

Содержание

Введение	4
1 Постановка задачи и обзор аналогичных решений	5
1.1 Постановка задачи	5
1.2 Аналитический обзор аналогов	5
1.2.1 Интернет-ресурс «Quiz»	5
1.2.2 Интернет-ресурс «Kahoot!»	6
1.2.3 Интернет-ресурс «Jackbox Games»	7
1.3 Выводы по разделу	9
2 Проектирование веб-приложения	10
2.1 Пользовательские роли и функциональные возможности веб-приложения ..	10
2.2 Логическая схема базы данных	13
2.3 Выбор технологий	19
2.3.1 Выбор стека для создания пользовательского интерфейса	19
2.3.2 Выбор стека для создания API	20
2.3.3 REST-архитектура	20
2.4 Архитектура веб-приложения	21
2.5 Выводы по разделу	22
Заключение	24
Список используемых источников	25

Введение

Прохождение практики в период с 10 февраля 2025 по 21 марта 2025 происходило в компании ООО «Модсен». Целью преддипломной практики является закрепление теоретических знаний и навыков, полученных в вузе, а также приобретение нового опыта, который пригодится в будущей профессиональной деятельности. Важным аспектом также является развитие практических навыков самостоятельной работы и умения применять их для решения конкретных задач.

Целью проекта является разработка функционального решения для обеспечения возможности проведения интеллектуальных игр и викторин.

Для достижения указанной цели поставлены следующие задачи:

1. Сформулировать постановку задачи и рассмотреть аналогичные решения (рассмотрено в разделе 1).
2. Спроектировать архитектуру приложения, обеспечивающую поддержку всех заявленных функций (рассмотрено в разделе 2).

Приложение ориентировано на широкую аудиторию, включая любителей интеллектуальных игр, активных пользователей социальных сетей, а также тех, кто заинтересован в саморазвитии и участии в командных или индивидуальных викторинах.

Разрабатываемое приложение представляет собой веб-приложение, доступное через интернет-браузеры на компьютерах, планшетах и смартфонах. Оно обеспечит взаимодействие пользователей с сервером и сохранение данных в базе данных.

В качестве технологической платформы используется последняя версия *ASP.NET Core* для серверной части и *React* с *Vite* для клиентской [1, 2, 3].

1 Постановка задачи и обзор аналогичных решений

1.1 Постановка задачи

Цель разработки веб-приложения для проведения интеллектуальных игр и викторин на платформе SaaS заключается в создании универсального решения, доступного для широкой аудитории пользователей через интернет. Основная задача приложения – предоставить пользователям возможность участвовать в разнообразных интеллектуальных играх и викторинах, а также отслеживать свои результаты и рейтинг. Приложение будет ориентировано как на индивидуальных пользователей, так и на организации, включая образовательные учреждения и компании, заинтересованные в проведении тренингов, соревнований и развлекательных мероприятий.

Приложение должно отвечать потребностям различных категорий пользователей, включая тех, кто ищет образовательный контент, и тех, кто хочет участвовать в развлекательных играх. Важным аспектом является создание гибкой архитектуры, которая обеспечит возможность использования платформы в разных контекстах, таких как корпоративные тренинги, образовательные мероприятия и развлекательные игры для широкой аудитории.

1.2 Аналитический обзор аналогов

1.2.1 Интернет-ресурс «Quiz»

«Quiz» – это онлайн-платформа, предназначенная для создания и прохождения интерактивных викторин [4].

Интерфейс интернет-ресурса «Quiz» представлен на рисунке 1.1.

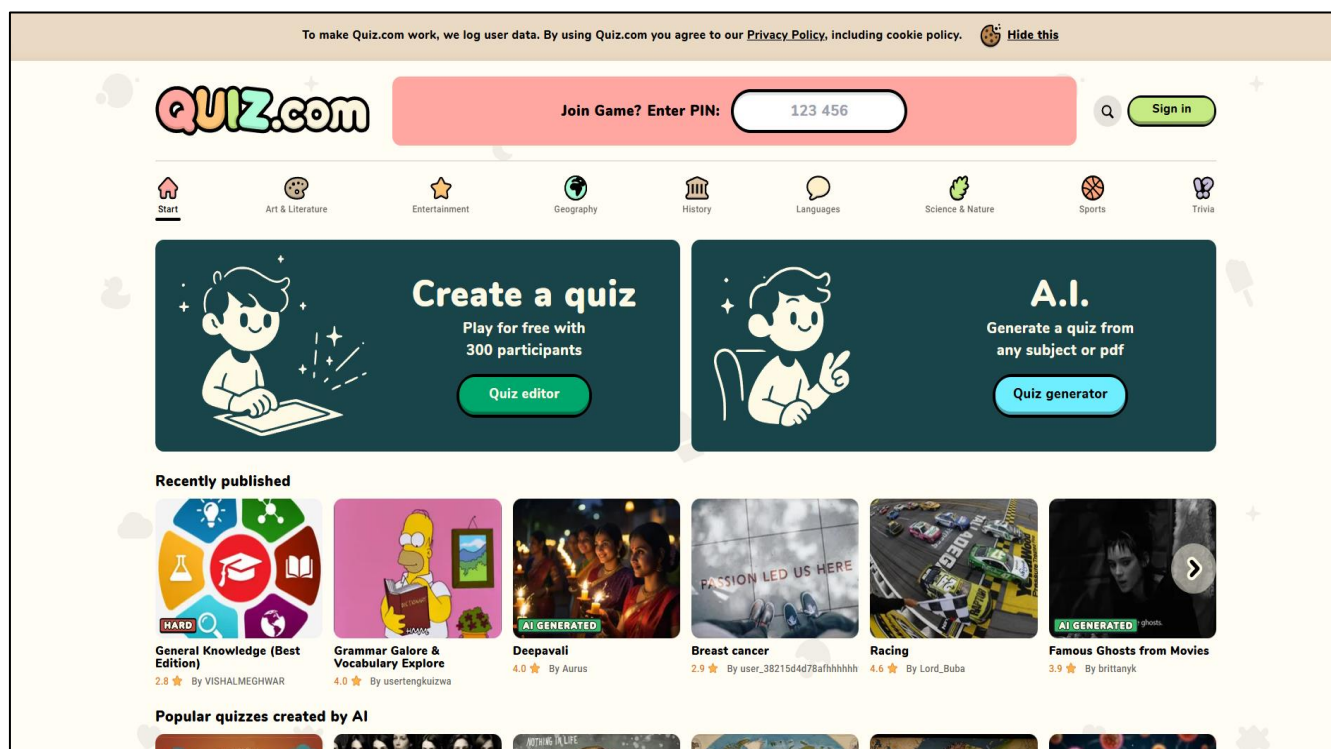


Рисунок 1.1 – Интерфейс интернет-ресурса «Quiz»

Пользователи могут выбирать из широкого ассортимента готовых викторин или создавать собственные, используя разнообразные настройки для персонализации. Платформа ориентирована на развлечение и обучение, что делает её популярной как среди учащихся, так и в корпоративной среде.

Достоинства:

- Широкий выбор тем: «Quiz» предлагает викторины на самые разные темы, от науки и искусства до поп-культуры и спорта, что позволяет привлечь интересы широкого круга пользователей.

- Возможность создания собственных викторин: пользователи могут легко создавать уникальные тесты, что делает платформу удобной для преподавателей и организаторов мероприятий.

- Интерактивность и мгновенные результаты: платформа предоставляет мгновенные результаты, что делает процесс прохождения викторин более увлекательным и познавательным.

Недостатки:

- Ограниченные социальные функции: платформа имеет ограниченные возможности для взаимодействия пользователей вне процесса прохождения викторин.

- Зависимость от интернета: «Quiz» требует стабильного интернет-соединения, что может быть неудобным в условиях ограниченного доступа к сети.

Еще одним важным аспектом платформы «Quiz» является её доступность на различных устройствах. Пользователи могут проходить викторины как на настольных компьютерах, так и на мобильных устройствах, что делает процесс взаимодействия более удобным. Адаптивный дизайн позволяет комфортно использовать платформу на экранах разных размеров, обеспечивая плавную навигацию и удобный интерфейс. Это особенно важно для образовательных учреждений и корпоративных пользователей, которым требуется гибкость при проведении викторин и тестов в различных условиях.

1.2.2 Интернет-ресурс «Kahoot!»

«Kahoot!» – образовательная платформа для создания викторин, которая часто используется в школах и университетах для проведения интерактивных занятий [5]. Она ориентирована на группы участников, что делает её идеальной для обучения и совместных мероприятий.

Помимо этого, «Kahoot!» предлагает интеграцию с различными образовательными сервисами, такими как *Google Classroom* и *Microsoft Teams*, что облегчает использование платформы в учебном процессе. Также предусмотрена возможность анализа результатов, что позволяет преподавателям отслеживать прогресс учеников. Однако ограниченный набор настроек для создания викторин и зависимость от стабильного интернет-соединения могут стать препятствием для использования в некоторых условиях.

Интерфейс интернет-ресурса «Kahoot!» представлен на рисунке 1.2.



Рисунок 1.2 – Интерфейс интернет-ресурса «Kahoot!»

Достоинства:

- Поддержка живых игр: «Kahoot!» позволяет проводить викторины в режиме реального времени, что делает процесс участия более динамичным и интерактивным.

- Ориентированность на образование: платформа широко используется в образовательных учреждениях для тестирования знаний и вовлечения учеников.

- Простота создания игр: пользователи могут легко создавать свои викторины и адаптировать их под нужды учебного процесса.

Недостатки:

- Ограниченные возможности персонализации: несмотря на возможность создания собственных викторин, пользователи не могут изменять многие аспекты интерфейса или игровых сценариев.

- Фокус на группе: платформа больше подходит для групповых мероприятий, чем для индивидуального использования.

1.2.3 Интернет-ресурс «Jackbox Games»

«Jackbox Games» – это набор многопользовательских игр, которые можно играть на различных устройствах с участием большого количества людей [6]. Игры отличаются оригинальными сценариями и креативными задачами, ориентированными на взаимодействие с друзьями.

Еще одной важной особенностью «Jackbox Games» является простота подключения – для участия в игре достаточно ввести код комнаты на официальном

сайте, используя любое устройство с браузером. Это делает платформу удобной для проведения онлайн-мероприятий, вечеринок и корпоративных тимбилдингов. Кроме того, игры не требуют сложных настроек и могут быть быстро запущены даже для новых пользователей. Однако отсутствие локализации на многие языки может стать препятствием для некоторых игроков, а необходимость отдельной покупки каждого игрового набора делает использование платформы менее доступным для широкой аудитории.

Интерфейс интернет-ресурса «*Jackbox Games*» представлен на рисунке 1.3.

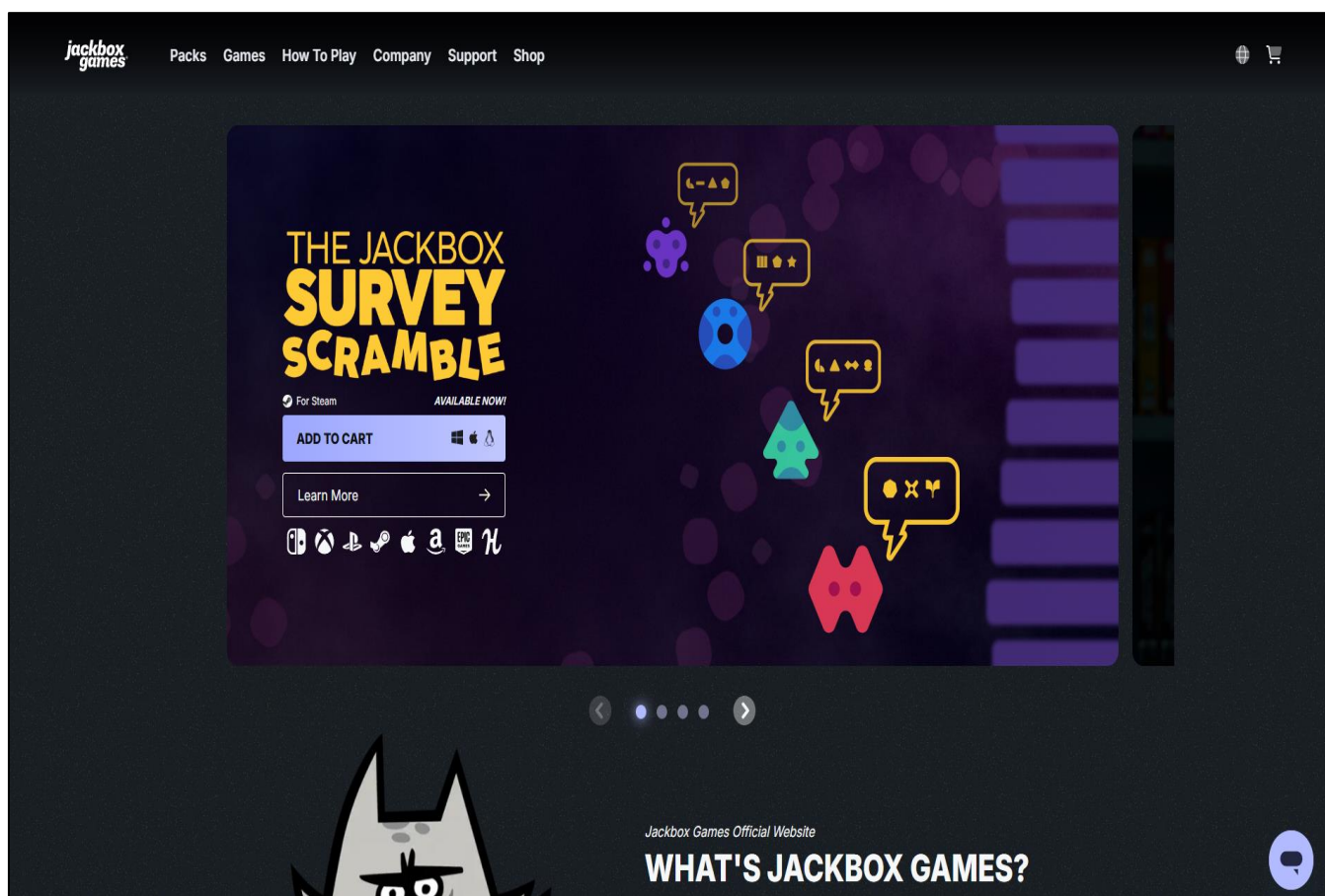


Рисунок 1.3 – Интерфейс интернет-ресурса «*Jackbox Games*»

Достоинства:

- Веселые и креативные игры: «*Jackbox Games*» предлагает оригинальные игровые сценарии, которые делают каждый игровой сеанс уникальным.
- Поддержка большого числа участников: играть в «*Jackbox Games*» можно в больших компаниях, что делает его идеальным для вечеринок и социальных мероприятий.
- Мульти-платформенность: игры можно запускать на любом устройстве, а участники подключаются через свои смартфоны.

Недостатки:

- Отсутствие социальных функций: платформа не предоставляет возможностей для взаимодействия пользователей вне игрового процесса.
- Ограниченное количество игр: несмотря на разнообразие игр в каждом наборе, их количество ограничено, и игроки могут быстро потерять интерес.

1.3 Выводы по разделу

В ходе анализа существующих решений, таких как «*Quiz*», «*Kahoot!*» и «*Jackbox Games*», были выявлены их сильные и слабые стороны. «*Quiz*» предоставляет простую платформу для создания викторин. «*Kahoot!*» ориентирован на интерактивность и образовательное использование. «*Jackbox Games*» фокусируется на развлекательных многопользовательских играх.

Основной задачей разработки веб-приложения является создание платформы с поддержкой ролей «Гость», «Пользователь», «Ведущий» и «Администратор», а также реализация функциональной части для взаимодействия с викторинами и игровыми комнатами, управления профилями и данными пользователей.

Разрабатываемое приложение объединяет сильные стороны аналогов, расширяя функционал за счет интеграции элементов взаимодействия между пользователями и ведения интеллектуальных игр.

2 Проектирование веб-приложения

2.1 Пользовательские роли и функциональные возможности веб-приложения

В системе предусмотрены следующие роли: «Гость», «Пользователь», «Ведущий» и «Администратор». Каждая из этих ролей имеет определённый набор прав и ограничений, в зависимости от своего набора функций. Для лучшего понимания взаимодействия ролей и функций, была построена диаграмма вариантов использования, которая отражает основные действия пользователей в приложении. Она отображает ключевые сценарии использования, охватывая все важные процессы и действия. Диаграмма вариантов использования представлена на рисунке 2.1 и также продублирована в приложении А.

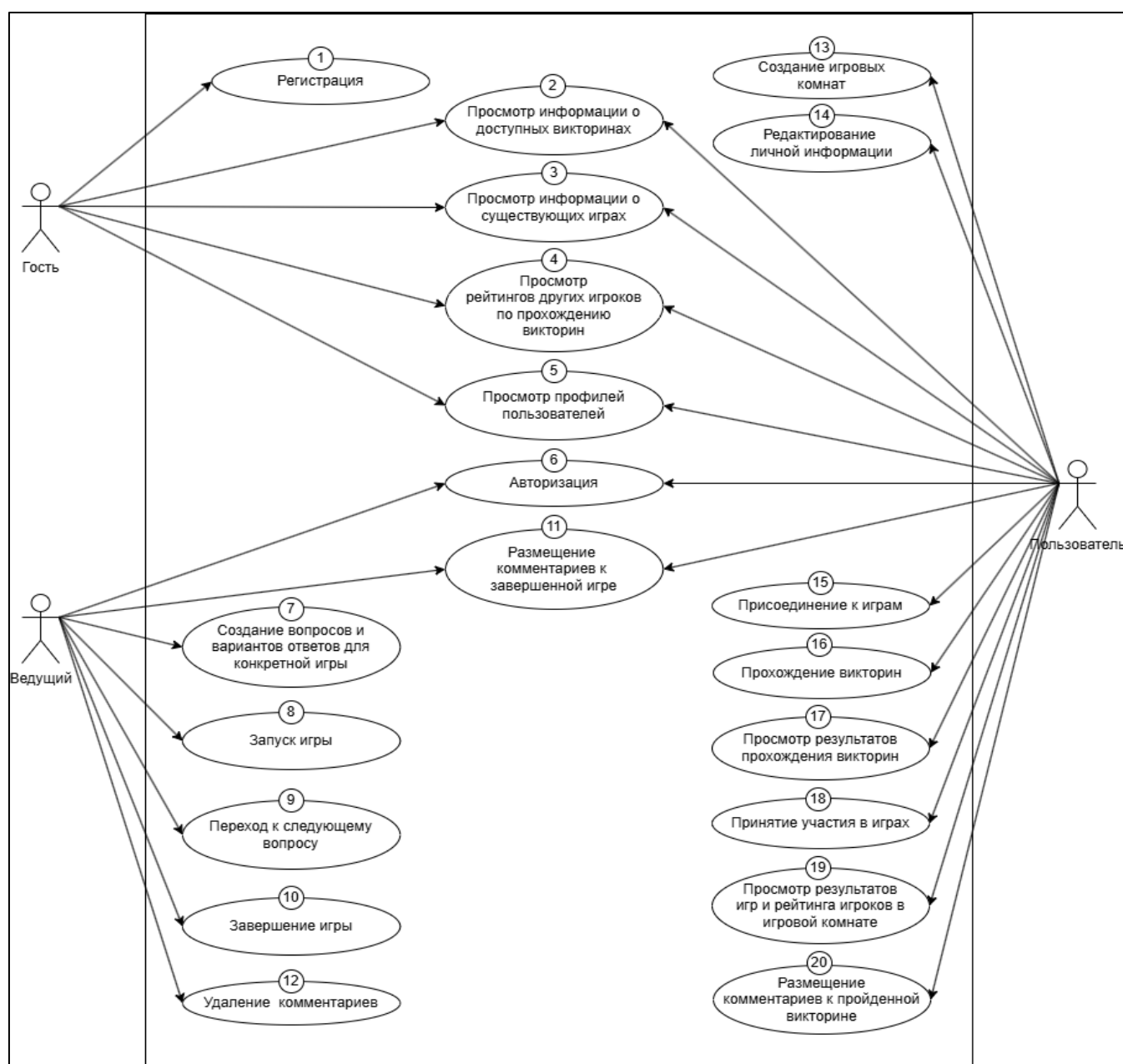


Рисунок 2.1 – Диаграмма вариантов использования

В таблице 2.1 представлено описание ролей.

Таблица 2.1 – Описание ролей

Роль	Описание роли
Гость	Пользователь, не авторизованный в системе, имеющий доступ к ограниченному количеству функций. Может просматривать информацию о викторинах, играх, рейтингах и профилях пользователей. Не может создавать игровые комнаты или участвовать в играх
Пользователь	Авторизованный пользователь, имеющий расширенные права. Может просматривать информацию о викторинах, играх, профилях пользователей, создавать игровые комнаты, редактировать личную информацию, участвовать в играх, проходить викторины, оставлять комментарии и просматривать результаты игровых сессий
Ведущий	Пользователь с расширенными правами для проведения игр. Может создавать вопросы и варианты ответов, запускать и завершать игру, переходить между вопросами, а также удалять комментариями к завершенным играм. Ведущий несет ответственность за управление игровым процессом
Администратор	Авторизованный пользователь, имеющий возможность управлять другими пользователями, контролировать контент, создавать и редактировать викторины и игры, а также модерацию комментариев и результатов. Администратор может назначать роли пользователям, удалять учетные записи, а также управлять системой

В таблице 2.2 представлены все функции, их номера, наименования и описания, а также соответствующие роли пользователей.

Таблица 2.2 – Описание функций

№	Роль	Наименование функции	Описание функций
1	Гость	Регистрация	Процесс создания учетной записи пользователя для доступа к системе
2	Гость, Пользователь	Просмотр информации о доступных викторинах	Отображение списка доступных викторин с кратким описанием
3	Гость, Пользователь	Просмотр информации о существующих играх	Получение сведений о текущих играх, включая их статус
4	Гость, Пользователь	Просмотр рейтингов других игроков по прохождению викторин	Ознакомление с таблицей лидеров, содержащей данные о лучших результатах игроков
5	Гость, Пользователь	Просмотр профилей пользователей	Получение информации о других пользователях, включая их достижения и статистику

№	Роль	Наименование функции	Описание функций
6	Пользователь, Ведущий	Авторизация	Вход в систему с использованием учетных данных пользователя
7	Ведущий	Создание вопросов и вариантов ответов для конкретной игры	Процесс добавления вопросов с вариантами ответов для проведения игры
8	Ведущий	Запуск игры	Инициация начала игры для участников
9	Ведущий	Переход к следующему вопросу	Переход к следующему вопросу игры
10	Ведущий	Завершение игры	Фиксация результата игры и завершение игрового процесса
11	Пользователь, Ведущий	Размещение комментариев к завершенной игре	Добавление отзывов или замечаний к завершенной игре
12	Ведущий	Удаление комментариев	Удаление ранее опубликованных комментариев к играм
13	Пользователь	Создание игровых комнат	Создание игровых комнат для проведения игр
14	Пользователь	Редактирование личной информации	Изменение данных профиля, включая имя, фамилию и картинку пользователя
15	Пользователь	Присоединение к играм	Возможность присоединения к игровой комнате
16	Пользователь	Прохождение викторин	Участие в викторинах, включая ответ на вопросы и завершение викторин
17	Пользователь	Просмотр результатов прохождения викторин	Получение данных о своих ответах, правильности и общем результате викторины
18	Пользователь	Принятие участия в играх	Возможность пользователя быть участником игры
19	Пользователь	Просмотр результатов игр и рейтинга игроков в игровой комнате	Анализ итогов завершенных игр и сравнение с результатами других участников игровой комнаты
20	Пользователь	Размещение комментариев к пройденной викторине	Возможность оставить отзыв о конкретной викторине, в которой пользователь принимал участие

Проектирование функциональности веб-приложения позволяет создать удобный и интуитивно понятный интерфейс, удовлетворяющий потребности различных групп пользователей. Диаграмма вариантов использования и таблицы ролей помогают визуализировать и структурировать ключевые процессы

взаимодействия с системой. Такое структурирование упрощает разработку и тестирование приложения, а также улучшает его масштабируемость и возможность добавления новых функций в будущем.

2.2 Логическая схема базы данных

Структура базы данных состоит из четырнадцати таблиц, каждая из которых отвечает за хранение и управление определенными данными. Логическая схема базы данных представлена на рисунке 2.2 и также продублирована в приложении Б.

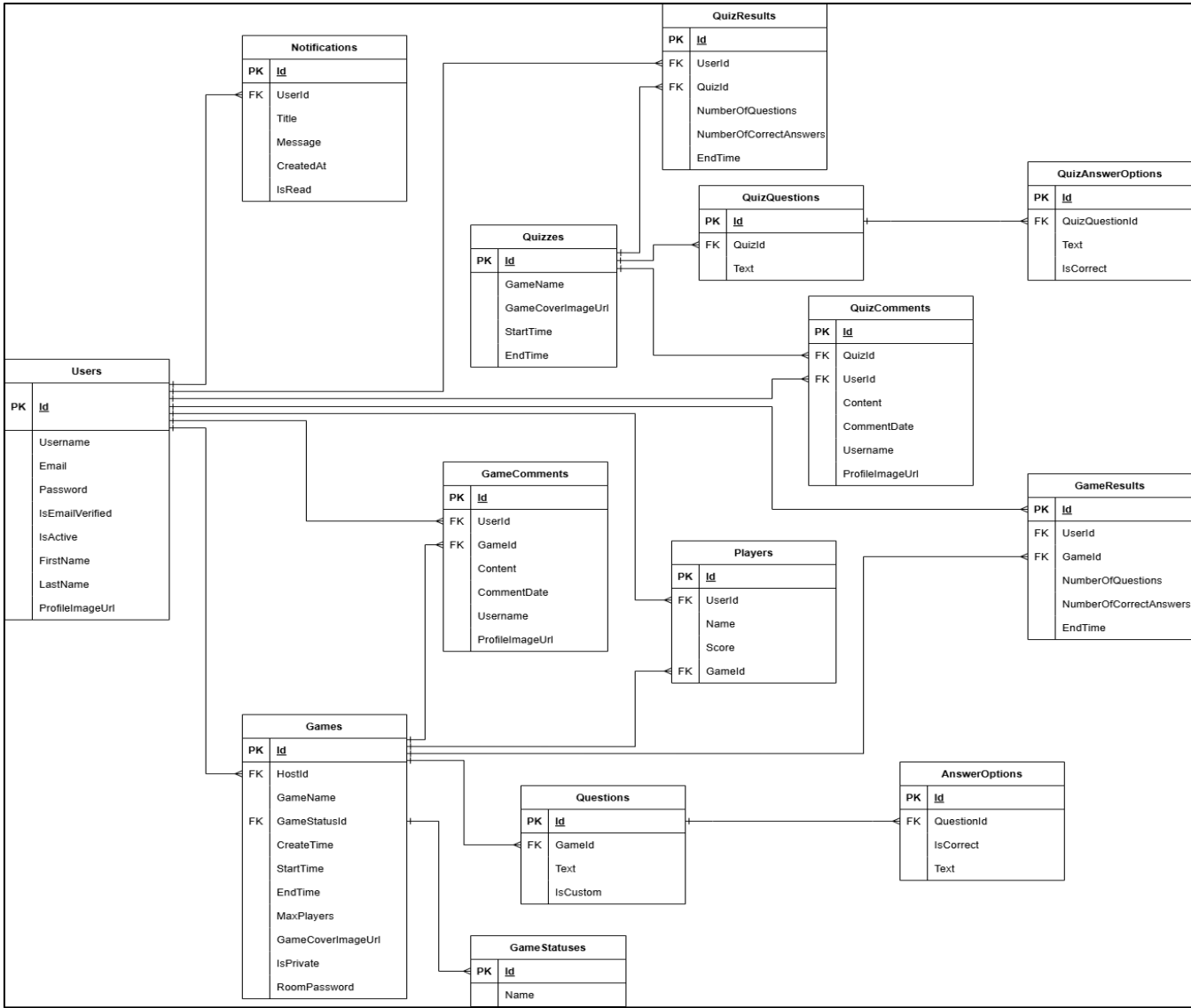


Рисунок 2.2 – Логическая схема базы данных

В таблице 2.3 предоставлена краткая информация о каждой таблице, включая наименования и описание все сущностей базы данных. Эти таблицы взаимодействуют между собой с помощью внешних ключей, обеспечивая целостность данных и поддержку ключевых функциональных возможностей системы, таких как управление пользователями, хранение викторин, вопросов и ответов, а также отслеживание прогресса и результатов.

Таблица 2.3 – Таблицы базы данных

Название таблицы	Описание использования
<i>AnswerOptions</i>	Хранит варианты ответов для вопросов, включая текст ответа, его корректность и связь с вопросами
<i>GameComments</i>	Содержит комментарии пользователей к играм, включая текст комментария, имя пользователя, аватар и дату
<i>GameResults</i>	Хранит результаты игр для пользователей, включая количество правильных ответов и общее количество вопросов
<i>Games</i>	Содержит информацию об играх, таких как название, ведущий, время начала/окончания, статус, приватность и пароль
<i>GameStatuses</i>	Хранит возможные статусы игр
<i>Notifications</i>	Содержит уведомления для пользователей, включая заголовок, текст сообщения, дату создания и статус прочтения
<i>Players</i>	Хранит данные о игроках игр, включая имя, счёт, связь с игрой и пользователем
<i>Questions</i>	Содержит текст вопросов для игр, а также информацию о кастомных вопросах
<i>QuizAnswerOptions</i>	Хранит варианты ответов для викторин, включая текст ответа и информацию о корректности
<i>QuizComments</i>	Содержит комментарии пользователей к викторинам, включая текст комментария, имя пользователя и дату
<i>QuizQuestions</i>	Хранит вопросы викторин с текстом и привязкой к конкретной викторине
<i>QuizResults</i>	Содержит результаты прохождения викторин, включая количество правильных ответов, время завершения и пользователя
<i>Quizzes</i>	Хранит информацию о викторинах, включая название, обложку, время начала/окончания
<i>Users</i>	Хранит информацию о пользователях

AnswerOptions – хранит варианты ответов для вопросов, включая текст ответа, его корректность и связь с вопросами. Её структура представлена в таблице 2.4.

Таблица 2.4 – Структура таблицы *AnswerOptions*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор варианта ответа
<i>Text</i>	<i>NVARCHAR(200)</i>	Текст варианта ответа
<i>IsCorrect</i>	<i>BIT</i>	Признак корректности ответа
<i>QuestionId</i>	<i>INT</i>	Идентификатор связанного вопроса, к которому относится данный ответ

GameComments – содержит комментарии пользователей к играм, включая текст комментария, имя пользователя, аватар и дату. Структура представлена в таблице 2.5.

Таблица 2.5 – Структура таблицы *GameComments*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор комментария
<i>GameId</i>	<i>INT</i>	Идентификатор игры, к которой относится комментарий
<i>Username</i>	<i>NVARCHAR(MAX)</i>	Имя пользователя, оставившего комментарий
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя
<i>Content</i>	<i>NVARCHAR(500)</i>	Текст комментария
<i>CommentDate</i>	<i>DATETIME2(7)</i>	Дата и время создания комментария
<i>ProfileImageUrl</i>	<i>NVARCHAR(MAX)</i>	Ссылка на аватар пользователя

GameResults – хранит результаты игр для пользователей, включая количество правильных ответов и общее количество вопросов. Структура представлена в таблице 2.6.

Таблица 2.6 – Структура таблицы *GameResults*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор результата игры
<i>NumberOfCorrectAnswers</i>	<i>INT</i>	Количество правильных ответов
<i>NumberOfQuestions</i>	<i>INT</i>	Общее количество вопросов
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя, к которому относится результат
<i>GameId</i>	<i>INT</i>	Идентификатор игры, связанной с результатом

Games – содержит информацию об играх, таких как название, ведущий, время начала/окончания, статус, приватность и пароль. Её структура представлена в таблице 2.7.

Таблица 2.7 – Структура таблицы *Games*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор игры.
<i>GameName</i>	<i>NVARCHAR(100)</i>	Название игры
<i>HostId</i>	<i>INT</i>	Идентификатор ведущего игры
<i>CreateTime</i>	<i>DATETIME2(7)</i>	Время создания игры
<i>StartTime</i>	<i>DATETIME2(7)</i>	Время начала игры
<i>EndTime</i>	<i>DATETIME2(7)</i>	Время окончания игры
<i>MaxPlayers</i>	<i>INT</i>	Максимальное количество участников

Окончание таблицы 2.7

Название	Тип данных	Назначение
<i>GameCoverImageUrl</i>	<i>NVARCHAR(MAX)</i>	Ссылка на изображение обложки игры
<i>IsPrivate</i>	<i>BIT</i>	Признак приватности игры
<i>RoomPassword</i>	<i>NVARCHAR(50)</i>	Пароль для доступа к приватной игре
<i>GameStatusId</i>	<i>INT</i>	Идентификатор текущего статуса игры

GameStatuses – содержит возможные статусы игр. Структура представлена в таблице 2.8.

Таблица 2.8 – Структура таблицы *GameStatuses*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор статуса
<i>Name</i>	<i>NVARCHAR(100)</i>	Название статуса игры

Notifications – содержит уведомления для пользователей, включая заголовок, текст сообщения, дату создания и статус прочтения. Структура представлена в таблице 2.9.

Таблица 2.9 – Структура таблицы *Notifications*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор уведомления
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя, получившего уведомление
<i>Title</i>	<i>NVARCHAR(100)</i>	Заголовок уведомления
<i>Message</i>	<i>NVARCHAR(1000)</i>	Текст сообщения уведомления
<i>CreatedAt</i>	<i>DATETIME2(7)</i>	Дата и время создания уведомления
<i>IsRead</i>	<i>BIT</i>	Признак прочтения уведомления

Players – Хранит данные о игроках игр, включая имя, счёт, связь с игрой и пользователем. Структура таблицы представлена в таблице 2.10.

Таблица 2.10 – Структура таблицы *Players*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор игрока
<i>GameId</i>	<i>INT</i>	Идентификатор игры, к которой относится игрок
<i>Name</i>	<i>NVARCHAR(100)</i>	Имя игрока
<i>Score</i>	<i>INT</i>	Текущий счёт игрока
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя, связанного с игроком

Questions – содержит информацию о вопросах, связанных с играми. Структура представлена в таблице 2.11.

Таблица 2.11 – Структура таблицы *Questions*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор вопроса
<i>Text</i>	<i>NVARCHAR(500)</i>	Текст вопроса
<i>GameId</i>	<i>INT</i>	Идентификатор игры, к которой относится вопрос
<i>IsCustom</i>	<i>BIT</i>	Признак кастомного вопроса

QuizAnswerOptions – хранит варианты ответов для викторин, включая текст ответа и информацию о корректности. Структура представлена в таблице 2.12.

Таблица 2.12 – Структура таблицы *QuizAnswerOptions*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор варианта ответа для викторины
<i>Text</i>	<i>NVARCHAR(200)</i>	Текст варианта ответа
<i>IsCorrect</i>	<i>BIT</i>	Признак корректности ответа
<i>QuizQuestionId</i>	<i>INT</i>	Идентификатор вопроса викторины, связанного с ответом

QuizComments – Содержит комментарии пользователей к викторинам, включая текст комментария, имя пользователя и дату. Структура представлена в таблице 2.13.

Таблица 2.13 – Структура таблицы *QuizComments*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор комментария к викторине
<i>QuizId</i>	<i>INT</i>	Идентификатор викторины, к которой относится комментарий
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя, оставившего комментарий
<i>Content</i>	<i>NVARCHAR (500)</i>	Текст комментария
<i>Username</i>	<i>NVARCHAR (max)</i>	Имя пользователя, оставившего комментарий
<i>ProfileImageUrl</i>	<i>NVARCHAR (max)</i>	Ссылка на аватар пользователя
<i>CommentDate</i>	<i>DATETIME2(7)</i>	Дата и время создания комментария

QuizQuestions – содержит вопросы викторины. Структура представлена в таблице 2.14.

Таблица 2.14 – Структура таблицы *QuizQuestions*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор вопроса викторины
<i>Text</i>	<i>NVARCHAR(500)</i>	Текст вопроса викторины
<i>QuizId</i>	<i>INT</i>	Идентификатор викторины, к которой относится вопрос

QuizResults – таблица для хранения результатов викторины. Структура представлена в таблице 2.15.

Таблица 2.15 – Структура таблицы *QuizResults*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор результата викторины
<i>NumberOfCorrectAnswers</i>	<i>INT</i>	Количество правильных ответов
<i>NumberOfQuestions</i>	<i>INT</i>	Общее количество вопросов в викторине
<i>EndTime</i>	<i>DATETIME2(7)</i>	Дата и время завершения викторины
<i>UserId</i>	<i>INT</i>	Идентификатор пользователя, связанного с результатом
<i>QuizId</i>	<i>INT</i>	Идентификатор викторины, связанной с результатом

Quizzes – хранит информацию о викторинах, связанных с играми. Структура представлена в таблице 2.16.

Таблица 2.16 – Структура таблицы *Quizzes*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор викторины
<i>GameName</i>	<i>NVARCHAR(200)</i>	Название викторины
<i>GameCoverImageUrl</i>	<i>NVARCHAR(500)</i>	Ссылка на изображение обложки викторины
<i>StartTime</i>	<i>DATETIME2(7)</i>	Дата и время начала викторины
<i>EndTime</i>	<i>DATETIME2(7)</i>	Дата и время окончания викторины

Users – хранит информацию о пользователях, зарегистрированных в системе. Структура представлена в таблице 2.17.

Таблица 2.17– Структура таблицы *Users*

Название	Тип данных	Назначение
<i>Id</i>	<i>INT</i>	Уникальный идентификатор пользователя
<i>Email</i>	<i>NVARCHAR(MAX)</i>	Адрес электронной почты пользователя
<i>Username</i>	<i>NVARCHAR(MAX)</i>	Имя пользователя
<i>Password</i>	<i>NVARCHAR(MAX)</i>	Пароль пользователя
<i>IsEmailVerified</i>	<i>BIT</i>	Признак подтверждения электронной почты
<i>IsActive</i>	<i>BIT</i>	Признак активности пользователя
<i>FirstName</i>	<i>NVARCHAR(MAX)</i>	Имя пользователя
<i>LastName</i>	<i>NVARCHAR(MAX)</i>	Фамилия пользователя
<i>ProfileImageUrl</i>	<i>NVARCHAR(MAX)</i>	Ссылка на аватар пользователя

Помимо самих таблиц базы данных, были определены связи этих таблиц. Так, каждая таблица, которая имеет столбец, значения в котором, зависят от значений в другом столбце, этой или другой таблицы, образует связь. В таблице 2.18 представлены связи между таблицами базы данных.

Таблица 2.18 – Связи между таблицами базы данных

Таблица источник	Связанная таблица	Тип связи
<i>Users</i>	<i>GameComments</i>	Один ко многим
<i>Users</i>	<i>GameResults</i>	Один ко многим
<i>Users</i>	<i>Players</i>	Один ко многим
<i>Users</i>	<i>Notifications</i>	Один ко многим
<i>Users</i>	<i>QuizComments</i>	Один ко многим
<i>Users</i>	<i>QuizResults</i>	Один ко многим
<i>Users</i>	<i>Games</i>	Один ко многим
<i>Games</i>	<i>GameComments</i>	Один ко многим
<i>Games</i>	<i>GameResults</i>	Один ко многим
<i>Games</i>	<i>Questions</i>	Один ко многим
<i>Games</i>	<i>Players</i>	Один ко многим
<i>GameStatuses</i>	<i>Games</i>	Один ко многим
<i>Questions</i>	<i>AnswerOptions</i>	Один ко многим
<i>Quizzes</i>	<i>QuizQuestions</i>	Один ко многим
<i>Quizzes</i>	<i>QuizComments</i>	Один ко многим
<i>Quizzes</i>	<i>QuizResults</i>	Один ко многим
<i>QuizQuestions</i>	<i>QuizAnswerOptions</i>	Один ко многим

Читать данную таблицу следует как: связанная таблица ссылается на таблицу источник таким образом, что одной записи в таблице источнике соответствует одна или множество записей в связанной таблице, в зависимости от типа связи.

Таким образом, представленные таблицы и связи формируют основу базы данных, обеспечивая возможность гибкого хранения и обработки всех данных, необходимых для функционирования системы.

2.3 Выбор технологий

2.3.1 Выбор стека для создания пользовательского интерфейса

Для разработки пользовательского интерфейса веб-приложения было принято решение использовать библиотеку *React 18.3.1*.

Основные причины данного выбора:

- компонентный подход: позволяет создавать *UI* на основе переиспользуемых компонентов;
- высокая производительность: использует виртуальный *DOM* для эффективного обновления интерфейса, минимизируя количество изменений в реальном *DOM* и повышая производительность приложения;
- гибкость и кроссплатформенность;

– большая экосистема и сообщество: имеет широкую поддержку со стороны разработчиков, множество готовых библиотек и инструментов для работы с *UI*, что позволяет ускорить процесс разработки.

2.3.2 Выбор стека для создания API

Для разработки серверного приложения (*API*) было решено использовать *ASP.NET Core Web API 8.0.400* в сочетании с *Entity Framework 8.0.8* и *MS SQL Server 2019 (RTM) 15.0.2000.5* для работы с базой данных [7, 8].

Основные причины выбора *ASP.NET Core Web API*:

– высокая производительность и масштабируемость, обеспечиваемые оптимизированным серверным фреймворком, который подходит для разработки *RESTful API*;

– полная поддержка языка *C#*, который позволяет реализовать надежную и типобезопасную серверную логику [9];

– богатая экосистема и встроенная поддержка безопасности, включая интеграцию с *JWT* для авторизации и аутентификации.

Основные причины выбора *Entity Framework*:

– предоставляет удобный *ORM*-инструмент для работы с базами данных;

– поддержка миграций, что облегчает управление схемой базы данных на протяжении жизненного цикла приложения;

– гибкость в настройке и возможность прямого выполнения *SQL*-запросов при необходимости.

Основные причины выбора *MS SQL Server*:

– надежное решение для управления реляционными базами данных, с широким набором инструментов для мониторинга и оптимизации производительности;

– отличная интеграция с *Entity Framework*, минимизирует настройку и упрощает процесс разработки;

– высокая совместимость с современными корпоративными системами и облачными решениями.

2.3.3 REST-архитектура

В данном проекте для разработки приложения используется стек *ASP.NET Core* и *React*. Архитектура сервера построена в соответствии с принципами *REST (Representational State Transfer)*, что обеспечивает удобное взаимодействие между клиентом и сервером через *HTTP*-протокол версии 1.1 [10].

Основные компоненты:

– контроллеры обрабатывают *HTTP*-запросы от клиента и управляют потоком данных между сервисами и внешним миром. В *REST*-архитектуре контроллеры обеспечивают реализацию *CRUD*-операций (*Create, Read, Update, Delete*);

– сервисы представляют собой бизнес-логику приложения и отвечают за обработку данных. В данном проекте сервисы работают с моделью данных и выполняют все основные операции;

2.4 Архитектура веб-приложения

Система, изображённая на схеме, представляет собой распределённую архитектуру с несколькими компонентами, работающими на различных операционных системах, использующих различные протоколы связи и версии ПО.

Структурная схема приложения представлена на рисунке 2.3 и также продублирована в приложении В.

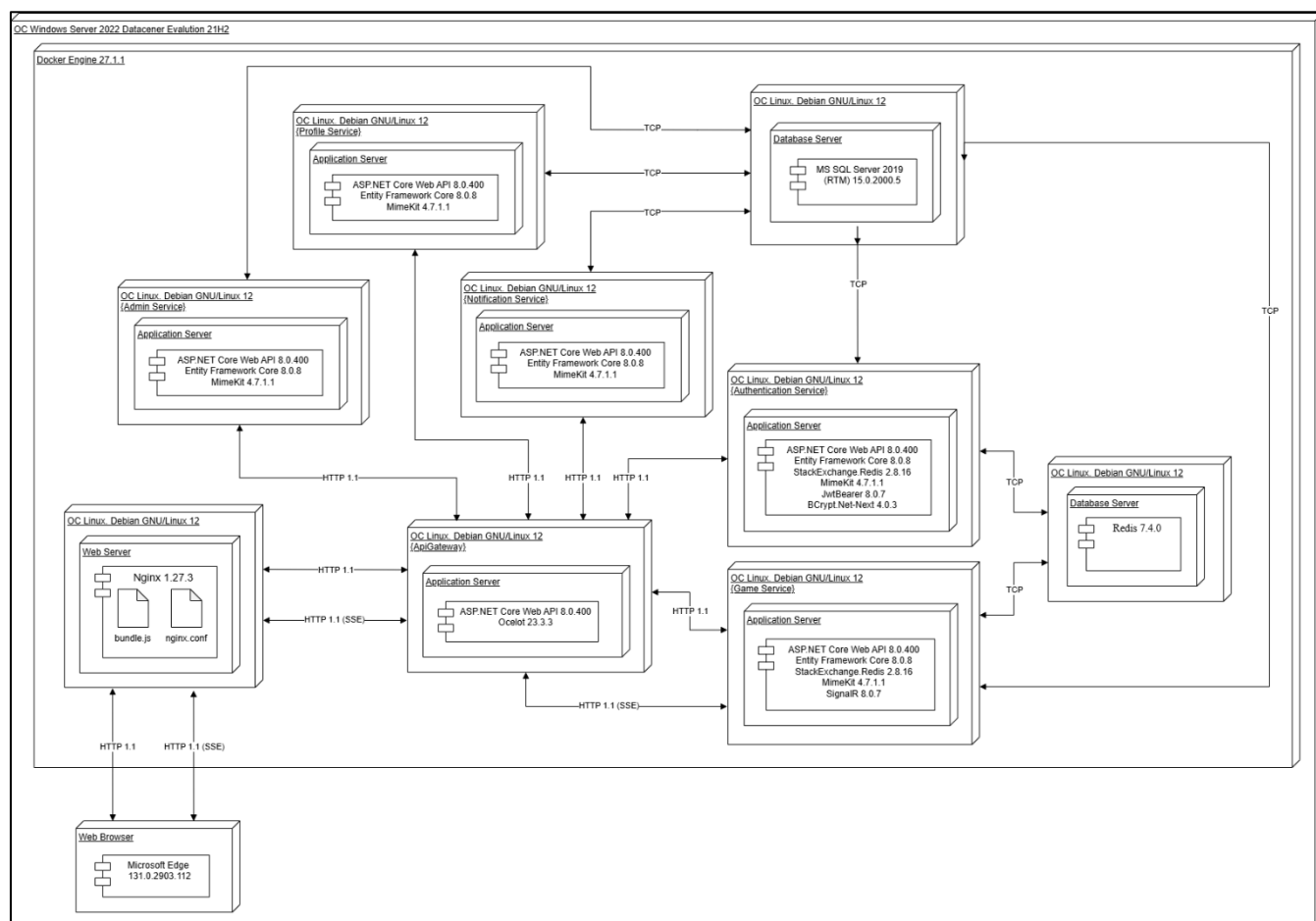


Рисунок 2.3 – Структурная схема приложения

OC Linux, Debian GNU/Linux 12 – используется для работы большинства сервисов в системе, включая серверную часть и базы данных [11].

MS SQL Server (система управления базами данных) также развернута на этой операционной системе, обеспечивая хранение и управление данными.

Клиенты работают на операционных системах, включая: *Windows 11 Pro 23H2*, где используется *Microsoft Edge 131.0.2903.112* и происходит развертывание *Docker* контейнеров на движке версии 27.1.1 [12, 13, 14].

Система представляет собой распределённую архитектуру, состоящую из серверных, клиентских компонентов, а также баз данных и систем кэширования. Компоненты развернуты на операционной системе *Debian GNU/Linux 12* и взаимодействуют через стандартные сетевые протоколы: *HTTP 1.1*, *SSE* и *TCP* [15, 16].

Серверная часть построена на *ASP.NET Core 8.0* и включает несколько сервисов:

- *Profile Service* отвечает за логику взаимодействия с пользователями.
- *Game Service* обрабатывает игровую функциональность.
- *Notification Service* передаёт уведомления пользователям.
- *Authentication Service* управляет аутентификацией и авторизацией.

Эти сервисы взаимодействуют между собой и с клиентскими компонентами по *HTTP 1.1*. Для отправки событий в реальном времени используется протокол *Server-Sent Events (SSE)*.

Хранение данных обеспечивается *MS SQL Server*, развернутым на той же операционной системе. Для передачи данных с сервисами используется *TCP*-соединение, что гарантирует надёжный и упорядоченный обмен информацией. Кэширование реализовано с помощью *Redis 7.4.0*, который ускоряет доступ к данным и уменьшает нагрузку на основную базу [17].

Клиентская часть разработана на *ReactJS 18.3.1* с использованием *Vite 5.4.0* для сборки и *Node.js 20.17.0* для обработки сборочного процесса [18]. Динамическое обновление данных и взаимодействие с сервером происходит через *HTTP 1.1* и *SSE*.

Маршрутизация запросов осуществляется с помощью *Nginx 1.27.3* [19]. Он обслуживает статические ресурсы фронтенда: файлы *index.html*, *bundle.js* и другие. Файл *bundle.js* будет находиться внутри контейнера *Nginx* в директории */usr/share/nginx/html*.

Клиенты системы – это веб-приложения, запускаемые в браузерах (например, *Microsoft Edge*). Данные передаются с использованием *HTTP 1.1* для *REST*-запросов и *SSE* для односторонней связи. Двусторонняя коммуникация реализована с помощью *SignalR 8.0.7*, где *SSE* используется для отправки событий от сервера клиенту [20].

Протоколы взаимодействия:

- *HTTP 1.1* используется для *REST API* и обмена данными между клиентами и сервисами.
- *SSE* применяется для передачи событий в реальном времени, что особенно важно для интерфейсов с живыми обновлениями.
- *TCP* обеспечивает надёжный обмен данными между сервисами, базой данных и *Redis*.

Таким образом, система использует надёжную распределённую архитектуру, включающую серверные сервисы на *ASP.NET Core 8.0*, *React Vite* для фронтенда, *MS SQL Server* для хранения данных и *Redis* для кэширования. *Nginx* и использование протоколов *HTTP 1.1*, *SSE* и *TCP* обеспечивают эффективное взаимодействие компонентов и передачу данных в реальном времени.

2.5 Выводы по разделу

Веб-приложение «Интеллектуальные игры и викторины», поддерживает четыре роли пользователей: гость, пользователь, ведущий и администратор. Гость может зарегистрироваться, просматривать информацию о доступных викторинах, играх, рейтингах и профилях пользователей. Пользователь получает доступ к

основным функциям, таким как создание игровых комнат, участие в играх, прохождение викторин, просмотр результатов и размещение комментариев. Ведущий, кроме функций пользователя, может создавать вопросы, управлять игровым процессом и редактировать комментарии. Администратор имеет возможность управлять викторинами, пользователями и играми.

Приложение реализует ключевые функции: регистрация, авторизация и управление учетными записями, создание и управление викторинами, игровыми комнатами, а также взаимодействие с пользователями через комментарии, просмотр рейтингов, результатов игр и викторин для повышения интерактивности и вовлечения пользователей. Полный перечень функций представлен в таблице 2.2.

Логическая структура базы данных включает таблицы для хранения информации о пользователях, викторинах, играх, результатах и комментариях.

Серверная часть разработана с использованием *ASP.NET Core Web API*, *MS SQL Server* и *Entity Framework*. Клиентская часть использует *React* для создания удобного пользовательского интерфейса.

Заключение

В ходе прохождения практики в период с 10 февраля 2025 по 21 марта 2025 в компании ООО «Модсен» было разработано веб-приложение «Интеллектуальные игры и викторины».

1. Веб-приложение поддерживает 4 роли: гость, пользователь, ведущий, администратор.

2. В ходе рассмотрения аналогов разрабатываемого приложения были выявлены основные функции и элементы.

3. База данных включает 14 таблиц.

4. Использована *REST*-архитектура, обеспечивающая удобное взаимодействие клиента и сервера через *HTTP*. Основные компоненты: Модели данных: описывают структуру и связи таблиц базы данных. Контроллеры: реализуют *CRUD*-операции. Сервисы: выполняют бизнес-логику приложения. Поддержка связи через *SignalR* для обновлений в реальном времени.

В результате разработки были выполнены все поставленные задачи, в том числе:

- приложение должно поддерживать четыре роли: Гость, Пользователь, Ведущий, Администратор;

- функции пользователя с ролью «Гость»: регистрация, просмотр информации о доступных викторинах, просмотр информации о существующих играх, просмотр рейтингов других игроков по прохождению викторин, просмотр профилей пользователей;

- функции пользователя с ролью «Пользователь»: авторизация, просмотр информации о доступных викторинах, просмотр информации о существующих играх, просмотр профилей пользователей, просмотр рейтингов других игроков по прохождению викторин, создание игровых комнат, редактирование личной информации, присоединение к играм, прохождение викторин, просмотр результатов прохождения викторин, принятие участия в играх, просмотр результатов игр и рейтинга игроков в игровой комнате, размещение комментариев к завершенной игре, размещение комментариев к пройденной викторине;

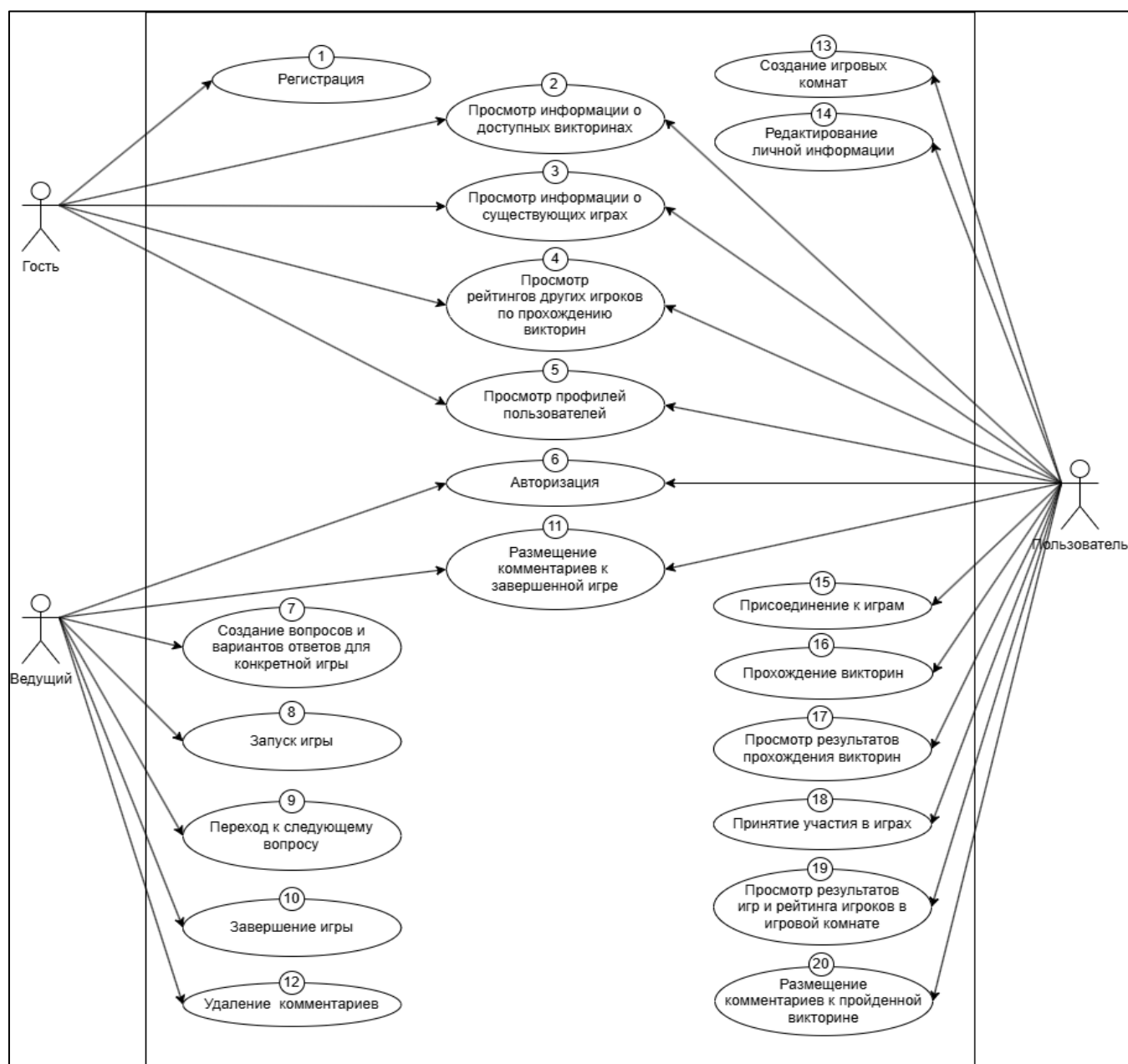
- функции пользователя с ролью «Ведущий»: авторизация, создание вопросов и вариантов ответов для конкретной игры, запуск игры, переход к следующему вопросу, завершение игры, размещение комментариев к завершенной игре, удаление комментариев;

- функции пользователя с ролью «Администратор»: авторизация, создание, изменение, удаление пользователей, игр и викторин, просмотр информации о доступных викторинах, просмотр информации о существующих играх, просмотр профилей пользователей, просмотр рейтингов других игроков по прохождению викторин, создание игровых комнат, редактирование личной информации, присоединение к играм, прохождение викторин, просмотр результатов прохождения викторин, принятие участия в играх, просмотр результатов игр и рейтинга игроков в игровой комнате, размещение комментариев к завершенной игре, размещение комментариев к пройденной викторине.

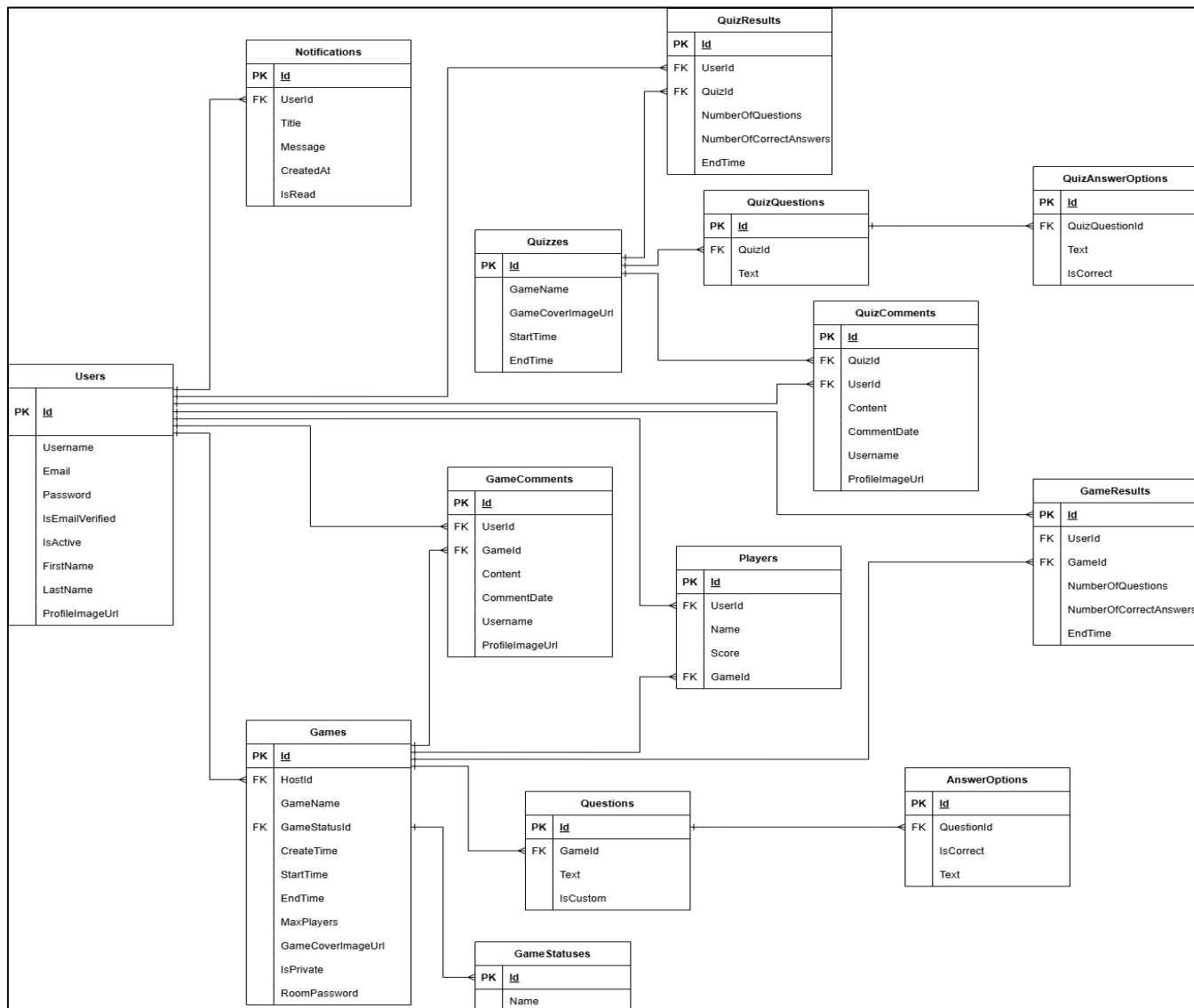
Список используемых источников

1. ASP.NET Core Web API [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-9.0> – Дата доступа: 14.02.2025
2. React [Электронный ресурс] / Режим доступа: <https://react.dev> – Дата доступа: 14.02.2025
3. Vite [Электронный ресурс] / Режим доступа: <https://vite.dev> – Дата доступа: 14.02.2025
4. Quiz [Электронный ресурс] / Режим доступа: <https://quiz.com> – Дата доступа: 18.02.2025
5. Kahoot! [Электронный ресурс] / Режим доступа: <https://kahoot.com> – Дата доступа: 18.02.2025
6. Jackbox Games [Электронный ресурс] / Режим доступа: <https://www.jackboxgames.com/> – Дата доступа: 18.02.2025
7. Entity Framework [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/en-us/ef/> – Дата доступа: 21.02.2025
8. MS SQL Server [Электронный ресурс] / Режим доступа: <https://www.microsoft.com/en-us/sql-server> – Дата доступа: 21.02.2025
9. C# [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/en-us/dotnet/csharp/> – Дата доступа: 22.02.2025
10. HTTP 1.1 [Электронный ресурс] / Режим доступа: <https://www.w3.org/Protocols/> – Дата доступа: 22.02.2025
11. Debian [Электронный ресурс]. – Режим доступа: <https://www.debian.org/releases/bookworm/> – Дата доступа: 25.02.2025
12. Windows 11 [Электронный ресурс] / Режим доступа: <https://www.microsoft.com/ru-ru/software-download/windows11> – Дата доступа: 25.02.2025
13. Microsoft Edge [Электронный ресурс] / Режим доступа: www.microsoft.com/en-us/edge – Дата доступа: 25.02.2025
14. Docker [Электронный ресурс]. – Режим доступа: <https://docs.docker.com> – Дата доступа: 25.02.2025
15. Server-Sent Events (SSE) [Электронный ресурс] / Режим доступа: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events – Дата доступа: 25.02.2025
16. TCP [Электронный ресурс] / Режим доступа: <https://www.rfc-editor.org/rfc/rfc793> – Дата доступа: 25.02.2025
17. Redis [Электронный ресурс] / Режим доступа: <https://redis.io> – Дата доступа: 26.02.2025
18. Node.js [Электронный ресурс] / Режим доступа: <https://nodejs.org/en> – Дата доступа: 03.03.2025
19. Nginx [Электронный ресурс] / Режим доступа: <https://nginx.org> – Дата доступа: 03.03.2025
20. SignalR [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-9.0> – Дата доступа: 03.03.2025

ПРИЛОЖЕНИЕ А Диаграмма вариантов использования



ПРИЛОЖЕНИЕ Б Логическая схема базы данных



ПРИЛОЖЕНИЕ В Структурная схема приложения

