|  |  |
| --- | --- |
|  | **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

КАФЕДРА Информационная безопасность

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***Разработка защищенной БД информационной системы «Фитнес Центра»***

Студент \_\_\_ИУ8-81\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** В. А. Фильков

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Е. В. Глинская

(Подпись, дата) (И.О.Фамилия)

Москва, 2024 г.

**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)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ8

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_М.А.Басараб

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_февраля\_\_ 2024 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_Безопасность систем баз данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы: Разработка защищенной БД информационной системы «\_\_\_\_\_\_\_\_\_\_\_\_\_».

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_учебно-практическая\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_кафедра ИУ8 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения КР: 25% к \_3\_ нед., 50% к \_9\_ нед., 75% к \_12\_ нед., 100% к \_15\_ нед.

Техническое задание: Проанализировать существующую проблему информационной безопасности и защиты информации в базах данных информационной системы «\_\_\_\_\_\_\_\_». Представить решение проблемы информационной безопасности и защиты информации в базах данных информационной системы «\_\_\_\_\_\_\_\_\_\_».\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_30-35 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_Презентация\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_Флешка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_» \_февраля\_ 2024 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В.Глинская

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Оглавление

[Введение 3](#_Toc166447204)

[Основная часть 4](#_Toc166447205)

[Анализ предметной области 4](#_Toc166447206)

[Проектирование базы данных 7](#_Toc166447207)

[Подход к проектированию 7](#_Toc166447208)

[Построение инфологической модели предметной области 8](#_Toc166447209)

[Ограничение целостности 10](#_Toc166447210)

[Физическая реализация БД 11](#_Toc166447211)

[Запросы к БД 13](#_Toc166447212)

[Графический интерфейс 15](#_Toc166447213)

[Разработка средств защиты 19](#_Toc166447214)

[Архитектура 19](#_Toc166447215)

[Авторизация 20](#_Toc166447216)

[Резервное копирование 20](#_Toc166447217)

[Заключение 21](#_Toc166447218)

[Список использованных источников 22](#_Toc166447219)

# Введение

В настоящее время невозможно реализовать информационную систему, предназначенную для обработки большого объема данных, не используя готовых инфраструктурных и программных решений, существенно повышающих эффективность организации. К таким решениям относится электронная база данных.

База данных - совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными. Любая крупная организация использует электронные базы данных для хранения, агрегации и обработки большого количества информации. Такие базы предоставляют высокую скорость работы, возможность хранения различных типов данных, встроенные механизмы безопасности и аудита. В частности, безопасности баз данных уделяется большое значение: система должна обладать таким набором программных и аппаратных средств, чтобы обеспечивалась целостность, конфиденциальность и доступность информации.

Для создания базы данных и работы с ними существуют различные системы управления базами данных (СУБД) – комплекс программ, позволяющих создать базу данных и манипулировать данными. СУБД служит интерфейсом между базой данных и пользователями или программами, предоставляя пользователям возможность получать и обновлять информацию, а также управлять ее упорядочением и оптимизацией. СУБД обеспечивает контроль и управление данными, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка, а также резервное копирование и восстановление.

Целью данной работы является проектирование и разработка базы данных для построения безопасной и удобной системы взаимодействия клиента с фитнес центром.

# Основная часть

## **Анализ предметной области**

Для разработки защищенной базы данных информационной системы «Фитнес центр» необходимо понимать специфику бизнес-процессов внутри организации. Ниже представлены диаграммы, описывающие поведение системы.

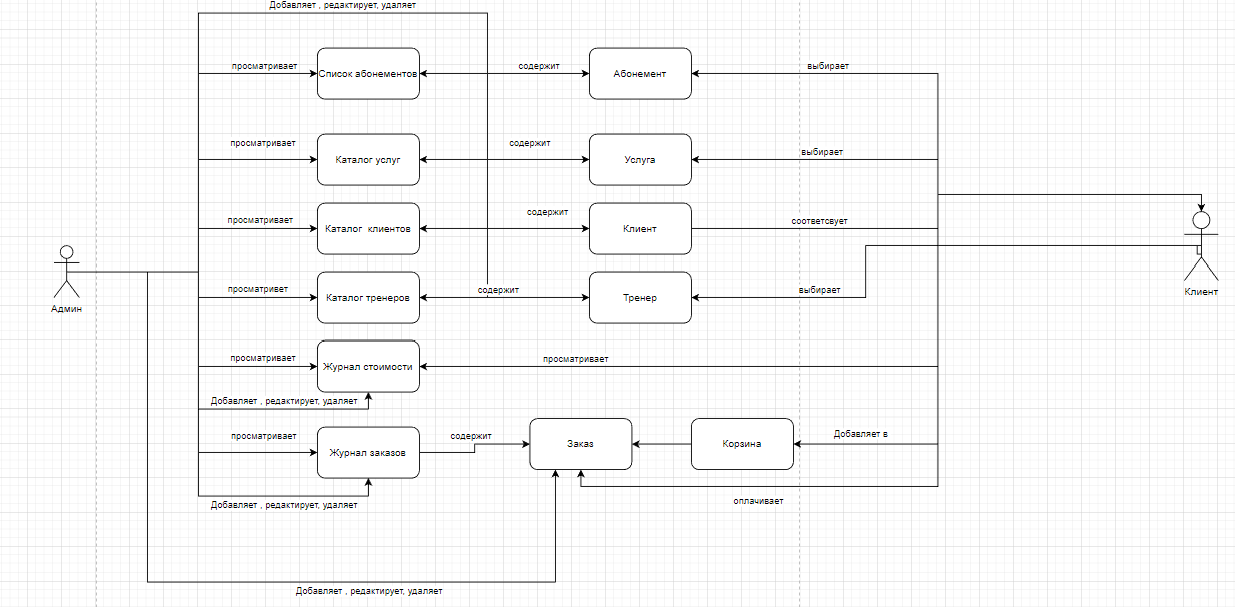


Рисунок 1 - Концептуальная модель

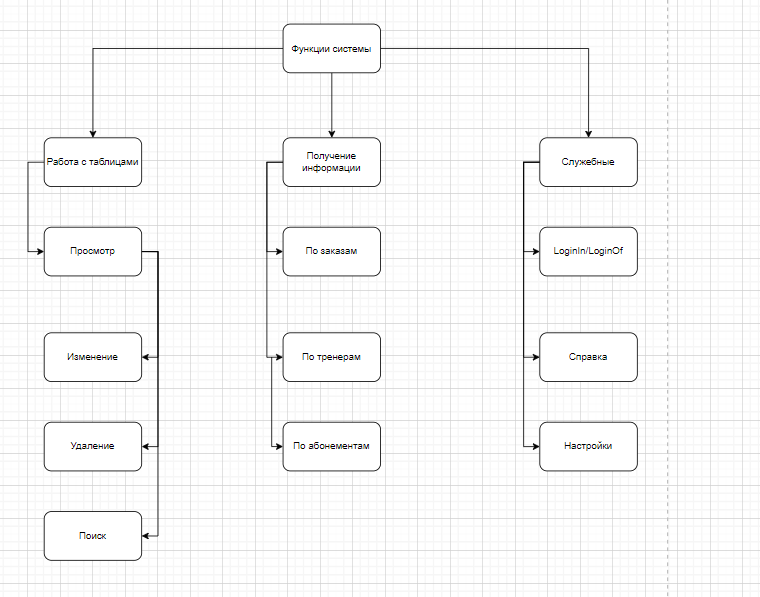


Рисунок 2 - Дерево функций

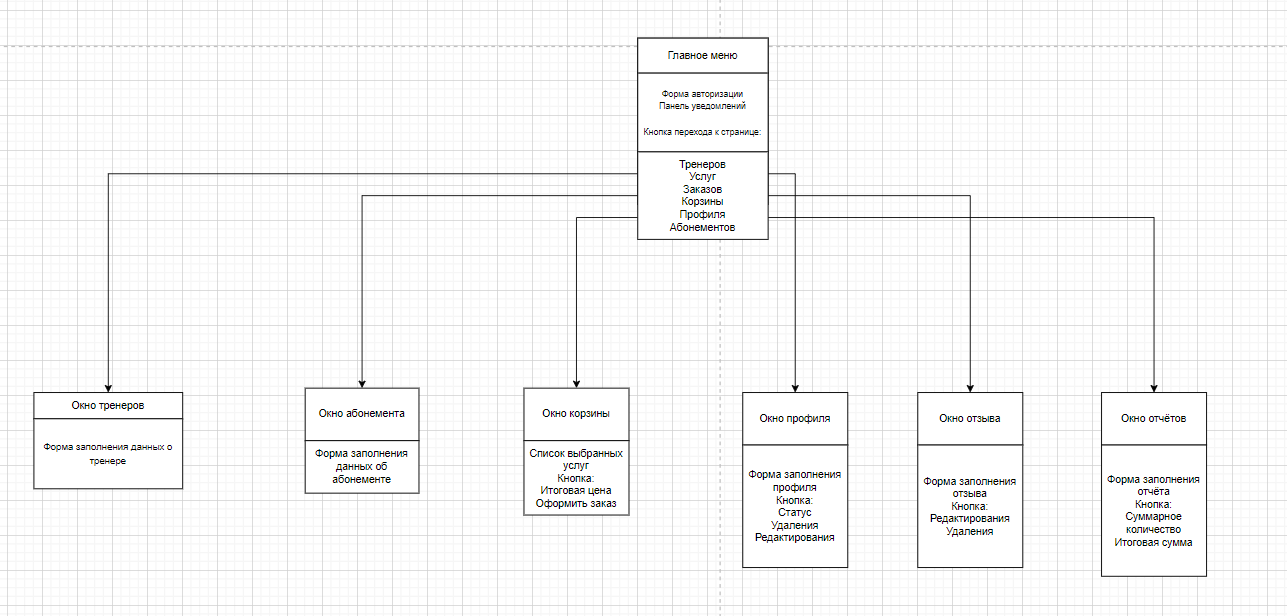
****

Рисунок 3 - Диаграмма классов

Таким образом, в системе существуют роли: администратор, клиент. Администратор имеет самые высокие привилегии и взаимодействует с БД во внутреннем контуре организации через СУБД. Клиент решают задачи через систему с графическим интерфейсом, не имея прямого доступа к БД.

Для реализации такой системы необходимо:

1. Спроектировать и реализовать БД для автоматизации учета клиентов, взаимодействующих с фитнесс залом, хранения необходимых персональных данных сотрудников и пользователей системы, агрегации информации о выполненных работах. В рамках реализации системы администратор сможет добавлять клиентов и тренеров в систему; клиент сможет заключать договор на оказание спортивных услуг. Таким образом, БД будет работать с сущностями: клиент, администратор.
2. Реализовать механизмы защиты данных для поддержания их целостности, доступности и конфиденциальности. Запросы со стороны клиента, проходят ряд проверок, необходимых для предотвращения SQL-инъекций и других атак, после этого сервер обращается к БД, расположенной во внутреннем контуре, и отсылает ответ клиенту. Также на клиентской стороне предусмотрена система авторизации.

Ссылка на источник [1].

# Проектирование базы данных

## **Подход к проектированию**

В качестве СУБД была взята POSTGRESQL, т.к. она имеет ряд преимуществ:

1. Встроенные механизмы обеспечения надежности и безопасности (механизмы резервного копирования, шифрования, аудита)
2. Возможность масштабирования: вертикально за счет увеличения серверных мощностей, горизонтально за счет кластеризации
3. Интеграция с серверными решениями Microsoft, например, Azure.

СУБД может использоваться на большом числе ОС от Microsoft Windows Server 2008, Windows 7 и более новых версиях, при необходимости можно запустить СУБД посредством Docker на любой ОС.

# Построение инфологической модели предметной области

На рисунке 4 показана схема БД и установлены зависимости.

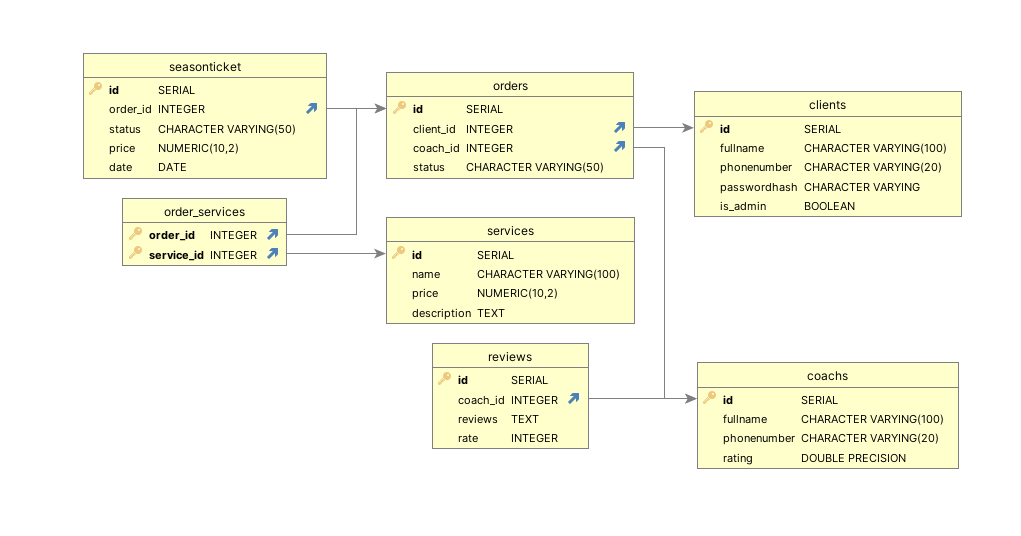


Рисунок 4 - Схема БД

В каждой таблице (за исключением вспомогательных таблиц, которые используются для отношения «many-to-many») имеется уникальное поле id, которое является первичным ключом.

В базе данных имеются следующие таблицы:

1. Клиенты (clients) – клиента фитнес зала
   1. fullname – ФИО клиента
   2. phonenumber – номер телефона
   3. hashpassword -хэш пароля
   4. is\_admin – админ ли наш пользователь
2. Тренеры (coachs) – тренеры фитнес зала

2.1) fullname – ФИО тренера

2.2) phonenumber – номер телефона

2.3) rating – рейтинг по 5 бальной шкале

3) Услуги (services)

3.1) name – название

3.2) price – стоимость

3.3) description - описание

4) Отзывы (review) – отзывы о тренерах

4.1) coach\_id – id тренера

4.2) reviews – комментарий

4.3) rate – рейтинг от 0 до 10

5) Абонемент (seasonticket) – абонементы

5.1) order\_id – id заказа

5.2) price – стоимость

5.3) date – дата оплаты

5.4) status – статус абонемента

6) Заказы (orders) - заказы

6.2) client\_id – id клиента, который оформил заказ

6.3) coach\_id – id тренера

6.4) status – статус заказа

В БД установлены следующие связи:

1. Таблица «Заказ» связана с таблицей «Клиент» связью «один-к-одному»
2. Таблица «Заказ» связана с таблицей «Тренера» связью «один-к-одному»
3. Таблица «Заказ» связана с таблицей «Услуги» (через вспомогательную таблицу) связью «много-ко-многим»
4. Таблица «Тренера» связана с таблицей «Отзывы» связью «один-к-одному»
5. Таблица «Абонемент» связана с таблицей «Заказ» связью «один-к-одному»

## **Ограничение целостности**

Целостность – состояние защищенности информации, характеризуемое способностью автоматизированной системы обеспечивать сохранность и неизменность информации при попытках несанкционированных воздействий на нее в процессе обработки или хранения.

Ограничение целостности – набор специальных методик, описывающих допустимые изменения значений отдельных информационных единиц и связей между ними. Нарушение ограничения целостности может происходить по многим причинам: человеческий фактор (непреднамеренные ошибки), действия злоумышленников, технические сбои и др.

Для информационной системы «Фитнес Центр» были введены следующие ограничения:

1. Ключевые значения каждой записи имеют уникальные, непустые значения
2. Каждый пользователь имеет уникальный login (номер телефона) для авторизации в системе

Дополнительно используются встроенные методы обеспечения целостности данных в MS SQL Server:

1. Механизм транзакций, гарантирующий, что изменения к БД применяются только в том случае, кода все запросы выполнены успешно
2. Ограничение целостности: ограничения уникальности (UNIQUE), ограничения внешнего ключа (FOREIGN KEY).
3. Журналирование событий, что позволяет восстановить данные в случае ошибки или сбоя

## **Физическая реализация БД**



Рисунок 5 - таблица клиентов



Рисунок 6 - информация о клиентах

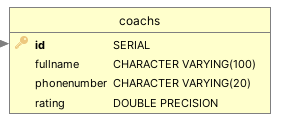


Рисунок 7 – таблица тренеров

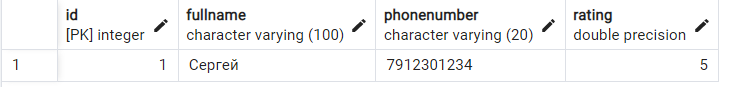


Рисунок 8 - информация о тренерах

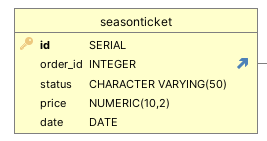


Рисунок 9 - таблица абонементов



Рисунок 10 - абонемент

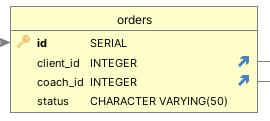


Рисунок 11 - таблица заказов



Рисунок 12 - отзывы о заказах

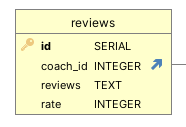


Рисунок 13 - таблица комментариев

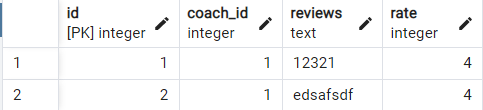


Рисунок 14 - комментарии

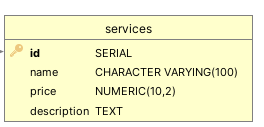


Рисунок 15 - таблица услуг



Рисунок 16 - информация о услуге

## **Запросы к БД**

Для работы с системой пользователю необходимо получить некоторые данные из БД, для этого на серверной части реализовано HTTPS API, к которому обращается клиент. Сервер взаимодействует с БД для получения информации [3].

1. Запрос к регистрации на клиента.

"INSERT INTO clients (fullname, phonenumber, passwordhash, is\_admin) VALUES ('"

+ fullname + "', '"

+ phone\_number + "', '"

+ hash\_password + "', "

+ "false" + ");";

Рисунок 17 – регистрация клиента

1. Запрос на добавление отзыва к тренеру.

"INSERT INTO reviews (coach\_id, reviews , rate) VALUES('"

+coach\_id+ "', '"

+review+ "', '"

+rate+ "');";

Рисунок 18 – добавление отзыва

3.

3. Найти id тренера по имени

"SELECT id FROM coachs WHERE fullname = '"+TrainerName+"';";

Рисунок 19 - получение заказов для клиента

4. Логин клиента по введённому паролю и номеру телефона.

"SELECT \* FROM clients WHERE phonenumber = '" + phone\_number +

"' AND passwordhash = '" + hash\_password + "';";

Рисунок 20 - получение заказов для клиента

5. Вставка в заказы.

INSERT INTO orders (client\_id , coach\_id , status)

Рисунок 21 – вставка в заказы

VALUES ('client\_id','coach\_id','status');

6. Вставка в комментарии для тренера

Рисунок 22 – вставка в комментарии

INSERT INTO reviews (coach\_id, reviews, rate) VALUES('coach\_id' ,'reviews','rate')

SELECT \* FROM agency.dbo.realtors WHERE id = CAST(\'{id}\'

7. Вставка в абонементы

INSERT INTO seasonticket (order\_id ,status, price, date)

Рисунок 23 – вставка в абонементы

VALUES ('1','Индивидуальный', '2500', '20

1. Также нам может понадобиться в логике центра получать все нужные данные из этих таблиц.

Select \* from seasonticket

Рисунок 25 – получение данных из таблицы услуг

Рисунок 24 – получение данных из таблицы абонементов

SELECT \* from services

SELECT \* from reviews

Рисунок 26 - получение данных из таблицы

## **Графический интерфейс**

Для унификации взаимодействия с БД различных пользователей с различными привилегиями был разработан графический интерфейс, предоставляющий возможность работы с информационной системой «Фитнес центра».

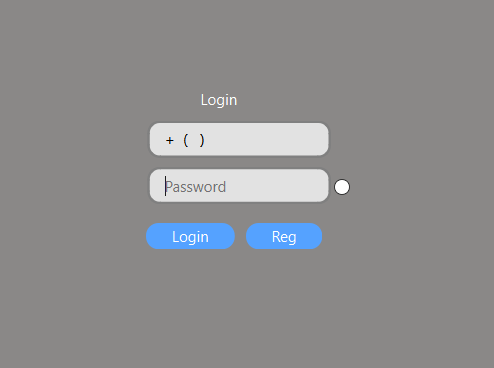


Рисунок 27 - форма авторизации



Рисунок 28 – форма регистрации

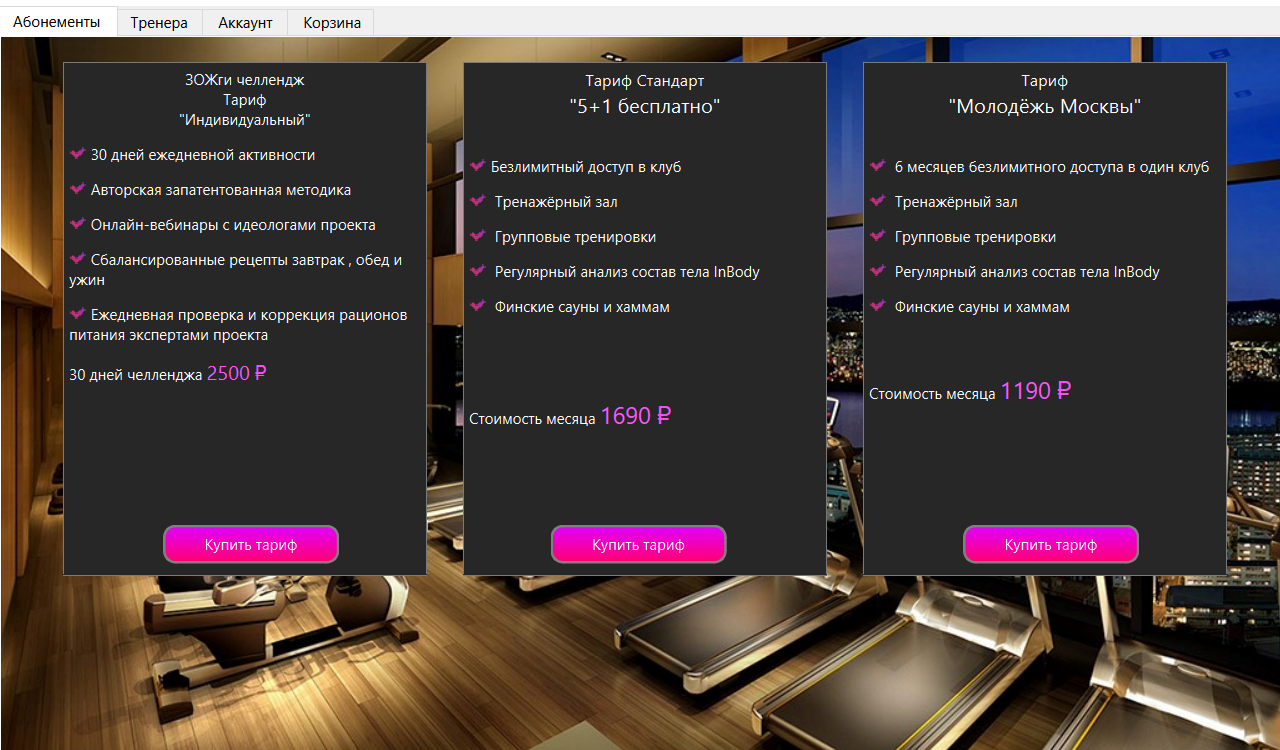


Рисунок 29 – форма Абонементов



Рисунок *30* *–* форма *тренеров*

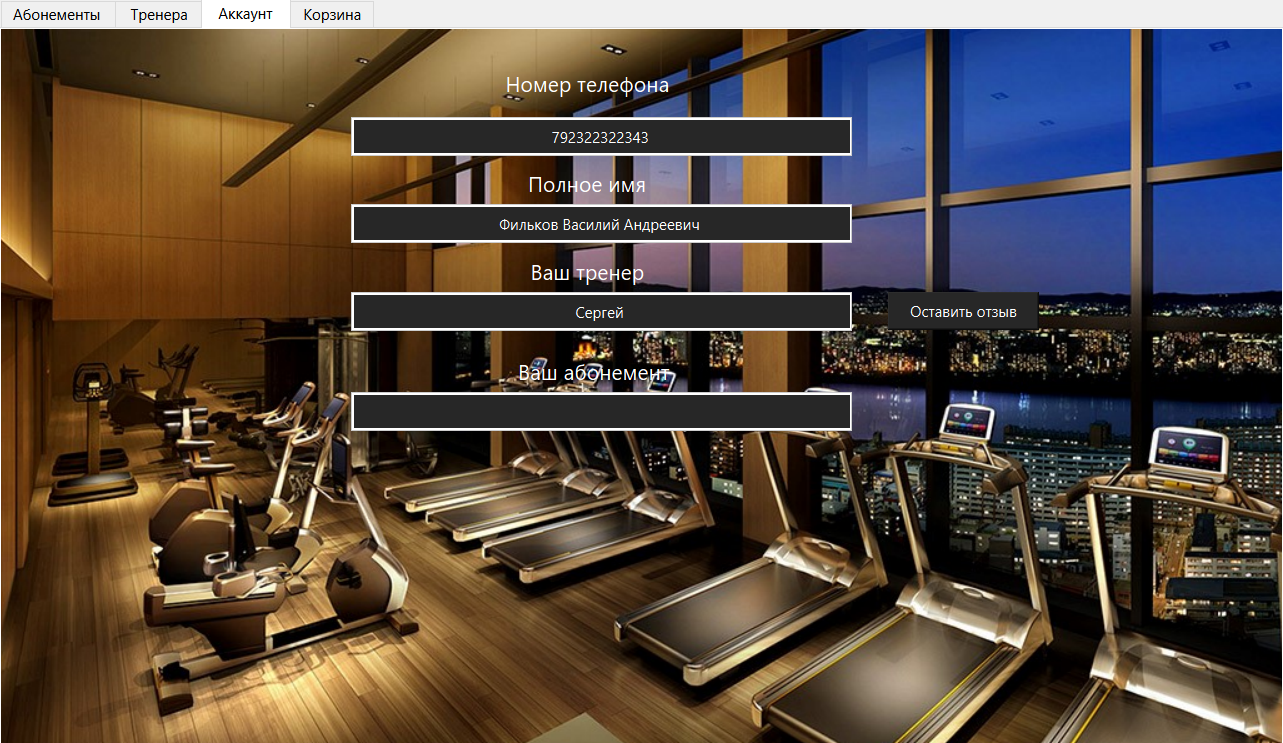


Рисунок *31* *–* форма *аккаунта*

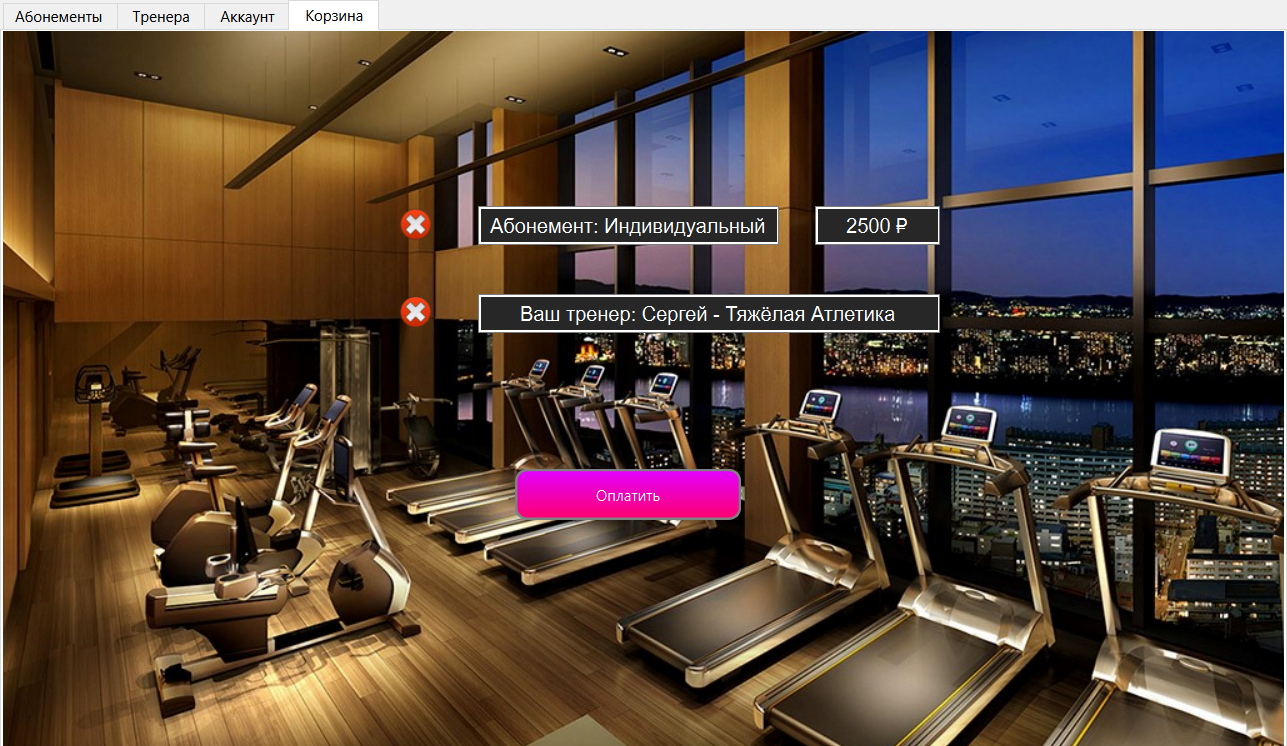


Рисунок *32* *–* форма *корзины*

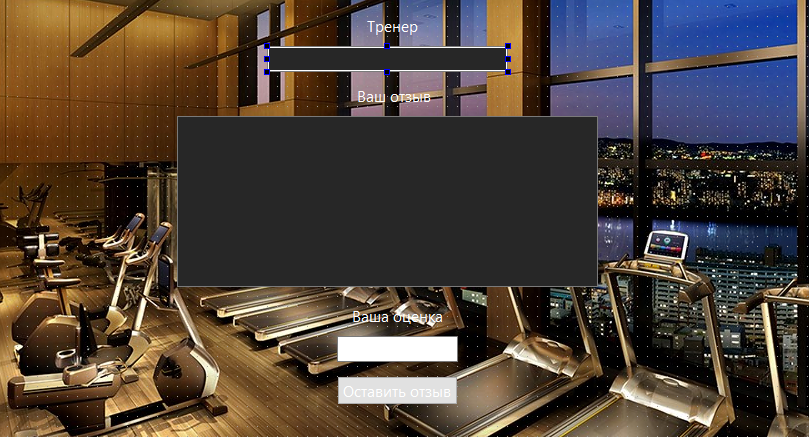


Рисунок *33* *–* форма *комментариев для тренера*

# Разработка средств защиты

Для обеспечения защиты информационной системы «Фитнес цетр» были реализованы следующие методы :

1. Разделение архитектуры на клиентскую и административную части.
2. Авторизация по логину и паролю
3. Разграничение прав пользователей
4. Интеграция со сторонними технологиями защиты БД - pgadmin
5. Резервное копирование

## **Архитектура**

На рисунке представлено верхнеуровневое представление архитектуры взаимодействия клиента и сервера. Клиент не имеет доступа к БД. Перед обращением к БД на клиентской стороне проверяет запрос на:

1. Наличие различных типов SQL инъекций (Blind, Error-based, Union-based и др.)
2. Наличие у пользователя необходимых прав для выполнения запроса

Если все проверки пройдены успешно, то клиент обращается к БД, расположенной в том же контуре, для получения необходимой информации. Если какая-то из проверок провалилась, то программа не даст выполнить этот код.

Такая архитектура легко масштабируется и управляется. Можно дополнительно настроить защиты SQL инъекций, межсайтовый скриптинг (cross-site scripting XSS) и др, тем самым значительно повысить безопасность структуры.

Для масштабирования системы достаточно указать ещё одну БД, с которой необходимо работать. Дополнительно можно установить балансировщик нагрузки или прокси, который в автоматическом режиме будет перенаправлять запросы на один из серверов, что не только увеличит скорость обработки запросов, но и сделает систему более стойкой к DDoS атакам.

Выше представлены некоторые (но далеко не все) способы повышения безопасности системы.

## **Авторизация**

Перед началом работы с информационной системой пользователь должен авторизоваться по логину и паролю. После ввода данных клиент устанавливает соединение с базой данных отправляет логин и хеш пароля. В качестве алгоритма хеширования используется SHA256. База данных проверяет полученные данные и в случае успешной авторизации возвращает должность пользователя. Если не удалось установить соединение с сервером или введенные данные оказались неверными, то пользователь не сможет авторизоваться. В программе также может быть настроена политика блокировки пользователей после определенного числа неудачных попыток авторизации. Так же из данных можно ввести только в логине можно ввести только цифры из номера телефона так, как везде в графах для пользователя настроены input\_mask. Вовремя введения пароля пароль скрыт точками.

  
Рисунок *34* *– форма для логина*

## **Резервное копирование**

Для создания резервных копий используется встроенный механизм СУБД POSTGRESQL.

# Заключение

Информационная система «Фитнес Центр» работает с большим объемом данных, которые необходимо хранить, обрабатывать и обновлять. Поэтому для эффективной работы организации была выстроена система на основе БД POSTGRESQL, позволяющая эффективно и безопасно работать клиентам и сотрудникам.

На первом этапе была спроектирована модель базы данных на основе анализа предметной области, разработан подход к проектированию БД и выбраны средства и методики разработки

На втором этапе была создана и заполнена БД, разработаны запросы и отчеты, реализован графический интерфейс для пользователей.

На третьем этапе описаны средства защиты системы, предложены дальнейшие способы развития и улучшения модели, обоснована эффективность выбранных подходов.

Таким образом, полученная информационная система «Фитнес Центр» соответствует всем поставленным требованиям безопасности, отказоустойчивости, скорости и удобства.

# Список использованных источников

1. Базы данных / под ред. проф. А.Д. Хомоненко. – М.: КОРОНА-Век, 2011.
2. Туманов В.Е. Основы проектирования реляционных баз данных / В.Е. Туманов. – М.: Бином-Пресс, 2012 – 420 c.
3. Цирлов В.Л. Основы информационной безопасности автоматизированных систем – М.: Феникс, 2006 256 с.
4. Бондарь А. Г. Microsoft SQL Server 2022 – М.: BHV-СПб, 2024 – 528 с.