

Красноярск 2023

РЕФЕРАТ

Выпускная квалификационная работа по теме «Онлайн-сервис для планирования мероприятий и участия в них» содержит 49 страниц, 2 таблицы, 30 рисунков, 21 использованный источник, 1 приложение.

СОЦИАЛЬНАЯ СЕТЬ, КЛИЕНТ, СЕРВЕР, VUE.JS, JAVA SCRIPT, ВЕБ-ПРИЛОЖЕНИЕ.

Целью бакалаврской работы является разработка и создание удобного онлайн сервиса для поиска и проведения мероприятий, встреч и собраний.

Основные задачи:

- анализ технического задания;
- обзор предметной области, обзор аналогов;
- выбор средств разработки;
- проектирование системы;
- создание дизайна приложения;
- реализация;
- тестирование;
- создание руководства пользователя.

При выполнении работы были исследованы существующие методы решения задачи и сформированы требования к разрабатываемой системе.

В результате выполнения ВКР был спроектирован и разработан онлайн сервис, удовлетворяющий поставленным требованиям к системе.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ задания на выпускную квалификационную работу	6
1.1 Описание предметной области	6
1.2 Аналитический обзор существующих сервисов	7
1.2.1 Критерии оценки аналогов.....	8
1.2.2 Kuda Moscow	8
1.2.3 Kudago	9
1.2.4 ВК сообщества.....	10
1.2.5 Яндекс Афиша	11
1.2.6 Планета семинаров.....	12
1.2.7 Timerpad.....	13
1.3 Вывод по обзору аналогов	14
1.4 Функциональные требования	15
1.5 Вывод по главе	15
2 Проектирование системы	17
2.1 Общая структура разрабатываемой системы	17
2.1 Выбор средств и технологий разработки	18
2.1.1 Клиентская часть	18
2.1.2 Серверная часть.....	18
2.1.3 Формат запроса данных у сервера.....	19
2.1.4 База данных.....	19
2.1.5 Методология проектирования ICONIX.....	20
2.2 Диаграммы прецедентов	21
2.2.1 Диаграммы прецедентов актора «гость»	21
2.2.2 Диаграммы прецедентов актора «пользователь»	23
2.2.3 Диаграммы прецедентов актора «модератор».....	25
2.3 Проектирование интерфейса системы.....	26
2.3.1 Модель предметной области	26
2.3.2 Разработка диаграммы классов.....	27
2.3.3 Диаграмма потока экранов	27
2.4 Проектирование базы данных	28
2.4.1 Логическая модель данных предметной области.....	28
2.5 Вывод по главе	29

3 Реализация.....	30
3.1 Реализация клиентской части	30
3.2 Описание HTTP методов.....	31
3.3 Формат обмена данными.....	32
3.4 Яндекс карта	33
3.5 Реализация серверной части.....	34
3.6 Реализация базы данных	34
3.7 Инструкция пользователя	36
3.7.1 Основные функции	36
3.7.2 Дополнительные функции.....	42
3.8 Вывод по главе	44
Заключение.....	45
Список использованных источников.....	46
Приложение А Общая диаграмма прецедентов	48
Приложение Б Диаграмма потока экранов	49

ВВЕДЕНИЕ

21 век – время технологий и интернета. Цивилизация не стоит на месте, и с каждым годом появляются сотни новых изобретений, меняющих жизнь современных людей. Технологии изменили и социальные взаимоотношения между людьми. Технически общаться стало гораздо проще, больше нет необходимости встречаться лично, чтобы поделиться информацией; скорость доставки сообщений между городами и странами преодолела возможные барьеры. В то же время эти изменения помимо положительных эффектов привели к замещению реальной коммуникации виртуальным общением. Обеспечение баланса реального и виртуального общения на сегодняшний день очень актуально. Чтобы его достичь, необходимо увеличить количество мероприятий, проводимых в реальной среде и предполагающих реальное общение. Однако организация офлайн-мероприятия, особенно силами небольшой инициативной группы или отдельного человека, и привлечение достаточного числа участников является сложной задачей.

Существует множество сайтов, позволяющих посмотреть спортивные события, концерты, выставки, и зарегистрироваться на них. Однако на большинстве таких сайтов публикуются только официальные мероприятия, и отсутствует возможность создания мероприятия сторонним пользователем. Использование для этих целей социальных сетей ограничивает число потенциальных участников мероприятия группой пользователей конкретной социальной сети.

Для решения данной проблемы предлагается разработать онлайн-сервис, который должен позволить человеку самостоятельно зарегистрировать мероприятие и провести его. Таким мероприятием может быть бесплатный концерт музыкальной группы, любительская игра в волейбол, киновечер, кружок вязания и др. Это позволит пользователям найти новых знакомых и друзей, повысить социальную активность, реализовать свой творческий и интеллектуальный потенциал.

Цель работы: реализация онлайн-сервиса для поиска и организации проведения мероприятий различной направленности (культурных, зрелищных, спортивных и др.).

Для достижения поставленной цели необходимо решить следующие задачи:

- анализ задания, определение функционала разрабатываемой системы;
- изучение предметной области, обзор аналогов;
- выбор средств разработки и проектирование системы;
- реализация системы;
- создание руководства пользователя.

Первый раздел выпускной квалификационной работы посвящен обзору аналогов, полному выявлению требований к системе, основываясь на анализе уже реализованных приложений, поиск подходящих средств разработки.

Второй раздел посвящен проектированию системы, выявлению прецедентов и их описанию, проектированию моделей для трех акторов: гость, пользователь и модератор. Формирование функций для каждого актора.

Третий раздел представляет собой реализацию системы и описание функциональной части системы.

Данная работа подразумевает разработку как клиентской, так и серверной части. Итогом выполнения ВКР является полностью рабочее веб приложение.

1 Анализ задания на выпускную квалификационную работу

1.1 Описание предметной области

Предметная область разрабатываемого сервиса охватывает не только поиск мероприятий, но и создание их.

«Мероприятие» – в данной системе подразумевает собой событие, организованное одним из пользователей, с определенной тематикой, для группы людей, зарегистрированных в сервисе. Мероприятие является встречей пользователей по приглашению пользователя организатора мероприятия.

«Организатор мероприятия» – в данной системе подразумевает собой зарегистрированного пользователя, проявившего желание устроить встречу определенной тематики.

Проведение официальных мероприятий (официальные концерты, выступления, выставки и т.д.) возможно при условии, что их «организатором» будет официальный представитель площадки проведения мероприятия.

Сложность данной предметно области состоит в том, что её составляющие (мероприятия) постоянно изменяются, такая система должна отслеживаться и быть контролируема в реальном времени.

Проведение мероприятий, особенно массовых, регулируется законодательством.

Федеральный закон от 19.06.2004 № 54-ФЗ (ред. от 11.10.2018) «О собраниях, митингах, демонстрациях, шествиях и пикетированиях» указывает, что проводить мероприятия нельзя:

В местах, которые не отвечают безопасности участников и зрителей. Это, например, аварийное здание, заброшенная заводская территория и др. Если под мероприятие нужно определенное место, следует уточнить у органов местного самоуправления, не пересекается ли мероприятие с другими.

На ряде территорий, например, возле транспортных магистралей, путепроводов, электростанций, железнодорожных дорог, нефтепроводов,

газопроводов в других местах, где требуется соблюдение специальных правил безопасности, проведение мероприятий запрещено.

Если организатор планирует проводить мероприятие на территории объектов культурного и исторического наследия, где часто пустует большое пространство, то потребуется разрешение от органов исполнительной власти.

Федеральный закон от 29.12. 2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и развитию» требует, чтобы в случае, если мероприятие имеет ограничение по возрасту участников, в описании была указана эта информация. Также в этом случае чтобы избежать проблем, организатору следует контролировать возраст детей. Поэтому на странице мероприятия должна быть информация о том, что на входе будет контроль и всем участникам нужно предъявить паспорт.

Время проведения мероприятия тоже регулируется законом и допускается в промежутке с 7:00 до 22:00. Исключением являются только салюты в памятные даты и общегородские культурные мероприятия.

Поскольку не все инициаторы мероприятий могут быть знакомы с правовыми основами регулирования проведения мероприятий, сервис должен иметь модерацию.

1.2 Аналитический обзор существующих сервисов

Для более точного формулирования требований к системе, проведем анализ наиболее популярных аналогов.

Для сравнения выбраны следующие аналоги:

- kuda Moscow [2];
- kudago [3];
- ВК сообщества [4];
- Яндекс афиша [5];
- Планета семинаров [6];
- Timerpad [7].

1.2.1 Критерии оценки аналогов

На основании поставленной цели и выделенных задач были определены следующие критерии оценки сервисов со сходным функционалом.

Основные критерии:

- стоимость услуг: размещение и поиск мероприятий без взимания платы;
- всероссийская платформа: доступность платформы в более чем 10 городов России;

Реализуемый функционал:

- создание мероприятий: функция создания и регистрации своего мероприятия;
- личный кабинет: личный кабинет и основной информацией о пользователе и его мероприятиях;
- запись на мероприятие: функция регистрации на мероприятие;
- чат: возможность общения между участниками мероприятия или возможность общения между участником и организатором или возможность оставлять отзывы о прошедшем мероприятии.

Удобство использования (юзабилити):

- удобство поиска мероприятий: интуитивно понятный интерфейс, без необходимости поиска основных функций по сайту;
- мобильная версия: адаптивный дизайн интерфейса для телефонов или отдельное приложение для телефонов;
- адаптивный интерфейс: интерфейс, корректно изменяющийся, под разные размеры экрана;
- общая карта: карта города с указанными на ней мероприятиями.

1.2.2 Kuda Moscow

Kuda Moscow [2] – это сервис, на котором публикуются самые популярные события и мероприятия города Москва. На сайте можно узнать про события, которые проходят сегодня или ближайшее время. Как следует из

названия, Kuda Moscow – это локальный сервис, предназначенный только для жителей Москвы, это является его главным минусом. Из плюсов следует отметить удобство пользования, понятный дизайн, хорошее описание событий, разделение событий на категории, отсутствие рекламы. Однако на сервисе предоставлены только официальные события, что не отвечает требованиям для создаваемого нами сайта. Интерфейс приложения представлен на рисунке 1.

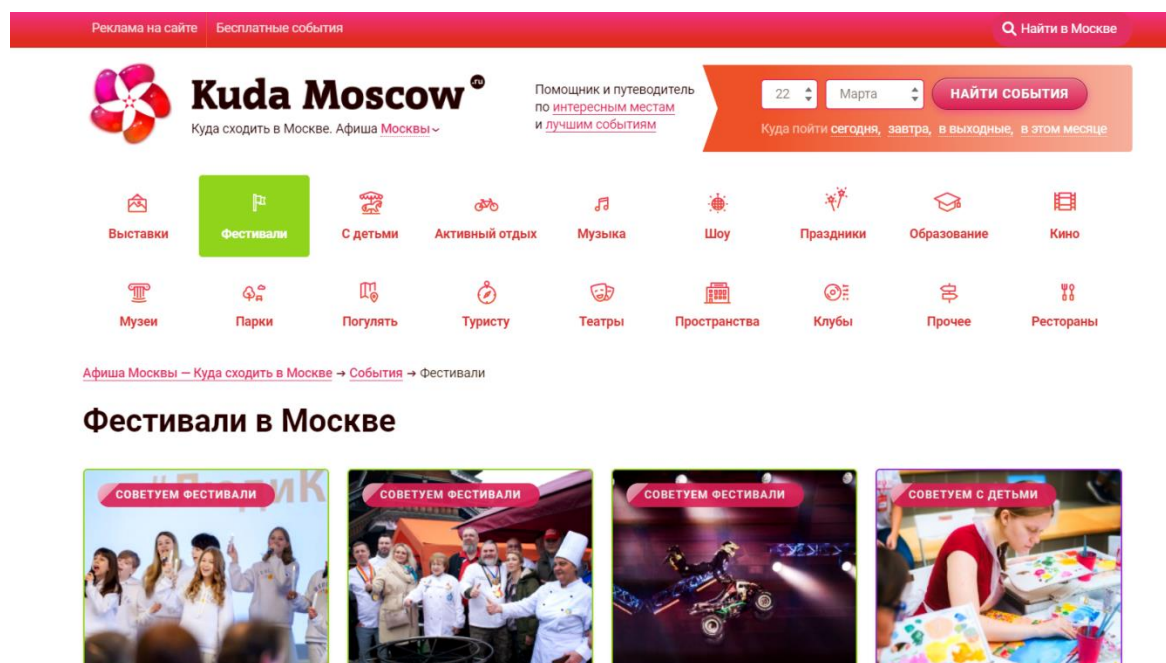


Рисунок 1 – Интерфейс приложения Kuda Moscow

1.2.3 Kudago

Kudago [3] – сервис, для поиска мероприятий, в отличие от предыдущего аналога, Kudago распространяется на 10 городов, таких как: Москва, Санкт-Петербург, Казань, Новосибирск, Екатеринбург, Нижний Новгород, Красноярск, Краснодар, Сочи, Минск. Из плюсов следует отметить минималистический дизайн, что делает его понятным для пользователей, которые редко пользуются компьютером. Из минусов отметим большое количество рекламы. Также, сервис поддерживает только официальные события, что не отвечает нашим требованиям. Интерфейс приложения представлен на рисунке 2.

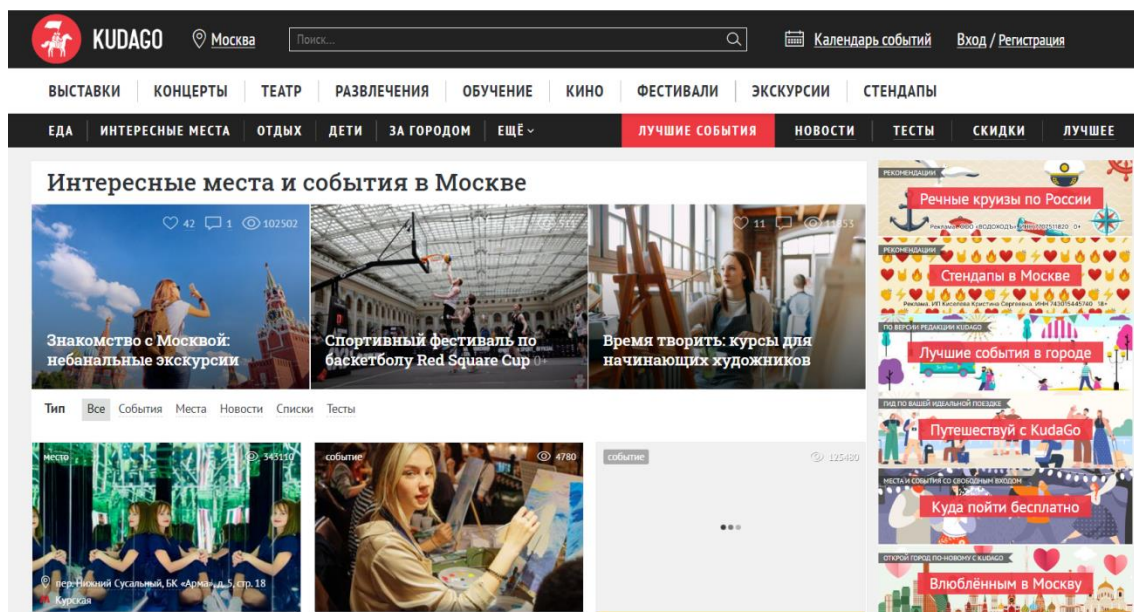


Рисунок 2– Интерфейс сервиса Kudago

1.2.4 ВК сообщества

ВК сообщества [4] – это группы в социальной сети «ВКонтакте», которые создают организаторы событий. В качестве событий может быть официальное мероприятие, такое как концерт или театральное выступление, а так же, неофициальное собрание группы людей, например, встреча фанатов или любительская игра в футбол. В качестве функционала присутствует: чат между участниками, кнопка «участвовать», можно присоединиться к событию, отслеживать новости сообщества, оставляя комментарии. Данный функционал отвечает требованиям проектируемого нами приложения. Однако существует ряд минусов, из-за которых данный аналог не может полностью заменить разрабатываемую систему, таких как:

- отсутствие контроля безопасности, «ВКонтакте» не отвечает за безопасность пользователей, собирающихся на мероприятия;
- нет поиска по категориям, реализована только функция поиска по названию конкретного мероприятия, что очень неудобно, ведь пользователь может не знать, как организаторы назвали мероприятие, следовательно, пропустит его;
- отсутствует карта со всеми мероприятиями;

- «ВК сообщества» являются частью социальной сети, и ими нельзя воспользоваться, не войдя в свой аккаунт.

Интерфейс приложения представлен на рисунке 3.

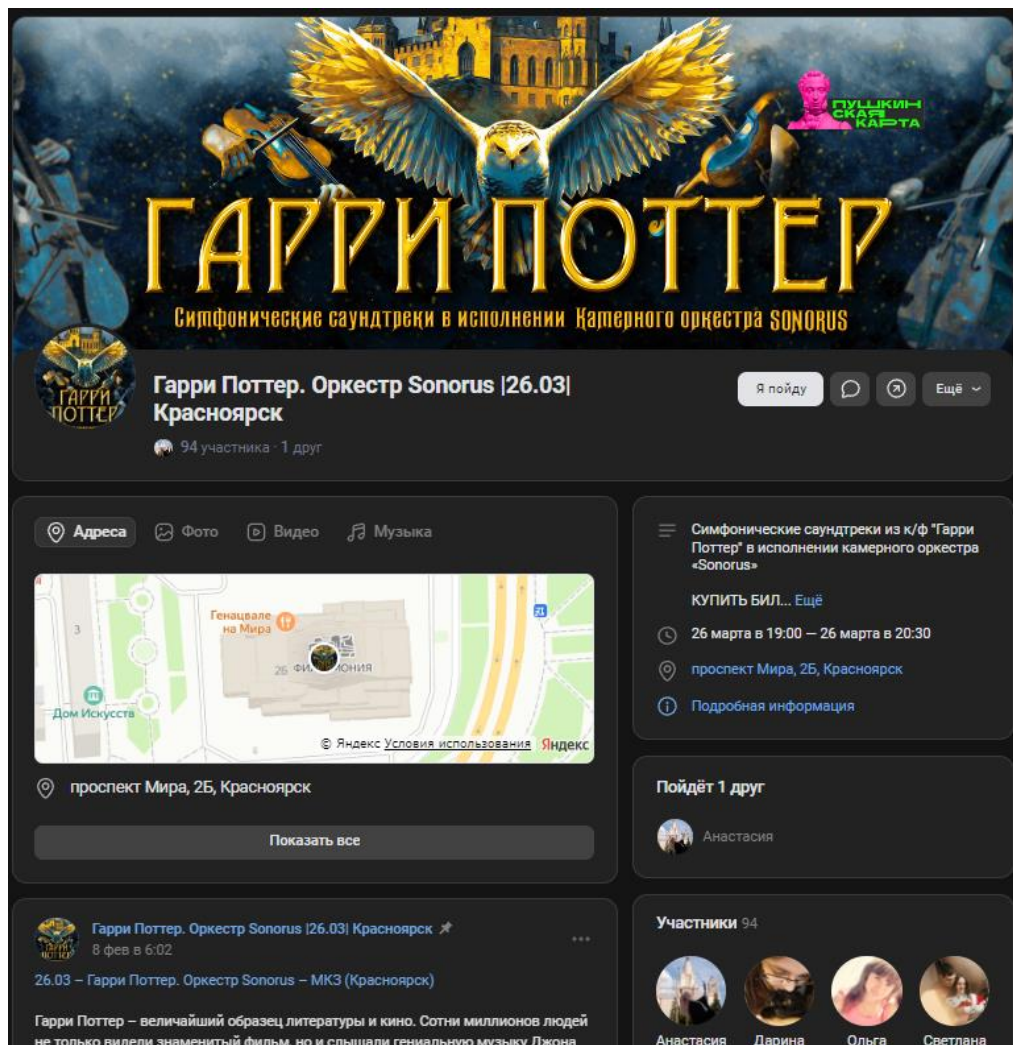


Рисунок 3 – Пример интерфейса ВК сообщества

1.2.5 Яндекс Афиша

Яндекс Афиша [5] – один из самых популярных аналогов, который рассматривается в данной работе. Функционал стандартный для данной категории сайтов, черты, отличительные от предыдущих аналогов – приятный минималистичный интерфейс, большое количество событий и мероприятий, а так же, система скидок и акций. Из минусов следует отметить наличие рекламы, однако, она не мешает пользователю пользоваться функционалом сайта. Также отсутствует возможность создания собственных мероприятий, что

не отвечает требованиям проектируемого приложения. Интерфейс приложения представлен на рисунке 4.

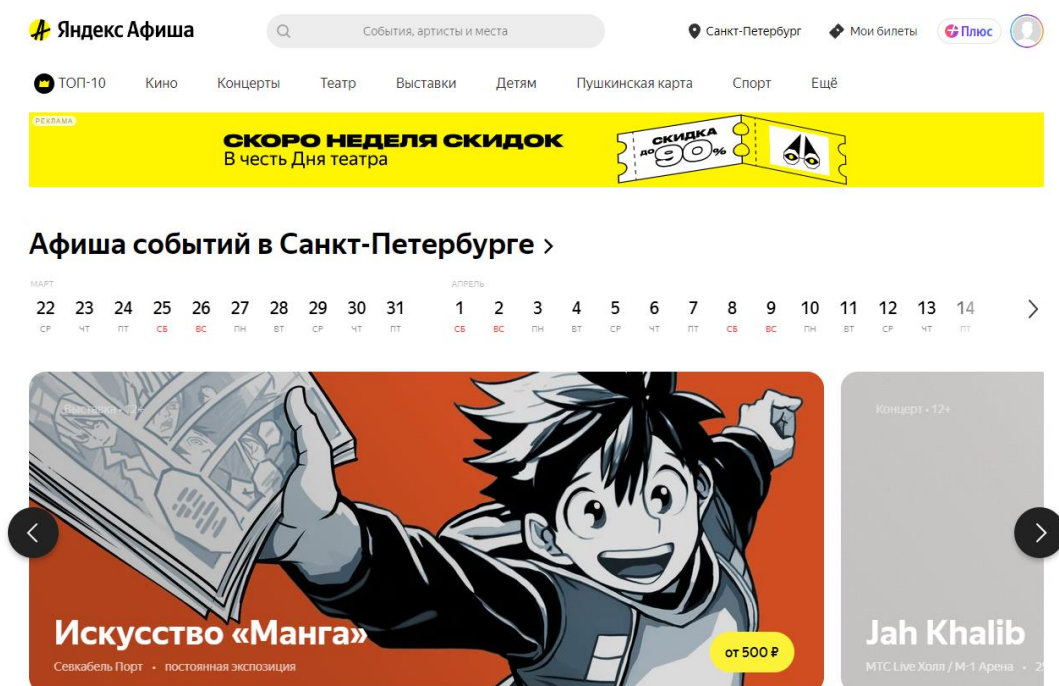


Рисунок 4 – Интерфейс сервиса Яндекс Афиша

1.2.6 Планета семинаров

Планета семинаров [6] – наиболее подходящая платформа по функционалу, однако список мероприятий ограничивается тренингами и семинарами. Отличительной чертой является возможность добавления своего семинара. Интерфейс приложения представлен на рисунке 5.

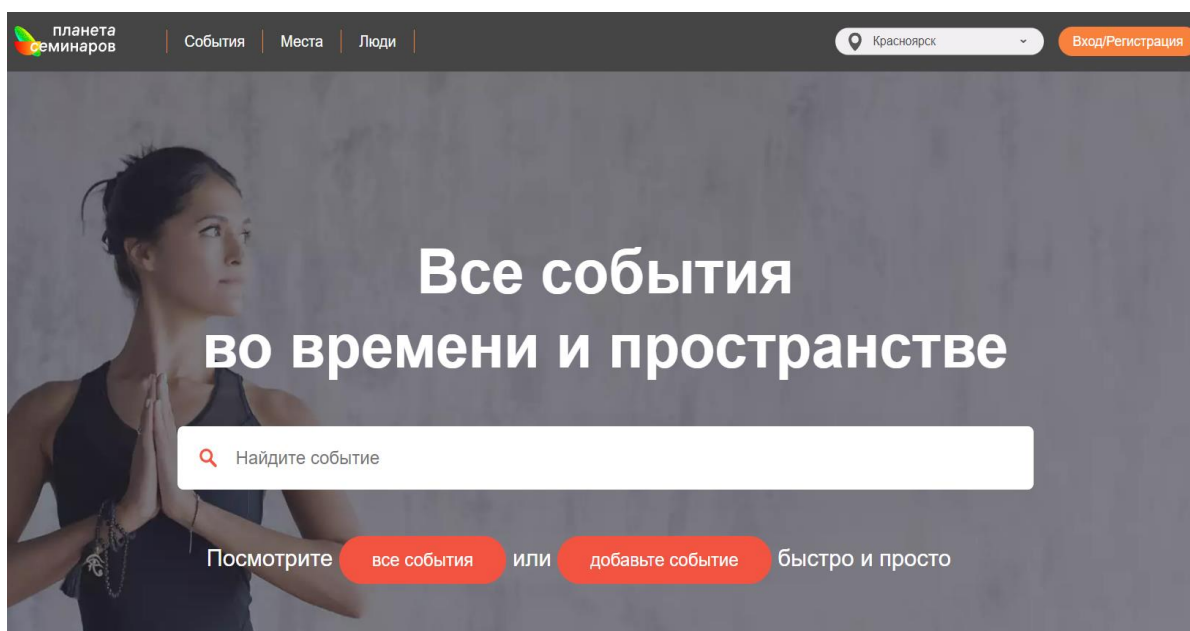


Рисунок 5 – Интерфейс сервиса Планета семинаров

1.2.7 Timerpad

Timerpad [7] – популярный сервис с большим функционалом по продвижению мероприятий.

Возможность размещать мероприятия на этом сервисе присутствует только для официальных организаций. Интерфейс приложения представлен на рисунке 6.

Рекомендуем события

[Тема](#)
[Дата](#)
[Цена](#)

Концерты >

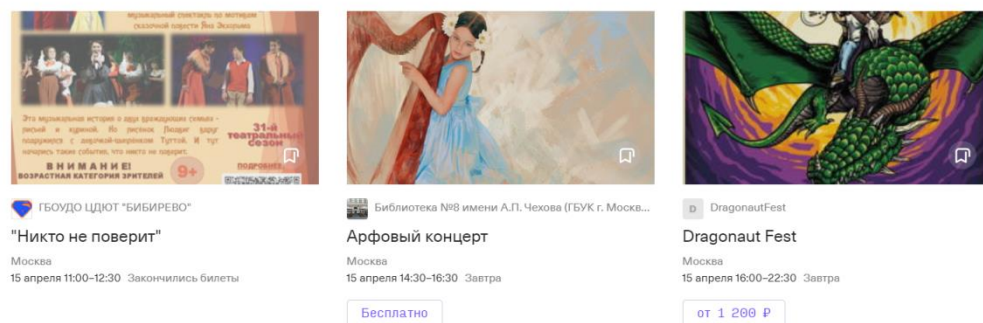


Рисунок 6 – Интерфейс сервиса TimeraD

1.3 Вывод по обзору аналогов

Проанализировав данные аналоги, можно вывести список признаков, по которым можно описать систему, подходящую под требования. Рассмотрим признаки аналогов (Таблица 1).

Таблица 1 – Обзор аналогов

Признаки	Название	kuda Moscow	kudago	ВК сообщества	Яндекс афиша	Планета семинаров	TimeraD
Чат		+	+	+	-	-	-
Бесплатное размещение		-	-	+	-	+	-
Мобильная версия		+	-	+	+	-	+
Личный кабинет		-	+	+	+	+	+
Запись на мероприятие		+	+	+	+	+	+
Безопасность		+	+	-	+	+	+
Общая карта		-	-	-	-	-	-
Создание мероприятий		-	+	+	-	+	+
Адаптивный интерфейс		+	-	+	-	+	+
Всероссийская платформа		-	-	+	+	+	+
Удобство поиска мероприятий		+	+	-	+	+	+

Из данной таблицы видно, что большинство аналогов имеют необходимые для нашего приложения признаки, такие, как мобильная версия приложения, личный кабинет, запись на мероприятие, безопасность и удобство поиска мероприятий. Так же из таблицы видно, что все платформы имеют возможность записи на мероприятие.

Но ни один рассматриваемый нами аналог не имеет общей карты, что является большим минусом. Наиболее близким к нашим требованиям аналогом является ВК сообщества. Однако ни один исследуемый нами аналог не соответствует на 100% к требованиям проектируемой нами системы.

1.4 Функциональные требования

Требования к системе:

- регистрация на сайте;
- разделение на акторы: модератор, гость, пользователь;
- аутентификация пользователя по паролю;
- поиск мероприятий с помощью фильтров;
- чат между участниками одного мероприятия;
- запись на мероприятие;
- управление личным кабинетом;
- создание своего мероприятия;
- указание мероприятий на карте.

1.5 Вывод по главе

По результатам анализа задания сформулированы требования к разрабатываемому сервису.

Выполнен поиск и обзор аналогов, приближенных по функционалу.

Рассмотренные аналоги имеют ряд существенных минусов:

- сложность поиска мероприятий и событий;
- отсутствие возможности создания неофициальных мероприятий;

- отсутствие карты с ближайшими к пользователю событиями.

А также они не удовлетворяют главной идеи разрабатываемого сервиса, а именно – создание и проведение любительских тематических встреч и мероприятий с удобным поиском событий и встреч по всему городу. Это дает большое преимущество на рынке социальных сетей.

2 Проектирование системы

2.1 Общая структура разрабатываемой системы

Общая структура разрабатываемой социальной сети представляет собой клиент-сервер архитектуру (Рисунок 7). Была выбрана трехуровневая архитектура, состоящая из клиента, сервера и базы данных.

Архитектура клиент-сервер подразумевает собой разделение отправки запросов и предоставления результатов пользователю на разных компьютерах, для работы независимо друг от друга. При такой архитектуре запросы с неограниченного множества клиентов передаются специально подготовленным служебным машинам – серверам, в разрабатываемой системе будет использоваться один сервер.

Преимуществом такой архитектуры является:

- централизованное управление данными через один сервер;
- информационная безопасность;
- повышение производительности работы с общими ресурсами;
- масштабируемость, свободное увеличение количества клиентов и серверов.

Недостатки архитектуры клиент-сервер:

- опасность отказа единственного сервера;
- перегрузка сервера при большом количестве клиентов.



Рисунок 7 – Общая структура системы

2.1 Выбор средств и технологий разработки

2.1.1 Клиентская часть

Для создания клиентской части были выбраны наиболее подходящие языки программирования и фреймворки, отличающиеся своей популярностью и большим функционалом и, в некоторых случаях, русской документацией.

JavaScript:

JavaScript [9] – это основная технология для клиент-серверной разработки. JavaScript поддерживает объектно-ориентированный, императивный и декларативный (например, функциональное программирование) стили программирования. Подходит как для клиента, так и для сервера.

Vue.js:

Vue [10] – это прогрессивный фреймворк для создания пользовательских интерфейсов. Содержит в себе множество функций упрощающих процесс программирования, за что и был выбран для данного проекта.

HTML:

HTML [12] – язык разметки гипертекста. HTML- платформа, для создания веб-приложений, использующих аудио, видео, графику, анимацию, геолокацию.

SASS (SCSS):

SASS [13] – это такой язык, код которого специальной программой транслируется в обычный CSS код. Синтаксис этого языка очень гибок, он учитывает множество мелочей, которые так желанны в CSS.

2.1.2 Серверная часть

Node.js:

Node.js [14] – значительно расширяет возможности JavaScript. Это среда выполнения кода JavaScript вне браузера, которая позволяет писать серверный

код для веб-страниц и веб-приложений, за что Node.js и был выбран для данного проекта.

Nuxt.js:

Nuxt.js [15]— это фреймворк для создания универсальных приложений на Vue.js с использованием Node.js. Главный плюс технологии – это серверный рендеринг.

2.1.3 Формат запроса данных у сервера

HTTP расшифровывается как Hyper Text Transfer Protocol, что в переводе означает «протокол передачи гипертекста». HTTP является протоколом прикладного уровня для обмена информацией между клиентом и сервером путем передачи сообщений.

Виды http сообщений:

- запрос: сообщения, которые отправляет клиент серверу, чтобы получить определенную информацию или выполнить заданное действие.
- ответ: сообщения, которые отправляет сервер клиенту в ответ на его запрос.

Для более точного описания запросов существуют методы, они позволяют указать конкретное действие, которое необходимо выполнить серверу.

2.1.4 База данных

MongoDB:

MongoDB [16] – документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Была выбрана из-за:

- MongoDB используется для хранения данных, не связанных жестко между собой;

- выдача прав разным категориям пользователя;
- не нужен язык SQL;
- возможность создавать коллекции с документами, различающейся структурой;
- легкая масштабируемость;

2.1.5 Методология проектирования ICONIX

ICONIX [17] – это методология разработки и проектирования ПО, в котором используются четыре основных UML диаграммы, это: модель прецедентов (диаграмма использования), диаграмма последовательности, диаграмма пригодности, диаграмма классов. Диаграммы в свою очередь разбиваются на две группы: динамическую модель и статическую (Рисунок 8). Для данного проекта будет достаточно построения диаграммы прецедентов, модели предметной области и диаграммы классов.

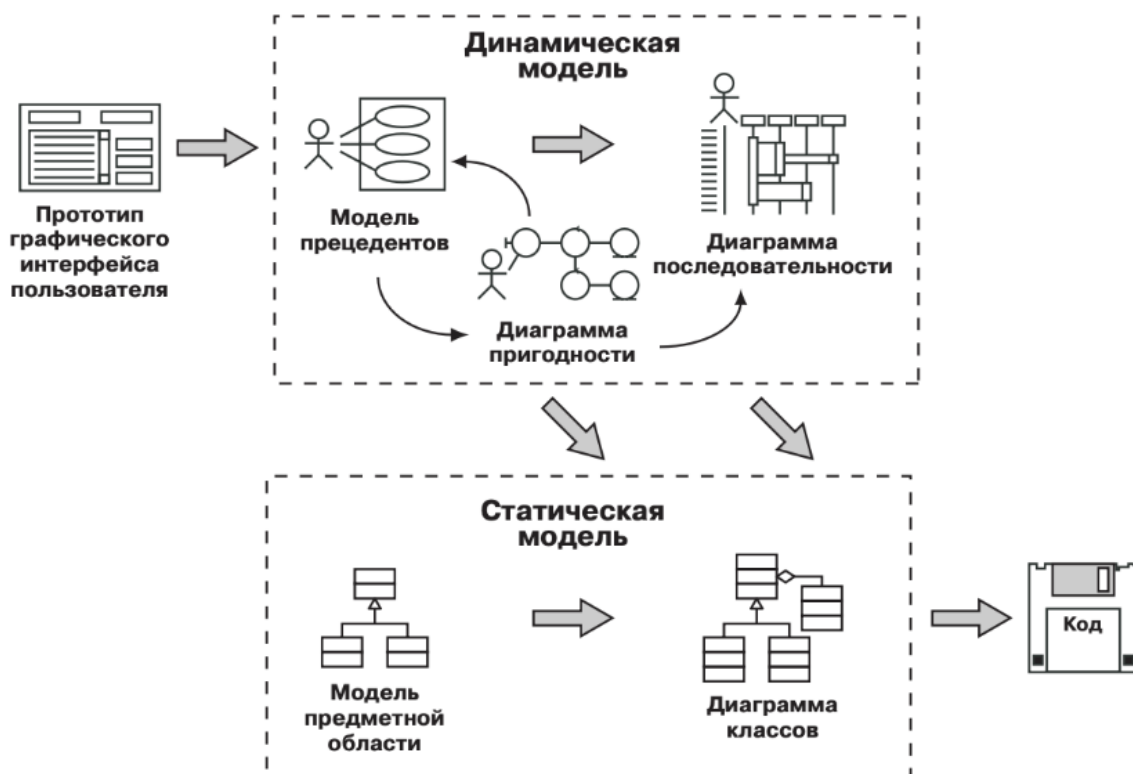


Рисунок 8 – Проектирование ICONIX

2.2 Диаграммы прецедентов

На основании анализа функционального требования к системе, были выделены следующие акторы:

- гость;
- пользователь;
- модератор.

Диаграммы прецедентов показывают, какие функции доступны каждому из акторов.

2.2.1 Диаграммы прецедентов актора «гость»

Описание для диаграммы прецедентов актора «гость» (Рисунок 9).

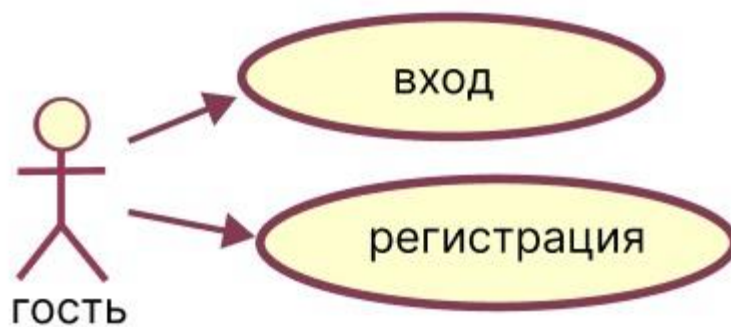


Рисунок 9 – Диаграмма прецедентов для гостя

Название прецедента: войти в программу.

Цель сценария: войти в программу.

Предусловия: открыто окно вебсайта.

Основной сценарий:

1. пользователь просматривает страницу;
2. нажать кнопку «Войти»;
3. открывается окно входа с полями для ввода логина и пароля и кнопками «войти» и «зарегистрироваться»;
4. пользователь вводит данные и нажимает кнопку войти.

Альтернативная последовательность:

1. пользователь просматривает страницу;
нажать кнопку «Войти»;
2. открывается окно входа с полями для ввода логина и пароля и кнопками «войти» и «регистрация»;
3. нажата кнопка «регистрация»;
4. переход на страницу регистрации.

Постусловия:

Выполнен вход в систему.

Название прецедента: Регистрация на сайте

Цель сценария: добавление нового пользователя.

Предусловия: открыто окно регистрации.

Основной сценарий:

1. пользователь вводит личную информацию;
2. нажата кнопка зарегистрироваться;
3. альтернативный сценарий;
4. возврат на главную страницу.

Постусловия:

В базу данных вносится новый пользователь сервиса.

2.2.2 Диаграммы прецедентов актора «пользователь»

У пользователя появляются новые прецеденты. Описание для диаграммы прецедентов актора «пользователь» (Рисунок 10):

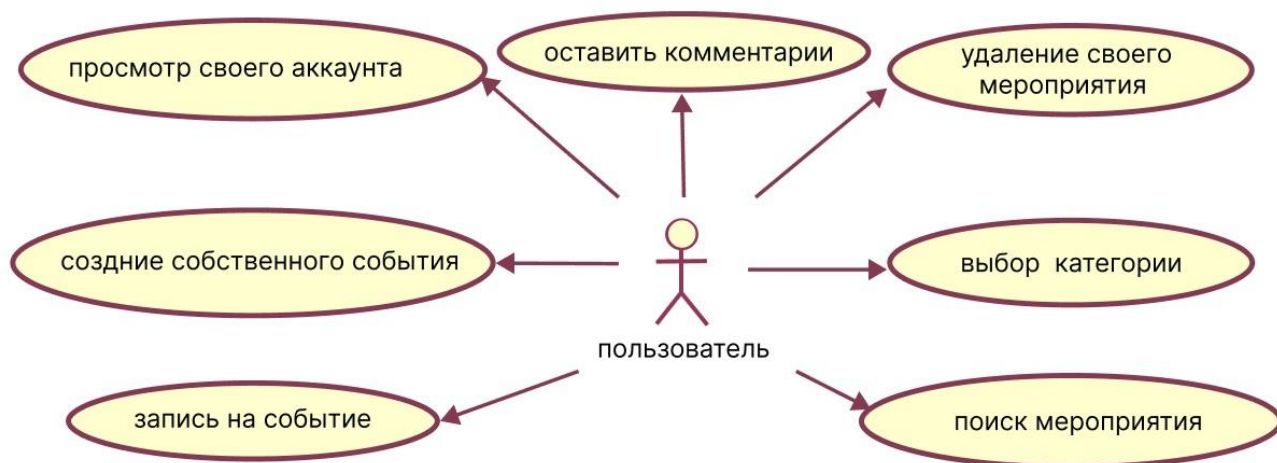


Рисунок 10 – Диаграмма прецедентов для пользователя

Название прецедента: Выбор категории.

Цель сценария: уточнение поиска мероприятия.

Предусловия: открыто главное окно сайта.

Основной сценарий:

1. нажата кнопка с названием категории;
2. переход на страницу с мероприятиями;
3. указаны фильтры при необходимости.

Постусловия: идет фильтрация мероприятий по принципу, выбранному пользователем.

Название прецедента: Поиск мероприятия.

Цель сценария: знакомство с мероприятиями на карте.

Предусловия: открыта главная страница.

1. основной сценарий;
2. нажата кнопка «поиск»;
3. переход на страницу с картой и метками.

Постусловия: пользователь знакомится с картой и отмеченными на ней событиями.

Название прецедента: удаление своего мероприятия.

Цель сценария: удаление мероприятия с сайта и базы данных.

Предусловия: мероприятие создано данным пользователем.

Основной сценарий:

1. пользователь просматривает свое мероприятие и принимает решение о его удалении;
2. нажата кнопка «удалить».

Постусловия: в базе данных и у пользователей удаляется данное мероприятие.

Название прецедента: создание собственного события.

Цель сценария: добавление события на сервис.

Предусловия: выполнен вход в систему.

Основной сценарий:

1. нажата кнопка «добавить мероприятие»;
2. переход в окно для создания мероприятий;
3. пользователь вводит информацию, заполняет поля;
4. нажата кнопка «создать».

Постусловия: В базу данных отправляется новое мероприятие с запросом для модератора на публикацию.

Название прецедента: запись на событие.

Цель сценария: подтверждение участия в мероприятии.

Предусловия: выполнен вход в систему.

Основной сценарий:

1. пользователь находит интересующее его мероприятие;
2. нажата кнопка «участвовать».

Постусловия: в базу данных отправляется новая связь пользователя с данным мероприятием.

Название прецедента: оставить комментарии.

Цель сценария: общение между пользователями сервиса.

Предусловия: открыта страница мероприятия.

Основной сценарий:

1. пользователь пишет комментарий в поле под описанием мероприятия;
2. нажата кнопка «отправить»;
3. комментарий высвечивается среди других комментариев.

Постусловия: на сайте отображается комментарий пользователя.

Название прецедента: просмотр своего аккаунта.

Цель сценария: просмотр своего аккаунта.

Предусловия: открыта главная страница сайта.

Основной сценарий:

1. нажата кнопка «мой профиль»;
2. переход на страницу профиля;
3. редактирование информации о пользователе.

Постусловия: в базе данных меняется информация о пользователе.

2.2.3 Диаграммы прецедентов актора «модератор»



Рисунок 11 – Диаграмма прецедентов для модератора

Новые прецеденты для актора «модератор» (Рисунок 11):

Название прецедента: блокировка пользователя.

Цель сценария: блокировка недоброжелательных пользователей

Предусловия: открыт аккаунт выбранного пользователя.

Основной сценарий:

1. модератор просматривает информацию о пользователе;
2. нажата кнопка «временно заблокировать».

Постусловия: в базе данных пользователь переносится в архив.

Название прецедента: удаление мероприятия.

Цель сценария: удаление конкретного мероприятия.

Предусловия: открыта страница мероприятия.

Основной сценарий:

1. модератор просматривает информацию о мероприятии;
2. нажата кнопка «удалить».

Постусловия: в базе данных и у клиентов удаляется данное мероприятие.

2.3 Проектирование интерфейса системы

2.3.1 Модель предметной области

Модель предметной области (Рисунок 12) описывает сущности на данном этапе проектирования и очевидные отношения включения. В качестве сущностей выступают окна экрана, включающие элементы на экране.



Рисунок 12 – Модель предметной области

2.3.2 Разработка диаграммы классов

Диаграмма классов (Рисунок 13) показывает структуру программы и связь между элементами. В данном случае в качестве элементов выступают классы. На диаграмме видны переменные и функции, которыми пользуется класс.

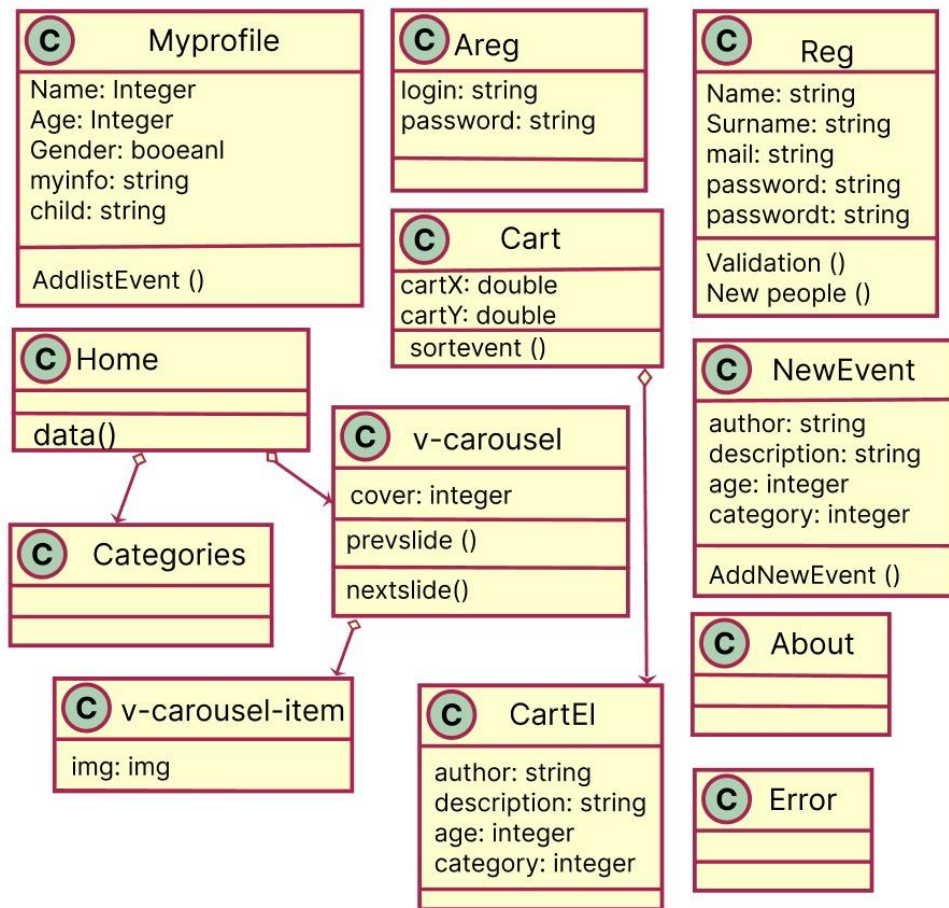


Рисунок 13 – Диаграмма классов

Между классами используется отношение агрегации (один класс включает в себя другой)

2.3.3 Диаграмма потока экранов

Диаграмма потока экранов, представленная в приложении Б, показывает, как перейти от одного экрана к другому, а также, какие функции присутствуют на каждом экране.

Для наиболее удобного использования сайтом, переход от начального экрана до другого экрана не должен превышать 4 переходов.

2.4 Проектирование базы данных

2.4.1 Логическая модель данных предметной области

Логическая модель данных (Рисунок 14)– это набор сущностей и связей. Так как в данной работе используется документноориентированная база данных, то в качестве данных представлены 2 сущности: мероприятия и пользователи.

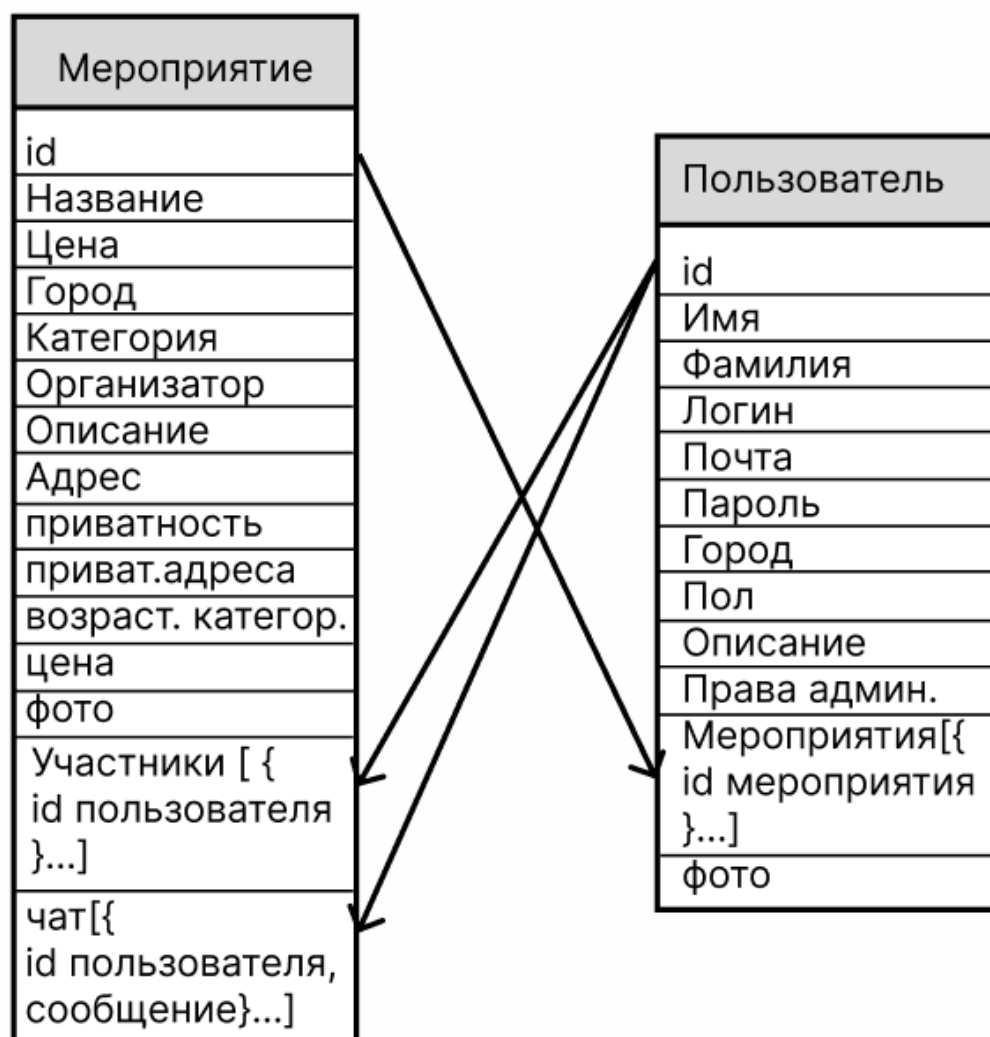


Рисунок 14 – Логическая модель базы данных

2.5 Вывод по главе

В соответствии со спецификацией требований:

- построены диаграммы прецедентов и потоков экранов;
- построена диаграмма классов;
- разработана макеты интерфейса;
- построена логическая модель данных предметной области.

3 Реализация

3.1 Реализация клиентской части

Разработка визуального интерфейса была проведена в приложении «Figma», ссылка на дизайн интерфейса приведена в использованных источниках [18].

Реализация интерфейса производится с помощью популярного фреймворка «Vue.js 3». Разработка основывается на формировании структуры приложения на языке HTML с помощью методологии БЭМ [19].

БЭМ (Блок, Элемент, Модификатор) – компонентный подход к веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки. Он позволяет легко и быстро разрабатывать интерфейсы любой сложности и повторно использовать существующий код, избегая повторения кода в программе.

Для уменьшения временных затрат

Перемещение между страницами осуществляется с помощью библиотеки «vue router»[20].

Компоненты переменной routes:

```
const routes = {  
  '/': HomeComponent,  
  '/about': AboutComponent,  
  '/page: PageComponent,  
  '/profil: ProfilComponent  
}
```

В клиентской части были реализованы следующие компоненты:

- About-comp – окно о компании;
- Areg-comp – окно регистрации;
- Card-comp – окно поиска мероприятий;
- CardEl-comp – окно мероприятия;

- error-comp – окно ошибки;
- Home-comp – домашнее окно;
- Myprofile-comp – окно личного кабинета;
- Reg-comp – окно входа;
- v-carousel – структура карусели (слайдер фотографий);
- v-carousel-item – элементы карусели.

Разработка приложения велась с использованием системы контроля версий «Git», а именно «GitHub». Ссылка на репозиторий добавлена в список использованных источников [21].

3.2 Описание HTTP методов

Методы, используемые в сервисе:

- Get: Позволяет запросить некоторый конкретный ресурс,
- Post: Позволяет отправить данные на сервер

Таблица 2 – описание используемых методов

Окно	Метод	Данные	Описание
Главное окно	Get	Данные о мероприятии: Фотография, название	Топ 5 мероприятий
Окно входа	Post	Логин, пароль	Отправка данных на сервер
	Get	Ответ сервера	Данные прошли проверку, получен ответ об ошибке входа либо входе
Окно регистрации	Post	Данные о пользователе: Город, пол, логин, ФИО, почта, пароль	Создание нового пользователя
Окно выбора мероприятия	Get	Данные о мероприятии: Фотография, название, цена, категория, время, место	Список доступных мероприятий
Окно мероприятия	Get	Все данные о мероприятии, чат	Отображение данных
	Post	Запрос на участие в мероприятии	Добавление в базу данных нового участника мероприятия
Личный кабинет	Get	Информация о пользователе: ФИО, логин, город, возраст, пол,	Отображение личного кабинета

		описание, мероприятия пользователя	
	Post	Редактирование данных	Перезапись информации в базе данных
Окно создания мероприятия	Post	Информация о мероприятии: дата, время, возрастная категория, направление, описание, адрес, организатор	Добавление мероприятия в базу данных
Окно редактирования личного кабинета	Get	Информация о пользователе	Отображение старой информации
	Post	Измененная информация о пользователе	Перезапись информации

3.3 Формат обмена данными

JSON [8] – это текстовый формат обмена данными. В данной работе используется для обмена данными между браузером(клиентом) и сервером. Этот формат отлично подходит для передачи паролей, логинов и другой информации пользователя, и дальнейшей записи её в базу данных.

Именно в этом формате будут передаваться данные о пользователях в виде

```

“Users”: [
{
“id”: “значение”,
“name”: “значение”,
“surname”: “значение”,
“login”: “значение”,
“mail”: “значение”,
“password”: “значение”,
“city”: “значение”,
“gender”: “значение”,
“description”: “значение”,
“admrighths”: “значение”
}

```

А также информация о мероприятиях:

```
“Meeting”: [{  
  “id”: “значение”,  
  “city”: “значение”,  
  “category”: “значение”,  
  “organizer”: “значение”,  
  “members”: “значение”,  
  “description”: “значение”,  
  “address”: “значение”  
}]
```

3.4 Яндекс карта

Благодаря API Яндекс карты была реализована карта с метками (Рисунок 15).

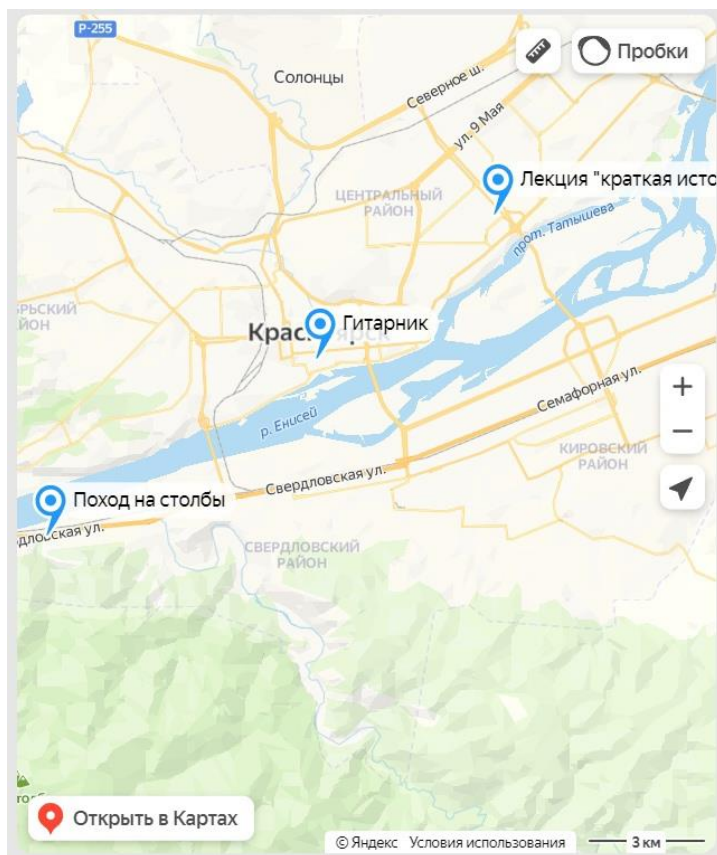


Рисунок 15 – Реализованная карта

Подключение карты с помощью тега Script:

```
<script src="https://api-maps.yandex.ru/2.1/?apikey= API-ключ&lang=ru_RU" type="text/javascript">
```

3.5 Реализация серверной части

Для создания серверной части используется комбинация Node.js + Express + Nuxt.js. Express – это подходящий для данной разработки фреймворк для Node.js, его плюсами является:

- схожий синтаксис;
- простота, возможность быстрой интеграции;
- автоматические обработки кейсов кодировок и чтение файлов.

A Nuxt.js позволяет быстро развернуть клиент-серверное приложение. Также, используется утилита командной строки – Nodemon, для автоматического обновления сервера в процессе реализации.

Компоненты серверной части:

- registration- отвечает за добавление новых пользователей в базу данных и валидацию вводимых данных;
- authorization – отвечает за сверку вводимых данных с данными в базе данных, выдает ответ о ошибке, если данные не совпадают, или проводит авторизацию, если данные совпали;
- drawdata – отрисовка данных о мероприятиях на главной странице;
- drawactivities – отрисовка мероприятий на окне выбора мероприятий;
- activities – отрисовка информации о мероприятии, запрос на участие в мероприятии;
- account – отображение и изменение информации о пользователе.

3.6 Реализация базы данных

База данных была разработана в MongoDB Compass – визуальное представление базы данных. Обращение к базе данных идет по адресу localhost:27017. Создана база «DataDip» и две коллекции, «Events» с

информацией о мероприятиях и «Users» с информацией о пользователях (Рисунок 16).

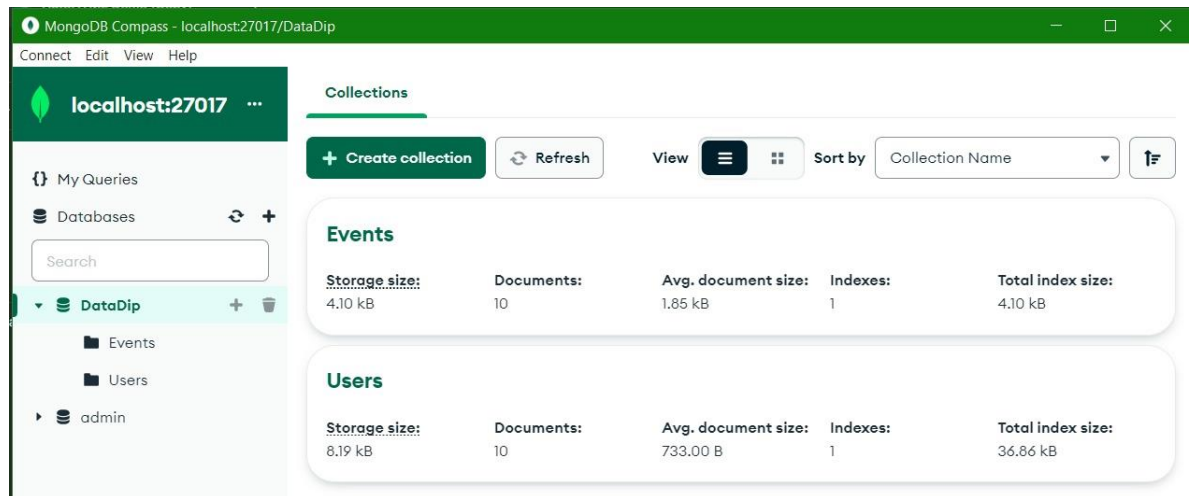


Рисунок 16 – Структура базы данных

Пример структуры элемента «мероприятие» в коллекции Events (Рисунок 17).

```
_id: "6493ca2004a7a63b64b488e7"
Name: "Гитарник"
City: "Красноярск"
Category: "музыка"
Organazer: "6493ca20ade31d143badfe6f"
privacy: "no"
agecategory: "18"
addressprivacy: "no"
price: "0"
photo: "git.png"
Description: "Non ea voluptate aliqua exercitation in sunt irure Lorem. Est amet dui..."
address: "510 Linden Boulevard, Disautel, Maine, 3302"
Users: Array
  0: Object
    User: "6493ca20ade31d143badfe6f"
  1: Object
  2: Object
Chat: Array
  0: Object
    User: "6493ca2044245cbd8d459a9d"
    message: "Sit reprehenderit ad veniam ullamco elit. Laborum magna labore laboris..."
  1: Object
  2: Object
```

Рисунок 17 – Пример структуры элемента коллекции Events

Пример структуры элемента «пользователь» в коллекции Users (Рисунок 18).

```
_id: "6493c114950828a22d935194"
Name: "Mcguire"
Surname: "Clarke"
Login: "gser"
Mail: "mcguireclarke@zior.com"
Password: "gsgr43gg"
City: "Красноярск"
Gender: "male"
Description: "Et tempor aliqua in enim proident esse id. Non ipsum enim enim ullamco..."
Adminrights: "no"
age: 31
photo: "my.png"
▼ events: Array
  ▣ 0: Object
    event: "6493c11412623c7271622d4e"
  ▸ 1: Object
  ▸ 2: Object
```

Рисунок 18 – Пример структуры элемента коллекции Users

3.7 Инструкция пользователя

3.7.1 Основные функции

Чтобы впервые воспользоваться сервисом, гостю нужно зайти на главное окно сайта (Рисунок 19). На главном окне гость видит такие элементы как:

- кнопка «войти»;
- кнопка «о компании»;
- кнопка «зарегистрироваться»;
- слайдер с информацией о сайте;
- слайдер с топ 5 популярных мероприятий.

После просмотра гостем информации на страницы, чтобы продолжить нужно нажать кнопку «зарегистрироваться».



Рисунок 19 – Главное окно

После нажатия на кнопку «Зарегистрироваться» гость попадает в окно регистрации (Рисунок 20).

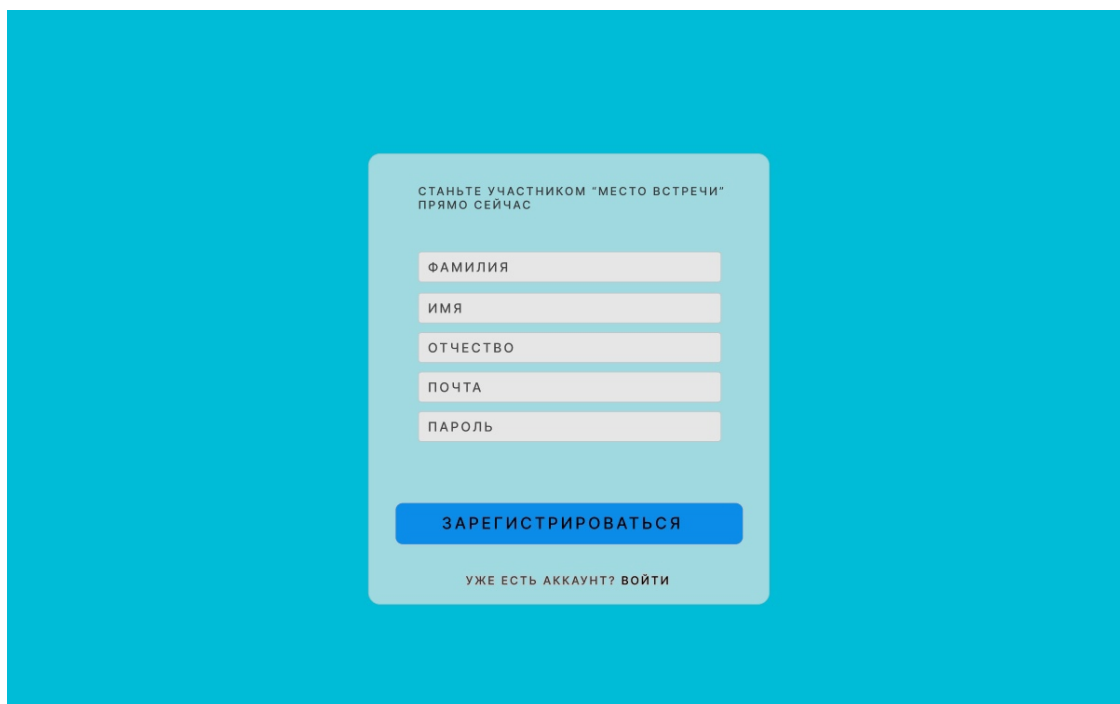
A registration form window with a light blue background. At the top, it says "СТАНЬТЕ УЧАСТНИКОМ 'МЕСТО ВСТРЕЧИ' ПРЯМО СЕЙЧАС". Below this are five input fields: "ФАМИЛИЯ", "ИМЯ", "ОТЧЕСТВО", "ПОЧТА", and "ПАРОЛЬ". At the bottom, there is a blue button labeled "ЗАРЕГИСТРИРОВАТЬСЯ" and a link that says "УЖЕ ЕСТЬ АККАУНТ? ВОЙТИ".

Рисунок 20 – Окно регистрации

В данном окне гостю требуется ввести свои данные и нажать кнопку «Зарегистрироваться», после чего он попадает на окно входа (Рисунок 21).

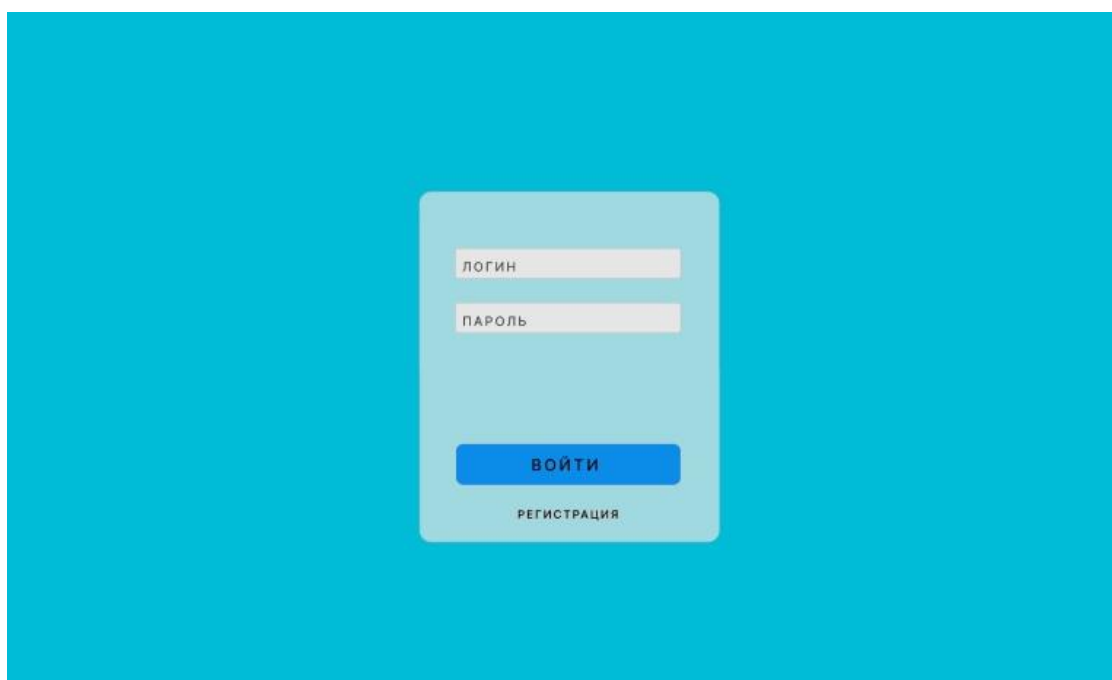
A login form window with a light blue background. It has two input fields: "ЛОГИН" and "ПАРОЛЬ". Below these is a blue button labeled "ВОЙТИ". At the bottom, there is a link that says "РЕГИСТРАЦИЯ".

Рисунок 21 – Окно входа

После регистрации, гость получает статус пользователя. Пользователь вводит данные, нажимает кнопку «войти». После чего попадает в окно выбора мероприятий (Рисунок 22).

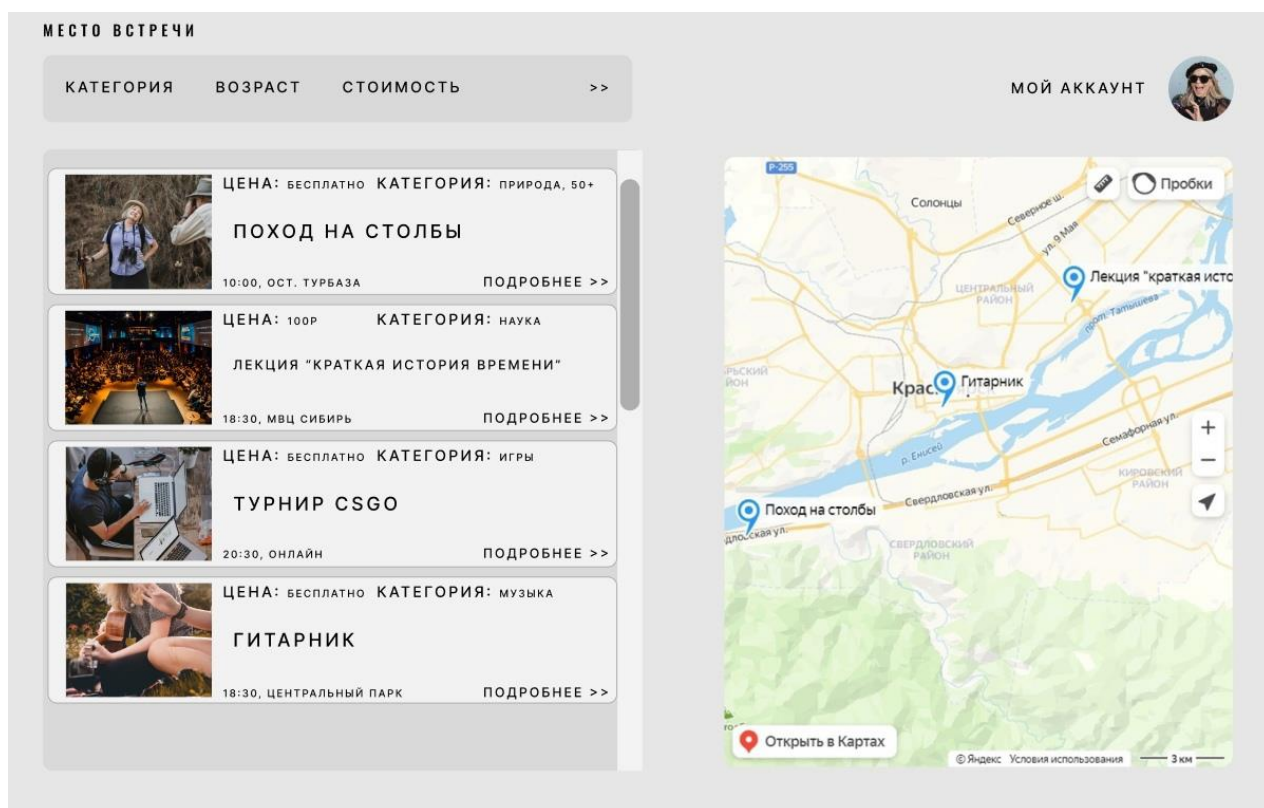


Рисунок 22 – Окно выбора мероприятий

На данном окне пользователь видит данные элементы:

- кнопка «место встречи», переносящая на главное окно;
- кнопка «мой аккаунт», переносит в окно мой профиль;
- карта с метками;
- фильтр выбора мероприятий;
- список мероприятий;

Для участия в мероприятии, пользователю нужно выбрать из списка мероприятие и нажать на него, затем откроется окно выбранного мероприятия (Рисунок 23).

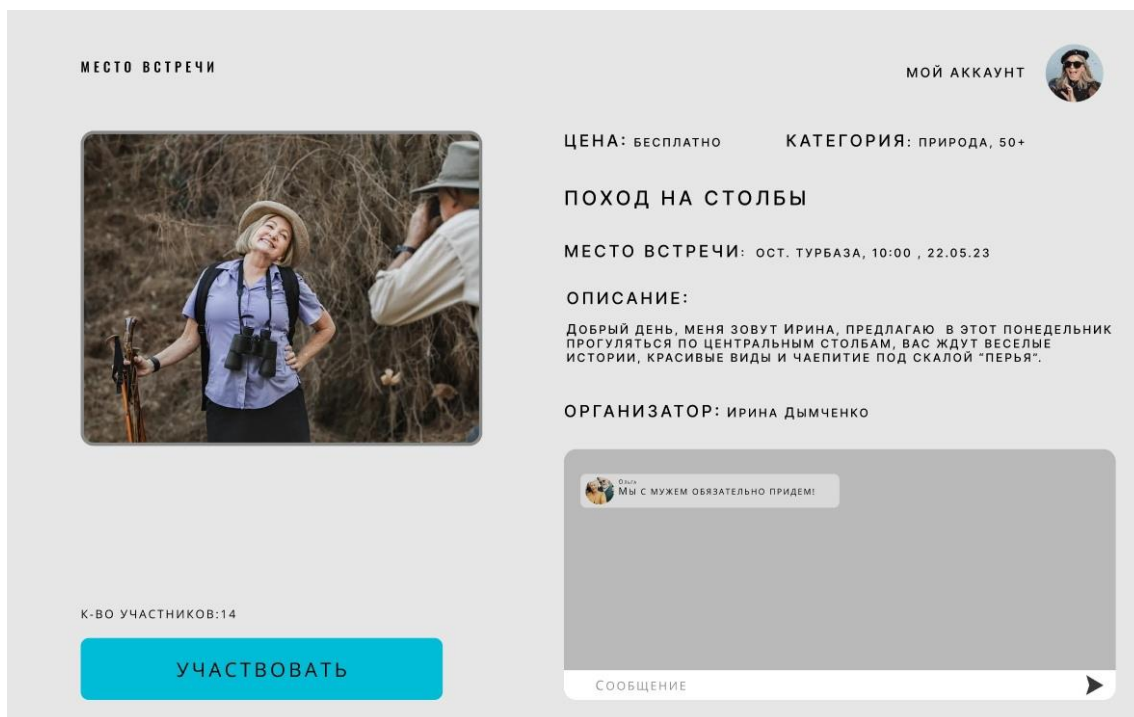


Рисунок 23 – Окно выбранного мероприятия

В данном окне пользователь видит следующие элементы:

- фотография, добавленная создателем мероприятия;
- описание мероприятия (цена, категория, название, текстовое описание, имя организатора);
- чат с участниками мероприятия.

После знакомства с информацией, пользователь нажимает на кнопку «Участвовать», и попадает в список участников, став участником мероприятия, ему открывается возможность писать в чате.

Для того чтобы создать свое собственное мероприятие, пользователю нужно нажать на кнопку «мой аккаунт», после чего будет выполнен переход в окно личного аккаунта (Рисунок 24).

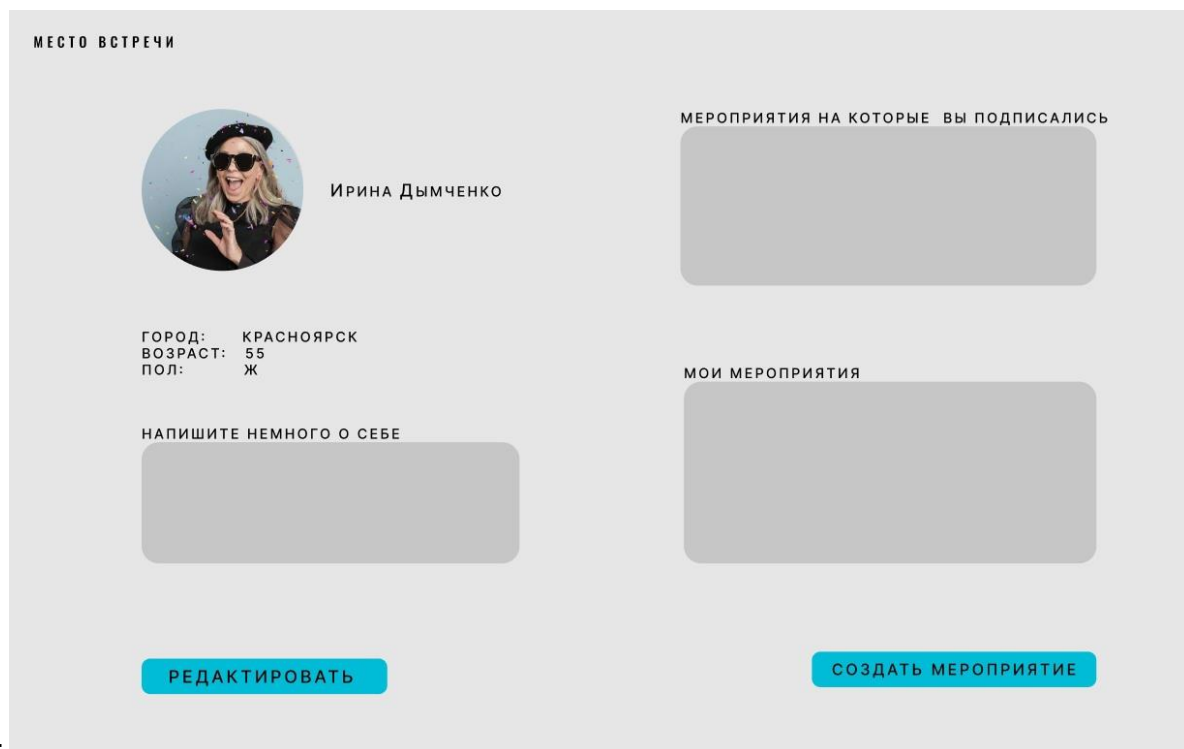


Рисунок 24 – Окно мой аккаунт

На данном окне пользователь видит следующие элементы:

- фотография, добавленная пользователем;
- информация о пользователе;
- список мероприятий, на которые пользователь подписан;
- кнопка «редактировать»;
- кнопка «создать мероприятие»;
- список мероприятий, которые пользователь создал.

Пользователь нажимает кнопку «Создать мероприятие» и переходит в окно создания мероприятий (Рисунок 25).

МЕСТО ВСТРЕЧИ

ФОТО

НАЗВАНИЕ:

КАТЕГОРИЯ: v

ПРИВАТНОСТЬ: v

АДРЕС:

ЦЕНА:

ПРИВАТНОСТЬ АДРЕСА: v

ВОЗРАСТНАЯ КАТЕГОРИЯ: v

ОПИСАНИЕ:

РАССКАЖИТЕ, ЧЕМ ВЫ ПЛАНИРУЕТЕ ЗАНИМАТЬСЯ НА ДАННОМ МЕРОПРИЯТИИ

СОЗДАТЬ

Рисунок 25 – Окно создания мероприятия

Для создания мероприятия пользователю необходимо заполнить все окна и нажать кнопку «создать». После данного действия мероприятие попадает в список всех мероприятий, а пользователь переносится обратно на окно «мой профиль».

3.7.2 Дополнительные функции

Из окна «Мой аккаунт» можно перейти в окно редактирования личной информации (Рисунок 26). После редактирования, пользователь нажимает кнопку сохранить и переходит обратно на окно «мой аккаунт».

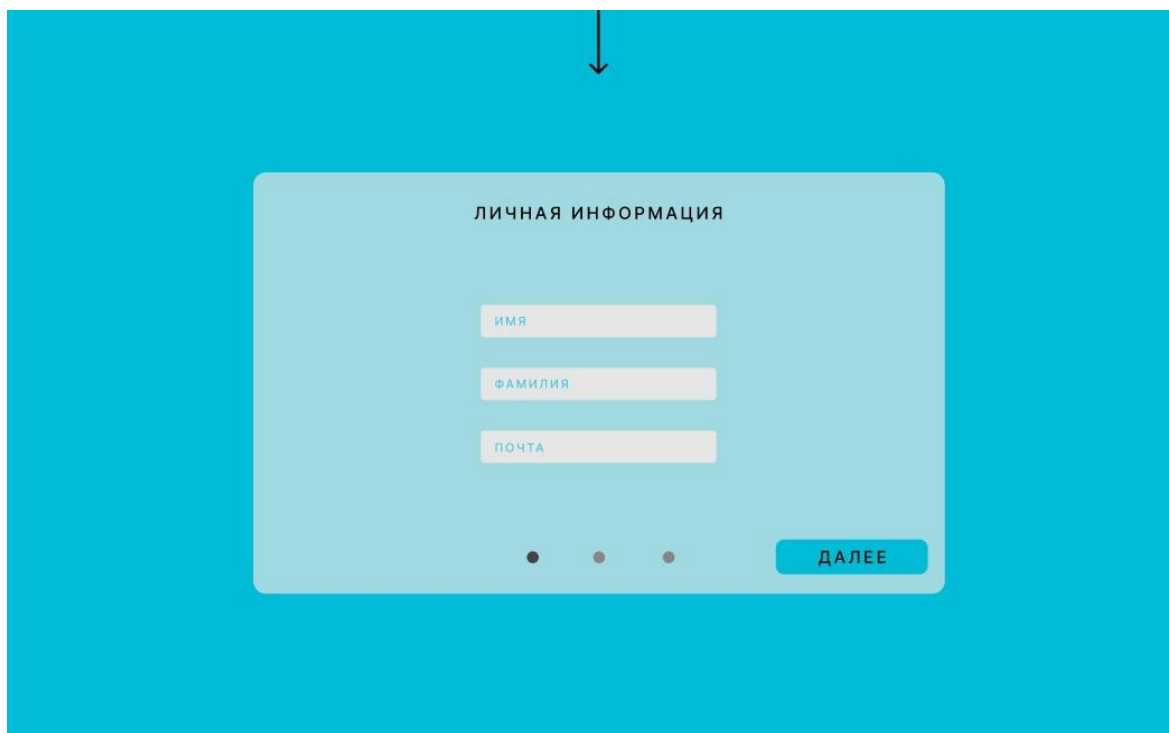


Рисунок 26 – Окно редактирования личной информации

На главном окне пользователю доступна кнопка «о компании», при нажатии на которую пользователь переходит в окно с описанием проекта (Рисунок 27).



Рисунок 27 – Окно о компании

При переходе пользователем на не существующую страницу, выдается окно об ошибке (Рисунок 28).



Рисунок 28 – Окно сообщения об ошибке пользователя

В данном окне пользователь может нажать на кнопку «Домой» для перехода на главную страницу сервиса.

3.8 Вывод по главе

Была разработана структура приложения, соответствующая сформированным требованиям и функционалу, а также соответствующая программной платформе реализации, а также выбраны методы и средства разработки требуемого функционала. В результате было разработано веб-приложение, соответствующее необходимому функционалу и реализованы необходимые модули.

ЗАКЛЮЧЕНИЕ

Интернет приобрел большое значение в современном обществе. На сегодняшний день проблема виртуального общения стала очень актуальна, так как общение через Интернет стало замещать реальное. Для решения этой проблемы нами была разработана социальная сеть, позволяющая самостоятельно зарегистрировать и провести то или иное мероприятие, найти людей с одинаковыми интересами и расширить круг общения. Для достижения нашей цели мы провели разбор аналогов разрабатываемого нами приложения, выделили критерии для их оценки, выделили основные плюсы и минусы функционала. Таким образом, самым близким к разрабатываемой нами социальной сети аналогом является ВК сообщества. Функционал групп в социальной сети «ВКонтакте» отвечает требованиям проектируемого нами приложения, но имеет ряд существенных минусов, из-за которых данный аналог не может полностью заменить разрабатываемую систему.

В результате проделанной работы было выполнено:

- разбор аналогов разрабатываемого приложения;
- выделение требований и задач;
- проектирование, путем выделения прецедентов;
- проектирование базы данных;
- выбор технологий реализации на основе требований к веб-сервису;
- создание дизайна и интерфейса веб-сервиса;
- была реализована верстка и логика модулей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 09.01.2014. – Красноярск : СФУ, 2014. – 60 с.
2. kuda Moscow [Электронный ресурс] // kudamoscow. – Режим доступа: <https://kudamoscow.ru/event/fest/> (Дата обращения: 19.03.2023).
3. kudago [Электронный ресурс] // kudago. – Режим доступа: <https://kudago.com/msk/activity/extraordinary> (Дата обращения: 19.03.2023).
4. ВКонтакте [Электронный ресурс] // vk. – Режим доступа: <https://vk.com/groups> (Дата обращения: 19.03.2023).
5. Яндекс афиша [Электронный ресурс] // yandex. – Режим доступа: <https://afisha.yandex.ru/krasnoyarsk> (Дата обращения: 19.03.2023).
6. Планета семинаров [Электронный ресурс] // planetaseminarov. – Режим доступа: <https://planetaseminarov.com> (Дата обращения: 19.03.2023).
7. Timerpad [Электронный ресурс] // Timerpad. – Режим доступа: <https://afisha.timerpad.ru> (Дата обращения: 19.03.2023).
8. JSON [Электронный ресурс] // Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/JSON>
9. Современный учебник JavaScript [Электронный ресурс] // lavascript. – Режим доступа: <https://learn.javascript.ru/> (Дата обращения: 13.01.2023).
10. Документация Vuejs [Электронный ресурс] // Vuejs. – Режим доступа: <https://ru.vuejs.org/v2/guide/> (Дата обращения: 13.01.2023).
11. Документация Vue 3 [Электронный ресурс] // V3. – Режим доступа: <https://v3.ru.vuejs.org/> (Дата обращения: 13.01.2023).
12. HTML Documentation [Электронный ресурс] // HTML. – Режим доступа: <https://html.spec.whatwg.org/multipage/> (Дата обращения: 13.01.2023).
13. SASS Documentation [Электронный ресурс] // sass-scss. – Режим доступа: <https://sass-scss.ru/documentation/> (Дата обращения: 13.01.2023).
14. Node Documentation [Электронный ресурс] // nodejs. – Режим доступа:

<https://nodejs.org/en/docs> (Дата обращения: 13.01.2023).

15. Nuxt Documentation [Электронный ресурс] //nuxt – Режим доступа: <https://nuxt.com/> (Дата обращения: 13.02.2023).

16. MongoDB Documentation [Электронный ресурс] // mongodb. – Режим доступа: <https://www.mongodb.com/> (Дата обращения: 13.01.2023).

17. ICONIX [Электронный ресурс] // Википедия. – Режим доступа: <https://en.wikipedia.org/wiki/ICONIX>

18. Дизайн сайта «Место встречи» [Электронный ресурс] // Figma. – Режим доступа: <https://www.figma.com/file/1L5LBPL7KeVAFAfgFnCaOA/Untitled?node-id=0-1&t=6hgPkb3EeWG93A3P-0> (Дата обращения: 25.03.2023).

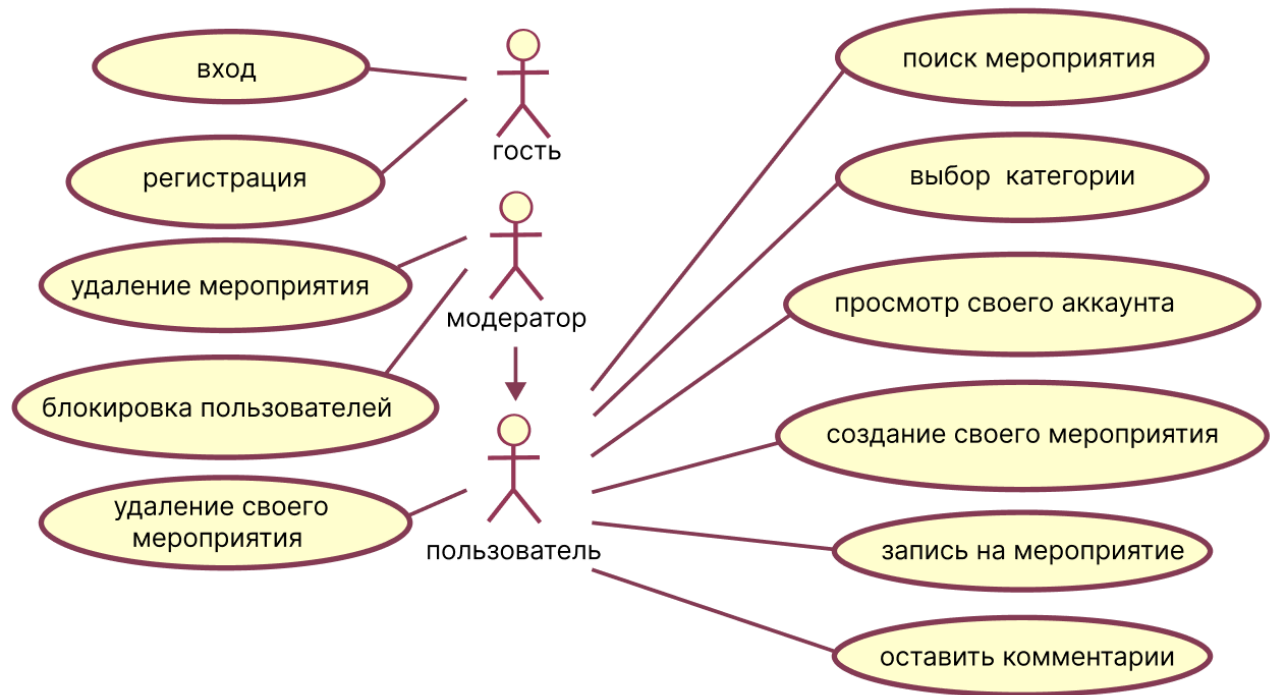
19. Быстрый старт методология БЭМ [Электронный ресурс] // bem. – Режим доступа: <https://ru.bem.info/methodology/quick-start/>

20. Документация Vuerouter [Электронный ресурс] // Vuerouter. – Режим доступа: <https://router.vuejs.org/guide/>

21. Meeting-places [Электронный ресурс] // GitHub. – Режим доступа: <https://github.com/sunmantra/meeting-places>

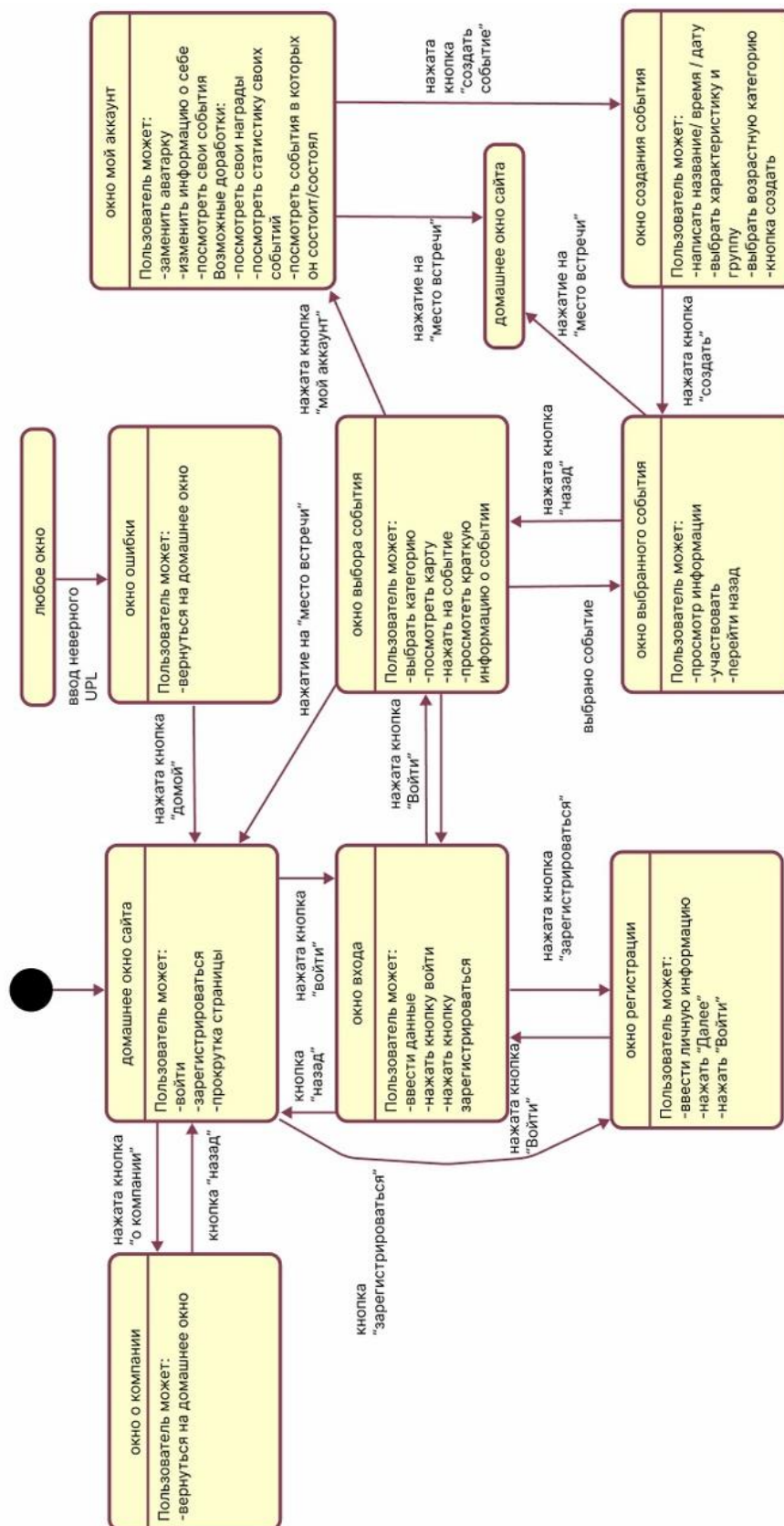
ПРИЛОЖЕНИЕ А

Общая диаграмма прецедентов



ПРИЛОЖЕНИЕ Б

Диаграмма потока экранов



Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

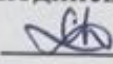
 О. В. Непомнящий
подпись инициалы, фамилия
« 21 » 06 2023 г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Онлайн-сервис для планирования мероприятий и участия в них
тема

Руководитель Н.С. - 24.06.2023 доцент,
подпись, дата канд. техн. наук
должность, ученая степень

Выпускник 
подпись, дата

Нормконтролер Н.С. - 24.06.2023 канд. техн. наук
подпись, дата должность, ученая степень

Н.Ю. Сиротинина
инициалы фамилия
Д.А. Черных
инициалы, фамилия
Н.Ю. Сиротинина
инициалы, фамилия

Красноярск 2023