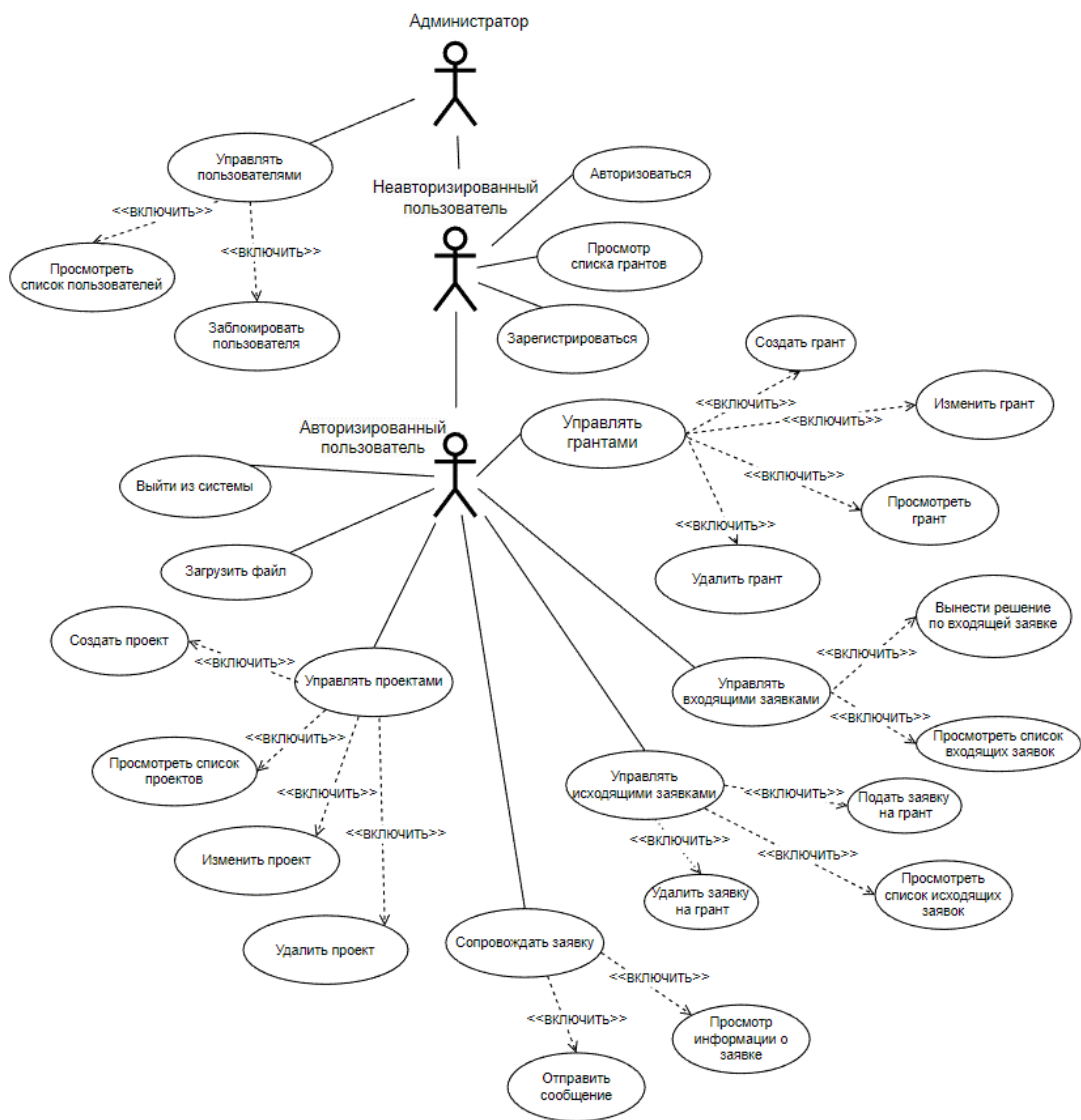


Рисунок 14 – Диаграмма вариантов использования системы

Для получения наглядной схемы переходов между состояниями грантов построим диаграммы переходов состояний.

Переходы между состояниями публикации и получения грантов представлены на рисунках 15 и 16.

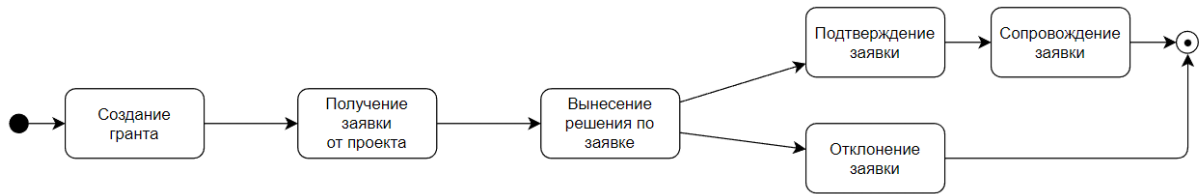


Рисунок 15 – Диаграмма переходов состояний публикации гранта

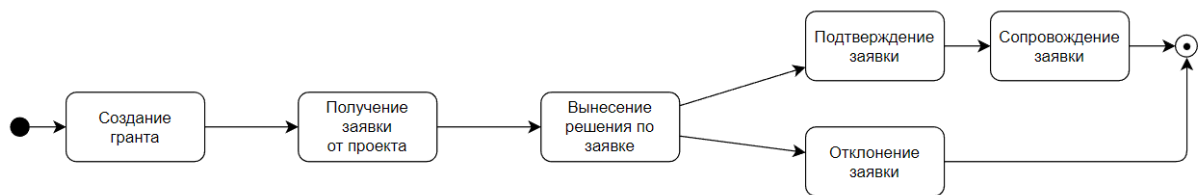


Рисунок 16 – Диаграмма переходов состояний получения гранта

Как мы можем видеть из вышеуказанных диаграмм грантодатель рассматривает информацию о проекте, подавшем заявку на получение гранта, и выносит решение, которое в свою очередь определяет происходит ли переход к состоянию поддержки заявки на получение гранта или соискатель, получив отказ начинает поиск нового гранта для своего проекта.

## 2.4 Разработка серверного приложения

Серверное приложение разработано на языке Go. Язык Go является кроссплатформенным языком с сильной поддержкой многопоточности, сборщиком мусора и другими атрибутами современного высокоуровневого языка программирования. [17]

В качестве основы серверного приложения был использован фреймворк Goswagger. Данный фреймворк позволяет описать программный интерфейс для взаимодействия клиентского приложения и серверного приложения на декларативном языке YAML, используя стандарт Swagger версии 2.

На рисунке 17 приведен пример описания языком YAML метода вынесения решения по входящей заявке на получение гранта.

```
/claims/{id}/decision:
  put:
    tags: [ claims ]
    summary: 'Вынесение решения по входящей заявке на получение гранта'
    operationId: GrantClaimDecision
    security:
      - AuthScope: [ user ]
    parameters:
      - in: path
        name: id
        required: true
        description: Идентификатор заявки
        type: integer
      - in: body
        name: body
        required: true
        schema:
          type: object
          properties:
            comment:
              type: string
              description: Комментарий
            decision:
              type: bool
              description: Одобрено
    responses:
      200:
        description: Ок
      404:
        description: Не найдено
        schema:
          $ref: '#/definitions/Error'
      500:
        description: Внутренняя ошибка
        schema:
          $ref: '#/definitions/Error'
      422:
        description: Невозможно обработать запрос
        schema:
          $ref: '#/definitions/Error'
```

Рисунок 17 – Пример описания метода языком YAML

После описания методов программного интерфейса необходимо описать обработчик для каждого метода на языке Go. [5]

На рисунке 18 отображен обработчик вынесения решения.



```

type GrantClaimDecisionHandler struct {
    claimService *cl.Service
}
func NewGrantClaimDecisionHandler(clServ *cl.Service ) *GrantClaimDecisionHandler {
    return &GrantClaimDecisionHandler{
        |   claimService: clServ,
        |   }
    }
func (g *GrantClaimDecisionHandler) Handle(params claims.GrantClaimDecisionParams,
principal *models.Principal) middleware.Responder {
    exClaim, err := g.claimService.GetByID(uint32(params.ID))
    if err != nil {
        |   return claims.NewGrantClaimDecisionInternalServerError().
        |   |   WithPayload(utils.MapError(err))
        |   }
    if exClaim == nil
        |   || exClaim.Grant.OwnerID != uint32(principal.AccountID) {
        |   return claims.NewGrantClaimDecisionNotFound().WithPayload(
        |   |   |   utils.MapError(fmt.Errorf("заявка не найдена")))
        |   }
    decision := "approved"
    if !params.Body.Decision {
        |   decision = "declined"
        |   }
    newClaim := cl.GrantClaim{
        |   Decision:      decision,
        |   DecisionComment: params.Body.Comment,
        |   }
    err := g.claimService.Update(exClaim.ID, newClaim)
    if err != nil {
        |   return claims.NewGrantClaimDecisionInternalServerError().
        |   |   WithPayload(utils.MapError(err))
        |   }
    return claims.NewGrantClaimDecisionOK()
}

```

Рисунок 18 – Обработчик вынесения решения

В обработчик передается инициализированные сервис. Метод Handle принимает параметры запроса, декодированные из формата JSON в структуру в памяти приложения. Если заявка найдена и грант, связанный с заявкой, принадлежит текущему пользователю, заявка обновляется с записью вынесенного решения и комментария к нему. В противном случае возвращается один из вариантов ответа об ошибке.

На рисунке 19 мы видим пример инициализации обработчиков запросов при старте серверного приложения.

```
api.HealthHealthCheckHandler = handlers.NewHealthHandler()
api.GrantsGrantCreateHandler = handlers.NewGrantCreateHandler(grantService, catServ, regionServ, stageServ, validate)
api.GrantsGrantUpdateHandler = handlers.NewGrantUpdateHandler(grantService, catServ, regionServ, stageServ, validate)
api.GrantsGrantDeleteHandler = handlers.NewGrantDeleteHandler(grantService)
api.GrantsListGrantsHandler = handlers.NewListGrantsHandler(grantService, validate)
api.ClaimsGrantClaimCreateHandler = handlers.NewCreateGrantClaimsHandler(claimService, grantService, projectServ)
api.ClaimsGrantClaimDeleteHandler = handlers.NewDeleteGrantClaimsHandler(claimService, grantService)
api.ClaimsGrantClaimUpdateHandler = handlers.NewUpdateGrantClaimsHandler(claimService, grantService, projectServ)
api.ClaimsListGrantClaimHandler = handlers.NewListGrantClaimsHandler(claimService, grantService)
api.ClaimsGrantClaimDecisionHandler = handlers.NewGrantClaimDecisionHandler(claimService, grantService)
api.UserConfirmEmailHandler = handlers.NewConfirmEmailImplHandler(confirmService, userService, authService)
api.UserSignupHandler = handlers.NewSignupHandler(userService, validate)
api.UserSignInHandler = handlers.NewSignInHandler(userService, authService, validate)
api.ProjectsProjectListHandler = handlers.NewProjectsListHandler(projectServ)
api.ProjectsProjectCreateHandler = handlers.NewProjectCreateHandler(projectServ, fileServ, catServ, regionServ, stageServ)
api.ProjectsProjectUpdateHandler = handlers.NewProjectUpdateHandler(projectServ, fileServ, catServ, regionServ, stageServ)
api.ProjectsProjectDeleteHandler = handlers.NewProjectDeleteHandler(projectServ, fileServ)
api.MessagesMessageListHandler = handlers.NewMessagesListHandler(claimMessageService, claimService)
api.MessagesMessageCreateHandler = handlers.NewMessagesCreateHandler(claimMessageService, claimService, fileServ, validate)
api.MessagesMessageDeleteHandler = handlers.NewMessagesDeleteHandler(claimMessageService, claimService, fileServ)
```

Рисунок 19 – Пример инициализации обработчиков запросов

При инициализации обработчиков запросов, интерфейсу API назначаются обработчики, соответствующие реализующие описанные в формате OpenAPI соглашениям. В качестве параметров в конструкторы обработчиков передаются сервисы, реализующие слой бизнес-логики, таким образом приложение разделяется на слои:

- слой программного интерфейса предоставляющий доступ к функционалу проекта, обрабатывающий HTTP запросы, кодирующий и декодирующий сообщения в соответствии с соглашением, описанным в формате OpenAPI;
- слой бизнес-логики, реализующий обработку полученных и декодированным слоем API сообщений;
- слой абстракции базы данных, обеспеченный библиотекой gorm, предоставляет доступ к объектам базы данных по технологии ORM.

На рисунке 20 мы можем увидеть общую структуру проекта серверного приложения.

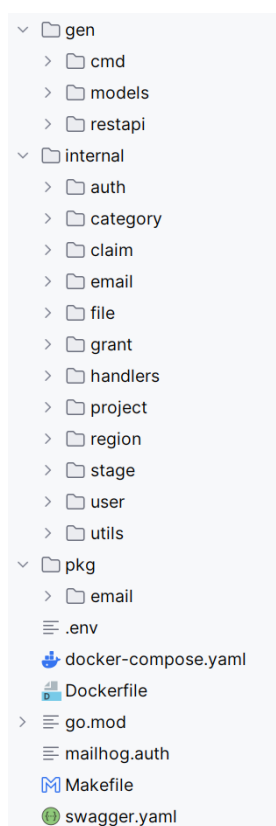


Рисунок 20 – структура проекта серверного приложения

Код приложения разделен на пакеты в целях изолирования компонентов приложения, простоты сопровождения, соответствия общепринятым стандартам.

Приложение на уровне коннектора базы данных использует SQL запросы для работы с реляционной моделью базы данных MySQL. [1]

## 2.5 Разработка интерфейса пользователя

В целях унификации и упрощения разработки для разработки интерфейса конечного пользователя используем фреймворк для разработки веб-интерфейсов Vuetify, предлагающий большое количество готовых компонентов и изначально разработанный для интеграции с фреймворком клиентских приложений VUE. [18]

На рисунке 21 мы можем увидеть интерфейс поиска грантов.

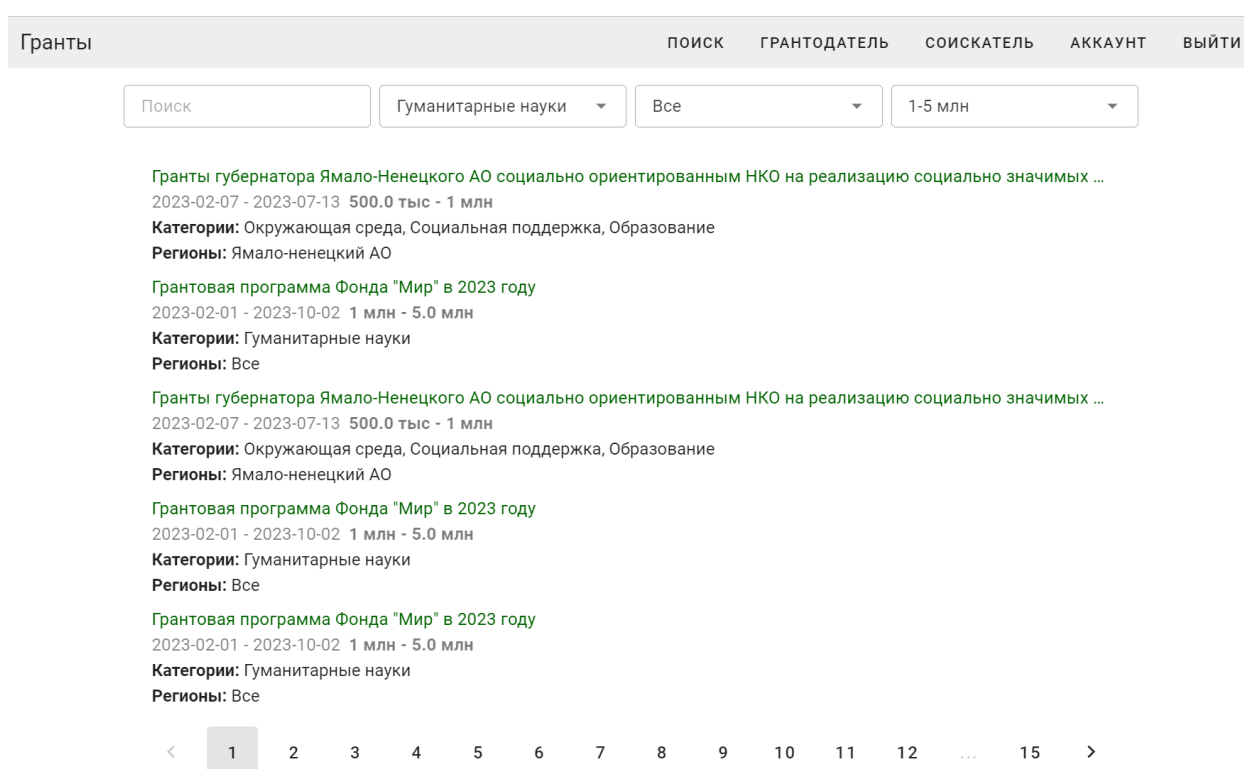


Рисунок 21 – Интерфейс поиска грантов

Для управления своими данными пользователем разработаем интерфейс управления аккаунтом, представленный на рисунке 22.

Гранты ПОИСК   ГРАНТОДАТЕЛЬ   СОИСКАТЕЛЬ   АККАУНТ   ВЫЙТИ

**Управление аккаунтом**

Имя  
ООО Рувентс

Email  
test@email.com

Телефон  
79527776655

Организация  
ООО Рувентс

Инн  
1234567890

Кпп  
123456789

Пароль

[СОХРАНИТЬ](#)

Рисунок 22 – Интерфейс управления аккаунтом пользователя

Интерфейс просмотра списка входящих заявок показан на рисунке 23.

Гранты ПОИСК   ГРАНТОДАТЕЛЬ   СОИСКАТЕЛЬ   АККАУНТ   ВЫЙТИ

ГРАНТЫ   ЗАЯВКИ

[ВСЕ](#)   [НА РАССМОТРЕНИИ](#)   [ОДОБРЕНО](#)   [ОТКАЗАНО](#)

Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально ...  
**Проект:** Социально значимый проект "Пишем вместе"  
**Статус:** Одобрено

Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально ...  
**Проект:** Проект повышения проникновения информационных технологий в изучении русского языка  
**Статус:** На рассмотрении

Грантовая программа Фонда "Мир" в 2023 году  
**Проект:** Проект повышения проникновения информационных технологий в изучении русского языка  
**Статус:** Отказано

< 1 >

Рисунок 23 – Интерфейс списка входящих заявок

Для управления проектами разработаем интерфейс списка проектов (рисунок 24).

**Мои проекты**

#12 Социально значимый проект "Пишем вместе"

ЗАЯВКИ

#16 Проект повышения проникновения информационных технологий в изучении русского языка

ЗАЯВКИ

&lt; 1 &gt;

**Рисунок 24 – Интерфейс списка проектов**

Интерфейс просмотра заявки позволяет грантодателю и соискателю ознакомиться с деталями заявки и проекта, а также перейти к разделу обмена сообщениями для поддержки заявки, рассмотрим интерфейс на рисунке 25.

Гранты

ПОИСК ГРАНТОДАТЕЛЬ СОИСКАТЕЛЬ АККАУНТ ВЫЙТИ

Заявка №11 от 2023-03-12 15:01:41

ВЕРНУТСЯ К СПИСКУ ЗАЯВОК

ИНФОРМАЦИЯ СОПРОВОЖДЕНИЕ

**Статус:** Одобрена

**Грант:** [Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально значимых проектов](#)

**Проект:** [Социально значимый проект "Пишем вместе"](#)

Проект развития грамотности в малонаселенных районах Ямало-Ненецкого автономного округа

Требуемый бюджет: 500.0 тыс - 1 млн  
Период реализации: 2023-07-10 - 2024-07-10  
Стадия реализации: Прототип

**Соискатель:** ООО Вектор

ИНН: 7734415820  
КПП: 118774679

**Сопроводительный текст:** Здравствуйте! Мы хотим подать заявку на грант, но у нас есть вопросы. Можно ли подать заявку на грант и на другие источники финансирования одновременно? Если да, то какие? Если нет, то почему?

**Рисунок 25 – Интерфейс просмотра заявки**

Интерфейс обмена сообщениями в рамках поддержки заявки, позволяет видеть историю сообщений и обмениваться сообщениями и файлами грантодателю и соискателю, подавшему заявку на грант.

Интерфейс обмена сообщениями представлен на рисунке 26.

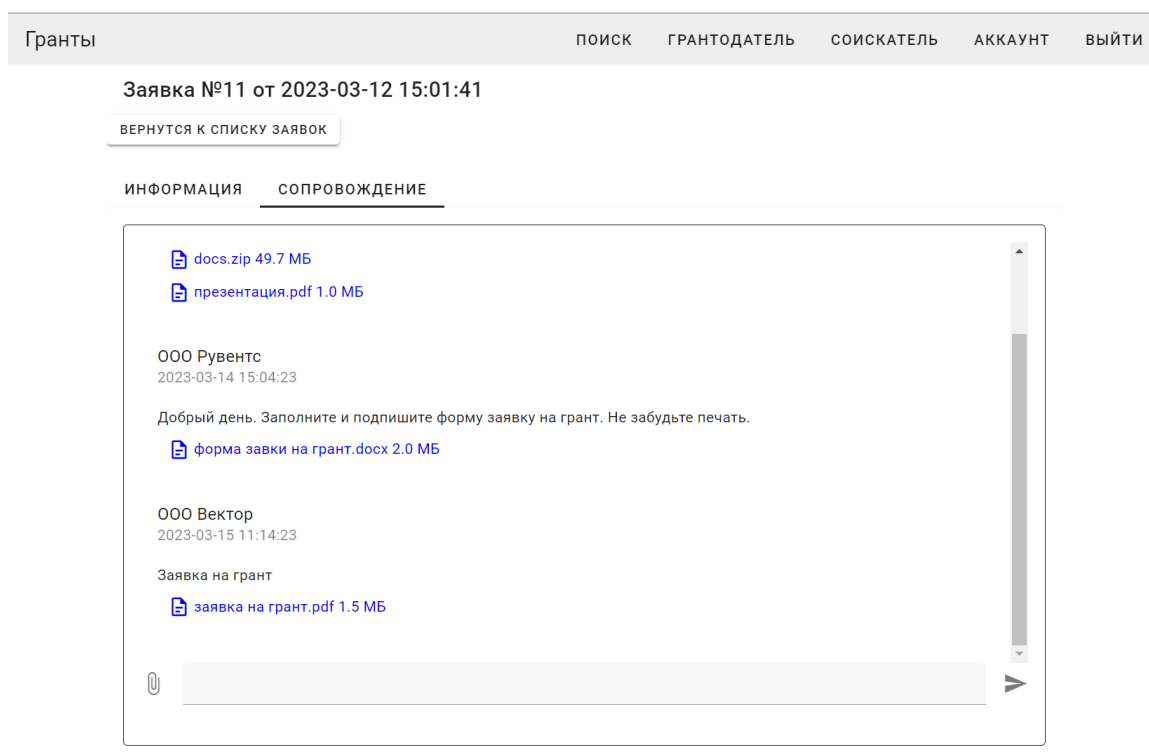


Рисунок 26 – Интерфейс обмена сообщениями

Простота интерфейсов обусловлена особенностью web-приложений, где пользователь зачастую удовлетворен простым быстрым решением, даже если альтернативные решения полнее покрывают его требования на длительный срок. [16]

Для контроля грантодателем своих грантов и их заявок разработаем интерфейс списка грантов.

Интерфейс списка грантов представлен на рисунке 27.

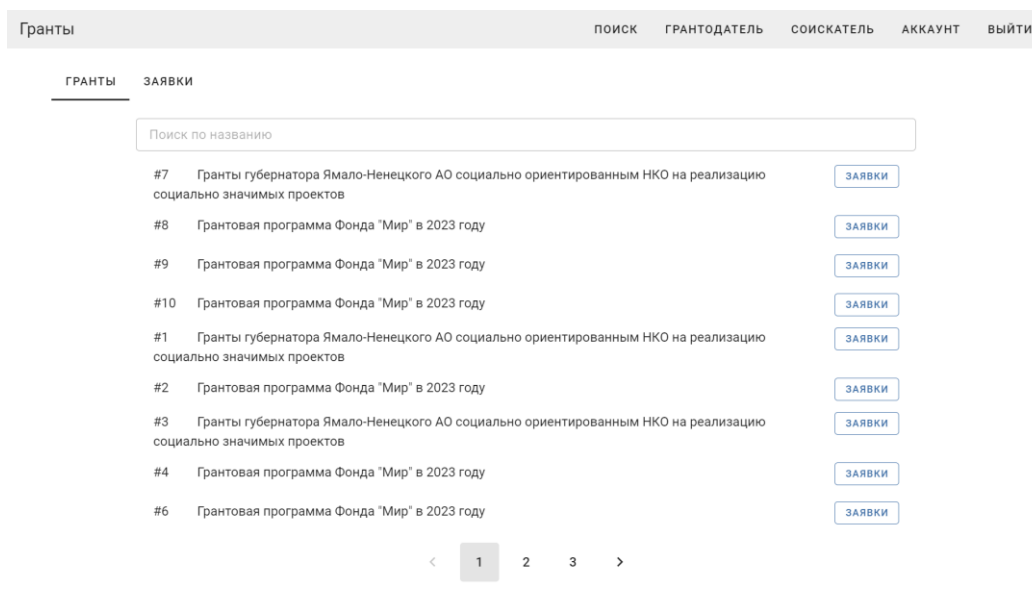


Рисунок 27 – Интерфейс списка грантов

Интерфейс создания и редактирования гранта показан на рисунке 28.

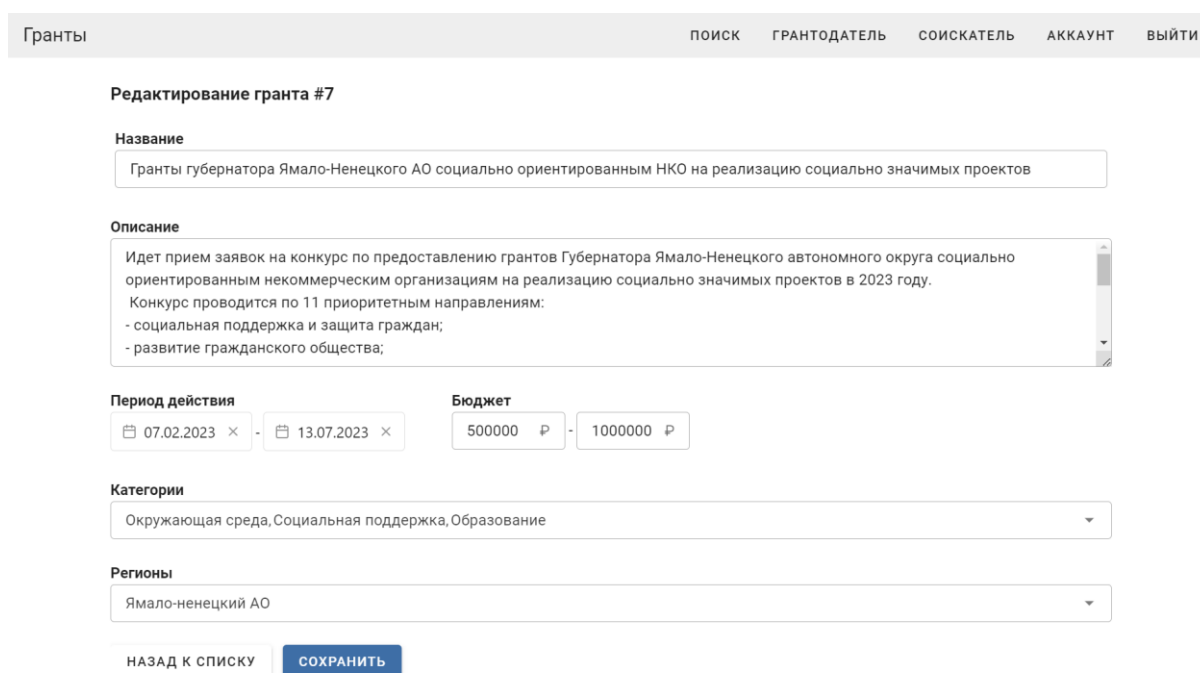


Рисунок 28 – Интерфейс создания и редактирования гранта



После сохранения данных гранта, грант попадает в общедоступный поиск, откуда соискатели могут подавать заявки от имени своих проектов.

## 2.6 Разработка клиентского приложения

Клиентское приложение написано на языке JavaScript. JavaScript, это динамический язык со слабо типизированными, динамически расширяемыми объектами, которые неформально объявляются по мере необходимости. [19]

Одним из больших преимуществ JavaScript является то, что его код может распознаваться непосредственно на HTML странице – т.е. браузер может не обращаться к серверу за интерпретацией каждой строки кода. [8]

В качестве основы клиентского приложения используется фреймворк VUEJS, построенный на принципе «реактивности». Принцип действия «реактивности» в web-приложениях мы можем увидеть на рисунке 29.

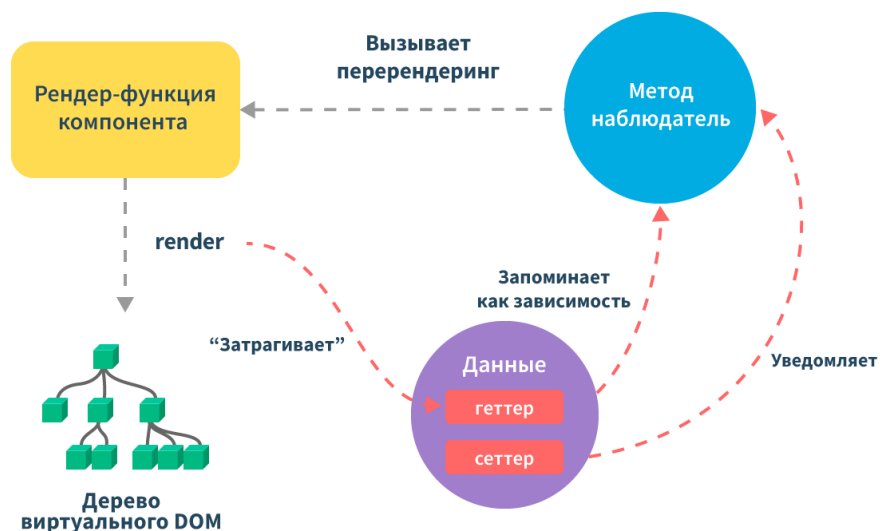


Рисунок 29 – Принцип работы реактивности в web-приложениях

Данный подход позволяет определить порядок взаимодействия компонентов, систематизировать расположение html и javascript кода в модуле, упростить связь бизнес-событий и реакций интерфейса на эти события.

Рисунок 30 отражает структуру проекта клиентского приложения сервиса грантов.

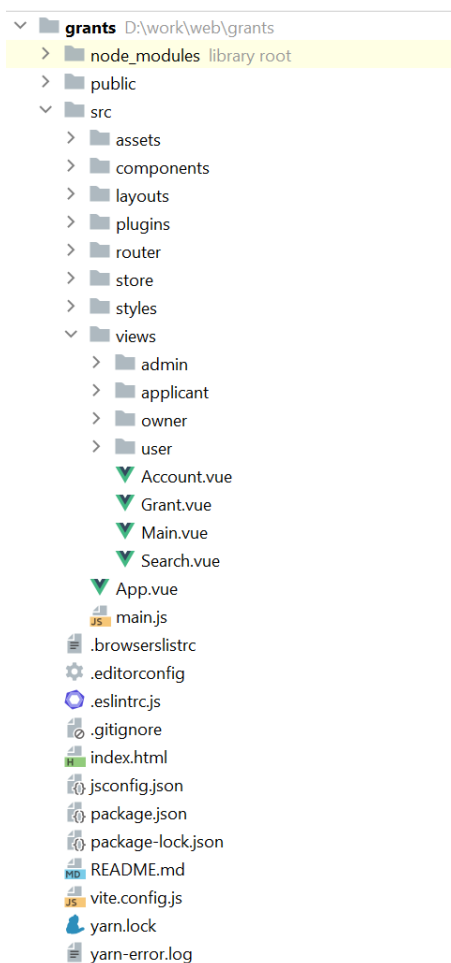


Рисунок 30 – Структура проекта клиентского приложения

Для построения страниц в фреймворке VueJS используются компоненты, компоненты могут быть включены друг в друга, повторно использоваться, иметь исходящие и входящие события и входные параметры.

Каждый компонент состоит из шаблона HTML и программного кода JavaScript выполняющего динамическое наполнение HTML шаблона и реагирующего на события DOM модели.

Компонент может иметь внутренние стили не влияющие на общий дизайн приложения, а также быть подверженными общей стилизации проекта, где они используются.

Так же при разработке был использован компилируемый `<script setup>` синтаксис, позволяющий упростить разработку и повысить производительность приложения.

Пример HTML кода компонента страницы редактирования гранта мы можем увидеть на рисунке 31.

```
1 <template>
2   <v-row>
3     <v-col offset="1" cols="10">
4       <v-card variant="text">
5         <br/>
6         <h3>
7           <span v-if="isCreate()">Создание</span>
8           <span v-else>Редактирование</span> гранта {{ buildTitle() }}
9         </h3>
10        <v-row style="margin-top:5px;padding:5px">
11          <v-col cols="12">
12            <b>Название</b>
13            <v-text-field density="compact" variant="outlined" v-model="grant.name" placeholder="3аролонок" single-line
14              hide-details/>
15          </v-col>
16        </v-row>
17        <v-row>
18          <v-col cols="12">
19            <b>Описание</b><v-textarea density="compact" ...>
20          </v-col>
21        </v-row>
22        <v-row>
23          <v-col cols="4">
24            <b>Период действия</b><br/><div style="display: inline-block"...> - <div style="display: inline-block"...>
25          </v-col>
26          <v-col cols="4">
27            <b>Бюджет</b><br/>
28          </v-col>
29          <div style="display: inline-block;width:120px"...>
30          <span style="position: relative;top: -15px;"> - </span>
31          <div style="display: inline-block;width:120px"...>
32        </v-row>
33      </v-row>
34    </v-col>
35  </v-row>
36 </template>
```

Рисунок 31 – HTML код компонента редактирования гранта

Javascript составляющую рассматриваемого компонента можно увидеть на рисунке 32.

```

102 <script setup>
103   import {useAppStore} from '@/store/app'
104   import router from '@/router'
105   import VueDatePicker from '@vuepic/vue-datepicker';
106   import '@vuepic/vue-datepicker/dist/main.css';
107   const store = useAppStore()
108   const grant = store.currentGrant
109   function save() {
110     store.storeGrant(grant)
111     router.push({name: 'OwnerGrantList'})
112   }
113   function isCreate() {
114     return router.currentRoute.value.params.id === undefined
115   }
116   function buildTitle() {
117     return router.currentRoute.value.params.id !== "" ? ("#" + router.currentRoute.value.params.id) : ""
118   }
119 </script>
120 <style>

```

Рисунок 32 – JavaScript составляющая компонента

Как мы можем видеть взаимодействие с серверной частью скрыто от компонента, и расположено в хранилище состояния.

Хранилище состояния – архитектурный подход, когда единый объект в клиентском приложении содержит всё глобальное состояние приложения и служит «единственным источником истины». Данный подход позволяет разделить функционал приложения в зависимости от его назначения, например взаимодействие с серверной частью не связано с отображением, что дает возможности для широкого повторного использования компонентов в разных контекстах.

На рисунке 33 мы можем увидеть пример метода хранилища состояния, предназначенного для передачи данных на серверное приложение при модификации гранта.

```
438     storeGrant(grant) {
439         |   this.axios.post( url: '/grants', grant)
440         |   .then(() => {
441         |   |   this.grant = grant
442         |   |   })
443         |   .catch((err) => alert(err))
444     },
```

Рисунок 33 – Метод передачи гранта в серверное приложение при редактировании

При нажатии на кнопку сохранения, вызывается метод storeGrant после чего делается HTTP POST запрос, содержащий в теле запроса данные в формате JSON, указанные в HTML шаблоне формы.

Пример запроса на создание гранта выполняемый между клиентским и серверным приложением приведен на рисунках 34, 35 и 36.

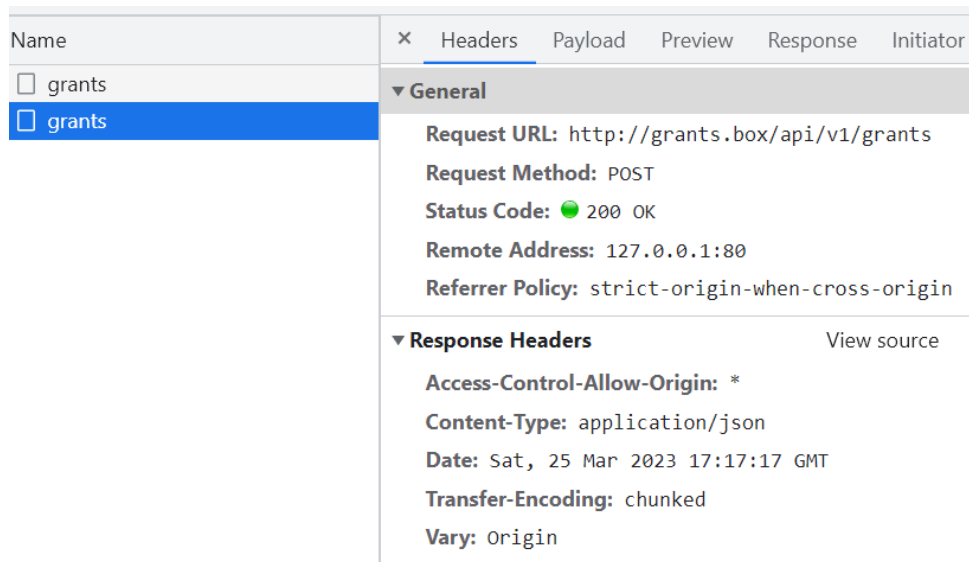


Рисунок 34 – Запрос создания гранта

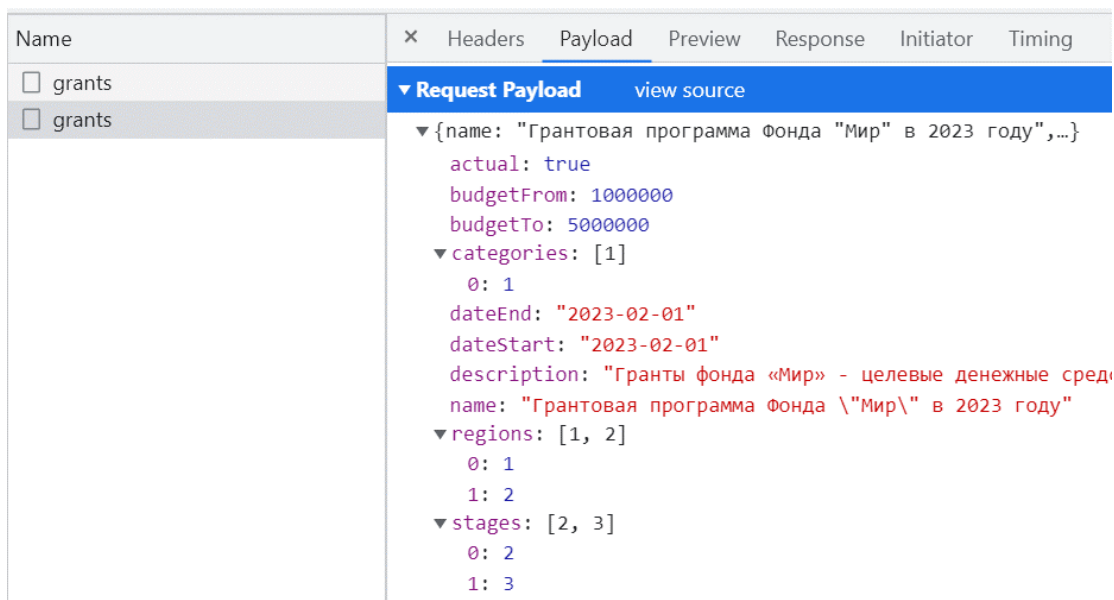


Рисунок 35 – Запрос создания гранта тело запроса

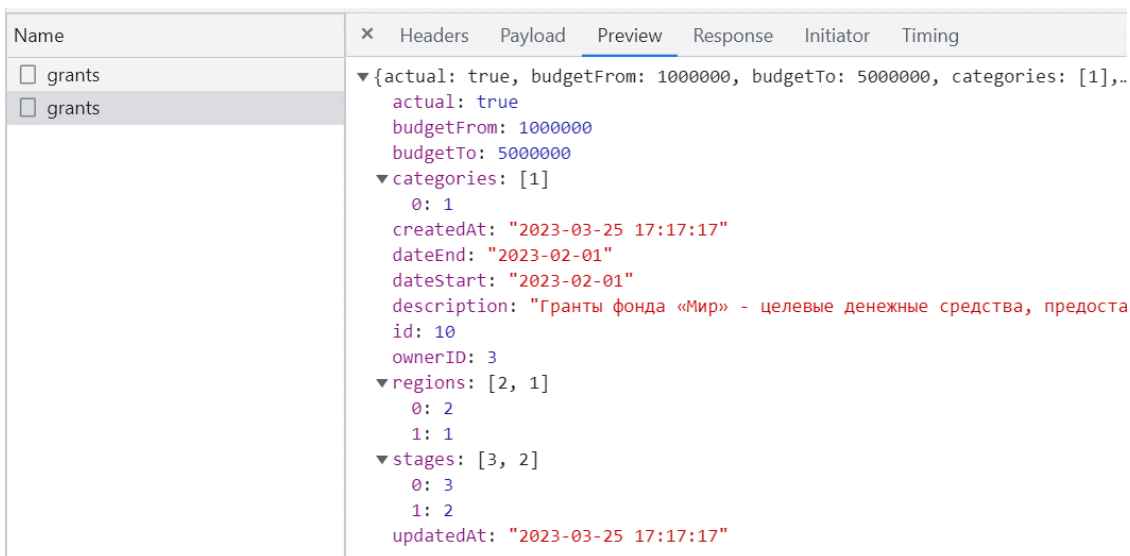


Рисунок 36 – Ответ на запрос создания гранта

Результаты создания гранта мы можем так же увидеть в виде записи в таблице grants базы данных серверного приложения. Запись в таблице отображена на рисунке 37.

		1
id	↕	11
owner_id	↕	3
title	↕	Грантовая программа Фонда "Мир" в 2023 году
description	↕	Гранты фонда «Мир» - целевые денежные средства, предо...
budget_from	↕	1000000
budget_to	↕	5000000
date_start	↕	2023-02-01 00:00:00
date_end	↕	2023-02-01 00:00:00
actual	↕	1
created_at	↕	2023-03-25 23:33:50.787
updated_at	↕	2023-03-25 23:33:50.787

Рисунок 37 – Запись в таблице grants

Создавать гранты, проекты, подавать заявки и обмениваться сообщениями могут только авторизованные пользователи.

Авторизация пользователей происходит путем выдачи JSON Web Token (JWT) – токена представляющего себя закодированную base64 строку, содержащую объект в кодировке JSON, а также подпись, полученную путем применения хеширования sha256 над содержательной частью токена.

Основное преимущество применения JWT токенов – отсутствие необходимости хранения дополнительных данных о выданных сессиях: все, что должен сделать сервер – это проверить подпись. Помимо этого, решается проблема распределенной авторизации, а также масштабирования.





WHERE id=6

	1
id	6
email	artem@email.com
password	\$2a\$10\$bqwxXEj9w3l1wFjhUx7t0.LVogcg5JUBunA24Fx4YAqgprkN3tEGK
name	Артем
phone	79998887766
is_admin	0
email_confirmed	1
created_at	2023-03-25 23:26:43.542
updated_at	2023-03-25 23:26:43.542

Рисунок 40 – Пример данных пользователя в базе данных

Вывод по главе 2

Во второй главе данного исследования было выполнено проектирование веб-приложения платформы для получения грантов.

В результате проведенного анализа информационных потоков и построения диаграммы использования приложения была сформирована модель для проектирования базы данных, на основе которой разработана физическая схема базы данных.

Спроектирован интерфейс пользователя, разработаны серверный и клиентский компоненты веб-приложения. В результате чего, проектирование и разработка веб-приложения были полностью выполнены.

## Глава 3 Анализ работы веб-приложения

### 3.1 Проведение контрольного примера использования

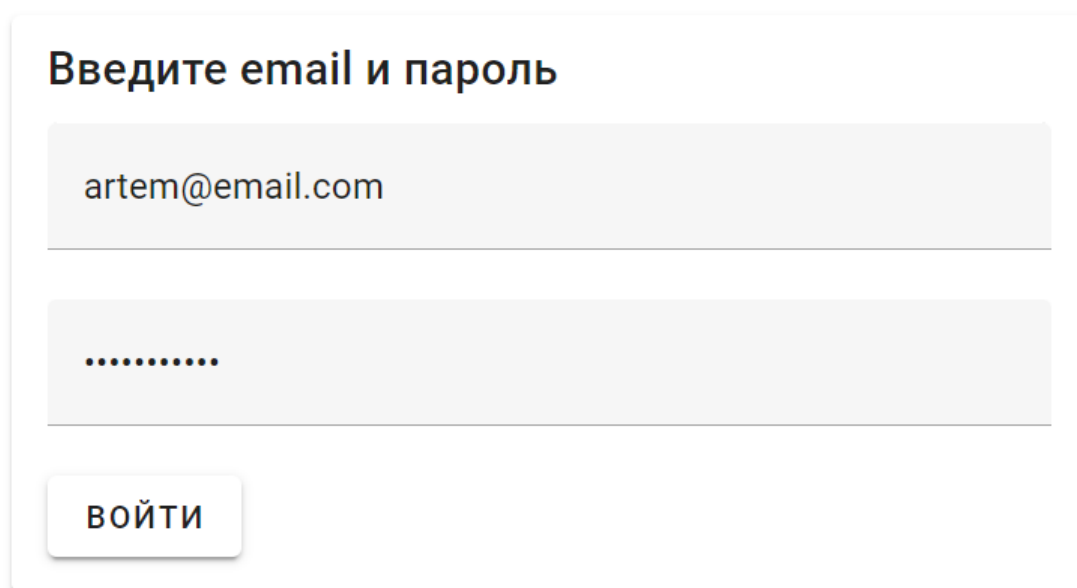
В веб-приложении предусмотрена работа под двумя ролями – «администратор» и «пользователь», обе роли имеют доступ к возможности искать, создавать и сопровождать гранты, а также к управлению данными аккаунта, и управлению своими проектами.

Администратор помимо вышеперечисленного так же имеет доступ к разделу пользователей, а также к проектам и грантам всех пользователей.

Администратор может менять и удалять гранты, проекты и самих пользователей.

Старт программы начинается с окна авторизации (рисунок 41).

Произведем авторизацию в роли «администратор».



**Введите email и пароль**

Рисунок 41 – Форма авторизации

При нажатии кнопки «войти» клиентское приложение делает запрос к серверному приложения передавая логин и пароль, в ответ получая токен аутентификации. Который в дальнейшем передается со всеми запросами от клиентского приложения к серверному приложению.

После авторизации в роли «администратор» в заголовке сайта будет отображены ссылки на страницы с соответствующим функционалом (рисунок 42).

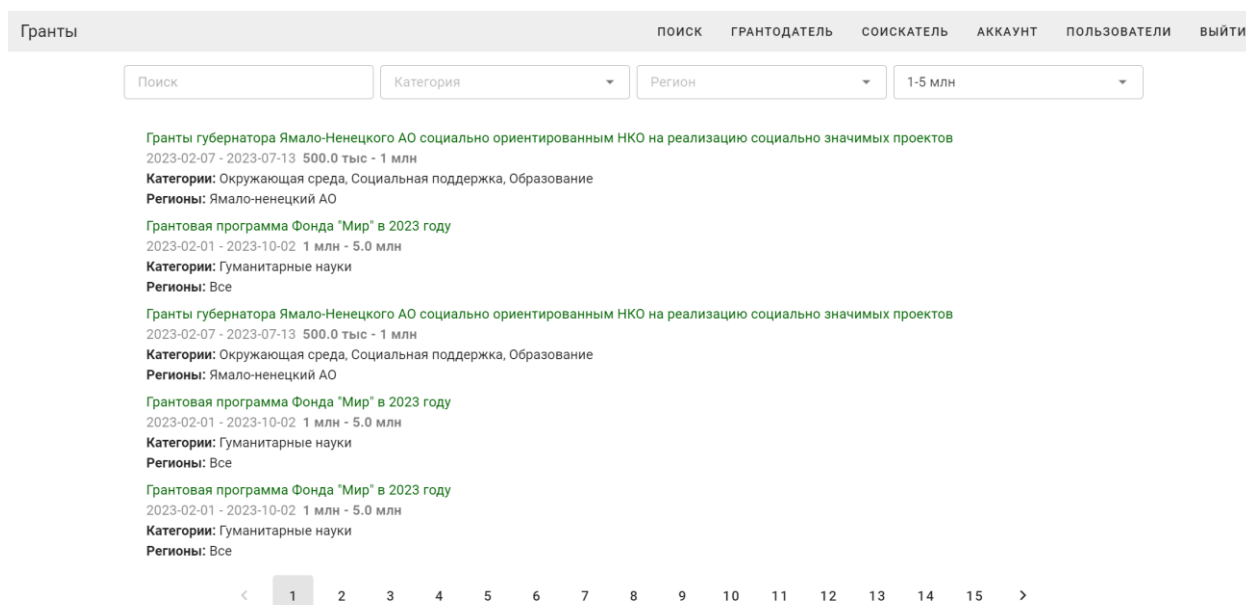


Рисунок 42 – Интерфейс веб-приложения авторизованного пользователя

В разделе «пользователи» располагается список пользователей (рисунок 43) и форма редактирования пользователя (рисунок 44).

Список пользователей представлен в виде таблицы, строки которой представляют пользователей, для просмотра информации о пользователе необходимо осуществить нажатие на кнопку «просмотр». Для удаления пользователя необходимо осуществить нажатие на кнопку «удалить».

Гранты	ПОИСК	ГРАНТОДАТЕЛЬ	СОИСКАТЕЛЬ	АККАУНТ	ПОЛЬЗОВАТЕЛИ	ВЫЙТИ
Поиск по id или имени						
<a href="#">ПРОСМОТР</a>	ООО Рувентс	test@email.com	7(952)777-66-55	<a href="#">УДАЛИТЬ</a>		
<a href="#">ПРОСМОТР</a>	ООО Вектор	con@vector.gov.ru	8(34938)56630	<a href="#">УДАЛИТЬ</a>		
<a href="#">ПРОСМОТР</a>	ООО Цифровые Решения	sales@dsol.ru	7(952)777-66-55	<a href="#">УДАЛИТЬ</a>		
< 1 >						

Рисунок 43 – Список пользователей

Гранты	ПОИСК	ГРАНТОДАТЕЛЬ	СОИСКАТЕЛЬ	АККАУНТ	ПОЛЬЗОВАТЕЛИ	ВЫЙТИ
<b>Редактирование аккаунта #1</b>						
Имя ООО Рувентс						
Email test@email.com						
Телефон 7(952)777-66-55						
Организация ООО Рувентс						
Иин 1234567890						
Кпп 123456789						
Пароль *****						
<a href="#">СОХРАНИТЬ</a>						

Рисунок 44 – Редактирование пользователя

После изменения данных, нажатия кнопки «сохранить» и отправки данных на сервер, данные пользователя будут записаны в базу данных.

Все пользователи сайта могут редактировать данные своего аккаунта в разделе «аккаунт». Раздел «аккаунт» представлен на рисунке 45.

Гранты      ПОИСК      ГРАНТОДАТЕЛЬ      СОИСКАТЕЛЬ      **АККАУНТ**      ПОЛЬЗОВАТЕЛИ      ВЫЙТИ

**Мой аккаунт**

Имя  
ООО Рувентс

Email  
test@email.com

Телефон  
7(952)777-66-55

Организация  
ООО Рувентс

ИИН  
1234567890

Кпп  
123456789

Пароль

**СОХРАНИТЬ**

Рисунок 45 – Редактирование данных аккаунта

В разделе «грантодатель» представлен список грантов (рисунок 46) и список заявок (рисунок 47).

Гранты      ПОИСК      **ГРАНТОДАТЕЛЬ**      СОИСКАТЕЛЬ      АККАУНТ      ПОЛЬЗОВАТЕЛИ      ВЫЙТИ

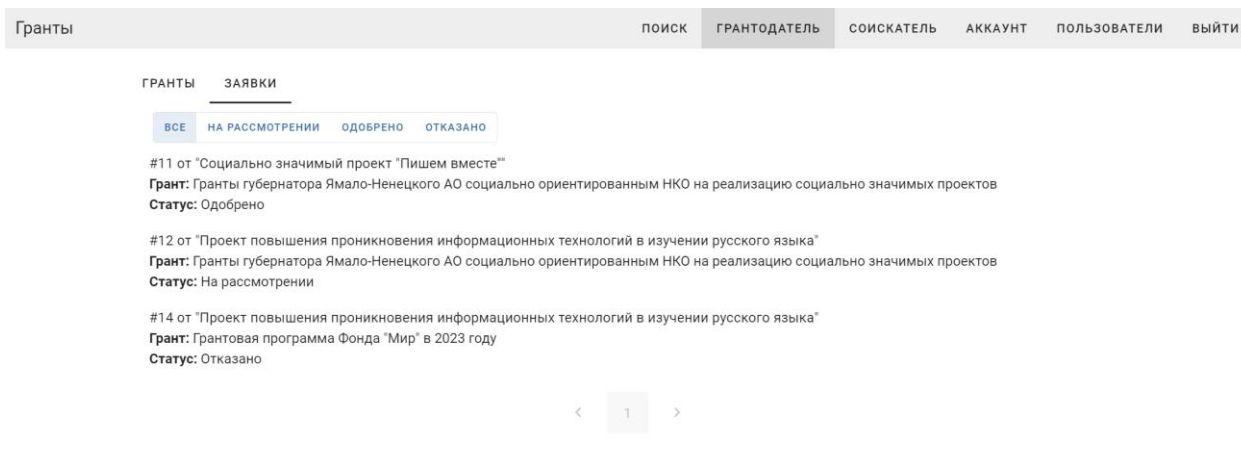
**ГРАНТЫ**      ЗАЯВКИ

**ДОБАВИТЬ НОВЫЙ ГРАНТ**      Поиск по названию

#7	Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально значимых проектов	<b>ЗАЯВКИ</b>
#8	Конкурс на соискание грантов Правительства Республики Бурятия в 2023 году	<b>ЗАЯВКИ</b>
#9	Грант программы «Особый взгляд»: Организация тифлокомментирования в театрах, 2023 г. Номинация «Старт»	<b>ЗАЯВКИ</b>
#10	Всероссийский конкурс по поддержке театрального искусства "Российского фонда культуры"	<b>ЗАЯВКИ</b>
#1	Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально значимых проектов	<b>ЗАЯВКИ</b>
#2	Национальная премия "Жить вместе"	<b>ЗАЯВКИ</b>
#3	Конкурс "Арт-резиденция" ГКБУК «Центр по реализации проектов в сфере культуры»	<b>ЗАЯВКИ</b>
#4	Премия "Вечный вклад" за лучшие исследования по истории целевых капиталов в российской дореволюционной филантропии	<b>ЗАЯВКИ</b>
#6	Конкурс грантов на проекты в сфере культуры и искусства для НКО Мурманской области	<b>ЗАЯВКИ</b>

< 1 2 3 >

Рисунок 46 – Список грантов



## Рисунок 47 – Список заявок

При нажатии на заявку в списке, пользователь попадает на страницу сопровождения заявки (рисунок 48).



## Рисунок 48 – Сопровождение заявки

На этой странице есть возможность совершать обмен сообщениями и файлами с соискателем (рисунок 49).

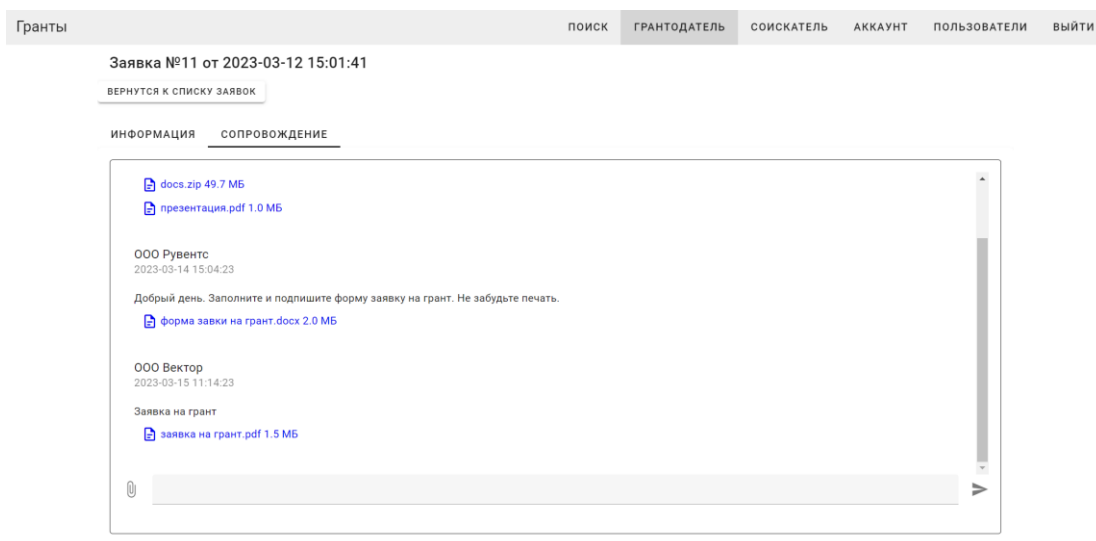


Рисунок 49 – Обмен сообщениями с соискателем

Раздел «соискатель» представляет собой список проектов, созданных пользователем (рисунок 50).

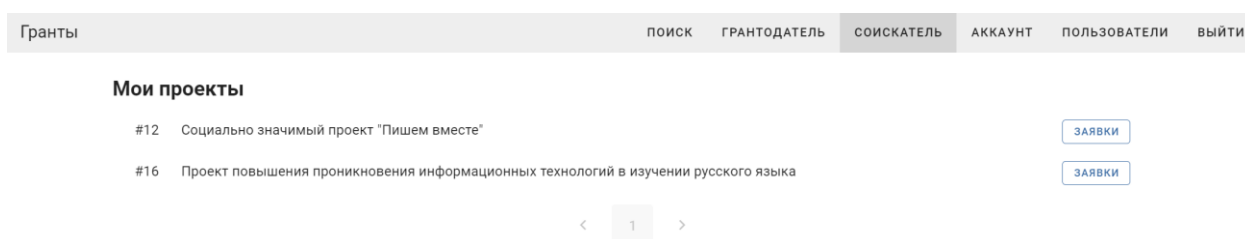


Рисунок 50 – Раздел «соискатель»

При нажатии на название проекта пользователь попадает на форму редактирования данных о проекте (рисунок 51).

Гранты ПОИСК ГРАНТОДАТЕЛЬ **СОИСКАТЕЛЬ** АККАУНТ ПОЛЬЗОВАТЕЛИ ВЫЙТИ

### Редактирование проекта #12

**Название**

**Описание**

**Период действия**  × -  ×

**Бюджет**  Р -  Р

**Категории**

**Регионы**

[НАЗАД К СПИСКУ](#) [СОХРАНИТЬ](#)

Рисунок 51 – Форма редактирования проекта

Так же из списка проектов можно перейти к списку заявок, поданных от имени проекта (рисунок 52).

Гранты ПОИСК ГРАНТОДАТЕЛЬ **СОИСКАТЕЛЬ** АККАУНТ ПОЛЬЗОВАТЕЛИ ВЫЙТИ

### Заявки от проекта #12 Социально значимый проект "Пишем вместе"

[К СПИСКУ ПРОЕКТОВ](#)

[ВСЕ](#) [НА РАССМОТРЕНИИ](#) [ОДОБРЕНО](#) [ОТКАЗАНО](#)

#12 на грант Гранты губернатора Ямало-Ненецкого АО социально ориентированным НКО на реализацию социально значимых проектов	<a href="#">УДАЛИТЬ</a>
#14 на грант Грантовая программа Фонда "Мир" в 2023 году	<a href="#">УДАЛИТЬ</a>

< 1 >

Рисунок 52 – Список заявок проекта

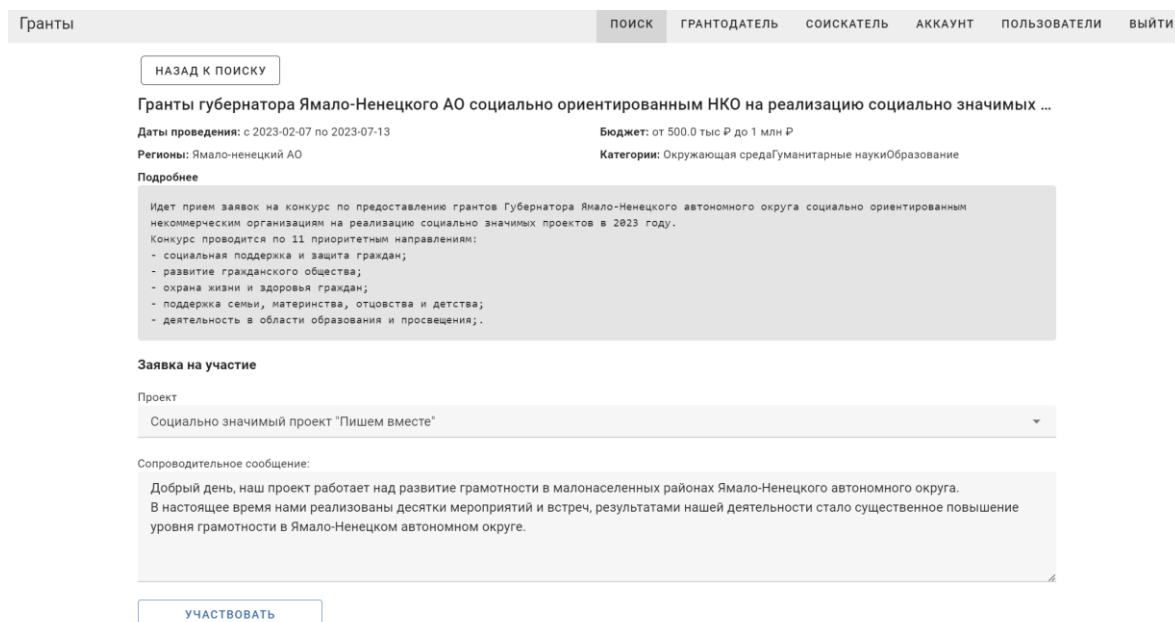
Общедоступным является раздел поиска опубликованных грантов, представленный на рисунке 53. Данный раздел показывает гранты добавленные всеми грантодателями в системе.





## Рисунок 53 – Поиск опубликованных грантов

При клике на заголовок гранта, пользователь переходит на страницу подробной информации о гранте (рисунок 54), где он может подать заявку от имени одного из своих проектов.



## Рисунок 54 – Страница информации о гранте

В контрольном примере была протестирована работа реализации спроектированного веб-сервиса. В результате прохождения сценариев в роли «администратор» и роли «пользователь» ошибки не были выявлены. Веб-приложение работает корректно.

### **3.2 Расчёт экономической эффективности**

В этом разделе требуется выполнить расчет капитальных затрат, связанных с разработкой веб-сервиса, а также рассчитать текущие и будущие ежемесячные расходы. После чего оценить экономический эффект от внедрения разработки для определения ее экономической целесообразности.

Для определения экономической эффективности проекта, необходимо рассчитать две составляющие:

Прямую эффективность, выражаемую в снижении трудовых затрат, за счет сокращения объема задач и снижении трудоемкости работы.

Косвенную эффективность, которая отражается на увеличении общей прибыли организации, уменьшении затрат на материалы, уменьшении сумм отчислений партнерам и т.д.

Для расчёта экономической эффективности работы используется методика расчёта прямой эффективности от внедрения ИС по сравнению с существующим вариантом.

Для определения абсолютного снижения трудозатрат используем формулу (1):

$$\Delta T = T \alpha \cdot T \beta \quad (1)$$

где  $T\alpha$  – период времени, затрачиваемый на выполнение задачи в проектном варианте,  $T\beta$  – период времени, затрачиваемый на выполнение задачи в базовом варианте.

Рассчитаем разницу стоимостных затрат на сопровождение операций после внедрения веб-приложения, по формуле (2):

$$\Delta C = C\alpha \cdot C\beta \quad (2)$$

где  $C\alpha$  – базовая стоимость выполнения задачи,  $C\beta$  – стоимость выполнения операций после внедрения ИС.

На основе описанных параметров мы можем рассчитать также коэффициенты изменения затрат.

Коэффициент изменения трудовых затрат, вычисляется по формуле (3):

$$Kt = (\Delta T / T\alpha) \quad (3)$$

Коэффициент изменения стоимостных затрат, рассчитывается по формуле (4):

$$Kc = (\Delta C / C\alpha) \quad (4)$$

Помимо вышеописанных затрат, важными экономическими показателями эффективности внедрения ИС являются показатели сокращения трудозатрат  $Yt$  и стоимостных затрат  $Yc$

Рассчитаем показатель сокращения трудозатрат, по формуле (5):

$$Yt = (T\alpha / T\beta) \quad (5)$$

Показатель стоимостных затрат рассчитываем, по формуле (6):

$$Yc = (C\alpha / C\beta) \quad (6)$$

Исходя из всех вышеуказанных параметров, мы можем рассчитать ориентировочный срок окупаемости проекта, используя формулу (7):

$$T_{ok} = (K_{\pi} / \Delta C) \quad (7)$$

где  $K_{\pi}$  - объем капитальных затрат на реализацию проекта.

Расчёты трудовых затрат до и после внедрения ИС указаны в таблице 5.

Таблица 5 – Расчёты трудовых затрат

Процесс	$T_{\alpha}$ , час	$T_{\beta}$ , час
Составление описания гранта с грантодателем	1	1
Создание мероприятия, публикация мероприятия	0,5	0
Поддержка актуальности информации о гранте	0,5	0
Поддержка участников мероприятия	1	0
Итого	3	1

Определим абсолютное снижения трудозатрат из расчёта на одно сопровождение гранта.

$$\Delta T = 3 - 1 = 2 \text{ часа}$$

Также получим коэффициент снижения трудозатрат.

$$K_t = \frac{1}{3} \cdot 100\% = 33,3\%$$

Определим индекс сокращения трудозатрат.

$$Y_t = \frac{3}{1} = 3$$

Итоговые показатели сокращения трудозатрат находятся в зависимости от объёмов заведения и сопровождения грантов на предприятии.

Например, если на одного сотрудника приходится 10 грантов в неделю, то сокращение трудозатрат составит 80 человеко-часов в месяц.

Если принять за среднюю стоимость человеко-часа за 150 рублей, экономия предприятия составляет 12000 рублей на каждого сотрудника в месяц, что позволит сократить количество работников выполняющих сопровождение грантов.

Рассчитаем далее изменение стоимостных затрат с учетом расходов на обслуживание внедряемого решения (таблица 6) ориентировочную стоимость рассчитаем за год обслуживания. [20]

Таблица 6 – Расчёт стоимостных затрат

Вид затрат	$C_a$ , руб.	$C_b$ , руб.
Стоимость поддержки ИС	0	20000
Фонд оплаты труда	200000	122000
Обеспечение технических средств	20000	40000
Итого	230000	182000

Фонд оплаты труда после оптимизации процессов представлен в виде ставок 3 сотрудников отдела онлайн сервисов в объеме 72000 рублей, а также ставку одного администратора ИС в объеме 40000 рублей.

Рассчитаем разницу стоимостных затрат.

$$\Delta C = 230000 - 182000 = 48000 \text{ руб.}$$

На основе полученной разницы стоимостных затрат, вычислим коэффициент изменения стоимостных затрат.

$$K_c = \left( \frac{48000}{230000} \right) \cdot 100\% = 20.86\%$$

Вычислим показатель стоимостных затрат.

$$Y_c = \frac{230000}{182000} = 1,26$$

Рассчитанные капитальные затраты на проектирование и разработку веб-приложения, сводим в таблицу 7.

Таблица 7 – Капитальные затраты на разработку веб-приложения

Статья затрат	Сумма, руб.
Регистрация и право владения доменом	450
Хостинг	3000
Оборудование и программное обеспечение	42800
Стоимость расходных материалов	330
Накладные расходы	5193
Итого	51773

Рассчитаем период окупаемости внедренного решения.

$$T_{ок} = \frac{(51773 \cdot 1.1)}{48000} = 1,19 \text{ месяца}$$

Исходя из произведенных вычислений мы можем убедиться, что реализация веб-приложения окупается в течении второго месяца внедрения, чем оказывает положительный экономический эффект.

Вывод по главе 3

В главе 3 проведено контрольное тестирование реализации спроектированного веб-приложения. Использование веб-приложения возможно под двумя ролями: «администратор» и «пользователь». В рамках проверки выполнены шаги от авторизации в роли «администратор» до управления проектами и грантами в роли «пользователь». «Администратор»

может модифицировать данные аккаунтов пользователей, а также удалять их. «Пользователь» может выступать в системе как «грантодатель» и «соискатель», управлять своими грантами, проектами, заявками на грант и сообщениями.

В результате тестирования веб-приложение не было выявлено ошибок, что позволяет сделать вывод о том, что веб-приложение работает корректно, в соответствии с заявленными требованиями.

В заключительном этапе был произведен расчет экономической эффективности. С учетом результатов оценки экономической эффективности, можно сделать вывод о том, что разработка веб-приложения является обоснованной, и ее внедрение позволит окупить разработку и высвободить дополнительные финансовые ресурсы в течение 1,26 месяцев после начала использования.

## Заключение

В данной работе была выполнен анализ деятельности ООО Цифродата и разработан веб-сервис «Платформа получения грантов».

В первой главе была проанализирована организационная структура ООО Цифродата и особенности структуры отдела онлайн сервисов, исследованы бизнес-процессы отдела онлайн сервисов, изучено техническое и аппаратное обеспечение отдела онлайн сервисов, на базе полученных данных разработана контекстная модель («AS IS» и «TO BE») процессов. В ходе проведенного анализа были обнаружены недостатки в бизнес-процессах, преимущественно связанные с отсутствием автоматизации в процессе сопровождения грантов. Для определения удачной стратегии автоматизации бизнес-процессов отдела онлайн сервисов в области получения грантов, был выполнен сравнительный анализ существующих веб-сервисов получения грантов. Анализ данных систем позволил использовать лучшие практики сферы получения грантов при построении подобного веб-сервиса.

Во второй главе была осуществлена разработка веб-приложения. Был составлен список необходимых инструментов для его реализации, определены основные информационные потоки, а также построена диаграмма вариантов использования. На основе результатов проделанной работы был спроектирован пользовательский интерфейс, разработаны программы серверного и клиентского приложения, спроектирована схема и физическая реализация базы данных.

В третьей главе была выполнена контрольная проверка спроектированного и разработанного веб-сервиса. Последовательно пройдены сценарии от авторизации в роли «администратор» до управления грантами, проектами, заявками и сообщениями в роли «пользователь». После завершения тестирования был проведен анализ экономической



эффективности, который подтвердил обоснованность разработки веб-сервиса и показал прогноз положительного экономического эффекта в течении срока окупаемости продолжительностью 1,26 месяца. [13]

Разработка веб-приложения позволила решить проблему автоматизации ряда задач. В дальнейшем, для расширения функционала приложения потребуется реализовать функционал авторизации через систему ЕГАИС, для улучшенной персонализации грантодателей и соискателей. Система спроектирована под нужды и задачи отдела онлайн сервисов, однако применение слоистой архитектуры, современных подходов и разбивки кода на пакеты позволит изменять и расширять необходимый функционал сервиса, а также поддерживать общеизвестные подходы к разработке, упрощающие включение новых разработчиков в работу над программой.

Теоретическая значимость данной работы выражается в упорядочении и закреплении знаний в области моделирования бизнес-процессов организаций, полученных в процессе обучения.

Практическая значимость данной работы заключается в возможности применения разработанного веб-приложения в существующем рабочем процессе отдела онлайн сервисов.

## Список используемых источников

1. ISO (1992). Database Language SQL (ISO 9075:1992(E)). [Текст] / International
2. ГОСТ Р 54869-2011. Проектный менеджмент. Требования к управлению проектом.
3. Гольцман В. MySQL 5.0. Библиотека программиста – Санкт-Петербург: Питер, 2010. – 256 с. – ISBN 978-5-49807-135-0.
4. Дергачев А.М., Кореньков Ю.Д., Логинов И.П., Сафронов А.Г. Технологии веб-сервисов: учебно-методическое пособие. - Санкт-Петербург: Университет ИТМО, 2021. - 100 с. - экз.
5. Донован, Алан А. А., Керниган, Брайан, У. Язык программирования Go - ООО И.Д. Вильямс, 2016. - 432 с
6. Конноли Т. Базы данных. Проектирование, реализация и сопровождение. / Т. Конноли, К. Бегг. – М.: Вильямс, 2003. – 1440 с.
7. Киселев Д.Ю., Киселев Ю.В. – Самара: СГАУ, 2014 – 20 с.
8. Коэн Исси, Лазаро; Джозеф. А.О. Полный справочник по HTML, CSS и Javascript. – Паблшерз: Эксмо, 2017. – 246 с.
9. Купер А. Интерфейс. Основы проектирования взаимодействия. Санкт-Петербург: Питер, 2022. – 720 с.
10. Маркин А.В. Построение запросов и программирование на SQL: учебное пособие. – Москва: Диалог-МИФИ, 2014. - 384 с. – ISBN 978-5-86404-227-4
11. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. – СПб: Питер, 2018. – 352 с.
12. Мартин Роберт Чистый код: создание, анализ и рефакторинг / Роберт Мартин. – Санкт-Петербург: Питер СПб, 2018. – 464 с. – ISBN 978-5-4461-0069-9

13. Скрипкин К.Г. Экономическая эффективность информационных систем. – Москва: ДМК Пресс, 2022. – 500 с.
14. Спинеллис Д. Идеальная архитектура. Ведущие специалисты о красоте программных архитектур. – СПб.: Символ–Плюс, 2010. – 528 с.
15. Стешин А.И. Информационные системы в организации [Электронный ресурс]: учебное пособие. – Саратов: Вузовское образование, 2019. – 194 с.
16. Терещенко П.В. Интерфейсы информационных систем: учебное пособие. – Новосибирск: НГТУ, 2012. – 67 с. – ISBN 978-5-7782-2036-2.
17. Фриман Адам Pro Go Полное руководство по программированию надежного и эффективного программного обеспечения с использованием Golang. – Нью-Йорк: Apress, 2022. – 1285 с. – ISBN 978-1-4842-7354-8
18. Хэнчетт Э., Листоун Б. Vue.js в действии – Санкт-Петербург: Питер, 2019. – 304 с. – ISBN 978-5-4461-1098-8
19. Хэррон Д. Node.js Разработка серверных веб-приложений на JavaScript. – Москва: ДМК, 2014. – 144 с.
20. Ясенев В.Н. Информационные системы и технологии в экономике: учебное пособие. – Москва: Юнити-Дана, 2015. – 560 с.

## Приложение А

### Дамп базы данных

```
create table categories (  
  id smallint unsigned auto_increment primary key,  
  title varchar(255) null  
) collate = utf8_bin;  
create table regions (  
  id smallint unsigned auto_increment primary key,  
  title varchar(255) null  
) collate = utf8_bin;  
create table stages (  
  id smallint unsigned auto_increment primary key,  
  title varchar(255) null ) collate = utf8_bin;  
create table users (  
  id int unsigned auto_increment primary key,  
  email longtext null,  
  password longtext null,  
  name longtext null,  
  phone longtext null,  
  is_admin tinyint(1) null,  
  email_confirmed tinyint(1) null,  
  created_at datetime(3) null,  
  updated_at datetime(3) null  
) collate = utf8_bin;  
create table files (  
  id int unsigned auto_increment primary key,  
  owner_id int unsigned null,  
  size bigint unsigned null,  
  name varchar(255) null,  
  uri varchar(1000) null,  
  mime varchar(255) null,  
  linked tinyint(1) null,  
  created_at datetime(3) null,  
  constraint fk_users_files foreign key (owner_id) references users (id)  
  on update cascade on delete cascade  
) collate = utf8_bin;  
create index idx_files_owner_id on files (owner_id);  
create table grants (  

```

## Продолжение Приложения А

```
id      int unsigned auto_increment primary key,
owner_id int unsigned null,
title   varchar(255) null,
description text null,
budget_from int unsigned null,
budget_to  int unsigned null,
date_start datetime(3) null,
date_end   datetime(3) null,
actual     tinyint(1) null,
created_at datetime(3) null,
updated_at datetime(3) null,
constraint fk_users_grants foreign key (owner_id) references users (id)
) collate = utf8_bin;
create table grant_categories (
  grant_id int unsigned not null,
  category_id smallint unsigned not null,
  primary key (grant_id, category_id),
  constraint fk_grant_categories_category
    foreign key (category_id) references categories (id)
    on update cascade on delete cascade,
  constraint fk_grant_categories_grant
    foreign key (grant_id) references grants (id)
);
create table grant_files (
  grant_id int unsigned not null,
  file_id int unsigned not null,
  primary key (grant_id, file_id),
  constraint fk_grant_files_file
    foreign key (file_id) references files (id),
  constraint fk_grant_files_grant foreign key (grant_id) references grants (id),
);
create table grant_regions (
  grant_id int unsigned not null,
  region_id smallint unsigned not null,
  primary key (grant_id, region_id),
  constraint fk_grant_regions_grant
    foreign key (grant_id) references grants (id),
```

## Продолжение Приложения А

```
constraint fk_grant_regions_region
foreign key (region_id) references regions (id),
);
create table grant_stages (
grant_id int unsigned not null,
stage_id smallint unsigned not null,
primary key (grant_id, stage_id),
constraint fk_grant_stages_grant
foreign key (grant_id) references grants (id)
on update cascade on delete cascade,
constraint fk_grant_stages_stage
foreign key (stage_id) references stages (id)
on update cascade on delete cascade,
);
create table projects (
id int unsigned auto_increment primary key,
owner_id int unsigned null,
stage_id smallint unsigned null,
title varchar(255) null,
description text null,
created_at datetime(3) null,
updated_at datetime(3) null,
constraint fk_projects_stage key (stage_id) references stages (id),
constraint fk_users_projects foreign key (owner_id) references users (id)
) collate = utf8_bin;
create table grant_claims (
id int unsigned auto_increment primary key,
grant_id int unsigned null,
owner_id int unsigned null,
project_id int unsigned null,
covering_letter text null,
decision varchar(255) null,
decision_comment text null,
budget_from int unsigned null,
budget_to int unsigned null,
created_at datetime(3) null,
updated_at datetime(3) null,
```

## Продолжение Приложения А

```
constraint idx_grant_claim unique (grant_id, owner_id, project_id),
```

```

    constraint fk_grant_claims_grant foreign key (grant_id) references grants (id),
    constraint fk_grant_claims_owner foreign key (owner_id) references users (id),
    constraint fk_grant_claims_project foreign key (project_id) references projects
(id)
);
create table grant_claim_messages (
    id int unsigned auto_increment primary key,
    claim_id int unsigned null,
    owner_id int unsigned null,
    message text null,
    created_at datetime(3) null,
    constraint fk_grant_claim_messages_claim
        foreign key (claim_id) references grant_claims (id),
    constraint fk_grant_claim_messages_owner
        foreign key (owner_id) references users (id) collate = utf8_bin;
create index idx_grant_claim_messages_claim_id
    on grant_claim_messages (claim_id);
create index idx_grant_claim_messages_owner_id
    on grant_claim_messages (owner_id);
create table grant_claim_participants (
    id int unsigned auto_increment primary key,
    claim_id int unsigned null,
    user_id int unsigned null,
    constraint idx_claim_participant
        unique (claim_id, user_id),
    constraint fk_grant_claim_participants_claim
        foreign key (claim_id) references grant_claims (id),
    constraint fk_grant_claim_participants_user
        foreign key (user_id) references users (id)
);
create table message_files (
    grant_claim_message_id int unsigned not null,
    file_id int unsigned not null,
    primary key (grant_claim_message_id, file_id),
    constraint fk_message_files_file foreign key (file_id) references files (id)
on update cascade on delete cascade,
    constraint fk_message_files_grant_claim_message
        foreign key (grant_claim_message_id) references grant_claim_messages (id)
on update cascade on delete cascade
);
create table project_categories (
    project_id int unsigned not null,

```

## Продолжение Приложения А

```
category_id smallint unsigned not null,  
primary key (project_id, category_id),  
constraint fk_project_categories_category  
    foreign key (category_id) references categories (id),  
constraint fk_project_categories_project  
    foreign key (project_id) references projects (id)  
);  
create table project_files (  
    project_id int unsigned not null,  
    file_id int unsigned not null,  
    primary key (project_id, file_id),  
    constraint fk_project_files_file foreign key (file_id) references files (id)  
    on update cascade on delete cascade,  
    constraint fk_project_files_project foreign key (project_id) references projects  
(id)  
);  
create table project_regions (  
    project_id int unsigned not null,  
    region_id smallint unsigned not null,  
    primary key (project_id, region_id),  
    constraint fk_project_regions_project foreign key (project_id) references  
projects (id),  
    constraint fk_project_regions_region foreign key (region_id) references regions  
(id)  
);  
create index idx_projects_owner_id on projects (owner_id);  
create index idx_projects_stage_id on projects (stage_id);
```