|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***Информационный портал сети досуговых центров***

Студент \_\_ИУ7-23М\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**Ф.С.Апальков**\_\_\_\_\_\_**

(Группа) (Подпись, дата) ( И.О. Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2022 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_И.В.Рудаков\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине Распределенные системы обработки информации\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы ИУ7-23М\_\_\_\_ Апальков Фёдор Станиславович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема курсовой работы Информационный портал сети досуговых центров\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КР: учебная\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики: кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к 9 нед., 50% к 12 нед., 75% к 15 нед., 100% к 17 нед.

***Задание*** \_Разработать прототип информационного портала сети досуговых центров на базе веб-интерфейса. Система должна состоять из микросервисов, каждый из которых отвечает за свою задачу: Сервис пользовательского интерфейса, сервис агрегирования запросов, сервис авторизации и данных пользовательских аккаунтов, сервис новостей, сервис расписаний, сервис центров, сервис записей на занятия. Каждый сервис при необходимости может иметь доступ к связанной с ним базе данных, но не должен иметь доступа к базам данных других сервисов. Запросы между сервисами требуют авторизацию. Запросы пользователей делятся на две категории: запросы, требующие авторизации пользователя, и запросы, доступные для всех пользователей, даже неавторизованных. Все ошибки должны обрабатываться\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Руководитель курсового проекта**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

# Оглавление

[Оглавление 3](#_Toc104022603)

[Аналитический раздел 5](#_Toc104022604)

[Описание системы 5](#_Toc104022605)

[Назначение разработки 5](#_Toc104022606)

[Существующие аналоги 6](#_Toc104022607)

[Общие требования к системе 6](#_Toc104022608)

[Требования к функциональным характеристикам 6](#_Toc104022609)

[Функциональные требования к порталу с точки зрения пользователя 7](#_Toc104022610)

[Требования к программной реализации 8](#_Toc104022611)

[Общие требования к подсистемам 9](#_Toc104022612)

[Конструкторский раздел 10](#_Toc104022613)

[Топология системы 10](#_Toc104022614)

[Общие требования к подсистемам 10](#_Toc104022615)

[Функциональные требования к сервисам 11](#_Toc104022616)

[Сценарии функционирования системы 14](#_Toc104022617)

[Регистрация пользователя: 14](#_Toc104022618)

[Авторизация пользователя: 15](#_Toc104022619)

[Получение информации о досуговых центрах: 15](#_Toc104022620)

[Получение новостей и расписания занятий для досугового центра: 16](#_Toc104022621)

[Создание или изменение информации о досуговом центре: 16](#_Toc104022622)

[Получение записей на занятия: 17](#_Toc104022623)

[Создание или удаление новости 17](#_Toc104022624)

[Создание или удаление элемента расписания: 17](#_Toc104022625)

[Спецификации сценариев 18](#_Toc104022626)

[Спецификация сценария регистрация пользователя 18](#_Toc104022627)

[Спецификация сценария регистрация пользователя: 19](#_Toc104022628)

[Спецификация сценария получение информации о досуговых центрах: 19](#_Toc104022629)

[Спецификация сценария получение новостей и расписания занятий для досугового центра: 20](#_Toc104022630)

[Спецификация сценария создание или изменение информации о досуговом центре (только с правами администратора): 21](#_Toc104022631)

[Спецификация сценария получение записей на занятия: 22](#_Toc104022632)

[Спецификация сценария создание или удаление новости 23](#_Toc104022633)

[Спецификация сценария Создание или удаление элемента расписания: 24](#_Toc104022634)

[Диаграммы последовательности действий 25](#_Toc104022635)

[Асинхронные запросы 26](#_Toc104022636)

[Технологический раздел 27](#_Toc104022637)

[Логический дизайн 27](#_Toc104022638)

[Спецификация таблицы User: 27](#_Toc104022639)

[Спецификация таблицы Center: 28](#_Toc104022640)

[Спецификация таблицы News: 28](#_Toc104022641)

[Спецификация таблицы Schedule: 29](#_Toc104022642)

[Спецификация таблицы Application: 29](#_Toc104022643)

[Реализация отказоустойчивости 29](#_Toc104022644)

[Сборка системы 30](#_Toc104022645)

Аналитический раздел

# Описание системы

Система должна представлять собой портал предоставляющий информацию заинтересованным лицам – людям которые хотят записаться на занятия в досуговом центре сети. Портал должен предоставлять возможность Администраторам центров обновлять новости и расписания досугового центра. Портал должен предоставлять возможность добавления в сеть досуговых центров новых узлов (центров).

Пользователь может выступать в качестве администраторов, создающих расписания и новости и в качестве суперпользователей, добавляющих новые досуговые центры в сеть и редактирующий информацию о них. Администраторы имеют возможность получать список пользователей, записавшихся на занятия.

Записи о новостях, расписаниях и информации о центрах, хранятся в базах данных связанных с ними. В целях экономии ресурсов, для каждого досугового центра не выделяется отдельная база данных с его новостями, расписанием и информацией о нем.

# **Назначение разработки**

Главное назначение разрабатываемого портала - предоставление любому пользователю возможности просмотра информации о досуговом центре и получения информации в контексте досугового центра о расписании его занятий и новостей. В контексте центра пользователь должен иметь возможность создать запись на интересующее его занятие. Администратор должен иметь возможность создавать и редактировать новости и элементы в расписании. Суперпользователь должен иметь возможность добавлять досуговые центра на портал и редактировать информацию о любых центрах. Администратор использует элементы расписания и новости для привлечения пользователей на занятия. Все пользователи портала просматривают портал для личных целей. Портал производит работу в отдельно взятом регионе.

# Существующие аналоги

Среди аналогов разрабатываемого проекта можно отметить такие порталы, как 2gis.ru, zoon.ru, yell.ru. Данный проект должен иметь следующие преимущества перед существующими аналогами:

1. Предоставление расписания для **каждого** досугового центра
2. Предоставление новостей для **каждого** досугового центра
3. Предоставление возможности записаться на занятие для **каждого** досугового центра

# Общие требования к системе

1. Система должна поддерживать возможность «горячего» переконфигурирования системы. Необходимо поддерживать возможность добавления нового центра, элемента расписания, новости и записи на занятие во время работы системы без рестарта;
2. Время восстановления системы после сбоя не должно превышать 15 минут;
3. Каждый узел должен автоматически восстанавливаться после сбоя;
4. Обеспечить безопасность работоспособности системы за счет отказоустойчивости узлов;

# Требования к функциональным характеристикам

1. Медиана времени отклика системы на запросы пользователя на получение информации не должна превышать 3 секунд
2. Медиана времени отклика системы на действия пользователя должна быть менее 800мс при условии работы на рекомендованной аппаратной конфигурации, задержках между взаимодействующими сервисами менее 200мс.

# Функциональные требования к порталу с точки зрения пользователя

Портал должен обеспечивать реализацию следующих функций:

1. Система должна обеспечивать регистрацию и авторизацию пользователей с валидацией вводимых данных через интерфейс приложения.
2. Система должна обеспечивать аутентификацию пользователей.
3. Система должна обеспечивать разделение пользователей на четыре роли:
   1. Неавторизированный пользователь;
   2. Пользователь;
   3. Администратор;
   4. Суперпользователь
4. Система должна предоставлять любому пользователю, в том числе и неавторизированному, следующие функции:
   1. Просмотр информации о досуговых центрах в сети
   2. Просмотр расписаний досуговых центров
   3. Просмотр новостей досуговых центров
5. Система должна предоставлять Пользователю с ролью **пользователь** следующие функции:
   1. Запись на занятие в досуговом центре
   2. Функции всех, в том числе и неавторизированных пользователей
6. Система должна предоставлять Пользователю с ролью **администратор** следующие функции:
   1. Создание элемента расписания в досуговом центре
   2. Создание новости в досуговом центре
   3. Удаление элемента расписания
   4. Удаление новости
   5. Функции пользователя с ролью пользователь
7. Система должна предоставлять Пользователю с ролью **суперпользователь** следующие функции:
   1. Создание нового досугового центра
   2. Редактирование информации о досуговом центре
   3. Функции пользователя с ролью администратор
   4. Выдача ролей другим пользователям
   5. Получение информации о пользователях

# Требования к программной реализации

1. Требуется использовать СОА (сервис-ориентированную архитектуру) для реализации системы;
2. Система состоит из микросервисов. Каждый микросервис отвечает за свою область логики работы приложения;
3. Взаимодействие между сервисами осуществляется посредством HTTP-запросов
4. Данные сервисов должны храниться в базе данных. Каждый сервис взаимодействует только со своей схемой данных. Взаимодействие сервисов происходит по технологии REST;
5. При недоступности систем портала должна осуществляться деградация функционала или выдача пользователю сообщения об ошибке;
6. Необходимо предусмотреть авторизацию пользователей через интерфейс приложения
7. Для запросов, выполняющих обновление данных на нескольких узлах распределенной системы, в случае недоступности одной из систем, необходимо выполнять полный откат транзакции;
8. Необходимо сохранять профили всех пользователей в базе данных в хешированном виде.
9. Приложение должно поддерживать возможность горизонтального и вертикального масштабирования за счет увеличения количества функционирующих узлов и совершенствования технологий реализации компонентов и всей архитектуры системы

# Общие требования к подсистемам

1. Фронт-энд – серверное приложение, при разработке которого следует учесть следующие нюансы:
   1. Фронт-энд должен принимать запросы по протоколу HTTP и формировать ответ пользователю в формате HTML-страниц;
   2. Фронт-энд является посредником между пользователями, передавая их запросы последовательно на сервис агрегации запросов;
   3. В соответствии с выбранными технологиями реализации целесообразно использовать библиотеку JQuery, языки HTML, CSS и JS, для создания верстки HTML-страниц проекта.
2. реализации бэк-эндов должны быть предъявлены следующие требования:
   1. Прием и возврат данных должен происходить в формате JSON по протоколу HTTP;
   2. Если результаты работы сервиса необходимо сохранять в базе данных, то доступ к ней должен осуществляться по протоколу HTTP. Доступ к базе данных может осуществляться только из подсистем, работающих напрямую с данными ее таблиц.

Конструкторский раздел

# Топология системы

Топология разрабатываемой системы, изображена на рисунке 1.

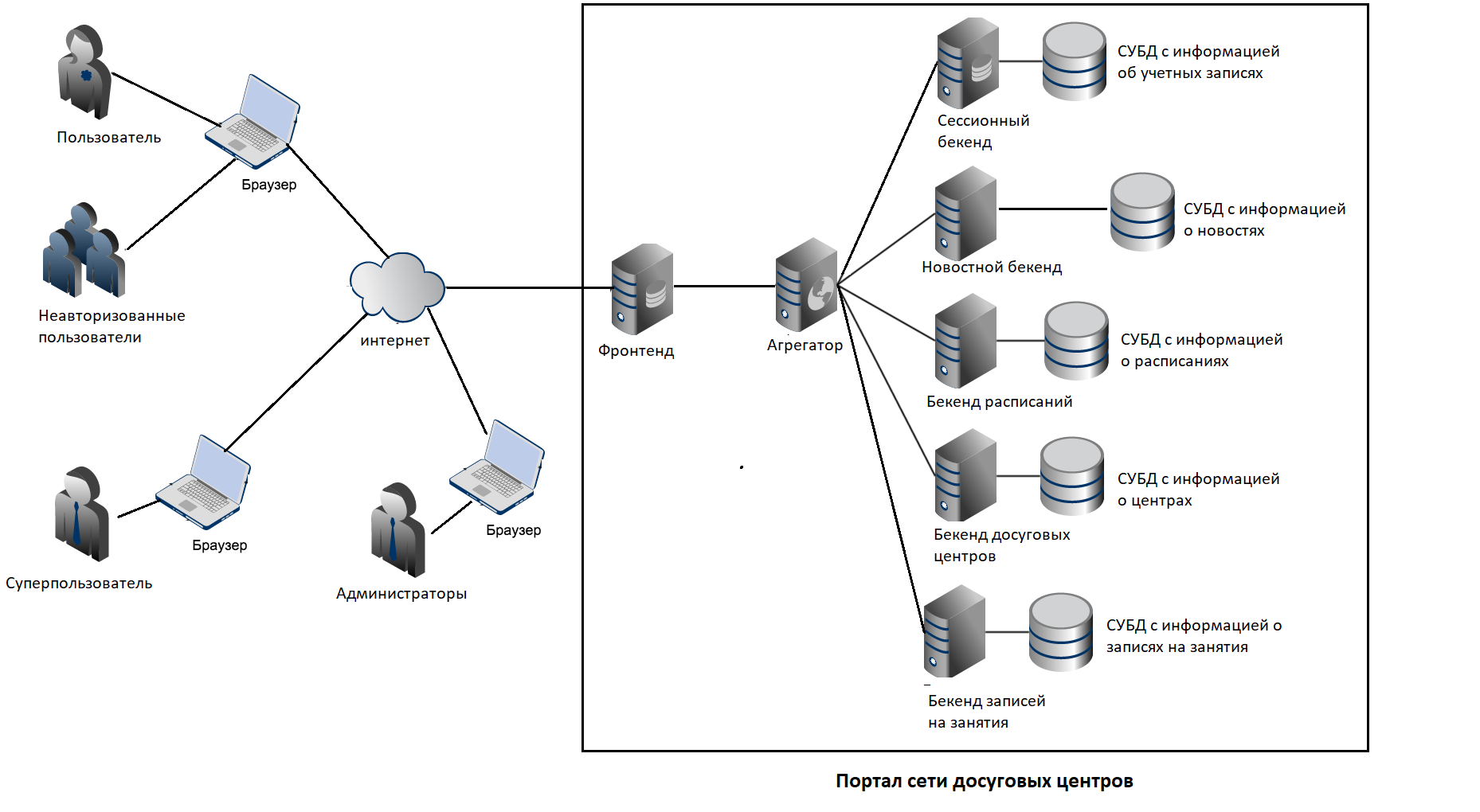


Рисунок 1. Топология системы

Разрабатываемая система состоит из Фронт-энда, Агрегатора, Сессионного сервиса и 4 подсистем:

* Сервис центров
* Сервис новостей
* Сервис расписаний
* Сервис записей на занятия

# Общие требования к подсистемам

1. Фронт-энд – серверное приложение, при разработке которого следует учесть следующие нюансы:
   1. Фронт-энд должен принимать запросы по протоколу HTTP и формировать ответ пользователю в формате HTML-страниц;
   2. Фронт-энд является посредником между пользователями, передавая их запросы последовательно на сервис агрегации запросов;
   3. В соответствии с выбранными технологиями реализации целесообразно использовать библиотеку JQuery, языки HTML, CSS и JS, для создания верстки HTML-страниц проекта.
2. реализации бэк-эндов должны быть предъявлены следующие требования:
   1. Прием и возврат данных должен происходить в формате JSON по протоколу HTTP;
   2. Если результаты работы сервиса необходимо сохранять в базе данных, то доступ к ней должен осуществляться по протоколу HTTP. Доступ к базе данных может осуществляться только из подсистем, работающих напрямую с данными ее таблиц.

# Функциональные требования к сервисам

1. **Сервис агрегации запросов** - предоставляет пользовательский интерфейс и внешний API системы.

Сервис должен реализовывать следующий функционал:

* Проверка существования пользователя
* Регистрация пользователя
* Аутентификация пользователя
* Получение информации о всех центрах
* Получение новостей для конкретно выбранного центра
* Получение расписания для конкретно выбранного центра
* Создание и удаление новости (только для ролей: Администраторы и Суперпользователи)
* Создание и удаление элемента расписания (только для ролей: Администраторы и Суперпользователи)
* Создание, удаление и редактирование конкретно выбранного центра (только для ролей: Суперпользователи)
* Просмотр информации о пользователях (только для ролей: Суперпользователи)
* Выдача ролей пользователю (только для ролей: Суперпользователи) (Изменение в базе данных записи о пользователе)
* Создание в базе данных записей о пользователях (Суперпользователи)
* Просмотр записей на занятия (только для ролей: Администраторы и Суперпользователи)

1. **Сервис работы с центрами** – отвечает за хранение информации о досуговых центрах и ее редактирование

Хранимая в базе данных сущность, ассоциированная с сервисом, имеет следующие обязательные поля:

* Название центра
* Информация о центре
* Адрес центра

Сервис должен реализовывать следующий функционал:

* Получение всей информации о центрах
* Редактирование записи в базе данных с информацией о центрах
* Удаление записи в базе данных записи с информацией о центре

1. **Сервис работы с новостями** – отвечает за хранение новостей и ее редактирование

Хранимая в базе данных сущность, ассоциированная с сервисом, имеет следующие обязательные поля:

* Содержание новости
* Дата создания новости
* Время создания новости
* Название центра, к которому принадлежит новость

Сервис должен реализовывать следующий функционал:

* Добавление новости (создание записи в базе данных)
* Удаление новости (удаление записи в базе данных)

1. **Сервис работы с расписаниями** – отвечает за хранение информации о расписаниях и ее редактирование

Хранимая в базе данных сущность, ассоциированная с сервисом, имеет следующие обязательные поля:

* Содержание элемента расписания
* Дата проведения элемента расписания
* Время проведения элемента расписания
* Название центра, к которому принадлежит элемент расписания

Сервис должен реализовывать следующий функционал:

* Добавление элемента расписания (создание записи в базе данных)
* Удаление элемента расписания (удаление записи в базе данных)

1. **Сервис работы с записями на занятия** – отвечает за хранение информации о записях на занятия и ее редактирование

Хранимая в базе данных сущность, ассоциированная с сервисом, имеет следующие обязательные поля:

* ID элемента расписания
* ID пользователя

Сервис должен реализовывать следующий функционал:

* Добавление записи на занятие (создание записи в базе данных)
* Удаление записи на занятие (удаление записи в базе данных)
* Просмотр всех записей на занятия

1. **Сервис регистрации и авторизации пользователей (Сессионный)**

Хранимая в базе данных сущность, ассоциированная с сервисом, имеет следующие обязательные поля:

* Логин
* Пароль
* Имя
* Роль

Сервис должен реализовывать следующий функционал:

* Идентификация и аутентификация пользователя (выдача токена)
* Регистрация пользователя (создание записи в базе данных)
* Выдача прав пользователю (изменение записи в базе данных)

Получение информации и о всех пользователях

# Сценарии функционирования системы

Сценарии функционирования или использования системы описывают конкретную последовательность действий, иллюстрирующую поведение пользователя при работе с приложением. Далее приведены подробные сценарии основных возможных действий пользователя.

## Регистрация пользователя:

1. Пользователь вводит логин, пароль и имя через интерфейс портала
2. Пользователь нажимает кнопку «Регистрация», тем самым подтверждая верность своих данных, а также согласие на их обработку и хранение;
3. Фронт-энд отправляет запрос на Агрегатор, Агрегатор отправляет запрос на сессионный сервис.
4. Из ответа Фронт-энд генерирует веб страницу с полученным ответом
5. Если пользователь с введенным для регистрации именем уже существует или не вводит имя или не вводит пароль, то клиент получает сообщение об ошибке. При успешной регистрации пользователь получает сообщение об успешной регистрации.

## Авторизация пользователя:

1. Пользователь вводит логин и пароль в интерфейсе портала;
2. Пользователь нажимает кнопку «Войти» в интерфейсе портала;
3. Фронт-энд отправляет запрос на Агрегатор, Агрегатор отправляет запрос на сессионный сервис.
4. При обнаружении ошибки в данных, пользователь получает сообщение об ошибке. При совпадении данных с записью в базе данных пользователей пользователь получает доступ к системе с правами, полученными из значения его роли, получает сообщение об успехе

## Получение информации о досуговых центрах:

1. При переходе пользователя на главную страницу портала, сервис Фронт-энд отправляет запрос на Агрегатор, Агрегатор отправляет запрос на сервис центров и получает всю информация о центрах;
2. Из ответа Фронт-энд генерирует страницу ответа
3. Фронт-энд возвращает пользователю веб страницу из полученных данных

## Получение новостей и расписания занятий для досугового центра:

1. Пользователь нажимает на интересующий его центр
2. Из базы данных новостей Сервис Агрегатор получает новости для интересующего центра
3. Из базы данных расписаний Сервис Агрегатор получает расписание для интересующего центра
4. Из ответа Фронт-энд генерирует страницу ответа
5. Фронт-энд возвращает пользователю страницу с полученными данными

## Создание или изменение информации о досуговом центре:

1. Пользователь нажимает кнопку «Центры»
2. Фронт-энд генерирует страницу с выбором действий, и отправляет запрос на сервис Агрегатор
   1. Пользователь вводит данные нового центра и нажимает кнопку «создать»
      1. Сервис Агрегатор создает новый центр, отправив запрос на сервис центров
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе создания
   2. Пользователь вводит ID центра на удаление и нажимает кнопку «удалить»
      1. Сервис Агрегатор удаляет центр, отправив запрос на сервис центров
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе удаления
   3. Пользователь вводит данные для обновления центра и нажимает кнопку «изменить»
      1. Сервис Агрегатор обновляет данные для центра, отправив запрос на сервис центров
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе обновления данных

## Получение записей на занятия:

1. Пользователь нажимает кнопку «записи»
2. Фронт-энд отправляет запрос на Агрегатор
3. Сервис Агрегатор получает все записи на занятия
4. Фронт-энд возвращает пользователю страницу со всеми записями на занятия

## Создание или удаление новости

1. Пользователь нажимает кнопку «новость»
2. Фронт-энд генерирует страницу с выбором действий, и отправляет запрос на Агрегатор
   1. Пользователь вводит данные новой новости и нажимает кнопку «создать»
      1. Сервис Агрегатор создает новость, отправив запрос на сервис новостей
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе создания
   2. Пользователь вводит ID новости и нажимает кнопку «удалить»
      1. Сервис Агрегатор удаляет новость, отправив запрос на сервис новостей
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе удаления

## Создание или удаление элемента расписания:

1. Пользователь нажимает кнопку «расписание»
2. Фронт-энд генерирует страницу с выбором действий, и отправляет запрос на Агрегатор
   1. Пользователь вводит данные нового элемента расписания и нажимает кнопку «создать»
      1. Агрегатор отправляет запрос на сервис расписаний
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе создания
   2. Пользователь вводит ID элемента расписания и нажимает кнопку «удалить»
      1. Сервис Агрегатор удаляет новость, отправив запрос на сервис расписаний
      2. Фронт-энд сообщает пользователю об успехе или неудаче в процессе удаления

# Спецификации сценариев

Приведенные сценарии могут иметь как основной поток выполнения, который выполняется чаще всего, так и альтернативные потоки, описывающие выполнение запроса при отклонении от основного хода сценария. Все возможные ходы выполнения сценария описываются при помощи спецификаций. Примеры спецификаций для описанных выше сценариев приведены в данном разделе.

## Спецификация сценария регистрация пользователя

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь вводит логин и пароль в интерфейсе портала, нажимает кнопку «Вход» | Пользователь получает сообщение об успешной авторизации |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь вводит неверный логин или пароль в интерфейсе портала, нажимает кнопку «Вход» | Система отвечает сообщением о невозможности авторизации |

## Спецификация сценария регистрация пользователя:

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь вводит логин, пароль и имя в интерфейсе портала, нажимает кнопку «Регистрация» | Пользователь получает сообщение об успешной регистрации |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь вводит логин или имя неправильного вида в интерфейсе приложения, нажимает кнопку «Регистрация» | Пользователь получает сообщение о невозможности регистрации пользователя с такими данными |

## Спецификация сценария получение информации о досуговых центрах:

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь переходит на главную страницу приложения | Пользователь получает веб страницу с информацией о досуговых центрах |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь переходит на главную страницу приложения, сервер центров находится в нерабочем состоянии | Пользователь получает сообщение о невозможности загрузки данных |

## Спецификация сценария получение новостей и расписания занятий для досугового центра:

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает на интересующий его центр | Пользователь получает веб страницу с информацией о центре |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает на интересующий его центр, сервис новостей или расписаний находится в нерабочем состоянии | Пользователь получает веб страницу без новостей или (и) расписаний |

## Спецификация сценария создание или изменение информации о досуговом центре (только с правами администратора):

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает на кнопку «Центры» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит данные нового центра и нажимает кнопку «создать» | Пользователь получает сообщение об успехе создания записи в базе данных |
| Пользователь вводит название центра на удаление и нажимает кнопку «удалить» | Пользователь получает сообщение об успехе удаления записи в базе данных |
| Пользователь вводит данные для обновления центра и нажимает кнопку «изменить» | Пользователь получает сообщение об успехе изменения записи в базе данных |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает на кнопку «Центры» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит данные нового центра и нажимает кнопку «создать», сервер центров находится в нерабочем состоянии или введено неверное название центра | Пользователь получает сообщение об ошибке создания записи в базе данных |
| Пользователь вводит название центра на удаление и нажимает кнопку «удалить», сервер центров находится в нерабочем состоянии или введено неверное название центра | Пользователь получает сообщение об ошибке удаления записи в базе данных |
| Пользователь вводит данные для обновления центра и нажимает кнопку «изменить», сервер центров находится в нерабочем состоянии или введено неверное название центра | Пользователь получает сообщение об ошибке изменения записи в базе данных |

## Спецификация сценария получение записей на занятия:

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «записи» | Пользователь получает веб страницу с информацией о всех записях на занятия |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «записи», сервер с записями находится в нерабочем состоянии | Пользователь получает сообщение о невозможности получения данных |

## Спецификация сценария создание или удаление новости

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «новость» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит данные новой новости и нажимает кнопку «создать» | Пользователь получает сообщение об успехе создания записи в базе данных |
| Пользователь вводит ID новости и нажимает кнопку «удалить» | Пользователь получает сообщение об успехе удаления записи в базе данных |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «новость» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит неправильные данные новой новости и нажимает кнопку «создать» | Пользователь получает сообщение об ошибке создания записи в базе данных |
| Пользователь вводит неправильный ID новости и нажимает кнопку «удалить» | Пользователь получает сообщение об ошибке удаления записи в базе данных |

## Спецификация сценария Создание или удаление элемента расписания:

|  |  |
| --- | --- |
| Нормальный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «расписание» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит данные нового элемента расписания и нажимает кнопку «создать» | Пользователь получает сообщение об успехе создания записи в базе данных |
| Пользователь вводит ID элемента расписания и нажимает кнопку «удалить» | Пользователь получает сообщение об успехе удаления записи в базе данных |

|  |  |
| --- | --- |
| Альтернативный ход сценария | |
| Действие пользователя | Отклик системы |
| Пользователь нажимает кнопку «расписание» | Пользователь получает веб страницу с выбором действий |
| Пользователь вводит неправильные данные нового элемента расписания и нажимает кнопку «создать» | Пользователь получает сообщение об ошибке создания записи в базе данных |
| Пользователь вводит неправильный ID элемента расписания и нажимает кнопку «удалить» | Пользователь получает сообщение об ошибке удаления записи в базе данных |

# Диаграммы последовательности действий

Для описания поведения компонентов системы на единой оси времени используются диаграммы последовательности действий, при помощи которых можно описать последовательность действий для каждого прецедента, необходимую для достижения цели. На рисунке 9 изображен процесс изменения данных для центра.

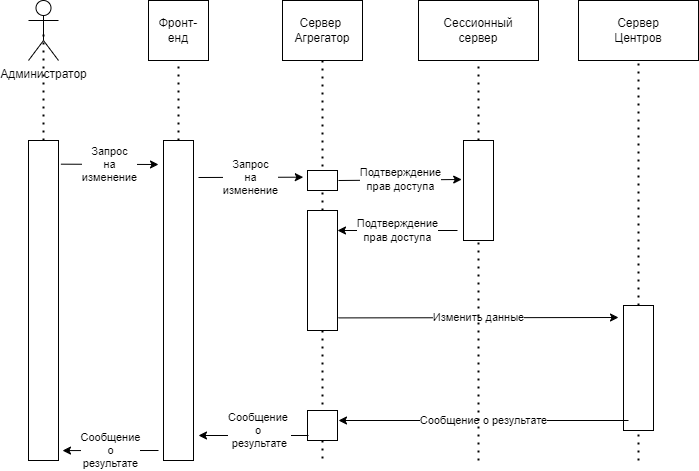


Рисунок 9. Диаграмма последовательности действий при изменении данных для центра

Пользователь отправляет запрос на получение списка записей на занятия на фронт-энд, фронт-энд отправляет запрос на сервис Агрегатор. Он в свою очередь запрашивает у сессионного сервиса разрешения на просмотр данной информации и получив ответ запрашивает у сервиса записей нужные данные. Возвращает их на фронт-энд, который возвращает пользователю веб страницу с полученными данными.

Аналогично происходят изменения записей в базах данных для сервиса новостей, расписаний и записей на занятия.

# Асинхронные запросы

https://developer.mozilla.org/ru/docs/Web/API/XMLHttpRequest/Synchronous\_and\_Asynchronous\_Requests

Технологический раздел

# Логический дизайн

На основе функциональных требований к выделенным подсистемам, а также объектов, о которых необходимо хранить данные в системе, была разработана схема данных приложения. Результат ее проектирования отображен на условной спецификации таблиц базы данных, приведенных на рисунке 7.

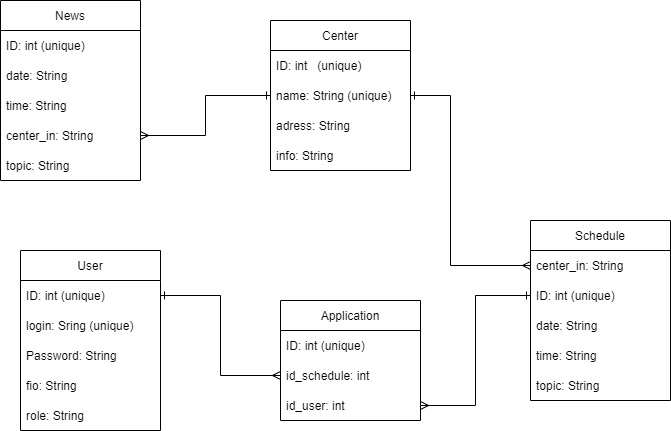


Рисунок 7. Схема базы данных системы.

## Спецификация таблицы User:

Таблица содержит данные о пользователях

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | Int | Идентификатор пользователя |
| login | String | Логин пользователя |
| password | String | Хешированный пароль пользователя |
| fio | String | Имя пользователя |
| role | String | Роль пользователя |

## Спецификация таблицы Center:

Таблица содержит данные о центрах

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | Int | Идентификатор центра |
| name | String | Название центра |
| adress | String | Адрес центра |
| info | String | Информация о центре |

## Спецификация таблицы News:

Таблица содержит данные о новостях

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | Int | Идентификатор новости |
| date | String | Дата новости |
| time | String | Время новости |
| center\_in | String | Названия центра в котором была новость |
| topic | String | Текст новости |

## Спецификация таблицы Schedule:

Таблица содержит данные о элементах расписаний

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | Int | Идентификатор элемента расписания |
| center\_in | String | Названия центра в котором находится элемент расписания |
| date | String | Дата элемента расписания |
| time | String | Время элемента расписания |
| topic | String | Текст элемента расписания |

## Спецификация таблицы Application:

Таблица содержит данные о записях на занятия

|  |  |  |
| --- | --- | --- |
| Имя атрибута | Тип атрибута | Описание атрибута |
| ID | Int | Идентификатор записи на занятие |
| id\_schedlue | Int | Идентификатор элемента расписания |
| id\_user | Int | Идентификатор пользователя |

## Реализация отказоустойчивости

Отказоустойчивость обеспечена автоматическим перезапуском приложения. Для этого в файле сборки docker-compose для каждого сервиса установить инструкцию restart: always, что будет обеспечивает его перезапуск (сервиса)

# Сборка системы

Каждый сервис имеет свой dockerfile.

Сборка системы происходит через docker-compose