

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Базы данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

«База данных для организации деятельности интернет-магазина
компании по разработке компьютерных игр»

БГУИР КП 1-40 01 22 ПЗ

Студент: гр. 081073

Самойло А. А.

Руководитель:

Кунцевич О. Ю.

Минск 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Постановка задачи	7
1.1 Описание предметной области	7
1.2 Обзор существующих аналогов	8
1.3 Информационная база задачи	10
1.4 Инструменты разработки	11
1.5 Функциональное назначение	12
2 Проектирование задачи	13
2.1 Концептуальное проектирование	13
2.2 Функциональная модель	17
3 Программная реализация	21
3.1 Физическая модель базы данных	21
3.2 Реализация проекта базы данных	24
3.2.1 Создание и заполнение таблиц базы данных	24
3.2.2 Создание представлений	26
3.2.3 Назначение прав доступа	27
3.2.4 Создание индексов	27
3.3 Структура приложения	28
4 Тестирование	30
5 Применение	33
5.1 Назначение и область применения	33
5.2 Руководство пользователя	33
Заключение	36
Список использованных источников	37
Приложение А	38
Приложение Б	39
Приложение В	40
Приложение Г	41

ВВЕДЕНИЕ

В настоящее время интернет получил очень широкое применение. Поскольку интернет в наши дни является наиболее обширным источником информации, его применение не ограничивается использованием электронной почты или чатов для общения.

Практически каждая организация сегодня имеет личный сайт, на котором размещается информация о предлагаемых ею товарах и услугах. Это очень удобно, так как позволяет пользователю познакомиться с деятельностью компании и предлагаемым спектром услуг, не выходя из дома.

Поэтому в настоящее время для многих организаций, индивидуальных предпринимателей и прочих, одной из главных задач является разработка такой информационной системы, которая смогла бы обеспечить максимально простой способ взаимодействия представителя компании и клиента через сеть Интернет.

Любая крупная информационная система должна иметь место для хранения какой-либо информации, в современной ИС таким местом является база данных.

База данных ^[1] — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляет системой управления базами данных. Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов, формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать.

Данная курсовая работа рассматривает создание базы данных для интернет-магазина по разработке компьютерных игр. В курсовой работе разрабатывается БД, с помощью которой, можно будет хранить и редактировать информацию о пользователях, товарах и т.д. Данная тема актуальна по причине быстрого роста популярности сферы разработки компьютерных игр.

Интернет-магазин — сайт, торгующий товарами посредством сети Интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ. При этом продажа товаров осуществляется дистанционным способом, и она накладывает ограничения на продаваемые товары.

Целью курсовой работы является изучение основ проектирования баз данных. В процессе изучения дисциплины и выполнения курсовой работы должны быть усвоены теоретические основы организации баз данных, включая принципы построения на концептуальном, логическом и физическом уровнях.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- Провести анализ существующих аналогов;
- выбрать наиболее подходящие инструменты для реализации цели;
- реализовать программный продукт;
- провести тестирование продукта;
- разработать руководство пользователя.

Перед выполнением вышеперечисленных задач необходимо изучить особенности разработки баз данных, а также подходы к реализации различных процедур таких как: поиск, обновление, удаление, добавление, разграничение прав доступа. Изучить подход к разработке приложения для взаимодействия с базой данных, а также базы данных с точки зрения реляционной модели.

Результат курсовой работы должен представлять из себя законченное, готовое к эксплуатации программное средство с сопровождающей документацией.

1 Постановка задачи

1.1 Описание предметной области

Предметная область — часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации. Предметная область представляется множеством фрагментов, например, предприятие - цехами, дирекцией, бухгалтерией и т.д. Каждый фрагмент предметной области характеризуется множеством объектов и процессов, использующих объекты, а также множеством пользователей, характеризуемых различными взглядами на предметную область.

Интернет-магазин — сайт, торгующий товарами посредством сети Интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ. При этом продажа товаров осуществляется дистанционным способом.

В любом интернет-магазине покупатель в первую очередь должен иметь возможность купить товар. Дополнительно должна быть возможность сохранить в корзину понравившийся товар, способ оплаты, просмотреть историю покупок, а также оставить оценку и отзыв.

В качестве товаров будут представлены как компьютерные игры различных жанров и созданные под различные устройства, так и тематические сувениры, а также одежда.

Редактирование списка товаров, категорий, жанров, а также управление пользователями будет осуществляться с помощью панели администратора. Права администратора присваиваются пользователю после регистрации вручную.

Процесс покупки товара состоит из нескольких этапов:

1. Перед покупкой пользователь должен пройти авторизацию на сайте и оставить информацию о себе.
2. Далее происходит выбор необходимых товаров, и добавление их в корзину.
3. Из корзины пользователь может перейти к процессу оплаты товара, для этого необходимо выбрать способ оплаты и оплатить товар.
4. Все купленные товары будут сохранены в личном кабинете пользователя.

Интернет-магазин будет содержать несколько различных типов товаров, а именно компьютерные игры и сувениры. Также каждый товар будет иметь категорию и жанр что позволит пользователям быстрее находить похожие товары. Из наиболее популярных категорий можно выделить следующие: спортивные игры, мобильные игры, шутеры, RPG, симуляторы. Так как категории будут настраиваемыми, то конечный список категорий, либо жанров будет отличаться.

Данный интернет-магазин в первую очередь рассчитан на молодую аудиторию, имеющую средний доход. Также помимо покупателей игр, сайт рассчитан на привлечение потенциальных заказчиков и возможных сотрудников компании.

Помимо интернет-магазина, на сайте будет размещена информация о компании и разрабатываемых играх, а также список актуальных вакансий.

Дополнительно для улучшения работы сайта должна быть реализована форма для связи с разработчиком.

1.2 Обзор существующих аналогов

Аналогов интернет-магазинов в сети интернет достаточно много и их основное отличие в том, что при продаже каких-либо товаров взимается комиссия, которая составляет от 10 до 30 процентов цены товара. Такое решение не подходит для крупной компании по разработке игр, т к. конечная цена будет гораздо выше, что будет отталкивать клиентов от покупки. В связи с этим актуальность интернет-магазина становится весьма высокой.

Рассмотрим несколько похожих решений:

- Steam (<https://store.steampowered.com/>)
- Ubisoft (<https://www.ubisoft.com/en-us/>)
- Zaka-Zaka (<https://zaka-zaka.com/>)

Steam - онлайн-сервис цифрового распространения компьютерных игр и программ, разработанный и поддерживаемый компанией Valve. Steam выполняет роль средства технической защиты авторских прав, платформы для многопользовательских игр и потокового вещания, а также социальной сети для игроков.

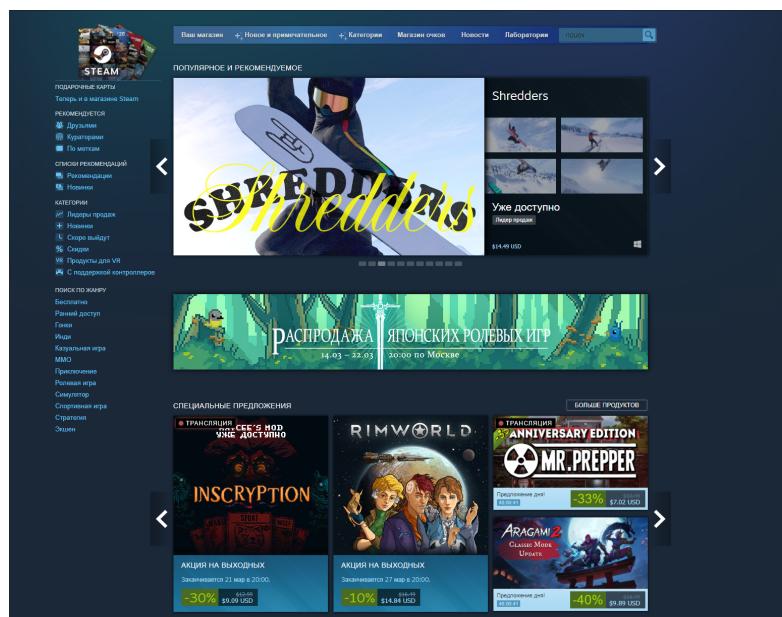


Рис.1.1 - Главный экран Steam

Ключевые особенности Steam:

- В роли продавца программного обеспечения или компьютерных игр может выступать как крупная компания, так и обычный пользователь.
- У каждого пользователя есть свой собственный настраиваемый профиль.

Основным достоинством Steam, является магазин компьютерных игр, он содержит весь необходимый функционал для покупки, возврата, а также передачи другому пользователю игр.

Недостатком является то, что в Steam невозможно продвижение своей компании, а также невозможна продажа различных сувениров.

Ubisoft - французская компания, специализирующаяся на разработке и издании компьютерных игр.

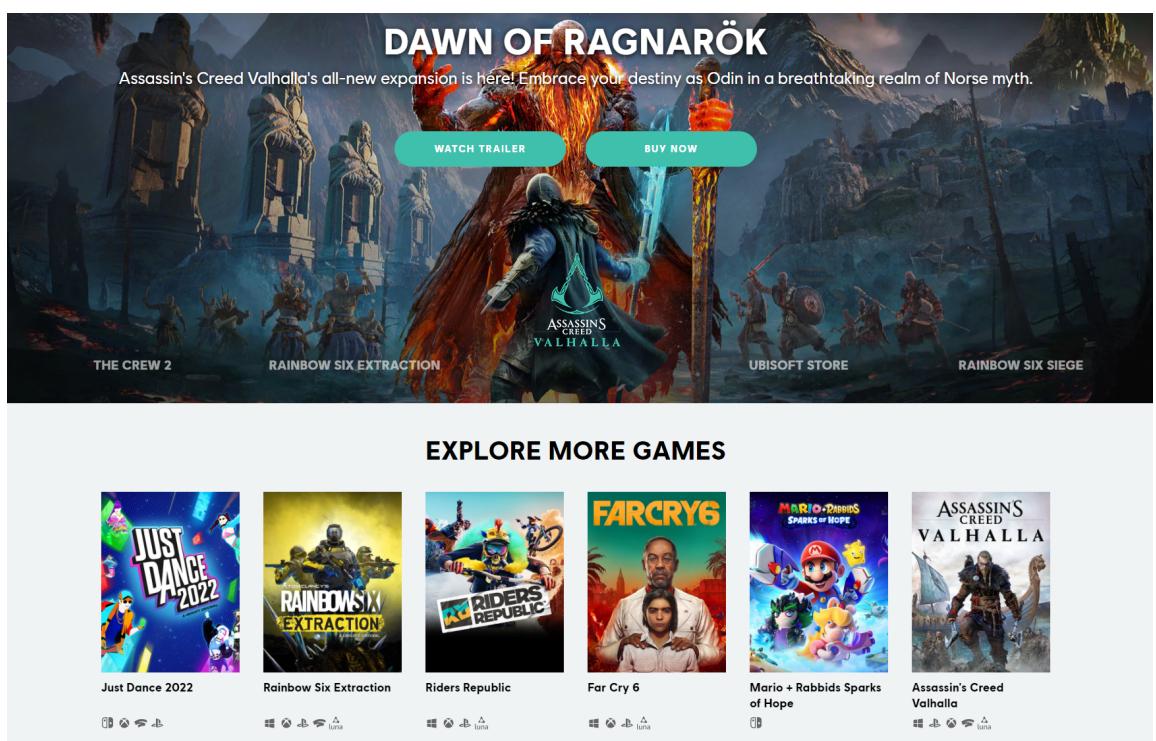


Рис. 1.2 - Главный экран Ubisoft

Основной особенностью является то, что помимо интернет-магазина на сайте размещена информация о самой компании, доступных вакансиях, а также новости.

Недостатком является отсутствие собственного интернет-магазина, при нажатии на кнопку купить, происходит переход на другой сервис для покупки компьютерных игр.

Zaka-Zaka.com – это интернет-магазин компьютерных игр и ключей активации для Steam, Origin, Uplay, Battle.net и других игровых платформ.

Данный аналог отличается от предыдущих тем что предоставляет только услуги по продаже компьютерных игр и не имеет собственной студии по разработке игр.

Достоинством данного интернет-магазина являются низкие цены, но существенным недостатком является низкое качество поддержки, а также невозможность возврата товара.

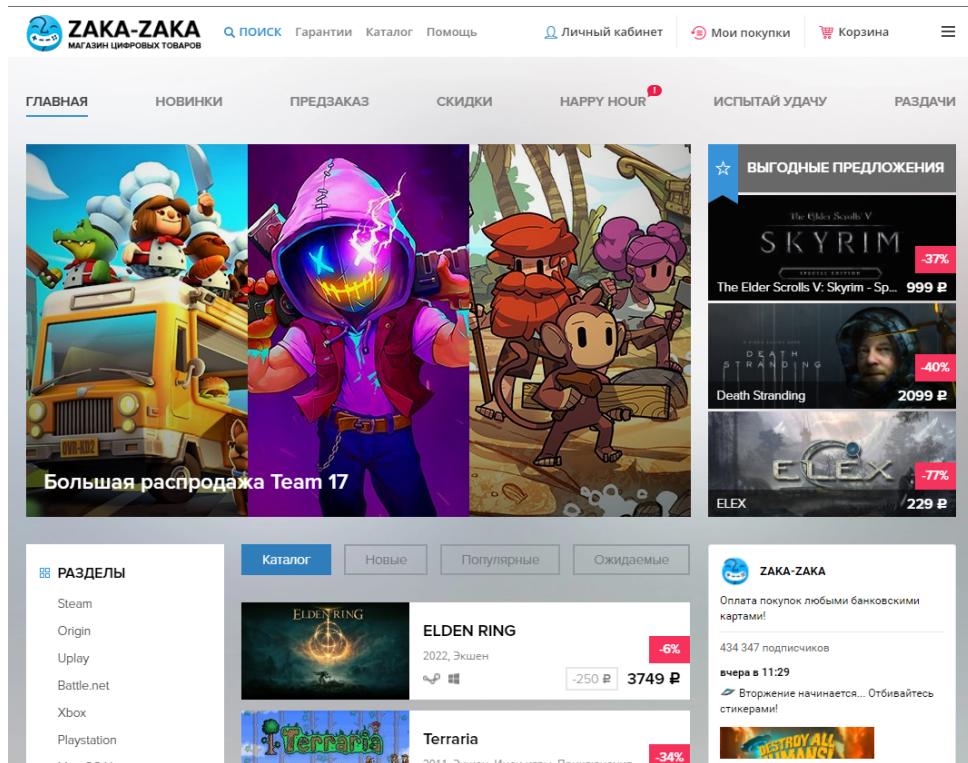


Рис. 1.3 - Главный экран Zaka-Zaka

На основании анализа существующих аналогов следует сделать вывод о том, что данные сервисы пользуются спросом у пользователей. Для создания собственного сайта, необходимо будет воспользоваться достоинствами обоих сервисов.

1.3 Информационная база задачи

Информационная база задачи состоит из входной, выходной и постоянной информации.

Входная информация – информация, поступающая в систему в виде данных, внесенных пользователем.

Выходная информация – информация, получаемая в результате выполнения функций или запросов.

Постоянная информация – информация, которая остается неизменной в течение длительного периода времени и многократно используется при обработке переменной информации.

Входные данные разрабатываемого программного средства:

- данные пользователей, данные товаров, данные о категориях и типах товаров, данные о оценках и отзывах;
- различные запросы пользователя, выполняемые во время использования сайта.

Выходные данные разрабатываемого программного средства:

- результаты выполнения запросов или функций;
- информация, отображаемая на экране.

Постоянные данные разрабатываемого программного средства представлены в виде информации о компании.

1.4 Инструменты разработки

Разрабатываемый продукт будет разрабатываться с помощью операционной системы Microsoft Windows 11.

Microsoft Windows 11 обладает множеством преимуществ над своими младшими версиями такими, как визуальное оформление, быстродействие, поддержка новых технологий и тому подобное, что позволяет не только разрабатывать программный продукт в полном соответствии со всеми заданными условиями, но и создать удобный и интуитивно понятный интерфейс программы, что играет немаловажную роль в проектировании информационно-справочных систем.

В качестве среды разработки приложения будет использоваться IntelliJ IDEA. IntelliJ IDEA – это умная и удобная среда разработки для Java, включающая поддержку всех последних технологий и фреймворков. Она предоставляет инструменты для продуктивной работы и идеально подходит для создания коммерческих, мобильных и веб-приложений.

Программное обеспечение будет написано на java^[2]. В java 15^[3] включены все необходимые компоненты и библиотеки для создания полноценного приложения.

Основой для приложения послужит фреймворк Spring boot^[4]. Spring — универсальный фреймворк с открытым исходным кодом для Java-платформы. Spring обеспечивает решение многих задач, с которыми сталкиваются Java-разработчики, фреймворк предлагает целый набор инструментов, таких как: встроенный контейнер, внедрение зависимостей, организация работы с БД и т.д.

Для сборки проекта будет использоваться maven. Apache Maven — фреймворк для автоматизации сборки проектов на основе описания их структуры в РОМ файлах.

Запросы для манипулирования данными будут написаны на JPQL. JPQL — это объектно-ориентированный язык запросов, его особенностью является абстрагирование от структуры БД, запросы создаются на основе java объектов, и с помощью фреймворка Hibernate автоматически преобразуются в SQL.

В качестве базы данных будет использоваться MySql^[5]. MySql представляет собой одну из самых распространенных сегодня систем управления базами данных в сети Интернет. Данная система используется для работы с достаточно большими объемами информации, поддерживая при этом высокий уровень производительности. Однако MySQL идеально подходит как для небольших, так и для крупных интернет-проектов. Немаловажной характеристикой системы является ее бесплатность.

1.5 Функциональное назначение

Сайт предназначен для продажи компьютерных игр и различных сувениров через сеть Интернет, а также для размещения актуальных вакансий в компании.

Основные требования к функциональности сайта:

- авторизация клиента и администратора;
- ведение базы данных клиентов;
- ведение базы данных товаров и отзывов;
- ведение базы данных доступных вакансий;
- обработка покупок;
- поиск товаров по заданным критериям;

Также, программное средство должно иметь возможность быстрого формирования отчета по товарам. Отчет должен быть представлен в виде Excel таблицы и содержать полную информацию о товаре.

Для разграничения доступа должна использоваться система ролей.

Пользователь с ролью «гость» должен иметь возможность просматривать информацию о компании, а также список, товаров и вакансий.

Для покупки товаров, написания отзывов пользователю необходимо зарегистрироваться и пройти авторизацию.

Администратор должен иметь возможность редактировать информацию о товарах, категориях, жанрах и т.д, а также выполнять собственные SQL запросы.

Помимо вышеперечисленных функций необходимо предусмотреть возможность расширения. В будущем может потребоваться хранение дополнительной информации, а также выполнение других различных функций. Добавление нового функционала не должно сильно влиять на корректную работоспособность старого.

2 Проектирование задачи

2.1 Концептуальное проектирование

Концептуальное проектирование технических систем — начальная стадия проектирования, на которой принимаются решения, определяющие последующий облик, и проводится исследование и согласование параметров созданных технических решений с возможной их организацией. Термин «концепция» применяется для описания принципа действия не только в технических системах, но и в научных, художественных и прочих видах деятельности. Концепт — содержание понятия, смысл. Таким образом, проектирование на концептуальном уровне — на уровне смысла или содержания понятия систем.

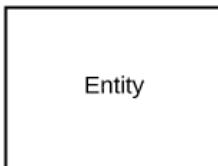
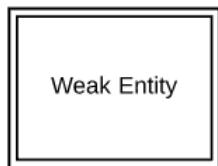
Основной объем задач концептуального проектирования относится к ранним стадиям разработки технических систем: при постановке задачи на проектирование, выработке массива вариантов технических и оформительских решений.

Одним из наиболее важных этапов концептуального проектирования является построение диаграммы «сущность-связь».

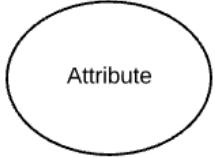
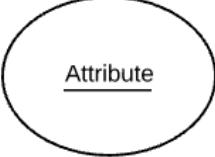
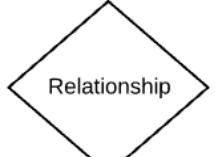
Модель «сущность — связь»^[6] используется при высокогенерализованном (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Важным свойством модели «сущность — связь» является то, что она может быть представлена в виде графической схемы. Это значительно облегчает анализ предметной области. Существует несколько вариантов обозначения элементов^[7] (или сущностей) диаграммы «сущность — связь», каждый из которых имеет свои положительные черты. Элементы представлены в таблице 2.1.

Таблица 2.1 - Элементы диаграммы «сущность — связь».

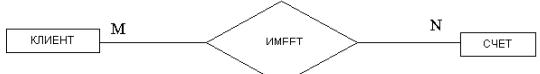
Обозначение	Значение
	Независимая (сильная) сущность
	Зависимая (слабая) сущность

Продолжение таблицы 2.1

	Атрибут
	Ключевой атрибут
	Связь

Помимо различных элементов, в диаграмме также определены связи между элементами. Виды связей представлены в таблице 2.2.

Таблица 2.2 - Виды связей.

Обозначение	Значение
	ОДИН-К-ОДНОМУ
	ОДИН-КО-МНОГИМ
	МНОГИЕ-КО-МНОГИМ

Применимо к разрабатываемой задаче можно выделить следующие сущности, представленные в таблице 2.3.

Таблица 2.3 — Описание сущностей

Сущность	Описание

Пользователи	В этой сущности идентифицируются все зарегистрированные пользователи и администраторы.
Учетные данные	В этой сущности идентифицируются данные пользователей необходимые для авторизации.
Карточки	В этой сущности идентифицируются сохраненные банковские карточки пользователей.
Изображения	Эта сущность идентифицирует ссылки на изображения, хранящиеся в файловой системе.
Товары	Эта сущность описывает товары (игры, сувениры).
Категории	Эта сущность описывает категории товаров.
Жанры	Данная сущность описывает жанры товаров.
Разработчики	Данная сущность идентифицирует разработчиков.
Отзывы	В этой сущности идентифицируются отзывы на товары.
Вакансии	Данная сущность описывает доступные вакансии.
Сообщения разработчикам	Данная сущность описывает жалобы и предложения, отправленные разработчикам.

Следующим шагом является идентификация связей. Связь – это ассоциация между двумя сущностями, значимая для рассматриваемой предметной области.

Логические зависимости (связи), которые прослеживаются между объектами нашей предметной области, будут как обязательными, так и необязательными. Причем, по характеру объединения данных установленные связи относятся к видам «Один – ко – многим», «Один – к – одному» и «Многие – ко – многим». Описание связи между сущностями представлено в таблице 2.4.

Таблица 2.4 — Описание связи между сущностями

Сущность 1	Связь	Вид связи	Сущность 2
Пользователи	У пользователей может быть множество товаров в корзине.	Многие ко многим	Товары
	У пользователей может быть множество купленных товаров.	Многие ко многим	
	У пользователя могут быть только одни учетные данные.	Один к одному	Учетные данные
	У пользователя может быть несколько карточек.	Один ко многим	Карточки
Пользователи	Пользователь может написать несколько отзывов на товары.	Многие ко многим	Товары
	У пользователя может быть только одно изображение.	Один к одному	Изображения
	Пользователь может отправить несколько сообщений разработчикам.	Один ко многим	Сообщения разработчикам

Продолжение таблицы 2.4

Товары	У товара может быть только одно изображение.	Один к одному	Изображения
	У товара может быть несколько отзывов от пользователей	Многие ко многим	Пользователи
	У товара может быть только одна категория	Один к одному	Категории
	У товара может быть только один жанр	Один к одному	Жанры
	У товара может быть только один разработчик	Один к одному	Разработчики

Последним шагом моделирования является идентификация атрибутов. Атрибуты сущностей будут изображены на конечной диаграмме. Полученная диаграмма «сущность — связь» представлена на рисунке 2.1.

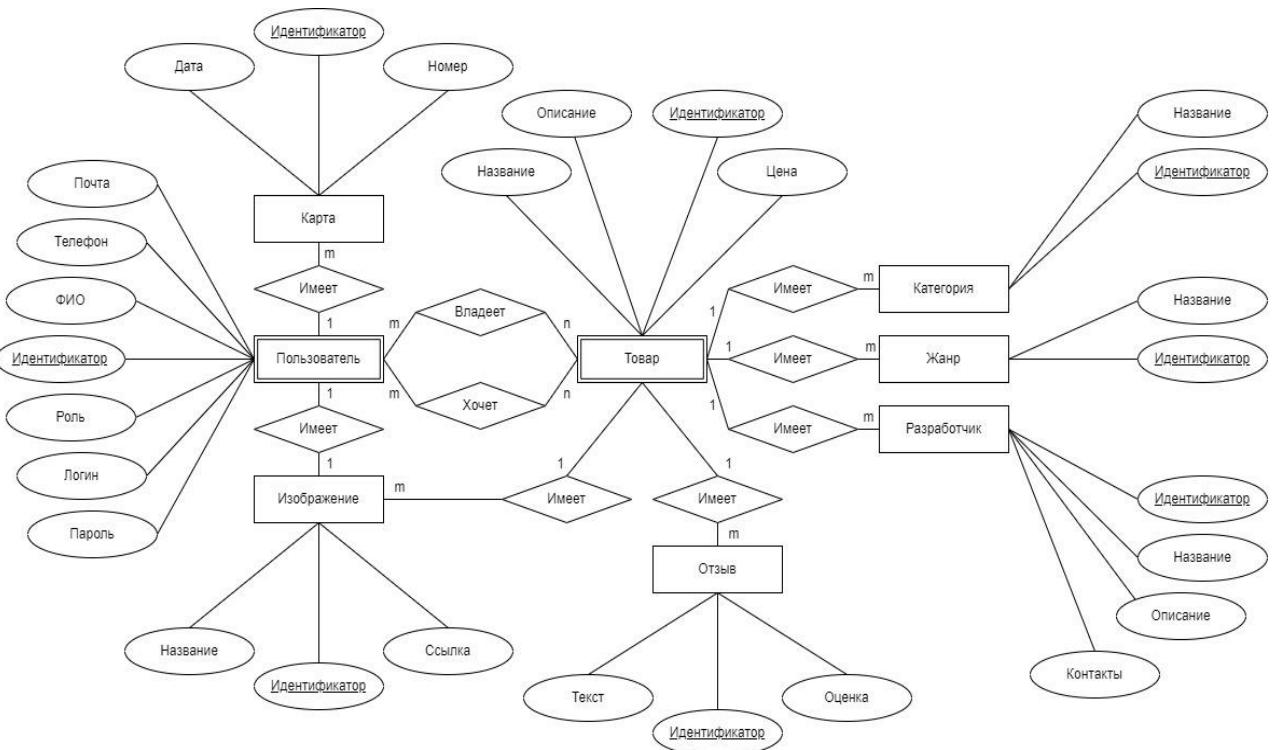


Рис. 2.1 - Диаграмма «сущность — связь»

2.2 Функциональная модель

Диаграмма вариантов использования [8] (use case diagrams), предназначенная для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой) и отражает

функциональные возможности и требования системы с использованием действующих лиц и вариантов использования.

Диаграммы вариантов использования показывают взаимодействия между вариантами использования и действующими лицами, отражая функциональные требования к системе с точки зрения пользователя. Цель построения диаграмм вариантов использования — это документирование функциональных требований в самом общем виде, поэтому они должны быть предельно простыми.

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). В нашем случае действующим лицом (актером), будет выступать пользователь, а именно пользователь, у которого будет открыт доступ к базе данных.



Рис. 2.2 - Актер

Вариант использования изображается на диаграмме эллипсом внутри, которого содержится его краткое название или имя.



Рис. 2.3 — Вариант использования

Вариант использования описывает типичное взаимодействие между Пользователем и системой и отражает представление о поведении системы с точки зрения пользователя. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать, или целей, которые он преследует по отношению к разрабатываемой системе.

Также есть еще один базовый элемент — операция отношения. Отношение — это семантическая связь между отдельными элементами модели. В нашем случае будем пользоваться только тремя из них:

1. Отношение ассоциации. Оно указывает на то, что актер инициирует соответствующий вариант использования. Пользователь, перед тем перейти к базе данных, запускает программу

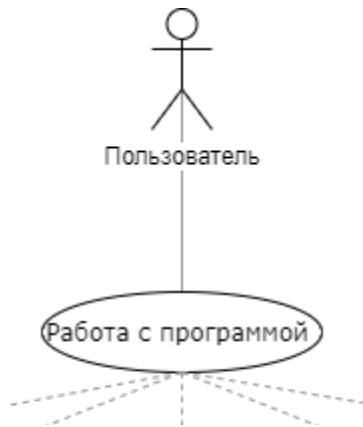


Рис. 2.4 — Отношение ассоциации

2. Отношение включения означает, что поведение или выполнение одного варианта использования включения в качестве составного объекта в поведение или выполнение другого варианта использования



Рис. 2.5 — Отношение включения

3. Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров.



Рис. 2.6 — Отношение расширения

Диаграмма вариантов использования является самым общим представлением функциональных требований к системе. Диаграмма вариантов использования изображена на рисунке 2.7.

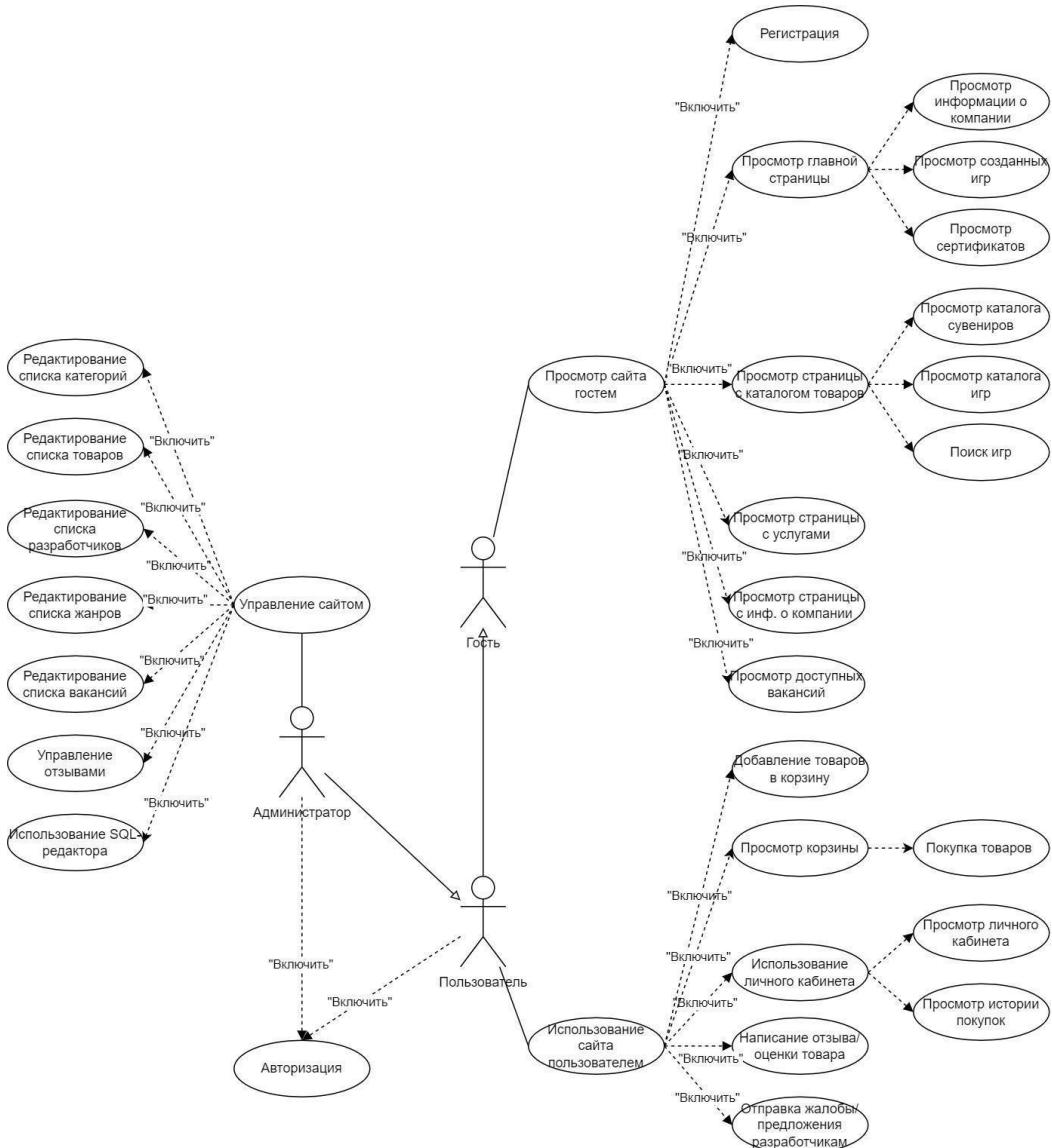


Рис. 2.7 — Диаграмма вариантов использования

3 Программная реализация

3.1 Физическая модель базы данных

На этапе физического проектирования базы данных приводится состав таблиц базы данных. Для каждого поля таблицы необходимо указать используемый тип данных и, размер поля (количество символов), т.е. составить таблицу спецификаций. Для первичных ключей необходимо ввести запрет неопределенных значений. Для остальных полей возможность запрета неопределенных значений определяется семантикой предметной области.

Физическая модель базы данных [9] — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных.

На первом этапе происходит выделение сущностей. Сущность (entity) представляет тип объектов, которые должны храниться в базе данных. Каждая таблица в базе данных должна представлять одну сущность. Как правило, сущности соответствуют объектам из реального мира.

У каждой сущности определяют набор атрибутов. Атрибут (элемент данных) - наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных также часто называют полем.

Каждый столбец должен хранить один атрибут сущности. А каждая строка представляет отдельный объект или экземпляр сущности.

Физическая модель базы данных будет построена на основе диаграммы “сущность-связь”, а также таблицы с описанием атрибутов сущностей. Описание базы данных представлено в таблице 3.1.

Таблица 3.1 — Описание базы данных

Сущность	Атрибут	Тип данных
image	id (PK)	bigint
	name	varchar(45)
	link	varchar(128)
user	id (PK)	bigint
	name	varchar(45)
	lastname	varchar(45)
	phone	varchar(15)
	birthday	timestamp
	email	varchar(45)

	metadata_id (FK)	bigint
Продолжение таблицы 3.1		
user	image_id (FK)	bigint
credentials	user_id (FK)(PK)	bigint
	login	varchar(45)
	password	varchar(100)
	role	enum (GUEST, USER, ADMIN)
category	id (PK)	int8
	value	varchar(45)
developer	id (PK)	bigint
	title	varchar(45)
	description	varchar(256)
	contacts	varchar(256)
	address	varchar(256)
	metadata_id (FK)	bigint
genre	id (PK)	bigint
	value	varchar(45)
product	id (PK)	bigint
	title	varchar(45)
	description	varchar(256)
	price	decimal(13,2)
	type	enum(GAME, MERCH)
	category_id (FK)	bigint
	developer_id (FK)	bigint
	genre_id (FK)	bigint
	image_id (FK)	bigint
	metadata_id (FK)	bigint
cart	product_id (FK)(PK)	bigint
	user_id (FK)(PK)	bigint
purchase	product_id (FK)(PK)	bigint
	user_id (FK)(PK)	bigint

Продолжение таблицы 3.1

feedback	id (PK)	bigint
	title	varchar(45)
	description	varchar(256)
	user_id (FK)	bigint
	metadata_id (FK)	bigint
card	id (PK)	bigint
	number	varchar(45)
	date	varchar(45)
	metadata_id (FK)	bigint
	user_id	bigint
vacancy	id (PK)	bigint
	title	varchar(45)
	description	varchar(2048)
	metadata_id (FK)	bigint
review	id (PK)	bigint
	text	varchar(256)
	score	int
	user_id	bigint
	metadata_id (FK)	bigint
	product_id (FK)	bigint
session	id (PK)	bigint
	created_at	timestamp
	success	tinyint
	user_id (FK)	bigint
session	id (PK)	bigint
	created_at	timestamp
	updated_at	timestamp
	created_by	varchar(45)
	enabled	tinyint

На основе концептуальной и логической моделей, была создана физическая модель базы данных, адаптированная под СУБД MySql. Физическая модель базы данных представлена на рисунке 3.1.

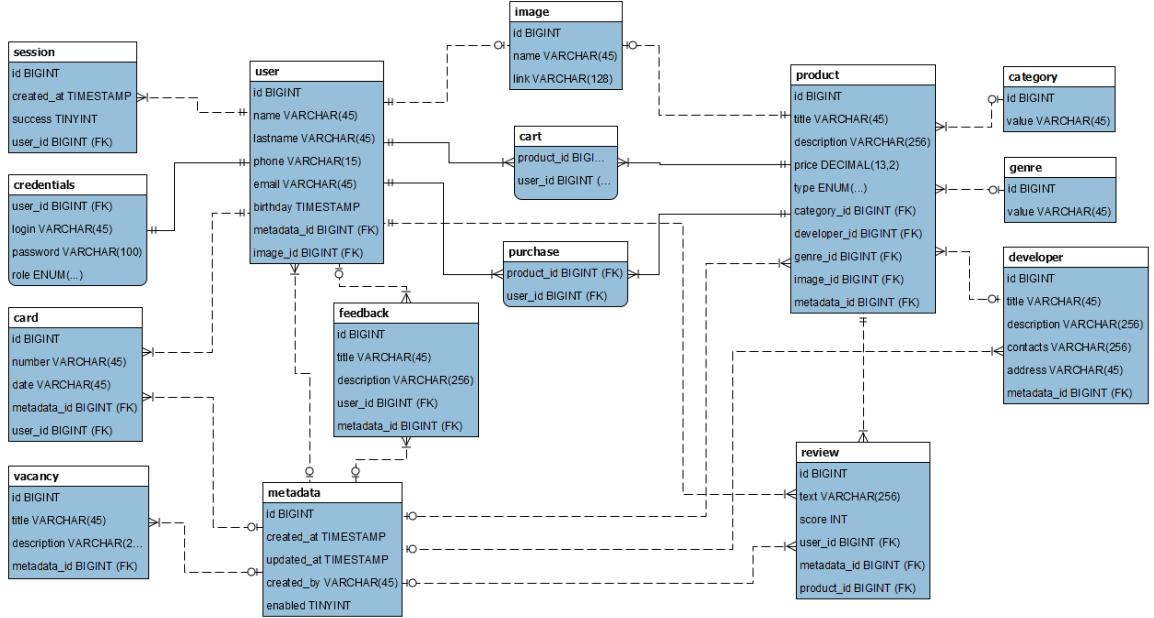


Рис. 3.1 — Физическая модель базы данных

3.2 Реализация проекта базы данных

3.2.1 Создание и заполнение таблиц базы данных

Для создания таблиц базы данных использован SQL. Создание таблицы в базе данных включает в себя присваивание имени таблице и определение колонок и типа данных каждой из них.

С помощью команды CREATE TABLE создаются таблицы в базе данных. С этой командой можно использовать ряд операторов, которые определяют столбцы таблицы и их атрибуты. И кроме того, можно использовать ряд операторов, которые определяют свойства таблицы в целом.

Скрипт создания таблицы user представлен ниже:

```

CREATE TABLE IF NOT EXISTS `qualleinc`.`user` (
  `id` BIGINT AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `lastname` VARCHAR(45) NULL,
  `phone` VARCHAR(15) NULL,
  `email` VARCHAR(45) NULL,
  `birthday` TIMESTAMP NULL,
  `metadata_id` BIGINT NULL,
  `image_id` BIGINT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_user_metadata_idx` (`metadata_id` ASC) VISIBLE,
  INDEX `fk_user_image_idx` (`image_id` ASC) VISIBLE,

```

```

CONSTRAINT `fk_user_metadata`
FOREIGN KEY (`metadata_id`)
REFERENCES `qualleinc`.`metadata`(`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_user_image`
FOREIGN KEY (`image_id`)
REFERENCES `qualleinc`.`image`(`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

После имени таблицы в скобках указываются параметры всех столбцов и в самом конце атрибуты, которые относятся ко всей таблице. Атрибуты столбцов и атрибуты таблицы являются необязательными компонентами.

Для обеспечения версионирования при работе с базой данных использован фреймворк liquibase.

Liquibase - независимая от базы данных библиотека для отслеживания, управления и применения изменений схемы базы данных. Изменения структуры базы данных записываются в файлы, которые называются changelog. Поддерживаемые форматы: XML, YAML, JSON или SQL.

Скрипт создания базы данных будет автоматически выполняться при запуске приложения, если в базе данных будут отсутствовать все необходимые таблицы.

Скрипт заполнения таблицы тестовыми данными будет запускаться вручную при необходимости. Пример скрипта для заполнения таблицы с товарами представлен ниже:

```

INSERT INTO product(title, price, description, type, developer_id,
category_id, genre_id, image_id)
values ('Witcher 3', 12.00, 'The Witcher is a series of fantasy action
role-playing games developed by CD Projekt Red and published by CD
Projekt.', 'GAME', 4, 4, 4, 1);
INSERT INTO product(title, price, description, type, developer_id,
category_id, genre_id, image_id)
values ('Battlefront 2', 23.00, 'Become the Hero in a galaxy at war in Star
Wars Battlefront II. Available for PlayStation 4, Xbox One, and on Origin
for PC November 2021.', 'GAME', 5, 4, 2, 2);
INSERT INTO product(title, price, description, type, developer_id,
category_id, genre_id, image_id)
values ('Assassins Creed Origins', 34.00, 'Assassin"s Creed Origins is a 2021
action role-playing video game developed by Ubisoft Montreal and
published by Ubisoft.', 'GAME', 2, 4, 8, 3);

```

```
INSERT INTO product(title, price, description, type, developer_id,
category_id, genre_id, image_id)
values ('Mortal Kombat', 27.00, 'Mortal Kombat is an American media
franchise centered on a series of video games, originally developed by
Midway Games in 1992.', 'GAME', 1, 4, 9, 4);
```

3.2.2 Создание представлений

Представление — виртуальная таблица, представляющая собой поименованный запрос, который будет подставлен как подзапрос при использовании представления.

В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе. Содержимое представления динамически вычисляется на основании данных, находящихся в реальных таблицах. Изменение данных в реальной таблице базы данных немедленно отражается в содержимом всех представлений, построенных на основании этой таблицы.

В данном проекте использовано представление товаров которое включает в себя ссылку на изображение, название разработчика, категорию и жанр. Скрипт для создания представления представлен ниже:

```
CREATE VIEW complete_product AS
SELECT p.title,
       p.description,
       p.price,
       p.type,
       i.link,
       c.value AS category_value,
       d.title AS developer_title,
       g.value AS genre_value
FROM product AS p
      LEFT JOIN image AS i
            ON p.image_id = i.id
      LEFT JOIN category c
            ON p.category_id = c.id
      LEFT JOIN developer d
            ON d.id = p.developer_id
      LEFT JOIN genre g
            ON g.id = p.genre_id
WHERE p.id = 2;
```

3.2.3 Назначение прав доступа

База данных представляет собой сущность для хранения информации в виде таблиц. Для того чтобы чужие базы данных не были доступны абсолютному каждому пользователю на сервере, существует система пользователей для этих баз данных. Доступ к какой-либо базе может быть назначен администратором (либо уполномоченным пользователем) другому пользователю, причем он может быть полным или в некоторой степени ограниченным. Более конкретно эта степень доступа выражается в привилегиях («правах» или «разрешениях»).

Для доступа к базе данных приложения используется 2 пользователя.

Первый пользователь является администратором и имеет все возможные привилегии. Данный пользователь используется для ручного изменения схемы базы данных либо данных.

Второй пользователь используется приложением для доступа к данным, а также редактирования данных.

```
CREATE USER 'qualle'@'localhost' IDENTIFIED BY '43q21';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER,
INDEX, DROP, TRIGGER, EVENT ON qualleinc.* TO 'qualle'@'localhost';
FLUSH PRIVILEGES;
```

3.2.4 Создание индексов

Системы баз данных обычно используют индексы для обеспечения быстрого доступа к реляционным данным. Индекс представляет собой отдельную физическую структуру данных, которая позволяет получать быстрый доступ к одной или нескольким строкам данных. Таким образом, правильная настройка индексов является ключевым аспектом улучшения производительности запросов.

Индекс базы данных во многом сходен с алфавитным указателем книги. Когда нужно быстро найти какую-либо тему в книге, мы сначала смотрим в индексе, на каких страницах книги эта тема рассматривается, а потом сразу же открываем нужную страницу. Подобным образом, при поиске определенной строки таблицы компонент Database Engine обращается к индексу, чтобы узнать ее физическое местонахождение.

В базе данных приложения индексы используются в каждой таблице для обеспечения быстрого поиска по первичному ключу (id). Индекс для таблицы создается с помощью инструкции CREATE INDEX Пример создания индекса изображен ниже:

```
CREATE INDEX `fk_user_metadata_idx` ON user (`metadata_id` ASC)
VISIBLE;
```

3.3 Структура приложения

Структуру приложения можно разделить на логическую и физическую структуры.

Логическая структура сайта представлена в виде интеллектуальной карты, на которой будут изображены все страницы сайта, а также информация, которая будет размещена на данных страницах.

Интеллектуальная карта [10] (MindMap) — карта, с помощью которой можно систематизировать в виде схемы любое понятие. Используется для самостоятельного или группового анализа любых вопросов, создания структуры сайта, визуальной систематизации информации, составление плана работы или действий по решению задач. Вид карты полностью зависит от воображения человека, в основном используется графика в виде блок-схем. Структура очень важна для любого сайта — неважно, информационный ресурс, интернет-магазин, корпоративный блог или сайт-визитка.

Логическая структура сайта изображена на рисунке 3.2.

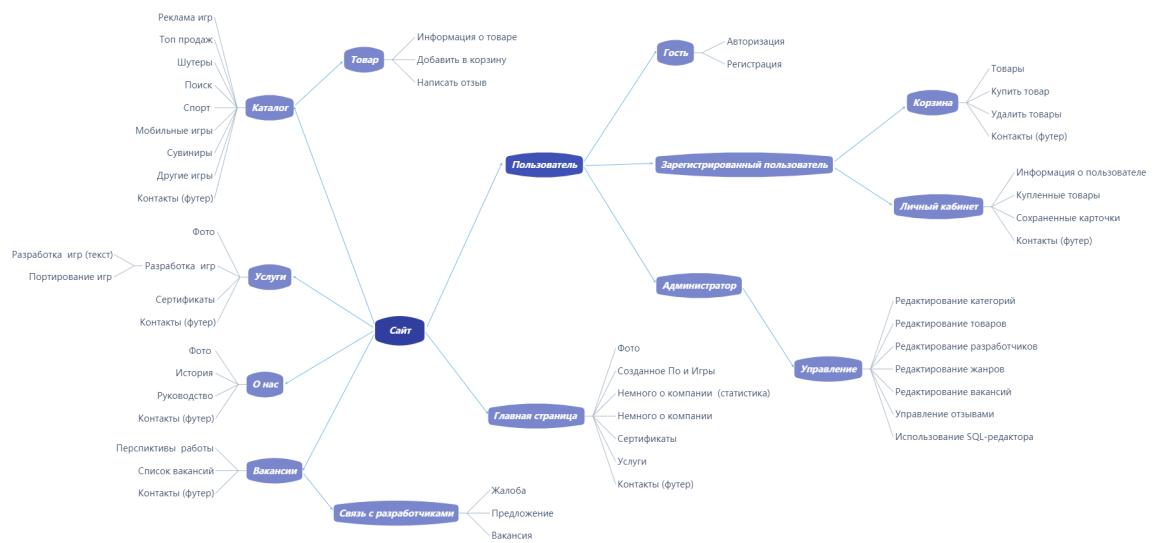


Рис. 3.2 — Интеллектуальная карта сайта

Физическая структура сайта представляет собой алгоритм размещения физических файлов по директориям. Поскольку в данном web-приложении использована архитектура MVC [11], то во многом данная архитектура определяет физическое расположение файлов приложения.

MVC (Model-View-Controller) — это шаблон в разработке программного обеспечения, обычно используемый для реализации пользовательских интерфейсов, данных и управляющей логики. Данный шаблон предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает

за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения.

Компоненты MVC:

Модель — этот компонент отвечает за данные, а также определяет структуру приложения.

Представление — этот компонент отвечает за взаимодействие с пользователем. То есть код компонента view определяет внешний вид приложения и способы его использования.

Контроллер — этот компонент отвечает за связь между model и view. Код компонента controller определяет, как сайт реагирует на действия пользователя. По сути, это мозг MVC-приложения.

Физическая структура сайта изображена на рисунке 3.3.

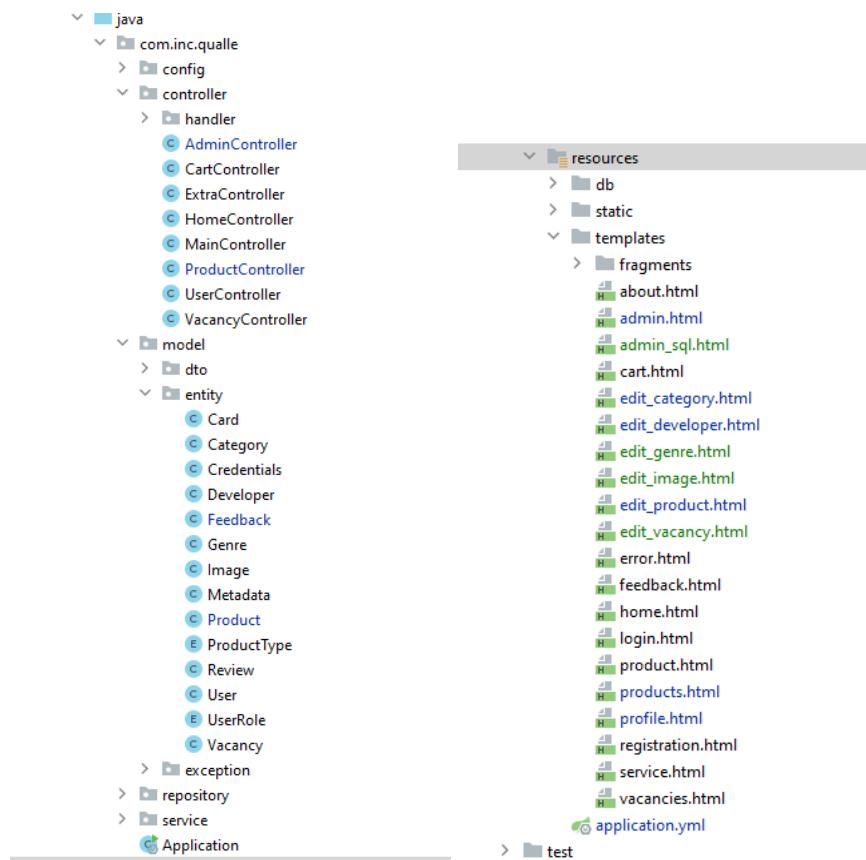


Рис. 3.3 Физическая структура сайта

4 Тестирование

Тестирование ПО^[11] — это процесс его исследования с целью получения информации о качестве продукта. Целью тестирования является выявление дефектов в ПО. С помощью тестирования нельзя доказать отсутствие дефектов и корректность функционирования анализируемой программы. Тестирование сложных программных продуктов является творческим процессом, не сводящимся к следованию строгим и четким процедурам.

Дефект — изъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию.

Покрытие — это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

Существует большое количество различных видов тестирования. Рассмотрим два основных:

Модульное тестирование. Каждая сложная программная система состоит из отдельных частей - модулей, выполняющих ту или иную функцию в составе системы. Для того, чтобы удостовериться в корректной работе всей системы, необходимо сначала протестировать каждый модуль системы по отдельности. В случае возникновения проблем при тестировании системы в целом это позволяет проще выявить модули, вызвавшие проблему, и устранить соответствующие дефекты в них.

В Java за модульное тестирование отвечают специальные Test классы, которые работают с помощью JUnit 5. JUnit — это библиотека для модульного тестирования программного обеспечения на языке Java. Принцип работы данной библиотеки заключается в сравнении ожидаемого результата с полученным. Тестирование запускается при компиляции кода чтобы обеспечить работоспособность готового приложения.

С помощью данной библиотеки будет осуществлено тестирование некоторых функций приложения т.к. конвертация строки с временем в java объект Date и конвертация сущности из БД в DTO. DTO (data transfer object) в переводе объект передачи данных — это один из шаблонов проектирования, используется для передачи данных между подсистемами приложения.

На рисунке 4.1 приведен пример тестирования конвертации сущности Product в объект ProductDto который будет использован для передачи информации в пользовательский интерфейс. Сущность тестирования заключается в сравнении всех полей исходной сущности и выходного объекта. Тест считается успешным если данные всех полей идентичны. На рисунке 4.2 изображен результат тестирования метода.

```

    @Test
    public void testWhenConvertDtoThanSuccess(){
        Product product = getProduct();

        ProductDto dto = productMapper.toDto(product);

        assertEquals( expected: 1, dto.getId());
        assertEquals( expected: "Test", dto.getTitle());
        assertEquals( expected: "For Test", dto.getDescription());
        assertEquals( expected: 10, dto.getPrice());
        assertEquals( ProductType.GAME, dto.getType());
        assertEquals( expected: 1, dto.getCategory().getId());
        assertEquals( expected: "Test category", dto.getCategory().getValue());
        assertEquals( expected: 1, dto.getDeveloper().getId());
        assertEquals( expected: "Test developer", dto.getDeveloper().getTitle());
        assertEquals( expected: "Description", dto.getDeveloper().getDescription());
        assertEquals( expected: 1, dto.getGenre().getId());
        assertEquals( expected: "Test genre", dto.getGenre().getValue());
        assertEquals( expected: 1, dto.getImage().getId());
        assertEquals( expected: "Test link", dto.getImage().getLink());
        assertEquals( expected: 1, dto.getMetadata().getId());
        assertEquals( expected: "User", dto.getMetadata().getCreatedBy());
    }

    @Test
    public Product getProduct() {
        return Product.builder()
            .id(1)
            .title("Test")
            .description("For Test")
            .price(10)
            .type(ProductType.GAME)
            .category(Category.builder().id(1).value("Test category").build())
            .developer(Developer.builder().id(1).title("Test developer").description("Description").build())
            .genre(Genre.builder().id(1).value("Test genre").build())
            .image(Image.builder().id(1).link("Test link").build())
            .metadata(Metadata.builder().id(1).createdAt(LocalDateTime.now()).createdBy("User").build())
            .build();
    }
}

```

Рис. 4.1 – Тестирование конвертации сущности Product

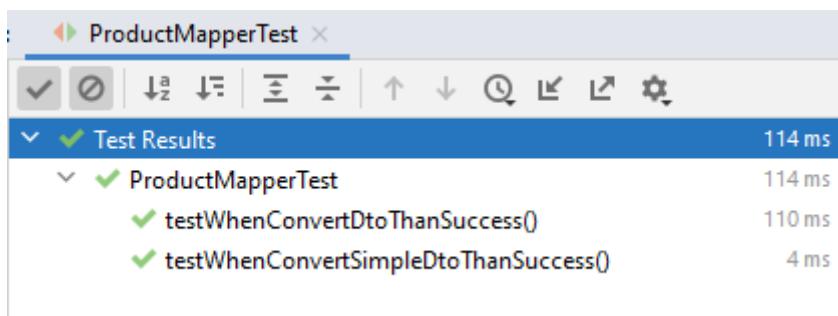


Рис. 4.2 – Результат тестирования конвертации сущности Product

Функциональное тестирование - тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тест-кейс – это описанный алгоритм тестирования программы, специально созданный для определения возникновения в программе определённой ситуации, определённых выходных данных.

В Таблице 4.1 представлено тестирование основных действий пользователя на странице товаров, ожидаемый результат и полученный.

Таблица 4.1 – Тестирование функций страницы товаров

Тест-кейс	Предусловие	Ход выполнения	Ожидаемый результат
Нажатие на иконки с товарами	Необходимы товары в бд.	Нажать на иконку товара	Произойдет переход на страницу с описанием выбранного товара
Нажатие на строку поиска	Необходимы товары в бд.	Нажать на строку поиска	Произойдет переход на страницу поиска товаров
Нажатие на ссылки в шапке страницы	-	Нажать на ссылку	Произойдет переход на выбранную страницу
Нажатие на добавить в корзину	Зарегистрированный пользователь, товар	Нажать на кнопку “Добавить в корзину”	Для зарегистрированного пользователя произойдет переход на страницу с корзиной в которую будет добавлен товар Для гостя произойдет переход на страницу входа

5 Применение

5.1 Назначение и область применения

Главным назначением сайта является продажа товаров, а конкретнее продажа компьютерных игр и различной сувенирной продукции. Помимо продажи товаров сайт необходим компании для продвижения компании среди аудитории, а также для поиска новых заказчиков. Дополнительно была разработана страница с вакансиями, которая поможет расширению компании путем поиска новых сотрудников. Область применения в основном связана с компьютерной индустрией, разработкой компьютерных игр, а также программированием.

5.2 Руководство пользователя

При первой загрузке главной страницы сайта, у пользователя появляется доступ к следующим страницам: Главная страница, игры, услуги, о нас, вакансии, а также страницы регистрации и входа. На данном этапе пользователю присвоена роль гость. Гость может просматривать страницу с товарами (Рисунок 5.1) и страницы товаров с описанием товаров. При попытке добавить товар в корзину гость будет перенаправлен на страницу входа, также как и в случае отправки сообщения разработчикам.

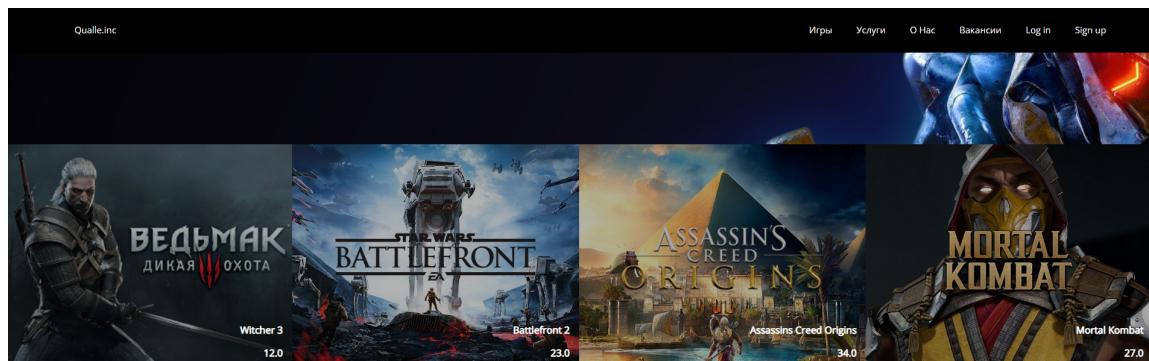


Рис. 5.1 - Страница с товарами

Гость также может воспользоваться поиском, при этом поиск производится методом фильтрации данных, т.е. пользователю необязательно знать точное название товара, программа сама будет фильтровать данные и

выводить соответствующие результаты на страницу. Изображение страницы представлено на рисунке 5.2.

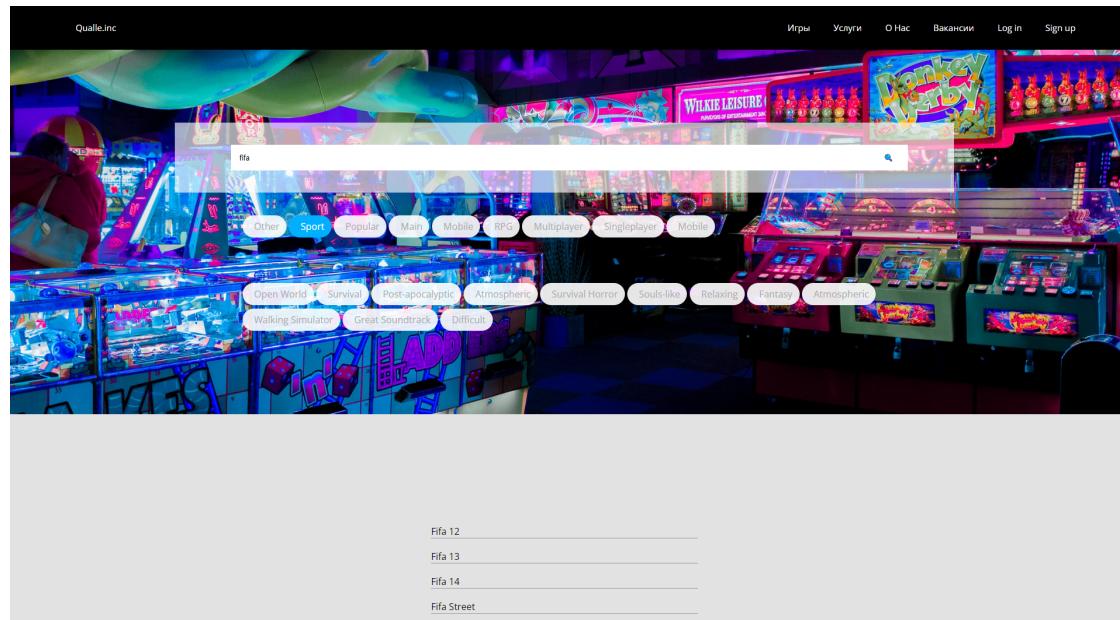


Рис. 5.2 - Страница с поиском товаров

После регистрации либо входа, гостю присваивается роль зарегистрированного пользователя, и он получает доступ к собственной корзине с товарами (Рисунок 5.3) а также профилю (Рисунок 5.4).

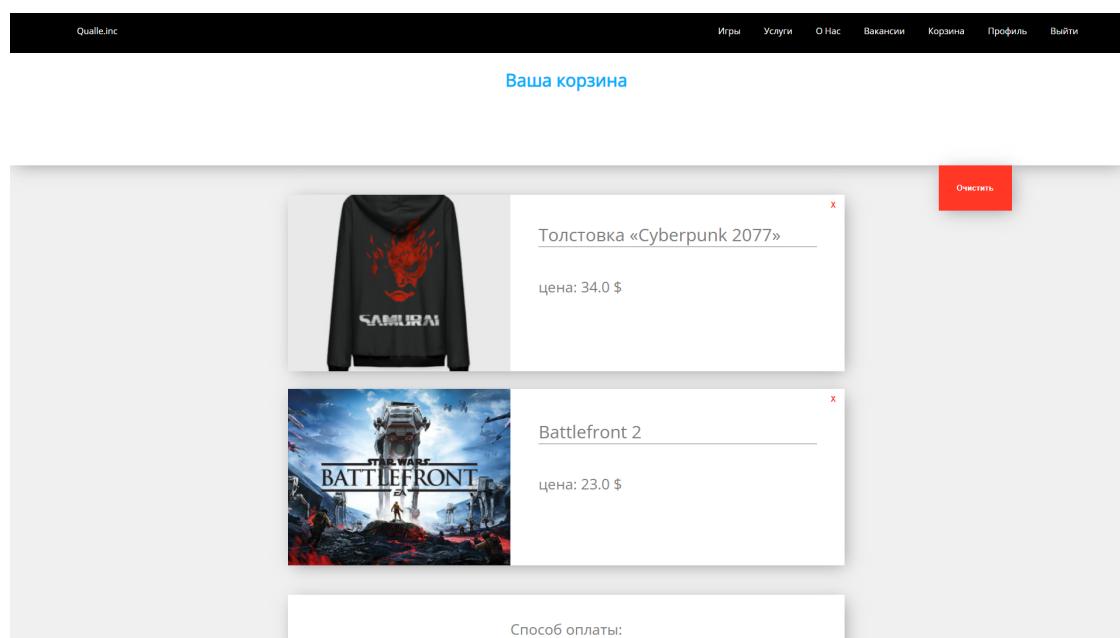


Рис. 5.3 - Корзина

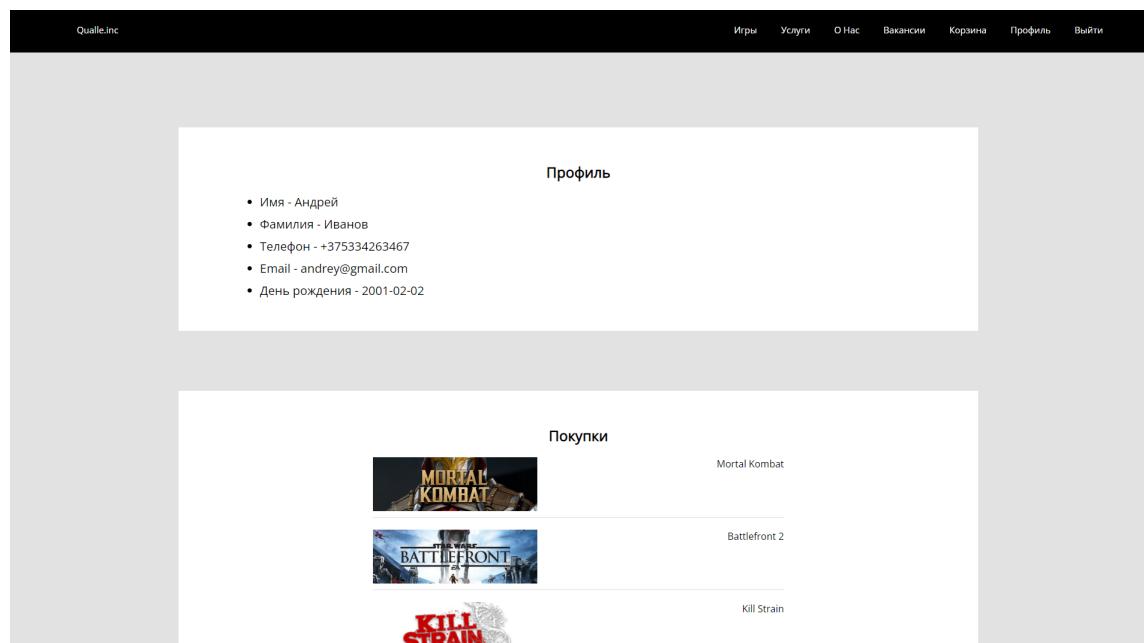


Рис. 5.4 - Профиль

Для того чтобы пользователь получил права администратора, необходимо что бы любой другой пользователь с ролью администратор вручную изменил роль через панель управления. Любой администратор имеет доступ к панели управления сайта (Рисунок 5.5).

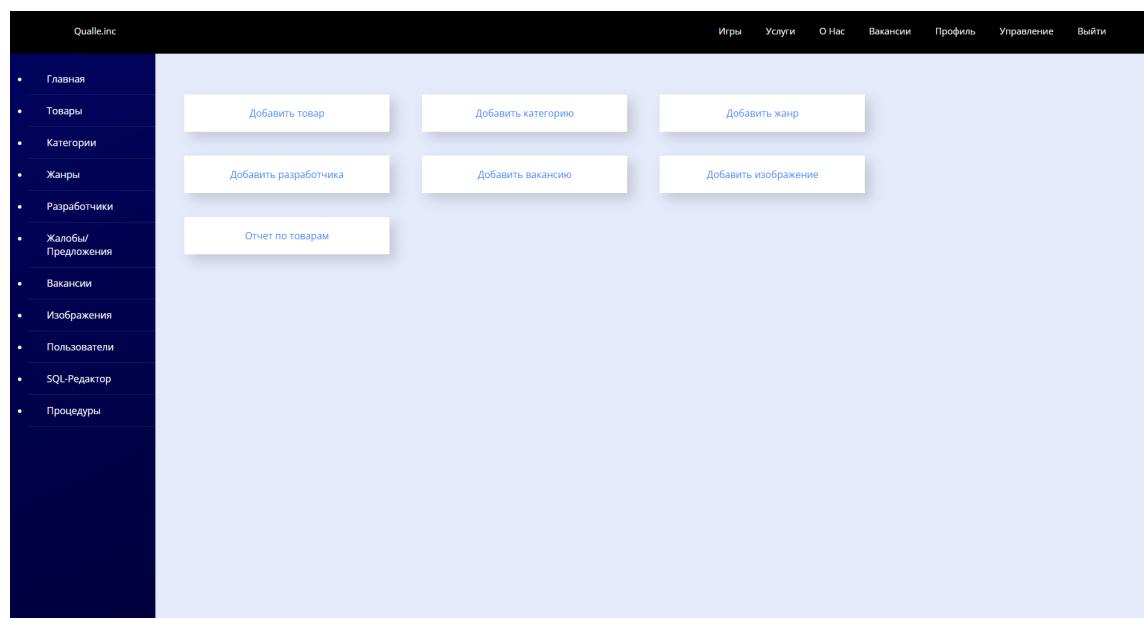


Рис. 5.5 - Панель управления

Через панель управления администратор имеет возможность редактировать любые данные, размещенные на сайте, а также выполнять процедуры и собственные запросы.

Заключение

Целью курсового проекта являлось создание базы данных для организации деятельности интернет-магазина компании по разработке компьютерных игр. В процессе выполнения курсового проекта была подробно изучена деятельность компании, а также поведение пользователей при использовании аналогичных продуктов, что позволило составить подробные требования к программному продукту. Перед началом разработки были проанализированы схожие по функционалу программы и были выявлены их достоинства и недостатки.

В процессе разработки программного продукта была спроектирована база данных, которая позволила хранить всю необходимую информацию. Особенностью данного продукта является то, что любую информацию можно редактировать в любой момент, а также все запросы на изменение данных выполняются как транзакции. Был использован `read committed` уровень изолированности транзакций, данный уровень имеет высокую скорость выполнения и среднюю согласованность данных, предоставляя защиту от чтения незафиксированных изменений другой транзакцией. Помимо этого, в программном продукте присутствует журнализация, что позволяет упростить процесс исправления неисправностей в случае ошибок. Также в базе данных было настроено ежедневное сохранение резервной копии.

Цель курсового проекта достигнута, была разработана база данных для организации деятельности интернет-магазина компании по разработке компьютерных игр. После завершения разработки, программный продукт был полностью протестирован. Также для конечных пользователей разработано подробное руководство по использованию приложения.

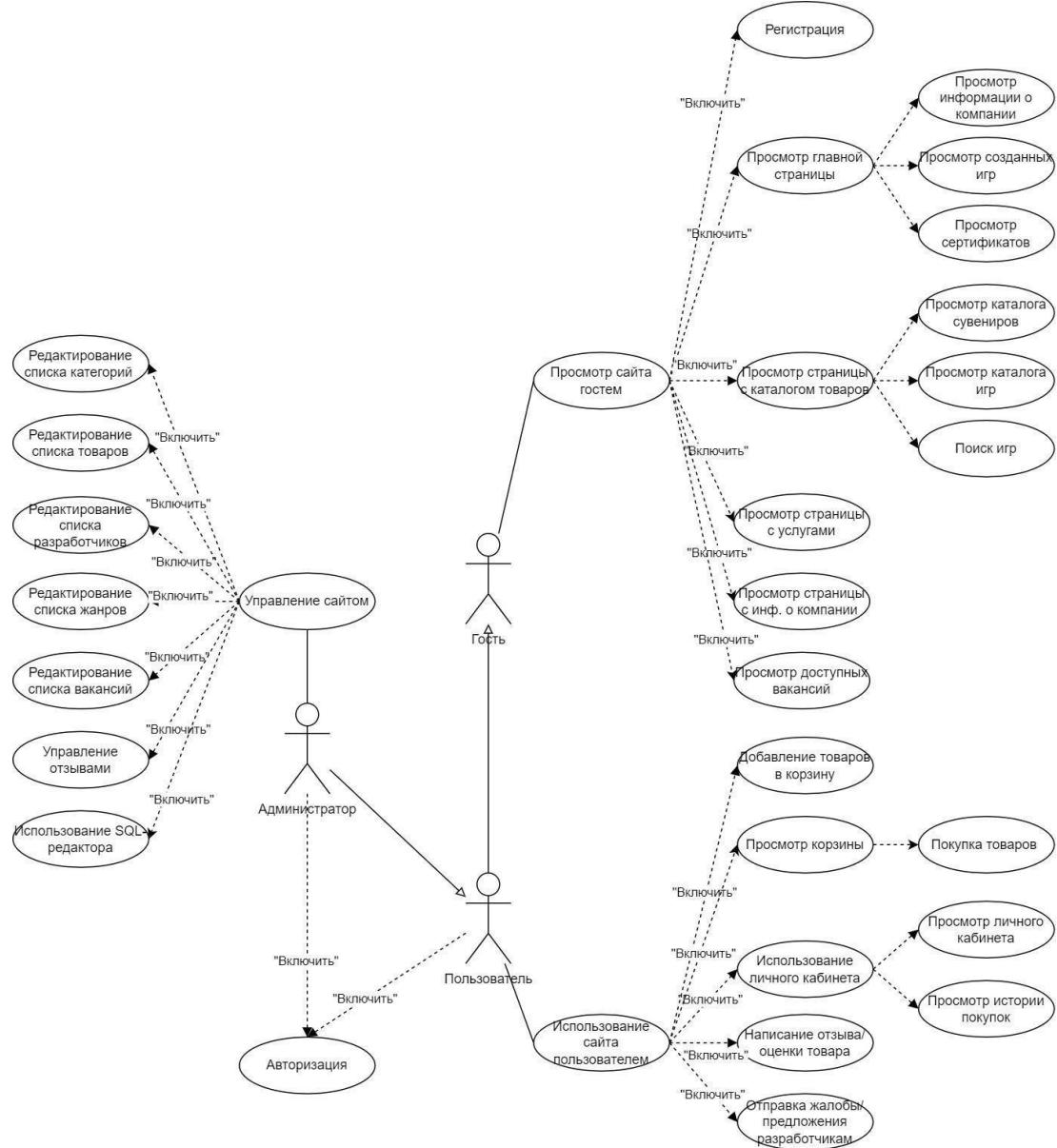
Разработанное программное обеспечение соответствует требованиям, и может быть использовано конечными пользователями.

Список использованных источников

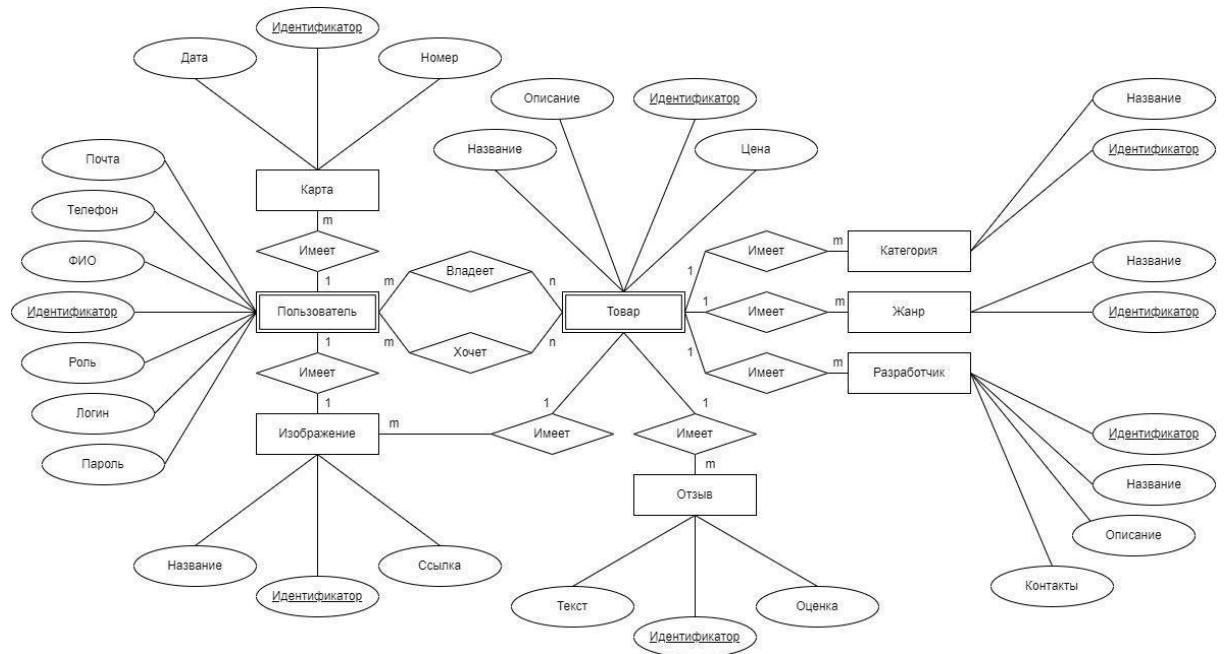
- [1] Введение в системы баз данных / К. Дж. Дейт. – Питер, 2018. – 472 с.
- [2] Документация по Java. [Электронный ресурс]. Oracle Help Center, 2022. – Режим доступа: <https://docs.oracle.com/en/java> Дата доступа: 15.03.2022
- [3] Руководство по Java. [Электронный ресурс]. Baeldung.com, 2022. – Режим доступа: <https://www.baeldung.com/> Дата доступа: 11.03.2022
- [4] Руководство по Spring Boot. [Электронный ресурс]. Spring Getting Started Guides, 2022. – Режим доступа: <https://spring.io/guides/> Дата доступа: 11.03.2022
- [5] Документация по MySQL. [Электронный ресурс]. MySQL Tutorial, 2022. – Режим доступа: <https://www.mysqltutorial.org/> Дата доступа: 11.03.2022
- [6] Модель «сущность-связь». [Электронный ресурс]. Учебно-методический комплекс БГЭУ, 2020. – Режим доступа: http://bseu.by/it/tohod/lekcii4_3.htm Дата доступа: 11.03.2022
- [7] Элементы модели сущность связь. [Электронный ресурс]. Lucidchart, 2022. – Режим доступа: <https://www.lucidchart.com/pages/ru/erd-диаграмма> Дата доступа: 11.03.2022
- [8] Диаграммы UML. [Электронный ресурс]. Evergreen, 2022. – Режим доступа: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> Дата доступа: 11.03.2022
- [9] Модель данных. [Электронный ресурс]. Интернет-технологии.ру, 2022. – Режим доступа: <https://www.internet-technologies.ru/articles/modeli-baz-dannyh-sistemy-upravleniya-bazami-dannyh.html> Дата доступа: 11.03.2022
- [10] Интеллектуальная карта сайта. [Электронный ресурс]. Интернет-маркетинг FireSE, 2022. – Режим доступа: <https://fireseo.ru/blog/struktura-sajta-primer-kak-sostavit-mindmap-sajta/> Дата доступа: 11.03.2022
- [11] Чистый код. Создание, анализ и рефакторинг / Роберт Мартин. – Питер, 2021. – 324 с.
- [12] Святослав Куликов. Тестирование программного обеспечения. Базовый курс / Святослав Куликов. – Четыре четверти, 2014. – 643 с.

Приложение А (обязательное)

Диаграмма вариантов использования

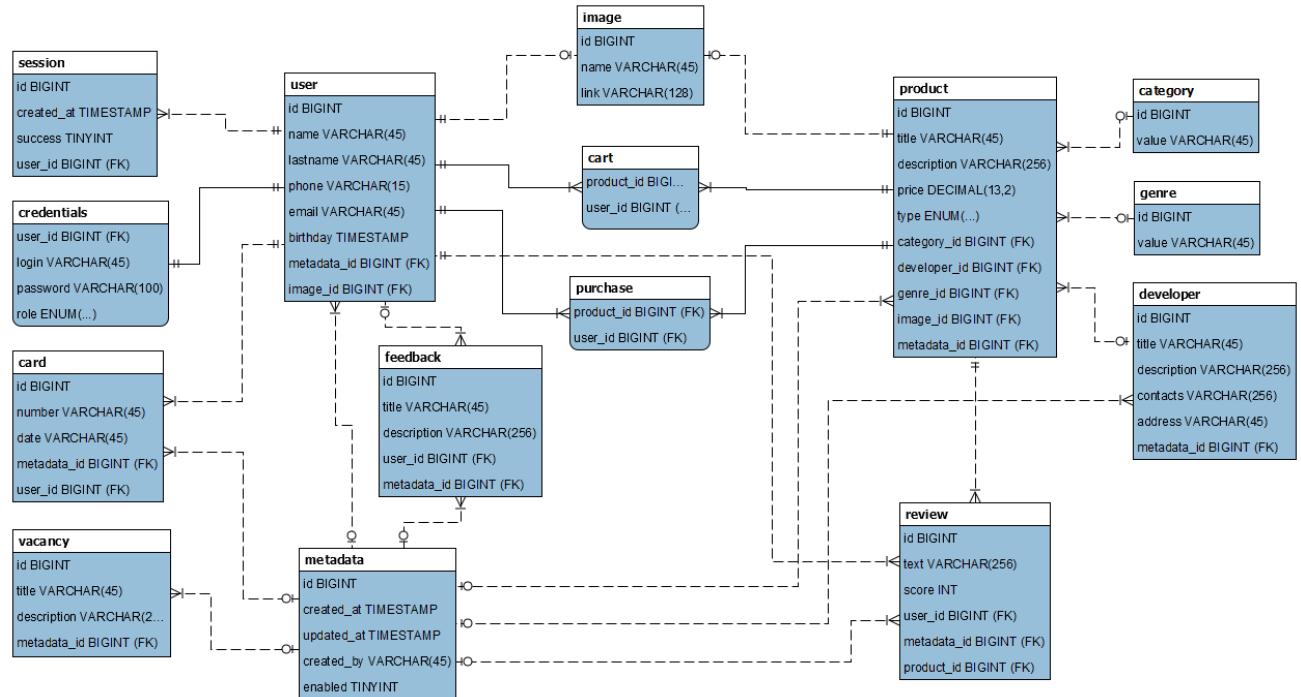


Приложение Б
(обязательное)
Диаграмма «сущность-связь»



Приложение В (обязательное)

Физическая модель данных



Приложение Г
(обязательное)
Листинг программы