

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

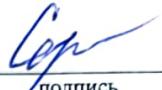
**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

Допускаю к защите

Руководитель


подпись

I.A. Серышева

И.О. Фамилия

Основные проектные решения по разработке платформы для
нетворкинга начинающих ИТ специалистов для повышения
уровня компетенций в командной разработке

наименование темы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту по дисциплине**

Проектирование информационных систем

1.008.00.00 ПЗ

обозначение документа

Выполнил студент

ИСТб-19-2

шифр группы


подпись

Ю.В. Комогорцева

И.О. Фамилия

Нормоконтроль


подпись

И.А. Серышева

И.О. Фамилия

Курсовой проект защищен с оценкой



Иркутск 2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

По курсу Проектирование информационных систем

Студенту Комогорцевой Ю.В.
(фамилия, инициалы)

Тема проекта Основные проектные решения по разработке платформы для нетворкинга начинающих ИТ специалистов для повышения уровня компетенций в командной разработке

Исходные данные Разработать систему, которая предоставляет информацию о проектах и хакатонах, позволяет объединяться участникам в команды.

Рекомендуемая литература

1) Проектирование АСОИУ : [Электронный ресурс] : методические указания по выполнению курсового проекта: 09.03.01 «Информатика и вычислительная техника» для бакалавров по специальности «Автоматизированные системы обработки информации и управления» / Иркут. нац. исслед. техн. ун-т ; сост. Р. Д. Гутгарц. - Иркутск : ИРНИТУ, 2018. - 13 с. - URL: <http://elib.istu.edu/viewer/view.php?file=/files3/er-17727.pdf>.

Графическая часть на листах.

Дата выдачи задания « 26 » сентября 2022 г.

Задание получил


подпись

Ю.В. Комогорцева

И.О. Фамилия

Дата представления проекта руководителю « 23 » декабря 2022 г.

Руководитель курсового проектирования

Лев
Подпись

И.А. Серышева

И.О. Фамилия

Содержание

Введение	4
1 Предпроектное обследование	5
1.1 Описание процесса повышения навыков в ИТ	5
1.2 Идея решения.....	8
1.3 Обзор аналогов	9
2 Постановка задачи и формирование требований к системе	12
2.1 Постановка задачи	12
2.2 Требования к системе в целом	12
2.2 Требования к функциям, выполняемым системой	13
2.3 Требования к ролям и правам доступа	15
2.4 Требования к информационному обеспечению	18
2.4.1 Требования к составу, структуре и способам организации данных в системе	18
2.4.2 Требования к информационной совместимости со смежными системами	18
2.4.3 Требования по применению СУБД.....	19
2.4.4 Требования к структуре процесса сбора, обработки, передачи данных в системе и представлению данных	19
2.4.5 Требования к защите данных от разрушений при авариях и сбоях в электропитании системы	19
2.5 Требования к верстке	19
2.6 Требования к структуре сайта	20
2.7 Требования к техническому обеспечению	21
2.8 Требования к составу и содержанию работ создания системы	22
3 Проектирование системы.....	24
4.1 Обоснование выбора инструментария	24
4.2 Моделирование бизнес-процессов (ТО-ВЕ)	25
4.3 Проектирование вариантов использования	30
4.3.1 Вариант использования «Создание проекта».....	30
4.3.2 Вариант использования «Участие в проекте».....	32
4.3.3 Вариант использования «Участие в хакатоне»	37
4.3.4 Вариант использования «Проведение мероприятия».....	38
4.4 Проектирование архитектуры системы.....	39
4.5 Проектирование базы данных.....	44
4.6 Проектирование пользовательского интерфейса.....	48
5 Реализация приложения	52
5.1 Описание возможностей текущей версии	52
5.2 Демонстрация варианта использования «Создание проекта»	53
6 Экономическая часть	59
Заключение	64
Список использованных источников	65
Приложение А	66

Введение

Сфера информационных технологий и ее рынок профессий показывает положительные тенденции роста уже несколько лет. Исходя из этого, возникает большое количество людей, желающих развиваться в этой сфере. Вместе с этим, растет конкуренция. Теперь для построения карьеры недостаточно иметь только профессиональные навыки (hard skills). Сейчас важно иметь так называемые «мягкие» навыки — soft skills. Например, уметь общаться с клиентами, работать в команде, проявлять инициативу или руководить процессом. При таких требованиях каждый новичок в ИТ-индустрии должен стремиться к совершенствованию своих soft skills и получению первичного опыта разработки. Существует множество инструментов в данном деле. Самым простым является участие в каких-либо мероприятиях (хакатоны, конференции, семинары, мастер-классы) или создание пет-проектов с сообщниками. Предложенные варианты являются универсальными, потому что помимо повышения профессиональных компетенций они помогают развивать личностные качества, благодаря работе в команде.

После появления интереса и инициативы к возможностям саморазвития появляется ряд вопросов:

Где искать информацию о мероприятиях?

Как найти команду?

На первый вопрос можно найти ответ в интернете, а вот второй вопрос можно решить, имея высокий уровень социальных навыков.

Исходя из вышеописанной ситуации, можно сформировать проблему: отсутствует система, которая консолидирует информацию о возможных источниках опыта, позволяет объединяться в команды и приобретать полезные связи. Отсутствие системы и потребность людей приводят к необходимости разработки. Поэтому необходимо создать систему, целью которой будет: сопровождение процесса повышения уровня компетенций в командной разработке посредством нетворкинга.

После определения цели будущей системы, необходимо упомянуть о том, что разработка программного продукта будет происходить в рамках курсового проекта по дисциплине «Проектирование информационных систем». Необходимо выполнить этапы анализа предметной области, формирования требований, проектирования и реализации.

Предполагается, что курсовой проект является фундаментом выпускной квалификационной работы. А также планируется поддержание и развитие системы в будущем.

Планируемый результат заключается в рабочей общедоступной системе, которая сможет увеличить информированность начинающих ИТ-специалистов о возможностях получения первичного опыта и предоставит возможность объединяться в команды для выполнения ИТ-проектов.

1 Предпроектное обследование

1.1 Описание процесса повышения навыков в ИТ

В ИТ-сфере для построения карьеры необходимо иметь не только профессиональные навыки (hard skills), но и личностные качества (soft skills). Развитие этих компетенций тесно связано с работой в команде и командной разработкой. Повысить свои личностные качества можно занимаясь проектной деятельностью или участвуя в хакатонах и иных мероприятиях. Рассмотрим более подробно способы получения опыта командной разработки.

Часто разработчики помимо основной работы занимаются разработкой личных проектов. Такие проекты называются пет-проектами. Благодаря пет-проекту разработчик отдыхает от рутинных задач, прокачивает навыки и осваивает новые инструменты. Из-за возможности получения выгоды многие компании сами поощряют пет-проекты сотрудников. Например, Gmail и Google News появились благодаря стимулированию Google работников уделять часть рабочего времени личным проектам. Идеи проектов можно найти на различных сайтах в интернете.

Преимущества разработки пет-проектов:

- пет-проекты приносят удовольствие и позволяют развиваться;
- личные проекты помогают найти лучшую работу;
- компаниям выгодно стимулировать эксперименты сотрудников.

Недостатки пет-проектов:

- нехватка времени на выполнение;
- отношения с друзьями и семьей ухудшаются из-за смещения приоритетов;
- негативные эмоции в случае неудачи проекта.

Участие в мероприятиях – необходимая деятельность в процессе профессионального развития и роста специалиста. ИТ-конференции, слеты, митапы, хакатоны и фестивали – это неотъемлемая часть комьюнити в ИТ-сфере. Польза посещения мероприятий в первую очередь заключается в нетворкинге, смене обстановки, актуализации знаний и обмене опытом. Мероприятия проходят по различным направлениям развития информационных технологий.

Одним из видов мероприятий являются хакатоны. Хакатон – это событие, на котором программисты, дизайнеры и менеджеры сообща решают какую-то проблему за ограниченное время, соревнуясь между собой. Слово «хакатон» образовалось от hacker («хакер») и marathon («марафон»).

Обычно цель хакатона – создать полноценную версию ПО или минимально жизнеспособный продукт по инициативе какой-то компании. Каждый хакатон сфокусирован на конкретной области знаний – языке программирования, операционной системе, специфических типах приложений или программном интерфейсе. Хакатоны устраивают не только для программистов, но и для дизайнеров и менеджеров. Призы тоже бывают разные: крупная сумма, грант на развитие проекта, оффер в компанию, льготы при поступлении в вуз или просто удовлетворение (благотворительные хакатоны).

В соревнованиях практически всегда участвуют группы из 2-5 человек: разработчики, UI/UX-дизайнеры, аналитики, продуктологи, маркетологи. Иногда это сформированные команды, которые перемещаются из хакатона в хакатон. Иногда встречаются команды, которые выстраиваются вокруг лидера: он привлекает разных специалистов в зависимости от мероприятия. В некоторых случаях группы формируют организаторы перед соревнованием.

Обычно участники проходят предварительный отбор: организаторы оценивают качество заполнения заявки, актуальность предложенной идеи и другие параметры, а финалистов приглашают на само мероприятие [1].

В среднем хакатоны делятся 2–3 дня, но срок может варьироваться от нескольких часов до недели. Для отслеживания прогресса есть чекпоинты: в определенное время, несколько раз за хакатон, участники показывают результаты работы и получают отзывы от менторов. Командам это позволяет понять, в правильном ли направлении они двигаются, а компаниям-заказчику – убедиться, что участники работают над MVP и не используют готовые решения с открытым кодом.

Кроме консультаций и советов от менторов, участники получают навыки работы в команде, подготовки и защиты презентации, тайм-менеджмента, работы в стрессовой ситуации.

В результате все активности необходимо искать. Есть различные способы, которые будут рассмотрены ниже. После рассмотрения объектов предметной области необходимо проанализировать процессы.

Процессы, которые выполняются в данный момент для повышения компетенций с помощью описанных выше методов, выглядят следующим образом:

Поиск пет-проекта:

1. Необходимо найти статьи с рекомендациями о пет-проектах для своей области ИТ и ознакомиться с ними;
2. Прочитать форумы или обсуждения в социальных сетях;
3. Подобрать идею для пет-проекта исследуя потребности знакомых;
4. Выбрать из готовых предложенных или генерировать самому идею.

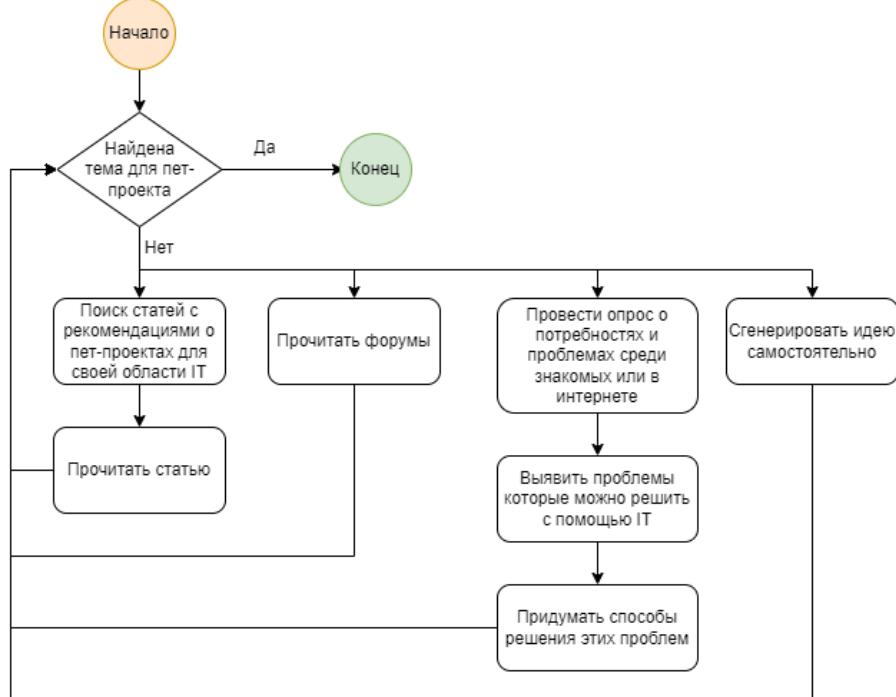


Рисунок 1.1 – Алгоритм поиска пет-проекта

Поиск хакатона:

1. Найти ресурсы с хакатонами (иногда это группы в социальных сетях или чаты компаний, занимающихся организацией хакатонов);
2. Найти хакатон с помощью нетворкинга;
3. Отслеживать подходящий по навыкам и интересам хакатон.

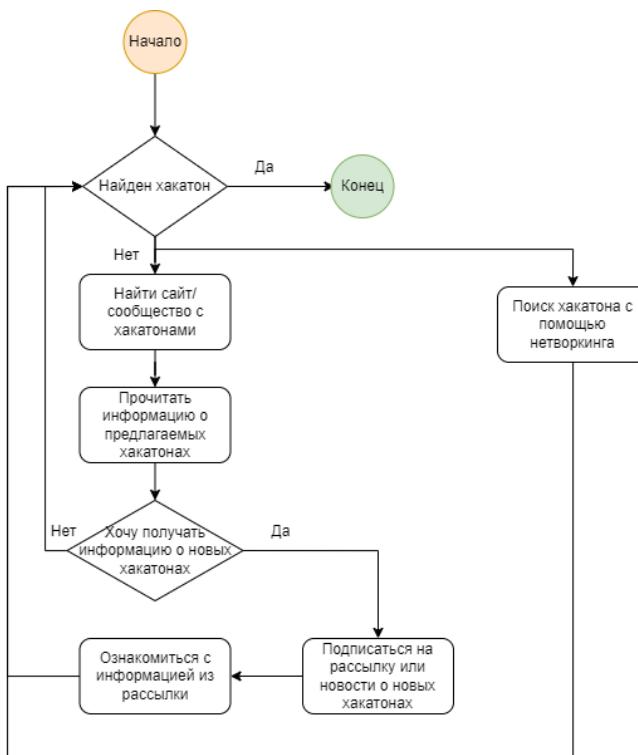


Рисунок 1.2 – Алгоритм поиска хакатона

Поиск команды для проекта или хакатона:

1. Поиск через платформу, на которой проводится хакатон;
2. Поиск через социальные сети (объявления, чаты);
3. Поиск с помощью нетворкинга (знакомые, мероприятия, конференции).

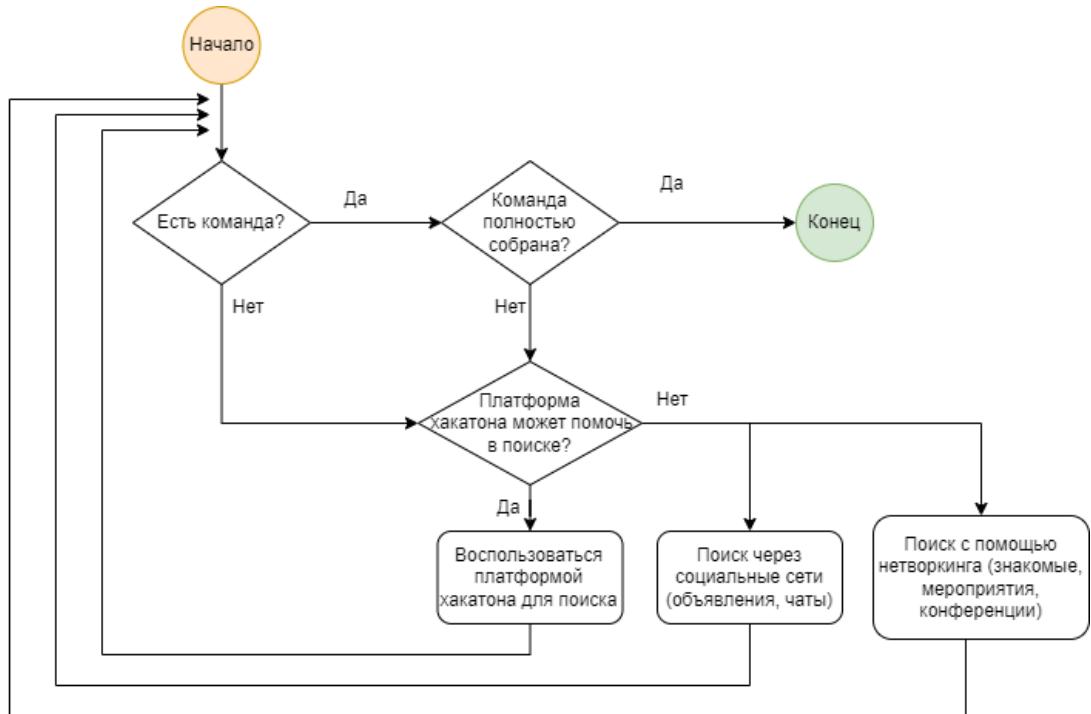


Рисунок 1.3 – Алгоритм поиска команды для проекта или хакатона

К сожалению, каждый из процессов занимает немало времени, выполняется много ненужных действий. В основе всего лежат статьи, соц. сети и сайты с хакатонами. Источники информации для каждого процесса разные, что является существенным недостатком.

Часто, несмотря на желание человека получить опыт, дальше ознакомления информации он не заходит. Чтобы решить эту проблему, необходимо учитывать систему мотивации.

На основании проведенного анализа выявлена проблема: **отсутствует система, которая консолидирует информацию о возможных источниках опыта, позволяет объединяться в команды и приобретать полезные связи.**

1.2 Идея решения

Исходя из выявленной проблемы и поставленной цели, предлагается решение: автоматизировать процесс поиска активностей для повышения компетенций в командной разработке. Активностями в системе должны являться проекты, хакатоны и мероприятия.

Проекты должны создаваться пользователями и проходить процесс модерации администратором. Пользователи могут вступить в команду проекта или создать свою. Проекты могут быть уникальными (аналог стартапов) или учебными (может быть много команд). Нужна возможность фиксации результатов участия в проекте.

Так как запись на участие в хакатоне в основном ведется на сторонних ресурсах, то на первоначальном этапе достаточно лишь предоставлять информацию о них. Автоматизировать этот процесс можно с помощью агрегирования хакатонов с других сайтов.

Мотивацию необходимо предусмотреть с помощью конкурсов и челленджей на платформе, а также системы достижений, правила выдачи которых необходимо определить на этапе проектирования. Так же, для увеличения знаний и компетенций можно предусмотреть проведение различных мероприятий с помощью системы.

Для реализации идеи необходимо создать личный кабинет, так как необходимо хранить информацию об участниках. Так же, необходимо роль администратора и панель управления системой.

1.3 Обзор аналогов

Исходя из поставленной цели, выявленной проблемы и предложенного решения, необходимо обязательно выполнить обзор аналогов. Это поможет выявить не только наличие аналогов, но и интересные особенности или недостатки, которые можно учесть при создании системы. Для проведения анализа была собрана информация о 17 сервисах. Их сравнение невозможно без критериев. Важно узнать имеет ли система возможности подбора проектов, мероприятий(хакатонов) и команд/участников. Актуальность контента так же является необходимым критерием при сравнении. Дополнительными критериями с точки зрения обычного пользователя можно считать язык системы, дизайн и простоту использования.

В итоге, все критерии должны иметь возможные значения и их описание. Предлагается, значения ставить в цифрах, чтобы после анализа можно было определить по сумме значений схожесть разрабатываемой системы с найденными аналогами.

Таблица 1.1 – Значение критериев

№	Критерий	Значения	Описание
1	Источник агрегатора	Да	Может являться источником для агрегатора, актуальный и подходящий контент
		Нет	Не может являться источником для агрегатора, контент не подходит
2	Подбор проектов	1	Наличие проектов: в виде идей, проектов для выполнения или в виде готовых решений
		0	Отсутствуют проекты или любая связанная с ними концепция
3	Подбор команды или участников для проекта	1	Наличие функций подбора команды, вступления в нее или поиска участника в команду

Продолжение таблицы 1.1

№	Критерий	Значения	Описание
3	Подбор команды или участников для проекта	0	Работа в команде внутри проектов отсутствует
4	Подбор мероприятий (хакатоны)	1	Наличие функций для проведения мероприятия, информация о них, в том числе о хакатонах
		0	Отсутствует информация о мероприятиях и функции для их проведения
5	Подбор команды или участников для мероприятия	1	Наличие функций подбора команды, вступления в нее или поиска участника в команду
		0	Работа в команде на мероприятиях отсутствует
6	Актуальность и полнота контента	1	Актуальный контент, достаточная проработка
		0	Отсутствует свежая информация, либо ее очень мало
7	Язык	ru	Русскоязычная версия
		en	Англоязычная версия
8	Дизайн	1	Приятный дизайн, контент воспринимается легко
		0	Плохая читабельность и непривлекательность
9	Простота использования	1	Удобный и понятный интерфейс, легко ориентироваться в системе
		0	Трудно ориентироваться, неструктурированная система

Проведенный анализ по критериям из таблицы № 1.1 представлен в приложении А. Из 17 сервисов были подобраны для агрегатора 7 источников, а также выявлены ближайшие аналоги. По сумме значений критериев ближе всего являются системы: Leader-ID, Codenrock, ФИЗТЕХ ДЖЕНЕЗИС, Devpost, Хакатоны.ру и Kaggle. Однако все эти сервисы не являются полным аналогом по запланированной системе.

Так же были выявлены интересные особенности, которые можно учитывать при создании системы:

- рулетка проектов – представление подбора проектов с элементами геймификации;
- проведение челленджей – мероприятие, где необходимо выполнить задание-вызов, чтобы оказаться на доске лидеров.

Подводя итог, выполненного анализа, стоит упомянуть о важности проведенной работы: дополнительное изучение предметной области, сбор источников хакатонов для агрегации, выявление особенностей в других системах.

2 Постановка задачи и формирование требований к системе

В данном разделе описаны различные требования к системе, которые основаны на цели будущей системы, проведенном анализе предметной области и выявленных особенностей аналогичных систем. Требования к системе были разработаны в рамках написания технического задания по межгосударственному стандарту [2].

2.1 Постановка задачи

Система предназначена для увеличения информированности начинающих ИТ специалистов о возможностях получения первичного опыта. Основным назначением системы является предоставление информации о проектах, мероприятиях и хакатонах, а также объединение людей в команды. Система должна предоставлять возможность вступать в команды для получения опыта разработки. Так же система должна обеспечивать возможность хранения информации о пользователе для лучшего взаимодействия между пользователями. Система должна мотивировать пользователей на участие с помощью соревнований и системы достижений.

В рамках проекта автоматизируется деятельность в следующих бизнес-процессах:

- поиск проекта для получения навыков работы в команде;
- поиск хакатона для получения навыков разработки через «соревновательный дух»;
- поиск команды для совместной работы;
- проведение мероприятий направленных на повышение компетенций командной деятельности.

Система создается с целью **сопровождения процесса повышения уровня компетенций в командной разработке посредством нетворкинга**.

В результате создания системы должны быть улучшены значения следующих показателей:

- время, затрачиваемое на приобретение полезных связей в ИТ-сфере;
- время поиска команды/участников в команду;
- время поиска проектов;
- время поиска хакатонов;
- время поиска мероприятий;
- время осваивания специалистов в новом коллективе;
- время принятия решения при принятии на работу;
- время поиска работы.

2.2 Требования к системе в целом

В Системе предлагается выделить следующие функциональные подсистемы:

- подсистема хакатонов, которая предназначена для реализации процесса поиска/подбора хакатона, а также поиска команды или участника (подсистема будет собирать информацию с помощью агрегации с других сайтов информации о хакатонах);

– подсистема проектов, которая должна предоставлять возможность подбора идей пет-проектов и сообщников для реализации пет-проектов (подразумевает хранение информации о результате участия в проекте каждого участника);

– подсистема мероприятий, которая включает в себя создание конкурсов, проведение мероприятий и отслеживание успехов пользователей для мотивации получения опыта;

– подсистема личного кабинета, которая предназначена для формирования информации о пользователе, его опыте и достижениях, для подбора пет-проектов и хакатонов (включает в себя так же механику достижений, а именно назначение достижений);

– подсистема администрирования, которая помогает управлять проектами, агрегатором и контента проекта в целом (должна быть доступна только для администратора, и нужна для модерации поданных заявок, учета статистики платформы).

Система должна быть веб-приложением, что позволит обеспечить доступ с различных устройств из любой точки мира. Система должна состоять из трех приложений в виде сервера, агрегатора и клиента. Клиент должен содержать программу-клиент, которая предоставляет для пользователя интерфейс, посылает запросы на сервер и получает ответы на запросы. В то же время, сервер должен хранить данные и содержит программу, которая отвечает за логику обработки данных из поступивших запросов. Агрегатор должен взаимодействовать с сайтами-источниками и запускаться сервером.

Источниками данных для подсистемы агрегатора хакатонов должны быть:

- <https://leader-id.ru/> ;
- <https://hackathons.pro/> ;
- <https://www.хакатоны.рф/> ;
- <https://codenrock.com/> ;
- <https://pgenesis.ru/events> ;
- <https://devpost.com/> ;
- <https://хакатоны.рус/>.

Данный перечень может редактироваться и пополняться после реализации подсистемы администрирования, а именно, управления агрегатором. В данном разделе подсистемы необходимо предоставлять возможность пользователю просматривать список источников, редактировать и добавлять новые источники.

2.2 Требования к функциям, выполняемым системой

Система должна включать 5 подсистем, определенных в предыдущем пункте. В данном подразделе описана детализация подсистем с описанием функциональности.

Форма представления выходной информации для всех функций определена структурой сайта.

Подсистема «Проекты» должна автоматизировать процесс поиска команды или идеи для пет-проекта. Функции, которые должны быть обязательно реализованы в первой версии:

- просмотр списка проектов с помощью фильтрации по тегам, по типу проектов;
- поиск проектов с помощью поисковой строки;
- просмотр информации о проекте: название; описание; изображения; тип проекта (的独特ый/учебный); список команд, которые его выполняют/выполнили;
- отправление заявки на создание проекта администратору;
- отправление заявки на вступление в команду для выполнения конкретного проекта;
- рассмотрение заявки на вступление в команду создателем команды или любым из участников (при создании команды необходимо выбрать роль участника, который будет рассматривать заявки);
- фиксация результатов деятельности команды по проекту в виде прикрепления ссылки на готовое решение и написания отзыва о проделанной работе.

Функции, которые необходимо реализовать в следующих версиях:

- оценка результата деятельности команды другими участниками с помощью выставления оценки «нравится/не нравится»;
- рулетка проектов – альтернативный инструмент подбора проектов.

Подсистема «Хакатоны» должна автоматизировать процесс поиска хакатона и команды или участников в команду. Функции, которые должны быть обязательно реализованы в первой версии:

- просмотр списка хакатонов с помощью фильтрации по тегам, по датам начала регистрации, по размеру команды;
- просмотр информации о хакатоне: описание, даты, описание приза;
- переход на другой ресурс для записи на хакатон;
- агрегация хакатонов со сторонних ресурсов с помощью библиотек Python для парсинга веб-сайтов.

Подсистема «Мероприятия» должна мотивировать участников на выполнение каких-либо требований с целью получения приза. Мероприятия создаются администратором или могут быть предложены участниками. Функции, которые должны быть обязательно реализованы в первой версии:

- просмотр списка мероприятий;
- участие в мероприятии (функции аналогичны участию в проекте);
- просмотр результатов мероприятия (отображение победителей на странице мероприятия).

Функции, которые необходимо реализовать в следующих версиях:

- доски со списком лидеров по достижениям;
- проведение мероприятия - хакатона с помощью системы;
- оценка результатов команд, экспертами с помощью выставления оценок.

Подсистема «Личный кабинет» должна хранить информацию о пользователях и с ее помощью пользователь будет получать доступ ко всему с ним связанному. Функции, которые должны быть обязательно реализованы в первой версии:

- регистрация с помощью email, пароля и логина пользователя;
- авторизация с помощью email и пароля;
- восстановление пароля с помощью отправления письма на почтовый адрес;
- заполнение профиля после регистрации: информация о пользователе, контактная информация;
- просмотр личного кабинета: информация о профиле, список достижений пользователя;
- просмотр команд, в которых состоит участник, в личном кабинете;
- просмотр проектов, которые создавал участник, в личном кабинете;
- переход к странице проекта из списка проектов личного кабинета;
- переход к странице команды из списка команд личного кабинета;
- подписка на рассылку.

Функции, которые необходимо реализовать в следующих версиях:

- уведомления пользователя;
- возможность регистрации через другие ресурсы;
- интеграция данных о пользователе с другими ресурсами.

Подсистема «Административная панель» должна быть панелью управления проектами, агрегатором и контентом Системы. Подсистема должна быть доступна только для администратора. Функции, которые должны быть обязательно реализованы в первой версии:

- просмотр списка поданных на рассмотрение заявок для создания проектов;
- рассмотрение заявки на создание проекта в административной панели, в результате проект приобретает статус (создан, не создан);
- создание хакатона вручную;
- просмотр статистики площадки, а именно посещаемости и активности;
- создание мероприятия администратором;
- завершение мероприятия, в том числе оценка результатов команд экспертами и публикация результатов мероприятия на странице мероприятия.

Функции, которые необходимо реализовать в следующих версиях:

- управление источниками для агрегатора хакатонов;
- управление частотой сбора информации агрегатора;
- рассмотрение заявок на создание мероприятия от участников;
- архивация проектов из-за низкого интереса.

2.3 Требования к ролям и правам доступа

В таблице 2.1 приведено разграничение прав доступа между 3 пользователями: гость, участник и администратор. Каждый пользователь

обладает правами предыдущего, и расположены они по возрастанию от меньшей группы к большей.

Таблица 2.1 – Роли и функции

Роль	Подсистема	Функция
Гость	Проекты	Просмотр списка проектов с помощью фильтрации по тегам, по типу проектов
		Поиск проектов с помощью поисковой строки
		Просмотр информации о проекте: название; описание; изображения; тип проекта (уникальный/учебный); список команд, которые его выполняют/выполнили
		Рулетка проектов – альтернативный инструмент подбора проектов
	Хакатоны	Просмотр списка хакатонов с помощью фильтрации по тегам, по датам начала регистрации, по размеру команды
		Просмотр информации о хакатоне: описание, даты, описание приза
	Хакатоны	Переход на другой ресурс для записи на хакатон
Гость	Мероприятия	Просмотр списка мероприятий
		Просмотр результатов мероприятия (отображение победителей на странице мероприятия).
		Доски со списком лидеров по достижениям
	Личный кабинет	Регистрация с помощью email, пароля и логина пользователя
		Авторизация с помощью email и пароля
		Возможность регистрации через другие ресурсы;
	Проекты	Отправление заявки на создание проекта администратору
		Отправление заявки на вступление в команду для выполнения конкретного проекта
Участник	Проекты	Рассмотрение заявки на вступление в команду создателем команды или любым из участников (при создании команды необходимо выбрать роль участника, который будет рассматривать заявки)
		Фиксация результатов деятельности команды по проекту в виде прикрепления ссылки на готовое решение и написания отзыва о проделанной работе

Продолжение таблицы 2.1

Роль	Подсистема	Функция
Участник	Проекты	Оценка результата деятельности команды другими участниками с помощью выставления оценки «нравится/не нравится»
	Мероприятия	Участие в мероприятии (функции аналогичны участию в проекте)
		Оценка результатов команд, экспертами с помощью выставления оценок
	Личный кабинет	Восстановление пароля с помощью отправления письма на почтовый адрес;
		Просмотр проектов, которые создавал участник, в личном кабинете
		Заполнение профиля после регистрации: информация о пользователе, контактная информация;
		Просмотр личного кабинета: информация о профиле, список достижений пользователя
		Просмотр команд, в которых состоит участник, в личном кабинете
		Переход к странице проекта из списка проектов личного кабинета
		Переход к странице команды из списка команд личного кабинета
		Подписка на рассылку
		Уведомления пользователя
		Интеграция данных о пользователе с другими ресурсами
Администратор	Административная панель	Проведение мероприятия - хакатона с помощью системы
		Просмотр списка поданных на рассмотрение заявок для создания проектов
Администратор	Административная панель	Рассмотрение заявки на создание проекта в административной панели, в результате проект приобретает статус (создан, не создан)
		Создание хакатона вручную
		Просмотр статистики площадки, а именно посещаемости и активности
		Создание мероприятия администратором
		Завершение мероприятия, в том числе оценка результатов команд экспертами и публикация результатов мероприятия на странице мероприятия

Продолжение таблицы 2.1

Роль	Подсистема	Функция
Администратор	Административная панель	Управление частотой сбора информации агрегатора
		Рассмотрение заявок на создание мероприятия от участников
		Архивация проектов из-за низкого интереса
		Управление источниками для агрегатора хакатонов

2.4 Требования к информационному обеспечению

2.4.1 Требования к составу, структуре и способам организации данных в системе

В системе должна храниться информация о следующих объектах: проект, хакатон, мероприятие, пользователь, команда. Детализация информации об объектах показана в таблице 2.2.

Таблица 2.2 – Структура данных

Объект	Данные
Проект	Название, теги, тип (учебный/уникальный), картинки, описание
Хакатон	Название, теги, даты начала и окончания, дата завершения регистрации, картинки, описание, описание призов, ссылка на ресурс для регистрации
Мероприятие	Название, теги, даты начала и окончания, дата завершения регистрации, картинки, описание, описание призов, список экспертов, списки команд, оценки экспертов для каждого решения
Пользователь	ФИО, контактные данные (email), список достижений, опыт, пароль, фотография
Команда	Создатель команды, список ролей в команде, пользователи с их ролью в команде, результат участия команды в проекте/хакатоне/мероприятие (ссылка, отзыв), описание команды, название команды

Логическая и физическая модель данных должна быть определена разработчиком на стадии «Проектирование. Разработка технического проекта».

2.4.2 Требования к информационной совместимости со смежными системами

Источники для агрегации информации о хакатонах определены в пункте 2.1.

Состав данных для осуществления информационной агрегации по каждому источнику должен быть определен разработчиком на стадии «Проектирование. Разработка технического проекта».

2.4.3 Требования по применению СУБД

Для реализации подсистемы хранения данных должна использоваться свободная объектно-реляционная система управления базами данных PostgreSQL - версия 14.

2.4.4 Требования к структуре процесса сбора, обработки, передачи данных в системе и представлению данных

Система при вводе в действие должна содержать теги, которые команда разработчиков должна самостоятельно сформировать, потому что они разбираются в данной предметной области.

Первые несколько учебных проектов необходимо создать в системе перед вводом в действие. Предполагается, что эти проекты заполнит команда разработки.

Остальные процессы сбора, обработки и передачи данных в системе определяются на этапе «Проектирование».

2.4.5 Требования к защите данных от разрушений при авариях и сбоях в электропитании системы

Информация в базе данных системы должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания на стороне арендованного сервера.

Резервное копирование данных должно осуществляться на регулярной основе, в объемах, достаточных для восстановления информации в подсистеме хранения данных.

2.5 Требования к верстке

Система должна отвечать требованиям кроссбраузерности, а именно, поддерживаться во всех популярных браузерах в последних версиях:

- Internet Explorer версии 10 и выше, платформа — операционные системы семейства Windows;
- Mozilla Firefox версии 28 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
- Safari версии 6.1 и выше, платформа — Mac OS X версии 10.8 и выше;
- Google Chrome версии 21 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
- Opera версии 15 и выше, платформа — Windows версии XP и выше и Mac OS X версии 10.8 и выше;
- браузеры мобильных устройств iOS 7 и выше;
- браузеры мобильных устройств Android 5 и выше.

При верстке должны учитываться особенности браузеров, их ограничения внешнего вида и поведения объектов. Для обеспечения правильного отображения элементов в разных браузерах может применяться грациозная деградация.

Система должна разрабатываться в соответствии с современным методом верстки БЭМ – «блок – элемент – модификатор». Необходимо соблюдать единые правила верстки, которые помогают быстро разрабатывать интерфейсы, гибко их настраивать и легко модифицировать.

Верстка в системе должна быть адаптивной, а именно корректно отображаться на любом устройстве со следующими расширениями: 2560x1440; 3840x2160; 1024x600; 1024x768; 1152x864; 1280x800; 1280x1024; 1440x900; 1920x1080; 1680x1050.

Остальные требования:

- при размерах более 375px не должно возникать горизонтальной прокрутки страницы (за исключением отдельных блоков верстки, где подобная реализация предусмотрена дизайн-макетами);
- все изображения должны сопровождаться описанием при проблемах загрузки;
- логотип на внутренних страницах должен вести на главную;
- у каждой страницы должен быть свое уникальное поле title;
- изображения должны масштабироваться в зависимости от размера окна;
- картинки должны быть максимально сжаты, должны использоваться современные форматы изображений;
- весь важный функционал сайта должен быть доступен без js-скриптов;
- анимация не должна быть сложной.

2.6 Требования к структуре сайта

Структура сайта (рис. 2.1) и основные компоненты определены в визуальном конструкторе и планировщике веб-сайтов – Octopus.do [3].

Действия обычного пользователя начинаются с главной страницы, где содержатся краткие сведения о проекте. Далее с помощью навигационного меню пользователь может перейти на страницу одной из подсистем (хакатоны, проекты, конкурсы) или зарегистрироваться/авторизоваться.

Действия администратора проекта начинаются с главной страницы со статистикой, и с помощью навигационного меню администратор может перейти на страницы с управлением проектами, хакатонами, конкурсами.

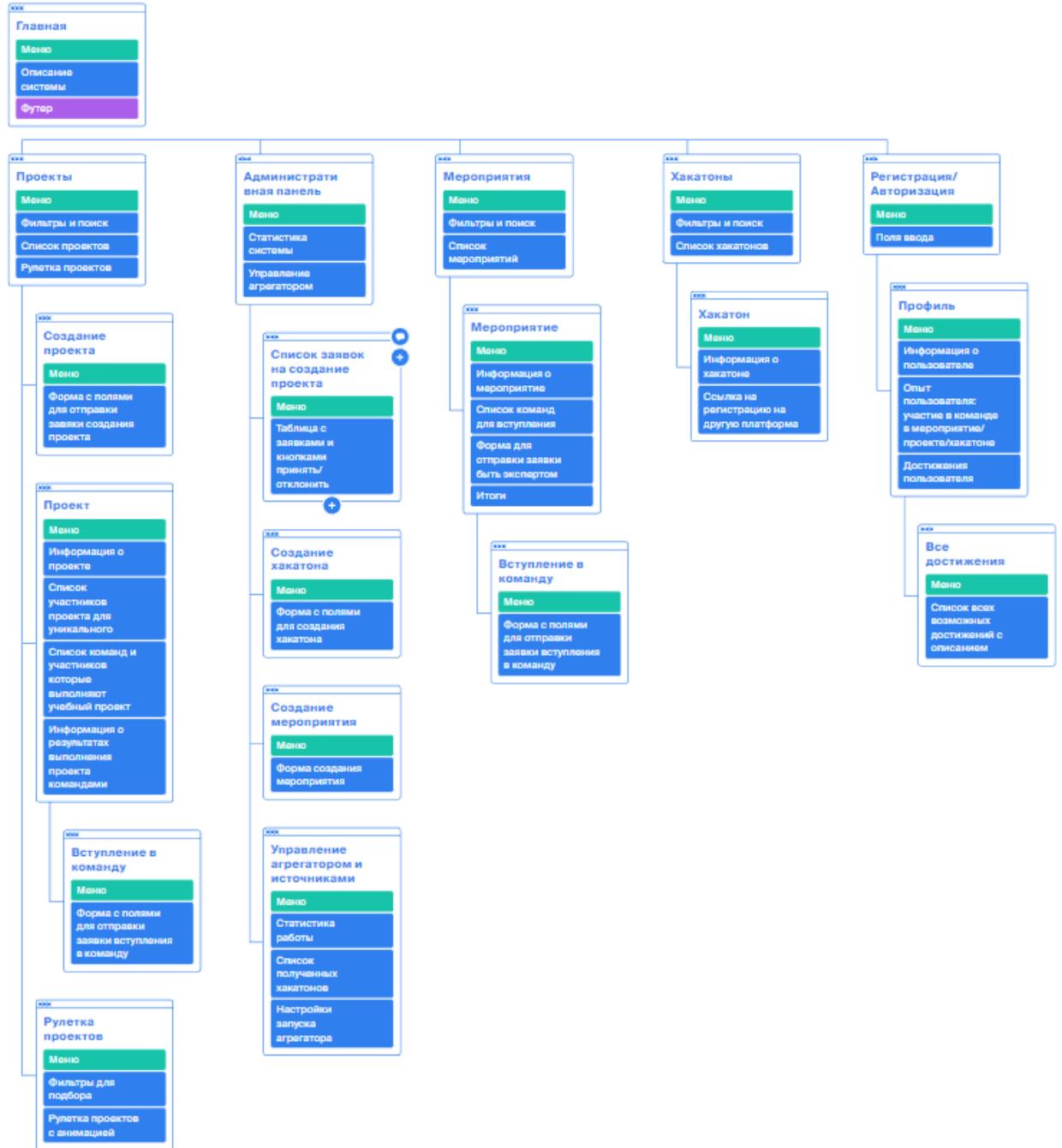


Рисунок 2.1 – Структура сайта

2.7 Требования к техническому обеспечению

Основные требования к хостингу:

- система должна быть расположена на виртуальном сервере VPS;
- объём дискового пространства – от 1GB;
- сервер должен быть с операционной системой Ubuntu;
- сервер должен быть с защитой от DDoS и блокиратором спама, а также с системой резервного копирования данных для восстановления в случае непредвиденных ситуаций и ошибок при обновлении.

2.8 Требования к составу и содержанию работ создания системы

Система должна быть реализована за 7,5 месяцев, в этот период необходимо выполнить анализ, проектирование, реализацию и ввод в действие. Некоторые процессы необходимо выполнять параллельно.

Таблица 2.3 – Стадии работ по созданию системы

Стадия	Результат	Сроки
Анализ предметной области, формирование требований и разработка технического задания.	Собрана и обработана информация о: – предметной области получения первичного опыта ИТ специалистов; – целевой аудитории и ее потребностях; – сущности пет-проекты; – сущности хакатоны. Определены источники данных. Техническое задание и приложения к нему	01.10.2022 – 14.12.2022
Проектирование. Создание технического проекта.	Архитектура системы, модели данных, сценарии использования системы, спроектированные интерфейсы пользователя	15.12.2022 – 12.01.2023
Реализация.	Разработка системы и ее документирование	12.01.2023 – 22.05.2023
Ввод в действие.	Работающий в интернете сайт	27.04.2023- 22.05.2023
Сопровождение.	Система работает стабильно, без сбоев. Выявленные недостатки устранены, постоянная модерация администратором.	27.04.2023

Для создания системы необходимо придерживаться итеративной модели разработки. Стадия проектирования и реализации, обязательно должны выполняться в каждой итерации для определённой цели. В разделе 2.2 определены функции, которые необходимо реализовать в каждой подсистеме. Список функций разделен на первую и последующие версии. В таблице 2.4 показаны сроки двух итераций проектирования и реализации системы. Разбиение на итерации необходимо для более раннего ввода в действие системы с минимальным функционалом. На второй реализации необходимо реализовать оставшиеся функции.

Таблица 2.4 – Итерации проектирования и реализации

Итерация проектирования и реализации	Цель	Длительность	Даты
1	Реализация функций первой версии	2,5 месяца	12.01.2023 - 12.04.2023
2	Реализация остальных функций	1,5 месяца	12.04.2023 - 27.04.2023

4 Проектирование системы

4.1 Обоснование выбора инструментария

В данном разделе приводится описание и аргументация выбора программных инструментов, используемых при проектировании и реализации системы.

Для построения логической модели хранилища данных и вариантов использования был выбран сервис Draw.io. Это бесплатное онлайн-приложение для создания диаграмм рабочих процессов, организационных, сетевых. Обладает современным и понятным интерфейсом, а также позволяет сохранить проект в форматах JPG, PNG и другие. Именно за эти особенности и был выбран данный инструмент.

Моделирование бизнес-процессов необходимо выполнить с использованием нотации IDEF0. Она предназначена для описания процессов с помощью определения логической последовательности и композиции задач. Моделирование необходимо производить по методологии функционального моделирования [4], определенной Госстандартом России. Для построения модели необходимо использовать бесплатную образовательную версию коммерческого программного обеспечения Ramus – Ramus Educational [5]. Он позволяет пользователям создавать диаграммы в форматах IDEF0 и DFD, а также поддерживает экспорт файлов в формате изображений.

Архитектура системы спроектирована с помощью модели C4 [6]. Эта модель была выбрана, потому что визуально показывает архитектуру на разных уровнях в понятном виде даже для простого пользователя.

Для разработки макетов пользовательского интерфейса выбран онлайн-сервис Figma, который предназначен для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. Данный онлайн-сервис был выбран, исходя из следующих преимуществ: наличие опыта использования, бесплатный тарифный план для одного пользователя, кроссплатформенность, сохранение истории изменений макетов.

Для программирования был выбран язык программирования TypeScript в связи с наличием опыта разработки, наличием типизации и удобной системы выявления ошибок. Для выполнения кода необходима среда разработки, а именно Node.js.

Исходя из выбранного языка для разработки веб-приложения был выбран фреймворк Vue.js из-за открытого исходного кода, легкого порога входления, высокой производительности, наличия документации, популярности, а также из-за использования компонентов и реактивности. Для разработки на нем необходимо пользоваться документацией и руководством с официального сайта фреймворка [7].

Для создания серверной части так же необходимо использовать фреймворк на основе TypeScript. Среди большого множества был выбран NestJS [8] за открытый исходный код, возможность создания

масштабируемого приложения, модульную архитектуру, шаблоны проектирования и адаптируемую экосистему.

Для разработки потребуется текстовый редактор для написания кода. В качестве текстового редактора будет использована такая программа как Visual Studio Code. Она бесплатна, имеет множество возможностей и может быть установлена как на Windows, так и на Linux и MacOS.

Программа Postman будет использована для тестирования работы API, а также для отправки запросов POST и GET. Она имеет графический интерфейс, поэтому легко осваивается даже новичками.

Так же в клиентской части системы будет использован Bootstrap [9]. Это свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Для работы с базой данных необходимо выбрать систему управления базами данных. СУБД – это совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. Среди множества СУБД была выбрана объектно-реляционная система управления базами данных PostgreSQL. Использование PostgreSQL бесплатно, она является одной из наиболее популярных. По этим причинам и был сделан выбор.

4.2 Моделирование бизнес-процессов (ТО-ВЕ)

В данном разделе будет выполнено моделирование бизнес-процессов с помощью графической нотации IDEF0 для формализации и описания логики разрабатываемой системы.

На рисунке 4.1 представлена модель «Система повышения уровня компетенций в командной разработке» уровня А0, где на вход поступает интерес/инициатива, информация о пользователе и информация о проектах, хакатонах, мероприятиях. Интерес и инициатива пользователя являются главным входным элементом, потому что без него пользователь не будет выполнять действия в системе. Информация о пользователе включает в себя контактные данные, пароль, логин, ФИО, дату рождения и т. д. Вся система должна работать благодаря двум ресурсам: пользователь и сайты с хакатонами. За счет управления правилами системы и пользовательским соглашением получается результат: информация об источниках получения опыта, достижения и опыт.

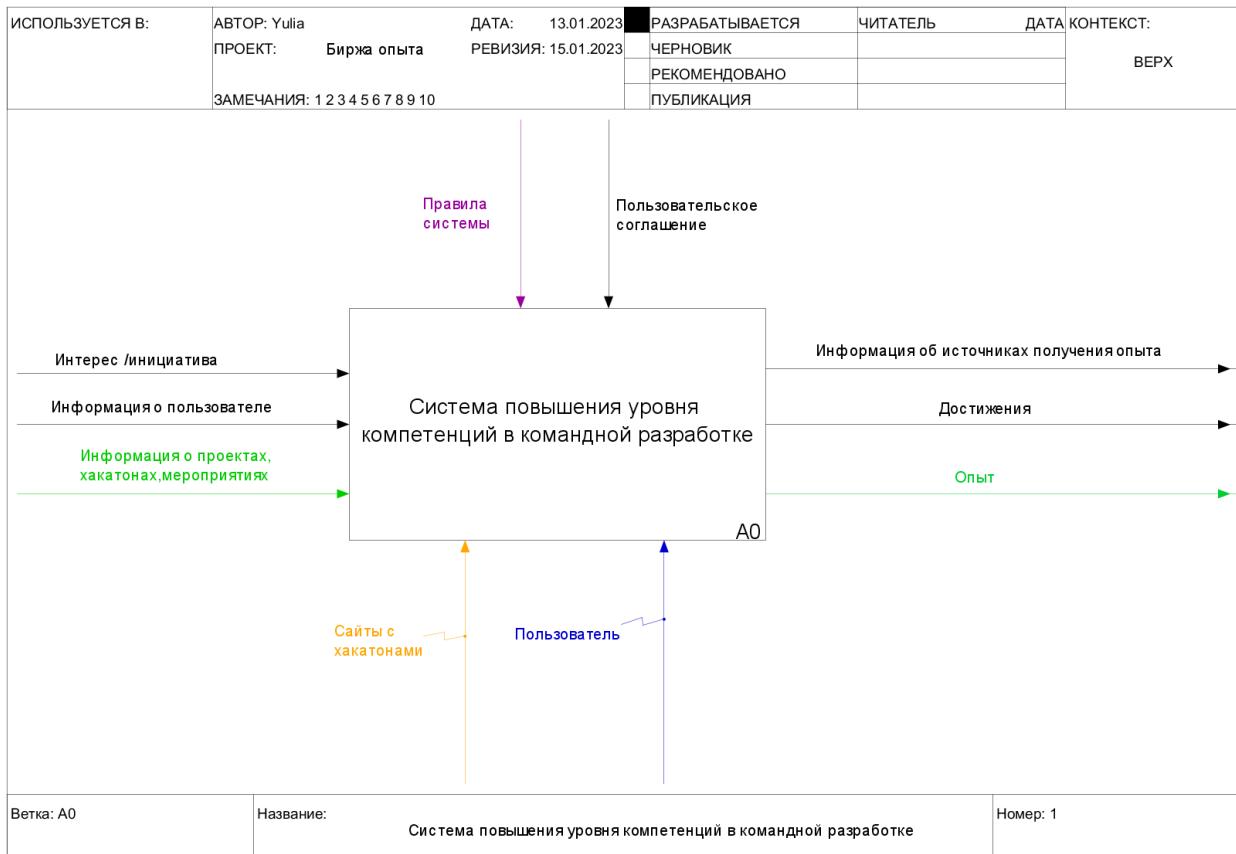


Рисунок 4.1 – Концептуальный уровень модели «Система повышения уровня компетенций в командной разработке»

На рисунке 4.2 изображена детализация концептуального уровня модели. Детализация содержит 4 процесса: авторизация, формирование контента, поиск активности и участие в ней. Во всех процессах участвует пользователь. Для формирования контента необходимы сайты с хакатонами. Всеми процессами управляет правила системы, а процессом авторизации еще и пользовательское соглашение.

Процесс авторизации необходим для разграничения в системе гостя, участника и администратора. Далее, все три роли распределяются по процессам и являются входными элементами для них. Тем самым исходя из модели видно права доступа пользователей в каждом процессе.

Процесс формирование контента отвечает за преобразование информации о проектах, мероприятиях и хакатонах в контент внутри системы.

Далее контент необходим для процесса поиска активности. Под активностью подразумевается либо проект, либо мероприятие, либо хакатон. В процессе поиска активности пользователь все равно получает информацию об источниках получения опыта, даже если не захочет никогда участвовать.

Все активности в системе поступают в процесс участия. После участия в проекте, мероприятии или хакатоне у пользователя появляется опыт командной разработки, тем самым уровень компетенций повышается. Приятным бонусом будут являться достижения пользователя после участия в мероприятиях.

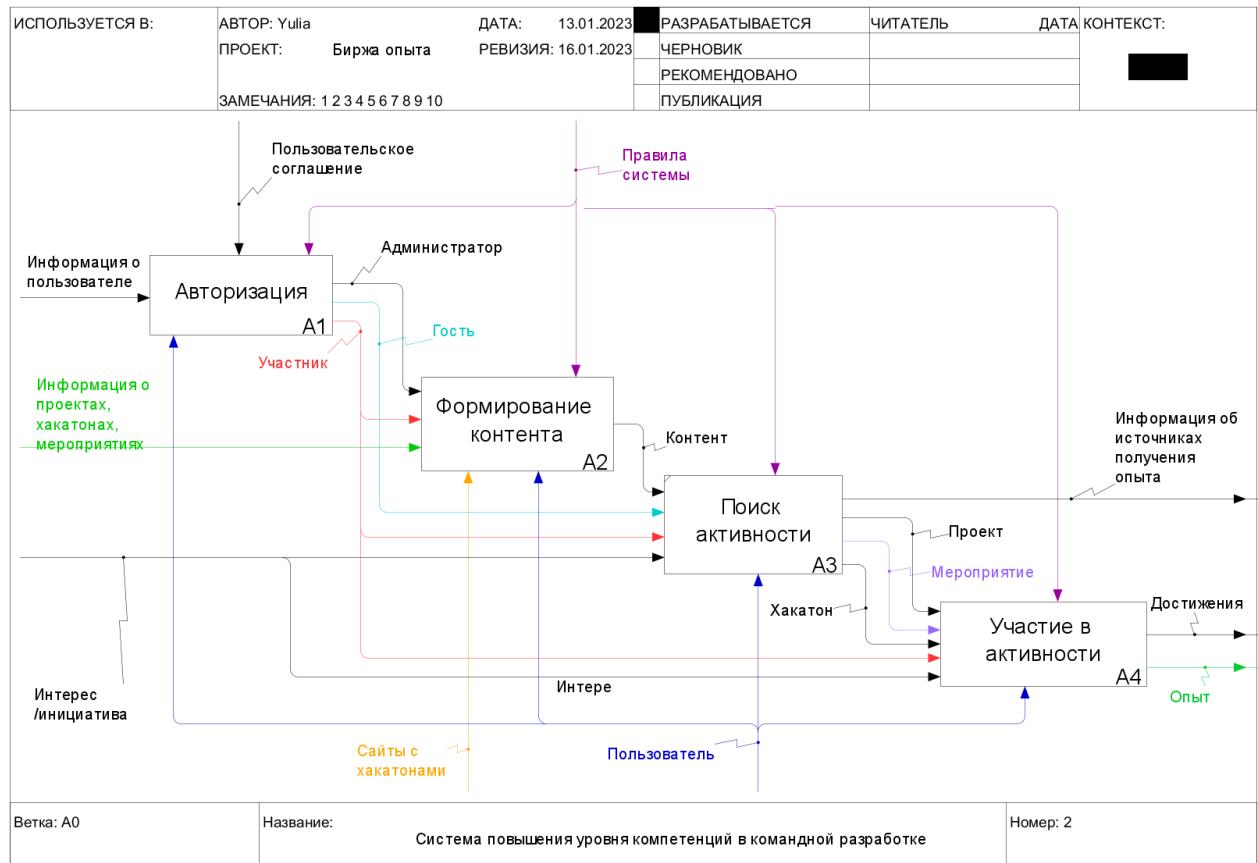


Рисунок 4.2 – Детализация концептуальной модели

Детализация процесса «Авторизация» показана на рисунке 4.3. Процесс состоит из трех подпроцессов: регистрация, вход в аккаунт и выход из аккаунта. При регистрации необходимо пользовательское соглашение. Правила системы управляют всеми процессами. При регистрации информация о пользователе превращается в аккаунт пользователя, который хранится в системе. Без аккаунта и информации о пользователе (email и пароль) невозможен вход в аккаунт. После входа в аккаунт в системе появляются участник или администратор, которые могут при необходимости выйти из системы и стать гостем.

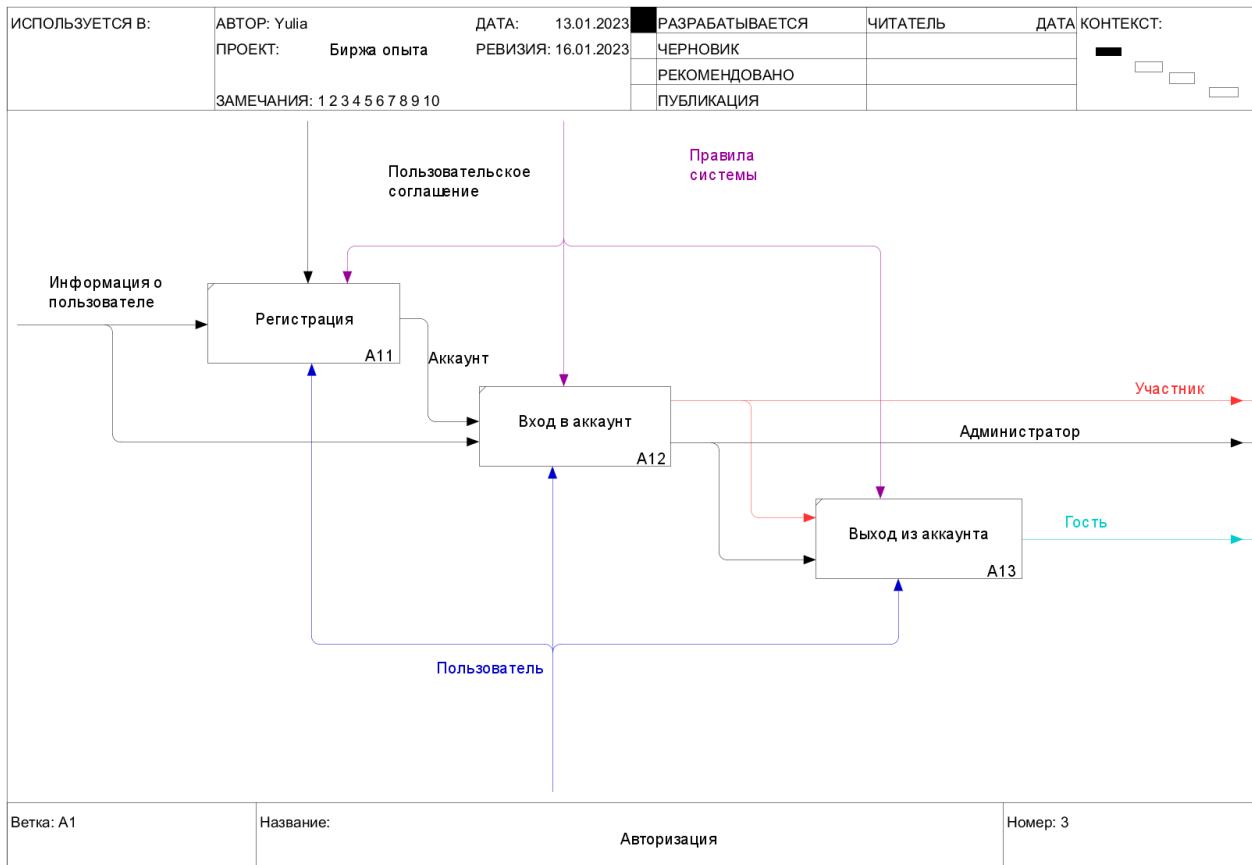


Рисунок 4.3 – Детализация процесса «Авторизация»

На рисунке 4.4 показана детализация процесса «Формирование контента». Под контентом подразумеваются проекты, хакатоны и мероприятия. Проект создается через отправление заявки на создание проекта участником и рассмотрение заявки администратором. Так же администратор создает мероприятия, хакатоны и управляет агрегатором. Для работы агрегатора необходимы сайты с хакатонами.

Детализация процесса «Участие в активности» на рисунке 4 содержит 3 подпроцесса: участие в проекте, участие в мероприятие и переход на другой сайт для участия в хакатоне.

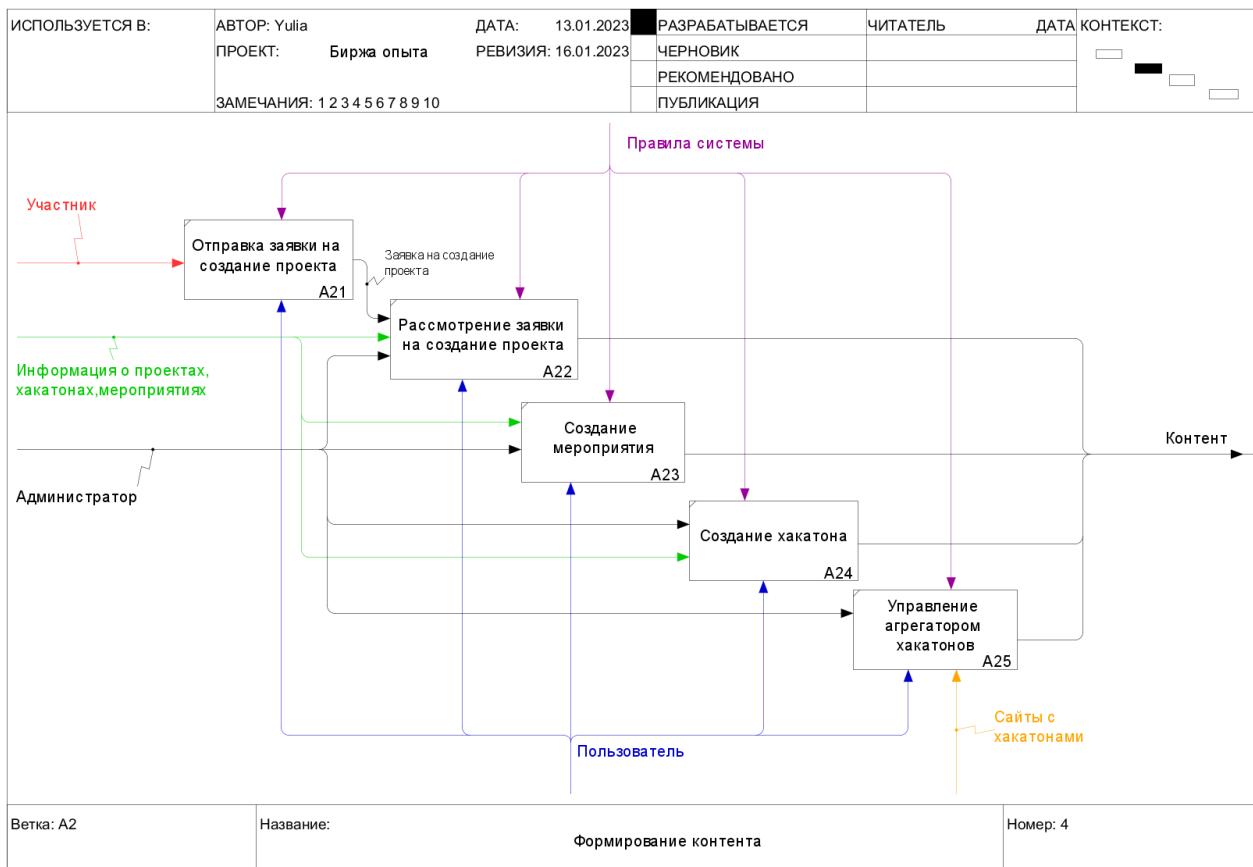


Рисунок 4.4 – Детализация процесса «Формирование контента»

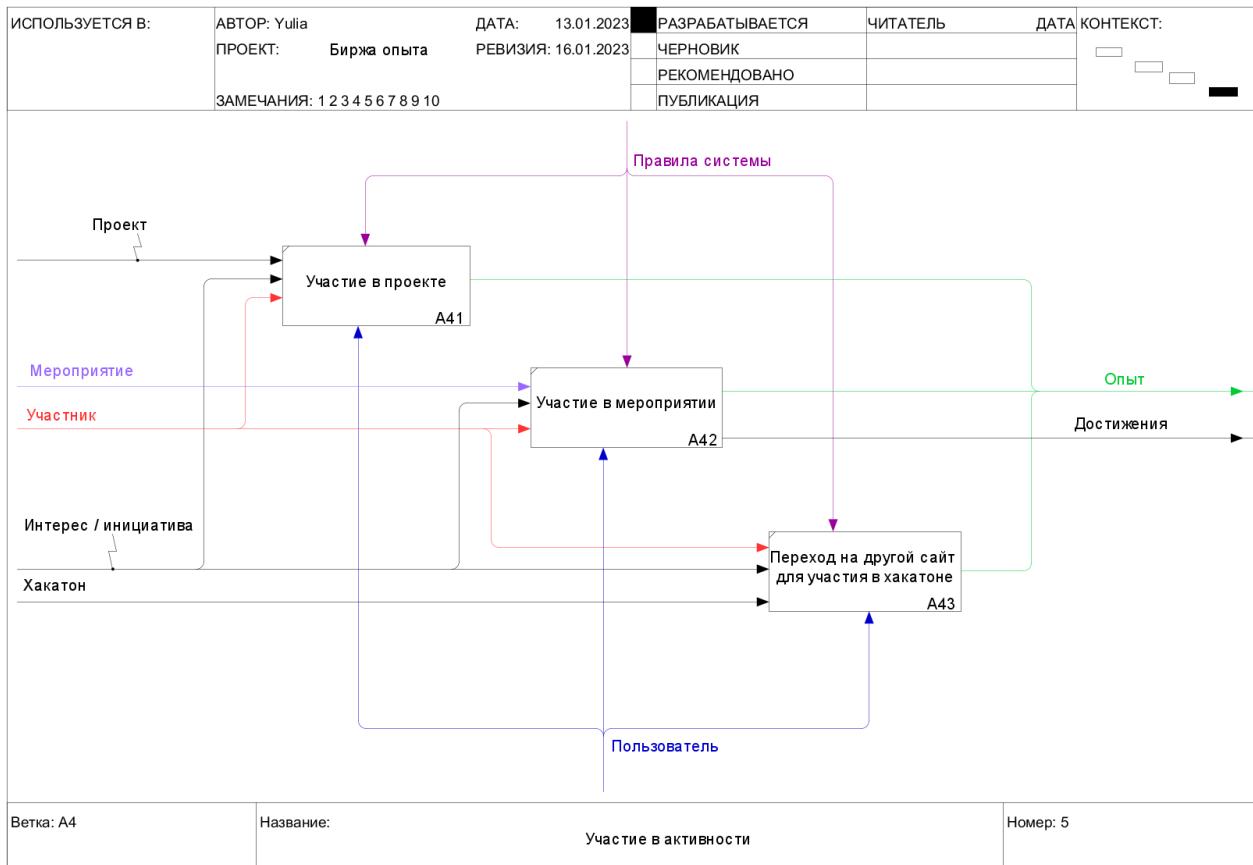


Рисунок 4.5 – Детализация процесса «Участие в активности»

4.3 Проектирование вариантов использования

В данном разделе описаны постановки задач и подзадач с помощью сценариев использования в нотации UML и спецификаций вариантов использования. В процессе разработки необходимо ориентироваться на данные сценарии, они показывают работу системы при определённых действиях пользователя. Выделено 4 задачи, с помощью которых пользователь сможет достигнуть тех или иных целей, которые возможны в рамках системы «Биржа опыта»:

- создание проекта;
- участие в проекте;
- участие в хакатоне;
- проведение мероприятия.

4.3.1 Вариант использования «Создание проекта»

Сценарий варианта использования «Создание проекта» показан на рисунке 4.6. Система должна позволять создавать проект только авторизованному пользователю. Создание проекта предполагает следующую последовательность действий: авторизованный пользователь подает заявку на создание проекта из страницы с проектами, затем администратор системы проверяет заявку и принимает решение о создании или отказе в создании проекта. Данная задача предполагает 3 пользователей (гость, участник и администратор) и реализацию как в серверной, так и в клиентской части.

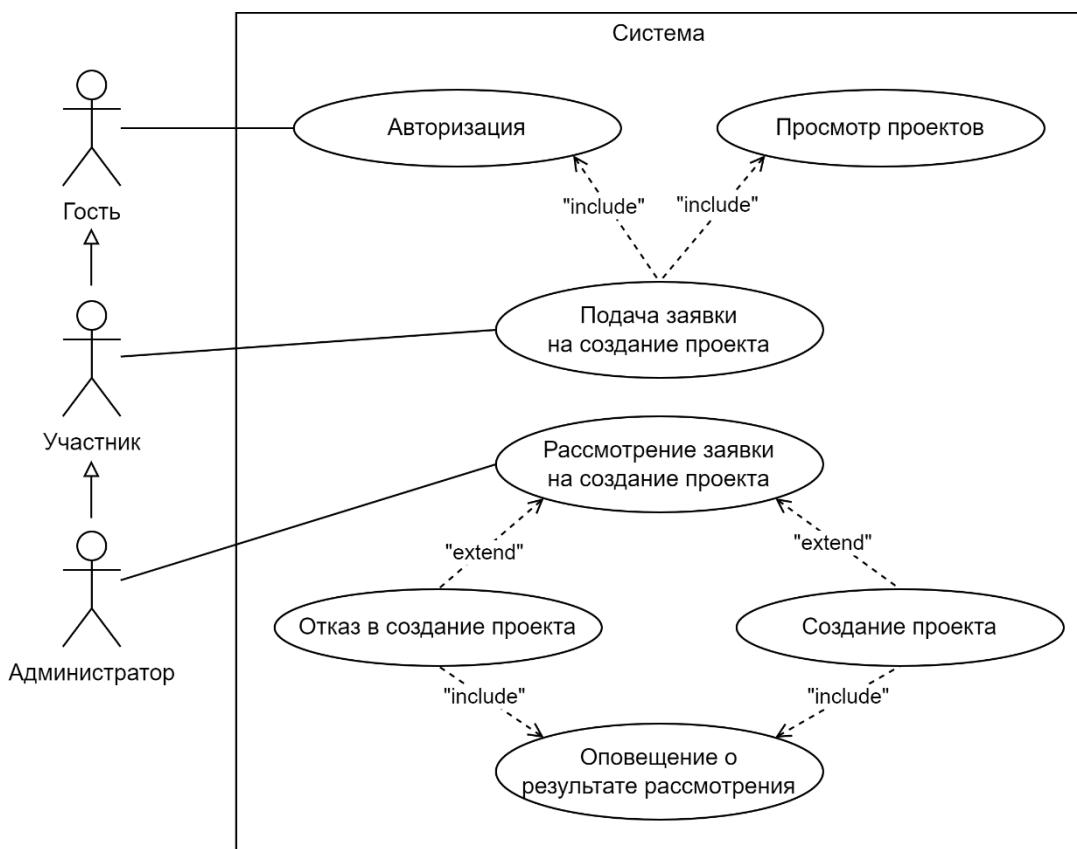


Рисунок 4.6 – Сценарий варианта использования «Создание проекта»

Спецификация варианта использования «Подача заявки на создание проекта».

Цель: подать заявку на создание проекта

Активные субъекты: Участник.

Краткое описание: Участник системы создает заявку на создание проекта.

Предусловия: Участник должен авторизоваться в системе.

Постусловия: должен быть выполнен вариант использования «Рассмотрение заявки на создание проекта».

Основной поток событий:

1. Система публикует страницу с проектами, на которой находится кнопка «Создать проект». Если пользователь не авторизован после нажатия кнопки система предлагает авторизоваться.
2. Активный субъект нажимает кнопку создания проекта.
3. Система показывает пользователю информацию о том, что ему нужно создать заявку, которую рассматривает администратор проекта и публикует форму с полями: название, тип проекта, описание, фотографии, теги проекта, максимальный размер команды. Поле тип проекта имеет значения «учебный» или «уникальный». Теги для проекта выбираются из уже существующих в системе.
4. Активный субъект заполняет поля формы.
5. Система проверяет данные в форме на валидность.
6. Активный субъект нажимает кнопку «Отправить заявку».
7. Система уведомляет пользователя об успешной или не успешной отправке заявки.
8. Система завершает работу варианта использования.

Альтернативные потоки событий:

1. Система демонстрирует активному субъекту о неправильном заполнении полей в форме.
2. Система демонстрирует активному субъекту сообщение: «Ошибка при отправке заявки».

Сценарий обработки ошибок: при возникновении ошибки отправки заявки на создание поля формы сохраняют свои данные и позволяют пользователю заново отправить заявку.

Спецификация варианта использования «Рассмотрение заявки на создание проекта».

Цель: рассмотреть заявку на создание проекта

Активные субъекты: администратор, участник.

Краткое описание: Администратор рассматривает поданные заявки на создание проекта и принимает решение. В конце система должна уведомить участника о результате рассмотрения заявки.

Предусловия: Администратор должен быть авторизован в системе для просмотра поданных заявок и принятия решения по ним. Участник должен авторизоваться в системе, чтобы увидеть результат заявки.

Постусловия: нет.

Основной поток событий:

1. Система публикует страницу для управления, на которой находятся дашборды. Один из них отвечает за показ статистики по поданным и не рассмотренным заявкам.
2. Администратор нажимает на элемент дашборда со статистикой поданных заявок.
3. Система показывает администратору страницу с заявками на создание проекта. Каждая заявка имеет статус: отклонено, создано, не рассмотрено. Заявки показываются в таблице и если у них статус не рассмотрено, то строка таблицы имеет кнопки «создать» и «отклонить».
4. Администратор нажимает на не рассмотренную заявку.
5. Система открывает страницу проекта с данными из заявки.
6. Администратор читает идею проекта и принимает решение создавать проект или нет.
7. Администратор возвращается на страницу списка заявок.
8. Если администратор принимает решение о создании проекта, то должен нажать на кнопку «Создать». Если администратор не хочет создавать предложенный проект, то ему необходимо нажать на кнопку «отклонить».
9. После нажатия любой из кнопок «создать» и «отклонить» система меняет статус проекта и уведомляет участника о результате рассмотрения заявки по почте.
10. Система завершает работу варианта использования.

Альтернативные потоки событий: нет.

Сценарий обработки ошибок: нет.

Требования к реализации задачи

Запросы, которые необходимо реализовать в REST API:

- /api/projects GET Получение списка проектов;
- /api/project POST Создание проекта;
- /api/admin GET Получение информации для дашбордов;
- /api/admin/project/request GET Получение списка заявок, поданных на создание проекта;
- /api/project/:id POST Изменение статуса проекта при рассмотрении проекта.

Страницы, которые необходимо реализовать в веб-приложении:

- страница «Проекты» с кнопкой создания проекта;
- страница «Подача заявки на создание проекта» с полями;
- страница «Управление системой» с дашбордами;
- страница «Заявки на создание проекта» со списком заявок.

4.3.2 Вариант использования «Участие в проекте»

Сценарий варианта использования «Участие в проекте» показан на рисунке 4.7. Система должна позволять участвовать в проекте только для

авторизованного пользователя. Участие в проекте состоит из следующих подзадач: подача заявки на участие в проекте, рассмотрение заявки на участие в проекте, подведение итогов участия в проекте. В задаче участвуют трое пользователей: гость, участник, который хочет попасть в команду для реализации проекта, и участники, уже состоящие в команде. Реализация предполагается как в серверной, так и в клиентской части.

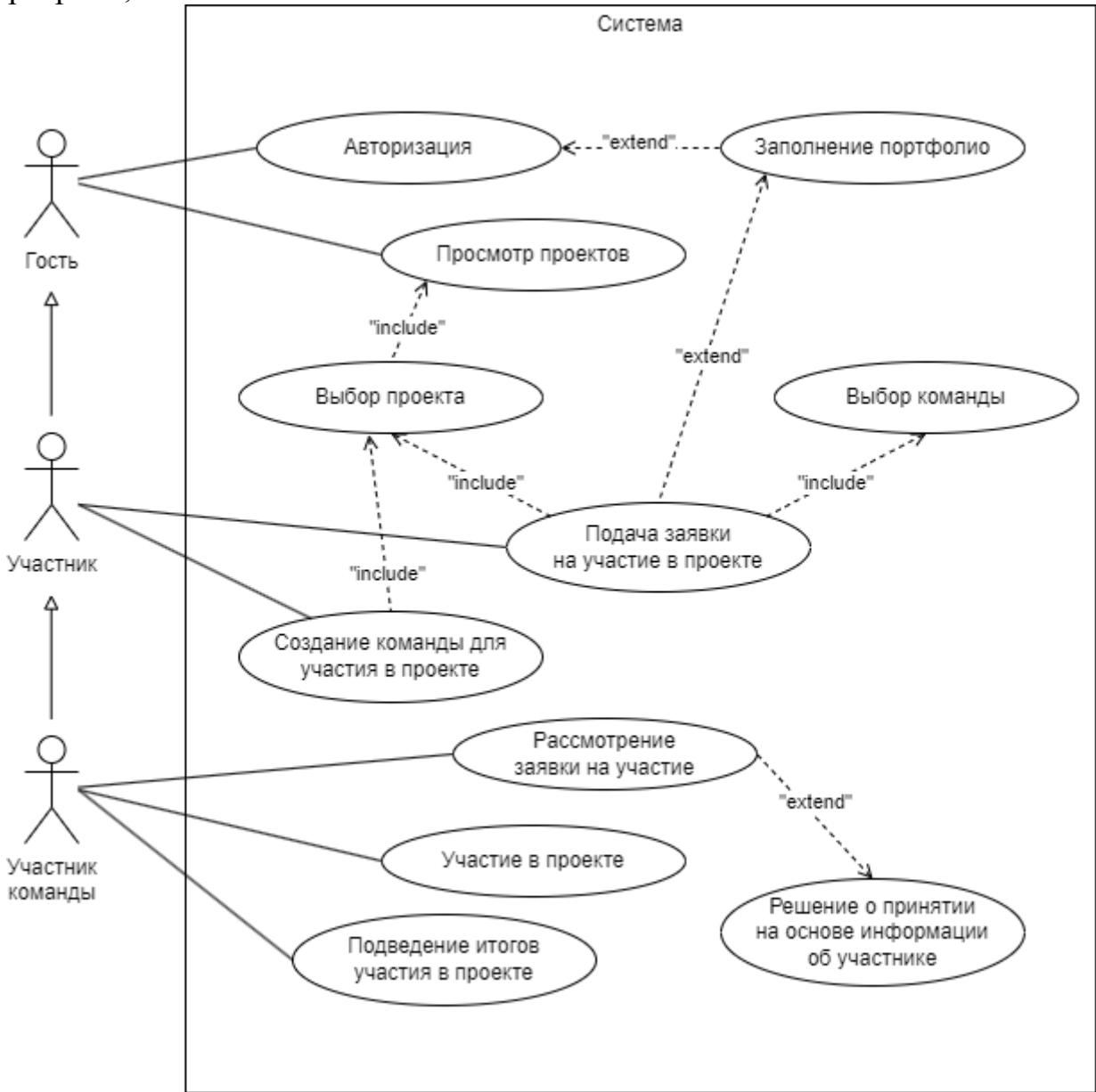


Рисунок 4.7 – Сценарий варианта использования «Участие в проекте»

Спецификация варианта использования «Создание команды для участия в проекте».

Цель: создание команды для участия в учебном проекте.

Активные субъекты: участник, который хочет создать команду.

Краткое описание: участник системы создает команду.

Предусловия: участник должен авторизоваться в системе.

Постусловия: может быть выполнен вариант использования «Рассмотрение заявки на участие».

Основной поток событий:

1. Система публикует страницу с проектами.
2. Активный субъект выбирает проект для участия, открывая проект на отдельной странице с помощью кнопки «Подробнее» на карточке проекта.
3. Система публикует страницу с информацией о выбранном проекте. Если проект учебный, то на странице отображается список команд, которые исходя из статусов участия команды в проекте либо содержат, либо нет кнопку «Подать заявку». Страница проекта должна содержать кнопку «Создать команду».
4. Активный субъект нажимает кнопку «Создать команду».
5. Система показывает пользователю информацию о том, что ему нужно создать команду и публикует форму с полями: название команды, количество участников и роли в команде. Роли в команде выбираются из готового списка, который может редактироваться только администратор.
6. Активный субъект заполняет поля формы.
7. Система проверяет данные в форме на валидность.
8. Активный субъект нажимает кнопку «Создать команду».
9. Система уведомляет пользователя об успешном или не успешном создании команды.
10. Система завершает работу варианта использования.

Альтернативные потоки событий:

1. Система демонстрирует активному субъекту о неправильном заполнении полей в форме.
2. Система демонстрирует активному субъекту сообщение: «Ошибка при создании команды».

Сценарий обработки ошибок: при возникновении ошибки о создании команды, формы сохраняют свои данные и позволяют пользователю заново отправить заявку.

Спецификация варианта использования «Подача заявки на вступление в команду для реализации проекта».

Цель: подать заявку на создание проекта

Активные субъекты: участник, который хочет попасть в команду для реализации проекта.

Краткое описание: участник системы создает заявку на участие в проекте в рамках выбранной команды.

Предусловия: участник должен авторизоваться в системе.

Постусловия: должен быть выполнен вариант использования «Рассмотрение заявки на участие».

Основной поток событий:

1. Система публикует страницу с проектами.
2. Активный субъект выбирает проект для участия, открывая проект на отдельной странице с помощью кнопки «Подробнее» на карточке проекта.
3. Система публикует страницу с информацией о выбранном проекте. Если проект учебный, то на странице отображается список команд, которые

исходя из статусов участия команды в проекте либо содержит, либо нет кнопку «Подать заявку». Если проект уникальный, то на странице проекта указана только одна команда, у которой тоже может быть или не быть кнопка «Подать заявку».

4. Активный субъект выбирает команду и нажимает кнопку «Подать заявку».
5. Система показывает пользователю информацию о том, что ему нужно создать заявку, которую рассматривают участники выбранной команды и публикует форму с полями: желаемая роль в проекте, резюме в виде файла или ссылки. Желаемая роль в проекте выбирается из уже созданных и не занятых ролей в команде.
6. Активный субъект заполняет поля формы.
7. Система проверяет данные на форме на валидность.
8. Активный субъект нажимает кнопку «Отправить заявку».
9. Система уведомляет пользователя об успешной или не успешной отправке заявки.
10. Система завершает работу варианта использования.

Альтернативные потоки событий:

1. Система демонстрирует активному субъекту о неправильном заполнении полей в форме.
2. Система демонстрирует активному субъекту сообщение: «Ошибка при отправке заявки».

Сценарий обработки ошибок: при возникновении ошибки отправки заявки на создание поля формы сохраняют свои данные и позволяют пользователю заново отправить заявку.

Спецификация варианта использования «Рассмотрение заявки на участие в проекте».

Цель: рассмотреть заявку на участие в проекте в определенной команде

Активные субъекты: участник, который хочет попасть в команду для реализации проекта, и участники, уже состоящие в команде.

Краткое описание: Участники, уже состоящие в проекте, рассматривают поданные заявки на участие в их команде и принимают решение. В конце система должна уведомить участника, который хочет попасть в команду для реализации проекта, о результате рассмотрения заявки.

Предусловия: Участники должны быть авторизованы в системе для просмотра поданных заявок и принятия решения по ним. Участник должен авторизоваться в системе, чтобы увидеть результат заявки.

Постусловия: должен быть выполнен вариант использования «Подведение итогов участия в проекте».

Основной поток событий:

1. Система публикует страницу с командами, в которых состоит участник, в рамках личного кабинета. Каждая команда имеет кнопку «Подробнее» и значок, появляющийся при наличии уведомления.
2. Участник, уже состоящий в выбранной команде, нажимает на кнопку «Подробнее» определенной команды.

3. Система показывает участнику, состоящему в команде, страницу с информацией о команде, ее участниках и желающих участвовать, а также сведениями об участии команды в проекте. Каждая заявка на участие имеет статус: отклонена, принята, не рассмотрена. Заявки показываются в таблице с полями: ФИО, желаемая роль, файл или ссылка на резюме. Если у заявки статус не рассмотрена, то строка таблицы имеет кнопки «Принять» и «Отклонить» заявку.
 4. Участник, состоящий в команде, принимает решение о принятии пользователя в команду или нет. Участнику необходимо нажать на кнопку «Принять» или «Отклонить» заявку. Из всех, состоящих в команде участников, это может сделать лишь создатель команды. Отменить действие невозможно.
 5. После нажатия любой из кнопок «Принять» или «Отклонить» система меняет статус заявки на участие и уведомляет участника о результате рассмотрения заявки по почте.
 6. Система завершает работу варианта использования.
- Альтернативные потоки событий:** нет.
- Сценарий обработки ошибок:** нет.

Спецификация варианта использования «Подведение итогов участия в проекте».

Цель: зафиксировать результат участия в проекте

Активные субъекты: участники, уже состоящие в команде.

Краткое описание: Один из участников, уже состоящих в команде, фиксирует результат об участии в проекте.

Предусловия: Участник должен авторизоваться в системе, чтобы появилась возможность подведения итогов.

Постусловия: нет.

Основной поток событий:

1. Система публикует страницу с командами, в которых состоит участник, в рамках личного кабинета. Каждая команда имеет кнопку «Подробнее» и значок, появляющийся при наличии уведомления.
2. Участник, уже состоящий в выбранной команде, нажимает на кнопку «Подробнее» определенной команды.
3. Система показывает участнику, состоящему в команде, страницу с информацией о команде, ее участниках и желающих участвовать, а также сведениями об участии команды в проекте. Если команда еще не подводила итоги, то на странице команды присутствует кнопка «Сформировать итоги участия»
4. Участник, состоящий в команде, нажимает кнопку «Сформировать итоги участия»
5. Система публикует страницу с формой подведения итогов со следующими полями: ссылка на решение, изображения результатов, отзыв в виде текста.
6. Активный субъект заполняет поля формы.
7. Система проверяет данные в форме на валидность.

8. Активный субъект нажимает кнопку «Сохранить».
9. Система уведомляет пользователя об успешном или не успешном сохранении результатов команды.
10. Система завершает работу варианта использования.

Альтернативные потоки событий: нет.

1. Система демонстрирует активному субъекту о неправильном заполнении полей в форме.
2. Система демонстрирует активному субъекту сообщение: «Ошибка при сохранении итогов».

Сценарий обработки ошибок: при возникновении ошибки о сохранении итогов, формы сохраняют свои данные и позволяют пользователю заново сохранить итоги.

Требования к реализации задачи

Что необходимо реализовать в REST API:

- /api/projects/:id/teams POST Создание команды для участия в проекте;
- /api/users/:id/teams GET Получение списка команд пользователя;
- /api/users/:id/teams/:id POST Изменение статуса у заявки участника на выполнение проекта в определенной команде;
- /api/events/:id/result POST Подведение итогов участия в проекте.

Что необходимо реализовать в веб-приложении:

- страница «Проекты» с кнопкой создания проекта;
- страница «Проект» с кнопкой создания команды и списком команд, выполняющих проект;
- страница «Команды» в личном кабинете, со списком команд, в которых состоит пользователь;
- страница «Команда» с информацией о команде и ее участниках, так же с поданными на вступление заявками;
- страница «Подача заявки на вступление в команду» с формой ввода;
- страница «Подведение итогов участия в проекте» с формой ввода.

4.3.3 Вариант использования «Участие в хакатоне»

Сценарий варианта использования «Участие в хакатоне» показан на рисунке 4.8. Система должна показывать хакатоны и позволять переходить на сайты организаторов хакатонов. Участие в хакатоне не детализируется, потому что состоит из следующих действий пользователя: просмотр списка хакатонов, выбор интересующего, открытие страницы с информацией о хакатоне, нажатие на кнопку «Записаться на хакатон» и система открывает страницу хакатона на сайте организаторов. Реализация предполагается как в серверной, так и в клиентской части. В дальнейшем планируется добавить возможность собирать команду на хакатон в разрабатываемой системе, пока что на данной итерации проекта участие в хакатоне предполагает только переход в другую ИС.

Отдельно в описании задачи необходимо определить, каким образом хакатоны будут появляться в системе. Администратор может создать хакатон вручную, для этого в клиентской части необходимо разработать форму ввода, а

на серверной стороне необходимо создать POST запрос для создания хакатона. Однако, предполагается что в следующих итерациях разработки будет создан агрегатор проектов. Определим первоначальные требования к агрегатору:

- агрегатор должен быть реализован отдельным приложением, которое будет запускаться по инициативе из приложения REST API;
- агрегатор необходимо реализовать на Python [10], с помощью библиотек:
 - requests — для отправления запросов на страницы в интернете;
 - beautifulsoup4 — для парсинга страниц;
 - sqlalchemy — для работ с БД;
 - источники для агрегатора определены в требованиях к системе в целом (п. 2.2);
- на административной панели должно располагаться управление агрегатором. Таблица с источниками для агрегатора должна содержать данные об источниках, количество хакатонов и даты парсинга. Так же необходимо предусмотреть отключение некоторых из источников или подключение новых.;
- агрегатор должен сам напрямую работать с базой данных;
- агрегатор получает данные с сайтов-источников с помощью двух методов: через обращение к API и через парсинг HTML кода.



Рисунок 4.8 – Сценарий варианта использования «Участие в хакатоне»

4.3.4 Вариант использования «Проведение мероприятия»

Сценарий варианта использования «Проведение мероприятия» показан на рисунке 4.9. Система должна позволять администратору создавать мероприятия, участвовать в которых может только участник.

Проведение мероприятия состоит из следующих подзадач: авторизация, создание мероприятия, выполнение задания, оценка результатов экспертами и подведение результатов мероприятия. В задаче участвуют двое пользователей: участник и администратор. Реализация предполагается как в серверной, так и в клиентской части.

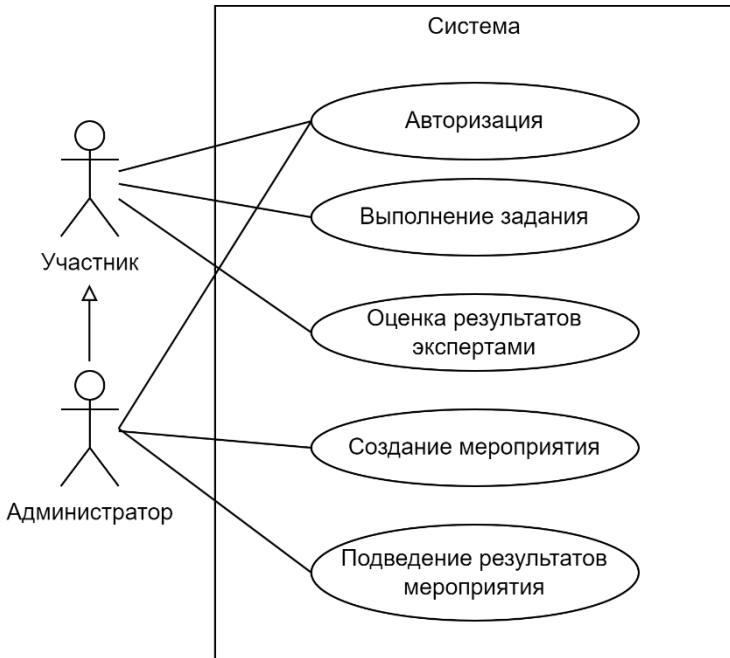


Рисунок 4.9 – Сценарий варианта использования «Проведение мероприятия»

4.4 Проектирование архитектуры системы

Модель состоит из 4х уровней, на данном этапе проектирования будут спроектированы 3 уровня. Четвертый уровень модели подразумевает диаграмму кода, который будет определен после частичной реализации.

Реализовать систему в соответствии с требованиями ТЗ необходимо в виде веб-приложения, HTTP сервера и агрегатора.

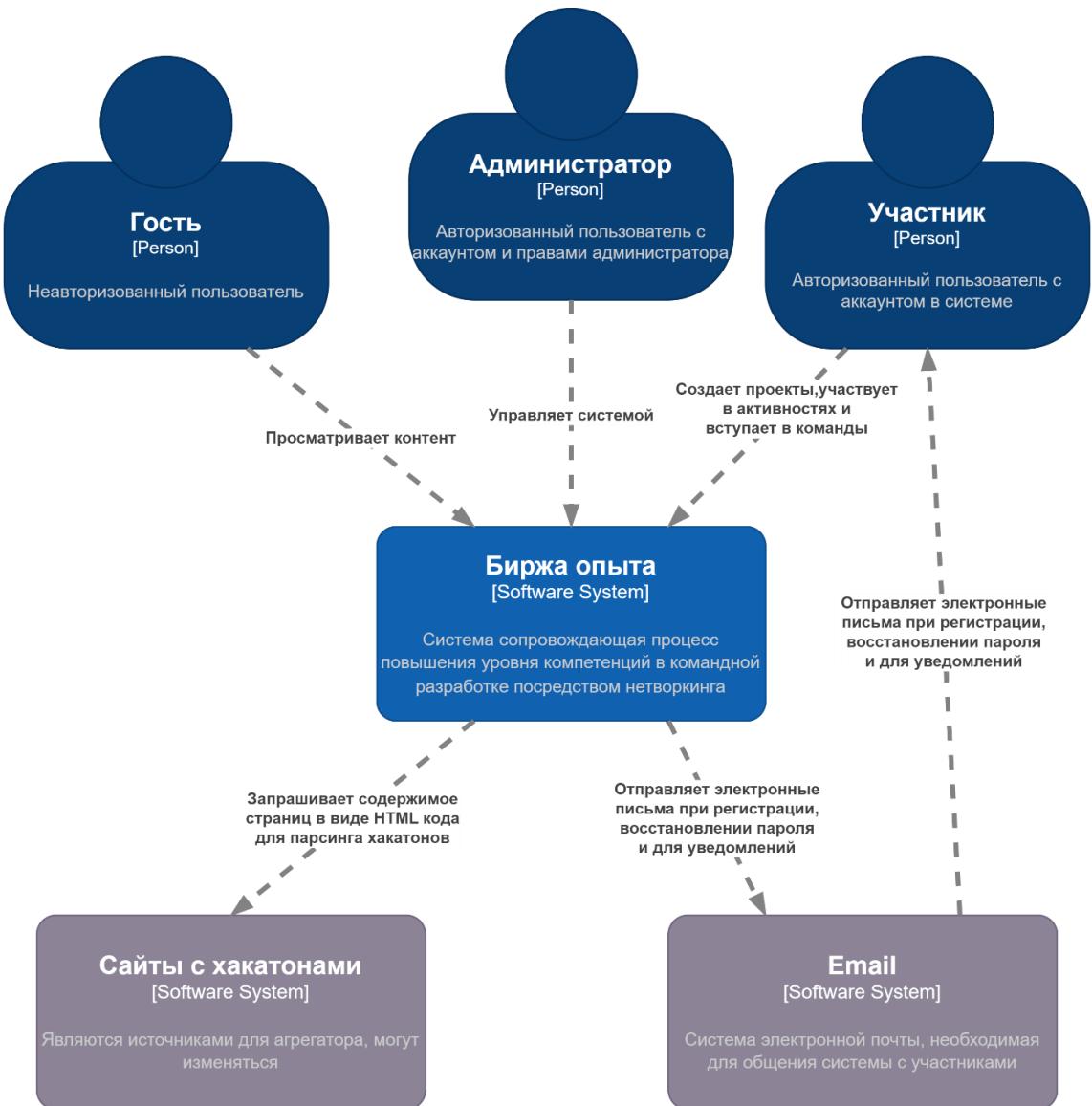


Диаграмма контекста системы "Биржа опыта"

Рисунок 4.10 – Уровень 1. Схема системного контекста

На рисунке 4.10 изображена диаграмма контекста системы, которая показывает, что «Биржа опыта» должна взаимодействовать с системой Email для отправки писем пользователю, а также сайтами с хакатонами для агрегирования данных.

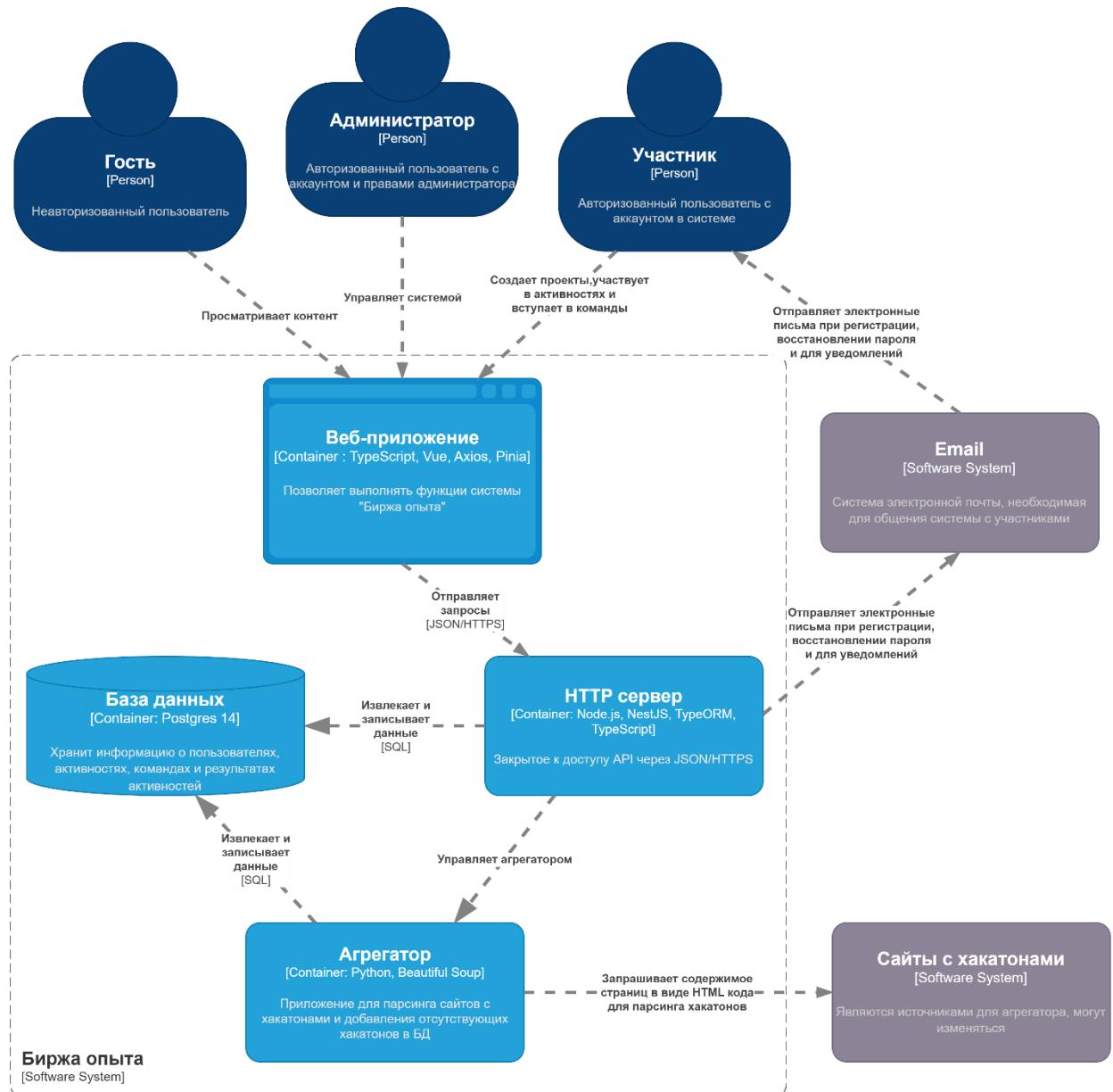


Диаграмма контейнеров системы "Биржа опыта"

Рисунок 4.11 – Уровень 2. Схема контейнеров

На рисунке 4.11 изображена диаграмма контейнеров, которая отображает концептуальную архитектуру системы «Биржа опыта». Система должна состоять из веб-приложения (клиент), HTTP сервера (сервер), агрегатора, а также самой БД. Веб-приложение должно взаимодействовать с HTTP сервером через HTTP протокол в JSON формате.

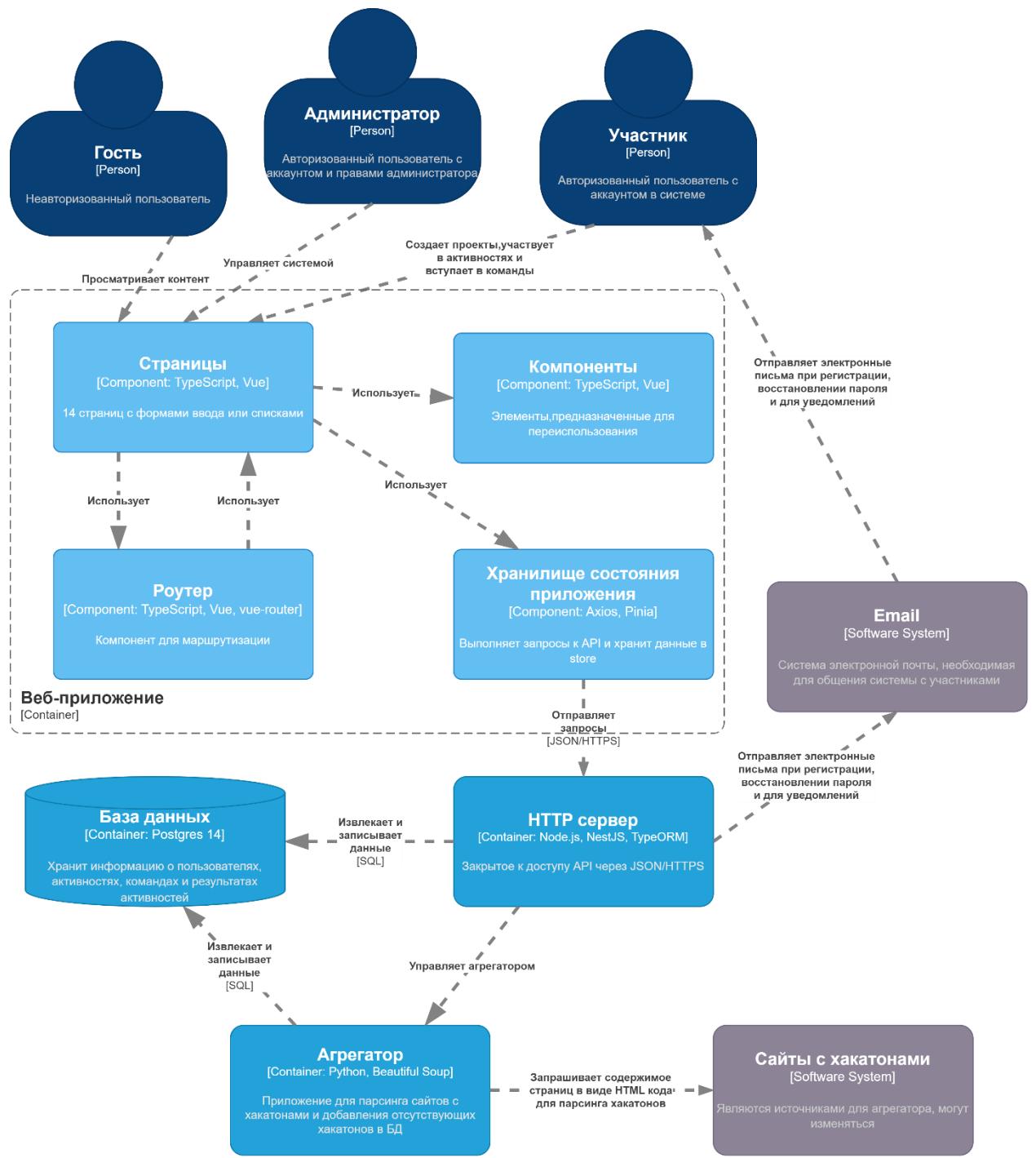


Диаграмма компонентов системы "Биржа опыта"- Веб-приложение

Рисунок 4.12 – Уровень 3. Схема компонентов веб-приложения

На рисунке 4.12 показана схема компонентов веб-приложения: страницы, компоненты, хранилище и роутер.

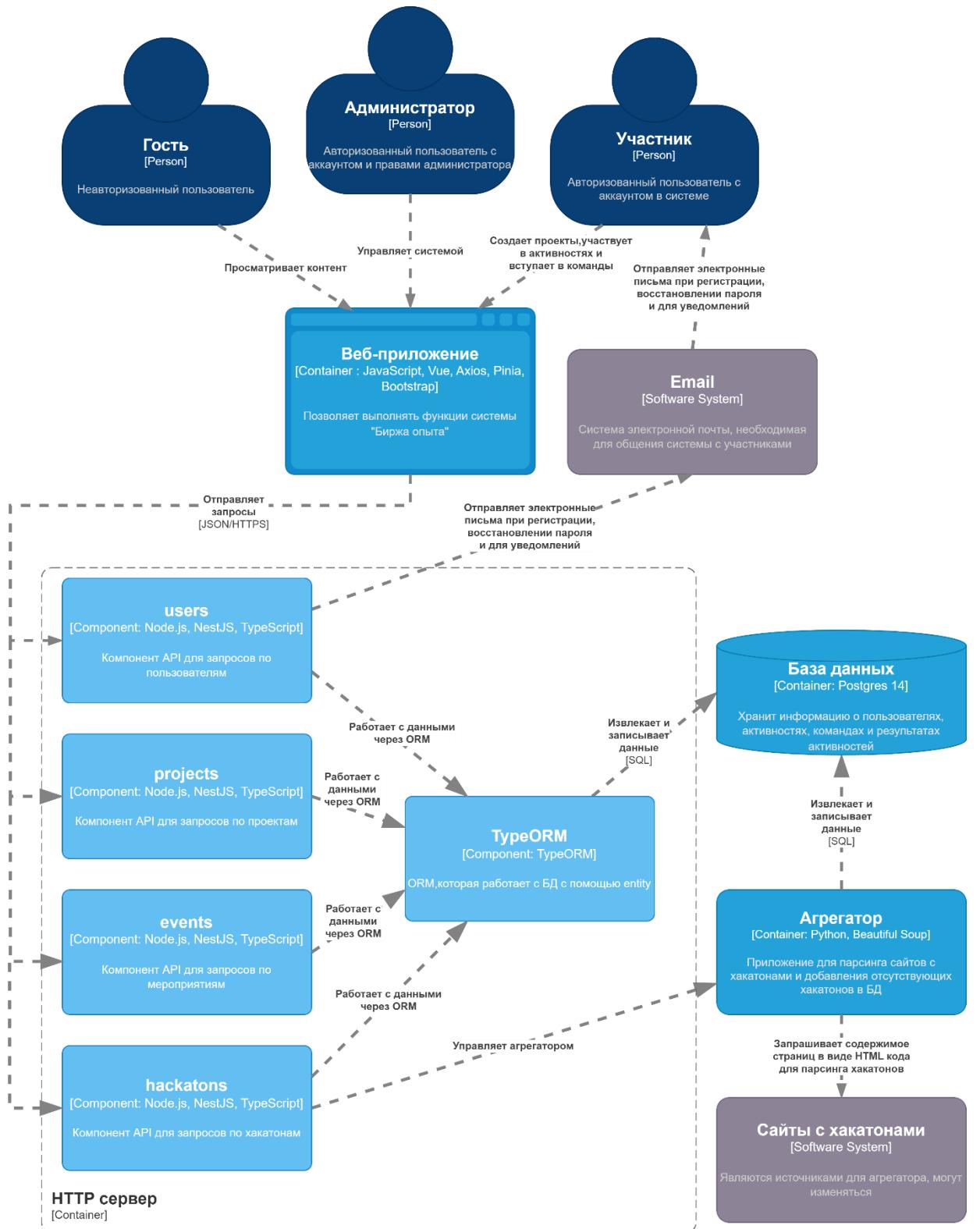


Диаграмма компонентов системы "Биржа опыта"- HTTP сервер

Рисунок 4.13 – Уровень 3. Схема компонентов HTTP сервер

На рисунке 4.13 находится схема компонентов HTTP сервер, а именно модулей: projects, hackatons, events, users. Все модули должны взаимодействовать с БД через библиотеку TypeORM. Так же модуль hackatons управляет агрегатором. С почтой взаимодействует модуль users.

4.5 Проектирование базы данных

В данном разделе выполнено проектирование модели данных из сущностей, определенных в техническом задании: проект, хакатон, мероприятие, пользователь, команда. В процессе проектирования были приняты следующие решения:

- проекты, мероприятия и хакатоны объединить в одну сущность с названием «Событие» по причине наличия одинаковых полей (для различия было создано поле «Тип события: мероприятие, проект, хакатон»);
- созданы идентификаторы у всех сущностей для более удобного хранения связей;
- некоторые поля имеют тип данных массив, для упрощения хранения;
- создать сущность голосования для оценки результатов события экспертами;
- пользователя и команду связывают сущности функция и функция пользователя (это необходимо для фиксирования роли участника в команде).
- результат события вынесен в отдельную сущность, для возможности создания нескольких результатов разных команд на одно событие.

В результате, получившаяся логическая модель данных показана на рисунке 4.14. Она содержит 10 сущностей. В дальнейшем модель будет расширяться по необходимости, однако на данный момент существующих сущностей и атрибутов достаточно для реализации первой версии системы.

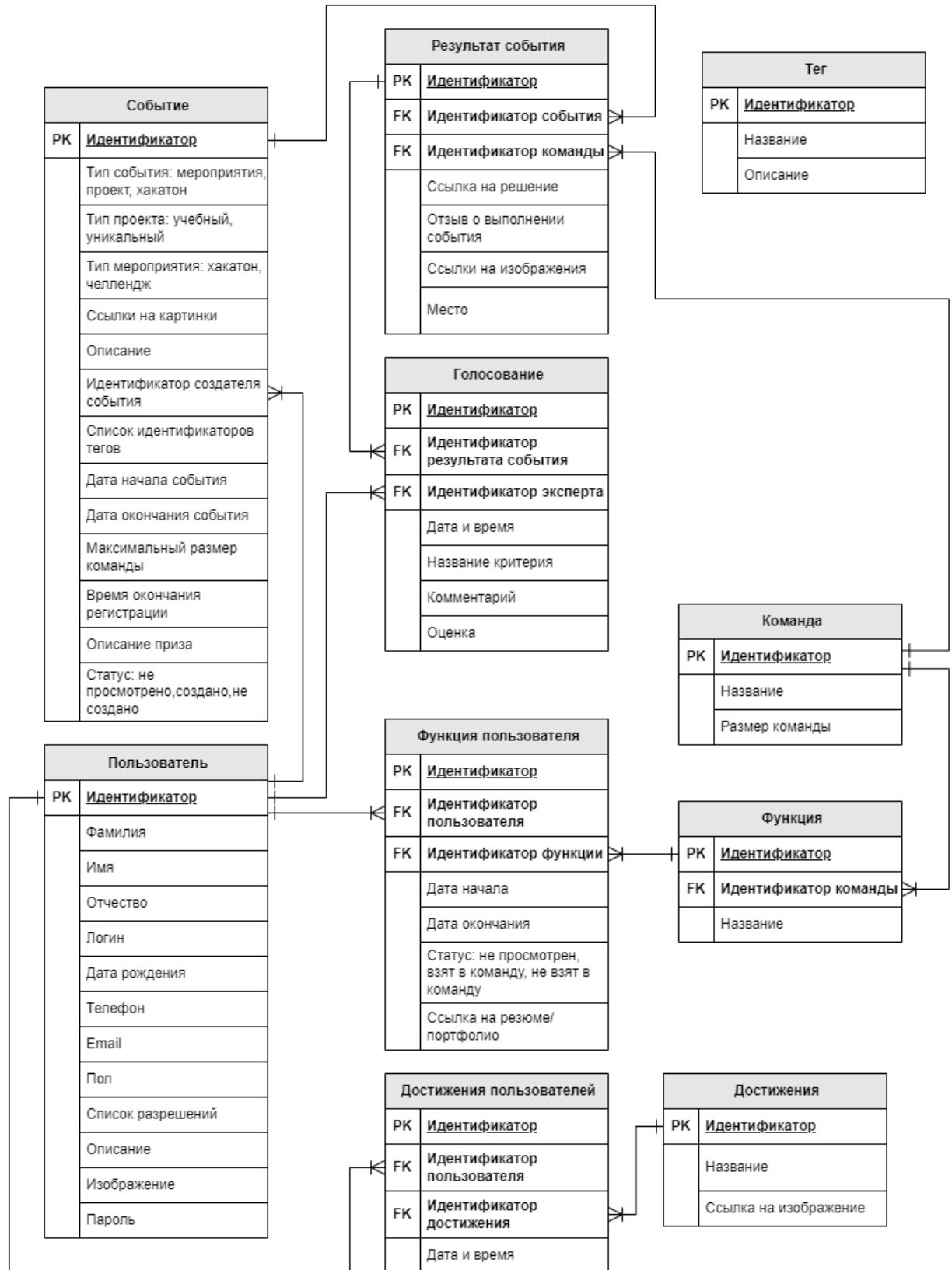


Рисунок 4.14 – Логическая модель данных

В таблице 4.1 представлено сопоставление сущностей из логической модели данных и физической модели данных.

Таблица 4.1 – Описание полей базы данных

Сущность	Таблица	Атрибут	Поле	Тип данных	Ключ
Событие	events	Идентификатор	id	int	PK
		Тип события: мероприятия, проект, хакатон	type	enum	
		Тип проекта: учебный, уникальный	type_project	enum	
		Тип мероприятия: хакатон, челлендж	type_events	enum	
		Ссылки на изображения	images	string[]	
		Описание	descriptions	string	
		Идентификатор создателя события	creator_id	int	FK
		Идентификаторы тегов	tags	int[]	
		Дата начала события	date_start	datetime	
		Дата окончания события	date_end	datetime	
		Максимальный размер команды	team_size	string	
		Время окончания регистрации	last_time_registration	datetime	
		Описание приза	prize	string	
		Статус: не просмотрено, создано, не создано	status	enum	
Пользователь	users	Идентификатор	id	int	PK
		Фамилия	last_name	string	
		Имя	name	string	
		Отчество	patronymic	string	
		Логин	username	string	
		Дата рождения	birthdate	date	
		Телефон	phone	string	
		Email	email	string	
		Пол	gender	string	
		Список разрешений	permission	string[]	
		Описание	bio	string	
		Изображение	image	string	
		Пароль	password	string	
Функция пользователя	user_functions	Идентификатор	id	int	PK
		Идентификатор пользователя	user_id	int	FK
		Идентификатор функций	function_id	int	FK
		Дата начала	date_start	date	
		Дата окончания	date_end	date	
		Статус: не просмотрен, взят в команду, не взят в команду	status	enum	

Продолжение таблицы 4.1

Сущность	Таблица	Атрибут	Поле	Тип данных	Ключ
Функция пользователя	user_functions	Ссылка на резюме/портфолио	linkInfoUser	string	
Функция	functions	Идентификатор	id	int	PK
		Идентификатор команды	team_id	int	FK
		Название	title	string	
Команда	teams	Идентификатор	id	int	PK
		Название	title	string	
		Размер команды	team_size	int	
Результат события	result_events	Идентификатор	id	int	PK
		Идентификатор события	event_id	int	FK
		Идентификатор команды	team_id	int	FK
		Ссылка на решение	link_solution	date	
		Отзыв о выполнении события	feedback	date	
		Ссылки на изображения	images	string[]	
		Место	place	string	
Голосование	voting	Идентификатор	id	int	PK
		Идентификатор результата события	result_event_id	int	FK
		Идентификатор эксперта	user_id	int	FK
		Дата и время	date	date	
		Название критерия	title_criterion	date	
		Комментарий	comments	string	
		Оценка	linkInfoUser	string	
Тег	tags	Идентификатор	id	int	PK
		Название	title	string	
		Описание	description	string	
Достижения	achievements	Идентификатор	id	int	PK
		Название	title	string	
		Ссылка на изображение	image	string	
Достижения пользователей	users_achievements	Идентификатор	id	int	PK
		Идентификатор пользователя	user_id	int	FK
		Идентификатор достижения	achievement_id	int	FK
		Дата и время	datetime	datetime	

Физическая модель данных была спроектирована с учетом СУБД PostgreSQL в сервисе DrawSQL [11] и отображена на рисунке 4.15.

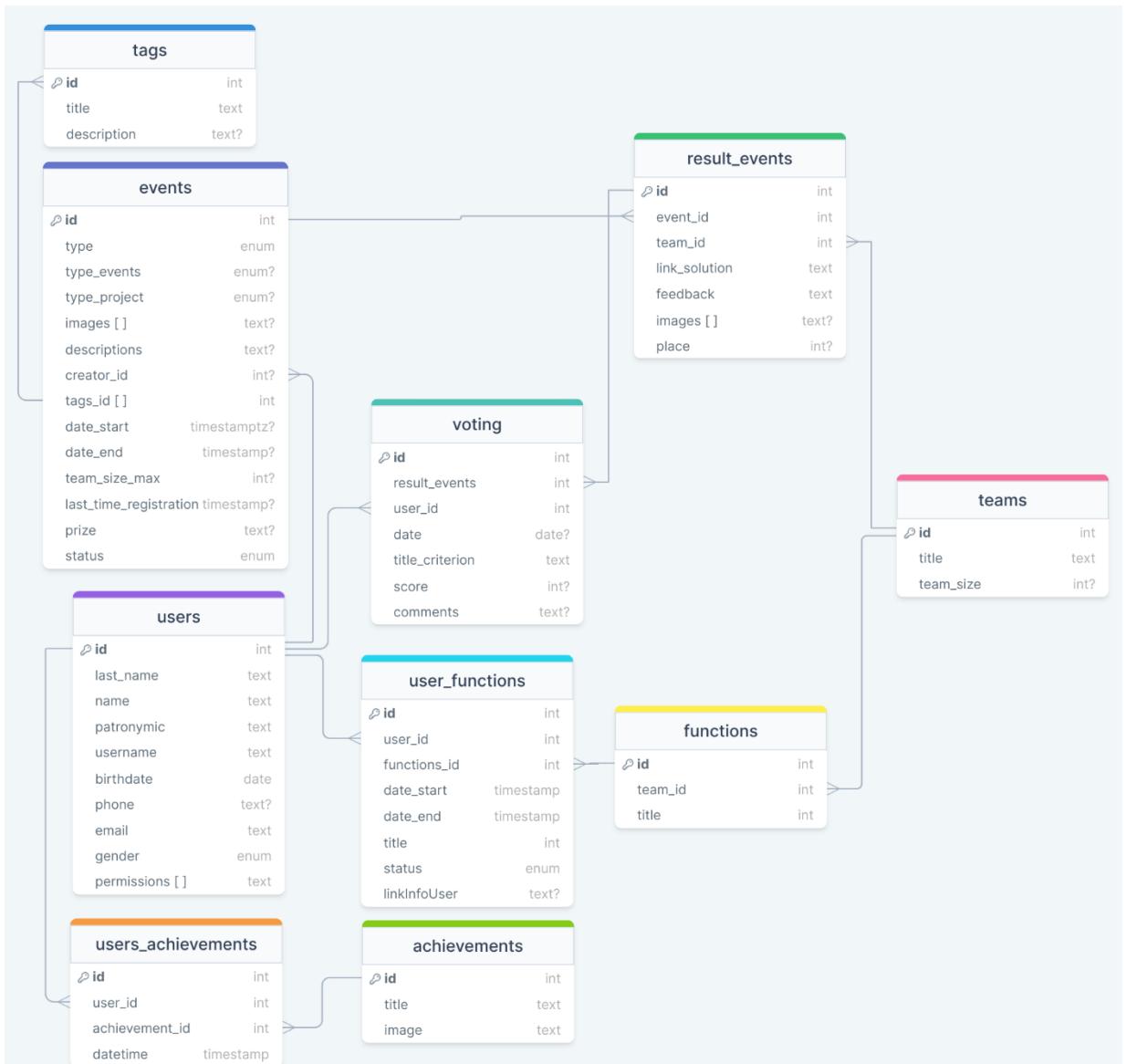


Рисунок 4.15 – Физическая модель данных

4.6 Проектирование пользовательского интерфейса

Ниже приведены примеры макетов, сделанных в онлайн – сервисе Figma [12]. Макеты сделаны в виде набросков структуры страниц разрабатываемой системы.

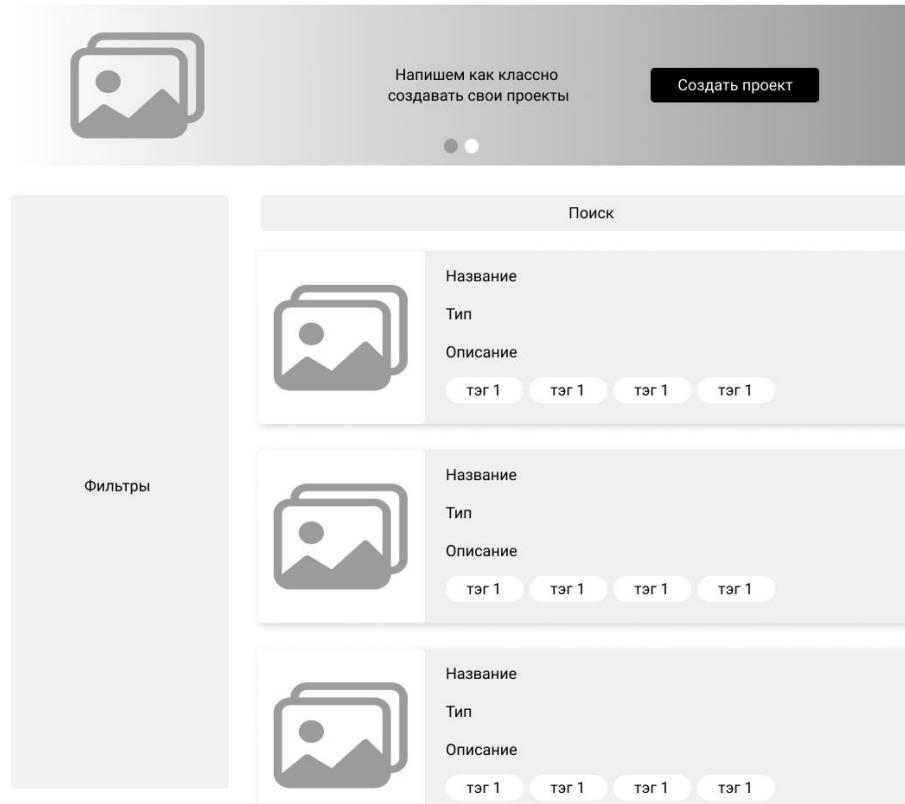


Рисунок 4.20 – Страница проектов

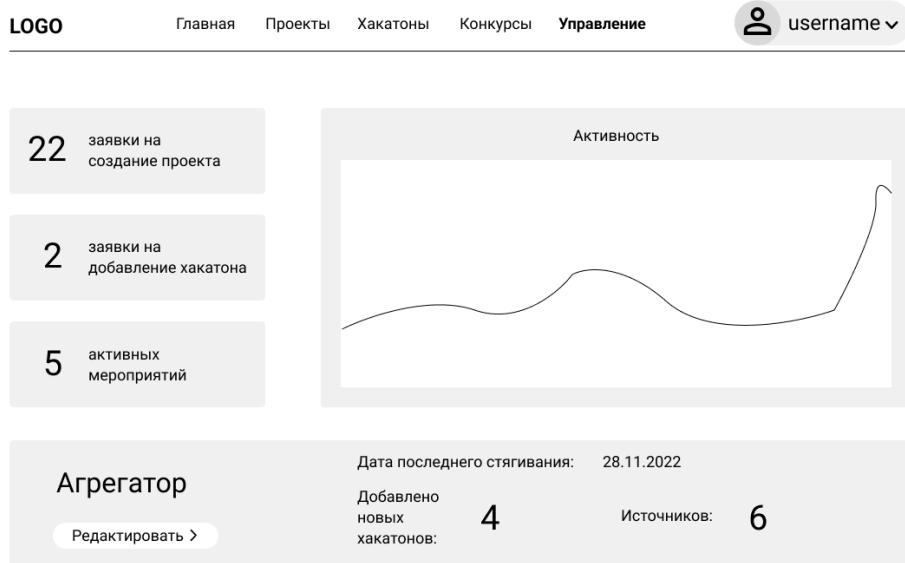


Рисунок 4.21 – Страница для управления системой

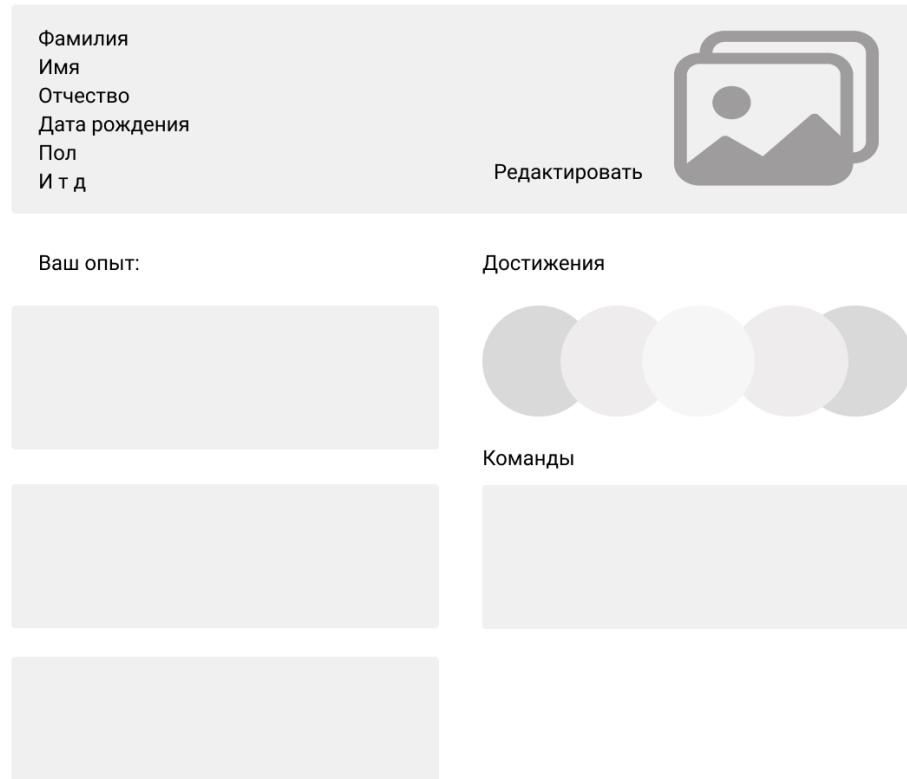


Рисунок 4.22 – Страница «Профиль»

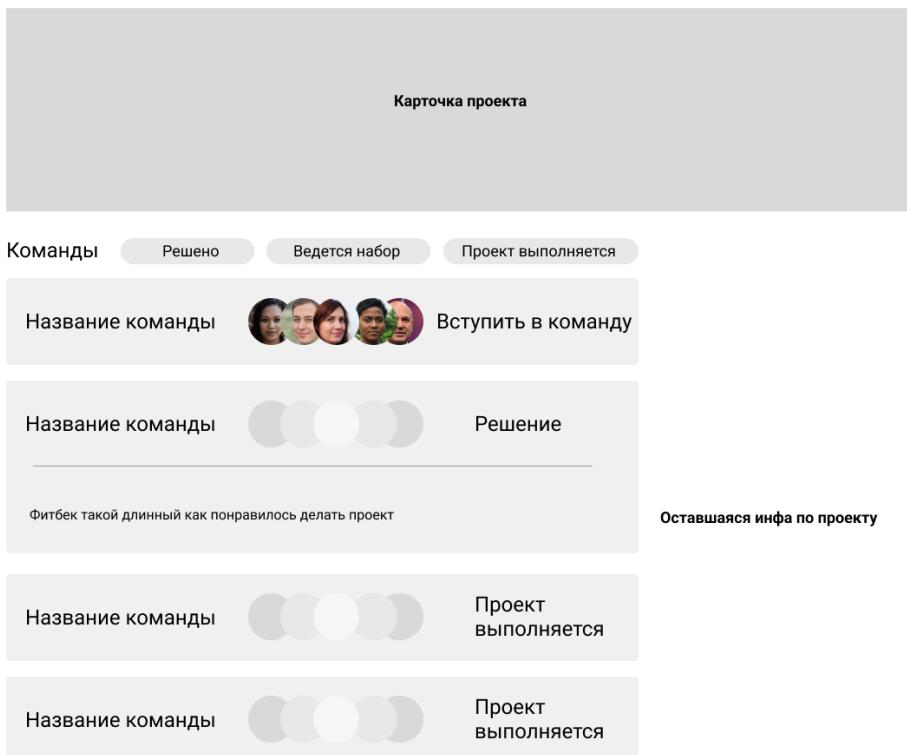


Рисунок 4.23 – Страница «Проект»

Создание проекта

Для того что бы создать проект вам необходимо заполнить форму ниже и отправить заявку на проверку.
В ближайшее время администратор рассмотрит заявку и вам придет уведомление о результате.

Название**Тип проекта**

- Учебный
- Уникальный

Ссылка на изображение проекта**Описание****Отправить заявку на создание**

Рисунок 4.24 – Страница «Создание проекта»



Рисунок 4.25 – Логотип

5 Реализация приложения

После проектирования была выполнена реализация MVP. Для демонстрации смысла системы был выбран вариант использования «Создание проекта». Для его реализации так же понадобилось реализовать авторизацию, регистрацию и личный кабинет. В данном разделе показано текущее состояние системы.

5.1 Описание возможностей текущей версии

Система состоит из 3 разных программ: веб-приложение, HTTP сервер и агрегатор. Исходя из этого описание реализованных функций разделено на веб-приложение и HTTP сервер. Работы по реализации агрегатора на данный момент не выполнялись.

Страницы, реализованные согласно макетам и структуре страниц:

- страница «Главная»;
- страница «Авторизация»;
- страница «Регистрация»;
- страница «Проекты»;
- страница «Хакатоны»;
- страница «Мероприятия»;
- страница «Профиль»;
- страница «Административная панель»;
- страница «Создание проекта»;
- страница «Создание хакатона»;
- страница «Заявки на создание проекта»;
- страница «Создание команды для проекта».

Перечень API запросов, реализованных на стороне сервера показан в таблице 5.1.

Таблица 5.1 – Реализованные API запросы

Путь	Метод	Назначение
/api/projects	GET	Получение списка проектов
/api/projects/:id/	GET	Получение информации о проекте
/api/projects/:id/team	GET	Получение списка команд, которые участвуют/участвовали в проекте
/api/tags	GET	Получение списка тегов
/api/project	POST	Создание проекта
/api/admin	GET	Получение информации для дашбордов
/api/admin/project/requestCreateProject/count	GET	Получение количества заявок, поданных на создание проекта
/api/admin/project/requestCreateProject	GET	Получение списка заявок, поданных на создание проекта

Продолжение таблицы 5.1

Путь	Метод	Назначение
/api/project/:id	POST	Изменение статуса проекта при рассмотрении проекта
/api/hackatons	GET	Получение списка хакатонов
/api/events	GET	Получение списка мероприятий
/api/users/login	POST	Авторизация
/api/users/logout	POST	Выход из профиля
/api/users	POST	Регистрация
/api/users/user	POST	Получение информации о пользователе, если он авторизован

Исходя из вышеперечисленного, можно повести итог о реализованных функциях в каждой подсистеме.

На данный момент в подсистеме проектов реализованы следующие функции:

- просмотр списка проектов;
- просмотр информации о проекте: название; описание; изображения; тип проекта (уникальный/учебный); список команд, которые его выполняют/выполнили;
- отправление заявки на создание проекта администратору.

В подсистеме личного кабинета реализованы следующие функции:

- регистрация с помощью email, пароля и логина пользователя;
- авторизация с помощью email и пароля;
- просмотр личного кабинета: информация о профиле, список достижений пользователя.

На этапе MVP в подсистеме административная панель реализованы следующие функции:

- просмотр списка поданных на рассмотрение заявок для создания проектов;
- рассмотрение заявки на создание проекта в административной панели, в результате проект приобретает статус (создан, не создан).

В подсистеме мероприятий реализована функция:

- просмотр списка мероприятий.

На данный момент в подсистеме хакатонов реализованы следующие функции:

- просмотр списка хакатонов.

Исходя из вышеописанных возможностей, можно сделать вывод что для полной реализации первой версии необходимо разработать еще 3 варианта использования. Так же необходимо продумать дизайн сайта, потому что на этапе MVP этому не было выделено внимание.

5.2 Демонстрация варианта использования «Создание проекта»

В пункте 4.3.1 был определен вариант использования «Создание проекта», который будет продемонстрирован ниже.

Первое, что видит пользователь, попадая на сайт – главная страница (рис. 5.1). Для того что бы создать проект, необходимо перейти на страницу с проектами (рис. 5.2) с помощью меню. Так как система должна позволять создавать проект только авторизованному пользователю, при нажатии на кнопку «Создать проект» система открывает страницу авторизации (рис. 5.3). Пользователь авторизуется и возвращается к странице проектов, где еще раз нажимает на кнопку «Создать проект». Система открывает форму с полями для создания проекта (рис. 5.4). Пользователь заполняет поля формы (рис. 5.4) и отправляет заявку. Система возвращает пользователя на страницу проектов.

Администратор авторизуется и видит уведомления в меню (рис. 5.5). Администратор переходит в административную панель и видит, что у него 4 заявки на создание проекта (рис. 5.6). Администратор открывает все заявки на создание проекта и видит таблицу с проектами (рис. 5.7). Администратор принимает решение о создании проекта «Тестовый проект» и нажимает кнопку «Создать» напротив проекта. Проект получает статус «created» (рис. 5.8). В списке проектов появляется созданный проект (рис. 5.9).

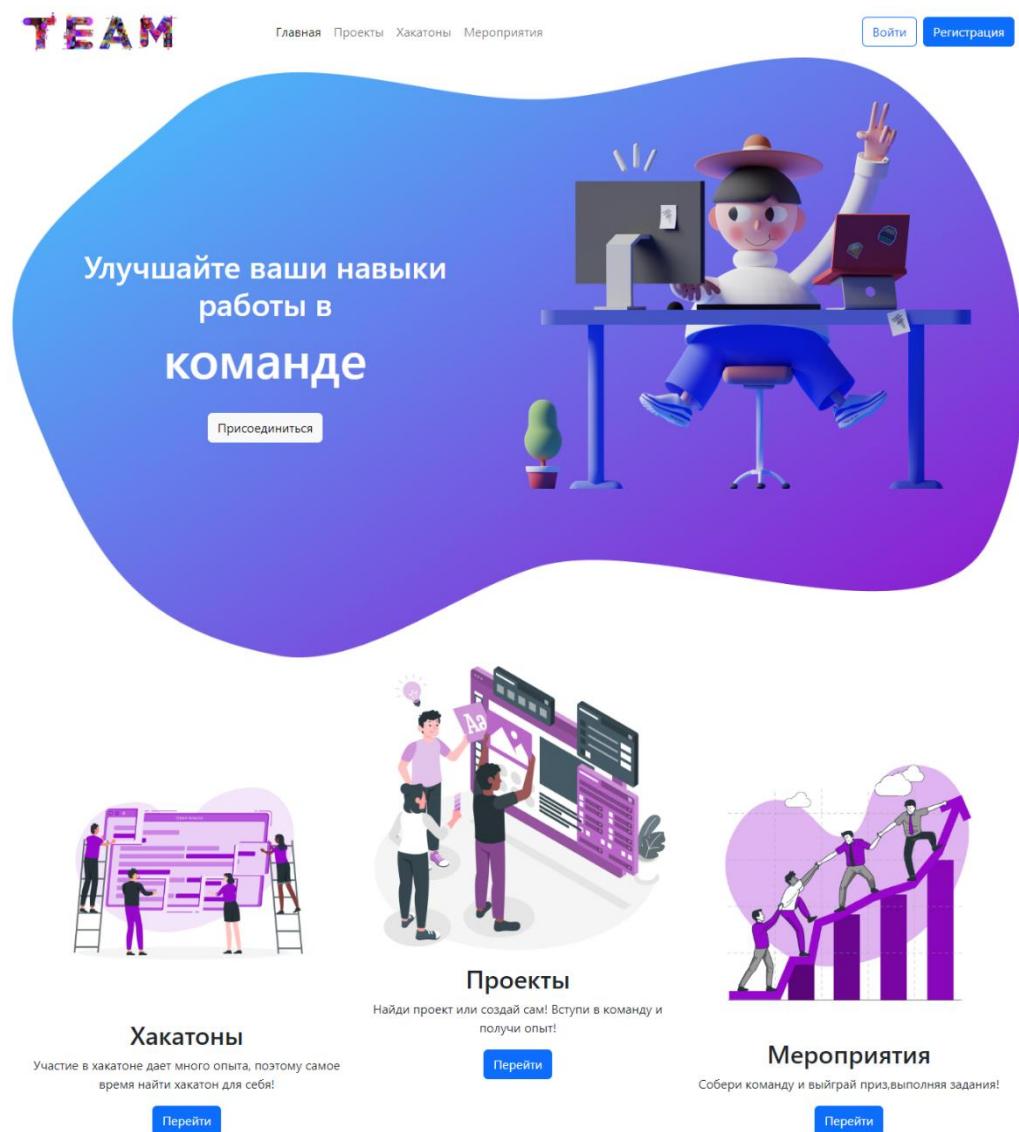


Рисунок 5.1 – Страница «Главная»



Создай свой проект и набери команду!

[Создать проект](#)**Фильтры****Тип проекта**

- Учебный
- Уникальный

Тэги

- [Веб-разработка](#)
- [AI & ML](#)
- [Инфобез](#)
- [Дизайн](#)
- [UI](#)
- [Бизнес-аналитика](#)
- [Маркетинг](#)
- [GameDev](#)

Поиск**Создайте свой аналог Trello с элементами геймификации**

#Что такое Trello? Трэлло - это сервис для совместной работы и организации задач. С помощью него можно составлять списки дел, распределять их по катег...

Уникальный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)**Какой фильм я должен посмотреть с помощью BeautifulSoup**

Основная идея: создать базовый Ethereum-обозреватель с предоставлением пользователю возможностей: видеть последние 10 блоков и список транзакций; пер...

Уникальный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)**Создайте свою собственную нейронную сеть с нуля на Python**

Интересный проект

Учебный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)**Проект**

Интересный проект

Учебный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)**Рисунок 5.2 – Страница «Проекты»****Авторизация**[Необходим аккаунт?](#) Email Password[Войти](#)**Рисунок 5.3 – Страница «Авторизация»**

[←Проекты](#)

Создание проекта

Для того что бы создать проект вам необходимо заполнить форму ниже и отправить заявку на проверку. В ближайшее время администратор рассмотрит заявку и вам придет уведомление о результате.

Название проекта

Тестовый проект

Тип проекта

- Уникальный
- Учебный

Ссылка на картинку проекта

https://ngokitchen.ru/wp-content/uploads/2021/08/shutterstock_3

Описание проекта

Проект для демонстрации варианта использования "Создание проекта"

Отправить заявку на создание**Рисунок 5.4 – Страница «Создание проекта»****Профиль admin**

Комогорцева Юлия Владимировна

Контакты:

Описание профиля

a@yandex.ru

male

8888888888

Мой опыт

Проект test

Команда 1



Проект test

Команда 2



Проект test

Команда 1



Проект test

Команда 2

**Достижения**

Quickdraw

Gitty up!



Pull Shark

@KomogorcevaJula opened pull requests that have been merged.

History

100% orbited

Unlocked on Sep 11, 2022

• KomogorcevaJula/hackathon#12 - Closed within 5 minutes of opening

History

Bronze unlocked - First unlocked on May 21, 2022

• Fazekas/Boris-holsten/2022-Wiki-Website - 3rd pdf

Рисунок 5.5 – Страница «Профиль»

4

поданных заявок на создание проекта

2

поданных заявок на добавление хакатона

5

активных мероприятий

**Агрегатор**[Настройки](#)

Дата последнего запуска: 20.01.2023

Добавлено новых хакатонов: 2 Источников: 6

Создать хакатон вручную**Рисунок 5.6 – Страница «Административная панель»**

[← Панель управления](#)

Заявки на создание проекта

ID	Название	Описание	Тип	Статус	Решение
114	Тестовый проект	Проект для демонстрации варианта использования "Со...	Учебный	not viewed	Создать Отклонить
6	Создайте редактор Markdown в реальном времени	Интересный проект	Учебный	not viewed	Создать Отклонить
29	Проект	Интересный проект	Уникальный	not viewed	Создать Отклонить
47	Проект	Интересный проект	Учебный	not viewed	Создать Отклонить
5	Создайте свой аналог Trello с элементами геймификации	#Что такое Trello? Трэлло - это сервис для совмест...	Уникальный	created	
7	Какой фильм я должен посмотреть с помощью BeautifulSoup	Основная идея: создать базовый Etherium-обозревате...	Уникальный	created	
8	Создайте свою собственную нейронную сеть с нуля на Python	Интересный проект	Учебный	created	
43	Проект	Интересный проект	Учебный	created	

Рисунок 5.7 – Страница «Заявки на создание проекта»

[← Панель управления](#)

Заявки на создание проекта

ID	Название	Описание	Тип	Статус	Решение
6	Создайте редактор Markdown в реальном времени	Интересный проект	Учебный	not viewed	Создать Отклонить
29	Проект	Интересный проект	Уникальный	not viewed	Создать Отклонить
47	Проект	Интересный проект	Учебный	not viewed	Создать Отклонить
43	Проект	Интересный проект	Учебный	created	
114	Тестовый проект	Проект для демонстрации варианта использования "Со...	Учебный	created	
5	Создайте свой аналог Trello с элементами геймификации	#Что такое Trello? Трэлло - это сервис для совмест...	Уникальный	created	
7	Какой фильм я должен посмотреть с помощью BeautifulSoup	Основная идея: создать базовый Etherium-обозревате...	Уникальный	created	
8	Создайте свою собственную нейронную сеть с нуля на Python	Интересный проект	Учебный	created	

Рисунок 5.8 – Страница «Заявки на создание проекта» с созданным проектом



Создай свой проект и набери команду!

[Создать проект](#)

Фильтры

Тип проекта

- Учебный
 Уникальный

Тэги

- Веб-разработка
AI & ML
Инфобез
Дизайн
UI
Бизнес-аналитика
Маркетинг
GameDev

Поиск



Создайте свой аналог Trello с элементами геймификации

#Что такое Trello? Трэлло - это сервис для совместной работы и организации задач. С помощью него можно составлять списки дел, распределять их по кате...

Уникальный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)

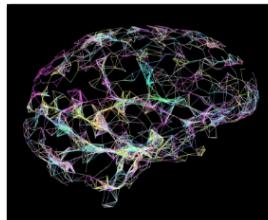


Какой фильм я должен посмотреть с помощью BeautifulSoup

Основная идея: создать базовый Etherium-обозреватель с предоставлением пользователю возможностей: видеть последние 10 блоков и список транзакций; пер...

Уникальный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)



Создайте свою собственную нейронную сеть с нуля на Python

Интересный проект

Учебный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)



Проект

Интересный проект

Учебный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)



Тестовый проект

Проект для демонстрации варианта использования "Создание проекта"

Учебный

[tag 1](#) [tag 2](#) [tag 3](#) [tag 4](#)

Рисунок 5.9 – Страница «Заявки на создание проекта»

6 Экономическая часть

При расчете экономической части предполагается использовать программный продукт Microsoft Project, для разработки календарного плана, распределения ресурсов и последующего анализа. Для понимания изображений и прочих материалов из данной программы рекомендуется ознакомиться со справкой для пользователя [13].

Выбор комплекса работ по разработке проекта производится в соответствии с требованиями к составу и содержанию работ по созданию системы (пункт 2.6).

Разработка системы не производится для заказчика или передачи другим лицам в дальнейшем. Система не преследует коммерческие цели, однако, важно рассчитать затраты на создание ПО из-за возможных доходов в дальнейшем. Например, при высоких метриках посещаемости сайта возможно внедрение рекламы от заинтересованных лиц. Так же можно предусмотреть возможность сбора финансов краудфандингом для окупаемости и развития проекта.

Созданием системы на всех этапах занимается один разработчик в свое свободное от основных занятий время, так как это является его инициативой. В связи с этим, для расчета затрат было принято решение присвоить разработчику статус «Плательщик налога на профессиональный доход» (самозанятый). Из-за с данного статусом, страховые взносы и налоги не учитываются. Однако, в случае продажи или передачи системы в пользование другим лицам по лицензионному договору необходимо учесть налог на профессиональный доход: 4 % при передачи физическому лицу и 6% при передаче юридическому лицу.

Так же, необходимо понимать, что на этапе разработки все затраты несет разработчик.

Для расчета стоимости проекта необходимо определить задачи и подзадачи (рис. 6.1). Так же важно распределить порядок выполнения каждой задачи (рис. 6.2). После этого необходимо определить длительность задач в часах. На рисунке 6.1 показан лист задач. Общая длительность проекта 527 рабочих часов с 01.10.22 по 15.05.23.

	Название задачи	Длительность	Начало	Окончание	Пред.
1	▫ Создание системы	527 часов	Сб 01.10.22	Пн 15.05.23	
2	▫ Анализ предметной области, формирование требований и разработка технического задания	150 часов	Сб 01.10.22	Ср 14.12.22	
3	Анализ предметной области	20 часов	Сб 01.10.22	Пн 10.10.22	
4	Обзор аналогов	20 часов	Сб 01.10.22	Пн 10.10.22	300
5	Формирование требований	40 часов	Вт 11.10.22	Вс 30.10.22	4;3
6	Разработка технического задания	90 часов	Пн 31.10.22	Ср 14.12.22	5
7	▫ Проектирование. Создание технического проекта.	60 часов	Чт 15.12.22	Чт 19.01.23	
8	Моделирование бизнес-процессов to-be	50 часов	Чт 15.12.22	Сб 14.01.23	6
9	Проектирование архитектуры системы	45 часов	Чт 15.12.22	Чт 12.01.23	6
10	Проектирование модели данных	50 часов	Чт 15.12.22	Сб 14.01.23	6
11	Проектирование пользовательского интерфейса	50 часов	Чт 15.12.22	Сб 14.01.23	6
12	Формирование технического проекта	10 часов	Вс 15.01.23	Чт 19.01.23	8;10;11
13	▫ Реализация	317 часов	Пт 20.01.23	Пн 15.05.23	
14	▫ Реализация первой версии	220 часов	Пт 20.01.23	Ср 03.05.23	
15	Реализация MVP	80 часов	Пт 20.01.23	Вт 28.02.23	12
16	Разработка дизайна+верстка	50 часов	Ср 01.03.23	Сб 25.03.23	15
17	Реализация агрегатора	40 часов	Вс 26.03.23	Пт 14.04.23	16
18	Заполнение БД	10 часов	Вс 26.03.23	Чт 30.03.23	16
19	Реализация остального функционала	50 часов	Сб 15.04.23	Ср 03.05.23	17
20	Реализация второй версии	97 часов	Ср 03.05.23	Пн 15.05.23	14
21	Тестирование	30 часов	Ср 03.05.23	Сб 06.05.23	14
22	▫ Ввод в действие и сопровождение	51 часов	Вс 07.05.23	Сб 13.05.23	
23	Аренда хостинга	10 часов	Вс 07.05.23	Пн 08.05.23	14;21
24	Внедрение на хостинг	41 часов	Пн 08.05.23	Сб 13.05.23	23

Рисунок 6.1 – Лист задач

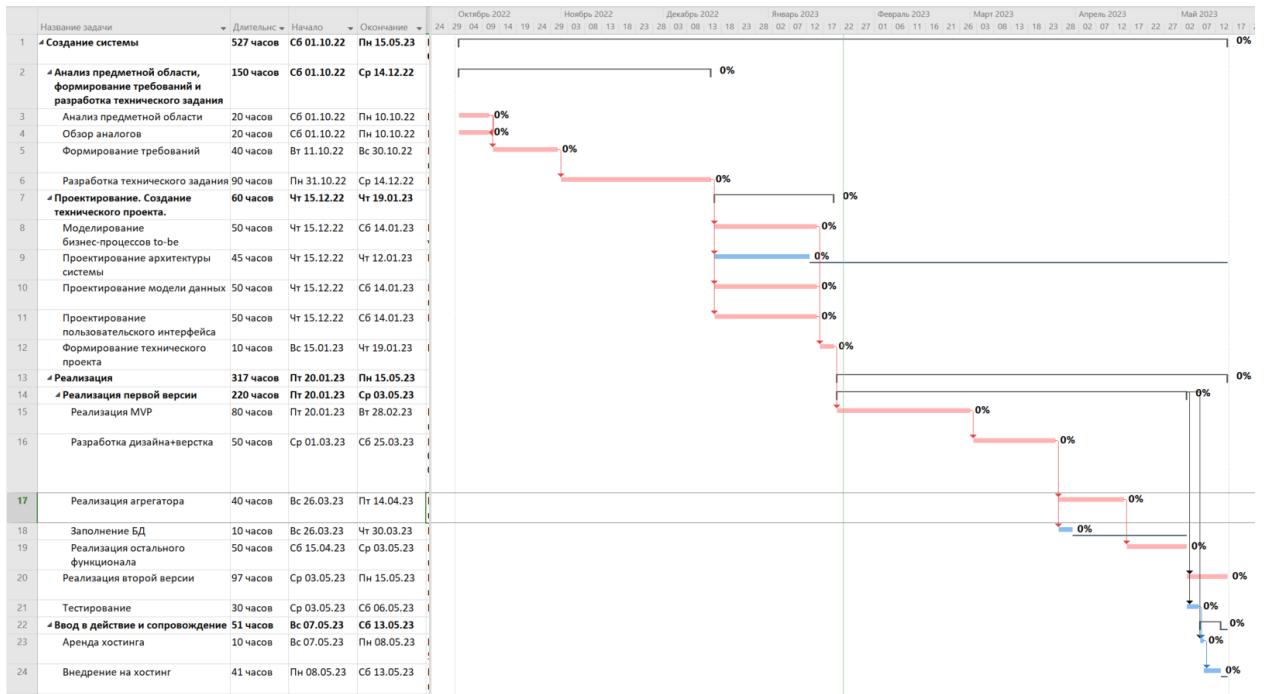


Рисунок 6.2 – Диаграмма Ганта

После определения календарного плана необходимо определить ресурсы и их стоимость (рис. 6.3). Заработная плата разработчика была взята исходя из средней по рынку для данной квалификации [14]. Помимо разработчика в процессе участвует технический консультант. Он помогает принимать решения в проектировании и решать сложные вопросы при разработке. Стоимость его заработной платы составляет 500 рублей в час. При реализации системы необходимо провести консультацию с юристом для создания юридических документов (пользовательское соглашение). Стоимость консультации равна 10 000р. Из-за отсутствия достаточных навыков разработки дизайна системы разработчику необходимо пройти курс по дизайну сайтов (10 000р.), создать свой дизайн и затем провести консультацию с дизайнером (2 000р) для выявления проблем. Такое решение проблемы дешевле, потому что стоимость работ по созданию дизайна с нуля начинается от 30 000р. Для разработки необходим обязательно ПК, но приобретение нового слишком дорого. Поэтому было принято решение использовать личный ноутбук. Так же для работы необходим интернет, бумага, канцелярия и арендованный хостинг. Все ресурсы показаны на рисунке 6.3.

Тип: Затраты	Затраты	Сроки	Ставка	Сверхурочная	Итоги
Хостинг	Затраты				
Интернет	Затраты				
Консультация с юристом	Затраты				
Консультация дизайнера	Затраты				
Курс по дизайну	Затраты				
▲ Тип: Материальный	Материальны				0,00р.
ПК	Материальн		0,00р.		0,00р.
Бумага	Материальн		500,00р.		0,00р.
Канцелярия	Материальн		300,00р.		0,00р.
▲ Тип: Трудовой	Трудовой	200%			0,00р.
Разработчик	Трудовой	100%	350,00р./час	350,00р./час	0,00р.
Технический консультант	Трудовой	100%	500,00р./час	0,00р./час	0,00р.

Рисунок 6.3 – Лист ресурсов

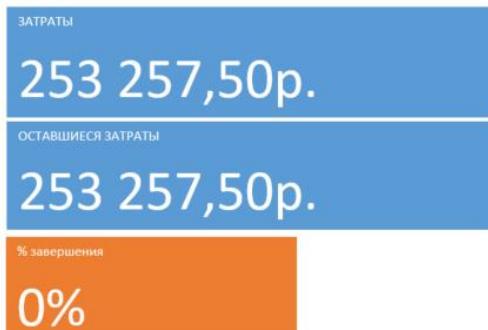
	Название задачи	Длительн	Начало	Окончание	Пред	Названия ресурсов	Затраты
1	▲ Создание системы	527 часов	Сб 01.10.22	Пн 15.05.23		Консультация с юристом[10 000,00р.];ПК[1];Интернет[4 000,00р.]	253 257,50р.
2	▲ Анализ предметной области, формирование требований и разработка технического задания	150 часов	Сб 01.10.22	Ср 14.12.22			57 690,00р.
3	Анализ предметной области	20 часов	Сб 01.10.22	Пн 10.10.22		Разработчик[50%]	3 500,00р.
4	Обзор аналогов	20 часов	Сб 01.10.22	Пн 10.10.22	300	Разработчик[50%];Бумага[0,2 уп.];Канцелярия[0,3]	3 690,00р.
5	Формирование требований	40 часов	Вт 11.10.22	Вс 30.10.22	4;3	Интернет;ПК[1];Разработчик;Технический консультант	19 000,00р.
6	Разработка технического задания	90 часов	Пн 31.10.22	Ср 14.12.22	5	Интернет;ПК[1];Разработчик	31 500,00р.
7	▲ Проектирование. Создание технического проекта.	60 часов	Чт 15.12.22	Чт 19.01.23			24 942,50р.
8	Моделирование бизнес-процессов to-be	50 часов	Чт 15.12.22	Сб 14.01.23	6	Разработчик[25%];Бумага[0,2 уп.];Канцелярия[0,3]	4 565,00р.
9	Проектирование архитектуры системы	45 часов	Чт 15.12.22	Чт 12.01.23	6	Разработчик[25%]	3 937,50р.
10	Проектирование модели данных	50 часов	Чт 15.12.22	Сб 14.01.23	6	Разработчик[25%];Технический консультант;Бумага[0,2 уп.];Канцелярия[0,3]	8 565,00р.
11	Проектирование пользовательского интерфейса	50 часов	Чт 15.12.22	Сб 14.01.23	6	Разработчик[25%]	4 375,00р.
12	Формирование технического проекта	10 часов	Вс 15.01.23	Чт 19.01.23	8;10;11	Разработчик	3 500,00р.
13	▲ Реализация	317 часов	Пт 20.01.23	Пн 15.05.23			142 000,00р.
14	▲ Реализация первой версии	220 часов	Пт 20.01.23	Ср 03.05.23			112 375,00р.
15	Реализация MVP	80 часов	Пт 20.01.23	Вт 28.02.23	12	Разработчик;Технический консультант	34 000,00р.
16	Разработка дизайна+верстка	50 часов	Ср 01.03.23	Сб 25.03.23	15	Разработчик;Курс по дизайну[10 000,00р.];Консультация дизайнера[2 000,00р.]	29 500,00р.
17	Реализация агрегатора	40 часов	Вс 26.03.23	Пт 14.04.23	16	Разработчик[75%];Технический консультант	20 500,00р.
18	Заполнение БД	10 часов	Вс 26.03.23	Чт 30.03.23	16	Разработчик[25%]	875,00р.
19	Реализация остального функционала	50 часов	Сб 15.04.23	Ср 03.05.23	17	Разработчик;Технический консультант	27 500,00р.
20	Реализация второй версии	97 часов	Ср 03.05.23	Пн 15.05.23	14	Разработчик[50%];Технический консультант[50%]	27 000,00р.
21	Тестирование	30 часов	Ср 03.05.23	Сб 06.05.23	14	Разработчик[25%]	2 625,00р.
22	▲ Ввод в действие и сопровождение	51 часов	Вс 07.05.23	Сб 13.05.23			14 625,00р.
23	Аренда хостинга	10 часов	Вс 07.05.23	Пн 08.05.23	14;21	Разработчик[25%];Хостинг[2 500,00р.]	3 375,00р.
24	Внедрение на хостинг	41 часов	Пн 08.05.23	Сб 13.05.23	23	Разработчик[50%];Технический консультант[50%]	11 250,00р.

Рисунок 6.4 – Диаграмма Ганта с ресурсами

После определения задач и ресурсов необходимо каждой задаче назначить ресурсы. Во всех задачах используется большинство ресурсов. Был установлен календарный план: разработчик работает над проектом каждый день по 2 часа 7 дней в неделю, с 31.12.2022 по 05.01.2023 выходные из-за новогодних праздников, а в мае 2023 года рабочий день с 11:00 до 19:00 из-за преддипломной практики. Благодаря назначению ресурсов (рис. 6.4) появилась возможность узнать общую стоимость разработки проекта. На рисунке 6.5 показан отчет «Обзор затрат». Общая сумма затрат составляет 253 257,50р. Так же на рисунке присутствуют диаграммы, которые отображают состояние затрат и ход выполнения относительно затрат. Самым дорогим является этап реализации, а затем идет этап анализа.

ОБЗОР ЗАТРАТ

СБ 01.10.22 – ПН 15.05.23



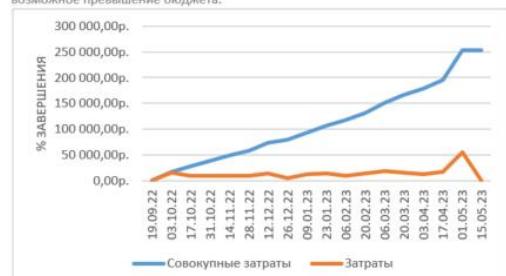
СОСТОЯНИЕ ЗАТРАТ

Состояние затрат для задач верхнего уровня.

Название	Фактические затраты	Оставшиеся затраты	Базовые затраты	Затраты	Отклонение по стоимости
Анализ предметной области, формирование требований и разработка технического задания	0,00р.	57 690,00р.	0,00р.	57 690,00р.	57 690,00р.
Проектирование, Создание технического проекта.	0,00р.	24 942,50р.	0,00р.	24 942,50р.	24 942,50р.
Реализация	0,00р.	142 000,00р.	0,00р.	142 000,00р.	142 000,00р.
Ввод в действие и сопровождение	0,00р.	14 625,00р.	0,00р.	14 625,00р.	14 625,00р.

ХОД ВЫПОЛНЕНИЯ И ЗАТРАТЫ

Ход выполнения в сравнении с затратами во времени. Превышение значения в строке "Совокупные затраты" над значением в строке "Процент завершения" указывает на возможное превышение бюджета.



СОСТОЯНИЕ ЗАТРАТ

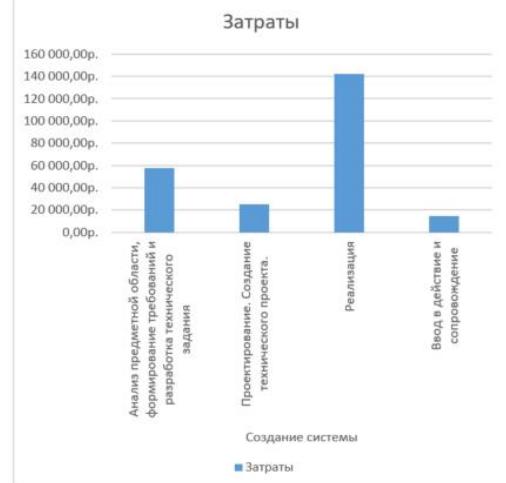


Рисунок 6.5 – Отчет «Обзор затрат»

Планирование и расчет затрат на создание системы показал, что стоимость готового программного продукта составляет 253 257 р. при длительности разработки около 7,5 месяцев.

Заключение

В результате курсового проекта была разработана и частично реализована система, которая отвечает поставленной цели, а именно: сопровождение процесса повышения уровня компетенций в командной разработке посредством нетворкинга. В процессе разработки были решены следующие задачи:

1. Выполнен анализ предметной области;
2. Сформированы требования к системе;
3. Выполнено проектирование системы;
4. Реализовано MVP;
5. Выполнен расчет экономической части;
6. Составлен отчёт по всей проделанной работе.

Выполнение задач привело к следующему результату: реализовано 12 страниц в веб-приложении, а также 15 API запросов в HTTP сервере. В результате получился минимально жизнеспособный продукт, который станет фундаментом при выполнение выпускной квалификационной работы.

В дальнейшем планируется полная реализация, проработка дизайна, ввод в действие, продвижение и развитие сайта. Так же была выявлена потребность в реализации дополнительных особенностей системы: интеграция с другими сервисами (агрегирование информации о пользователе), авторизация с помощью сторонних систем, мобильная верстка, рассылка новостей и автоматизирование модерации, с помощью применения библиотек фильтрации нецензурной лексики.

Исходный код приложения опубликован в публичном репозитории на сервисе GitHub [15].

Список использованных источников

1. Хакатоны: как они устроены, как проходят и чем отличаются от чемпионатов по анализу данных? // Tproger URL: <https://tproger.ru/articles/hakatony-kak-oni-ustroeny-kak-prohodjat-i-chem-otlichajutsja-ot-championatov-po-analizu-dannyh/> (дата обращения: 05.10.2022).
2. ГОСТ 34.602— 2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Взамен ГОСТ 34.602— 89; Москва: Российский институт стандартизации, 2021. – 12 с.
3. Структура сайта "Биржа опыта" // Octopus.do URL: <https://octopus.do/nwi79o50tnl> (дата обращения: 15.11.2022).
4. РД IDEF0 - 2000. Методология функционального моделирования IDEF0. Руководящий документ.; Москва: Госстандарт России, 2000. – 75 с.
5. Ramus Educational 1.1 // Software.Informer URL: <https://ramus-educational.software.informer.com/> (дата обращения: 07.12.2022).
6. The C4 model for visualising software architecture // C4model URL: <https://c4model.com/> (дата обращения: 22.12.2022).
7. Прогрессивный JavaScript-фреймворк // Vue.js URL: <https://v3.ru.vuejs.org/> (дата обращения: 25.12.2022).
8. A progressive Node.js framework // NestJS URL: <https://nestjs.com/> (дата обращения: 27.12.2022).
9. The most popular HTML, CSS, and JS library in the world. // Bootstrap URL: <https://getbootstrap.com/> (дата обращения: 05.01.2023).
10. Web Scraping с помощью python // Habr URL: <https://habr.com/ru/post/280238/> (дата обращения: 07.01.2023).
11. Физическая модель данных "Биржа опыта" // DrawSQL URL: <https://drawsql.app/teams/ne-izvestno/diagrams/fizicheskaya-model-dannyh-birzha-opyta> (дата обращения: 30.12.2022).
12. Макеты Биржа опыта // Figma URL: <https://www.figma.com/file/MwEZrUat1v0jSNV96TtIvO/%D0%91%D0%8B%D1%80%D0%B6%D0%B0-%D0%BE%D0%BF%D1%8B%D1%82%D0%B0?node-id=0%3A1&t=PCfel2ChGh8bvDHC-1> (дата обращения: 01.12.2022).
13. Справка и обучение по Project // Microsoft Office URL: <https://support.microsoft.com/ru-ru/project> (дата обращения: 02.11.2022).
14. Зарплаты: Junior разработчик, Россия | декабрь 2022 года // Zarplan URL: <https://zarplan.com/zarplata/JUNIOR%20РАЗРАБОТЧИК/РОССИЯ/> (дата обращения: 20.01.2023).
15. Репозиторий experience_market // GitHub URL: https://github.com/KomogortsevaYulia/experience_market (дата обращения: 01.11.2022).

Приложение А

Таблицы

Таблица А.1 – Обзор аналогов

№	Цель и/или описание сервиса	1	2	3	4	5	6	7	8	9	Итог
1	Хабр Фриланс(проекты) - Биржа фриланса, есть модуль проекты и специалисты, однако все направлено на ЗП https://freelance.habr.com/tasks	Да	1	1	0	0	0	ru	1	0	3
2	Russian Hackers - сообщество участников хакатонов, разработчиков, предпринимателей. Мы рассказываем, организуем и делимся экспертизой о хакатонах https://russianhackers.org/hackathons	Да	0	0	1	0	1	ru	1	1	4
3	Leader-ID - мероприятия, коворкинги, сообщества и «социальные лифты». Leader-ID помогает работать, учиться и находить единомышленников в любом городе https://leader-id.ru/	Да	0	0	1	1	1	ru	1	1	5
4	ODS - Международное сообщество, объединяющее специалистов, исследователей и инженеров, связанных с Data Science https://ods.ai/	Нет	1	1	1	1	0	ru	0	0	4
5	HACKATHONS.PRO - Мы агрегируем интересные ИТ-соревнования, хакатоны, чемпионаты, олимпиады по программированию и разработке, проходящие в России и мире. https://hackathons.pro/	Да	0	0	1	0	1	ru	0	1	3
6	Хакатоны.рф - На открытом ресурсе регулярно публикуются актуальные данные о предстоящих хакатонах. https://www.хакатоны.рф/	Да	0	0	1	0	1	ru	1	1	4

Продолжение таблицы А.1

№	Цель и/или описание сервиса	1	2	3	4	5	6	7	8	9	Итог
7	Codenrock - компания организации хакатонов. Участвуйте в онлайн и офлайн хакатонах, чтобы создавать новые продукты, практиковать навыки, изучать технологии, выигрывать призы и развивать нетворкинг https://codenrock.com/	Да	0	0	1	1	1	ru	1	1	5
8	ФИЗТЕХ ДЖЕНЕЗИС – компания организации хакатонов для взаимовыгодного развития бизнеса и ИТ-специалистов. https://pgenesis.ru/events	Да	0	0	1	1	1	ru	1	1	5
9	What to Code? - было создано во время пандемии для помощи разработчикам собирать идеи для их следующего проекта по кодированию. https://what-to-code.com/	Да	1	1	0	0	0	en	0	1	3
10	Programming Challenges - сервис с рулеткой проектов. Проекты добавлять нельзя, список фиксирован. https://programming-challenges.jeremyjaydan.dev/	Нет	1	0	0	0	0	en	0	1	2
11	Google Summer of Code - глобальная онлайн-программа наставничества, направленная на привлечение новых участников к разработке программного обеспечения. Участники работают над 12-недельным проектом по программированию под руководством наставников из их организаций. https://summerofcode.withgoogle.com/archive/2021/organizations	Нет	1	0	0	0	1	en	0	0	2
12	Devpost - создавайте продукты, практикуйте навыки, изучайте технологии, выигрывайте призы и расширяйте свою сеть. https://devpost.com/	Да	1	1	1	1	1	en	1	1	7

Продолжение таблицы А.1

№	Цель и/или описание сервиса	1	2	3	4	5	6	7	8	9	Итог
13	Календарь мероприятий экосистемы НТИ-2022 - здесь вы найдете информацию о готовящихся хакатонах и олимпиадах, школах, крупнейших отраслевых форумах и акселераторах в 2022 году. Выбирайте самые интересные события и до встречи на мероприятиях! https://2035.events/	Да	0	0	1	0	0	ru	1	1	3
14	Хакатоны.рус - проект создан АНО "Цифровая молодежь" для агрегации IT-соревнований, хакатонов, чемпионатов, олимпиад по программированию и разработке, проходящих в России. https://хакатоны.рус/	Да	0	0	1	0	1	ru	1	1	5
15	Kaggle - система организации конкурсов по исследованию данных, а также социальная сеть специалистов по обработке данных и машинному обучению. https://www.kaggle.com/	Да	1	1	0	0	1	en	1	0	5
16	КиберФорум - форум программистов и системных администраторов. Бесплатная помощь в решении задач связанных с IT. https://www.cyberforum.ru/	Нет	0	0	0	0	0	ru	0	0	0
17	Хеомате - бесплатный сервис для поиска единомышленников. Хеомате Вы можете найти команду для старта или хакатона, найти менторов, учеников, инвесторов, а также друзей для простого общения. https://app.xeomate.com/platform/profile-list	Нет	0	1	0	1	0	ru	1	0	3