



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

ФАКУЛЬТЕТ «Информатика и системы управления (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

K KURSOVOY RABOTE

НА ТЕМУ:

*«Создание информационной системы серверов для
видеоигры»*

Студент ИУ7-63Б
(Группа)

(Подпись, дата)

И.А. Цветков
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

О.В. Кузнецова
(И. О. Фамилия)

2022 г.

РЕФЕРАТ

В данной работе представлена реализация Web-приложения с информационной системой для поиска серверов кроссплатформенной игры. Готовое приложение позволяет просматривать список серверов, сортировать и фильтровать его по доступным параметрам, а также добавлять сервера в список избранных.

PostgreSQL была выбрана в качестве системы управления базой данных. Взаимодействие с базой обеспечивается посредством фреймворка EntityFramework Core для языка C#.

Ключевые слова: игра, сервер, C#, PostgreSQL, EntityFramework Core, ASP.NET.

Рассчетно-пояснительная записка к курсовой работе содержит 53 страницы, 36 иллюстраций, 4 таблицы, 21 источник, 2 приложения.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Аналитическая часть	7
1.1 Анализ существующих решений	7
1.1.1 Российский рынок	7
1.1.2 Зарубежный рынок	9
1.1.3 Вывод	10
1.2 Формализация задачи	11
1.3 Описание типов пользователей	12
1.4 Формализация данных	15
1.5 Выбор модели базы данных	16
1.5.1 Дореляционные модели	17
1.5.2 Реляционные модели	17
1.5.3 Постреляционные модели	17
1.5.4 Вывод	18
2 Конструкторская часть	19
2.1 Описание таблиц базы данных	19
2.2 Функции базы данных	21
2.3 Триггеры базы данных	24
2.4 Роли базы данных	26
3 Технологическая часть	27
3.1 Средства реализации	27
3.2 Архитектура приложения	27
3.3 Структура классов	27
3.4 Выбор СУБД	30
3.4.1 Виды СУБД	30
3.4.2 Вывод	30
3.5 Реализация функций	31
3.6 Реализация триггеров	33
3.7 Реализация ролевой модели	34
3.8 Полученный результат	36

4 Исследовательская часть	46
4.1 Цель эксперимента	46
4.2 Результаты эксперимента	46
4.2.1 Поиск по первичному ключу	46
4.2.2 Фильтрация по столбцу таблицы	47
ЗАКЛЮЧЕНИЕ	49
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51
ПРИЛОЖЕНИЕ А ФУНКЦИЯ СОРТИРОВКИ СПИСКА СЕРВЕРОВ	52

ВВЕДЕНИЕ

Мир компьютерных игр развивается с каждым годом. Всего за каких-то 30 лет простые 2D разработки превратились в полномасштабные проекты, которые поражают своей графической составляющей и невероятным вниманием к деталям. Важной частью этого бурного роста стало появление онлайна в компьютерных играх. Теперь появилась возможность сыграть в любимую игру вместе со своими друзьями со всего мира.

В связи с этим стали появляться специальные сайты, которые собирают информацию обо всех серверах, которые доступны для данной игры и представляют пользователю возможность подобрать идеально подходящий для себя сервер, исходя из доступной информации. При этом практически каждый сайт обладает своими проблемами, которые будут описаны позже.

Таким образом, целью данного курсового проекта является создание информационной системы для серверов игры. Для достижения данной цели необходимо решить следующие задачи:

- рассмотреть существующие решения;
- выбрать модель хранения данных;
- разработать базу данных;
- выделить роли пользователей;
- выбрать необходимый набор технологий для разработки;
- создать программный продукт, который решает поставленную цель.

1 Аналитическая часть

1.1 Анализ существующих решений

Существует большое количество сайтов, которые предоставляют возможность найти сервера для той или иной онлайн игры. При этом каждый сайт обладает своими преимуществами и недостатками. Далее будут рассмотрены примеры ПО для поиска серверов на российском и зарубежном рынках. Выбранные сайты являются популярными среди пользователей, что говорит о заинтересованности многих людей в поиске серверов для игр.

1.1.1 Российский рынок

Одним из сайтов для поиска серверов для игры «Minecraft» [1] является «MinecaftRating» [2]. При этом интерфейс главной страницы представлен на рисунке 1.1. На нем предоставляется большое количество функциональностей, таких как:

- поиск серверов (с сортировкой по самым популярным, по версии, по количеству игроков);
- получение полной информации о каждом сервере;
- добавление нового сервера;
- для зарегистрированных пользователей — добавление в избранное.

При этом не имеется возможности посмотреть сервера для разных платформ.

Для компьютерной игры «Counter-Strike» [3] также существуют сайты с серверами. Примером такого сайта является «Сервера КС» [4]. Интерфейс главной страницы представлен на рисунке 1.2. Для данной игры рынок развит слабее, поэтому функциональностей куда меньше. При этом предоставляются следующие возможности:

- просмотр серверов (присутствует лишь список, поиск и сортировка невозможны);

- для зарегистрированных пользователей — добавление нового сервера, добавление в избранное.

Из-за отсутствия сортировок серверов, имеется возможность купить место в верху таблицы, чтобы пользователи замечали сначала «проплаченные» сервера.

The screenshot shows the homepage of MinecraftRating.ru. At the top, there are tabs for 'ТОП серверов' (Top Servers), 'Новые сервера' (New Servers), 'Добавить сервер' (Add Server), 'Моды' (Mods), 'Помощь' (Help), 'Создать сервер' (Create Server), and 'Хостинг' (Hosting). Below these are buttons for 'Войти' (Login), 'Регистрация' (Registration), 'Раскрутить сервер' (Promote Server), and 'Лаунчер Minecraft' (Minecraft Launcher).

The main content area has several sections:

- Сервера Майнкрафт**: A table showing server statistics for various versions (1.18.2, 1.18.1, etc.) across various metrics like players online, latency, and ping.
- Основное**: Buttons for 'Выживание', 'Креатив', 'Дип', 'Без доната', 'Без привата', 'RPG', 'Клыны', 'Херобри', 'Торьмы', 'Fly', 'Седьбы', 'РВЕ', 'Экономика', 'Бесплатная админка', 'Без регистрации', 'Инвенти', 'Roleplay', 'Гиф', 'Анария', 'Ванильные', 'Читы', 'Наруто', 'Большой онлайн', 'Пустые', 'Без античита', 'Квесты', 'Classic', 'Хедор', 'Дуэли', 'Зарубежные', and 'Бесплатный донат'.
- С мини-играми**: Buttons for 'Голодные игры', 'Bed Wars', 'Против', 'TNT Ран', 'Сайл Варс', 'Build battle', 'Парору', 'Сайлбоя', 'Моб арена', and 'Quake'.
- С модами**: Buttons for 'Оружие', 'Industrial Craft', 'DayZ', 'Pixelmon', 'Машинки', 'Сталкер', 'Таумрафт', 'Зомби апокалипсис', 'Динайн РПГ', 'Lucky Block', 'Flet's', 'HiTech', 'BuildCraft', 'Galacticraft', 'MineZ', 'Морп', 'Смерч', 'Муравьев', 'Властелин колец', 'Star Wars', 'Сумеречный лес', 'GreP Tech', 'Forestry', 'Mo' Creatures', 'RedPower', 'FTA', 'Dragon's evolution', 'Millenaire', 'RailCraft'.
- С плагинами**: Buttons for 'Нукихербет', 'Урагансорт', 'Floodprophet', 'Кир спарт', 'Авто-шахта', 'Магазин', 'fp help', 'Питомцы', 'Antispam', and 'Работы'.

At the bottom, there's a section for 'Мониторинг серверов Minecraft' with a 'Поиск по серверам' (Search servers) field and a 'найти' (Find) button.

Рисунок 1.1 – Сайт для поиска серверов для игры «Minecraft» на русском рынке

The screenshot shows the homepage of a CS:GO server search website. At the top, there are tabs for 'Сервера Counter Strike', 'СЕРВЕРА КС ГО', 'Сервера CS GO', 'CS SOURCE', 'CSS v92', 'CSS v92 (No Steam)', 'Добавить сервер', 'Платные услуги', 'VIP / TOP', 'FAQ', 'NEW', 'Контакты', and 'Договор оферты'.

The main content area displays a grid of server listings:

Игроков: 36 / 64	Игроков: 9 / 64	Игроков: 31 / 48	Игроков: 16 / 40	Игроков: 18 / 64	Игроков: 10 / 64
БОЙЦОВЫЙ КЛУБ 46.174.53.216.27015	TOP Alpha Division (FREE) 46.174.55.77.27015	[OCTAGON] 62.122.214.32.27015	Русский Медведь 18+ 46.174.48.217.27015	МЯСОРУБКА © 46.174.49.219.27015	IDLE D+S #NOKILL 212.22.93.98.27030

Below the grid are social media sharing buttons (VK, OK, Facebook, G+, Twitter) and a '241' link. There are also buttons for 'Скачать лаунчер' (Download Launcher), '1060 онлайн' (1060 online), '1.12.2 версия' (1.12.2 version), '72 голосов' (72 votes), and '770 баллов' (770 points).

On the right side, there are sections for 'СКАЧАТЬ CS:SOURCE НАПРЯМУЮ' (Download CS:SOURCE Directly) and 'СКАЧАТЬ CS:SOURCE ТОРРЕНТОМ' (Download CS:SOURCE Torrent). There's also a search bar for 'IP/Port или название' (IP/Port or name) and a '60' button.

Further down, there are sections for 'ВОЙТИ' (Login), 'РЕГИСТРАЦИЯ' (Registration), 'Забыли пароль?' (Forgot password?), and 'Мы ВКонтакте' (We are on VKontakte). It also features a 'Топ 12 лучших карт' (Top 12 best maps) section with details about 'de_mirage' and 'de_dust2'.

Рисунок 1.2 – Сайт для поиска серверов для игры «Counter-Strike» на русском рынке

1.1.2 Зарубежный рынок

Примером сайта для поиска серверов для игры «Minecraft» [1] является «Minecraft Servers» [5]. На рисунке 1.3 представлен интерфейс главной страницы. Зарубежный аналог обладает тем же самым функционалом, что и пример сайта с российского рынка. При этом поиск серверов для различных платформ также отсутствует.

Для игры «Counter-Strike» [3] зарубежный рынок развит сильнее. Так, примером является сайт «Game Tracker» [6]. Интерфейс главной страницы представлен на рисунке 1.4. Он обладает тем же самым функционалом, что и его российский аналог, а также:

- присутствует сортировка по различным параметрам;
- существует возможность выбора серверов для различных стран мира.

При этом возможность покупки приоритетного места в списке отсутствует.

The screenshot shows the homepage of the Minecraft Servers website. At the top, there are navigation links for 'Home', 'My Servers', and 'Sponsored'. A search bar with a magnifying glass icon is located at the top right. Below the header, the title 'Minecraft Servers' is displayed in green. A breadcrumb trail indicates the user is at 'Minecraft Servers'. On the right side of the header, there is a 'Login' button. The main content area is titled 'Minecraft Server List' and features a grid of category buttons: Minecraft Anarchy Servers, Creative Servers, Economy Servers, Factions Servers, Hardcore Servers, KitPvP Servers; Minecraft MCMO Servers, Mini Games Servers, Parkour Servers, Pixelmon Servers, Prison Servers, PvE Servers; Minecraft PvP Servers, Roleplay Servers, Skyblock Servers, Skywars Servers, Survival Servers, Towny Servers; Minecraft Vanilla Servers, modded Servers, Bedwars Servers, SMP Servers, HungerGames Servers, New Servers. Below this is a section titled 'View Minecraft Server List' with a table header: Rank, Name, Server, Players, IP. Two servers are listed: 'Complex Gaming' (Rank 1) and 'Vortex Network' (Rank 2). Each entry includes a thumbnail image, the server name, its IP address, player count (16789/16790 for Complex Gaming, 2023/3000 for Vortex Network), and a link to the server's website.

Рисунок 1.3 – Сайт для поиска серверов для игры «Minecraft» на зарубежном рынке

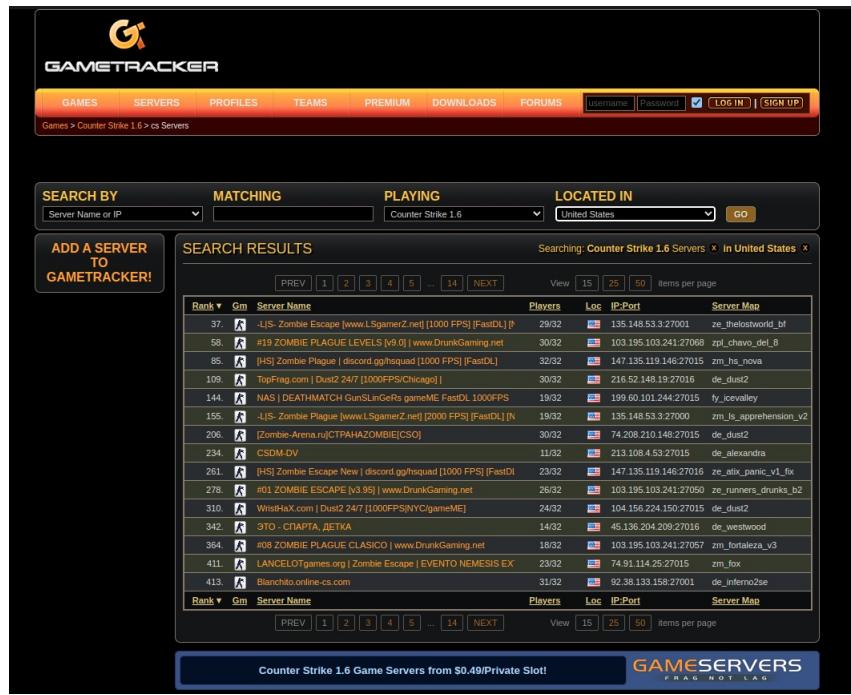


Рисунок 1.4 – Сайт для поиска серверов для игры «Counter-Strike» на зарубежном рынке

1.1.3 Вывод

В таблице 1.1 представлены результаты сравнения существующих решений. Выделенные критерии сравнения:

- К1 – просмотр списка серверов;
- К2 – сортировка списка серверов;
- К3 – просмотр серверов для разных платформ;
- К4 – добавление в избранное;
- К5 – получение подробной информации о сервере.

Таблица 1.1 – Сравнение существующих решений

Название	K1	K2	K3	K4	K5
«MinecaftRating»	+	+	-	+	+
«Сервера KC»	+	-	-	-	-
«Minecraft Servers»	+	+	-	+	-
«Game Tracker»	+	+	-	+	+

Таким образом, российский и зарубежный рынки предоставляют множество различных сайтов для нахождения серверов для игр. Стоит отметить то, что при этом сайты обладают рядом серьезных проблем.

1. Отсутствует возможность просмотреть списки серверов для разных платформ одной и той же игры.
2. Недостаточная функциональность для поиска необходимых серверов.
3. Слабая развитость рынка для некоторых игр.

1.2 Формализация задачи

Должно быть разработано Web-приложение для поиска серверов для игры. При этом ПО должно содержать в себе следующие возможности.

- **Список серверов.** Основным процессом является просмотр списка серверов. Список серверов представляет из себя таблицу из всех имеющихся серверов в базе данных. Информационные поля сервера указаны в колонках таблицы, а каждый отдельный сервер — строка этой таблицы. Также таблица должна иметь возможность сортировки по полям сервера (названию, IP-адресу, версии игры, рейтингу сервера) и имени платформы, на которой запущен сервер.
- **Авторизация и регистрация на сайте.** Должна быть введена возможность регистрации пользователя на сайте, чтобы открыть ему дополнительные возможности – просмотр информации о хостинге сервера, списка игроков сервера и добавления сервера в список избранного.
- **Список избранных серверов.** Каждый зарегистрированный пользователь должен обладать возможность добавить интересующий его сервер в список избранных. Это необходимо для того, чтобы пользователь не потерял интересующий его сервер и всегда имел быстрый доступ к информации о нем.
- **Рейтинг серверов.** Рейтинг сервера формируется из количества добавлений данного сервера в список избранных серверов отдельно взятого пользователя. Данный рейтинг выводится в качестве поля в таблице серверов,

предоставляя возможность пользователям узнать наиболее популярный сервер. При этом, если пользователь сайта удалил данный сервер из своего списка избранных серверов, то рейтинг сервера понизится.

- **Фильтрация серверов по платформам.** Каждый сервер может находиться лишь на одной единственной платформе. Поэтому важно разделить список серверов на отдельные списки для каждой платформы, чтобы пользователь мог выбрать именно те сервера, которые подходят для его рабочего устройства.
- **Добавление/изменение/удаление серверов администратором.** Данной возможностью наделен лишь администратор сайта. Должен быть предоставлен интерфейс для данного процесса. При добавлении/изменении должны быть добавлены ограничения на ввод информации, чтобы предотвратить ошибки ввода, а также не допустить появления серверов с таким же названием или на том же IP-адресе. При удалении должно быть реализовано подтверждение удаления сервера, чтобы предотвратить случайные нажатия.
- **Изменение ролей пользователей администратором.** Должна быть введена возможность изменения роли зарегистрированного пользователя. При этом администратор не может изменить собственную роль, а также администратору с никнеймом «admin» должно быть запрещено изменять роль, чтобы на сайте был всегда, как минимум, один администратор.

1.3 Описание типов пользователей

В задаче выделено 3 типа пользователей (таблица 1.2).

Таблица 1.2 – Типы пользователей

Тип	Функциональность
Гость (неавторизованный пользователь)	<ul style="list-style-type: none"> • Просмотр списка серверов • Выбор платформы для списка серверов • Сортировка списка серверов • Регистрация • Авторизация
Авторизованный пользователь	<ul style="list-style-type: none"> • Добавление сервера в список избранных • Выбор платформы для списка серверов • Просмотр списка серверов • Сортировка списка серверов • Просмотр информации о хостинге сервера, а также списка игроков сервера
Администратор	<ul style="list-style-type: none"> • Добавление нового сервера на сайт • Удаление сервера с сайта • Изменение информации о сервере • Изменение ролей зарегистрированных пользователей • Добавление сервера в список избранных • Просмотр списка серверов • Выбор платформы для списка серверов • Сортировка списка серверов • Просмотр информации о хостинге сервера, а также списка игроков сервера

На рисунках 1.5-1.7 представлены Use-Case диаграммы выделенных типов пользователей.

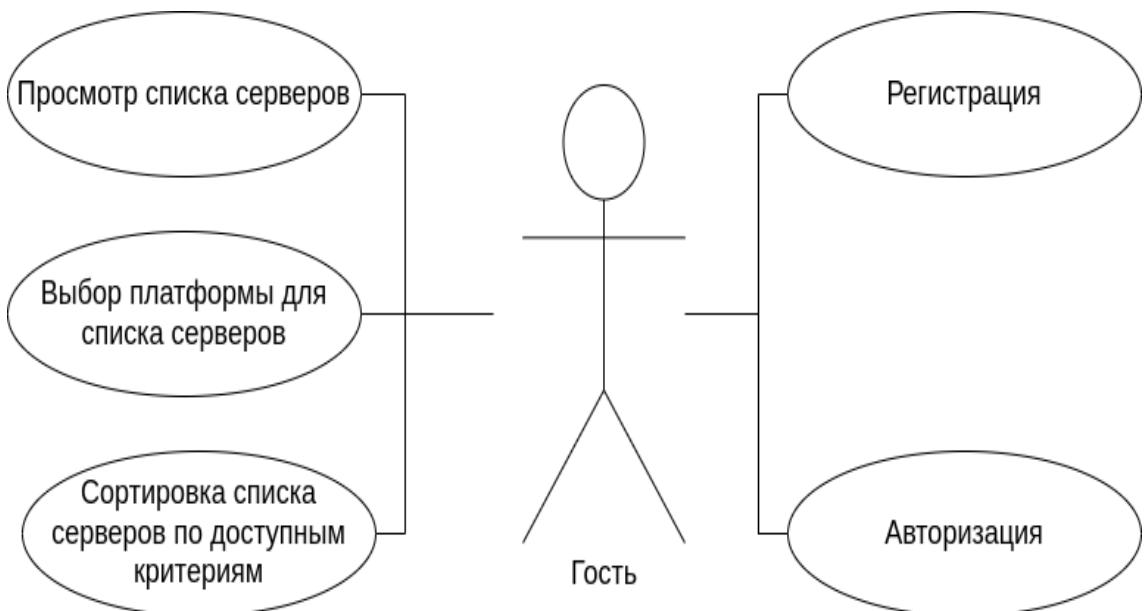


Рисунок 1.5 – Use-Case диаграмма для неавторизованного пользователя

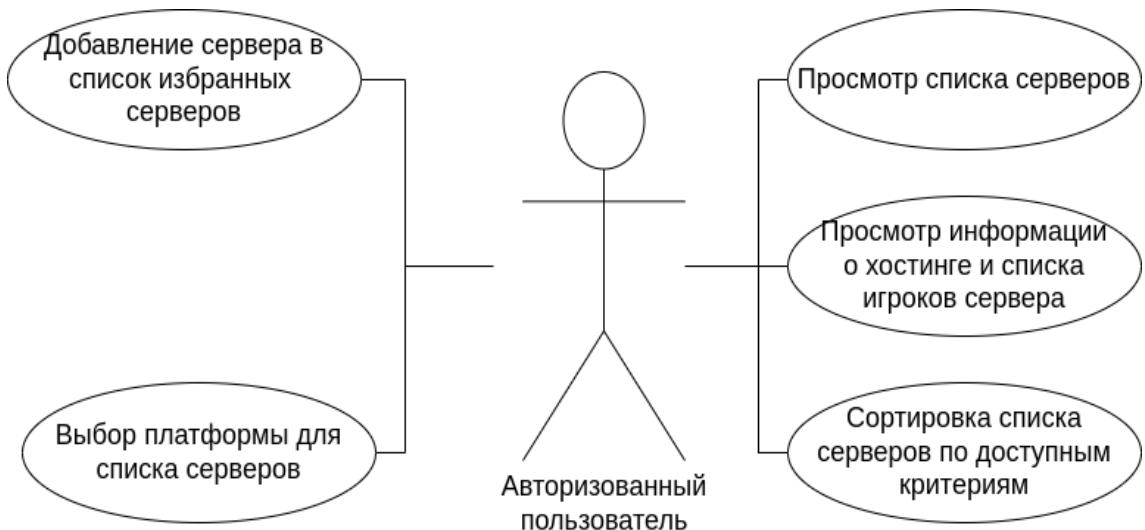


Рисунок 1.6 – Use-Case диаграмма для авторизованного пользователя

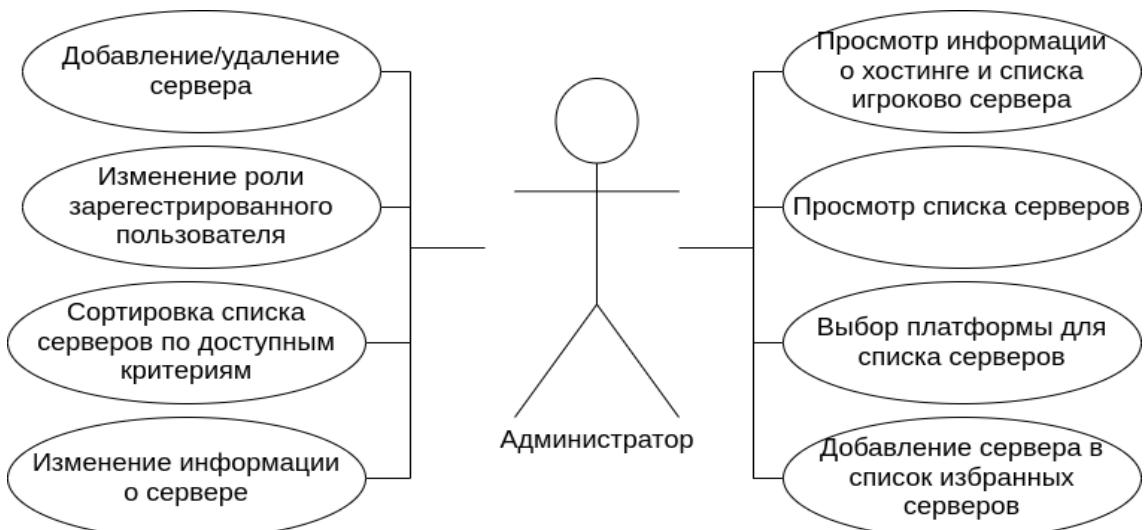


Рисунок 1.7 – Use-Case диаграмма для администратора

1.4 Формализация данных

В соответствии с задачей и типами пользователей, база данных должна содержать модели, описанные в таблице 1.3.

Таблица 1.3 – Модели базы данных

Тип	Функциональность
Сервер	<ul style="list-style-type: none">• ID• Название• IP-адрес• Версия игры• Рейтинг• ID платформы• ID хостинга
Платформа	<ul style="list-style-type: none">• ID• Название• Популярность• Стоимость
Хостинг	<ul style="list-style-type: none">• ID• Название• Плата в месяц
Игрок	<ul style="list-style-type: none">• ID• Никнейм• Сыграно часов на сервере• Дата последнего захода на сервер
Пользователь	<ul style="list-style-type: none">• ID• Логин• Пароль• Роль

Также на рисунке 1.8 представлена ER-диаграмма [7] разрабатываемой системы в нотации Чена.

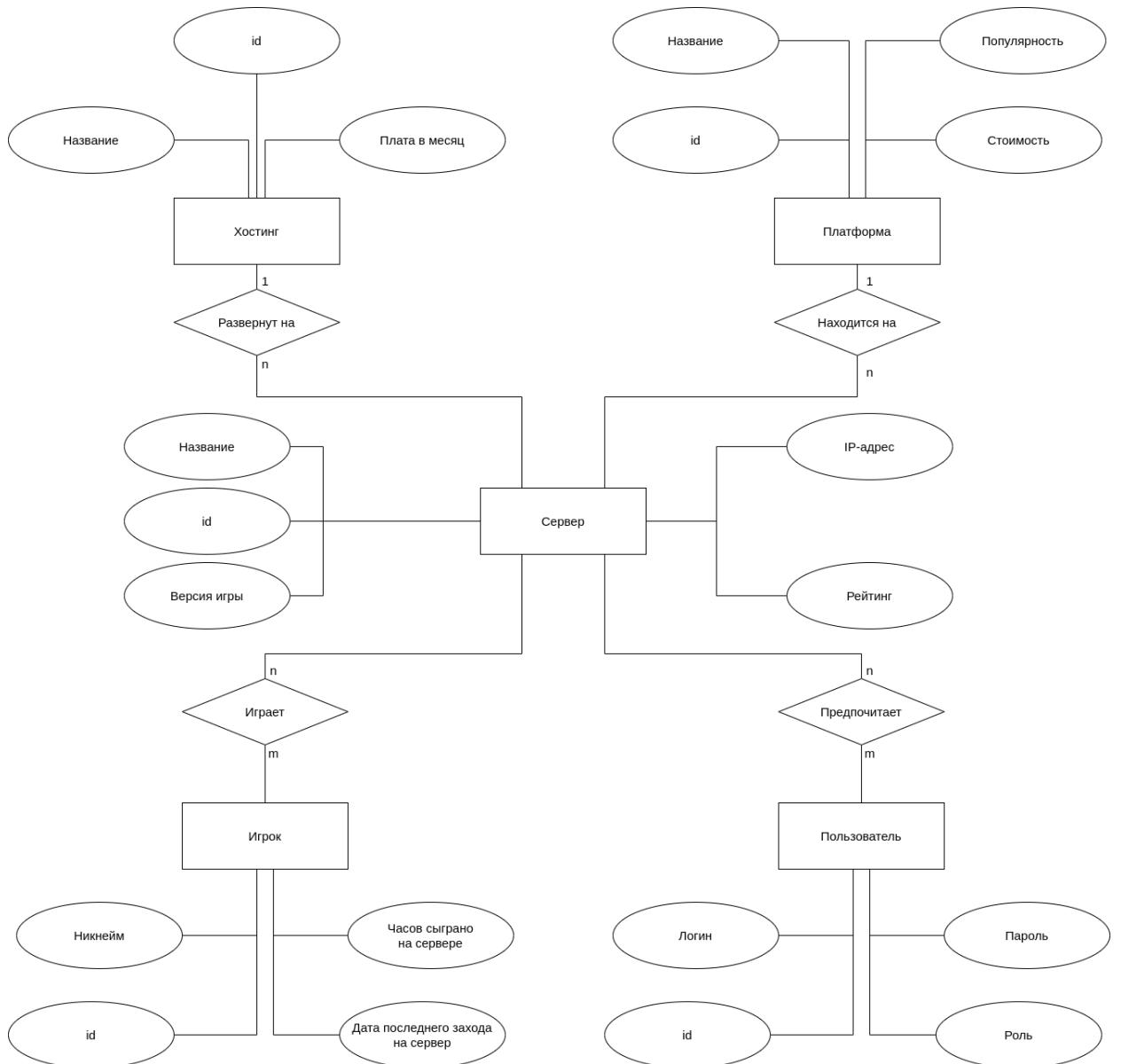


Рисунок 1.8 – ER-диаграмма в нотации Чена

1.5 Выбор модели базы данных

База данных [8] – упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляет системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу [8]. Этими данными можно управлять, изменять, обновлять, контролировать и упорядочивать.

Базы данных делятся на три модели организации данных:

- дореляционные;
- реляционные;
- постреляционные.

1.5.1 Дореляционные модели

Дореляционные модели базы данных [9] предоставляли доступ на уровне записей, которые располагались в виде древовидной структуры со связями предок-потомок. При этом взаимодействие с базой данных происходило с использованием языков программирования, которые были расширены функциями дореляционных СУБД.

Главными недостатками является то, что оптимизация доступа к данным со стороны системы отсутствует, а также древовидная структура является весьма трудоемкой.

1.5.2 Реляционные модели

Реляционная база данных [10] — совокупность отношений, которые содержат всю информацию, которая должна храниться в базе данных. Каждое отношение — двумерная таблица, в каждой строке которой хранится запись об объекте, а в каждом столбце — свойства данного объекта.

Реляционные базы данных обладают несомненным преимуществом — благодаря стандартизированного языка запросов SQL [11], существует возможность подмены СУБД.

1.5.3 Постреляционные модели

В постреляционных моделях баз данных [12] не используется табличная схема строк и столбцов. В этих базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных.

При этом они делятся на следующие основные категории:

- коллекции — документы, упорядоченные по группам;
- ключ-значение — хэш-таблица, в которой по ключу находится значение;

- колоночная — хранит информацию в виде разреженной матрицы, строки и столбцы которой используются как ключи;
- графовые — сетевая база, использующие узлы и ребра для хранения данных.

Данные модели используются для специфических задач, где явно подходит одна из приведенных выше категорий, что явно ускоряет работу программного продукта благодаря грамотной работе с данными.

1.5.4 Вывод

Для поставленной задачи разработки информационной системы Web-приложения поиска серверов для кроссплатформенных онлайн игр наилучшим образом подходит реляционная модель хранения данных. Выбор обусловлен необходимостью хранить данные в виде структурированных таблиц. Также реляционная модель с помощью стандартизированного языка запросов SQL позволяет подменять системы управления базами данных. Это является важным преимуществом, так как в будущем разделе будет произведен выбор СУБД.

Вывод

В данном разделе была проанализирована выполняемая задача — была проведена ее формализация, проведена формализация данных, описаны типы пользователей. Также были рассмотрены модели базы данных и выбрана реляционная модель.

2 Конструкторская часть

2.1 Описание таблиц базы данных

Схема разрабатываемой базы данных приведена на рисунке 2.1.

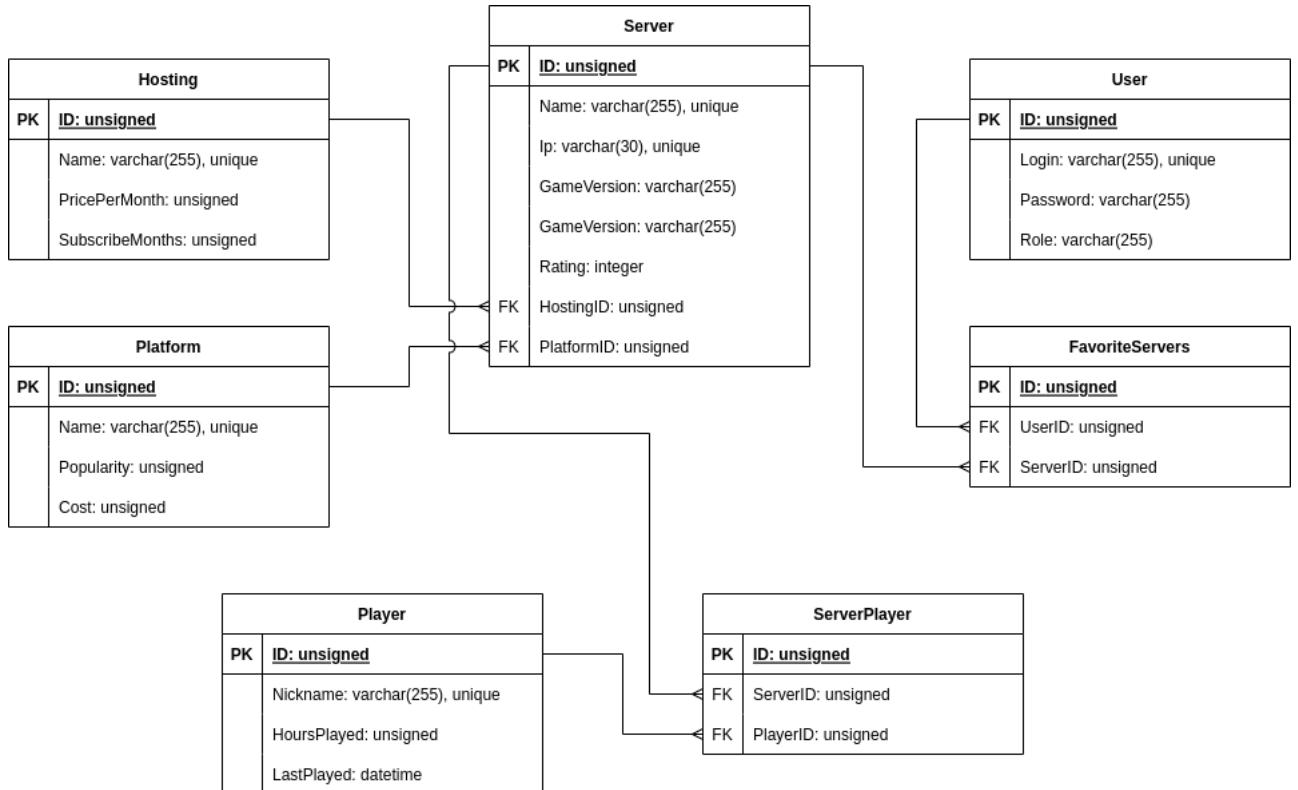


Рисунок 2.1 – Диаграмма базы данных

Реализуемая модель базы данных содержит 7 таблиц.

1. Таблица «Server» хранит информацию о серверах. Содержит следующие поля:
 - ID – первичный ключ; тип данных – unsigned;
 - Name – уникальное название сервера; тип данных – varchar(255);
 - IP – уникальный IP-адрес сервера; тип данных – varchar(30);
 - GameVersion – версия игры, для которой открыт данный сервер; тип данных – varchar(255);
 - Rating – рейтинг сервера; тип данных – integer;
 - HostingID – идентификатор хостинга; тип данных – unsigned;
 - PlatformID – идентификатор платформы; тип данных – unsigned;

2. Таблица «User» хранит информацию о пользователе сайта. Содержит следующие поля:
 - ID – первичный ключ; тип данных – unsigned;
 - Login – уникальный логин на сайте; тип данных – varchar(255);
 - Password – пароль на сайте; тип данных – varchar(255);
 - Role – роль пользователя; тип данных – varchar(255);
3. Таблица «FavoriteServer» хранит информацию об избранных серверах игроков. Является связующей таблицей между таблицами «Server» и «User». Содержит следующие поля:
 - ID – первичный ключ; тип данных – unsigned;
 - UserID – идентификатор пользователя; тип данных – unsigned;
 - ServerID – идентификатор сервера; тип данных – unsigned;
4. Таблица «Player» хранит информацию об игроке сервера. Содержит следующие поля:
 - ID – первичный ключ; тип данных – unsigned;
 - Nickname – уникальный никнейм игрока; тип данных – varchar(255);
 - HoursPlayed – количество часов, сыгранных на серверах; тип данных – unsigned;
 - LastPlayed – дата крайнего раза захода на сервер; тип данных – unsigned;
5. Таблица «FavoriteServer» хранит информацию об играх и на каких серверах они играют. Является связующей таблицей между таблицами «Server» и «Player». Содержит следующие поля:
 - ID – первичный ключ; тип данных – unsigned;
 - PlayerID – идентификатор игрока; тип данных – unsigned;
 - ServerID – идентификатор сервера; тип данных – unsigned;
6. Таблица «Hosting» хранит информацию о хостинге, на котором находится сервер. Содержит следующие поля:

- ID – первичный ключ; тип данных – unsigned;
 - Name – уникальное название хостинга; тип данных – varchar(255);
 - PricePerMonth – плата за размещение сервера на хостинге в месяц; тип данных – unsigned;
 - SubMonths – количество месяцев подписки сервера на хостинг; тип данных – unsigned;
7. Таблица «Platform» хранит информацию о платформах, на которых может быть запущена игра. Содержит следующие поля:
- ID – первичный ключ; тип данных – unsigned;
 - Name – уникальное название платформы; тип данных – varchar(255);
 - Popularity – популярность платформы; тип данных – unsigned;
 - Cost – стоимость платформы; тип данных – unsigned;

2.2 Функции базы данных

В проектируемой базе данных определены следующие функции.

1. **Функция выбора серверов.** Данная функция выбирает или все сервера из таблицы «Server», или только избранные сервера пользователя по его идентификатору. Схема алгоритма функции представлена на рисунке 2.2.
2. **Функция фильтрации серверов.** Данная функция фильтрует сервера из таблицы «Server», полученные после работы функции выбора серверов. Фильтрация происходит по названию сервера и/или идентификатору платформы. Схема алгоритма функции представлена на рисунке 2.5.
3. **Функция сортировки серверов.** Данная функция сортирует сервера из таблицы «Server», полученные после работы функции фильтрации серверов. Сортировка происходит по полям: название сервера, IP-адрес сервера, версия игры сервера, рейтинг сервера, названию платформы (все – по возрастанию и по убыванию). Схема алгоритма функции представлена на рисунке 2.4.

4. Функция парсинга серверов. Данная функция является некой «прослойкой», которая возвращает все сервера из таблицы «Server», полученных после работы функции сортировки серверов. Схема алгоритма функции представлена на рисунке 2.3.

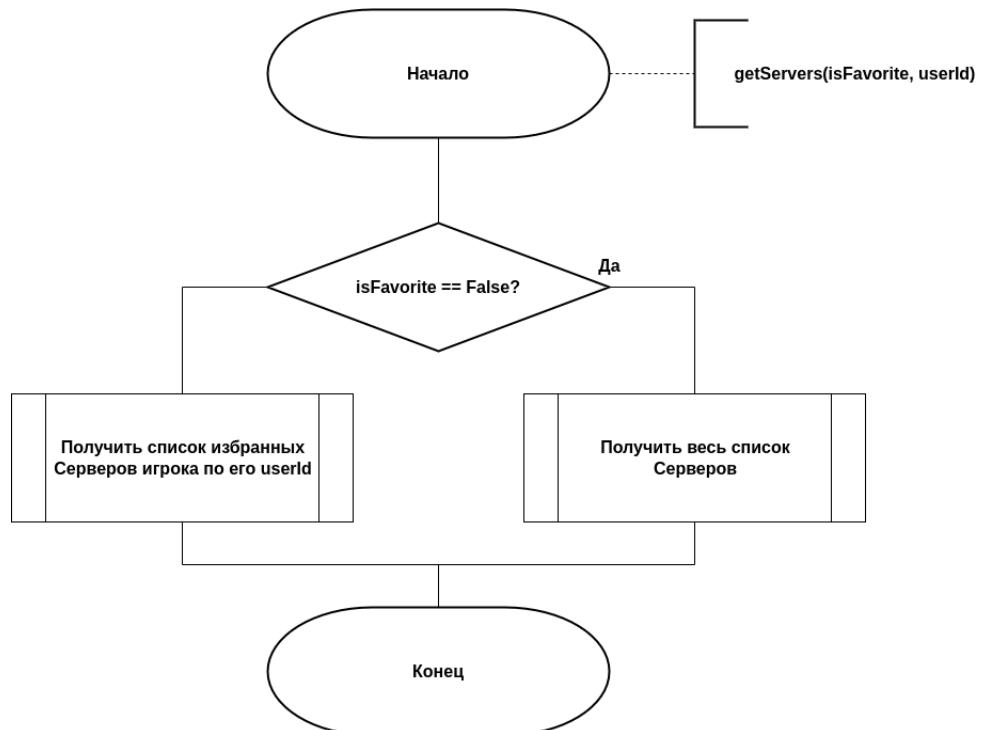


Рисунок 2.2 – Функция выбора серверов

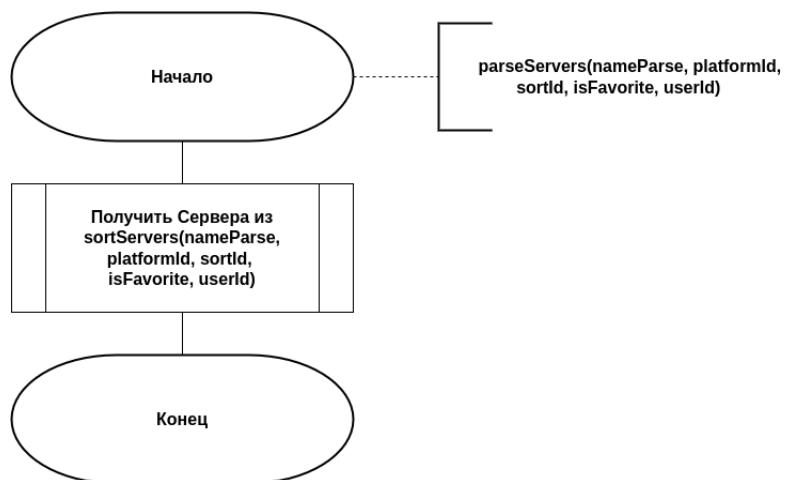


Рисунок 2.3 – Функция парсинга серверов

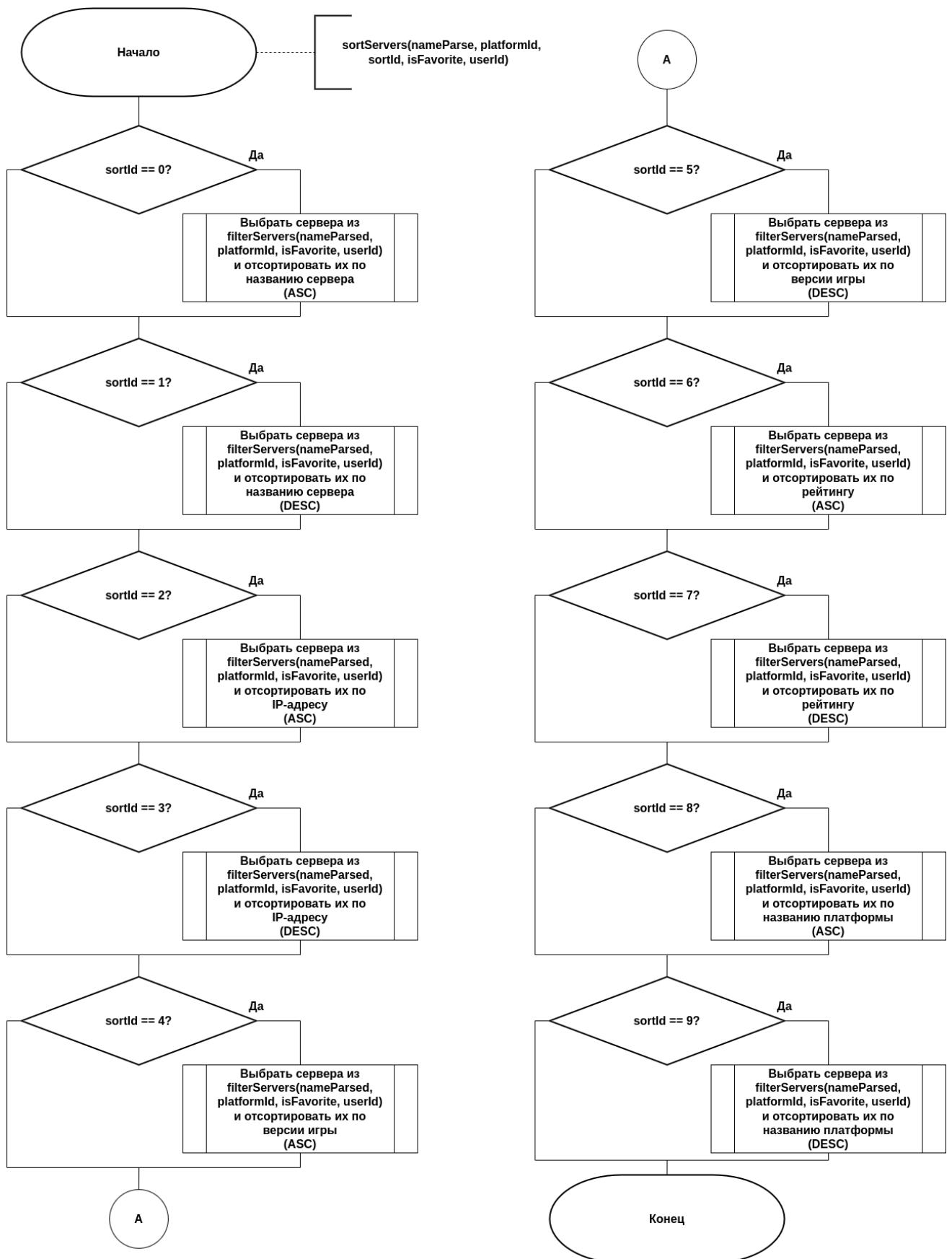


Рисунок 2.4 – Функция сортировки серверов



Рисунок 2.5 – Функция фильтрации серверов

2.3 Триггеры базы данных

В проектируемой базе данных определены следующие триггеры.

- Триггер установки роли пользователя.** Данный триггер срабатывает при операции «Insert» в таблицу «User» и вызывает функцию установки роли пользователя, которая выдает пользователю роль «User» или «Admin», если его логин «admin». Таким образом, в базе данных точно будет один администратор для управления сайтом. Схема алгоритма вызываемой функции представлена на рисунке 2.6.
- Триггеры изменения рейтинга сервера.** Один триггер срабатывает при операции «Insert» в таблицу «FavoriteServer» и вызывает функцию *увеличения* рейтинга добавленного сервера на единицу, а второй триггер –

при операции «Delete» из таблицы «FavoriteServer» и вызывает функцию **уменьшения рейтинга** удаляемого сервера на единицу. Схема алгоритма вызываемой функции представлена на рисунке 2.6.

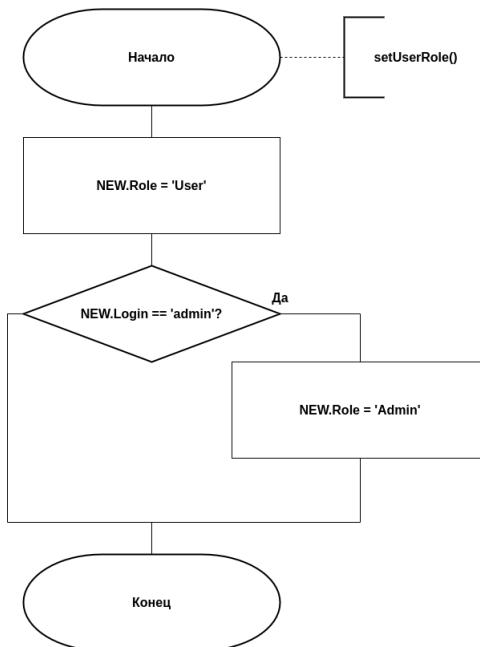


Рисунок 2.6 – Функция выдачи роли пользователю

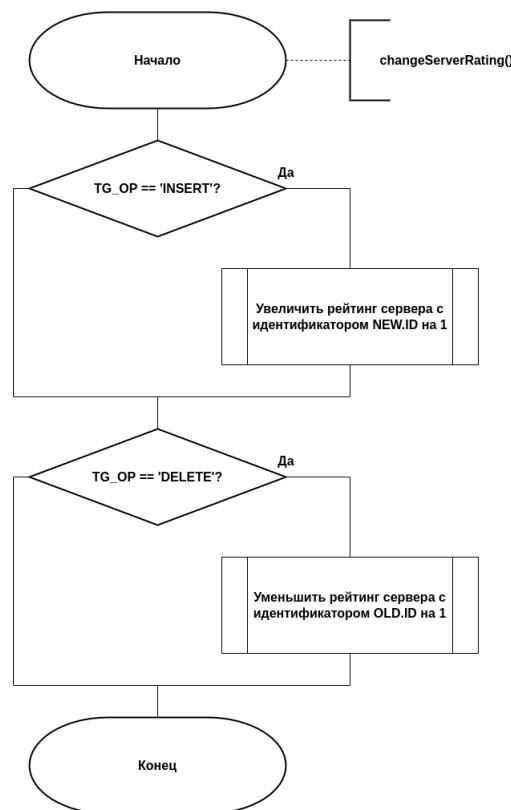


Рисунок 2.7 – Функция изменения рейтинга сервера

2.4 Роли базы данных

В проектируемой базе данных определены следующие роли.

- 1. Неавторизованный пользователь.** Данная роль обладает возможность просмотра таблиц серверов и платформ, так как общий список серверов доступен всем видам пользователей. Также обладает возможностью добавлять в таблицу «User» для регистрации нового аккаунта.
- 2. Авторизованный пользователь.** Данная роль обладает возможность просмотра таблиц серверов, платформ для списка серверов; просмотра таблиц «Hosting», «Player» и «ServerPlayer» для получения детальной информации о сервере; просмотра таблицы «FavoriteServer» для просмотра серверов из списка избранного. Также обладает возможностью добавлять и удаления в таблице «FavoriteServer» для добавления и удаления серверов из списка избранного.
- 3. Администратор сайта.** Данная роль обладает возможность просмотра, добавления и удаления на все определенные в базе данных таблицы.

Вывод

В данном разделе были подробно описаны все поля всех таблиц проектируемой базы данных. Были рассмотрены функции, которые выполняют работу на уровне базы данных. Также в данном разделе были описаны триггеры и введенные роли базы данных – для неавторизованного пользователя, для авторизованного пользователя и администратора сайта.

3 Технологическая часть

3.1 Средства реализации

В качестве языка программирования был выбран язык *C#* [13]. Он обладает возможность создания Web-приложений. Также данный язык программирования поддерживает принципы ООП, что крайне важно для поставленной задачи – имеется возможность создать понятную и простую структуру программного кода.

Фреймворком для работы с СУБД был выбран *Entity Framework Core* [14]. Данный фреймворк является основным для языка *C#* и имеет множество учебной литературы.

Visual Studio [15] был выбран в качестве среды разработки, так как имеется бесплатная версия для студентов, хороший дебаггер и полноценная поддержка языка *C#*.

Для разработки непосредственно Web-приложения используется фреймворк *ASP.NET* [16]. Данный фреймворк является абсолютно бесплатным и обладает широким функционалом для разработки. Имеет поддержку мультистраничных сайтов.

3.2 Архитектура приложения

Архитектура приложения построена на основе схемы MVC (модель-представление-контроллер) [17]. Использование данной схемы позволяет разрабатывать каждый из компонентов независимо от других. Это имеет следующие преимущества:

- изменения одного из компонентов не влияют на работоспособность других компонентов;
- имеется возможность подмены (Web-интерфейс может быть заменен на интерфейс десктопного приложения).

Схема взаимодействие компонентов MVC представлена на рисунке 3.1

3.3 Структура классов

Классы приложения разбиты на 3 основных слоя:

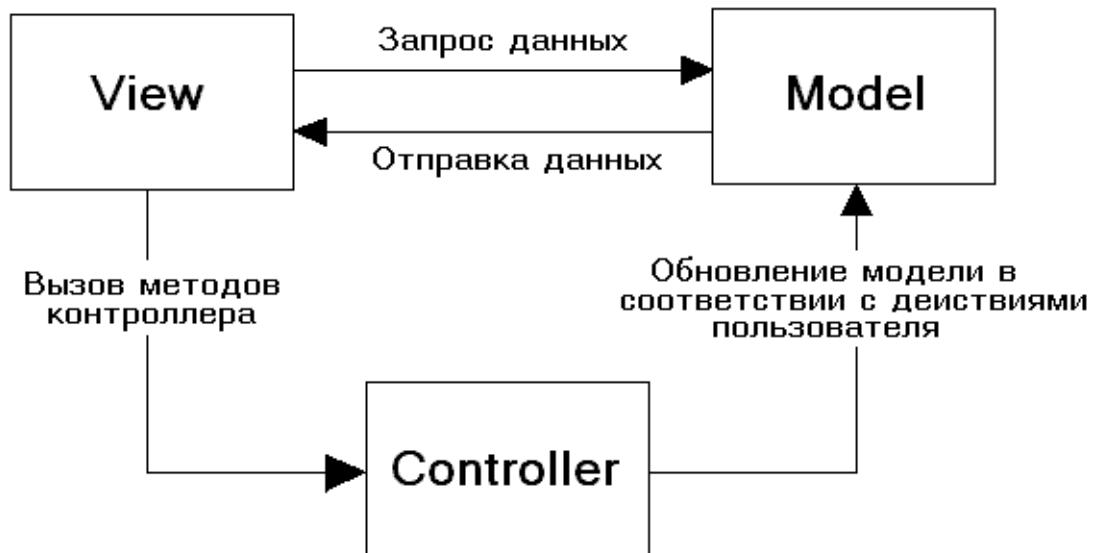


Рисунок 3.1 – Схема взаимодействие компонентов MVC

- слой доступа к данным, который состоит из классов моделей и репозиториев, для работы с данными;
- слой бизнес логики, состоящих из классов-сервисов, которые реализуют основную логику Web-приложения;
- слой контроллеров, которые, в ответ на действия пользователя, вызывают соответствующие методы слоя бизнес логики.

На рисунке 3.2 представлена схема взаимодействия классов между слоем доступа к данным и слоем бизнес логики приложения.

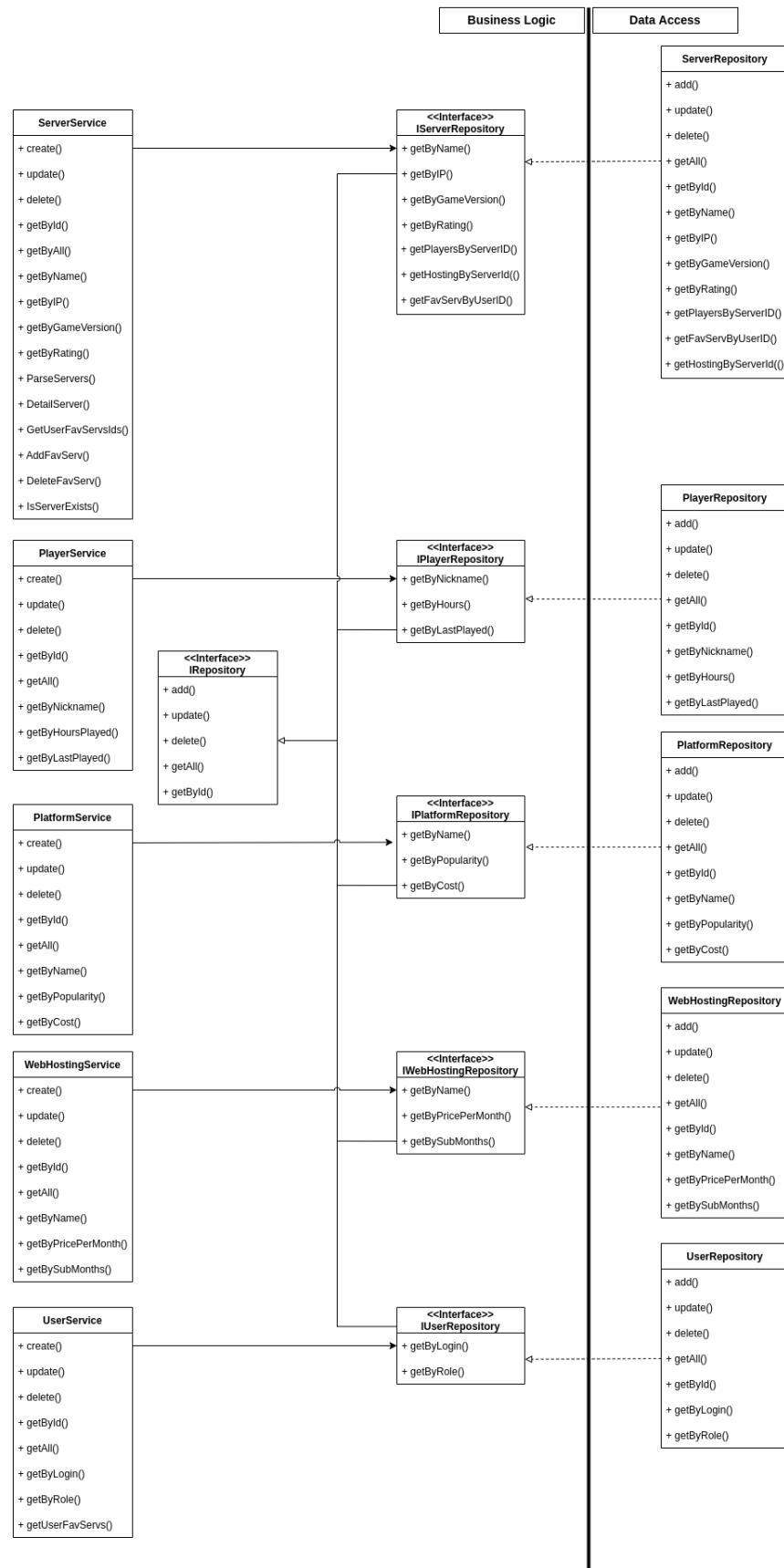


Рисунок 3.2 – Взаимодействие классов доступа к данным и бизнес логики

3.4 Выбор СУБД

3.4.1 Виды СУБД

Выбор системы управления базой данных является важной частью разработки программного продукта. Выбранная СУБД должна соответствовать всем требованиям, которые предъявляются при разработке. Рассмотрим наиболее популярные СУБД.

- **MySQL** [18] – является одной из самых популярных СУБД. Распространяется бесплатно. Из-за высокого спроса, часто выходят новые обновления, исправляющие ошибки и добавляющие много нового функционала. Не подходит для простых задач из-за трудной настройки. Не имеет ряда функционала, который в других СУБД реализован по умолчанию.
- **Microsoft SQL Server** [19] – собственная разработка компании Microsoft. Имеется поддержка для работы на операционной системе Linux, но наилучшую поддержку имеет для операционной системы Windows и иных продуктов компании Microsoft. Имеет проблемы с оптимизацией использования ресурсов. Проста в использовании.
- **Oracle** [20] – крайне популярна в крупных компаниях. Распространяется по подписке. Является одной из самых безопасных СУБД, так как каждая транзакция полностью изолирована друг от друга. Требует больших вычислительных ресурсов, тем самым не подходит для небольших проектов.
- **PostgreSQL** [21] – одна из самых популярных СУБД, которые распространяются бесплатно и имеют высокое качество. Имеет хорошую оптимизацию, особенно под операционную систему Linux. Часто используется при разработке Web-приложений и хорошо подходит для небольших проектов. Имеет трудную настройку для неподготовленного пользователя.

3.4.2 Вывод

Выбор СУБД будет произведен, исходя из следующих критериев:

- К1 – подробная документация;

- К2 – высокий уровень оптимизации;
- К3 – подходит для разработки небольших проектов;
- К4 – поддержка различных форматов файлов;
- К5 – распространяется бесплатно.

Таблица 3.1 – Сравнение СУБД

Название	К1	К2	К3	К4	К5
«MySQL»	+	+	-	-	+
«Microsoft SQL Server»	-	-	+	+	-
«Oracle»	+	-	-	+	-
«PostgreSQL»	-	+	+	+	+

Исходя из того, что разрабатывается небольшое Web-приложение, то наилучшим образом подходит СУБД PostgreSQL, так как она обладает всеми необходимыми требованиями.

3.5 Реализация функций

При разработке были реализованы следующие функции базы данных, описанные в разделе 2.2.

1. Функция выбора серверов (листинг 3.1).

Листинг 3.1 — Функция выбора серверов

```

1 CREATE OR REPLACE FUNCTION getServers(isFavorite boolean, userId integer)
2   RETURNS TABLE (SrvId integer, SrvName varchar, SrvIp varchar, SrvGameVersion
3   ↵ varchar, SrvRating integer, SrvHostingID integer, SrvPlatformID integer)
4 AS $$
5 BEGIN
6   IF (isFavorite is False) THEN
7     return query
8     SELECT *
9       FROM public."Server";
10    ELSE
11      return query
12        SELECT Fav."ServerID", Fav."Name", Fav."Ip", Fav."GameVersion",
13   ↵ Fav."Rating", Fav."HostingID", Fav."PlatformID"

```

```

12      FROM (public."Server" AS S JOIN public."FavoriteServer" AS FS ON S."Id" =
13      ↳ FS."ServerID") AS Fav
14          WHERE Fav."UserID" = userId;
15      END IF;
16  END;
$$ LANGUAGE plpgsql;

```

2. Функция фильтрации серверов (листинг 3.2).

Листинг 3.2 — Функция фильтрации серверов

```

1 CREATE OR REPLACE FUNCTION filterServers(nameParse varchar, platform_id integer,
2     ↳ isFavorite boolean, userId integer)
3 RETURNS TABLE (FltId integer, FltName varchar, FltIp varchar, FltGameVersion
4     ↳ varchar, FltRating integer, FltHostingID integer, FltPlatformID integer)
5 AS $$
6 BEGIN
7     IF (nameParse IS NULL) THEN
8         nameParse = '';
9     END IF;
10
11     IF (platform_id > 0) THEN
12         return query
13         SELECT *
14             FROM getServers(isFavorite, userId) as S
15             WHERE S.SrvPlatformID = platform_id AND S.SrvName LIKE '%' || nameParse
16             || '%';
17     ELSE
18         return query
19         SELECT *
20             FROM getServers(isFavorite, userId) as S
21             WHERE S.SrvName LIKE '%' || nameParse || '%';
22     END IF;
23 END;
$$ LANGUAGE plpgsql;

```

3. Функция сортировки серверов (листинги A.1-A.2)

4. Функция парсинга серверов (обертка) (листинг 3.3).

Листинг 3.3 — Функция парсинга серверов

```

1 CREATE OR REPLACE FUNCTION parseServers(nameParse varchar, platform_id integer,
2     ↳ sortId integer, isFavorite boolean, userId integer)
3 RETURNS TABLE (Id integer, Name varchar, Ip varchar, GameVersion varchar,
4     ↳ Rating integer, HostingID integer, PlatformID integer)

```

```

3   AS $$  

4     BEGIN  

5       return query  

6       SELECT * FROM sortServers(nameParse, platform_id, sortId, isFavorite,  

7       ↳ userId);  

8     END;  

9   $$ LANGUAGE plpgsql;

```

3.6 Реализация триггеров

В проектируемой базе данных определены следующие триггеры, которые описаны в разделе 2.2.

1. Триггеры изменения рейтинга сервера и функция, реализующая работу этих триггеров (листинг 3.4).

Листинг 3.4 — Триггеры изменения рейтинга сервера

```

1  CREATE OR REPLACE FUNCTION changeServerRating() RETURNS trigger AS $$  

2    BEGIN  

3      IF (TG_OP = 'INSERT') THEN  

4          UPDATE public."Server"  

5              SET "Rating" = "Rating" + 1  

6              WHERE "Id" = NEW."ServerID";  

7  

8          RETURN NEW;  

9      ELSIF (TG_OP = 'DELETE') THEN  

10         UPDATE public."Server"  

11             SET "Rating" = "Rating" - 1  

12             WHERE "Id" = OLD."ServerID";  

13  

14         RETURN OLD;  

15     END IF;  

16   END;  

17   $$ LANGUAGE plpgsql;  

18  

19  

20  -- Триггер повышения рейтинга сервера  

21  CREATE OR REPLACE TRIGGER incServerRatingTrigger AFTER INSERT ON  

22    ↳ public."FavoriteServer"  

23    FOR EACH ROW EXECUTE PROCEDURE changeServerRating();  

24  

25  -- Триггер понижения рейтинга сервера  

26  CREATE OR REPLACE TRIGGER decServerRatingTrigger AFTER DELETE ON  

27    ↳ public."FavoriteServer"  

28    FOR EACH ROW EXECUTE PROCEDURE changeServerRating();

```

2. Триггер установки роли пользователя и функция, реализующая его работу (листинг 3.5).

Листинг 3.5 — Триггер установки роли пользователя

```

1 CREATE OR REPLACE FUNCTION setUserRole() RETURNS trigger AS $$  

2 BEGIN  

3     NEW."Role" := 'User';  

4  

5     -- Если ник admin, то роль выдать админа  

6     IF (NEW."Login" = 'admin') THEN  

7         NEW."Role" := 'Admin';  

8     END IF;  

9  

10    RETURN NEW;  

11 END;  

12 $$ LANGUAGE plpgsql;  

13  

14 CREATE OR REPLACE TRIGGER setUserRoleTrigger BEFORE INSERT ON public."User"  

15     FOR EACH ROW EXECUTE PROCEDURE setUserRole();
```

3.7 Реализация ролевой модели

В базу данных были введены следующие роли, которые были рассмотрены в разделе 2.4.

1. Неавторизованный пользователь. В листинге 3.6 представлено создание роли и раздача необходимых прав.

Листинг 3.6 — Неавторизованный пользователь

```

1 CREATE ROLE non_auth_user WITH  

2     LOGIN  

3     NOSUPERUSER  

4     NOCREATEDB  

5     NOCREATEROLE  

6     NOROLE  

7     PASSWORD 'non_auth_user'  

8     CONNECTION LIMIT -1;  

9  

10    -- Привилегии  

11    GRANT SELECT ON public."Server" TO non_auth_user;  

12    GRANT SELECT ON public."Platform" TO non_auth_user;  

13    GRANT SELECT ON public."User" TO non_auth_user;  

14  

15    GRANT INSERT ON public."User" TO non_auth_user;
```

2. Авторизованный пользователь. В листинге 3.7 представлено создание роли и раздача необходимых прав.

Листинг 3.7 — Авторизованный пользователь

```
1 CREATE ROLE auth_user WITH
2   LOGIN
3   NOSUPERUSER
4   NOCREATEDB
5   NOCREATEROLE
6   NOREPLICATION
7   PASSWORD 'auth_user'
8   CONNECTION LIMIT -1;
9
10  -- Привилегии
11  GRANT SELECT ON public."Server" TO auth_user;
12  GRANT SELECT ON public."Platform" TO auth_user;
13  GRANT SELECT ON public."FavoriteServer" TO auth_user;
14  GRANT SELECT ON public."ServerPlayer" TO auth_user;
15  GRANT SELECT ON public."Player" TO auth_user;
16  GRANT SELECT ON public."User" TO auth_user;
17  GRANT SELECT ON public."WebHosting" TO auth_user;
18
19  GRANT INSERT ON public."FavoriteServer" TO auth_user;
20  GRANT INSERT ON public."User" TO auth_user;
21
22  GRANT UPDATE ON public."Server" TO auth_user;
23
24  GRANT DELETE ON public."FavoriteServer" TO auth_user;
```

3. Администратор сайта. В листинге 3.8 представлено создание роли и раздача необходимых прав.

Листинг 3.8 — Администратор сайта

```
1 CREATE ROLE admin WITH
2   LOGIN
3   SUPERUSER
4   CREATEDB
5   CREATEROLE
6   NOREPLICATION
7   PASSWORD 'admin'
8   CONNECTION LIMIT -1;
9
10  -- Привилегии
11  GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO admin;
```

3.8 Полученный результат

Спроектировано мультистраничное Web-приложение для поиска серверов игры. В зависимости от роли пользователя меняется и интерфейс сайта, и его функционал.

- 1. Возможности неавторизованного пользователя.** Неавторизованный пользователь может увидеть список серверов с указанием платформы, на которой располагается каждый сервер, что показано на рисунке 3.3. При этом также доступны сортировка этого списка серверов и его фильтрация, в том числе, по различным платформам, что отображено на рисунке 3.4. Также неавторизованному пользователю доступна страница детальной информации о сервере, но доступна лишь информация о самом сервере (рисунок 3.5). Для того, чтобы увидеть подробную информацию, пользователь должен авторизоваться на сайте (рисунок 3.6) или зарегистрироваться на сайте (рисунок 3.7).
- 2. Возможности авторизованного пользователя.** Главная страница сайта со списком серверов для авторизованного пользователя выглядит уже по-другому (рисунок 3.8). Теперь на строчке каждого сервера появилась кнопка, которая позволяет добавить данный сервер в список избранных серверов (или кнопка удаления сервера из избранного, если таковой был добавлен), также появилась вкладка для перехода к избранным серверам в «шапке» сайта.

Страница детальной информации о сервере также изменилась – теперь там выводится информация о хостинге, на котором расположен данный сервер, и список игроков, играющих на этом сервере, что видно на рисунке 3.9.

Также на рисунке 3.10 можно увидеть страницу избранных серверов некоторого пользователя. Здесь также доступна вся информация о серверах, список можно также отсортировать и отфильтровать.

- 3. Возможности администратора сайта** Администратор сайта, помимо вышеперечисленного списка для авторизованного пользователя, обладает рядом важных возможностей. В «шапке» сайта (рисунок 3.11) стала доступна новая ссылка – ссылка на панель управления сайтом. Как видно

из рисунка 3.12, данная панель отображает две страницы управления – страница управления серверами и страница управления ролями пользователей.

Страница управления серверами (рисунок 3.13) обладает рядом возможностей – добавить сервер (страница добавления показана на рисунке 3.14), обновить сервер (рисунок 3.15) или удалить его.

Страница управления ролями пользователями (рисунок 3.16) позволяет изменять роль выбранного пользователя (рисунок 3.17). При этом запрещено изменять свою собственную роль и изменять роль администратору с логином «admin», чтобы на сайте всегда был минимум один администратор.

The screenshot shows the homepage of the 'Server IN Game' website. At the top, there is a logo with three blue horizontal bars and the text 'ServerING'. To the right is a 'LOGIN' button. Below the header, the text 'Server IN Game' and 'Best Servers For You Are Here' is displayed. A search bar with 'Name' and 'Platofrm' dropdown menus, and a 'FILTR' button is present. Below this is a table listing five servers:

Name	IP	Game Version	Platform	Rating	Information
AdventureBedrock	155.24.130.237	1.17.8	PC	0	DETAIL
AdventureGrief	166.26.224.118	1.16.5	Android	0	DETAIL
AdventureNoDonate	54.48.64.106	1.19.2	Xbox One	0	DETAIL
AdventureServer	189.173.29.47	1.16.1	iOS	0	DETAIL

Рисунок 3.3 – Главная страница сайта (неавторизованный пользователь)

The screenshot shows the homepage of the ServerING website. At the top right is a blue 'LOGIN' button. The main title 'Server IN Game' is centered above the subtitle 'Best Servers For You Are Here'. Below the subtitle is a search bar with 'Name' and 'Platofrm' dropdown fields set to 'PlayStation 5', and a 'FILTR' button. A table lists five servers: AppleCreative, AppleNether, EnderdragonApple, and EnderdragonMinecraft, all running on PlayStation 5 with version 1.16.7. Each server entry includes a small rating icon and a 'DETAIL' button.

Name	IP	Game Version	Platform	Rating	Information
AppleCreative	62.42.63.217	1.10.1	PlayStation 5	0	DETAIL
AppleNether	17.53.57.123	1.12.3	PlayStation 5	0	DETAIL
EnderdragonApple	26.86.26.57	1.18.4	PlayStation 5	0	DETAIL
EnderdragonMinecraft	16.170.171.164	1.16.7	PlayStation 5	0	DETAIL

Рисунок 3.4 – Фильтрация списка серверов по платформам

The screenshot shows the detailed information page for the 'AppleCreative' server. At the top left is the 'ServerING' logo and a blue 'LOGIN' button. The main title is 'Server: AppleCreative'. Below it is a 'Server Info' section with the following details: IP 62.42.63.217, Game Version 1.10.1, and Rating 0. A large button at the bottom encourages users to 'Log In to see players of this Server'. At the very bottom are links for 'Back To Top' and copyright information: '© МГТУ Им. Н. Э. Баумана, ИУ7'.

Рисунок 3.5 – Страница подробной информации о сервере (неавторизованный пользователь)

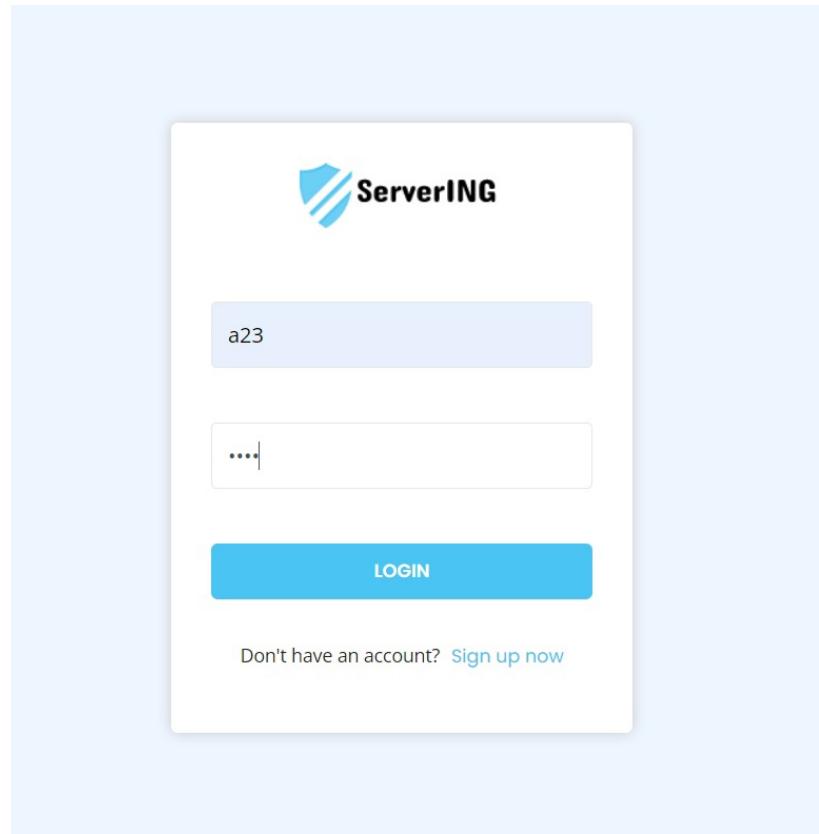


Рисунок 3.6 – Страница авторизации для пользователя

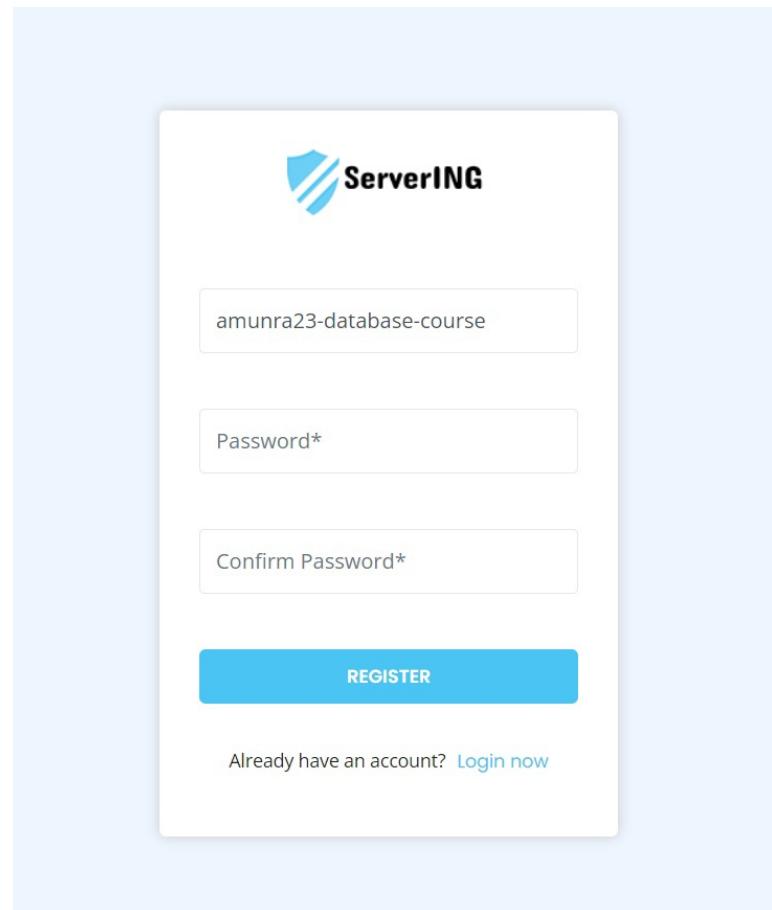


Рисунок 3.7 – Страница регистрации для нового пользователя

The screenshot shows the homepage of the ServerIN Game website. At the top right, there are buttons for "FAVORITE SERVERS", "LOGIN: AMUNRA23", and "LOGOUT". The main title "Server IN Game" is centered above the subtitle "Best Servers For You Are Here". Below the subtitle is a search bar with fields for "Name" and "Platform" (set to "All") and a "FILTR" button. A table lists four servers: AdventureBedrock, AdventureGrief, AdventureNoDonate, and AdventureServer, each with details like IP, Game Version, Platform, Rating, and buttons for "DETAIL" and "ADD".

Name	IP	Game Version	Platform	Rating	Information	Favorite
AdventureBedrock	155.24.130.237	1.17.8	PC	0	DETAIL	ADD
AdventureGrief	166.26.224.118	1.16.5	Android	0	DETAIL	ADD
AdventureNoDonate	54.48.64.106	1.19.2	Xbox One	0	DETAIL	ADD
AdventureServer	189.173.29.47	1.16.1	iOS	0	DETAIL	ADD

Рисунок 3.8 – Главная страница сайта (авторизованный пользователь)

The screenshot shows the detailed information page for the "AdventureBedrock" server. At the top right, there are buttons for "FAVORITE SERVERS", "LOGIN: AMUNRA23", and "LOGOUT". The main title is "Server: AdventureBedrock". The page is divided into two sections: "Server Info" and "Web Hosting Info". Under "Server Info", there are entries for IP (155.24.130.237), Game Version (1.17.8), and Rating (0). Under "Web Hosting Info", there are entries for Name (<https://www.dodson.com/category.htm>), Price Per Months (5000), and Subscribe Months (4). Below these sections is a table titled "Players" showing five players with their Nickname, Hours Played, and Last Played.

Nickname	Hours Played	Last Played
susanwolfe	23	06.09.2021 0:00:00
armitchell	51	16.03.2022 0:00:00
banderson	83	04.06.2022 0:00:00
floresrobin	75	01.07.2022 0:00:00

Рисунок 3.9 – Страница подробной информации о сервере (авторизованный пользователь)

The screenshot shows a user interface for managing favorite servers. At the top right, there are buttons for 'FAVORITE SERVERS', 'LOGIN: AMUNRA23', and 'LOGOUT'. Below this, a large heading says 'Your Favorite Servers'. There is a search bar with fields for 'Name' and 'Platofrm' (set to 'All') with a 'FILTR' button. A table lists four servers:

Name	IP	Game Version	Platform	Information	Favorite
AdventureBedrock	155.24.130.237	1.17.8	PC	DETAIL	DELETE
AdventureServer	189.173.29.47	1.16.1	iOS	DETAIL	DELETE
AppleGrief	73.79.19.160	1.17.2	Xbox One	DETAIL	DELETE
BedrockCreeper	98.167.247.43	1.14.5	PlayStation 4	DETAIL	DELETE

A small number '1' is displayed in a box at the bottom left.

Рисунок 3.10 – Страница избранных серверов авторизованного пользователя

The screenshot shows a landing page for 'Server IN Game'. At the top right, there are buttons for 'FAVORITE SERVERS', 'SERVER CONTROL', 'LOGIN: ADMIN', and 'LOGOUT'. The main heading is 'Server IN Game' followed by the subtext 'Best Servers For You Are Here'. There is a search bar with fields for 'Name' and 'Platofrm' (set to 'All') with a 'FILTR' button. A table lists five servers, showing a mix of red and green 'DETAIL' and 'DELETE' buttons:

Name	IP	Game Version	Platform	Rating	Information	Favorite
AdventureBedrock	155.24.130.237	1.17.8	PC	2	DETAIL	DELETE
AdventureGrief	166.26.224.118	1.16.5	Android	0	DETAIL	ADD
AdventureNoDonate	54.48.64.106	1.19.2	Xbox One	1	DETAIL	DELETE
AdventureServer	189.173.29.47	1.16.1	iOS	2	DETAIL	DELETE

Рисунок 3.11 – Главная страница сайта (администратор сайта)

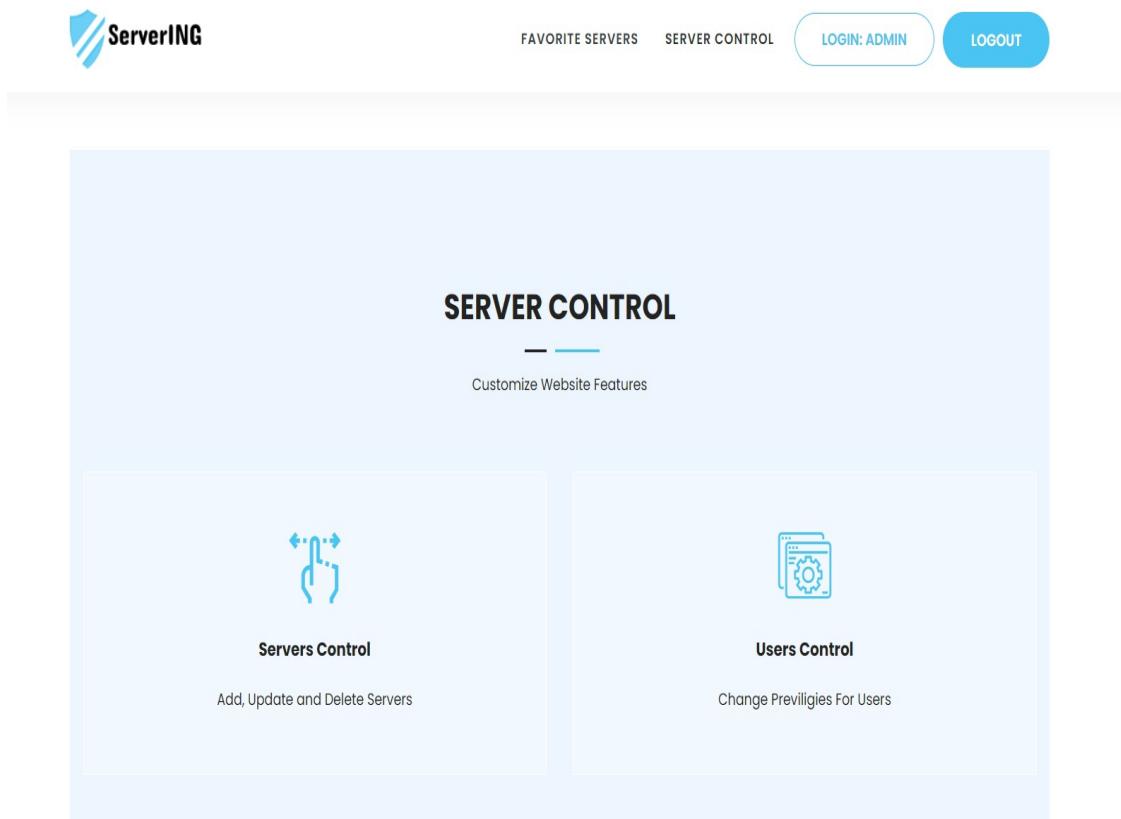


Рисунок 3.12 – Страница управления сайтом

The screenshot shows the 'Servers Control' page. At the top, there are navigation links for 'FAVORITE SERVERS', 'SERVER CONTROL', 'LOGIN: ADMIN', and 'LOGOUT'. The main title 'Servers Control' is centered at the top. Below it is a teal-colored button labeled 'ADD SERVER'. There is a search bar with fields for 'Name' and 'Platform' (set to 'All') and a 'FILTR' button. A table lists five server entries with columns for Name, IP, Game Version, Platform, Update, and Delete. Each entry includes a blue 'UPDATE' button and a red 'DELETE' button.

Name	IP	Game Version	Platform	Update	Delete
AdventureBedrock	155.24.130.237	1.17.8	PC	UPDATE	DELETE
AdventureGrief	166.26.224.118	1.16.5	Android	UPDATE	DELETE
AdventureNoDonate	54.48.64.106	1.19.2	Xbox One	UPDATE	DELETE
AdventureServer	189.173.29.47	1.16.1	iOS	UPDATE	DELETE
AppleBedrock	22.226.10.168	1.12.6	PC	UPDATE	DELETE

Рисунок 3.13 – Страница управления серверами

The screenshot shows the 'Server Adding' form. It includes fields for Name (Test-amunra23-dbc), IP (256.256.256.256), Game Version (1.11.1), Platform (Xbox One), and Web Hosting (https://www.dodson.com/category.htm). A green 'SUBMIT' button is at the bottom.

FAVORITE SERVERS SERVER CONTROL LOGIN: ADMIN LOGOUT

Server Adding

Name
Test-amunra23-dbc

IP
256.256.256.256

Game Version
1.11.1

Platform
Xbox One

Web Hosting
https://www.dodson.com/category.htm

SUBMIT

Рисунок 3.14 – Страница добавления нового сервера

The screenshot shows the 'Update Server' form. It includes fields for Name (AdventureBedrock), IP (155.24.130.237), Game Version (1.17.8), Platform (PC), and Web Hosting (https://www.dodson.com/category.htm). A green 'SUBMIT' button is at the bottom.

FAVORITE SERVERS SERVER CONTROL LOGIN: ADMIN LOGOUT

Update Server

Name
AdventureBedrock

IP
155.24.130.237

Game Version
1.17.8

Platform
PC

Web Hosting
https://www.dodson.com/category.htm

SUBMIT

Рисунок 3.15 – Страница обновления сервера

The screenshot shows a user role control interface. At the top, there is a navigation bar with the ServerING logo, 'FAVORITE SERVERS', 'SERVER CONTROL', and buttons for 'LOGIN: AMUNRA' and 'LOGOUT'. Below the navigation bar, the title 'User Role Control' is displayed. A search bar with 'Login' and 'FIND' buttons is present. A table lists three users: 'admin' (Admin, UNABLE), 'amunra' (Admin, YOU), and 'amunra23' (User, CHANGE). At the bottom, there is a pagination indicator '1' and a 'FORWARD' button.

Login	Current Role	Detail
admin	Admin	UNABLE
amunra	Admin	YOU
amunra23	User	CHANGE

Рисунок 3.16 – Страница управления ролями пользователей сайта

The screenshot shows a user role change form for the user 'amunra23'. At the top, there is a navigation bar with the ServerING logo, 'FAVORITE SERVERS', 'SERVER CONTROL', and buttons for 'LOGIN: AMUNRA' and 'LOGOUT'. The main content area displays the user information: 'User: amunra23' and 'User Role'. Two radio buttons are shown: 'Admin' (unchecked) and 'User' (checked). A green 'SUBMIT' button is at the bottom. At the very bottom, there is a link 'Back To Top'.

Рисунок 3.17 – Страница изменения роли пользователя

Вывод

В данном разделе были рассмотрены средства реализации программного продукта, его архитектура и структура классов. Также были приведены реализации функций, триггеров и ролей базы данных. Был проведен выбор СУБД, в результате которого был выбран PostgreSQL. Также приведен результат работы получившегося веб-приложения.

4 Исследовательская часть

4.1 Цель эксперимента

Целью эксперимента является сравнение времени выполнения запросов при использовании индексов и без них.

Созданные индексы будут иметь тип *BTree*. Данный тип индексов используется в ситуациях, когда данные можно отсортировать.

Эксперименты проводились на таблицах, которые имеют более 300 записей, чтобы продемонстрировать применение индексов с наибольшей наглядностью.

4.2 Результаты эксперимента

4.2.1 Поиск по первичному ключу

На рисунке 4.1 представлены замеры времени выполнения запроса при поиске записи в таблице «Server» по первичному ключу.

```
ServerING# explain (analyze) select * from public."Server" as S where S."Id" = 23;
                                         QUERY PLAN
-----
Index Scan using "PK_Server" on "Server" s  (cost=0.27..8.29 rows=1 width=50) (actual time=0.023..0.024 rows=1 loops=1)
  Index Cond: ("Id" = 23)
Planning Time: 2.448 ms
Execution Time: 0.080 ms
(4 rows)
```

Рисунок 4.1 – Поиск по первичному ключу (без индекса)

В листинге 4.1 показано создание индекса «server_id_index» для первичного ключа таблицы «Server».

Листинг 4.1 – Индекс для первичного ключа таблицы «Server»

```
1 CREATE INDEX server_id_index on public."Server" using btree("Id");
```

При этом на рисунке 4.2 представлены результаты замера времени запроса поиска записи в таблице «Server» после создания индекса.

Как видно из результата, создание индекса для первичного ключа не принесло прироста производительности выполнения запроса. Это связано с тем, что подобные индексы по умолчанию создаются для первичных ключей таблиц в базе данных.

```

ServerING=# explain (analyze) select * from public."Server" as S where S."Id" = 23;
                                         QUERY PLAN
-----
Index Scan using server_id_index on "Server" s  (cost=0.27..8.29 rows=1 width=50) (actual time=0.052..0.053 rows=1 loops=1)
  Index Cond: ("Id" = 23)
Planning Time: 1.183 ms
Execution Time: 0.081 ms
(4 行)

```

Рисунок 4.2 – Поиск по первичному ключу (с индексом)

4.2.2 Фильтрация по столбцу таблицы

На рисунке 4.3 приведены замеры времени выполнения запроса при фильтрации записей в таблице «Server» по столбцу «Rating».

```

ServerING=# explain (analyze) select * from public."Server" as S where S."Rating" < 23;
                                         QUERY PLAN
-----
Seq Scan on "Server" s  (cost=0.00..502.26 rows=73 width=50) (actual time=0.020..0.398 rows=74 loops=1)
  Filter: ("Rating" < 23)
  Rows Removed by Filter: 427
Planning Time: 0.109 ms
Execution Time: 0.423 ms
(5 行)

```

Рисунок 4.3 – Фильтрация по столбцу «Rating» (без индекса)

В листинге 4.2 показано создание индекса «server_rating_index» для столбца «Rating» таблицы «Server».

Листинг 4.2 – Индекс для столбца «Rating» таблицы «Server»

```

1 CREATE INDEX server_rating_index on public."Server" using btree("Rating");

```

После создания индекса, были снова выполнены замеры времени выполнения запроса с фильтрацией записей в таблице «Server» по столбцу «Rating», что показано на рисунке 4.4.

```

ServerING=# explain (analyze) select * from public."Server" as S where S."Rating" < 23;
                                         QUERY PLAN
-----
Bitmap Heap Scan on "Server" s  (cost=4.71..202.09 rows=73 width=50) (actual time=0.022..0.066 rows=74 loops=1)
  Recheck Cond: ("Rating" < 23)
  Heap Blocks: exact=58
    -> Bitmap Index Scan on server_rating_index  (cost=0.00..4.70 rows=73 width=0) (actual time=0.013..0.013 rows=74 loops=1)
      Index Cond: ("Rating" < 23)
Planning Time: 0.124 ms
Execution Time: 0.101 ms
(7 行)

```

Рисунок 4.4 – Фильтрация по столбцу «Rating» (с индексом)

Из полученного результата видно, что время выполнения запроса уменьшилось примерно в 4.19 раз после создания индекса для столбца «Rating».

Вывод

Как видно из результатов, применение индексов позволяет ускорить выполнение запросов. При этом индексы не следует создавать для первичных ключей таблиц, так как индексы для них создаются по умолчанию. Таким образом, индексы являются важным инструментов оптимизации работы с базой данных.

ЗАКЛЮЧЕНИЕ

Было создано Web-приложение для поиска серверов кросплатформенной игры. При этом были выполнены следующие поставленные задачи:

- рассмотрены существующие решения;
- выбрана модель хранения данных;
- разработана база данных;
- выделены роли пользователей;
- выбран необходимый набор технологий для разработки;
- создан программный продукт, который решает поставленную цель.

В ходе курсового проекта удалось выделить важную проблему существующих сайтов с серверами для игр – отсутствие фильтрации серверов по различным игровым платформам. При этом был приведен пример решения данной проблемы.

Также в результате проведенного эксперимента было обнаружено, что применение индексов может существенно уменьшить время выполнения запросов в базу данных. Но стоит отметить, что создание индексов для первичных ключей таблиц не приносит результатов, так как для данных полей индексы создаются автоматически.

Таким образом, все поставленные задачи выполнены, а цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальный сайт Minecraft [Электронный ресурс]. — URL: <https://www.minecraft.net/ru-ru> (дата обращения: 5.05.2022).
2. Сервера Майнкрафт «MinecraftRating» [Электронный ресурс]. — URL: <https://minecraftrating.ru/> (дата обращения: 5.05.2022).
3. Официальный сайт Counter-Strike [Электронный ресурс]. — URL: <https://blog.counter-strike.net/> (дата обращения: 5.05.2022).
4. Сервера Counter-Strike «Сервера КС» [Электронный ресурс]. — URL: <https://servera-csgo.ru/> (дата обращения: 5.05.2022).
5. Сервера Майнкрафт «Best Minecraft Servers» [Электронный ресурс]. — URL: <https://best-minecraft-servers.co/> (дата обращения: 5.05.2022).
6. Сервера Counter-Strike «Game Tracker» [Электронный ресурс]. — URL: <https://www.gametracker.com/search/cs/> (дата обращения: 5.05.2022).
7. Нотации модели сущность-связь (ER диаграммы) [Электронный ресурс]. — URL: <https://pro-prof.com/archives/8126> (дата обращения: 5.05.2022).
8. Что такое база данных? [Электронный ресурс]. — URL: <https://www.oracle.com/cis/database/what-is-database/> (дата обращения: 5.05.2022).
9. Дореляционные модели данных [Электронный ресурс]. — URL: https://spravochnick.ru/bazy_dannyh/dorelyacionnye_modeli_dannyh/ (дата обращения: 5.05.2022).
10. Что такое реляционная база данных? [Электронный ресурс]. — URL: <https://aws.amazon.com/ru relational-database> (дата обращения: 5.05.2022).
11. SQL [Электронный ресурс]. — URL: <https://blog.skillfactory.ru/glossary/sql/> (дата обращения: 5.05.2022).

12. Нереляционные данные и базы данных NoSQL [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/azure/architecture/data-guide/big-data/non-relational-data> (дата обращения: 5.05.2022).
13. C Sharp [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 5.05.2022).
14. Entity Framework Core [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/ef/core/> (дата обращения: 5.05.2022).
15. Visual Studio [Электронный ресурс]. — URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 5.05.2022).
16. ASP.NET [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-6.0> (дата обращения: 5.05.2022).
17. MVC [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-6.0> (дата обращения: 5.05.2022).
18. MySQL [Электронный ресурс]. — URL: <https://dev.mysql.com/doc/> (дата обращения: 5.05.2022).
19. SQL Server [Электронный ресурс]. — URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2022> (дата обращения: 5.05.2022).
20. Oracle [Электронный ресурс]. — URL: <https://www.oracle.com/cis/database/> (дата обращения: 5.05.2022).
21. PostgreSQL [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/> (дата обращения: 5.05.2022).

ПРИЛОЖЕНИЕ А

ФУНКЦИЯ СОРТИРОВКИ СПИСКА СЕРВЕРОВ

В листингах А.1-А.2 представлена функция сортировки списка серверов.

Листинг А.1 — Функция сортировки серверов

```
1 CREATE OR REPLACE FUNCTION sortServers(nameParse varchar, platform_id integer, sortId
2   → integer, isFavorite boolean, userId integer)
3 RETURNS TABLE (Id integer, Name varchar, Ip varchar, GameVersion varchar, Rating
4   → integer, HostingID integer, PlatformID integer)
5 AS $$
6 BEGIN
7   IF (sortId = 0) THEN -- Name ASC
8     return query
9     SELECT *
10    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
11    ORDER BY S.FltName;
12  ELSEIF (sortId = 1) THEN -- Name DESC
13    return query
14    SELECT *
15    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
16    ORDER BY S.FltName DESC;
17  ELSEIF (sortId = 2) THEN -- Ip ASC
18    return query
19    SELECT *
20    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
21    ORDER BY S.FltIp;
22  ELSEIF (sortId = 3) THEN -- Ip DESC
23    return query
24    SELECT *
25    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
26    ORDER BY S.FltIp DESC;
27  ELSEIF (sortId = 4) THEN -- GameVersion ASC
28    return query
29    SELECT *
30    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
31    ORDER BY S.FltGameVersion;
32  ELSEIF (sortId = 5) THEN -- GameVersion DESC
33    return query
34    SELECT *
35    FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
36    ORDER BY S.FltGameVersion DESC;
```

Листинг А.2 — Функция сортировки серверов

```
1      ELSEIF (sortId = 6) THEN -- Rating ASC
2          return query
3          SELECT *
4              FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
5                  ORDER BY S.FltRating;
6      ELSEIF (sortId = 7) THEN -- Rating DESC
7          return query
8          SELECT *
9              FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
10                 ORDER BY S.FltRating DESC;
11     ELSEIF (sortId = 8) THEN -- Platform ASC
12         return query
13         SELECT PS.FltId, PS.FltName, PS.FltIp, PS.FltGameVersion, PS.FltRating,
14             PS.FltHostingID, PS.FltPlatformID
15             FROM (filterServers(nameParse, platform_id, isFavorite, userId) as S JOIN
16             public."Platform" as P on S.FltPlatformID = P."Id") as PS
17                 ORDER BY PS."Name";
18     ELSEIF (sortId = 9) THEN -- Platform DESC
19         return query
20         SELECT PS.FltId, PS.FltName, PS.FltIp, PS.FltGameVersion, PS.FltRating,
21             PS.FltHostingID, PS.FltPlatformID
22             FROM (filterServers(nameParse, platform_id, isFavorite, userId) as S JOIN
23             public."Platform" as P on S.FltPlatformID = P."Id") as PS
24                 ORDER BY PS."Name" DESC;
25     ELSE -- Default: Name ASC
26         return query
27         SELECT *
28             FROM filterServers(nameParse, platform_id, isFavorite, userId) as S
29                 ORDER BY S.FltName;
30     END IF;
31
32 END;
33
34 $$ LANGUAGE plpgsql;
```