**Содержание**

[Введение 3](#_Toc101404051)

[1 Аналитический раздел 4](#_Toc101404052)

[1.1 Описание предметной области 4](#_Toc101404053)

[1.2 Существующие аналоги 5](#_Toc101404054)

[1.3 Описание бизнес-логики информационной системы 5](#_Toc101404055)

[1.4 Требования к разрабатываемой системе 6](#_Toc101404056)

[1.5 Выводы к разделу 1 8](#_Toc101404057)

[2 Конструкторский раздел 9](#_Toc101404058)

[2.1 Топология системы 9](#_Toc101404059)

[2.2 Описание используемых алгоритмов 11](#_Toc101404060)

[2.3 Диаграмма вариантов использования (Use-Case) 13](#_Toc101404061)

[2.4 Диаграмма последовательности (Sequence) 16](#_Toc101404062)

[2.5 Диаграмма компонентов (Component) 17](#_Toc101404063)

[2.6 Выводы к разделу 2 19](#_Toc101404064)

[3 Технологический раздел 20](#_Toc101404065)

[3.1 Структура базы данных 20](#_Toc101404066)

[3.2 Тестирование 22](#_Toc101404067)

[3.3 Выводы к разделу 3 25](#_Toc101404068)

[Заключение 26](#_Toc101404069)

[Приложение А (обязательное) Библиографический список 27](#_Toc101404070)

Введение

Информационный портал – это интернет-ресурс, предоставляющий пользователям целый комплекс сервисов со сложной разветвленной структурой.

В наше время информационные порталы пользуются большой популярностью во всех сферах жизни общества, которые подвергаются неминуемой автоматизации. Одной из таких сфер является музейный туризм, поскольку удобство покупки билетов в любой интересующий музей определяет потенциальное количество посетителей данного музея.

Таким образом, создание информационного портала, помогающему пользователю купить билет в интересующий его музей, а также получить подробную информацию о музеях и выставках, поможет сэкономить уйму времени, и повысить популярность музейного туризма.

Целью данного курсового проекта является проектирование и разработка информационного портала музеев Российской Федерации.

1. Аналитический раздел

Данный раздел курсовой работы предназначен для того, чтобы более точно понимать направленность решаемой задачи, а именно, разработки информационного портала музеев Российской Федерации. Анализ задачи всегда влечёт за собой определение её требований. Первым делом необходимо произвести описание предметной области, которое поможет сформировать более чёткое понимание разрабатываемого портала.

* 1. Описание предметной области

В настоящее время существует проблема поиска информации в сфере музейной деятельности на территории Российской Федерации. Клиенты не могут находить необходимую информацию о выставках, смотреть часы работы музея и покупать билеты на едином портале.

Некоторые из музеев не имеют своих собственных сайтов. Та часть российских музеев, которые представлены на информационных порталах имеют свою специфику и свои хранилища данных. Поэтому основной задачей является создание единого государственного портала, объединяющего все российские музеи, в том числе и те, у которых нет никакого представительства в интернете. Основной задача данного портала – рассказать обо всех музея Российской Федерации, представить их выставки и постоянные экспозиции, рассказать о часах работы, что будет интересно для туристов и повысить посещаемость музеев.

В настоящий момент существует множество сервисов, предоставляющих информацию только о конкретных музеях, что сильно затрудняет поиск потенциальным посетителям. Ввиду того, что роль автоматизации клиентского сервиса с каждым годом возрастает, появляется необходимость в решении задачи проектирования и разработки системы поиска информации о всех музеев Российской Федерации. К тому же, данная система поможет повысить популярность и посещаемость региональных музеев.

* 1. Существующие аналоги

Среди аналогов можно отметить порталы culture.gosuslugi.ru и museum.ru. Данный проект должен иметь следующие преимущества перед существующими аналогами:

* Поиск информации о музеях не только в больших городах, но и в регионах.
* Поиск актуальной информации о выставках и точном местоположении музеев.
* Возможность покупки билета в музей на портале.
* Удобный интерфейс и высокая скорость загрузки страниц портала.
  1. Описание бизнес-логики информационной системы

Проект должен представлять собой портал для поиска информации и покупки билетов в музеи. В данной информационной системе предполагается наличие двух типов пользователей: посетитель и администратор.

Потенциальный посетитель регистрируется на портале и указывает информацию о себе: имя, фамилия, логин.

Администратор ответственен за добавление музеев на портал. При добавлении нового музея, он должен указать всю информацию о добавляемой организации: название музея, его описание, адрес местонахождения, наименование юридического лица, ИНН, ОГРН, тип музея, электронную почту.

После успешного добавления музея, появляется базовая страничка с главной информацией о нем, в которую он может вносить правки, а также добавлять информацию о проходящих выставках.

При добавлении информации о выставках, администратор должен указывать название, краткое описание выставки, информацию о билетах, даты проведения (в случае временной выставки).

На основе этой информации клиенты, посещающие портал, производят поиск подходящих музеев, узнают их особенности, а также имеют возможность приобрести билеты на представленные на портале выставки.

Помимо этого, зарегистрированный пользователь должен иметь возможность просмотра своего профиля, купленных им билетов и менять цветовую тему портала.

Администратор имеет возможности удаления музеев, выставок, а также просмотра статистики о регистрации пользователей и поступлении средств после покупки билетов на счет музеев.

* 1. Требования к разрабатываемой системе

1. Система должна обеспечивать разделение пользователей на две роли:

* клиент;
* администратор;

1. Каждый музей, представленный на портале, должен быть классифицирован по типу его деятельности. Система должна обеспечивать музею выбор категории из представленного ниже списка:

* архитектурно-ансамблевый;
* естественнонаучный;
* краеведческий;
* исторический;
* художественный;
* научно-технический;
* литературный;
* театральный;
* музыкальный;
* музей-заповедник.

1. В случае недоступности некритичного функционала, должна осуществляться деградация функциональности.
2. Система должна обеспечивать валидацию вводимых данных через интерфейс приложения при операциях добавления, изменения контента портала, а также при аутентификации и регистрации.
3. Система должна обеспечивать регистрацию.
4. Система должна обеспечивать авторизацию и аутентификацию пользователей.
5. Система должна предоставлять **клиенту** следующие функции:

* просмотр информации о представленных музеях: тип музея, описание, местоположение, контактные данные, представленные в нем выставки;
* покупка билетов на выставки;
* просмотр профиля;
* просмотр истории купленных билетов;
* смена цветовой палитры портала.

1. Система должна предоставлять **администратору** следующие функции:

* добавление информации о музеях;
* изменение информации о музеях;
* удаление информации о музеях;
* добавление информации о проходящих выставках;
* удаление информации о выставках музея;
* возможность просмотра статистики поступлений денежных средств на счет музеев;
* возможность просмотра статистики о регистрации новых пользователей;
* просмотр профиля;
* смена цветовой палитры портала.

1. Каждый сервис разрабатываемой системы при необходимости может иметь доступ к связанной с ним базе данных, но не должен иметь доступа к базам данных других сервисов.
2. Все запросы между сервисами требуют авторизации.
   1. Выводы к разделу 1

В представленном разделе произведен анализ предметной области и существующих аналогов. Были определены основные требования, которым должна удовлетворять разрабатываемая система. Так же в данном разделе было произведено подробное описание бизнес-логики данного информационного портала.

1. Конструкторский раздел

Каждая информационная система должна обеспечивать требуемую производительность, функциональность, безопасность, безотказную работу, пропускную способность и множество других важнейших для эффективной работы факторов. Это достигается путем грамотного проектирования системы.

В данном разделе курсового проекта будет описана архитектура и алгоритмы разрабатываемой системы, описаны отдельные компоненты системы, а также спроектированы основные диаграммы, описывающие работу системы и взаимодействие отдельных компонентов.

* 1. Топология системы

Система будет состоять из фронтэнда и пяти сервисов, что наиболее целесообразно для реализации ее основного назначения. Топология разрабатываемой системы представлена на рисунке 2.1.

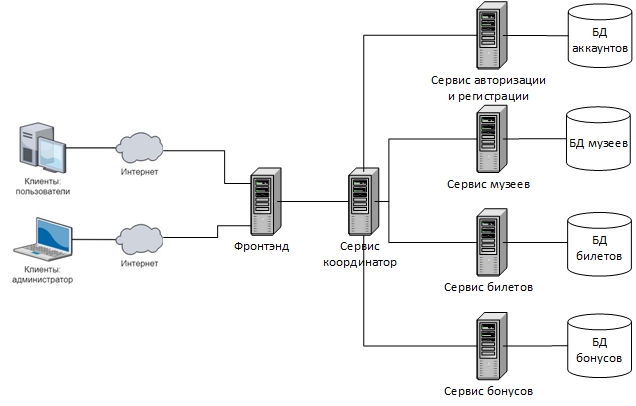


Рисунок 2.1 – Топология системы

**Сервис авторизации и регистрации** отвечает за пользователей портала и реализует следующие функции:

* + регистрация пользователя;
  + аутентификация пользователя;
  + просмотр профиля;
  + авторизация пользователя;
  + валидация токена;
  + получение и изменение цветовой темы портала.

**Сервис музеев** отвечает за хранение информации о представленных на портале музеях и реализует следующие функции:

* + получение списка музеев с условиями фильтрации по типу и местоположению;
  + получение информации о конкретном музее;
  + получение информации о проходящий в музее выставках и постоянных экспозициях;
  + добавление нового музея;
  + изменение информации конкретного музея;
  + удаление музея;
  + добавление выставки;
  + удаление выставки;
  + получение списка всех музейных типов;
  + начисление на счет музея средств после продажи билета.

**Сервис билетов** отвечает за хранение информации о билетах и реализует следующие функции:

* + покупка билета на конкретную экспозицию или выставку музея;
  + просмотр истории купленных пользователем билетов.

**Сервис статистики** отвечает за хранения информации о статистике и реализует следующие функции:

* + добавление и получение информации о всех поступлениях на счет музеев;
  + добавление и получение информации о новых пользователях системы.

**Сервис координатор** отвечает за диспетчеризацию запросов и предоставляет оптимальный унифицированный API.

**Фронтэнд** принимает запросы от пользователей по протоколу HTTP и анализирует их. На основе проведенного анализа фронтэнд выполняет запросы к координационному сервису, получает ответы и отсылает их пользователю.

* 1. Описание используемых алгоритмов

Все сервисы разрабатываемого портала используют протокол HTTP для получения и отправления информации. Все запросы, кроме получения токена проходят через координационный сервис.

Аутентификация на портале реализована с помощью Json Web Token. Json Web Token (JWT) — это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON [1]. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности [2].

Типичный алгоритм аутентификации на основе токенов представлен на рисунке 2.2.

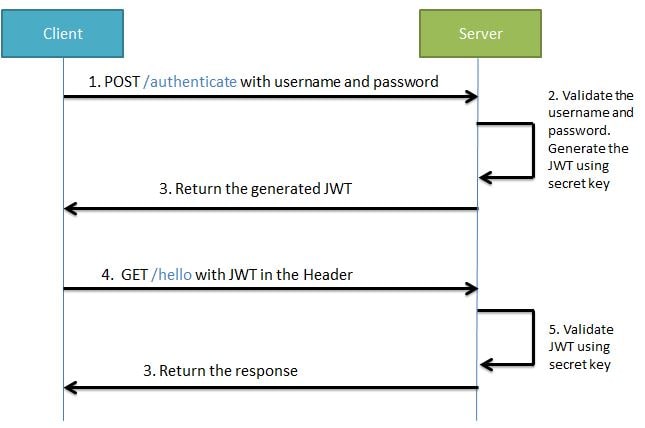


Рисунок 2.1 – Аутентификация на основе токенов

В разрабатываемой системе аутентификация на основе JWT-токена реализована следующим образом:

* + - 1. При успешной авторизации или регистрации нового пользователя, сервис авторизации и регистрации генерирует токен сроком действия в 10 минут и отправляет его на фронтэнд, который в свою очередь записывает токен в куки.
      2. При выполнении запросов, требующих авторизации, json web token подставляется в заголовок запроса. Сервис координатор перед выполнением запроса валидирует токен на сервисе авторизации и регистрации.
      3. Валидация токена осуществляется путем проверки даты действия токена и совпадения исходной уникальной подписи с тестовой. Если тестовая подпись совпадает с исходной, то это означает, что полезная нагрузка и заголовок не были изменены. Для генерации уникальной подписи используется секрет.
      4. В случае валидности токена, сервис координатор продолжает успешное выполнения запроса. В случае невалидности токена, сервис координатор возвращает 401 код ошибки HTTP.

Помимо этого, из важных особенностей реализации стоит отметить, что каждый сервис, кроме сервиса координатора имеет свое собственное хранилище данных, доступ к которому есть только у него.

База данных сервиса авторизации и регистрации хранит пользовательские пароли в хэшированном виде. В качестве алгоритма хэширования используется bcrypt.

bcrypt — адаптивная криптографическая хэш-функция формирования ключа, используемая для защищенного хранения паролей [3]. Функция основана на шифре Blowfish. Для защиты от атак с помощью радужных таблиц, bcrypt использует соль, кроме того, функция является адаптивной, время её работы легко настраивается и её можно замедлить, чтобы усложнить атаку перебором.

Также, в системе организована пагинация некоторых запросов для уменьшения поискового трафика и фильтрация данных, выполняемая на фронтэнде.

* 1. Диаграмма вариантов использования (Use-Case)

Диаграмма вариантов использования описывает функциональное назначение системы, то есть что система будет делать в процессе своего функционирования [4].

Use Case является исходной концептуальной моделью системы в процессе ее проектирования и разработки.

Цели построения диаграммы Use Case:

* Определение общих границ и контекста моделируемой предметной области на начальных этапах проектирования.
* Формулирование общих требований к функциональному проектированию системы.
* Разработка исходной концептуальной модели системы для ее последующей реализации.
* Подготовка документации для взаимодействия разработчика системы с ее заказчиком и пользователями.

В ходе проектирования система представляется в виде множества актеров, взаимодействующих с системой с помощью прецедентов.

Таким образом, основными компонентами диаграммы вариантов использования являются:

* Актеры – взаимодействуют с системой и используют ее функциональные возможности для достижения определенных целей и решения частных задач. Представляют собой внешнюю по отношению к моделируемой системе сущность. Может рассматриваться как некая роль относительно конкретного варианта использования.
* Прецеденты – определяют последовательность действий, которую должна выполнять система при взаимодействии ее с соответствующим актером.
* Отношения – показывают тип взаимодействия актеров и прецедентов. Один актер может взаимодействовать с несколькими вариантами использования и наоборот.

Существует 4 вида отношений между актерами и прецедентами:

* Ассоциативное отношение – устанавливает какую конкретную роль актер играет при взаимодействии с вариантом использования.
* Отношение расширения – определяет взаимосвязь базового варианта использования с некоторым другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а при выполнении некоторых дополнительных условий.
* Отношение обобщения – служит для указания того факта, что некоторый вариант использования может быть обобщен до другого варианта использования.
* Отношение включения – указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

Разработанные диаграммы вариантов использования для актеров клиент и администратор представлены на рисунках 2.2 и 2.3 соответственно.



Рисунок 2.2 – Диаграмма вариантов использования для клиента



Рисунок 2.3 – Диаграмма вариантов использования для администратора

* 1. Диаграмма последовательности (Sequence)

Диаграммы последовательности действий отображают взаимодействие объектов, упорядоченное во времени [5].

К основным компонентам диаграммы последовательности можно отнести:

* Объекты – основные компоненты системы.
* Линия жизни – вертикальная линия, которая показывает создание и уничтожение объекта, а также на ней находится фокус управления.
* Сообщения – законченный фрагмент информации, который отправляется одним объектом другому.

В ходе проектирования информационного портала музеев Российской Федерации была построена диаграмма последовательности системы, которая представлена на рисунке 2.4.

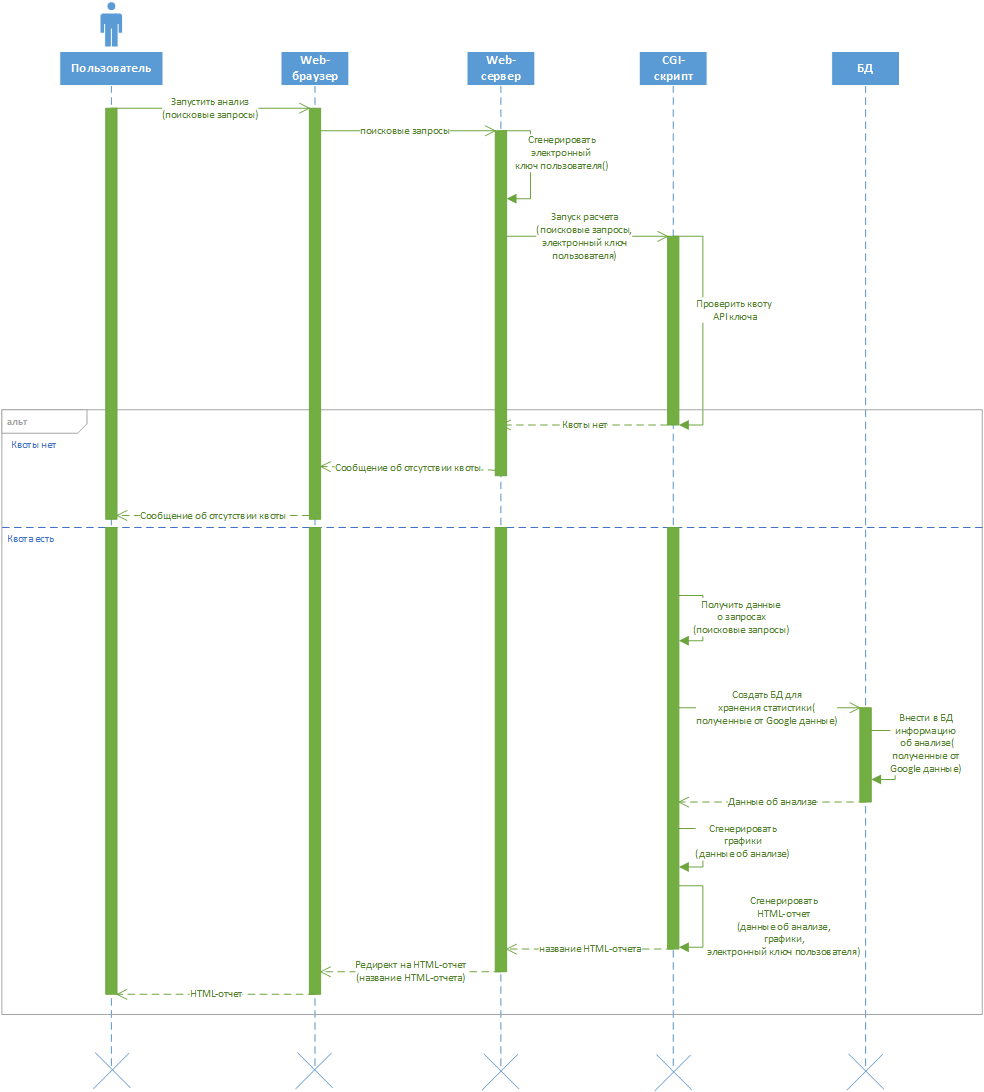


Рисунок 3.3.1 – Диаграмма последовательности системы

Таким образом, в данном подразделе курсового проекта была построена диаграмма последовательности информационной системы автоматизации аналитики роликов на платформе YouTube.

* 1. Диаграмма компонентов (Component)

Диаграмма компонентов — статическая структурная диаграмма, показывает разбиение системы на структурные компоненты и связи между ними. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы и пакеты. Такой вид диаграммы позволяет переходить от логического представления системы к ее реализации в виде программного кода [6].

Для представления физических сущностей в языке UML применяется термин – компонент. Компонент реализует некоторый набор интерфейсов и служит для общего обозначения элементов физического представления модели.

Компонент может иметь также свои собственные свойства, такие как атрибуты и операции.

Диаграмма компонентов разрабатываемой системы представлена на рисунке 2.7.

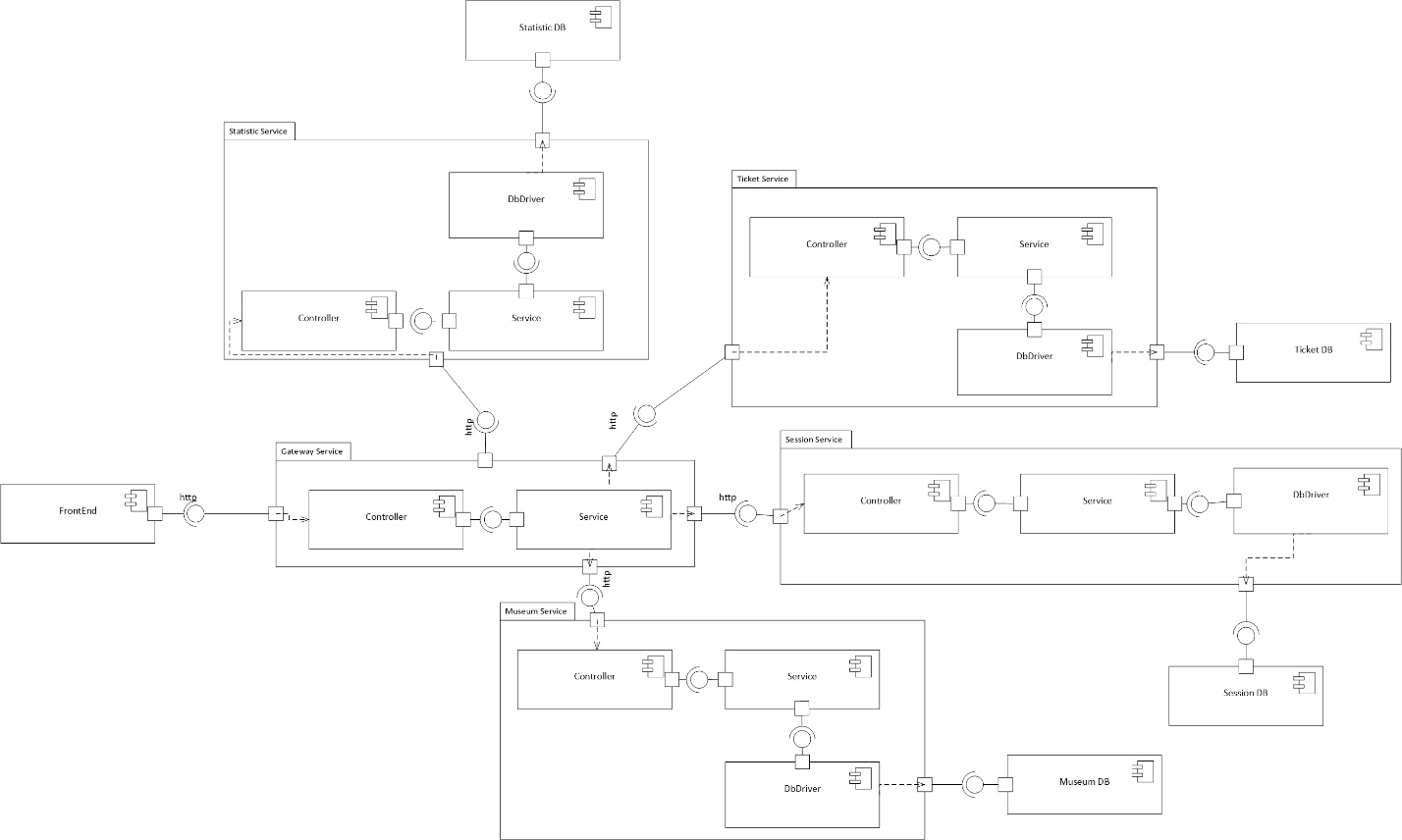


Рисунок 2.7 – Диаграмма компонентов

На данном этапе работы над курсовым проектом была спроектирована диаграмма компонентов, показывающая основные структурные компоненты информационной системы и связи между ними.

* 1. Выводы к разделу 2

В рассмотренном выше разделе описана архитектура, основные алгоритмы разрабатываемой системы и отдельные компоненты информационного портала.

Помимо этого, были спроектированы диаграммы вариантов использования для каждой из представленных на портале ролей. А также диаграммы последовательности действий, показывающие взаимодействие микросервисов.

1. Технологический раздел

На данном этапе разработки информационного портала будет произведено описание типов и структур данных в нотации IDEF1x, а также тестирование, обработка ошибок и поведение системы в случае отказа.

* 1. Структура базы данных

Модели данных служат для проектирования структуры постоянных хранилищ данных, используемых системой. При проектировании информационного портала всех музеев Российской Федерации были разработаны модели данных сервиса авторизации и регистрации, сервиса музеев, сервиса билетов и сервиса статистики в нотации IDEF1x [7].

Диаграммы базы данных сервиса авторизации и регистрации, сервиса музеев, сервиса билетов и сервиса статистики представлены на рисунках 3.1 – 3.4 соответственно.

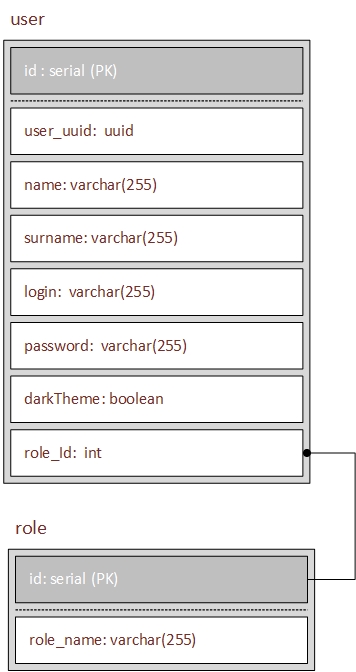


Рисунок 3.1 – БД сервиса авторизации и регистрации

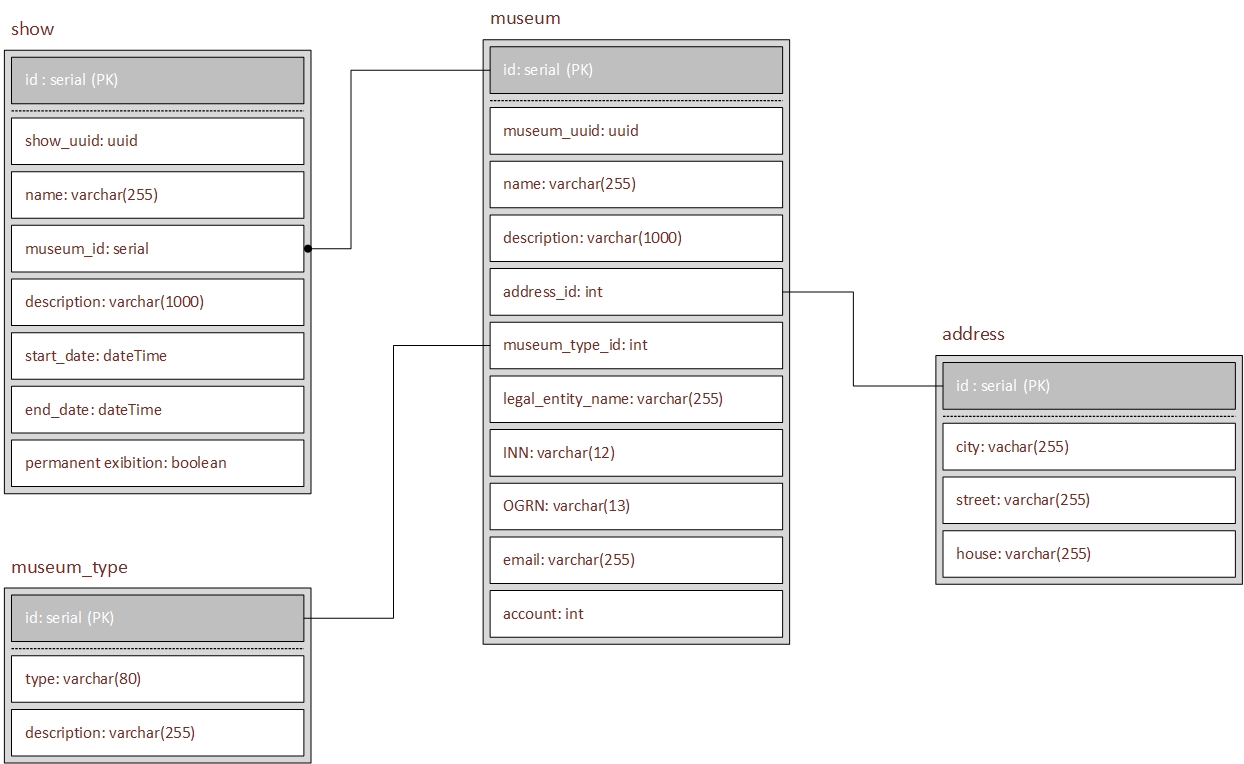


Рисунок 3.2 – БД сервиса музеев

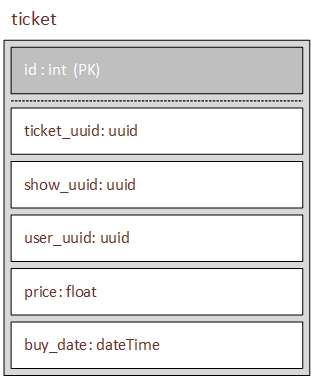


Рисунок 3.3 – БД сервиса билетов

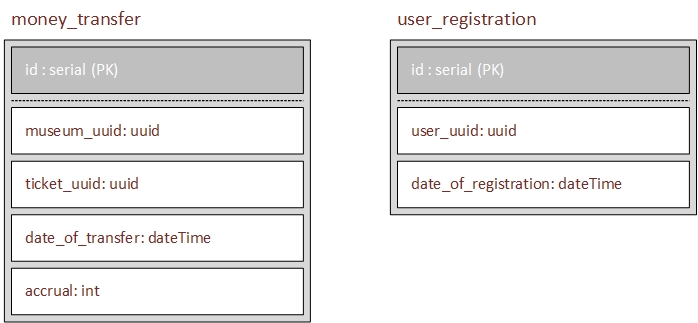


Рисунок 3.4 – БД сервиса статистики

* 1. Тестирование

При разработке информационного портала всех музеев Российской Федерации необходимо учитывать случаи отказа отдельных компонентов системы. В случае ошибки или недоступности некритичного функционала должна производиться деградация функциональности.

В данном проекте деградация функциональности реализована при запросах к сервису статистики. В случае, если при регистрации нового пользователя сервис статистики не отвечает, то пользователь получает ответ об успешной регистрации, а сервис координатор ставит запрос на добавление статистики о зарегистрированных пользователях в очередь. После чего, сервис координатор каждые 5 секунд выполняет запрос из очереди. В случае неудачи, запрос снова помещается в очередь.

Также, деградация функциональности реализована и при покупке билета на выставку. В случае, если при покупке билета сервис статистики не отвечает, то пользователь получает ответ об успешном приобретении билета, а сервис координатор ставит запрос на добавление истории начислений на счет музея в очередь. После чего, сервис координатор каждые 5 секунд выполняет запрос из очереди. В случае неудачно запрос снова помещается в очередь.

Для выполнения запросов, создается отдельный поток для каждой очереди.

Помимо этого, в разрабатываемой системе была предусмотрена обработка ошибок запросов получения информации. Для этого, на фронтэнде проверяется ответ, пришедший от сервисов. В случае возникновения ошибки, фронтэнд не прогружает компоненты, содержащие запрашиваемые данные. К примеру, в случае возникновения ошибки в запросе получения истории билетов, фронтэнд не станет добавлять данный компонент. Примеры успешной и ошибочной работы запросов представлены на рисунках 3.5 и 3.6 соответственно.

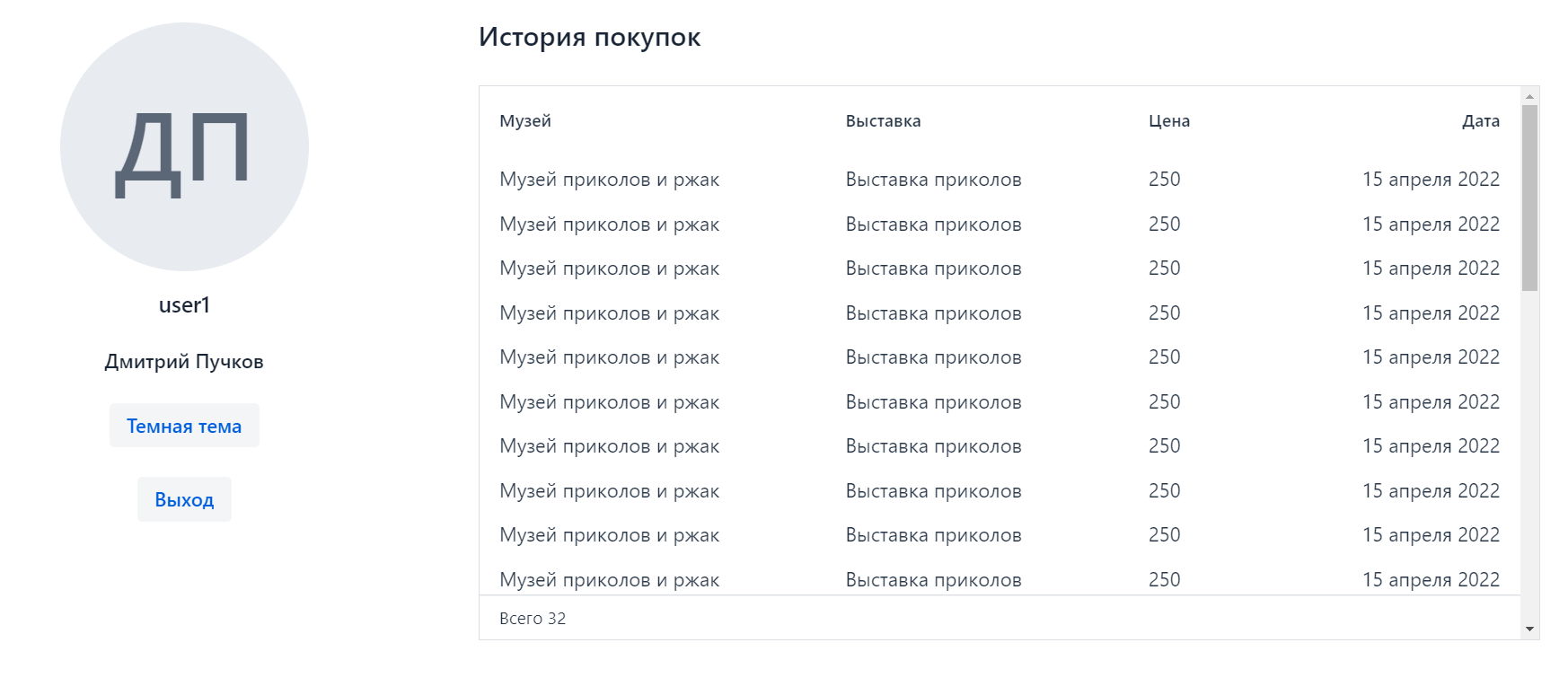


Рисунок 3.5 – Успешный запрос



Рисунок 3.6 – Ошибочный запрос

Также, для устойчивой работы сервисов в разрабатываемом портале предусмотрена валидация вводимых данных на всех представлениях. Валидация включает в себя проверку заполненности обязательных полей, проверку соответствия вводимой информации требуемому шаблону (реализовано с помощью регулярных выражений), а также проверку уникальности вводимых значений. В случае ввода невалидной информации, фронтэнд отобразит пользователю всплывающее уведомление о неправильности введенных данных. Примеры всех перечисленных выше типов валидации представлены на рисунках 3.7 – 3.9.

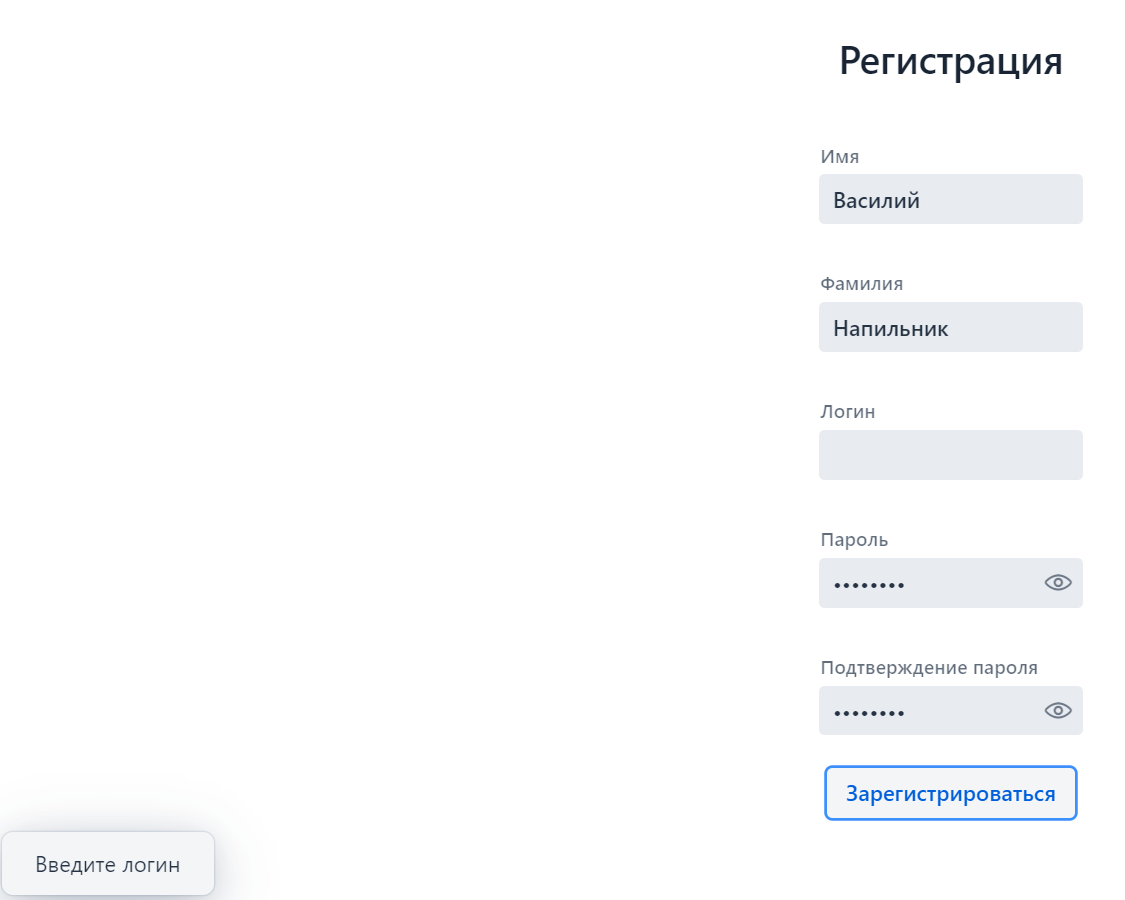


Рисунок 3.7 – Проверка заполненности обязательных полей

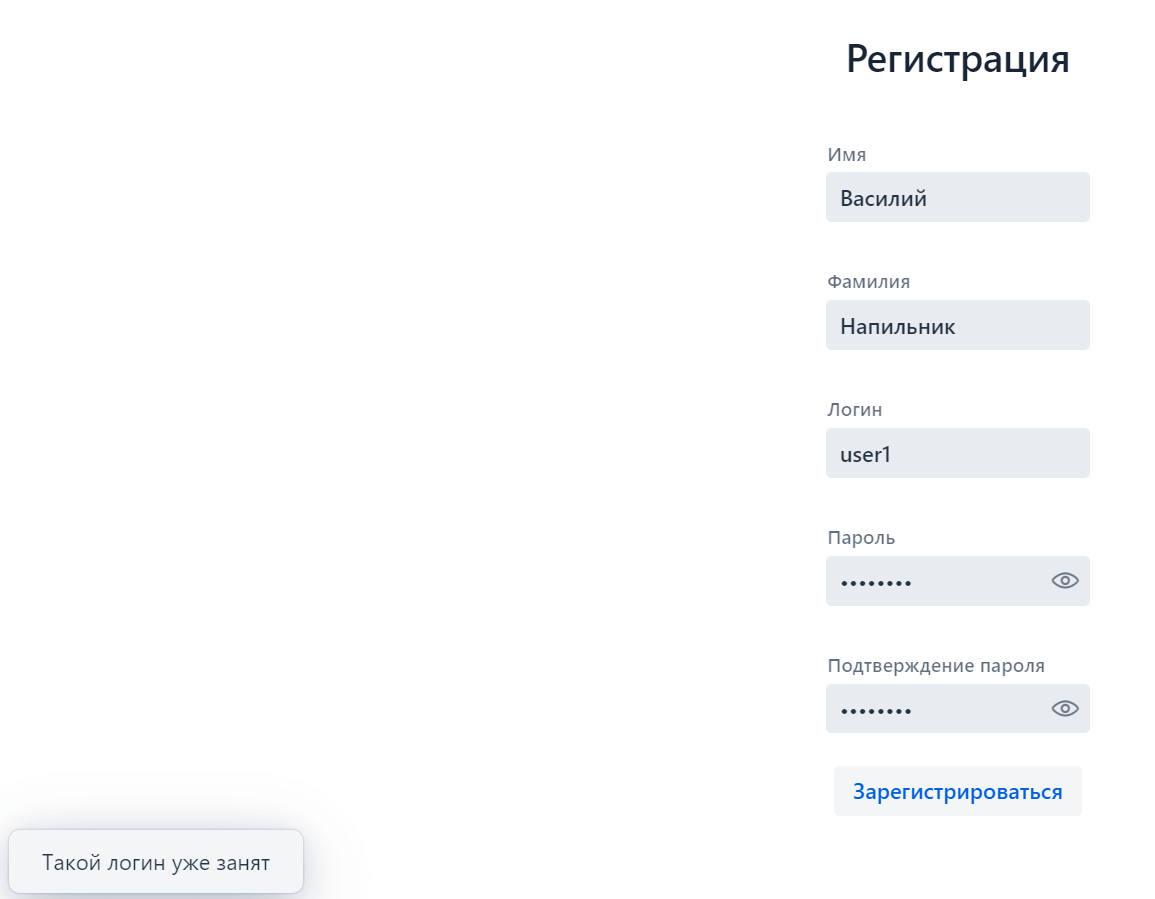


Рисунок 3.8 – Проверка уникальных данных



Рисунок 3.9 – Проверка соответствия шаблону

* 1. Выводы к разделу 3

В рассмотренном выше разделе было произведено описание структур баз данных всех сервисов, представленных в системе, а также описано поведение системы в случае ошибок и отказов.

Заключение

В ходе работы над курсовым проектом были проанализированы предметная область и определены требования к информационному порталу, что помогло при дальнейшем его проектировании.

Помимо этого, была описана топология системы, алгоритмы взаимодействия сервисов портала. Построены диаграммы вариантов использования, диаграммы последовательности действий и диаграмма компонентов.

Были описаны структуры данных всех сервисов, а также поведение системы в случае отказов и ошибок.

Приложение А  
(обязательное)  
Библиографический список

1. JavaInUse [Электронный ресурс] // JWT. URL: https://www.javainuse.com/spring/boot-jwt (дата обращения: 15.03.2022).
2. Stackoverflow [Электронный ресурс] // What is secret key for JWT and how to generate it. URL: https://stackoverflow.com/questions/31309759/what-is-secret-key-for-jwt-based-authentication-and-how-to-generate-it (дата обращения: 15.03.2022).
3. Wikipedia [Электронный ресурс] // bcrypt. URL: https://ru.wikipedia.org/wiki/Bcrypt (дата обращения: 17.03.2022).
4. Testengineer [Электронный ресурс] // Что такое use case? Теория и примеры. URL: https://testengineer.ru/chto-takoe-use-case (дата обращения: 24.03.2022).
5. Wikipedia [Электронный ресурс] // Диаграмма последовательности. URL: https://ru.wikipedia.org/wiki/Диаграмма\_последовательности (дата обращения: 27.03.2022).
6. Maccase [Электронный ресурс] // UML Component. URL: https://maccase.ru/android/uml-diagramma-komponentov-opisanie-modelirovanie-na-uml-diagrammy.html (дата обращения: 14.04.2022).
7. Analyst [Электронный ресурс] // IDEF1x. URL: http://analyst.by/diagrams/logicheskaya-model-predmetnoy-oblasti (дата обращения: 21.04.2022).