

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

## Отчет по курсовой работе

Дисциплина: «Программное обеспечение распределенных  
вычислительных систем»

Тема: «**Разработка информационной системы для  
спортивного клуба**»

Выполнил:  
студент группы №23541/3 Родина В.В.

Принял:  
к.т.н., доц. Стручков И. В.

Санкт-Петербург

2018

# **Анализ задания:**

Разработать информационную систему фитнес-клуба.

## **1. Роли и функциональные требования**

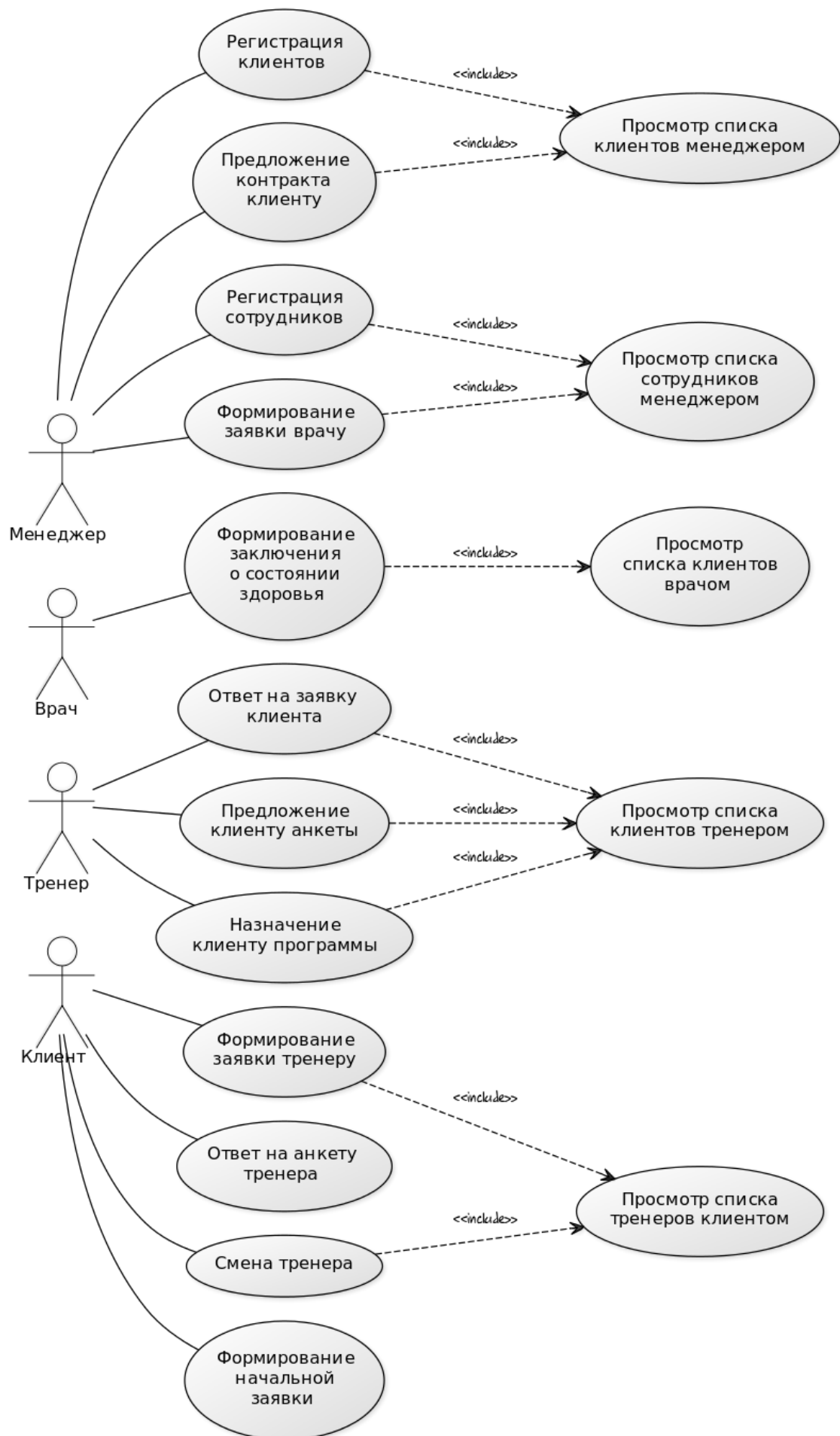
**В информационной системе предусматривается наличие следующих ролей:**

- Менеджер клуба
- Врач
- Тренер
- Клиент

**Функциональные требования:**

- Менеджер
  - Просмотр списка нанятых сотрудников
  - Регистрация новых сотрудников
  - Просмотр списка клиентов клуба
  - Регистрация новых клиентов
  - Формирование заявки врачу
- Тренер
  - Просмотр списка клиентов клуба
  - Ответ на заявку клиента
  - Предложение клиенту заполнить анкету
  - Назначение тренировочной программы клиенту
- Клиент
  - Формирование начальной заявки менеджеру
  - Заполнение анкеты и формулировка целей
  - Формирование заявки тренеру
  - Просмотр списка тренеров
- Врач
  - Формирование заключения о здоровье клиента
  - Просмотр списка заявок на обследование

## **2. Разработка вариантов использования (обобщенная диаграмма прецедентов)**



### **3. Подробное описание всех вариантов использования (текстовое описание с альтернативами)**

- **Менеджер**

- Просмотр списка нанятых сотрудников и информации о них
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает вкладку "Сотрудники" для просмотра соответствующего списка
  - Система отображает список
- Просмотр списка клиентов клуба и информации о них
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает вкладку "Клиенты" для просмотра соответствующего списка
  - Система отображает список
- Регистрация новых сотрудников
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает вкладку "Сотрудники" для просмотра соответствующего списка
  - Система отображает список
  - Пользователь нажимает кнопку "Новый сотрудник"
  - Система отображает диалоговое окно для ввода информации о сотруднике
  - Пользователь заполняет формы и нажимает "Ок"
  - Система заносит в БД нового сотрудника
- Регистрация новых клиентов
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает вкладку "Клиенты" для просмотра соответствующего списка
  - Система отображает список
  - Пользователь нажимает кнопку "Одобрить контракт"
- Формирование заявки врачу
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает кнопку "Все сотрудники" для просмотра соответствующего списка

- Система отображает список
  - Пользователь выделяет врача в списке и нажимает кнопку "Сформировать заявку"
  - Система отображает диалоговое окно, позволяющее сформировать заявку врачу.
- Расчет зарплаты тренера на основе оклада и оценок клиентов
  - Пользователь авторизуется в системе как "Менеджер"
  - Пользователь нажимает кнопку "Все тренеры" для просмотра соответствующего списка
  - Система отображает список
  - Пользователь выделяет тренера в списке и нажимает кнопку "Рассчитать зарплату"
- **Тренер**
  - Просмотр списка клиентов клуба и информации о них
    - Пользователь авторизуется в системе как "Тренер"
    - Система отображает список клиентов
  - Ответ на заявку клиента
    - Пользователь авторизуется в системе как "Тренер"
    - Система отображает список, в котором помечены клиенты, подавшие заявку
    - Пользователь выбирает клиента из списка и нажимает кнопку "Подтверждение".
  - Предложение клиенту заполнить анкету
    - Пользователь авторизуется в системе как "Тренер"
    - Система отображает список клиентов
    - Пользователь выбирает клиента из списка и нажимает кнопку "Послать анкету"
  - Назначение тренировочной программы клиенту
    - Пользователь авторизуется в системе как "Тренер"
    - Система отображает список клиентов
    - Пользователь выбирает клиента из списка и нажимает кнопку "Назначить программу"
    - Система показывает диалоговое окно для ввода информации о тренировочной программе

- Пользователь пишет в окне программу и нажимает кнопку "Отправить"

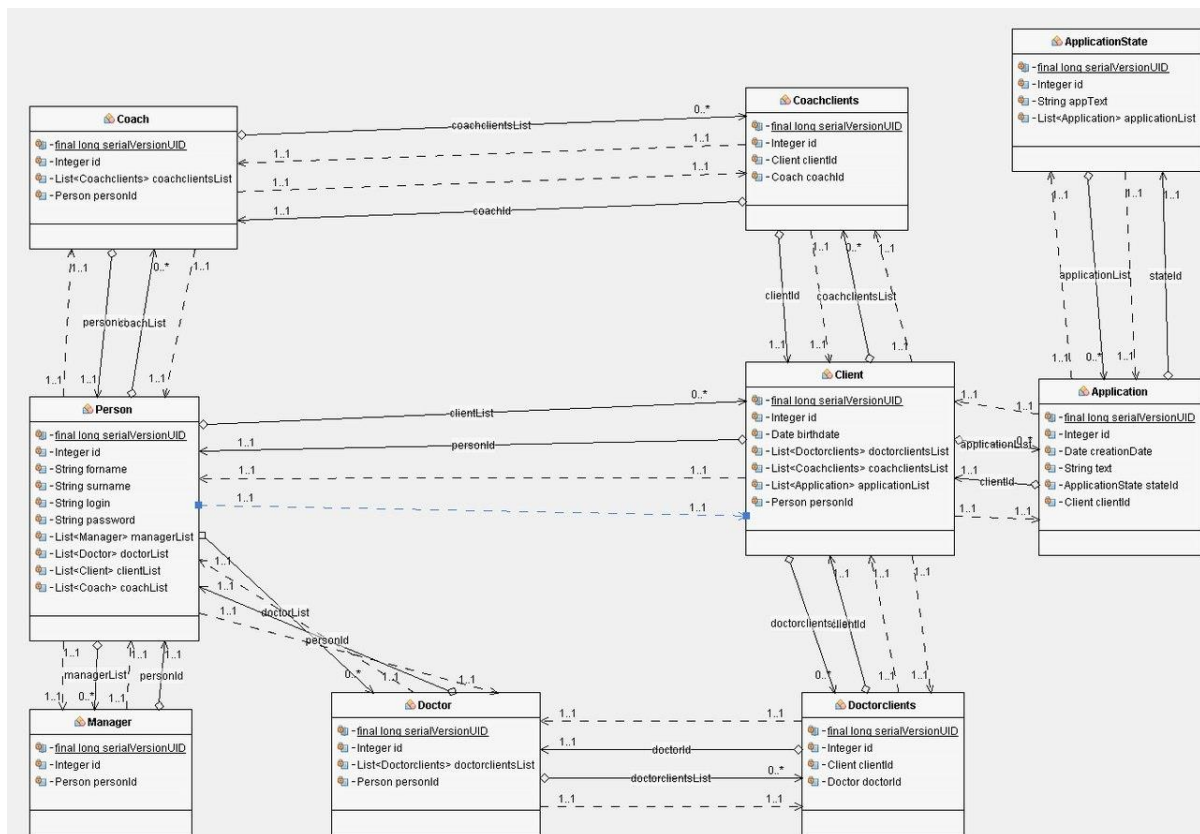
- **Клиент**

- Просмотр списка тренеров и информации о них
  - Пользователь авторизуется в системе как "Клиент"
  - Пользователь нажимает кнопку "Сменить тренера" для просмотра соответствующего списка, если его контракт был одобрен менеджером
  - Система отображает список
- Выбор тренера
  - Пользователь авторизуется в системе как "Клиент"
  - Пользователь нажимает кнопку "Сменить тренера" для просмотра соответствующего списка, если его контракт был одобрен менеджером
  - Система отображает список
  - Пользователь выбирает тренера из списка и нажимает кнопку "Выбрать"
- Заполнение анкеты и формулировка целей
  - Пользователь авторизуется в системе как "Клиент"
  - Пользователь нажимает кнопку "Заполнить анкету" для отображения соответствующей формы, если тренер отправил ему анкету
  - Система отображает окно с анкетой
  - Пользователь заполняет анкету и отправляет, нажав кнопку "Отправить"

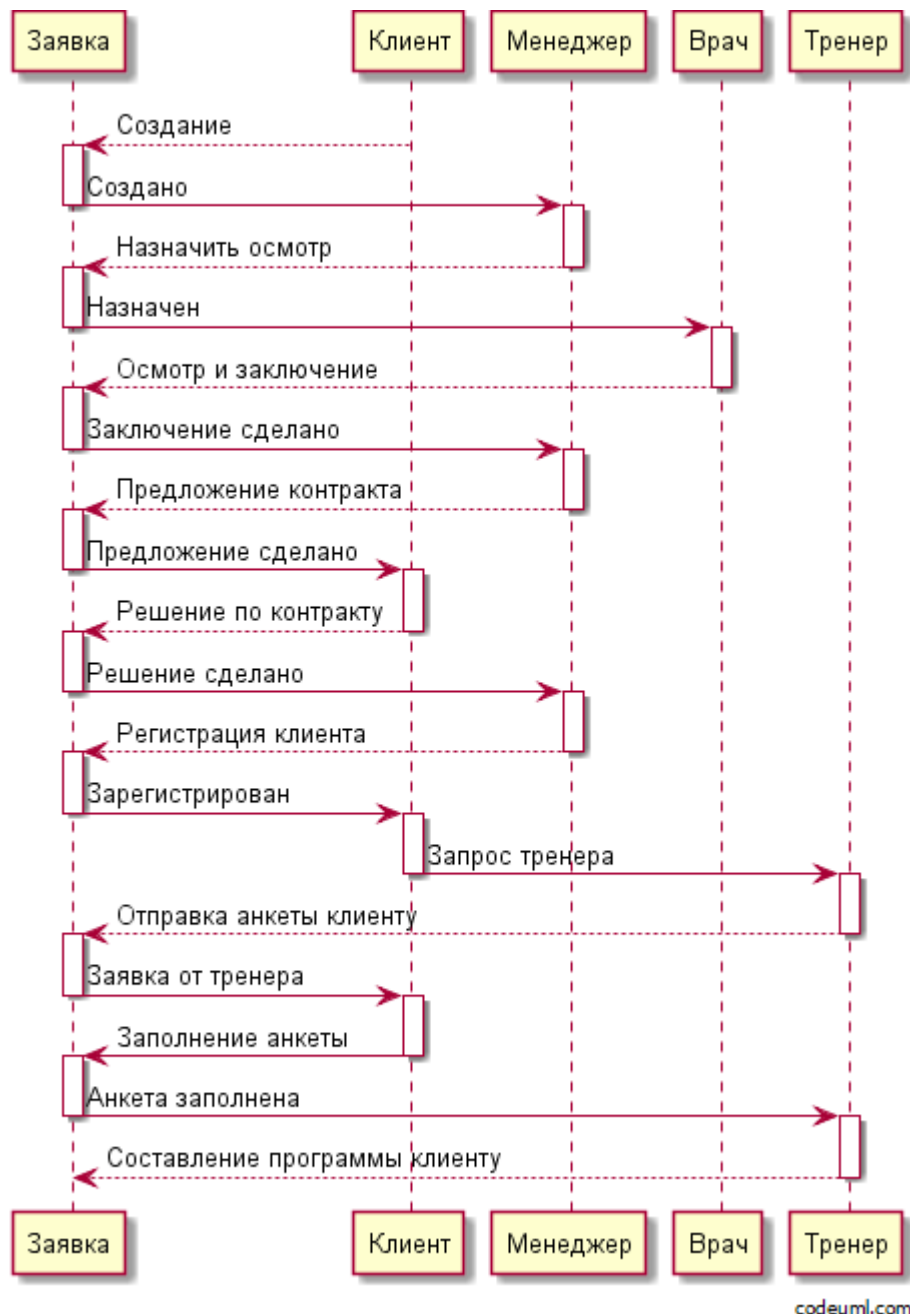
- **Врач**

- Формирование заключения о здоровье клиента
  - Пользователь авторизуется в системе как "Врач"
  - Система отображает список клиентов, которым назначен прием у этого врача
  - Пользователь выбирает клиента из списка и нажимает кнопку "Обследован", если клиент прошел обследование

## 4. Диаграмма классов сущностей



## 5. Диаграмма последовательности



В качестве СУБД для проекта была выбрана стандартная Java DB, поскольку для нее не надо дополнительно ничего устанавливать, она проста в обслуживании и управлении.

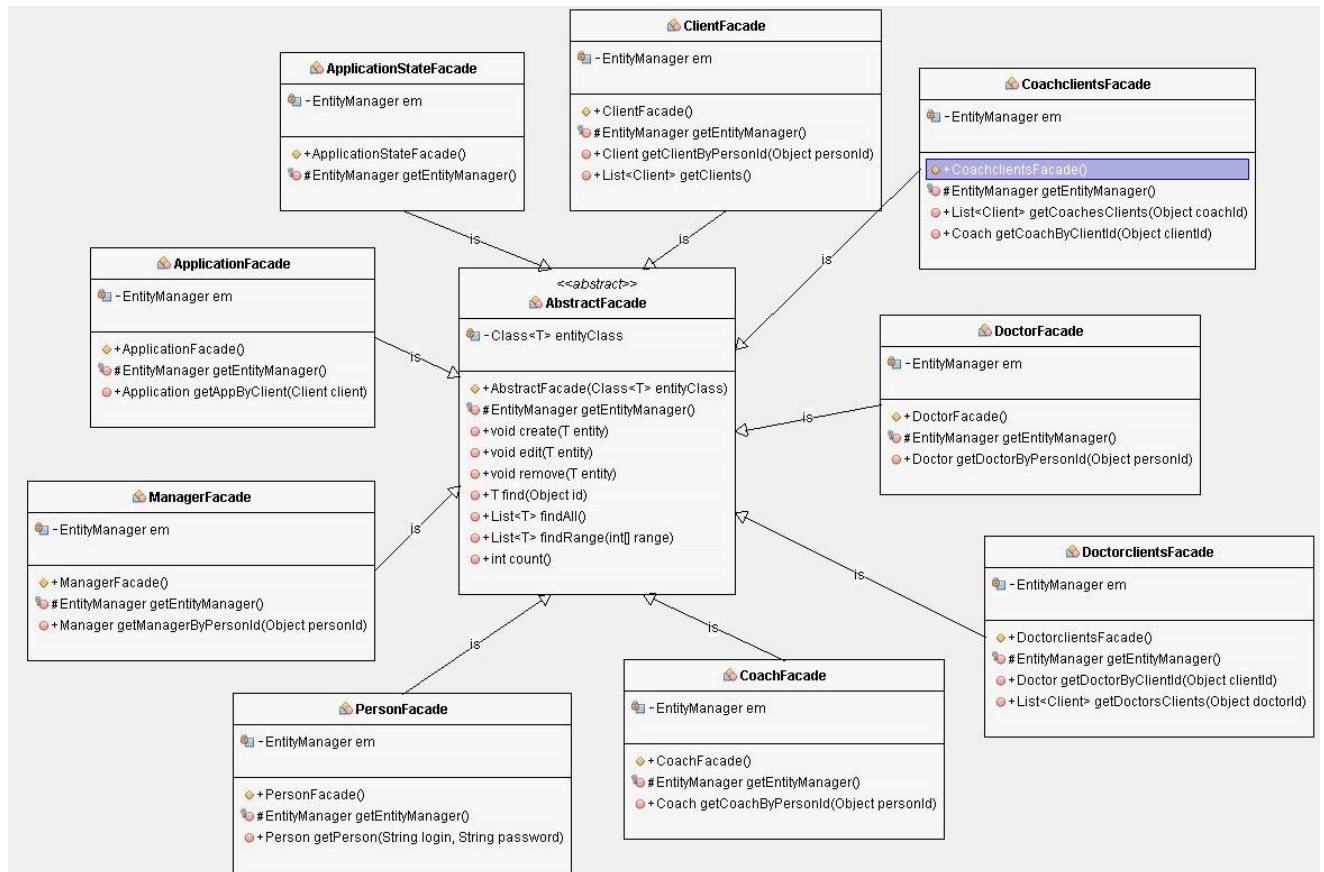
В качестве основы для бизнес-логики используются сеансовые компоненты без состояния, поскольку все данные хранятся в БД.

Что касается слоя доступа к данным, то он был сгенерирован на основе имеющейся базы данных. В частности, были сгенерированы автоматически все сущностные классы. В каждый сущностный класс помимо сгенерированных были добавлены несколько именованных запросов для повышения удобства использования. Помимо сущностных классов были сгенерированы фасады для управления транзакциями и доступа к сущностям.



# Описание программы

В ходе разработки было принято решение использовать фреймворк JSF для отображения информации на страницах. Основная бизнес-логика при этом сосредоточена в Managed Beans, которые, в свою очередь, обращаются к локальным фасадам сущностей.



## Текст программы

Смотреть репозиторий: [https://github.com/valeryrodina/Fitness\\_club\\_web\\_app](https://github.com/valeryrodina/Fitness_club_web_app)

## Методика и результаты тестирования

Тестирование проведено функциональное ручное исследовательское на соответствие ПО требованиям, поставленных в начале на этапе анализа данных.

# Инструкция системного администратора

Для работы приложения понадобится СУБД Java DB (Derby). Пользователь - qwerty, пароль – qwerty. Для создания бд и таблиц выполнить скрипты db\_create.sql и db\_fill\_appstate.sql. Для развертки приложения необходим GlassFish Server версии 4.1.2.

Запуск домена: asadmin start-domain domain1

Развернуть приложение на сервере GlassFish: asadmin deploy project.war  
(предварительно клонировать репозиторий на компьютер)

## Инструкция пользователя

В окне авторизации выбрать нужную вкладку человека и ввести логин и пароль

### Клиент

В окне клиента следовать инструкциям, написанным в поле Your app status.

### Менеджер

Таблицы врачей и тренеров отображают всех сотрудников.

Таблица клиентов отображает всех клиентов и статусы их заявок. Напротив каждого клиента есть кнопка для изменения статуса заявки.

### Доктор

Таблица клиентов отображает текущих назначенных доктору клиентов и статусы их заявок.

Нажимая на кнопку напротив клиента доктор подтверждает, что клиент здоров.

### Тренер

Таблица клиентов отображает текущих клиентов тренера и позволяет изменять статусы их заявок путем нажатия кнопки напротив. Для введения необходимой информации об анкете или программе необходимо использовать поле ввода рядом с кнопкой.

# Выводы

В результате выполнения данной работы были освоены основные технологии Java EE, использующиеся в построении распределенных информационных систем: фреймворк JSF 2, Enterprise Java Beans, Java Persistence API, etc.

В ходе работы были проанализированы роли, для каждой роли были составлены use cases. Далее были разработаны сущности, бизнес-логика и веб-интерфейс для приложения. Поскольку приложение разрабатывалось на платформе Java EE с применением EJB, то оно имеет большой потенциал для масштабирования. Бизнес логика четко отделена от слоя представления и от слоя доступа к данным. Однако, одним из наиболее уязвимых мест являются сущности. Их изменение может повлечь за собой крупные изменения в проекте.

В настоящее время данная технология хорошо зарекомендовала себя в нише enterprise-приложений, поскольку позволяет создавать высоконагруженные приложения высокого качества. Применяя технологии Java EE можно добиться значительной производительности приложения. Однако одним из существеннейших минусов является высокая сложность отладки, поскольку множество компонентов создаются неявно и от программиста скрыты реализации некоторых компонентов.