Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет   
им Г.И. Носова»

Многопрофильный колледж

Отделение №2 Информационные технологии и транспорт

**КУРСОВОЙ ПРОЕКТ**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

по ПМ.05. Проектирование и разработка информационных систем

МДК.05.02. Разработка кода информационных систем

на тему: Разработка автоматизированной информационной системы по поиску досуга

Исполнитель: Аверин.А.А . студент 3 курса, группа ИспВ-20-1

Руководитель: Зорина И.Г., преподаватель МпК

Работа допущена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Магнитогорск, 2023

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет   
им. Г. И. Носова»

Многопрофильный колледж

**КУРСОВОЙ ПРОЕКТ**

Тема Разработка автоматизированной информационной системы по поиску досуга

Студент Аверин Артём Александрович

Задание

Исходные данные:

по заданной предметной области:

1. выполнить моделирование информационной системы;
2. спроектировать и разработать базу данных;
3. разработать серверную и клиентскую части приложения;
4. разграничить права доступа;
5. выполнить тестирование информационной системы.

Состав и содержание проекта:

1. построение UML-диаграмм: вариантов использования, последовательности, состояния, деятельности, классов; потоков данных;
2. логическое проектирование, описание концептуальной модели данных, разработка базы данных в выбранной СУБД;
3. разработка интерфейса приложения, логики работы приложения: оформление заявок на услуги, вывод отчетной документации и т.д.;
4. администрирование приложения, разработка панели администратора;
5. разработка комплекта тестирования информационной системы.

Срок сдачи: « » 20 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.Г. Зорина

« » 2023 г.

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Аверин

« » 2023г.

Магнитогорск, 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc139613344)

[1 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ВЕБ-САЙТА ПО ПОИСКУ РЕСТАРАНОВ» 6](#_Toc139613345)

[1.1 Анализ предметной области 7](#_Toc139613346)

[1.2 Анализ существующих программных средств 8](#_Toc139613347)

[1.3 Моделирование информационной системы 8](#_Toc139613348)

[1.4 Логическое и физическое проектирование базы данных 16](#_Toc139613349)

[1.5 Выбор и обоснования средств разработки 19](#_Toc139613350)

[2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 21](#_Toc139613351)

[2.1 Разработка базы данных 21](#_Toc139613352)

[2.2 Логическая и физическая структура сайта 24](#_Toc139613353)

[2.3 Разработка интерфейса пользователя 25](#_Toc139613354)

[2.4 Разработка выходных документов 25](#_Toc139613355)

[2.5 Разграничение прав доступа 25](#_Toc139613356)

[2.6 Руководство пользователя 25](#_Toc139613357)

[2.7 Тестирование информационной системы 25](#_Toc139613358)

[ЗАКЛЮЧЕНИЕ 26](#_Toc139613359)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 27](#_Toc139613360)

# ВВЕДЕНИЕ

Сфера развлечений является одной из наиболее динамичных и перспективных в мире бизнеса. Современный рынок развлечений охватывает множество видов деятельности, от киноиндустрии до парков аттракционов, от концертов до игровой индустрии. В связи с развитием технологий и изменениями в образе жизни людей, сегодня сфера развлечений имеет большое значение для экономики и культуры в целом. Каждый день она привлекает миллионы людей по всему миру и представляет огромный потенциал для бизнеса и инноваций. В рамках проекта будем исследовать различные аспекты сферы развлечений и разработаем концепцию для создания успешного бизнеса в этой области.

Современный ритм жизни требует постоянного движения и развития, и часто видим, как люди проводят все свое свободное время в поисках новых развлечений и разнообразных мероприятий. Однако, существует проблема в том, что иногда тратится больше времени на поиск достойных мероприятий, чем на само развлечение. Именно поэтому, в рамках проекта ставим цель создать систему поиска развлечений, которая может значительно упростить этот процесс.

Целью нашего проекта является разработка и внедрение инновационной системы поиска развлечений, которая поможет людям находить различные мероприятия, соответствующие их интересам и предпочтениям. Система будет основана на использовании современных технологий и алгоритмов, которые обеспечат быстрый и удобный поиск.

Одной из главных задач проекта является сбор данных о различных мероприятиях и развлечениях с помощью базы данных, которая будет содержать информацию о культурных, спортивных и других мероприятиях в различных городах и регионах. Эта база данных будет также содержать информацию об интересах и предпочтениях пользователей, что позволит системе быстро найти подходящие мероприятия.

Важным аспектом проекта является создание удобного и интуитивно понятного интерфейса для пользователей, который будет демонстрировать мероприятия в зависимости от их интересов и местоположения. Будем использовать современные технологии для разработки приложения, которое будет доступно на планшете и веб-сайте.

Ожидаемый результат проекта - успешный запуск системы поиска развлечений, которая станет удобным и надежным инструментом для людей, которые ищут новые мероприятия и хотят проводить свое свободное время с удовольствием. Также предполагаем, что наш бизнес будет приносить значительную прибыль и станет лидером в отрасли, в которой конкуренция велика.

# 1 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ВЕБ-САЙТА ПО ПОИСКУ РЕСТАРАНОВ»

В современном мире информация является одним из наиболее ценных ресурсов, и ее правильное использование может стать ключевым фактором успеха в любой области деятельности. Информационные системы играют важную роль в обработке и хранении информации, а также в автоматизации бизнес-процессов.

Проектирование информационной системы – это процесс создания информационная система, который начинается с определения требований к системе и заканчивается ее внедрением и эксплуатацией. Основная цель проектирования информационная система – создание системы, которая будет эффективно выполнять свои функции, удовлетворять потребности пользователей и соответствовать бизнес-целям организации.

Процесс проектирования информационная система включает следующие этапы:

* Определение требований к системе. На этом этапе определяются цели и задачи информационная система, ее функциональные и нефункциональные требования, а также особенности пользователей и окружающей среды;
* Анализ и проектирование системы. На этом этапе проводится анализ существующих решений и выбор наиболее подходящей архитектуры информационная система. Также определяются методы и технологии, которые будут использоваться при разработке системы;
* Разработка и тестирование системы. На этом этапе происходит разработка программного обеспечения, создание баз данных и интерфейсов пользователя. После этого проводится тестирование системы на соответствие требованиям и ее готовности к внедрению;
* Внедрение и эксплуатация системы. На этом этапе информационная система устанавливается на рабочие места пользователей и начинается ее эксплуатация. Также проводится обучение пользователей и поддержка системы в рабочем состоянии.

При проектировании информационная система необходимо учитывать следующие факторы:

* Безопасность. Информационная система должна быть защищена от несанкционированного доступа и взлома;
* Масштабируемость. Система должна быть способна расширяться и адаптироваться к изменяющимся требованиям бизнеса;
* Надежность. Информационная система должна быть стабильной и надежной, чтобы избежать потери данных и сбоев в работе;
* Удобство использования. Система должна быть простой и удобной в использовании, чтобы пользователи могли быстро освоить ее функционал и эффективно работать с ней.

Проектирование информационной системы – это сложный процесс, который требует высокой квалификации и опыта в области информационных технологий. Однако, правильно спроектированная информационная система может стать ключевым фактором успеха организации и обеспечить ее конкурентоспособность на рынке.

## 1.1 Анализ предметной области

Для успешной реализации проекта системы поиска развлечений, необходимо провести анализ предметной области, который поможет определить основные проблемы и требования пользователей и конкурентов.

К началу 2023 года, существует множество онлайн-сервисов, которые предоставляют информацию о различных мероприятиях. Однако, согласно анализу, эти сервисы имеют следующие недостатки:

- Некоторые сервисы не содержат всю необходимую для пользователя информацию;

- Существующие сервисы могут быть сложны в использовании и не интуитивно понятны для пользователей.

Чтобы создать успешную систему поиска развлечений, необходимо решить эти описанные выше проблемы.

- Предоставление пользователю всей необходимой информации о мероприятиях;

- Создание простого и интуитивно понятного интерфейса для пользователей.

Кроме того, при проведении анализа предметной области необходимо учитывать конкурентов. Обзор конкурентной среды может помочь в определении преимуществ, которые можно использовать для создания более качественной системы поиска развлечений. Среди основных конкурентов можно выделить рейтинговые сервисы, такие как TripAdvisor и Yelp, а также базы данных мероприятий Eventful и Songkick.

Важной составляющей анализа является изучение потребностей пользователя. В данной предметной области, целевой аудиторией являются люди всех возрастных групп и социальных слоев, которые заинтересованы в проведении свободного времени с удовольствием. Они могут быть склонны к проведению своих развлечений с друзьями или семьей, а также могут искать новые знакомства и интересную компанию на мероприятиях.

Исходя из проведенного анализа, ключевыми требованиями к созданию успешной системы поиска развлечений являются: полнота и точность предоставляемой информации, интуитивно понятный интерфейс.

## 1.2 Моделирование информационной системы

Моделирование информационной системы – это процесс создания абстрактной модели, которая описывает функциональность и структуру информационной системы. Моделирование является одним из важных этапов в разработке информационных систем, так как позволяет увидеть все аспекты системы до ее реализации.

Для моделирования информационной системы используются различные методы и технологии, такие как UML, ER-моделирование, DFD-моделирование и другие. Они позволяют описать структуру и функциональность системы, а также взаимодействие ее компонентов.

UML – это язык моделирования, который позволяет описать структуру и поведение системы. UML используется для создания диаграмм классов, диаграмм последовательностей, диаграмм состояний и других типов диаграмм.

ER-моделирование – это метод моделирования, который используется для описания структуры базы данных. ER-моделирование позволяет создавать диаграммы сущностей и связей между ними.

DFD-моделирование – это метод моделирования, который используется для описания потоков данных в системе. DFD-моделирование позволяет создавать диаграммы уровней, на которых отображаются процессы, потоки данных и хранилища данных.

При моделировании информационной системы необходимо учитывать ее цели и задачи, а также требования к функциональности и производительности. Моделирование позволяет выявить возможные проблемы и недостатки системы на ранних этапах разработки, что позволяет сократить время и затраты на ее доработку.

В результате моделирования информационной системы создается абстрактная модель, которая может быть использована для создания реальной системы. Моделирование является важным этапом в разработке информационных систем и позволяет создать эффективную и функциональную систему, которая соответствует требованиям бизнеса.

Диаграмма прецедентов UML– это графическое представление функциональности системы, которую должна выполнять система для достижения конкретных целей. Данная диаграмма позволяет описать все возможные действия пользователей и системы, а также их взаимодействие.

Диаграмма прецедентов состоит из следующих элементов:

* Прецедент. Это описание конкретной функциональности системы, которую должна выполнять система для достижения конкретных целей;
* Актер. Это пользователь или система, который использует систему для выполнения определенных задач;
* Связь между актером и прецедентом. Это описание взаимодействия между пользователем и системой.

Для создания диаграммы прецедентов необходимо выполнить следующие шаги:

* Определить актеров. Необходимо определить всех пользователей и системы, которые используют систему для выполнения определенных задач;
* Определить прецеденты. Необходимо определить все возможные функциональности системы, которые должна выполнять система для достижения конкретных целей;
* Определить связи между актерами и прецедентами. Необходимо определить все возможные взаимодействия между пользователем и системой;
* Проверить корректность диаграммы. Необходимо проверить корректность диаграммы прецедентов, чтобы убедиться, что все прецеденты и актеры правильно описаны.

Диаграмма прецедентов UML позволяет лучше понять функциональность системы, а также выявить возможные проблемы и улучшить ее производительность. Она является важным инструментом для разработки информационных систем и может быть использована в любой области деятельности, где необходимо описать функциональность системы (Рисунок 1).

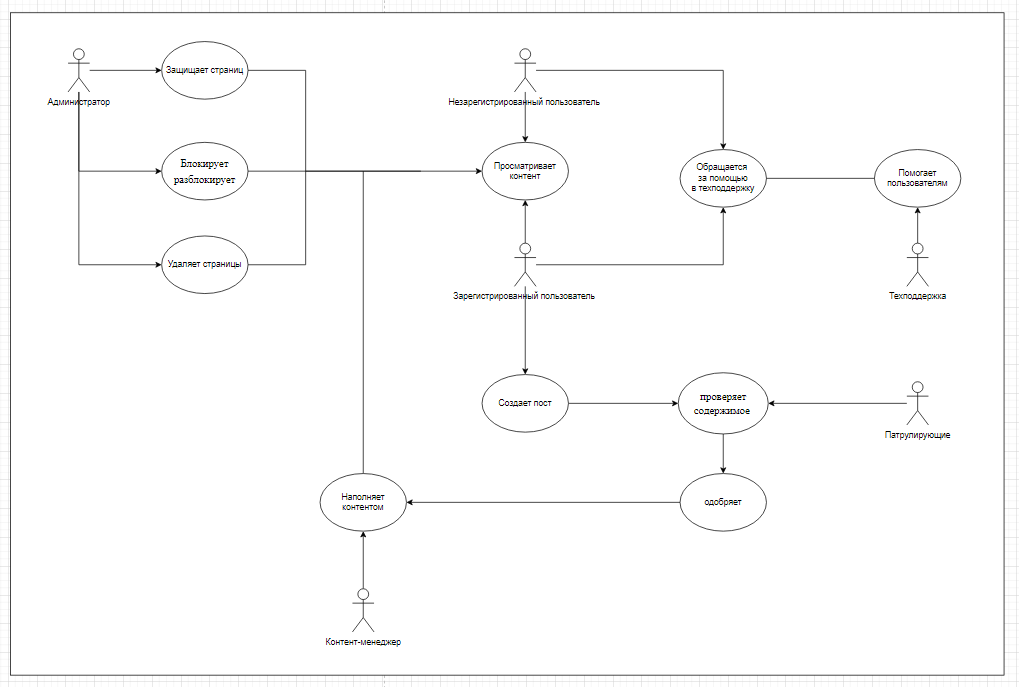


Рисунок 2 – Диаграмма прецедентов

Диаграмма деятельности UML – это графическое представление последовательности действий в процессе выполнения определенной задачи или процесса. Данная диаграмма позволяет описать порядок выполнения операций и принимаемые решения в процессе работы системы.

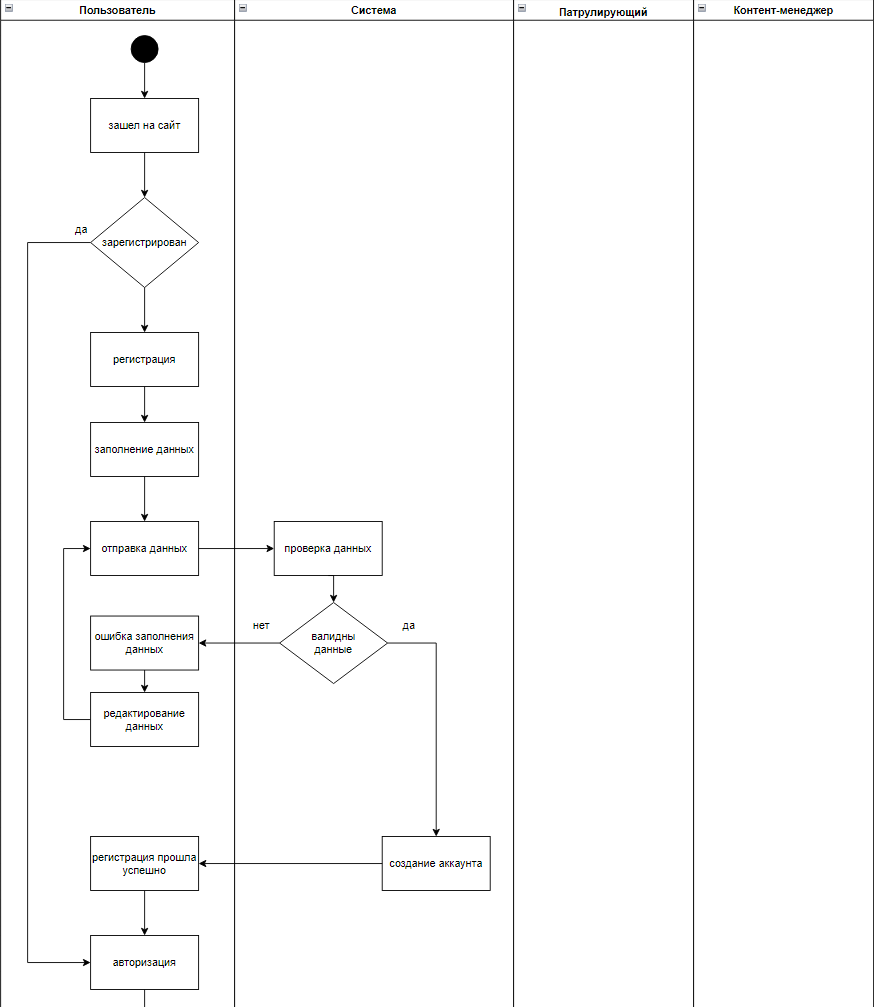
Диаграмма деятельности состоит из следующих элементов:

* Действия. Это операции, которые выполняются в процессе работы системы;
* Решения. Это условия, при которых выполняются определенные действия;
* Связи. Это отношения между действиями и решениями.

Для создания диаграммы деятельности необходимо выполнить следующие шаги:

* Определить цель и задачи. Необходимо определить цель и задачи, которые должна выполнять система;
* Определить действия. Необходимо определить все действия, которые должны быть выполнены в процессе работы системы;
* Определить решения. Необходимо определить все условия, при которых должны быть выполнены определенные действия;
* Связать действия и решения. Необходимо связать действия и решения между собой в определенном порядке выполнения;
* Проверить корректность диаграммы. Необходимо проверить корректность диаграммы деятельности, чтобы убедиться, что все действия и решения правильно описаны.

Диаграмма деятельности UML позволяет лучше понять порядок выполнения операций в процессе работы системы, а также выявить возможные проблемы и улучшить ее производительность. Она является важным инструментом для разработки информационных систем и может быть использована в любой области деятельности, где необходимо описать порядок выполнения операций (Рисунок 2**)**.



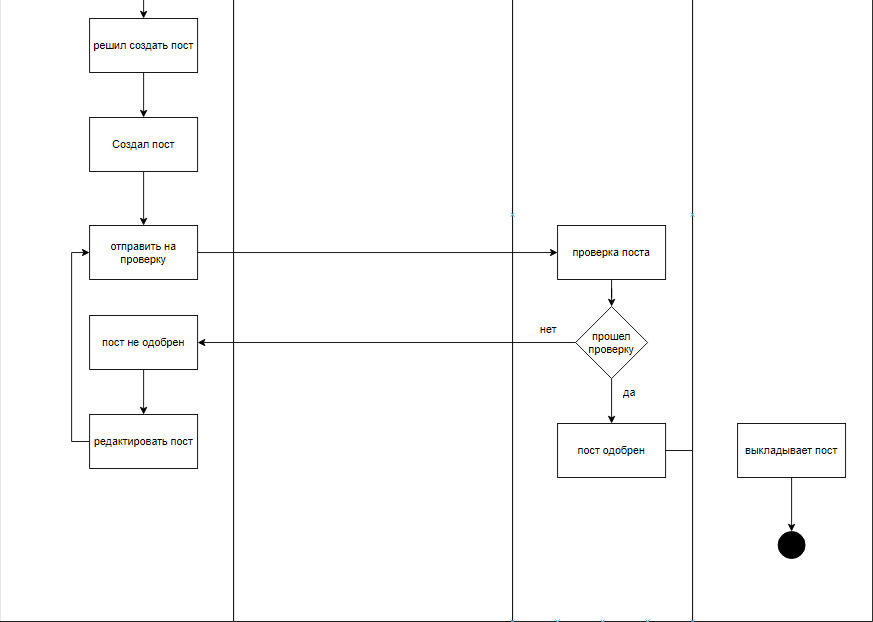


Рисунок 3 – Диаграмма деятельности

Диаграмма IDEF0 – это метод моделирования функциональных процессов, который используется для описания бизнес-процессов и систем. Она помогает разработчикам понять, как работает система и какие функции выполняются в процессе ее работы.

Диаграмма IDEF0 состоит из следующих элементов:

* Блоки функций. Это основные элементы диаграммы, которые представляют функции, выполняемые в процессе работы системы;
* Связи. Это отношения между блоками функций, которые показывают, какие функции зависят друг от друга и как они связаны в процессе работы системы;
* Входы и выходы. Это элементы, которые показывают, какие данные поступают в систему и какие данные она выдает;

Для создания диаграммы IDEF0 необходимо выполнить следующие шаги:

* Определить цель и задачи. Необходимо определить цель и задачи, которые должна выполнять система;
* Определить блоки функций. Необходимо определить все функции, которые должны быть выполнены в процессе работы системы;
* Определить связи. Необходимо определить отношения между блоками функций и показать, как они связаны между собой;
* Определить входы и выходы. Необходимо определить, какие данные поступают в систему и какие данные она выдает;
* Проверить корректность диаграммы. Необходимо проверить корректность диаграммы IDEF0, чтобы убедиться, что все функции и связи правильно описаны.

Диаграмма IDEF0 является важным инструментом для разработки информационных систем и может быть использована в любой области деятельности, где необходимо описать функциональные процессы. Она помогает разработчикам лучше понимать работу системы и выявлять возможные проблемы, что позволяет улучшить ее производительность. Вместе с диаграммой IDEF0 часто используется диаграмма UML, которая позволяет описывать процессы более детально и точно (Рисунок 3)..

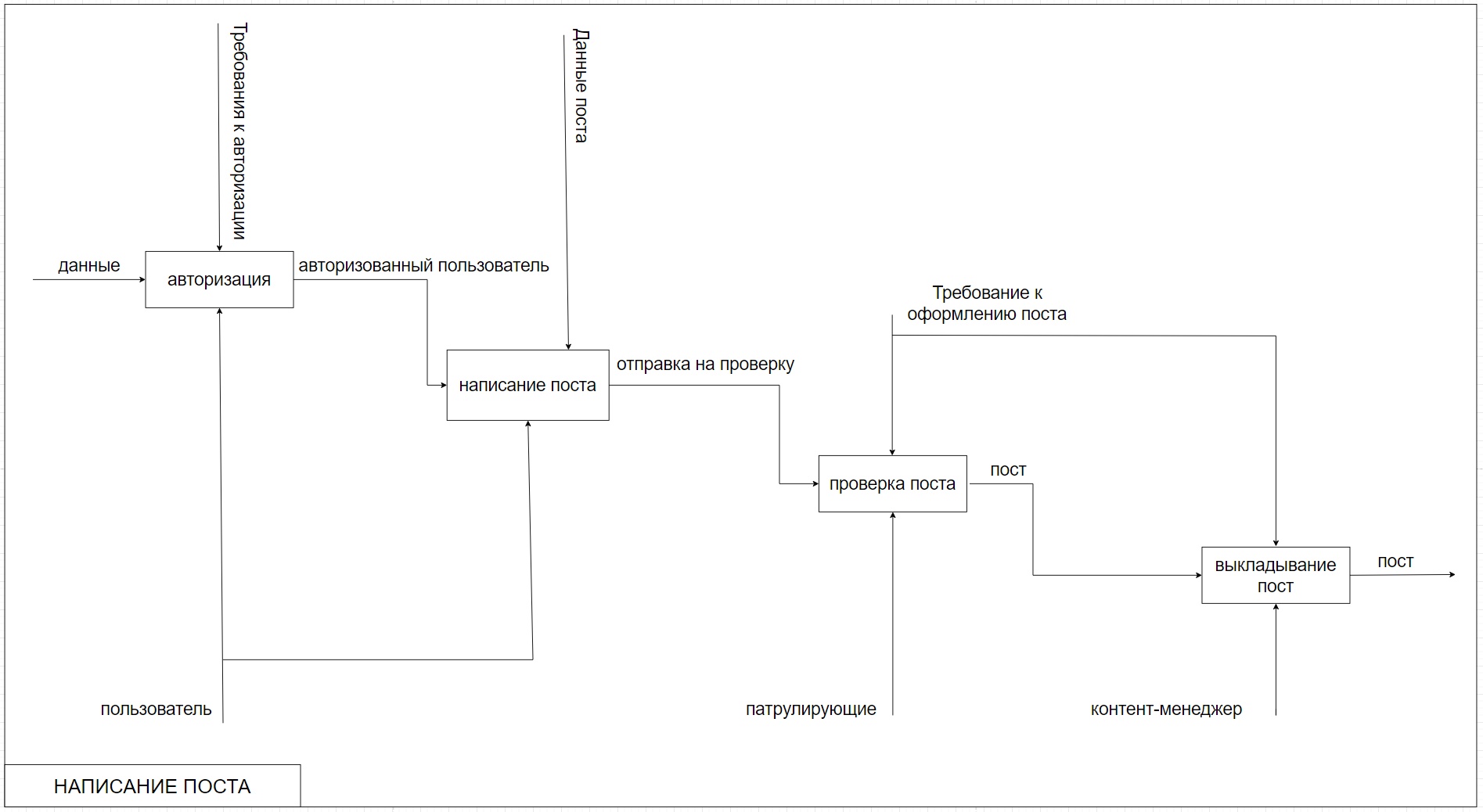


Рисунок 4 – Диаграмма IDEF0

Диаграмма последовательностей UML – это графическое представление взаимодействия объектов в системе в определенном порядке выполнения операций. Данная диаграмма позволяет описать последовательность действий между объектами и участниками системы.

Диаграмма последовательностей состоит из следующих элементов:

* Объекты. Это участники системы, которые выполняют определенные операции;
* Сообщения. Это действия, которые выполняют объекты в процессе взаимодействия;
* Линии жизни. Это временные интервалы, в которых объект выполняет определенные операции.

Для создания диаграммы последовательностей необходимо выполнить следующие шаги:

* Определить объекты. Необходимо определить все объекты, которые участвуют в процессе взаимодействия;
* Определить сообщения. Необходимо определить все сообщения, которые передаются между объектами;
* Определить порядок выполнения операций. Необходимо определить порядок выполнения операций между объектами;
* Проверить корректность диаграммы. Необходимо проверить корректность диаграммы последовательностей, чтобы убедиться, что все объекты и сообщения правильно описаны.

Диаграмма последовательностей UML позволяет лучше понять взаимодействие объектов в системе, а также выявить возможные проблемы и улучшить ее производительность. Она является важным инструментом для разработки информационных систем и может быть использована в любой области деятельности, где необходимо описать взаимодействие объектов (Рисунок 4)..

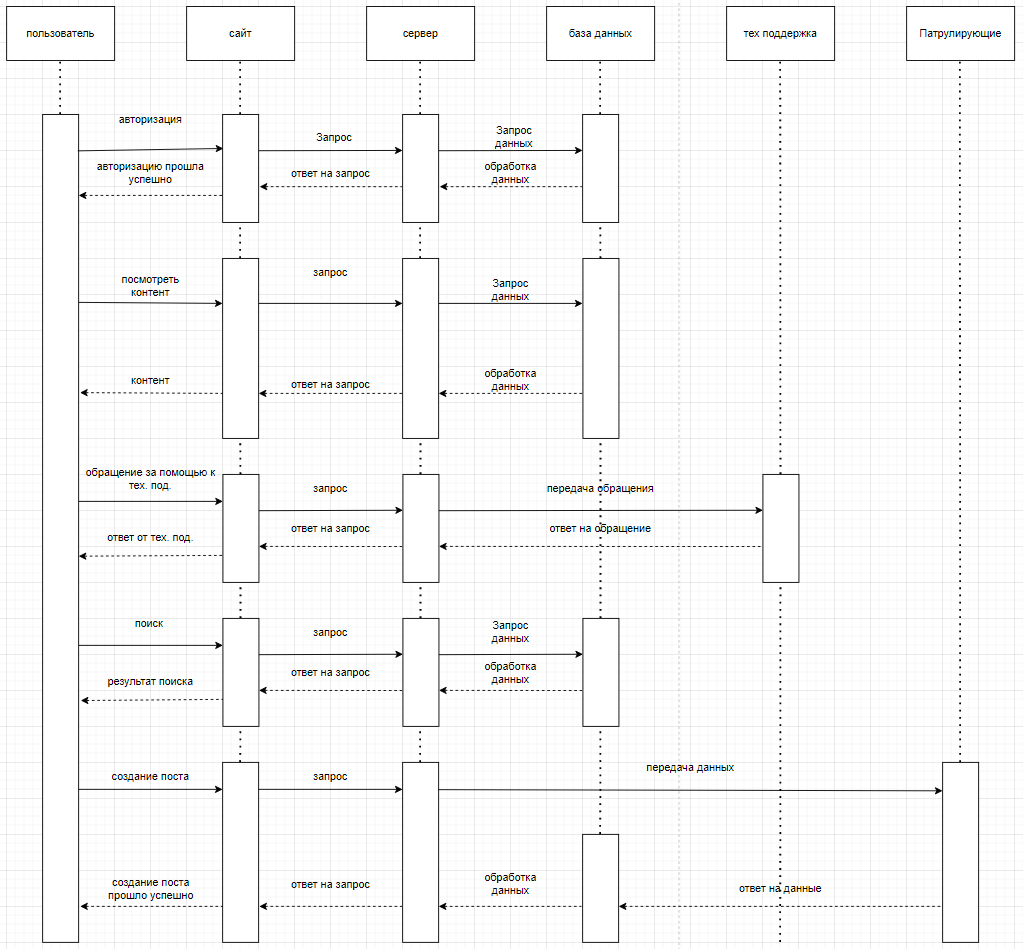


Рисунок 5 – Диаграмма последовательностей

## 1.3 Логическое и физическое проектирование базы данных

В современном мире базы данных являются неотъемлемой частью информационных технологий. Они используются практически во всех сферах деятельности, начиная от банковского сектора и заканчивая производственными предприятиями. Базы данных представляют собой совокупность данных, которые хранятся в определенном формате и обрабатываются с помощью специальных программных средств.

Одним из важных этапов создания базы данных является ее проектирование. Проектирование базы данных – это процесс создания ее структуры и определения отношений между различными элементами данных. При проектировании базы данных можно выделить два основных этапа: логическое и физическое проектирование.

Логическое проектирование базы данных – это этап, на котором определяются все сущности, атрибуты и связи между ними. На этом этапе создается модель данных, которая описывает все элементы базы данных и их взаимосвязь. В результате логического проектирования получается диаграмма, на которой отображены все сущности и связи между ними.

Физическое проектирование базы данных – это этап, на котором определяется физическая структура базы данных и ее хранение на диске. На этом этапе определяются таблицы, индексы, ключи и другие элементы, которые будут использоваться для хранения данных. В результате физического проектирования создается спецификация базы данных, которая содержит информацию о структуре базы данных и ее хранении.

Важно отметить, что логическое и физическое проектирование базы данных являются взаимосвязанными. Логическое проектирование определяет структуру базы данных и ее связи, а физическое проектирование определяет способ ее хранения. При проектировании базы данных необходимо учитывать требования к ее производительности, безопасности и масштабируемости.

В заключение можно сказать, что проектирование базы данных – это очень важный этап создания информационной системы. От правильности проектирования зависит эффективность и надежность работы системы. Логическое и физическое проектирование базы данных являются неотъемлемой частью этого процесса и должны выполняться в тесном взаимодействии друг с другом.

Входе разработки курсового проекта были составлены логическая и физическая модель будущего веб-сайта.

Структура таблицы dayweek:

id (int) - идентификатор дня недели (первичный ключ)

name (varchar(255)) - название дня недели

Таблица grade:

id (int) - идентификатор оценки (первичный ключ)

number (int) - число оценки

Таблица img:

id (int) - идентификатор изображения (первичный ключ)

path (varchar(255)) - путь к изображению

Таблица listimg:

id (int) - идентификатор (первичный ключ)

img\_id (int) - идентификатор изображения (внешний ключ, ссылается на таблицу img)

product\_menu\_id (int) - идентификатор меню продукта (внешний ключ, ссылается на таблицу productmenu)

review\_id (int) - идентификатор отзыва (внешний ключ, ссылается на таблицу review)

product\_id (int) - идентификатор продукта (внешний ключ, ссылается на таблицу product)

Таблица opentime:

id (int) - идентификатор (первичный ключ)

dayweek\_id (int) - идентификатор дня недели (внешний ключ, ссылается на таблицу dayweek)

time\_open (varchar(255)) - время открытия

time\_close (varchar(255)) - время закрытия

product\_id (int) - идентификатор продукта (внешний ключ, ссылается на таблицу product)

Таблица product:

id (int) - идентификатор продукта (первичный ключ)

title (varchar(255)) - название продукта

author\_id (int) - идентификатор автора (внешний ключ, ссылается на таблицу users)

description (text) - описание продукта

street (varchar(255)) – улица

phone (varchar(255)) – телефон

website (varchar(255)) - веб-сайт

gpsx (double) - координата X GPS

gpsy (double) - координата Y GPS

logo (varchar(255)) - логотип продукта

Таблица productmenu:

id (int) - идентификатор меню продукта (первичный ключ)

name (varchar(255)) - название меню

price (decimal(10,2)) - цена меню

product\_id (int) - идентификатор продукта (внешний ключ, ссылается на таблицу product)

Таблица producttag:

product\_id (int) - идентификатор продукта (внешний ключ, ссылается на таблицу product)

tag\_id (int) - идентификатор тега (внешний ключ, ссылается на таблицу tag)

Таблица review:

id (int) - идентификатор отзыва (первичный ключ)

user\_id (int) - идентификатор пользователя (внешний ключ, ссылается на таблицу users)

product\_id (int) - идентификатор продукта (внешний ключ, ссылается на таблицу product)

grade\_id (int) - идентификатор оценки (внешний ключ, ссылается на таблицу grade)

text (text) - текст отзыва

date (datetime) - дата отзыва

Таблица role:

id (varchar(255)) - идентификатор роли (первичный ключ)

description (varchar(255)) - описание роли

Таблица sex:

id (int) - идентификатор пола (первичный ключ)

name (varchar(50)) - название пола

Таблица tag:

id (int) - идентификатор тега (первичный ключ)

name (varchar(255)) - название тега

Таблица userrole:

* user\_id (int) - идентификатор пользователя (внешний ключ, ссылается на таблицу users)
* role\_id (varchar(255)) - идентификатор роли (внешний ключ, ссылается на таблицу role)

Таблица users:

* id (int) - идентификатор пользователя (первичный ключ)
* name\_first (varchar(50)) - имя пользователя
* name\_last (varchar(50)) - фамилия пользователя
* password (char(32)) - пароль пользователя
* sex (int) - идентификатор пола (внешний ключ, ссылается на таблицу sex)
* img (varchar(255)) - идентификатор изображения (внешний ключ, ссылается на таблицу img)
* birthdate (date) - дата рождения пользователя
* email (varchar(255)) - электронная почта пользователя

Модель данных предоставляет все необходимые сущности для управления мероприятиями и блогом. С помощью этой базы данных можно выполнять различные задачи, такие как добавление новых мероприятий, управление заказами и публикация постов блога (Рисунок 5).

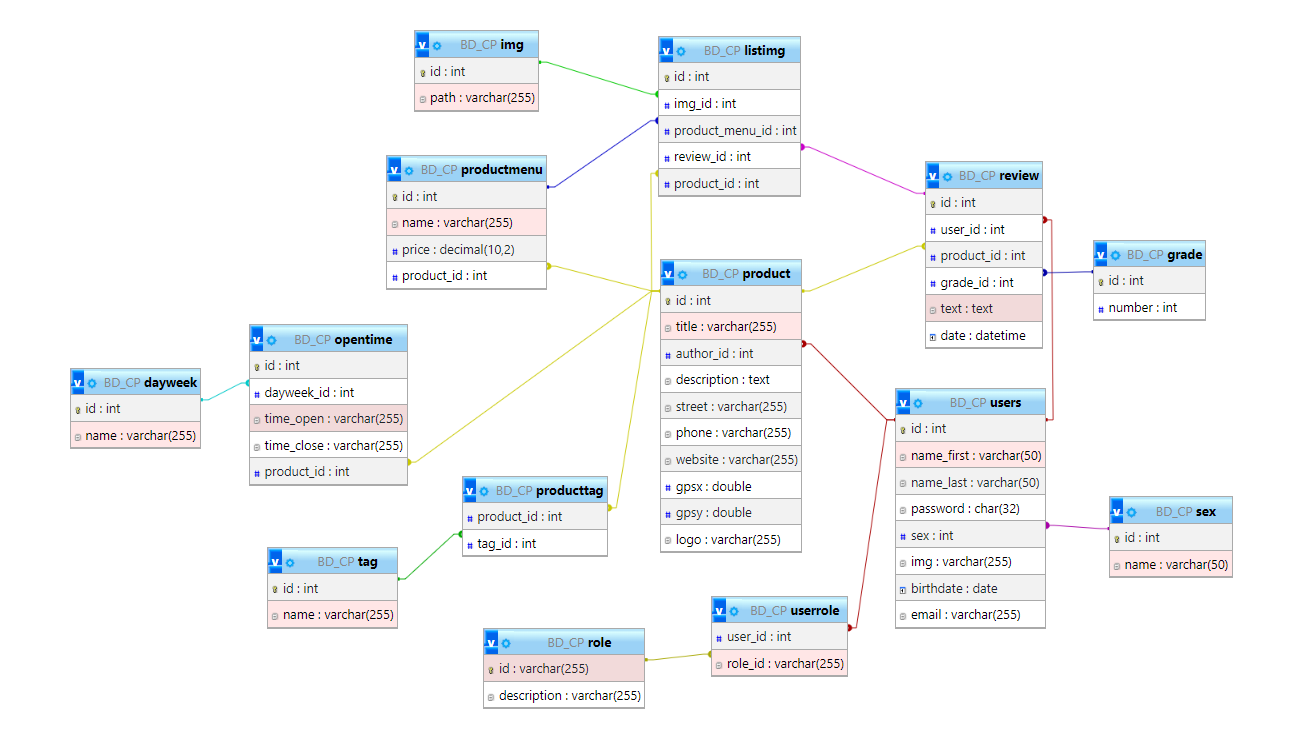


Рисунок 6 – Физическая модель базы данных

## 1.4 Выбор и обоснования средств разработки

При выборе средств разработки для создания веб-приложения, выбрал пал на комбинацию MySQL, PHP и Vue.js по следующим причинам:

* + - * MySQL - это одна из самых популярных реляционных баз данных, которая обладает высокой производительностью, надежностью и масштабируемостью. Она обеспечивает эффективное хранение и управление данными, что является важным для больших и сложных веб-приложений.
      * PHP - это один из самых распространенных языков программирования для веб-разработки. Он обладает широкими возможностями и гибкостью, позволяя разрабатывать различные типы веб-приложений, включая динамические и интерактивные. PHP также интегрируется хорошо с MySQL, что облегчает взаимодействие с базой данных.
      * Vue.js - это современный JavaScript-фреймворк для разработки пользовательских интерфейсов. Он обладает простым и интуитивно понятным синтаксисом, что упрощает разработку и поддержку кода. Vue.js также обеспечивает высокую скорость работы и отзывчивость пользовательского интерфейса, что является важным для удобного пользовательского опыта.

Комбинация MySQL, PHP и Vue.js обеспечивает мощный инструментарий для разработки веб-приложений. MySQL обеспечивает надежное хранение данных, PHP позволяет разрабатывать серверную логику и взаимодействовать с базой данных, а Vue.js обеспечивает разработку интерактивного пользовательского интерфейса. Эти инструменты хорошо интегрируются друг с другом, что облегчает разработку и поддержку приложения.

# 2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

Информационные системы - это комплексное программно-аппаратное обеспечение, предназначенное для сбора, хранения, обработки, передачи и использования информации. Они широко применяются в различных сферах и отраслях: бизнесе, образовании, здравоохранении, государственном управлении и т.д.

Одной из главных функций информационной системы является автоматизация бизнес-процессов. Она позволяет оптимизировать рабочие процессы, ускорить принятие решений, повысить эффективность деятельности предприятия. Например, системы управления ресурсами предоставляют возможность контролировать запасы, управлять комплектацией заказов, отслеживать исполнение планов производства.

Важным элементом информационных систем является база данных - хранилище информации, которое обеспечивает быстрый доступ к нужным данным. Базы данных позволяют хранить, организовывать и обрабатывать информацию различного типа: тексты, числа, изображения и т.д. Кроме того, современные системы обладают мощными средствами для анализа и визуализации данных, что помогает в принятии стратегических решений.

Также информационные системы позволяют организовать электронное взаимодействие между различными участниками процесса. Интернет и сетевые технологии позволяют передавать информацию в режиме реального времени, обеспечивать удаленный доступ к данным, а также взаимодействовать с клиентами, поставщиками и партнерами через электронную почту, социальные сети и прочие каналы коммуникации.

Однако, необходимо учитывать и риски, связанные с использованием информационных систем. К таким рискам относятся возможность несанкционированного доступа к данным, утечка информации, а также возможность сбоев и отказов системы. Поэтому безопасность данных и информационных систем является важным аспектом.

Современные информационные системы постоянно развиваются и усовершенствуются. Новые технологии, такие как искусственный интеллект, большие данные, нейронные сети и блокчейн, открывают новые возможности для анализа и использования информации. Они помогают предсказывать тренды, оптимизировать процессы и создавать инновационные решения.

Информационные системы играют важную роль в современном мире, обеспечивая эффективное использование информации и помогая организациям достигать своих целей. Правильное использование информационных систем позволяет повысить конкурентоспособность, улучшить качество услуг и управления, а также создать новые возможности для развития.

## 2.1 Разработка базы данных

Создание таблицы «Дни недели» Характеристики таблицы представлены в таблице 1

CREATE TABLE dayweek (  
 id int NOT NULL,  
 name varchar(255) NOT NULL)

Таблица 1 Характеристики полей таблицы «dayweek»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор дня недели (первичный ключ) |
| name | varchar(255) | название дня недели |

Создание таблицы «Оценка» Характеристики таблицы представлены в таблице 2

CREATE TABLE grade (  
 id int NOT NULL,  
 number int NOT NULL)

Таблица 2 Характеристики полей таблицы «grade»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор оценки (первичный ключ) |
| number | int | число оценки |

Создание таблицы «Изображение» Характеристики таблицы представлены в таблице 3

CREATE TABLE img (  
 id int NOT NULL,  
 path varchar(255) NOT NULL)

Таблица 3 Характеристики полей таблицы «img»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор изображения (первичный ключ) |
| path | varchar(255) | путь к изображению |

Создание таблицы «Список изображений» Характеристики таблицы представлены в таблице 4

CREATE TABLE listimg (  
 id int NOT NULL,  
 img\_id int NOT NULL,  
 product\_menu\_id int DEFAULT NULL,  
 review\_id int DEFAULT NULL,  
 product\_id int DEFAULT NULL)

Таблица 4 Характеристики полей таблицы «listimg»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор (первичный ключ) |
| img\_id | int | идентификатор изображения (внешний ключ, ссылается на таблицу img) |
| product\_menu\_id | int | идентификатор меню продукта (внешний ключ, ссылается на таблицу productmenu) |
| review\_id | int | идентификатор отзыва (внешний ключ, ссылается на таблицу review) |
| product\_id | int | идентификатор продукта (внешний ключ, ссылается на таблицу product) |

Создание таблицы «Время работы» Характеристики таблицы представлены в таблице 5

CREATE TABLE opentime (  
 id int NOT NULL,  
 dayweek\_id int NOT NULL,  
 time\_open varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 time\_close varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 product\_id int NOT NULL)

Таблица 5 Характеристики полей таблицы «opentime»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор (первичный ключ) |
| dayweek\_id | int | идентификатор дня недели (внешний ключ, ссылается на таблицу dayweek) |
| time\_open | varchar(255) | время открытия |
| time\_close | varchar(255) | время закрытия |
| product\_id | int | идентификатор продукта (внешний ключ, ссылается на таблицу product) |

Создание таблицы «Продукты» Характеристики таблицы представлены в таблице 6

CREATE TABLE product (  
 id int NOT NULL,  
 title varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 author\_id int NOT NULL,  
 description text CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 street varchar(255) NOT NULL,  
 phone varchar(255) NOT NULL,  
 website varchar(255) NOT NULL,  
 gpsx double NOT NULL,  
 gpsy double NOT NULL,  
 logo varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL)

Таблица 6 Характеристики полей таблицы «product»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор продукта (первичный ключ) |
| title | varchar(255) | название продукта |
| author\_id | int | идентификатор автора (внешний ключ, ссылается на таблицу users) |
| description | text | описание продукта |
| street | varchar(255) | улица |
| phone | varchar(255) | телефон |
| website | varchar(255) | веб-сайт |
| gpsx | double | координата X GPS |
| gpsy | double | координата Y GPS |
| logo | varchar(255) | логотип продукта |

Создание таблицы «Меню» Характеристики таблицы представлены в таблице 7

CREATE TABLE productmenu (  
 id int NOT NULL,  
 name varchar(255) NOT NULL,  
 price decimal(10,2) NOT NULL,  
 product\_id int NOT NULL)

Таблица 7 Характеристики полей таблицы «productmenu»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор меню продукта (первичный ключ) |
| name | varchar(255) | название меню |
| price | decimal(10,2) | цена меню |
| product\_id | int | идентификатор продукта (внешний ключ, ссылается на таблицу product) |

Создание таблицы «Список тегов» Характеристики таблицы представлены в таблице 8

CREATE TABLE producttag (  
 product\_id int NOT NULL,  
 tag\_id int NOT NULL)

Таблица 8 Характеристики полей таблицы «producttag»

|  |  |  |
| --- | --- | --- |
| product\_id | int | идентификатор продукта (внешний ключ, ссылается на таблицу product) |
| tag\_id | int | идентификатор тега (внешний ключ, ссылается на таблицу tag) |

Создание таблицы «Отзыв» Характеристики таблицы представлены в таблице 9

CREATE TABLE review (  
 id int NOT NULL,  
 user\_id int NOT NULL,  
 product\_id int NOT NULL,  
 grade\_id int NOT NULL,  
 text text NOT NULL,  
 date datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP)

Таблица 9 Характеристики полей таблицы «review»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор отзыва (первичный ключ) |
| user\_id | int | идентификатор пользователя (внешний ключ, ссылается на таблицу users) |
| product\_id | int | идентификатор продукта (внешний ключ, ссылается на таблицу product) |
| grade\_id | int | идентификатор оценки (внешний ключ, ссылается на таблицу grade) |
| text | text | текст отзыва |
| date | date | дата отзыва |

Создание таблицы «роль» Характеристики таблицы представлены в таблице 10

CREATE TABLE role (  
 id varchar(255) NOT NULL,  
 description varchar(255) NOT NULL)

Таблица 10 Характеристики полей таблицы «role»

|  |  |  |
| --- | --- | --- |
| id | varchar(255) | идентификатор роли (первичный ключ) |
| description | varchar(255) | описание роли |

Создание таблицы «Пол» Характеристики таблицы представлены в таблице 11

CREATE TABLE sex (  
 id int NOT NULL,  
 name varchar(50) NOT NULL)

Таблица 11 Характеристики полей таблицы «sex»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор пола (первичный ключ) |
| name | varchar(50) | название пола |

Создание таблицы «Тег» Характеристики таблицы представлены в таблице 12

CREATE TABLE tag (  
 id int NOT NULL,  
 name varchar(255) NOT NULL)

Таблица 12 Характеристики полей таблицы «tag»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор тега (первичный ключ) |
| name | varchar(255) | название тега |

Создание таблицы «Список ролей» Характеристики таблицы представлены в таблице 13

CREATE TABLE userrole (  
 user\_id int NOT NULL,  
 role\_id varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL)

Таблица 13 Характеристики полей таблицы «userrole»

|  |  |  |
| --- | --- | --- |
| user\_id | int | идентификатор пользователя (внешний ключ, ссылается на таблицу users) |
| role\_id | varchar(255) | идентификатор роли (внешний ключ, ссылается на таблицу role) |

Создание таблицы «Пользователь» Характеристики таблицы представлены в таблице 14

CREATE TABLE users (  
 id int NOT NULL,  
 name\_first varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 name\_last varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 password char(32) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 sex int NOT NULL,  
 img varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,  
 birthdate date NOT NULL,  
 email varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL)

Таблица 14 Характеристики полей таблицы «users»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор пользователя (первичный ключ) |
| name\_first | varchar(50) | имя пользователя |
| name\_last | varchar(50) | фамилия пользователя |
| password | char(32) | пароль пользователя |
| sex | int | идентификатор пола (внешний ключ, ссылается на таблицу sex) |
| img | varchar(255) | идентификатор изображения (внешний ключ, ссылается на таблицу img) |
| birthdate | date | дата рождения пользователя |
| email | varchar(255) | электронная почта пользователя |

## 2.2 Логическая и физическая структура сайта

Логическая и физическая структура сайта являются основными составляющими успешного веб-проекта. Обе структуры взаимодействуют между собой и играют важную роль в удобстве использования сайта для пользователей.

Логическая структура сайта определяет его организацию и иерархию контента. Она представляет собой дерево, где каждая ветвь соответствует определенной категории или подкатегории. Логическая структура позволяет пользователю быстро найти нужную информацию и легко перемещаться по сайту.

Правильное построение логической структуры требует тщательного планирования. Ключевыми элементами являются объединение похожих страниц в одну категорию, создание навигационной панели с ссылками на основные разделы, использование подкатегорий и тегов для дальнейшей сортировки контента.

Single page application - это подход в разработке веб-приложений, который предполагает создание одностраничного приложения, в котором весь контент загружается один раз и меняется динамически без перезагрузки страницы. Single page application основывается на принципе клиент-серверной архитектуры, где клиентская часть приложения выполняется на стороне браузера пользователя.

Одной из важных особенностей Single page application является его логика работы. В Single page application логика представляет собой управление отображением и обновлением контента на одной странице приложения. Традиционные веб-приложения работают на принципе перехода между страницами с помощью перезагрузки, в то время как Single page application позволяет пользователю взаимодействовать с различными частями приложения без перезагрузки страницы.

Одним из основных инструментов для реализации логики Single page application является JavaScript и его фреймворки, такие как Angular, React и Vue.js. Они предоставляют множество инструментов для управления состоянием приложения, маршрутизацией загрузкой данных с сервера и реактивным обновлением контента на странице.

Single page application логика также включает в себя механизмы маршрутизации, которые позволяют управлять переходами между разделами приложения и поддерживать историю посещенных страниц. Маршрутизация позволяет создавать более динамичные и гибкие интерфейсы, а также лучшую навигацию и пользовательский опыт.

Single page application логика является важной составляющей при разработке одностраничных приложений. Она позволяет создавать более интерактивные, быстрые и отзывчивые интерфейсы, а также улучшает пользовательский опыт. Правильное применение Single page application логики требует учета требований приложения, выбора подходящих технологий и правильной архитектуры приложения.

Физическая структура сайта определяет способ организации и хранения файлов на сервере. Она включает в себя размещение файлов по папкам, кодирование URL-адресов, наличие платформы управления контентом и т.д (Рисунок 6).Корректная физическая структура облегчает индексацию сайта поисковыми системами, ускоряет загрузку страниц и обеспечивает лучшую безопасность.

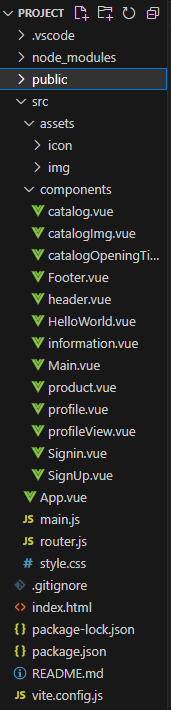


Рисунок 7 – Физическая структура сайта

Важными принципами построения физической структуры являются чистота и структурированность. Рекомендуется использовать дружественные URL-адреса, чтобы они были осмысленными для пользователей и поисковых роботов. Кроме того, файлы должны быть организованы в логические группы и расположены в соответствующих папках.

Обе структуры важны для успеха веб-сайта. Логическая структура обеспечивает удобство использования и легкость навигации, а физическая структура влияет на поисковую оптимизацию и общую производительность сайта. Правильное построение этих структур поможет создать удобное и привлекательное веб-пространство для посетителей сайта.

## 2.3 Разработка интерфейса пользователя

Интерфейс пользователя является ключевым элементом любой программы, веб-сайта или приложения. Он обеспечивает взаимодействие между пользователем и системой, позволяя пользователям выполнять различные операции и получать нужную информацию.

Качество и удобство интерфейса пользователя играют важнейшую роль в определении уровня удовлетворенности пользователей. Что же делает хороший интерфейс?

Во-первых, он должен быть интуитивно понятным. Пользователь должен легко понимать, как использовать интерфейс и выполнять нужные действия без особых усилий. Отображение информации и доступ к функционалу должны быть логичными и простыми.

Во-вторых, интерфейс должен быть информативным. Вся необходимая информация должна быть предоставлена пользователю в понятном и доступном виде. Важные элементы интерфейса, такие как кнопки, ссылки и подсказки, должны быть ясно видимыми и четко описанными.

В-третьих, интерфейс пользователя должен быть эстетически приятным и привлекательным. Дизайн должен соответствовать общему стилю и образу компании или бренда. Цветовая гамма, шрифты и графические элементы должны гармонично сочетаться и быть привлекательными для глаз пользователя.

В-четвертых, интерфейс должен быть адаптивным и отзывчивым. Он должен быть способен адаптироваться к различным устройствам и экранам, чтобы обеспечить комфортное использование на мобильных телефонах, планшетах и компьютерах. Кроме того, интерфейс должен реагировать на действия пользователя быстро и без задержек.

В-пятых, интерфейс пользователя должен обеспечивать хорошую функциональность и навигацию. Пользователь должен легко находить нужные функции и возможности, а также иметь возможность легко возвращаться к предыдущим шагам. Навигационное меню и разделение информации на логические блоки помогают пользователям легко перемещаться по интерфейсу и быстро находить нужное.

Хороший интерфейс пользователя должен постоянно обновляться и улучшаться на основе обратной связи от пользователей. Слушание и анализ отзывов пользователей позволяет выявлять недостатки интерфейса и вносить изменения, чтобы сделать его еще более удобным и функциональным.

Интерфейс пользователя является важным элементом любой системы, который влияет на ее успешность и эффективность. Создание хорошего интерфейса требует тщательного анализа требований и потребностей пользователей, а также применения лучших практик в области дизайна и удобства использования.

Разработка интерфейса пользователя является одной из важных задач в сфере создания программного обеспечения. UI представляет собой то, через что пользователи взаимодействуют с приложением или веб-сайтом, поэтому его дизайн и удобство использования имеют большое значение (Рис.7).

Перед началом разработки интерфейса пользователя необходимо провести исследование и анализ целевой аудитории. Это поможет понять, какие функции необходимо предоставить пользователям, какие проблемы они могут испытывать и как сделать интерфейс максимально интуитивным и удобным в использовании.

Далее начинается проектирование интерфейса. Оно включает разработку визуальной составляющей, такой как цветовая гамма, типографика, иконки и графика. Учитывается анимация и стилизация элементов интерфейса, которые помогают пользователям взаимодействовать с приложением.

Разработка интерфейса - это не только визуальные аспекты. Важным является также разработка архитектуры интерфейса, которая определяет, какие элементы будут включены в интерфейс и как они будут располагаться на странице или экране. Это включает в себя размещение кнопок, форм, меню и других элементов, которые пользователи будут использовать при взаимодействии с приложением.

Одной из важных задач при разработке интерфейса пользователя является тестирование и улучшение. В процессе разработки необходимо проводить тестирование с реальными пользователями, чтобы выявить проблемы и недостатки интерфейса. На основе обратной связи пользователей можно вносить изменения и улучшать интерфейс, чтобы сделать его более удобным и интуитивным.

В современных технологиях разработки пользовательских интерфейсов широко используется методика адаптивного дизайна. Это позволяет создавать интерфейсы, которые оптимально отображаются на различных устройствах и экранах, таких как компьютеры, смартфоны и планшеты.

Разработка интерфейса пользователя является неотъемлемой частью процесса создания программного обеспечения. Она включает в себя не только визуальные аспекты, но и архитектуру и функциональность. Качественный и удобный интерфейс помогает пользователям эффективно взаимодействовать с приложением и достигать своих целей.

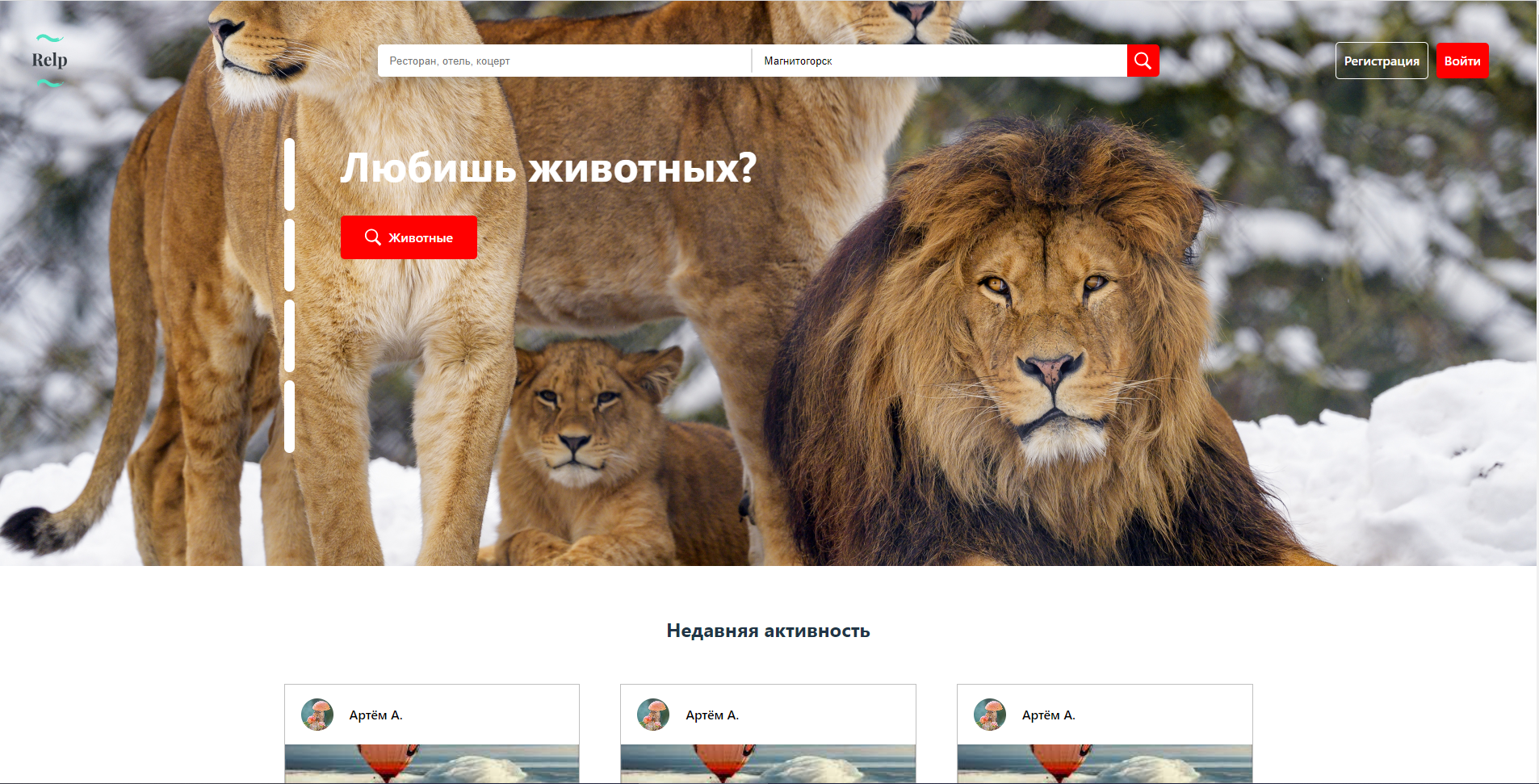


Рисунок 8 – Интерфейс на примере курсового сайта

## 2.4 Разработка запросов

Разработка запросов – важный этап в процессе создания и управления базами данных. Запросы представляют собой инструкции для извлечения, модификации или удаления данных из базы данных. Разработка эффективных запросов позволяет получить нужную информацию из базы данных быстро и точно.

Первым шагом в разработке запросов является определение целей и требований, которые необходимо удовлетворить. Например, какую информацию нужно извлечь, какие данные изменить или удалить. Нужно также определить условия и критерии для выборки данных.

<?php

    include './connect.php';

        header("Access-Control-Allow-Origin: \*");

        header("Access-Control-Allow-Methods: POST, GET, OPTIONS");

        header("Access-Control-Allow-Headers: Content-Type");

    try {

        $sql = "SELECT p.id AS product\_id, p.title AS product\_title,

        p.description AS product\_description,

        SUM(g.count) AS product\_grade\_count,

        (SELECT GROUP\_CONCAT(pg.id SEPARATOR '|') FROM productgrade AS pg  WHERE pg.product\_id=p.id) AS product\_grade\_id,

        (SELECT GROUP\_CONCAT(t.name SEPARATOR '|') FROM producttag AS pt JOIN tag AS t ON t.id=pt.tag\_id WHERE pt.product\_id=p.id) AS product\_tag,

        (SELECT GROUP\_CONCAT(i.path SEPARATOR '|') FROM listimg AS l JOIN img AS i ON i.id=l.img\_id WHERE l.product\_id=p.id) AS product\_img

        FROM product AS p

        JOIN status AS s ON p.status\_id=s.id

        JOIN productgrade AS g ON g.product\_id=p.id

        WHERE s.name='approved'

        GROUP BY p.id";

        $result = mysqli\_query($conn, $sql);

        $data = mysqli\_fetch\_all($result, MYSQLI\_ASSOC);

        $grade=0;

        foreach ($data as &$item) {

            $gradeid= explode('|', $item['product\_grade\_id']);

            $item['product\_grade\_id'] = $gradeid;

            foreach($item['product\_grade\_id'] as $num){

                $sql="SELECT (g.number\*pg.count) as sum FROM productgrade AS pg JOIN grade as g ON pg.grade\_id=g.id WHERE pg.id=$num";

                $result = mysqli\_query($conn, $sql);

                $fr=mysqli\_fetch\_assoc($result);

                $grade+=$fr['sum'];

            }

            $item['product\_grade'] = round($grade/$item['product\_grade\_count']);

            $tags = explode('|', $item['product\_tag']);

            $item['product\_tag'] = $tags;

            $imgs = explode('|', $item['product\_img']);

            $item['product\_img'] = $imgs;

            $openingTimes = [];

            $sql = "SELECT d.name AS day, o.time\_open AS opening, o.time\_close AS closing

                    FROM opentime o

                    JOIN dayweek d ON d.id=o.dayweek\_id

                    WHERE o.product\_id=".$item['product\_id'];

            $result = mysqli\_query($conn, $sql);

            while ($row = mysqli\_fetch\_assoc($result)) {

                $openingTimes[] = $row;

            }

            $item['openingTimes'] = $openingTimes;

            unset($item['product\_grade\_id']);

        }

        echo json\_encode($data, JSON\_NUMERIC\_CHECK);

    } catch(Exception $e) {

        echo "Ошибка при выполнении запроса: " . $e->getMessage();

    }

    mysqli\_close($conn);

?>

## 2.5 Разграничение прав доступа

Разграничение прав доступа – это процесс установления и контроля различных уровней доступа к информации, ресурсам или функционалу системы, с целью обеспечения безопасности данных и предотвращения несанкционированного доступа.

Одним из ключевых аспектов разграничения прав доступа является принцип наименьших привилегий, который предполагает, что каждому пользователю или субъекту должны быть предоставлены только те права и привилегии, которые необходимы для выполнения его задачи или функции. Это означает, что пользователи не должны иметь доступа к данным или ресурсам, которые не относятся к их задаче или ответственности. Этот принцип помогает уменьшить риски несанкционированного доступа и возможные угрозы безопасности.

Для реализации разграничения прав доступа используются различные методы и механизмы, такие как идентификация и аутентификация пользователей, системы управления доступом, многоуровневая система безопасности, системы ролевого управления доступом и другие.

Идентификация и аутентификация пользователей – это первый шаг в установлении разграничения прав доступа. Пользователь должен быть идентифицирован, чтобы система могла определить его права доступа. Аутентификация осуществляется путем проверки подлинности пользовательских учетных данных (Рис).

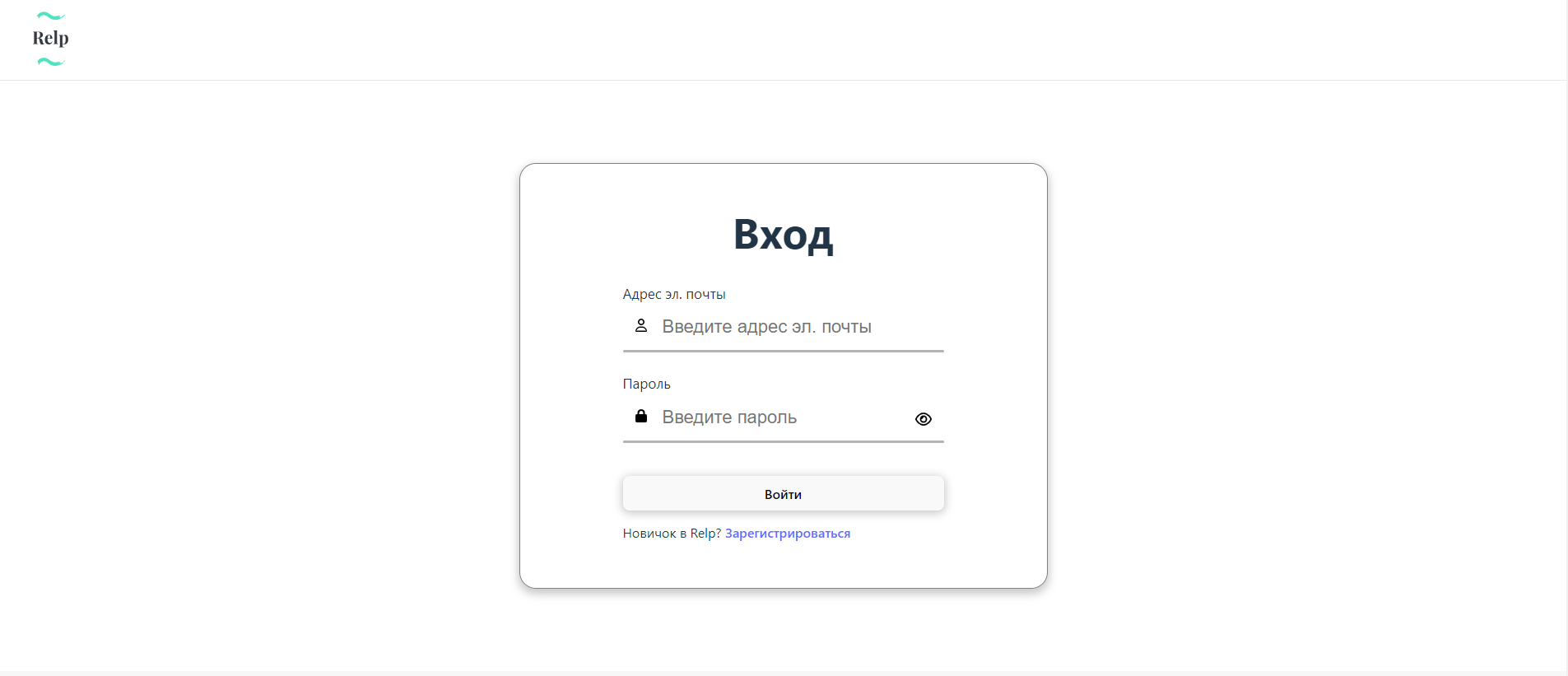


Рисунок 9 - Форма авторизации

Системы управления доступом позволяют установить и контролировать права доступа на уровне ресурса, объекта или функционала системы. Эти системы обычно осуществляются через список учетных записей с правами доступа или через определение прав доступа на уровне ролей или групп пользователей.

Системы ролевого управления доступом определяют права доступа на основе пользовательских ролей или групп. Каждой роли назначаются определенные права и привилегии, и пользователь получает эти права при назначении ему этой роли. Это упрощает управление правами доступа и снижает риск ошибок или несанкционированного доступа.

Разграничение прав доступа является неотъемлемой частью разработки и эксплуатации информационных систем, поскольку оно обеспечивает безопасность данных и защиту от внешних и внутренних угроз. Корректное и эффективное разграничение прав доступа позволяет предотвратить несанкционированный доступ, минимизировать риски и обеспечивать конфиденциальность и целостность информации.

## 2.6 Руководство пользователя

"Инструкция по использованию программы - это краткое руководство, предназначенное для пользователей программы, чтобы они могли корректно и эффективно использовать все ее функции и возможности.

В инструкции пользователю должна содержаться всесторонняя и пошаговая информация, которая поможет пользователю понять основную функциональность программы и научиться работать с ней без необходимости обращаться к дополнительным материалам или спецификациям. Вся необходимая информация для этого извлекается из полной документации, но представляется в удобной форме пользователю.

Первая часть инструкции - описательная. В ней должны присутствовать следующие пункты:

* Наименование программы: указывается название программы, чтобы пользователь мог легко ее идентифицировать.
* Краткое описание программы: дается краткое описание основных функций программы и ее назначения.
* Перечень выполняемых функций: перечисляются все функции, которые может выполнять программа, чтобы пользователь знал, какие возможности у него есть.
* Характеристика метода решения задачи: предоставляется краткое описание метода (или методов), которые использовались для решения поставленной задачи, а также указываются их достоинства и недостатки.
* Библиографическая ссылка: указывается полная библиографическая ссылка на описание метода, чтобы пользователь мог ознакомиться с ним подробнее, если потребуется.
* Входные и выходные данные: описываются данные, которые программа принимает на входе и данные, которые она выдает на выходе.
* Структура базы данных: если программа использует базу данных, то описывается структура базы данных и все ее таблицы.

Вторая часть инструкции должна содержать описание процесса работы с программой. В ней должны присутствовать следующие пункты:

* Запуск программы: описывается, как запустить программу, чтобы пользователь мог начать работу с ней.
* Режимы работы программы: описываются все режимы работы программы, чтобы пользователь мог выбрать нужный.
* Печати и диагностические сообщения: описываются все печати и сообщения, которые могут появляться в процессе работы программы, чтобы пользователь мог корректно на них реагировать.
* - Подготовка и ввод данных: описывается, как подготовить и ввести начальные данные в программу.
* Реакция на запросы программы: описывается, как пользователь должен реагировать на запросы программы, чтобы получить желаемый результат.
* Работа в исключительных ситуациях: описывается, как пользователь должен действовать в случае возникновения исключительных ситуаций или ошибок.
* Восстановление работы программы: описывается, как восстановить работу программы в случае ее аварийного завершения.
* Получение требуемого результата: объясняется, как получить конечный результат работы с программой.
* Правильное завершение работы с программой: описывается, как корректно закончить работу с программой.
* Другие сведения: предоставляются дополнительные сведения, которые могут быть полезными пользователю во время работы с программой.

Инструкция пользователю представляет собой независимый документ, содержащий всю необходимую и понятную информацию для пользователя программы, чтобы он мог успешно и эффективно ею пользоваться.

Руководство для обычного пользователя

При посещении веб-сайта обычным пользователем возникает некоторое количество ограничений. Например, неавторизованный пользователь не имеет возможности отправить заявку на создание свое странички продукта или оставить свой отзыв на сайте. Также у него отсутствует доступ к личному кабинету, поскольку учетной записи просто не существует. Чтобы получить возможность зарегистрироваться или войти в систему, необходимо нажать на соответствующую кнопку "Войти" или "Регистрация" в меню сайта (Рисунок 10).

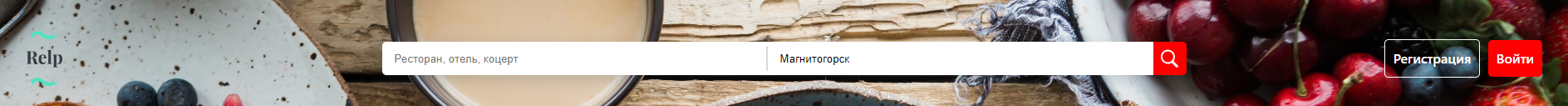


Рисунок 10– Кнопки для входа в систему

После того, как пользователь зарегистрировался, он может зайти в личный кабинет (Рисунок 11)

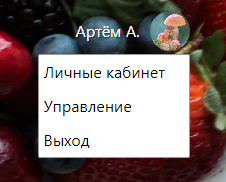


Рисунок 11 – Личный кабинет

Функционал во вкладке “личном кабинете” представляет собой набор возможностей, предоставляемых пользователю для удобного управления своим личными данными. Такой функционал дает пользователю доступ к различным инструментам, позволяющим контролировать свои данные и обновлять персональную информацию.

Функционал во вкладке “Управление” представляет собой набор возможностей, предоставляемых пользователю для удобного управления своими продуктами. Такой функционал дает пользователю доступ к изменению своих продуктов и их мониторинг.

## 2.7 Тестирование информационной системы

Тестирование информационной системы является важной и неотъемлемой частью процесса ее разработки и внедрения. Оно выполняет роль проверки работоспособности и соответствия системы заявленным требованиям и ожиданиям пользователей.

Перед началом тестирования информационной системы необходимо разработать план тестирования, который определит последовательность и объем работы, а также ресурсы, необходимые для проведения тестирования. В этом плане должны быть определены тестовые сценарии, критерии успеха и ошибок, а также ответственные за выполнение и анализ результатов.

Одной из основных задач тестирования информационной системы является проверка функциональности системы. Это включает тестирование работы отдельных функций и модулей системы, а также проверку их взаимодействия друг с другом. Для этого используются различные методы и техники тестирования, такие как тестирование черного ящика, тестирование белого ящика, тестирование пограничных значений и т.д.

Однако функциональное тестирование не является единственным аспектом тестирования информационной системы. Также проводится тестирование производительности системы, которое включает проверку ее работы при большой загрузке и в неблагоприятных условиях. Тестирование безопасности информационной системы является также важной составляющей, особенно если система содержит конфиденциальную информацию пользователей.

В процессе тестирования информационной системы необходимо учитывать множество факторов, таких как различные варианты использования системы, поведение системы при ошибочных действиях пользователей, совместимость системы с другими программными и аппаратными средствами, а также ее надежность и устойчивость к сбоям.

Оценка результатов тестирования информационной системы позволяет выявить ошибки и недостатки в ее работе, а также принять меры по их устранению и улучшению системы. Регулярное тестирование позволяет поддерживать систему в рабочем состоянии и обеспечивать ее высокое качество.

Таким образом, тестирование информационной системы играет важную роль в обеспечении ее работоспособности, эффективности, безопасности и соответствия требованиям пользователей. Оно помогает выявить и устранить ошибки и недостатки системы, обеспечивая ее стабильное и надежное функционирование.

Таблица 15 - Тестирование методом «черного ящика»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер | Назначение теста | Ожидаемый результат | Фактический результат | Итог |
| 1 | Нажать кнопку «Регистрация»» | Появятся формы для регистрации | Появились формы для регистрации | Выполнено |
| 2 | Нажать «Регистрация» после заполнения данных в формы регистрации | Пользователь добавится в базу данных и переход на главную страницу | Новый пользователь добавился в базу данных. и переход на главную страницу | Выполнено |
| 3 | Нажать кнопку «Войти» | Откроется окно входа в систему | Открылось окно входа | Выполнено |
| 4 | Нажать на иконку пользователя | Раскроется список | Раскроется список | Выполнено |
| 5 | Нажать кнопку «Выход» | Выход из системы и переход на главную страницу | Выход из системы и переход на главную страницу | Выполнено |

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данного курсового проекта был разработан веб-сайт, который отвечает современным требованиям и стандартам веб-разработки. Проект был основан на анализе потребностей и целей заказчика, а также учете требований пользователей.

В процессе разработки были использованы современные технологии и инструменты, такие как HTML, CSS, JavaScript, а также фреймворки и библиотеки, позволяющие создать удобный и функциональный интерфейс. Веб-сайт был разработан с учетом адаптивного дизайна, что позволяет его корректное отображение на различных устройствах, включая планшеты.

Также в процессе разработки был проведен тестирование функциональности веб-сайта. Были выявлены и исправлены возможные ошибки и недочеты, что позволило достичь высокой стабильности сайта.

В результате выполнения данного курсового проекта был создан веб-сайт. Он предоставляет удобный и интуитивно понятный интерфейс, а также функциональность, необходимую для достижения поставленных целей.

Разработка данного веб-сайта позволила применить и углубить знания в области веб-разработки, а также приобрести опыт работы и управлении проектом.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Варфоломеева, А. О. Информационные системы предприятия [Электронный ресурс] : учебное пособие / А. О. Варфоломеева, А. В. Коряковский, В. П. Романов. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2019. — 330 с. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/read?id=335060> – Загл. с экрана.
2. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем [Электронный ресурс] : учебное пособие / Л. Г. Гагарина. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 384 с. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/read?id=333679> – Загл. с экрана.
3. Исаев, Г. Н. Управление качеством информационных систем [Электронный ресурс] : учебное пособие / Исаев Г. Н. - Москва :НИЦ ИНФРА-М, 2016. - 248 с.: 60x90 1/16. - (Высшее образование: Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-16-011794-2 - Режим доступа: <https://znanium.com/read?id=36233> – Загл. с экрана.
4. Коваленко, В. В. Проектирование информационных систем [Электронный ресурс] : учебное пособие / В. В. Коваленко. — Москва : ФОРУМ : ИНФРА-М, 2018. — 320 с. — Режим доступа: <https://znanium.com/read?id=329691> – Загл. с экрана.
5. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности [Электронный ресурс]: учебное пособие / Г. Н. Федорова. — Москва :КУРС : ИНФРА-М, 2018. — 336 с. (Среднее Профессиональное Образование). - Режим доступа: <https://znanium.com/read?id=227266> – Загл. с экрана.
6. Золотухина, Е. Б. Управление жизненным циклом информационных систем (продвинутый курс) [Электронный ресурс]: конспект лекций / Е. Б. Золотухина, С. А. Красникова, А. С. Вишня. - Москва :КУРС, НИЦ ИНФРА-М, 2017. - 119 с.: ISBN 978-5-906818-36-2 - Режим доступа: <https://znanium.com/read?id=145707> – Загл. с экрана.
7. Сысоева, Л. А. Управление проектами информационных систем [Электронный ресурс] : учебное пособие / Л. А. Сысоева, А. Е. Сатунина. — Москва : ИНФРА-М, 2019. — 345 с. — (Высшее образование: Бакалавриат). — Режим доступа: <https://znanium.com/read?id=342011> – Загл. с экрана.
8. Голицына, О. Л. Базы данных [Электронный ресурс] : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. — 4-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2019. — 400 с. — Режим доступа: [https://znanium.com/read?id=339368](https://new.znanium.com/read?id=339368) – Загл. с экрана.
9. Федорова, Г. Н. Основы проектирования баз данных [Электронный ресурс] : учебник для студентов учреждений среднего профессионального образования / Г. Н. Федорова. - 3-е изд., стер. - Москва: Издательский центр "Академия", 2019. - 224 с. - Режим доступа: <https://www.academia-moscow.ru/reader/?id=434276> . - ISBN 978-5-4468-8690-6
10. Шустова, Л. И. Базы данных [Электронный ресурс] : учебник / Л.И. Шустова, О.В. Тараканов. — Москва : ИНФРА-М, 2019. — 304 с. + Доп. Материалы. - Режим доступа: [https://znanium.com/read?id=340159](https://new.znanium.com/read?id=340159) – Загл. с экрана.
11. Дадян, Э. Г. Проектирование современных баз данных. Практикум [Электронный ресурс]: учебно-методическое пособие / Э. Г. Дадян. - Москва : НИЦ ИНФРА-М, 2017. - 84 с. – ISBN 978-5-16-106528-0 - Режим доступа: [https://znanium.com/read?id=309392](https://new.znanium.com/read?id=309392) – Загл. с экрана.
12. Кукарцев, В. В. Теория баз данных [Электронный ресурс] : учебник / В. В. Кукарцев, Р. Ю. Царев, О. А. Антамошкин. - Красноярск : Сиб. федер. ун-т, 2017. - 180 с. - ISBN 978-5-7638-3621-9. - Режим доступа: [https://znanium.com/read?id=342132](https://new.znanium.com/read?id=342132) – Загл. с экрана.
13. Немцова, Т. И. Компьютерная графика и web-дизайн [Электронный ресурс] : учебное пособие / Т. И. Немцова, Т. В. Казанкова, А. В. Шнякин ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. + Доп. материалы. - Режим доступа: <https://znanium.com/read?id=329728> — (Среднее профессиональное образование) – Загл. с экрана.
14. Немцова, Т. И. Практикум по информатике. Компьютерная графика и web-дизайн [Электронный ресурс] : учебное пособие / Т. И. Немцова, Ю. В. Назарова ; под ред. Л. Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2018. — 288 с. + Доп. материалы. - Режим доступа: <https://znanium.com/read?id=328146> – Загл. с экрана.
15. Лаврентьев, А. Н. Цифровые технологии в дизайне. История, теория, практика [Электронный ресурс] : учебник и практикум для вузов / А. Н. Лаврентьев [и др.] ; под редакцией А. Н. Лаврентьева. - 2-е изд., испр. и доп. - Москва : Издательство Юрайт, 2019. — 208 с. — (Авторский учебник). — ISBN 978-5-534-07962-3. — Режим доступа: <https://www.urait.ru/bcode/424029> – Загл. с экрана.
16. Павловская, Е. Э. Графический дизайн. Современные концепции [Электронный ресурс] : учебное пособие для вузов / Е. Э. Павловская [и др.] ; ответственный редактор Е. Э. Павловская. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 119 с. — (Университеты России). — ISBN 978-5-534-11169-9. — Режим доступа: <https://www.urait.ru/bcode/444790> – Загл. с экрана.

# ПРИЛОЖЕНИЕ А

Листинг 1 – Регистрация пользователя в базе данных.

<?php

require\_once('./connect.php');

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: POST, GET, OPTIONS");

header("Access-Control-Allow-Headers: Content-Type");

$name = $\_POST['name'];

$name\_last = $\_POST['name\_last'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$date= $\_POST['date'];

$sex= $\_POST['sex'];

$img='/src/assets/img/img2.jpg';

$sql = "SELECT \* FROM users WHERE email='$email'";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) > 0) {

  $response = [

    'success' => false,

    'message' => 'Пользователь с таким email уже существует'

  ];

  header("Content-Type: application/json");

  echo json\_encode($response);

  exit();

}

$hashedPassword = md5($password);

$sql = "INSERT INTO users (name\_first, name\_last, email, password,birthdate,sex,img)

VALUES ('$name','$name\_last', '$email', '$hashedPassword','$date','$sex','$img')";

if (mysqli\_query($conn, $sql)) {

  $userId = mysqli\_insert\_id($conn);

  $roleId = "Пользователь";

  $sql = "INSERT INTO userrole (user\_id, role\_id) VALUES ('$userId', '$roleId')";

  if (mysqli\_query($conn, $sql)) {

    $response = [

        'success' => true,

        'message' => 'Регистрация прошла успешно',

        'role' => $roleId,

        'id' => $userId,

        'img' => $img,

        'name' => $name,

        'name\_last' => $name\_last,

    ];

  } else {

    $response = [

      'success' => false,

      'message' => 'Ошибка при добавлении роли пользователя'

    ];

  }

} else {

  $response = [

    'success' => false,

    'message' => 'Ошибка при выполнении запроса'

  ];

}

header("Content-Type: application/json");

echo json\_encode($response);

mysqli\_close($conn);

?>