PTGest

Management System for Personal Trainers and Businesses

Daniel Sousa, n.º 48642, e-mail: a48642@alunos.isel.pt, tel.: 962005244
Pedro Macedo, n.º 49471, e-mail: a49471@alunos.isel.pt, tel.: 918137550

Advisors: Filipe Freitas, e-mail: ffreitas@cc.isel.ipl.pt

March 2024

Introduction

This project aims to develop a comprehensive platform to facilitate efficient integration and management between trainees, personal trainers (*PTs*), and fitness companies, wich may be consultancy companies or companies that have their own gym. The platform will allow the registration of trainees, independent *PTs*, hired *PTs*, and companies, each with their respective permissions and functionalities. The goal of this platform is to provide a more personalized and efficient fitness experience for all involved parties. By digitizing and centralizing various functions, it is expected to improve communication, organization, and ultimately, the quality of service provided.

System Requirements

Functional Requirements

- **User Registration**: Should allow the registration of different types of users, including trainee, individual *PTs*, subcontracted *PTs*, and companies. Each type of user will have different data fields to fill in during registration:
  - **Trainee** can be registered by *PTs* or companies, with an email generated and sent by the system with a method of defining a password;
  - **Hired PT** can only be registered or unregistered through a company that add them into the system and an email is generated and sent by the system, with a

method to set a password, while **independent PT** register without the intervention of another entity;

  - **Companie** register themselves independently of any other entity.

- **Training Session Management**: The system should allow *PTs* to schedule, reschedule, and manage training sessions. It should also be possible to add details such as muscle group, exercises to be performed, training duration, date and time of the session, and the session's location;

- **Session calendar**: The platform will provide a calendar for both the trainee and the *PT*, displaying their scheduled sessions. These sessions could be training sessions based on a specific training plan, supervised training, or evaluation sessions.

- **Feedback and Reviews**: It allows trainees to evaluate the *PTs*, if the session is supervised by them, using a grading system (for example, assigning a grade between 1 and 5). This evaluation can be accessed by both the hiring company and the supervised trainee. Futhermore, each session should include feedback sections for individual exercises and for the session overall. These sections can be filled out by either the trainee or the *PT*, and can be reviewed by them at a later time;

- **Trainee Information Registration**: The platform allows *PTs* to record information related to a trainee such as weight, height, body mass index (*BMI*), percentage of lean and fat mass, and dimensions of muscle groups;

- **Progress Reports**: The *PTs* have the possibility to create progress reports of a trainee, which can be private (only the *PT* can see) or public (both can see);

- **Email Notifications**: The system should be able to send emails to notify sessions and password changes;

- **Creation of Custom Exercises**: *PTs* should be able to create custom exercises and add exercises to favorites for quick access, with the possibility of attaching demonstration videos.

Non-Functional Requirements

- **Cloud Deployment**: The system should be designed for deployment in a cloud environment, allowing for flexibility, scalability, and reliability. It should support seamless deployment processes, automatic updates, and efficient resource management;

- **Usability**: It should be easy to use and intuitive, with a clear and simple user interface;

- **Compatibility**: The system should be compatible with various devices and browsers;

- **Maintainability**: The system should be easy to modify and update, allowing for the addition of new functionalities and the correction of errors efficiently;

- **Scalability**: Should be able to handle an increase in the number of users or data volume without compromising performance;

- **Portability**: The application should be able to function on different operating systems and devices, being responsive.

Optional Features

- **Membership Management**: The system should allow the management of membership fees and values per class, both for companies and for individual *PTs*;

- **Implementation of a PWA (Progressive Web App)**: This would allow the platform to be installed and function as a native application on various devices and operating systems;

- **Media Upload**: The platform could allow for the upload of PDF documents, images, and videos, providing greater interactivity and understanding of exercises and training plans;

- **Invoice Generation**: The platform could generate invoices for fees and payments, facilitating financial management for users;

- **Evolution Charts / Progress History**: The platform could present evolution charts to visualize trainees progress over time;

- **Integration with Google Maps**: This functionality could allow for the mapping of *PT's* by gym(s) they can attend or franchise gyms;

- **Video Call Sessions**: The platform could allow for training sessions via video call, providing greater flexibility for users.

Technologies

In this project, the Spring framework [1] and the Kotlin programming language [2] will be utilized for managing application data. The construction of web pages will be accomplished using Vue [3] with Vite [4]. TypeScript [5] will also be incorporated into the project. For the persistent storage of application data, PostgreSQL [6] will be employed.

Risks

- **Data Security**: As the system stores personal information of users, there is a risk of data breach. It is crucial to ensure that all information is stored securely and encrypted;

- **Compatibility and Portability**: The system should be compatible with various devices and browsers. If not, there may be access issues for some users;

- **Scalability**: If the number of users or the volume of data increases significantly, the system may have performance issues.

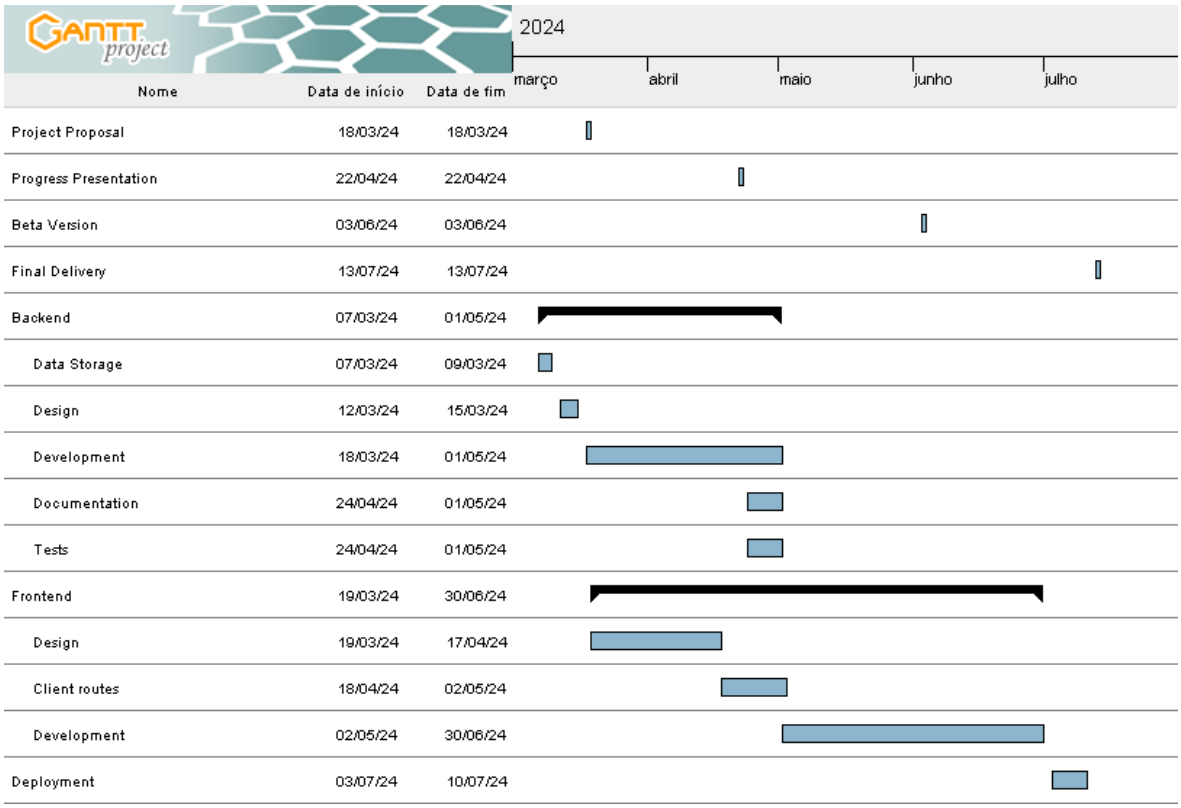

Figure 1: Gantt Chart Project Plan

References

[1] Spring. `https://spring.io/`. Accessed: 2024-03-11.

[2] Kotlin. `https://kotlinlang.org/`. Accessed: 2024-03-11.

[3] Vue.js. `https://vuejs.org/`. Accessed: 2024-03-11.

[4] Vite. `https://vitejs.dev/`. Accessed: 2024-03-11.

[5] Typescript. `https://www.typescriptlang.org/`. Accessed: 2024-03-11.

[6] Postgresql. `https://www.postgresql.org/`. Accessed: 2024-03-11.