



Министерство образования и науки 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 Информатики и систем управления

Кафедра Теоретической информатики и компьютерных технологий

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ
НА ТЕМУ:

**База данных для учета индивидуальных достижений
клиента тренажерного зала**

Студент _____

(Группа)

(Подпись)

(И.О.Фамилия)

Руководитель курсового проекта

(Подпись)

(И.О.Фамилия)

Москва 2018

Оглавление

Введение	3
Глава 1. Обзор предметной области.....	5
1.1 Обзор предметной области	5
1.2 Проектирование базы данных и составление ER - модели	5
Глава 2. Разработка реляционной базы данных.....	7
Глава 3. Разработка приложения	11
3.1 Используемые технологии	11
3.2 Регистрация и авторизация	11
3.3 Тренер.....	13
3.4 Клиент	15
Глава 4. Тестирование программной реализации.....	18
Заключение.....	20
Список литературы.....	21
ПРИЛОЖЕНИЕ А. Описание таблиц	22

Введение

Увеличение и изменение характера нагрузок на человека в связи с усложнением жизни общества приводит к тяжелым последствиям для организма. Образ жизни непосредственно влияет на состояние и здоровье человека. В связи с этим у общества возникает необходимость в соблюдении правильного режима дня, здорового питания, в занятиях спортом. В связи с загруженностью на работе, человеку трудно выделить свободное время для личных дел. Более трудной задачей является поиск времени для организации тренировок. Фитнес-клуб дает возможность эффективно использовать свое свободное время и получить помощь в занятиях спортом, так как упражнения выполняются по программе, составленной квалифицированными тренерами.

Здоровье и здоровый образ жизни в настоящее время актуален. Сегодня фитнес-индустрия в России активно развивается. Вложения в здоровье и красоту являются на сегодняшний день серьезными и окупаемыми.

В связи с улучшением качества обслуживания клиентов, разнообразием сфер услуг в фитнес-центрах и увеличением их количества растет конкуренция между ними. Ни один фитнес-центр в настоящее время не может обойтись без информационной поддержки, осуществляемой посредством соответствующих программных продуктов, чтобы быть конкурентоспособным в сфере данных услуг. Чаще всего для этого используются базы данных. В базах данных ведется учет работы всего фитнес-центра. База данных фитнес-центра позволяет просматривать перечень клиентов, инструкторов, предоставляемых услуг. Так же в базе данных ведется учет посещаемости занятий и оплата услуг клиентами. Создание базы данных позволит упростить выборку информации, внесение изменений и добавление новых записей. Кроме того база данных позволит просматривать расписание занятий в удобной форме. База данных значительно упрощает работу фитнес тренеров и позволяет фитнес-центру занимать достойное место в сфере фитнеса.

Целью данной работы является реализация подобной информационной системы. В частности системы взаимодействия клиента и его инструктора, которая позволит клиентам фитнес-клуба полностью забыть о проблемах планирования своего режима тренировок и составления плана занятий.

Глава 1. Обзор предметной области

1.1 Обзор предметной области

В качестве предметной области данной курсовой работы был выбран фитнес-клуб и его внутренняя система взаимодействия клиента с инструктором.

Каждый посетитель тренажерного зала хочет качественно организованного плана тренировок от квалифицированного профессионала в фитнес сфере. Но это довольно затратная услуга, и чтобы повысить качество и доступность предоставляемых услуг в фитнес-центре следует переходить на удаленное менторство тренера и его клиента. Это позволит значительно уменьшить финансовые затраты посетителей на персональные тренировки и увеличить общее количество подопечных каждому инструктору.

1.2 Проектирование базы данных и составление ER - модели

В проектируемой базе данных хранится информация о тренерах, клиентах и их тренировках. Схема базы данных представлена на рисунке 1 (создана при помощи Creately[1])

Устанавливая связи между сущностями, учтены следующие закономерности:

Тренер может иметь несколько клиентов, а может не иметь ни одного, например, в случае, если он только устроился на работу. Клиент, наоборот, изначально должен иметь инструктора и причем только одного.

Клиент имеет множество тренировок, но если, например, он только подписал договор с клубом, он, конечно, может не иметь тренировок.

Создание тренировки сопровождается созданием упражнения, в свою очередь упражнение в аналогичной связи с подходом, то есть тренировки без

упражнения быть не может, как и упражнения без подхода. Данные запросы будут заключены в одну транзакцию во избежании утечки информации.

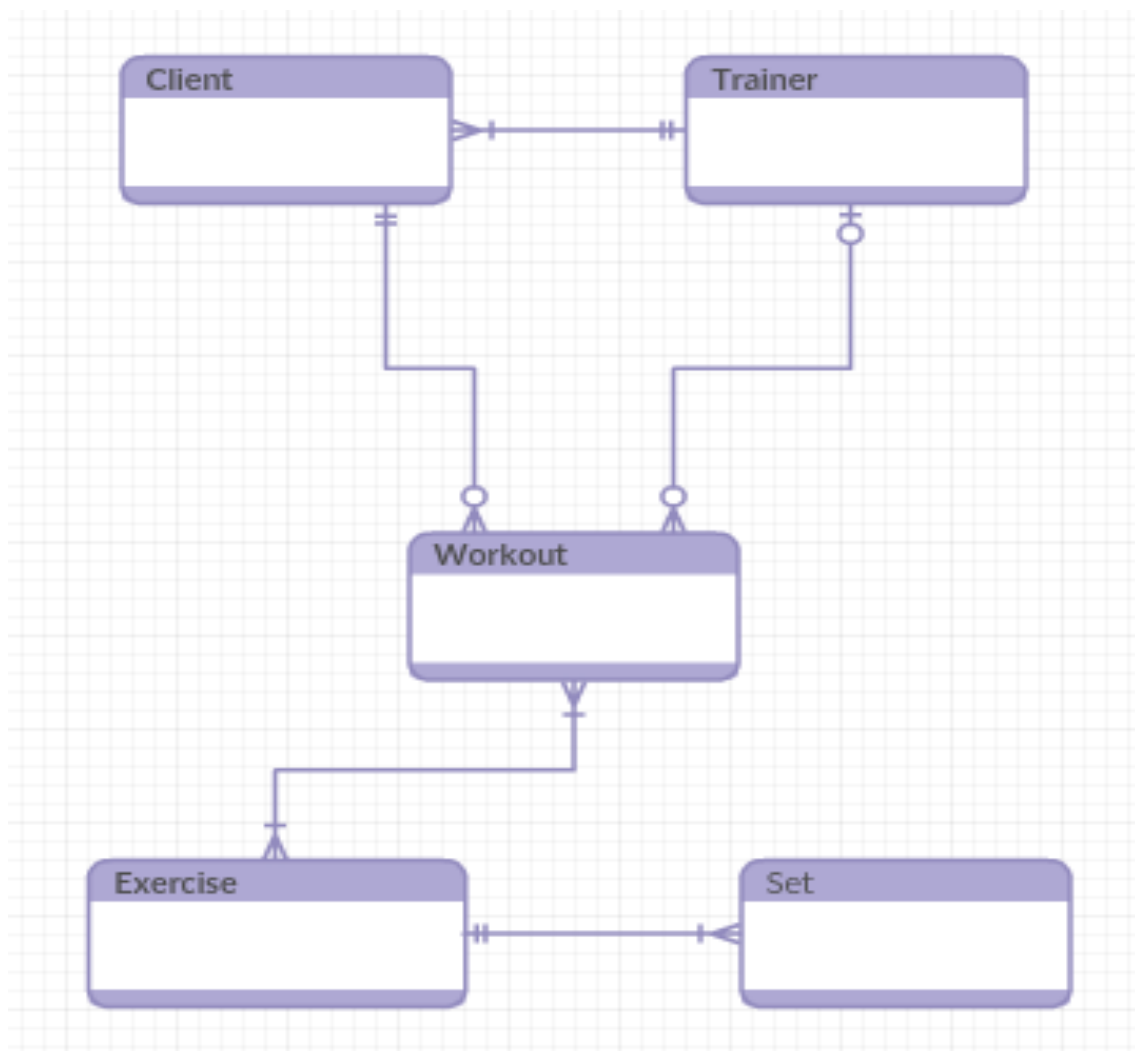


Рисунок 1. ER - модель базы данных

Глава 2. Разработка реляционной базы данных

Для реализации базы данных была выбрана свободная объектно-реляционная система управления базами данных PostgreSQL [2]. Данная СУБД предоставляет множество различных возможностей, достаточно надежна и имеет хорошие характеристики по производительности, свободно распространяется и имеет открытый исходный код.

Итоговая база данных `fitnes_club` содержит единственную схему `club_schema`, в которой содержатся таблицы, соответствующие вышеперечисленным сущностям.

После процесса нормализации спроектированной базы данных, были удалены зависимость «много ко многим», вынесены в отдельную сущность избыточные данные по персональным данным человека.

Реляционная модель данной базы данных представлена на рисунке 2.

Сущность «Person» содержит персональные данные человека:

- `personID` – идентификатор сущности;
- `firstName` - имя;
- `lastName` - фамилия;
- `middleName` - отчество.

Сущность «Trainer» содержит в себе поля:

- `trainerID` – идентификатор клиента;
- `isFired` - статус работы;
- `directionOfTraining` - направление тренировок (похудение, набор масса)
- `regalia` - регалии инструктора;
- `personID` - идентификатор персональных данных.
-

Сущность «Client» содержит в себе поля:

- `clientID` – идентификатор клиента;

- phone – телефонный номер клиента;
- trainerID – идентификатор инструктора;
- personID - идентификатор персональных данных.

Сущность «Workout» содержит в себе поля:

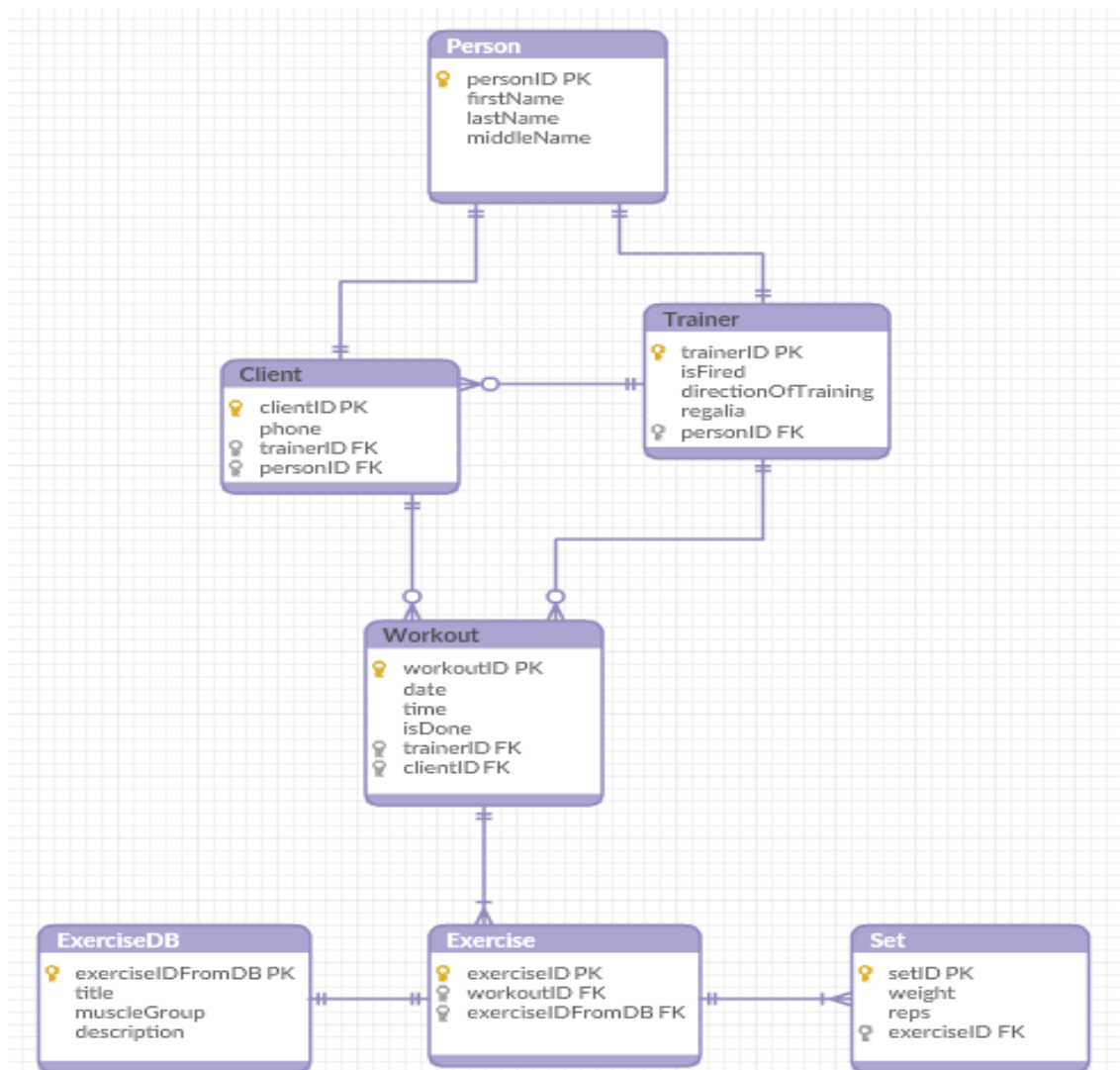


Рисунок 2. Реляционная модель базы данных

- workoutID – идентификатор тренировки;
- date – дата проведения тренировки;
- time – время проведения тренировки;
- isDone – тип тренировки (спланированная тренером, выполненная клиентом);
- clientID – идентификатор клиента;

- `trainerID` – идентификатор инструктора.

Сущность «Exercise» отвечает за упражнения в тренировке и содержит в себе поля:

- `exerciseID` – идентификатор упражнения;
- `workoutID` – идентификатор тренировки;
- `exerciseIDFromDB` – идентификатор упражнения из списка всех возможных упражнений.

Сущность «Set» отвечает за количество подходов в упражнении и содержит в себе поля:

- `setID` – идентификатор подхода;
- `weight` – вес снаряда в подходе;
- `reps` – количество повторений в подходе;
- `exerciseID` – идентификатор упражнения.

Сущность «ExerciseFromDB» содержит в себе поля:

- `exerciseIDFromDB` – идентификатор упражнения из списка всех возможных упражнений;
- `title` – наименование упражнения;
- `muscleGroup` – целевая мышечная группа;
- `description` – техника выполнения упражнения.

Связь сущностей `Trainer` и `Client` означает, что соответствующий тренер приставлен соответствующему клиенту.

Связь `Client` и `Workout` или `Trainer` и `Workout` означает принадлежность тренировки к данному пользователю и возможность иметь доступ к значениям сущностей `Exercise` и `Set`.

Атрибут `isDone` сущности `Workout` определяет источник создания тренировки, т.е. `FALSE` свидетельствует о том, что данная тренировка

спланирована тренером, TRUE указывает на то, что тренировка выполнена клиентом.

Таблица ExerciseDB представляет собой перечень всех возможных упражнений, которые можно записать в тренировку. Описание таблиц базы данных представлено в приложении А.

Глава 3. Разработка приложения

3.1 Используемые технологии

В ходе данной курсовой работы также было реализовано приложение с графическим интерфейсом. В качестве языка реализации был выбран Python [4] с библиотекой PyQt [4], которая и позволяет создавать графический интерфейс.

3.2 Регистрация и авторизация

С самого начала пользователя встречает окно авторизации (см. Рисунок 3), в котором предоставляется выбор между входом в систему и регистрацией. Само собой новому клиенту требуется прежде всего зарегистрироваться, введя свои персональные данные. По итогу регистрации клиент получает свой уникальный идентификационный номер, с помощью которого он может зайти в систему.

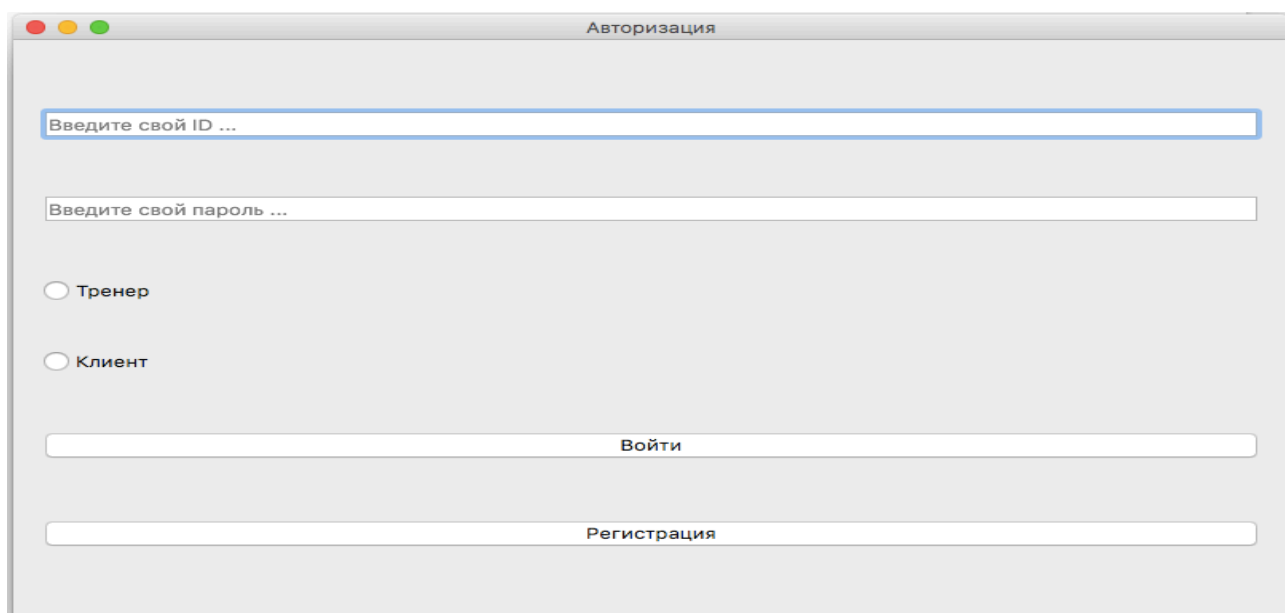


Рисунок 3. Окно авторизации

Но прежде чем зайти в систему, в ней нужно зарегистрироваться. На рисунке 4 показано окно регистрации с полями:

- 1) Имя пользователя
- 2) Фамилия
- 3) Отчество
- 4) Тип пользователя (Тренер или Клиент)
- 5) Пароль
- 6) Фамилия тренера (заполняется, если регистрируется клиент)

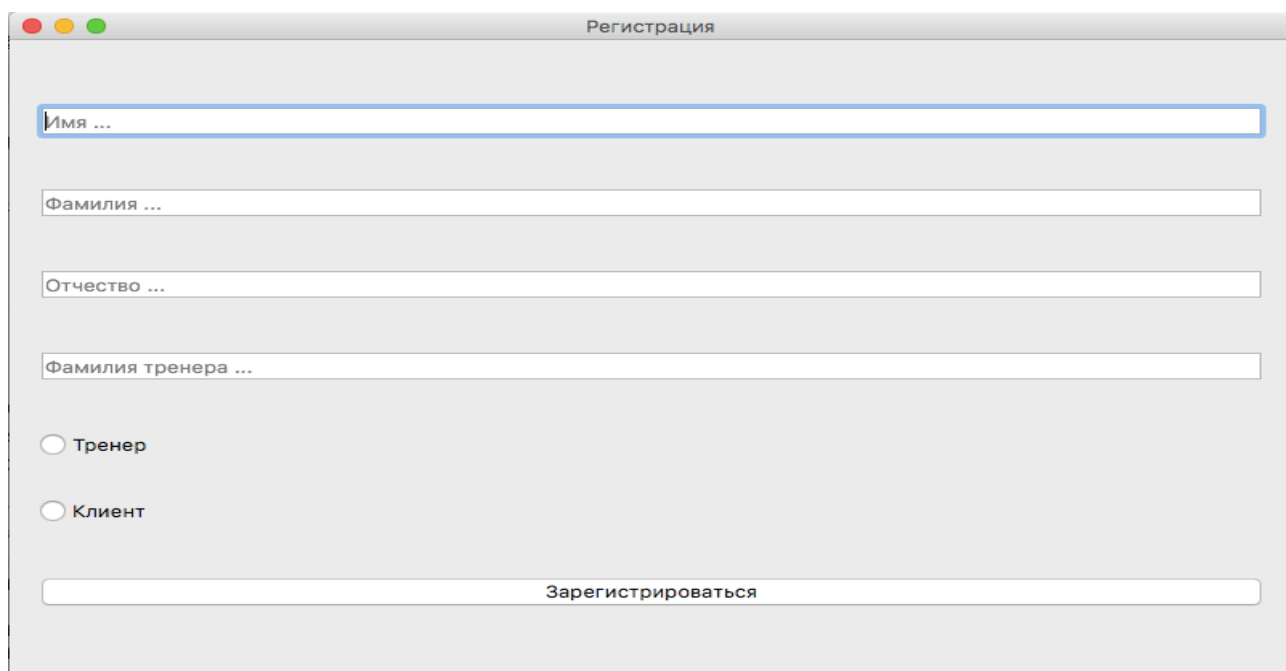
The image shows a screenshot of a web application window titled "Регистрация" (Registration). The window has a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. The registration form consists of several text input fields and two radio buttons. The fields are labeled: "Имя ..." (Name), "Фамилия ..." (Surname), "Отчество ..." (Patronymic), and "Фамилия тренера ..." (Coach's surname). Below these fields are two radio buttons labeled "Тренер" (Coach) and "Клиент" (Client). At the bottom of the form is a wide button labeled "Зарегистрироваться" (Register).

Рисунок 4. Окно регистрации

По окончании регистрации пользователю выдается уникальный идентификационный номер (см. Рисунок 5)

При входе в систему также требуется указать статус пользователя: «Тренер», «Клиент», так для каждого типа пользователя предоставлен свой интерфейс и своя функциональность системы.



Рисунок 5. Диалоговое окно с

3.3 Тренер

Войдя в систему как тренер перед пользователем открывается окно (см. Рисунок 6), разделенное на сегменты:

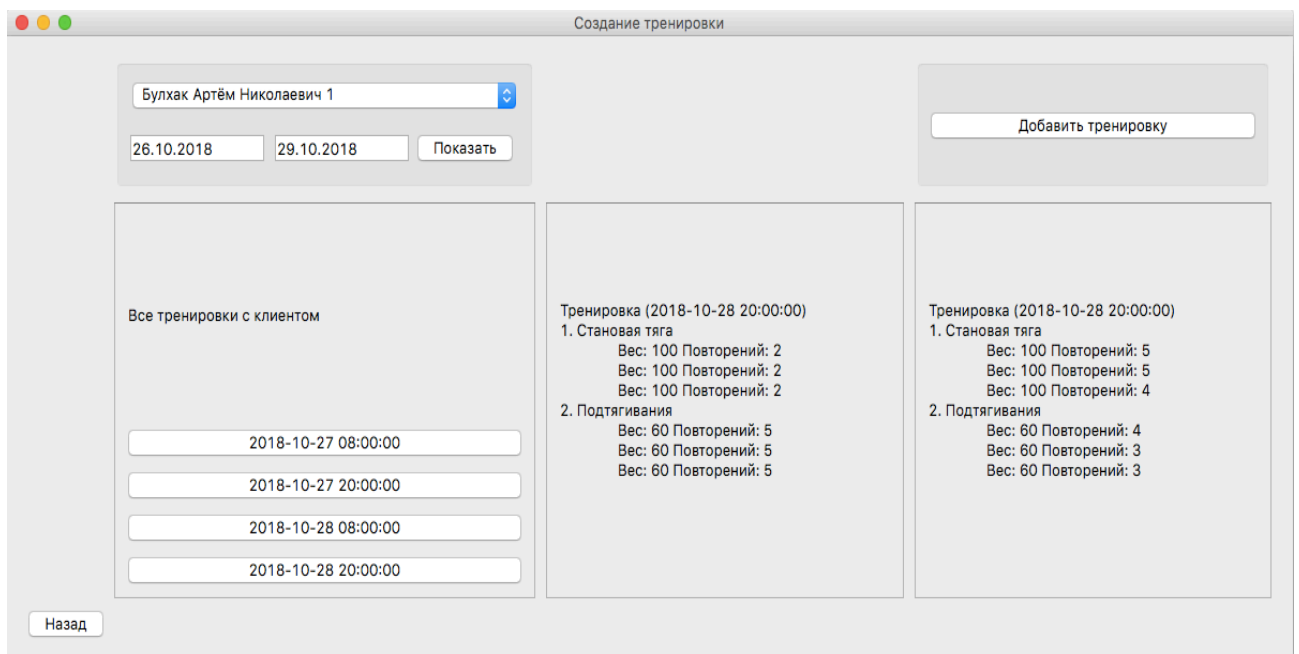


Рисунок 6. Меню тренера

- 1) Область выбора клиента
- 2) Область выбора промежутка времени, за который тренер хочет получить записи тренировок
- 3) Список тренировок

- 4) Окно с планом тренировки подготовленным тренером
- 5) Окно с результатом прохождения клиентом спланированной тренировки
- 6) Кнопка создания новой тренировки

При нажатии кнопки «Добавить тренировку» открывается небольшое окно (см. Рисунок 7) со следующими пунктами:

- 1) Дата тренировки
- 2) Время тренировки
- 3) Кнопка «Добавить упражнение»
- 4) Кнопка «Удалить упражнение»
- 5) Кнопка «Добавить тренировку»

Рисунок 7. Создание тренировки

При добавлении упражнения в области тренировки появляется «ComboBox», в котором перечислены все возможные упражнения, хранящиеся в базе упражнений, и кнопки добавления и удаления подходов выбранного упражнения. В каждом подходе требуется вписать вес снаряда и количество

повторений, которое требуется выполнить клиенту.

Видя все на одном экране, тренер имеет возможность, не совершая лишних движений, просматривать предыдущие результаты своих подопечных и планировать для них будущие тренировки, анализируя полученные данные.

3.4 Клиент

Войдя в систему как клиент перед вами открывается окно (см. Рисунок 8) со следующими действиями:

- 1) Пройти тренировку
- 2) Посмотреть журнал тренировок
- 3) Сменить тренера

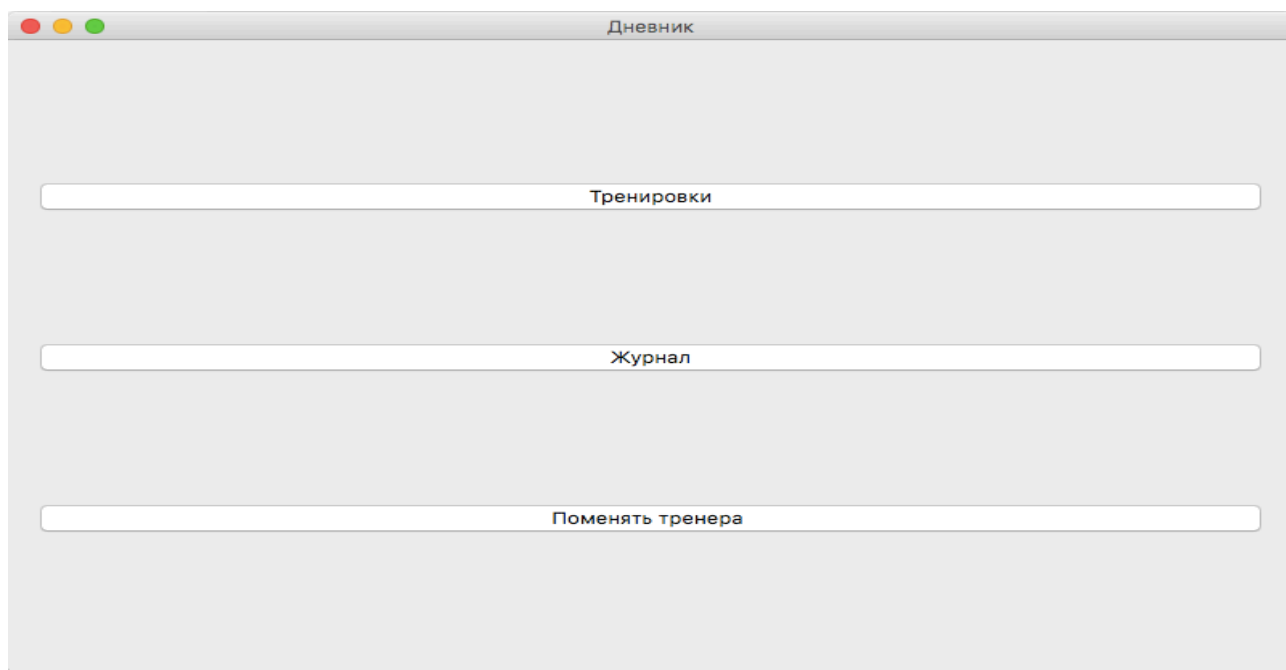


Рисунок 8. Меню клиента

Нажав первую кнопку, пользователю предоставляется список, спланированных тренером, тренировок текущего дня (см. Рисунок 9). Выбирая тренировку по времени дня, клиент может видеть план выбранной тренировки в

окне «Спланированная тренировка», а в окне «Текущая тренировка» автоматически выбираются спланированные упражнения с нужным количеством подходов и весом снаряда. Клиенту остается только четко следовать плану и записывать свои результаты в поля повторений. Если клиент в этот день не полностью восстановился, он, обговорив это с инструктором, может поменять упражнения текущей тренировки на те, которые ему будет комфортно выполнять. И в будущем тренер, просматривая журнал тренировок, учтет восстановительные способности своего клиента, при составлении тренировок.

Тренировка

Тренировка (09:00:00)

1) 09:00:00

1. Жим лежа

1) Вес: 100 Повторения: 5

2) Вес: 100 Повторения: 5

2. Становая тяга

1) Вес: 100 Повторения: 8

2) Вес: 100 Повторения: 8

Упражнение 1

Жим лежа

100 Повторения ...

100 Повторения ...

Упражнение 2

Становая тяга

100 Повторения ...

100 Повторения ...

Добавить упражнение

Удалить упражнение

Добавить тренировку

Назад

Рисунок 9. Тренировка дня

Перейдя во вкладку «Журнал тренировок» (см. Рисунок 10) клиент получает возможность, выбирая даты на календаре, просматривать свои тренировки и следить за прогрессом или, наоборот, регрессом в определенных упражнениях.

Вкладка «Сменить тренера» (см. Рисунок 11) нужна в тех случаях, когда по какой-либо причине пользователь желает сменить инструктора. При смене инструктора, все данные прошлых тренировок сохраняются, более того у нового инструктора появится возможность просмотреть старые тренировки нового подопечного.

Интерфейс приложения сконструирован таким образом, чтобы человеку любого возраста было максимально понятно, как пользоваться данным приложением.

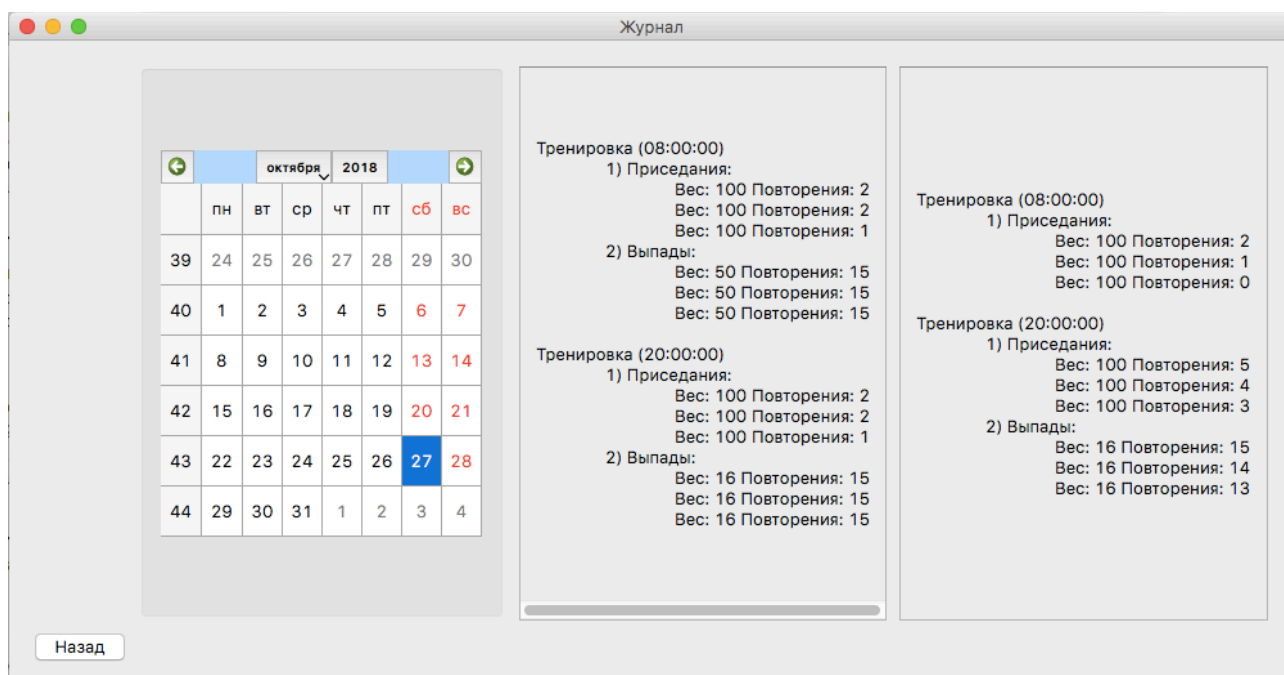


Рисунок 10. Журнал тренировок

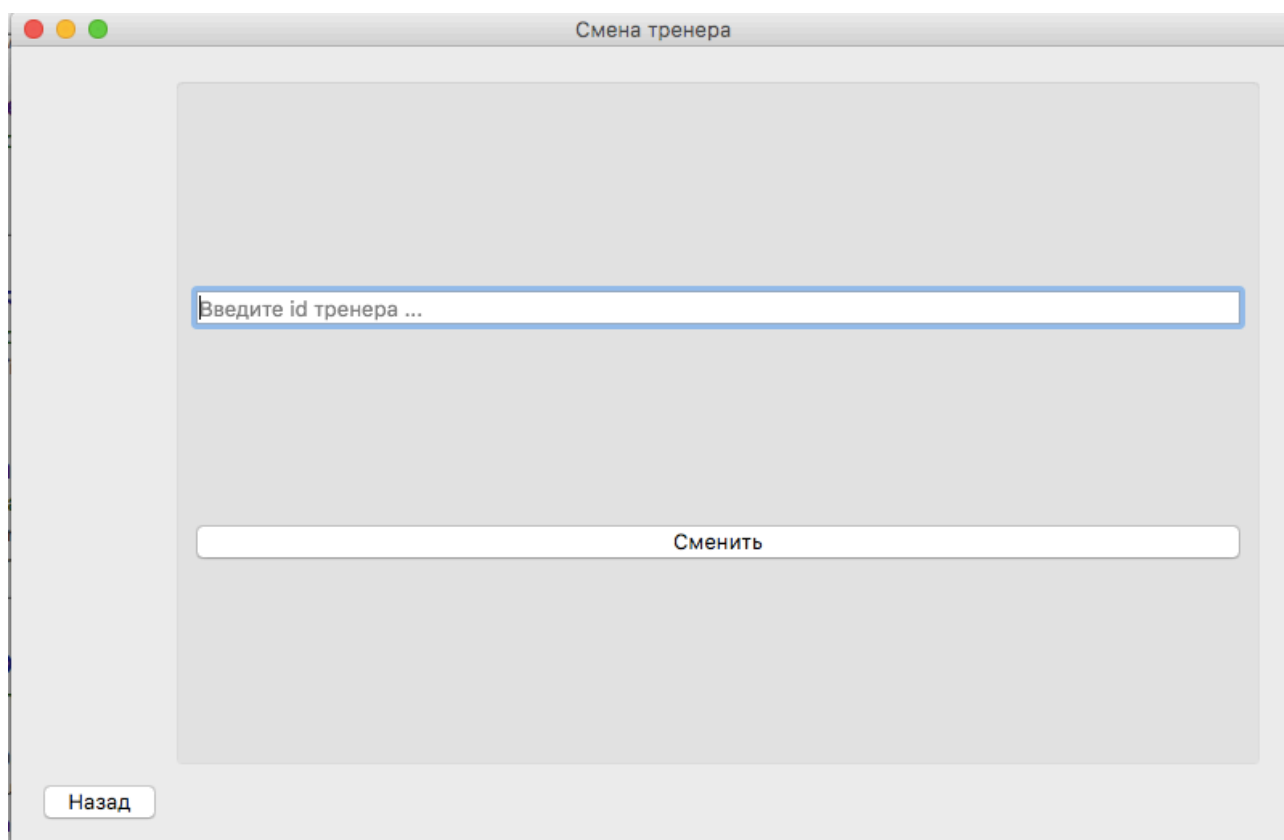


Рисунок 11. Смена тренера

Глава 4. Тестирование программной реализации

Тестирование системы производилось путём замера времени выполнения запросов, используемых в базе данных. В среднем человек тренируется 2-3 раза в неделю, поэтому для тестирования будем считать, что каждый пользователь отзанимался год, а в году тренировался через день, чтобы специально нагрузить систему.

Тестирование осуществлялось на системе с процессором Intel Core i5 1,6 GHz с 4 ГБ ОЗУ.

Проводилось сравнение времени выполнения запросов в базе данных при хранении записей. Были протестированы используемые запросы:

1. SELECT firstName, lastName, middleName, clientID FROM ClubSchema.Client WHERE trainerID = %s
2. SELECT date, time FROM ClubSchema.Workout WHERE clientID = %s AND isDone = FALSE AND date BETWEEN %s AND %s
3. SELECT time, name, weight, reps FROM ClubSchema.WorkoutView WHERE date = current_date AND isDone = FALSE AND clientID = %s
4. INSERT INTO ClubSchema.Client (firstName, lastName, middleName, trainerID) VALUES (%s, %s, %s, %s)
5. INSERT INTO ClubSchema.Workout (date, time, isDone, trainerID, clientID) VALUES (current_date, current_time, TRUE, %s, %s)

Результаты тестирования представлены в таблице 1.

Таблица 1. Тестирование времени выполнения SQL запросов

Номер запроса	Кол-во тренеров	Кол-во клиентов	Ср. результат, мс
1	5	10	2
	50	1000	20
	500	100000	50
2	5	10	3
	50	1000	35
	500	100000	60
3	5	10	3
	50	1000	70
	500	100000	150
4	5	10	5
	50	1000	10
	500	100000	21
5	5	10	30
	50	1000	40
	500	100000	100

В ходе тестирования было выявлено, что информационная система способна работать очень продолжительный срок без значительной потери производительности, так как результаты тестирования запросов, когда количество клиентов равно 10 отличаются от результатов в случае, когда клиентов 100000 в пределах допустимых значений.

Заключение

В ходе данной курсовой работы была реализована информационная система для поддержки фитнес-центра. Она позволяет упростить взаимодействие персонального инструктора и клиента, нагляднее прослеживать результаты подопечных и работать удаленно.

Была исследована объектно-реляционная система управления базами данных PostgreSQL и библиотека PyQt, которая позволяет создавать графический интерфейс.

Проведено тестирование, которое наглядно показало надежность системы.

В дальнейшем возможны доработки, направленные на разработку более органичного интерфейса.

Список литературы

- [1] Документация Creately [Электронный ресурс] // URL: <https://creately.com>;
- [2] Документация PostgreSQL [Электронный ресурс] // URL: <https://www.postgresql.org/docs/>;
- [3] Документация языка Python [Электронный ресурс] // URL: <https://www.python.org/doc/>;
- [4] Документация библиотеки PyQt5 [Электронный ресурс] // URL: <https://pyqt.sourceforge.net/Docs/PyQt5/>;

ПРИЛОЖЕНИЕ А. Описание таблиц

Описание каждой из таблиц в базе данных представлено в таблицах А.1-А.7.

Таблица А.1 – описание таблицы “Trainer”.

Имя столбца	Тип данных	Ключ	NULL	Описание
trainerID	INTEGER	PK	NOT NULL	Суррогатный ключ
isFired	BOOLEAN	-	NOT NULL	DEFAULT FALSE
directionOfTraining	TEXT	-	NULL	-
Regalia	TEXT	-	NULL	-
personID	INTEGER	FK	NOT NULL	-

Таблица А.2 – описание таблицы “Client”.

Имя столбца	Тип данных	Ключ	NULL	Описание
clientID	INTEGER	PK	NOT NULL	Суррогатный ключ
trainerID	INTEGER	FK	NOT NULL	-
personID	INTEGER	FK	NOT NULL	-

Таблица А.3 – описание таблицы “Workout”.

Имя столбца	Тип данных	Ключ	NULL	Описание
workoutID	INTEGER	PK	NOT NULL	Суррогатный ключ
date	DATE	-	NOT NULL	-
time	TIME	-	NOT NULL	-
isDone	BOOLEAN	-	NOT NULL	FALSE - запланированная TRUE - выполненная
clientID	INTEGER	FK	NOT NULL	-
trainerID	INTEGER	FK	NOT NULL	-

Таблица A.4 – описание таблицы “Exercise”.

Имя столбца	Тип данных	Ключ	NULL	Описание
exercisID	INTEGER	PK	NOT NULL	Суррогатный ключ
clientID	INTEGER	FK	NOT NULL	-
exercisIDFromDB	INTEGER	FK	NOT NULL	-

Таблица A.5 – описание таблицы “Set”.

Имя столбца	Тип данных	Ключ	NULL	Описание
setID	INTEGER	PK	NOT NULL	Суррогатный ключ
weight	INTEGER	-	NOT NULL	-
reps	INTEGER	-	NOT NULL	-
exercisID	INTEGER	FK	NOT NULL	-

Таблица A.6 – описание таблицы “ExerciseFromDB”.

Имя столбца	Тип данных	Ключ	NULL	Описание
exercisIDFromDB	INTEGER	PK	NOT NULL	Суррогатный ключ
title	VARCHAR	-	NOT NULL	-
muscleGroup	VARCHAR	-	NOT NULL	-
description	TEXT	-	NULL	-

Таблица А.7 – описание таблицы “Person”.

Имя столбца	Тип данных	Ключ	NULL
personID	INTEGER	PK	Суррогатный ключ
firstName	VARCHAR	-	NOT NULL
lastName	VARCHAR	-	NOT NULL
middleName	VARCHAR	-	NULL