**Project Deliverable 1 (PD1)**

Project specification

I choosed to make a Java and Spring web application, specifically an application that manages all the clients of different gyms. The application is inspired by the “UPfit.today” app. My application will allow gyms to enroll in the system, when a gym is enrolled, each customer of that gym has to register and login, there, the user can see his current gym membership and he can enroll to multiple training programs or make appointments for different trainers.

The application is basically a CRUD app. It will have three types of users: admins, clients and trainers. The admin will be able to add or delete training programs, the trainer will be able to see and accept training requests from the users and finally, the user will be able to see all the training programs available, all the trainers and his status as a gym member.

Domain model diagram

Diagram

Description automatically generated

Users and stakeholders

* Clients = regular people who want to go to the gym
* Admins = people who manages the gym, add new programs and validates trainers
* Trainers = a special kind of clients who have more features available
* Developers = people involved in the development of the application
* Trainers form other gyms that can use the application as normal users

Use case identification

Use case no.1

Use case name: User interacts for the first time with the application.

Level: Making an account and an appointment for a trainer – user goal

Main actor: A random client .

Main success scenario: The user first opens the app, presses the register button then he add all the credentials, he logs in then he searches for some qualified trainers and chooses one then he makes an appointment on a certain date for a session with that trainer.

Extension: The user tries to register and he is incapable of doing that because he doesn’t register with a valid e-mail and the application is unable to make a valid append to the database.

Use case no.2

Use case name: A new admin tries to register a new gym in the application

Level: Register a new gym – user goal

Main actor: An admin who tries to register his gym .

Main success scenario: The user first opens the app, presses the register a new gym button then he adds all the information and data about the gym.

Extension: The admin tries to register the new gym and he writes incorectly some information and a pop-up message appears.

Use case no.3

Use case name: A trainer tries to create a training program

Level: Create a training program – user goal

Main actor: A trainer approved by the admin.

Main success scenario: The trainer registers the new training program, enters all the information and the description for that specific program.

Extension: The trainer selects a date when the gym is actually closed so a pop-up message appears saying that the gym is closed in that time.

Use case diagramDiagram

Description automatically generated

Conceptual Architecture

My application will be a web application made with Java and Spring framework and the front-end part will be made with classic font-end web development languages: HTML and CSS. The app will respect the MVC architecture (Model View Controller). MVC is known as an architectural pattern, which embodies three parts Model, View and Controller, or to be more exact it divides the application into three logical parts: the model part, the view and the controller. Model: Handles data logic. View: It displays the information from the model to the user. Controller: It controls the data flow into a model object and updates the view whenever data changes.

The database from the app will be a MySQL Databse and I will use MySQL Workbench in order to manage it. The database will have X tables: one table for the users, one table for the trainers, one table for the administrator, one table for the trainer appointments, one table for the program appointments and one table for the training programs.

The MVC architectue fits my app because in the model package I will have all the logic, in the view I will have all the front-end files and all the classes related to the connection between the front-end and the actual back-end and in the controller package I will have all the database related logic.

Database Diagram

Diagram

Description automatically generated

Non-Functional Requirements

* Performance – I would like to make my app as fast as possible, so my clients would be happy and the users can use the app without waiting too long for certain processes
* Accessibility across devices – I would like to make my app cross platform (so it can run on desktop and smartphone aswell)
* Scalability – the app will be able to hold and maintain as many users as it can, like couple houndreds at a time
* Security – this requirement will be implemented with data validation
* Managebility – this will be implemented with functions available to the admins, so they can have a big picture and they will be able to access and control everything

Design Constraints

The App will be made using Java and Spring framework and the front-end part will be made using classic CSS and HTML. For the database part and data manuipulation I will use MySQL queries and MySQL Workbench. For the security part I will use data validation and maybe some hashing techniques for the passwords. In order to make the app I will use IntelliJ and for testing I will use Google Chrome.

Package Diagram

Diagram

Description automatically generated

Sequence Diagram

Diagram

Description automatically generated

Activity Diagram

Diagram

Description automatically generated