МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»**

Институт ИТКН

**Кафедра инженерной кибернетики**

Направление подготовки: 01.03.04 прикладная математика

Квалификация (степень): бакалавр

Группа: **БПМ-17-1**

**КУРСОВАЯ РАБОТА**

**на тему:**

Программная система для организации работы спортивных бассейнов.

Учебная дисциплина «Специальные главы баз данных»

VIII семестр 2020 – 2021 у.г.

**Учащийся: Голованов Н.С.**

**Проверил: доц., к.т.н. А.С. Кожаринов**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва 2021**

Содержание

[*Список используемых основных сокращений 4*](#_Toc71470914)

[*Введение 5*](#_Toc71470915)

[*Постановка задачи 6*](#_Toc71470916)

[*Использованные средства разработки и системные требования 8*](#_Toc71470917)

[*Модель данных проекта 9*](#_Toc71470918)

[*Организация ролевого доступа к данным 15*](#_Toc71470919)

[*Тематические запросы к данным 16*](#_Toc71470920)

[*Программное обеспечение для взаимодействия с базой данных 17*](#_Toc71470921)

[*1. Форма авторизации 17*](#_Toc71470922)

[*2. Интерфейс для роли клиента 18*](#_Toc71470923)

[*2.1. Форма покупки билетов. 18*](#_Toc71470924)

[*2.2. Форма тренировок 19*](#_Toc71470925)

[*2.3. Форма персональной страницы 20*](#_Toc71470926)

[*3. Интерфейс для роли администратора 21*](#_Toc71470927)

[*3.1. Форма для работы с клиентами. 21*](#_Toc71470928)

[*3.2. Форма для работы с платежными транзакциями. 23*](#_Toc71470929)

[*3.3. Форма для просмотра и формирования отчета по дневной выручки 23*](#_Toc71470930)

[*3.4. Форма для работы с загруженностью бассейнов 24*](#_Toc71470931)

[*Выводы 25*](#_Toc71470932)

# **Список используемых основных сокращений**

СУБД — система управления реляционными базами данных

GUI — Графический интерфейс пользователя

Фреймворк — программная платформа, определяющая структуру программной системы

API — программный интерфейс приложения

# **Введение**

Развитие информационной системы для любого предприятия в сегодняшние дни необходимо. Информационная система позволяет эффективнее обеспечивать работу администрирования и организации взаимодействия с клиентами. Одним из активно развивающих сфер услуг в наше время является спортивная сфера. Для примера была выбрана условная организация по предоставления услуг свободного плавания в бассейнах. В данной курсовой работе необходимо разработать программную систему для организации работы спортивных бассейнов.

Основой создания программной системы, рассчитанной на массовое использования и хранения большого объема информации, является выбор базы данных. На сегодняшний день существует возможность использовать мультимодельные решения для хранения данных. Такие решения позволяют разнести логику работы программной системы в зависимости от задач и требований. В данной курсовой работе будет использована общая модель данных проекта, состоящая из сочетания основной объектно-реляционной модели и вспомогательной документно-ориентированной модели. Основным предназначением программного комплекса будет предоставление возможностей администрирования спортивной организации, а также предоставления удобного интерфейса взаимодействия с клиентами для отслеживания результатов тренировок и покупки билетов. Также в работе реализован ролевой доступ к хранимым данным, что позволяет разделить пользователей на клиентов и администратора с различными правами доступа.

# **Постановка задачи**

Спортивные и массовые бассейны очень востребованы в России. Об этом свидетельствуют и результаты последних маркетинговых исследований, и опыт отечественных предпринимателей, открывших соответствующие направление в данной индустрии.

Поэтому программы автоматизирующий взаимодействие клиентов с организацией, предоставляющие услуги, а также администраторов для налаживания и мониторинга работы бассейнов имеют высокий спрос.

Основными пользователями будут – клиенты бассейнов и администратор организации.

Администратор – это сотрудник организации, который оказывает услуги:

* ведет отслеживания загруженности бассейных комплексов
* организовывает доступ для клиентов
* анализирует ежедневые финансовые отчеты
* проверяет финансовые операции

Клиент спортивного бассейна – это человек, который пользуется услугами:

• возможность просмотреть свои билеты, а также купить билет на сеанс;

• взаимодействовать со своим профилем;

• отслеживать свои автоматические записанные тренировки, а также вносить в них коррективы;

В данной курсовой работе для реализации программного комплекса необходимо составить базу данных, которая будет отвечать за хранения и сбор информации о посещения бассейнов, информации о пользователях и их тренировках, а также о билетах и финансовых отчетах. Для этой задачи подходящим решением будет использовать объектно-реляционную модель.

С другой стороны, для выполнения авторизации пользователей необходимо обособленное хранение информации, необходимой для авторизации с целью упрощения ее дальнейшей защиты от несанкционированного доступа. Также сведения о платежных транзакциях вынесены также в обособленную базу данных, что имитирует работу банка. Для перечисленных выше целей было решено использовать документно-ориентированной модель.

Необходимо предусмотреть интерфейс для авторизации и регистрации пользователя и учесть права доступа к базе данных у этого пользователя.

# **Использованные средства разработки и системные требования**

Разработанное программное обеспечение является десктопной версией, работающее на операционной системе не ниже Windows 7. Для написания программы был выбран язык Python с использованием редактора кода Visual Studio Code. Графический интерфейс пользователя был реализован с помощью графического фреймворка Qt с набор расширений PyQt5 для языка программирования Python.

В качестве хранилища данных в объектно-реляционной модельи задействована свободная объектно-реляционная система управления базами данных PostgreSQL, с использованием программы pgAdmin для ее администрирования. Для проектирования баз данных использовалась бесплатная версия онлайн платформы dbdesigner.net. Для хранения баз данных, построенных на документно-ориентированной модели использован популярный текстовый формат данных JSON.

Для взаимодействия программы с объектно-реляционной моделью базы данных использовалась SQLAlchemy - программная библиотека с открытым исходным кодом на языке Python. Для работы с JSON-файлами, которые хранят базы данных документно-ориентированной модели, иcпользована внутренняя библиотека для упрощения взаимодействия с данными.

# **Модель данных проекта**

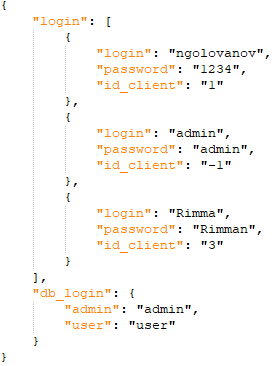
Для организации надежного долговременного хранения информации в создаваемой системе была разработана общая модель данных проекта, которая включает в себя 2 класса моделей данных. Для проектирования баз данных использовалась онлайн платформа dbdesigner.net. Она позволяет эффективно спроектировать базу данных с созданием всех связей. На рисунке 1 продемонстрирована ER-диаграмма основной (отмечены красными заголовками) и вспомогательной (отмечены голубыми заголовками) модели базы данных организации спортивных бассейнов.

Изображение выглядит как текст, внутренний

Автоматически созданное описание

Рисунок 1. ER-диаграмма базы данных.

На рисунке 2 показана структуры документно-ориентированной модели, одна из которых предназначена для хранения логинов, паролей, а также идентификационных номеров клиентов, а другая – для хранения информации о платежных транзакциях. Данная модель представляет из себя JSON строку, которая хранится в отдельном поле.

 Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. JSON формат для хранения информации для авторизации (слева) и платежных транзакций (справа).

Таблица Clients хранит персональную информацию о клиентах, а также их уникальный идентификационный номер (далее УИН).

Таблицы Swimming\_pools, Pools\_locations, Cities, Type\_of\_water и Monetary\_policies хранит достаточную информацию об бассейном комплексе, а именно его место расположение, тип воды, количество дорожек, а также финансовую политику.

Таблицы Tickets, Tickets\_types и Swimming\_sessions хранят информацию о проданных билетов, а также об загруженности бассейнов.

Таблица Daily\_earnings хранит информацию о дневной выручке и о финансовой политике.

Документы в таблице Authorization хранящаяся в формате типа JSON содержит информацию о логинах, паролях, а также уникальный идентификационный номер пользователя.

Документы в таблице Payment\_transactions хранящаяся в формате типа JSON содержит информацию о платежных операциях.

Ниже представлены таблицы с полями и их назначениями.

Таблица “Clients”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | Уникальный идентификационный номер клиента |
| 2 | name | varchar |  | Имя клиента |
| 3 | surname | varchar |  | Фамилия клиента |
| 4 | patronymic\_name | varchar |  | Отчество клиента |
| 5 | birth | date |  | Дата рождения клиента |
| 6 | date\_registration | date |  | Дата регистрация клиента |
| 7 | phone | varchar |  | Номер телефона |

Таблица “Tickets”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН билета |
| 2 | id\_client | varchar | FK | УИН клиента |
| 3 | id\_training | varchar | FK | УИН тренировки |
| 4 | id\_pool | varchar | FK | УИН бассейна |
| 5 | type | date | FK | Тип билета |
| 6 | date\_and\_time | Datetime |  | Дата сеанса |

Таблица “Ticket\_types”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | type | int | PK | УИН типа билета |
| 2 | description | varchar |  | Описание типа билета |

Таблица “Training\_sessions”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | Уникальный идентификационный номер тренировки |
| 2 | duration | int |  | Длительности тренировки |
| 3 | swam\_distance | int |  | Суммарная проплытая дистанция |
| 4 | heart\_rate | int |  | Пульс |
| 5 | spO2 | int |  | Уровень кислорода в крови |
| 6 | crawl | int |  | Проплытая дистанция кролем |
| 7 | backstroke | int |  | Проплытая дистанция на спине |
| 8 | breaststroke | int |  | Проплытая дистанция брассом |
| 9 | butterfly | int |  | Проплытая дистанция баттерфляем |
| 10 | type\_of\_water | int | FK | Тип воды |
| 11 | swolf | int |  | SWOLF плавца |
| 12 | kilocalories | int |  | Килокалории |

Таблица “Swimming\_pools”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН бассейна |
| 2 | lines | int |  | Количество дорожек |
| 3 | location | int | FK | Расположение |
| 4 | type\_of\_water | int | FK | Тип воды |
| 5 | capacity | int |  | Общая вместимость |
| 6 | monetary\_policy | int | FK | Финансовая политика |

Таблица “Pools\_locations”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН адреса |
| 2 | address | charvar |  | Адрес |
| 3 | city | int | FK | Город |

Таблица “Cities”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН города |
| 2 | city | charvar |  | Город |

Таблица “Ticket\_types”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | type | int | PK | УИН типа бассейна |
| 2 | description | varchar |  | Описание типа бассейна |
| 3 | capacity\_per\_line | int |  | Вместимость одной дорожки |

Таблица “Monetary\_policies”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН финансовой политики |
| 2 | ticket\_price | int |  | Полная цена билета |
| 3 | preferential\_discount | double |  | Льготная скидка (от 0 до 1) |
| 4 | date\_of\_adoption | date |  | Дата принятия |

Таблица “Swimming\_sessions”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | id | int | PK | УИН сеанса |
| 2 | id\_pool | int | FK | УИН бассейна |
| 3 | entry\_tickets | int |  | Общее количество проданных билетов |
| 4 | discount\_tickets | int |  | Количество проданных льготных билетов (из всего числа) |
| 5 | workload | double |  | Загруженность |
| 6 | date\_and\_time | datetime |  | Дата и время сеанса |

Таблица “Daily\_earnings”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | report\_id | uuid | PK | УИН(UUID) отчета для определенного бассейна за день |
| 2 | id\_pool | int | FK | УИН бассейна |
| 3 | clients | int |  | Общее количество проданных билетов за день |
| 4 | discount\_clients | int |  | Количество проданных льготных билетов (из всего числа) за день |
| 5 | proceeds | double |  | Выручка за день |
| 6 | monetary\_policy | int | FK | Финансовая политика |
| 7 | date\_ | date |  | Дата отчета |
| 8 | workload | double |  | Загруженность за день |

Таблица “Authorization”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | login-login | String |  | Логин учетной записи клиента |
| 2 | login-password | String |  | Пароль учетной записи клиента |
| 3 | login-id\_client | String |  | УИН клиента |
| 4 | db\_login-admin | String |  | Пароль для администратора |
| 5 | db\_login-user | String |  | Пароль для клиента |

Таблица “Payment\_transactions”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название поля | Тип данных | FK/PK | Назначение |
| 1 | Transactions-uuid | String |  | UUID транзакции |
| 2 | Transactions- ticket\_info- id\_client | String |  | УИН клиента |
| 3 | Transactions- ticket\_info- id\_pool | String |  | УИН бассейна |
| 4 | Transactions- ticket\_info- date | String |  | Дата сеанса |
| 5 | Transactions- ticket\_info-time | String |  | Время сеанса |
| 6 | Transactions- ticket\_info- id\_ticket | String |  | УИН билета |
| 7 | Transactions- payment \_info- card | String |  | Заглушка для карты |
| 8 | Transactions- payment \_info- time\_payment | String |  | Время оплаты |
| 9 | Transactions- payment\_info- time\_append\_ticket\_from\_db | String |  | Время подтверждения от базы данных |

# Организация ролевого доступа к данным

В текущем проекте существует 2 различные роли для доступа к базе данных.

1. Admin - данная роль предназначена для администрации спортивного бассейна с возможностью просматривать, изменять, удалять, обновлять данные.

CREATE ROLE Admin GO

GRANT SELECT, UPDATE, DELETE, INSERT ON DATABASE:: swimming\_pools\_db TO Admin

2. User - роль, которая назначается пользователям-клиентам, с возможностью просматривать данные и обновлять собственный профиль, а также покупать билеты.

CREATE ROLE User GO

GRANT SELECT ON DATABASE::swimming\_pools\_db TO User

GRANT INSERT, UPDATE, DELETE on public.Clients TO User

GRANT INSERT, UPDATE, DELETE on public.Training\_sessions TO User

GRANT INSERT on public.Tickets TO User

Также присутствует логика и бизнес-правила в приложение для работы с базами данных, которые также разграничивают права.

# **Тематические запросы к данным**

1. Почти все запросы для клиента отфильтрованы по ему собственному УИН, чтобы пользователь не мог получить информацию о других пользователях.
2. Запрос на вставку данных в таблицу Tickets, что, по сути, инициирует покупку билета, автоматически с помощью триггера insert\_ticket (см. Приложение) добавляет в нужную запись в таблице Swimming\_sessions +1 к общему числу проданных билетов, а также в зависимости от типа билета +1 к числу льготных билетов. А также иницилизирует запуск функции upd\_workload (см. Приложение) которая обновляет в автоматическом режиме загруженность сеанса.
3. При удалении тренировки необходимо выполнить каскадное удаление в таблице Tickets. Для этого реализована функция delete\_training\_session (см. Приложение)
4. При удалении пользователя также необходимо выполнить каскадное удаление всех записей о тренировках пользователя, а также выставления в поле id\_client в таблице Tickets значения -1, что означает неопределенного пользователя. И после выполнения всех этих процедур, удаляется сама запись клиента в таблице Clients. За это отвечает триггер delete\_client (см. Приложение)
5. Для вычисления ежедневной выручки также используется специальная функция culc\_daily\_earnings (см. Приложение). Она проходит по всем записям в таблице Swimming\_sessions и забирает все данные о конкретном бассейне в конкретный день.

# **Программное обеспечение для взаимодействия с базой данных**

## Форма авторизации

Разработанное программное обеспечение позволяет взаимодействовать клиентам и администратору с базой данных. Интерфейс программы обладает меню аутентификации (рисунок 3). Главное меню приложения формируется в процессе входа и отображает элементы в соответствии с ролью пользователя (рисунок 4). Каждый элемент открывает новую форму для изменения или отображения различной информации.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3. Окно авторизации.

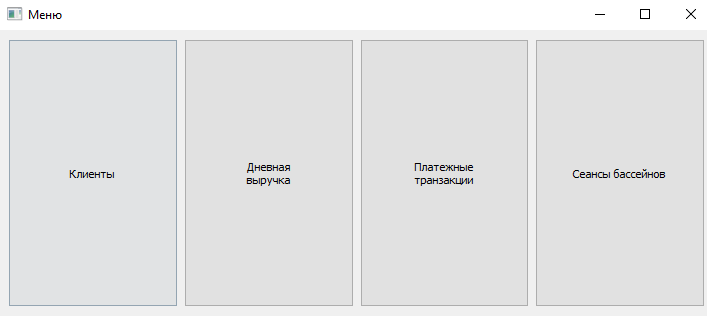
 

Рисунок 4. Меню после авторизации. Слева для роли администратора, справа для клиента

Во всем проекте активно используется библиотека SQLAlchemy (пример кода авторизации на рисунке 5). Так для соединения необходим логин и пароль роли базы данных. Для верного соотнесения с ролями и используется отдельные записи в базе authentication. Если поле client\_id = -1, значит это администратор, в остальных случаях – это пользователь.

Также для взаимодействия необходимы модели таблиц (см. Приложения).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Пример кода авторизации

## Интерфейс для роли клиента

В этом разделе будет продемонстрирован интерфейс и логика работы для роли клиента.

### Форма покупки билетов.

В данном окне (рисунок 6) клиент может увидеть информацию по уже приобретённым билетам (в левой части экранной формы), а также купить новый, если есть сеансы. Сначала пользователь должен выбрать бассейн. После этого на календаре появятся доступные даты для покупки. После выбора пользователем даты, в поле сеансов появляется список доступных сеансов для покупки. Он формируется из числа тех сеансов, загруженность которых меньше 1. Также можно выбрать тариф. После выбора всех полей станет активна кнопка Купить. После покупки – список слева обновиться. Если, например нет доступных сеансов, то программа уведомить об этом пользователя предупреждением (рисунок 7), а также сделает недоступным для выбора календарь.

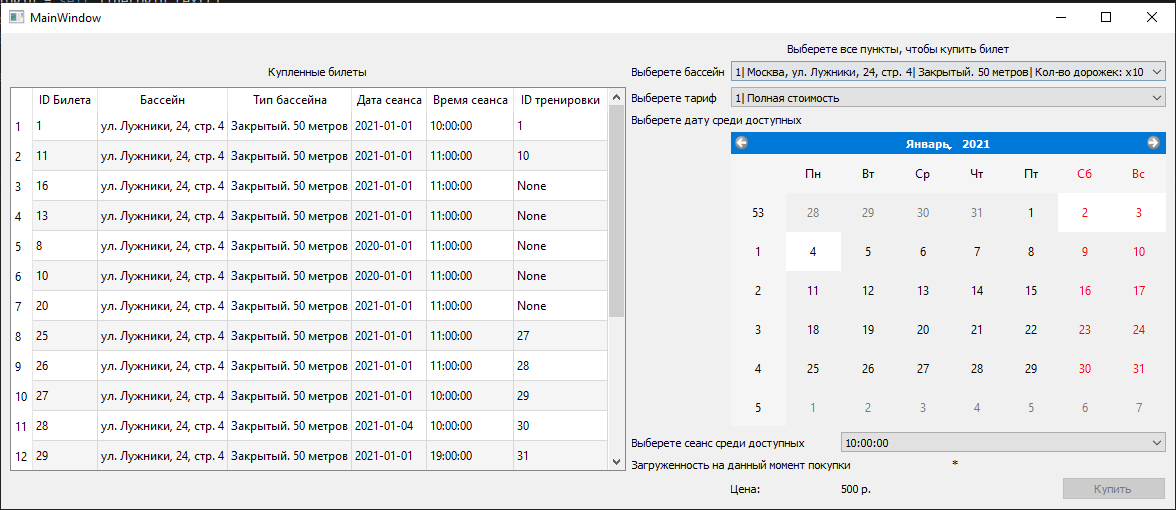


Рисунок 6. Форма покупки билетов.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7. Предупреждение об отсутствии дат

### Форма тренировок

В данном окне (рисунок 8) содержится вся информация о тренировках. Тренировка формируется автоматически при покупке билетов, так как предполагается, что выдается фитнес-браслет для клиентов, который автоматически заносит информацию о тренировке. Но пользователь в праве редактировать, удалять и создавать записи тренировок. Для удаления или редактирования необходимо выбрать любую из доступных записей из таблицы. Тогда она автоматически погрузится в нижнюю таблицу (рисунок 9). Можно изменить все поля, кроме id тренировки, так как оно формируется единожды и автоматически.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8. Форма просмотра тренировок.

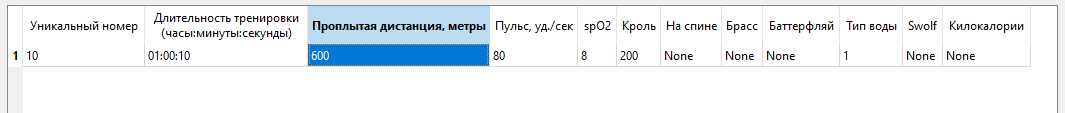


Рисунок 9. Автозаполнение формы для редактирования

### Форма персональной страницы

Данная форма (рисунок 10) содержит основную информацию о клиенте, а также предоставляет возможность скорректировать свой номер телефона и удалить свой аккаунта. Если пользователь неправильно внес свой номер телефона, выводится ошибка (рисунок 11 слева). Также при удалении еще раз уточняется у пользователя, точно ли он хочет это сделать (рисунок 11 справа). При удалении будет выполнено каскадное удаление как в основной части базы данных, так и во вспомогательной части (в базе данных об информации для авторизации).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10. Форма персональной карточки

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11. Ошибки и предупреждения пользователя.

## Интерфейс для роли администратора

В этом разделе будет продемонстрирован интерфейс и логика работы для роли администратора.

* 1. Форма для работы с клиентами.

Данная форма (рисунок 12) позволяет администратору удалять старых и создавать новых пользователей. Чтобы удалить необходимо в верхней таблице выбрать нужного пользователя и нажать кнопку удалить. Также при удалении у администратора будет уточнено, действительно ли необходимо его удалить. При нажатии кнопки Создать автоматически формируется форма в нижней таблице, и пользователь ее редактирует. После завершения редактирования нажимается еще раз кнопка Создать и пользователю предлагается придумать логин и пароль (рисунок 13). Они временные и в будущем будет создана форма и логика для его изменения.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 12. Форма работы с клиентами.

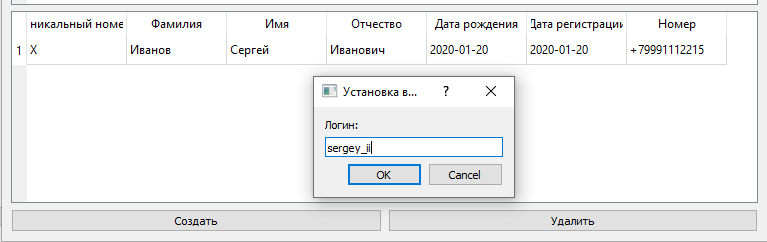


Рисунок 13. Форма для ввода логина.

### Форма для работы с платежными транзакциями.

В данной форме (рисунок 14) представлена работа с транзакциями. Здесь происходит взаимодействие только с вспомогательной частью базы данных, а именно с payment\_transactions. Присутствует возможность обновить данные.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 14. Окно для платежных транзакций.

### Форма для просмотра и формирования отчета по дневной выручки

Данная форма (рисунок 15) предназначена для работы с отчетами ежедневной выручки бассейнов. Администратор может просматривать как существующие отчеты в таблице, так и формировать (или обновлять старые) отчеты. Данный инструмент представлен в виде прототипа, так как в идеальном случае, должен отрабатывать триггер каждый день, который будет формировать такой отчет в автоматическом режиме.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15. Форма для отчетов дневной выручки

### Форма для работы с загруженностью бассейнов

Данная форма (рисунок 16) позволяет удобно просматривать администратору загруженность бассейнов, а также дополнительную интересующую его информацию. Сделана группировка по значениям. Так на рисунке 16 сформирована группировка по загруженности.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 16. Форма для работы с загруженностью бассейнов.

# **Выводы**

При выполнении курсовой работы, поставлена цель – разработка клиентского приложения для организации работы спортивных бассейнов. Обоснована актуальность и необходимость выбранной темы. Выбраны средства разработки, построены модели данных, схемы базы данных. Разработан интерфейс и функционал клиент-серверного приложения. Цель работы достигнута. В полной мере решены поставленные задачи:

* выполнен анализ предметной области;
* спроектирована модель данных;
* сделан выбор программных средств и инструментов для разработки;
* произведена разработка базы данных;
* спроектирован интерфейс;
* разработан функционал клиент-серверного приложения.

Программное приложение предназначено для использования непосредственно для организаций, осуществляющие предоставление услуг в сфере спортивного плавания. Таким образом, созданное приложение позволяет наиболее достоверно, быстро и безошибочно собирать и производить различные операции с данными. А значит, позволит быстрее и качественнее осуществлять взаимодействие клиентов с организацией. Также удобные инструменты для администрирования помогут качественно и эффективно отслеживать положение дел и вносить коррективы в случае необходимости, не отвлекаясь на перепроверку данных.

# **Приложения**