Software Requirements Specification

for

FitHub

Prepared by:

Arwa Ibrahim, Bosy Ayman, Farha Ahmed,

Habiba Arafa, Nourhan Deif

Zewail City

9 November 2024

1. Introduction
1.1 Purpose

The document details the features and constraints of a FitHub, which improves lifestyle through custom exercise and nutrition plans with the support of certified coaches via an interactive form. This guide helps developers, testers, and everyone involved in FitHub Development understand what this platform will be all about.

1.2 Scope

The FitHub program is aimed at helping people improve their health and fitness. It provides exercise plans, recipe catalogs, private coaching, forums for community interaction, and progress tracking as well as a conversation hub for community input.

The system will have the capability for the following:-

- User (Trainee and Coach): Sign-up, User Profile Management, and update personal details
- Training and Nutrition: Providing general exercise programs, and adding recipes the trainee wants and allowing coaches to tailor the program.
- Community Forum: Forums for users to post and comment, where people come together to share their experiences and advice.
- Messaging: Enabling the trainees to communicate directly with their assigned coach for personalized advice and support.

1.3 Definitions, Acronyms, and Abbreviations

- **API**: Application Programming Interface
- **UAT**: User Acceptance Testing
- **FitHub (Django backend): ORM:** Object-Relational Mapping used in FitHub for mapping the data models of the platform to SQLite database.
- **React**: JavaScript library for building user interfaces
- **SQLite**: Lightweight relational database system
- **UI:** User Interface: The visual layout and design of FitHub (how users interact with it)
- **UX:** user experience :(how easy and satisfying it is for the users to interact with our FitHub platform)
- **Trainee:** A FitHub user who is interested in lifestyle improvements through exercise and eating plans, possibly communicating with a coach that provides personalized support.

- **Coach:** A qualified expert who can share their professional advice, help, and encourage you. in your workouts on FitHub

1.4 References

- *Django documentation | Django documentation*. (n.d.).

  Django Project. https://docs.djangoproject.com/en/5.1/

- *What is REST?: REST API Tutorial*. (2023, December 11).

  REST API Tutorial. https://restfulapi.net/

- https://reactjs.org/docs/

- *SQLite Documentation*. (n.d.). https://www.sqlite.org/docs.html

---

2. System Overview

2.1 Product Perspective

FitHub is a standalone website that consists of:

- **Frontend**: Use React for interacting with the system.
- **Backend**: Developed using Django for controlling business processes and interacting with SQLite database.
- **Database**: Use SQLite to store data, which includes content created by users and records managed by the system.

2.2 Product Functions

The key functions of the Fithub include:

1. **Signup/Login**: Enable users (both coaches, trainees) to create their profiles with details on their work, and to log in securely
2. **Profile Management**: Gives users the ability to access and modify their profiles, with the possibility of changing personal information
3. **Password Management**: Provides an opportunity for all the users to create, change their passwords
4. **Forums (Posts & Comments)**: Allows users to create new posts, take part in discussions, and leave comments in the community forum offered to them.

5. **Notifications**: Delivers alerts for all significant events including messages , posts, exercise reminders.
6. **Coach Verification by Admin**: Utilizes an administrator approval to confirm the details of coach accounts
7. **Plan Management**: assign appropriate fitness schedules according to their aims and allows the coaches to modify and create such schedules if trainees assign with the coach .
8. **Progress Tracking**: showing the evolution of their performance in various workouts and nutrition
9. **Recipes Catalogue**: Provides a collection of Food Recipes categorized by Nutritional value, Goal and Fully Available for all users .
10. **Direct Messages (Trainee & Assigned Coach)**: Enables private messaging between trainees and the coach assigned to them for better assistance and feedback.
11. **Exercises Catalogue**: Provides an catalogue of workouts complete with information on each, thus enabling trainees to familiarize with different workouts and add them into plan

2.3 User Classes and Characteristics

Different types of users who will interact with the system include:

- **Admin**: Approve coaches to make sure form validation of their data
- **Coaches**: Create and modify workouts and plans, communicate with the trainees and track their performance and create posts
- **Trainees**:Choose workout programs, record the degree of completion, and correspond with coaches if they assigned and create posts

2.4 Operating Environment

The system will operate in the following environments:

- **Client Side**: Runs on web browsers like Chrome, Firefox, and Safari.
- **Server Side**: hosted on a web server that utilizes Django as backend and SQLite as the database engine.

---

3. Functional Requirements

3.1 Use Cases

3.1.1 Use case - User Registration

3.1.1.1 Textual use case

- **Description**: It describes the process in which new users (trainees or a coaches) create an account in the FitHub system by filling in information such as name, age, and weight among others. The user will have to select one of two available roles (trainee or a coach) and will be allowed to use only the respective functionalities upon successful registration. o **Actors**: Trainee, Coach, System.

- **Preconditions**:

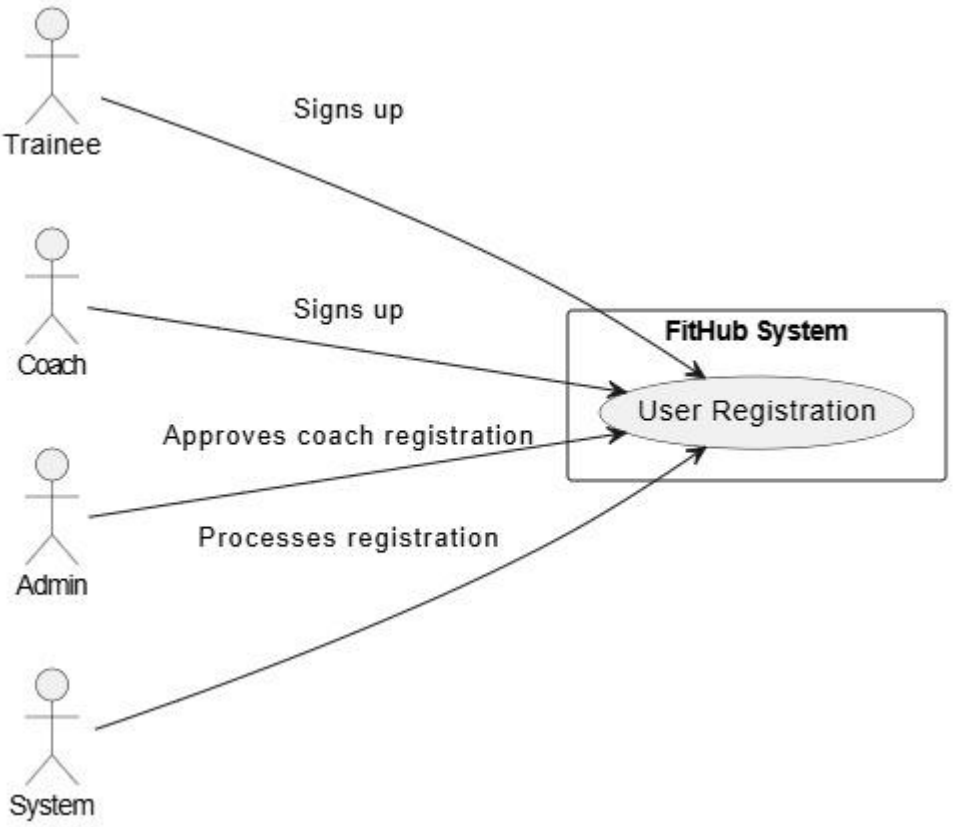
  - The user is required to have internet connectivity and go to FitHub webpage.
  - The user has to be a new user who has not registered in the system.

- **Postconditions**:

  - The system ensures that the user is registered and can go to the personal dashboard.
  - For trainees, a new basic training program is provided, whereas coaches wait for admin's approval.

- **Steps**:

  1. The user navigates to the FitHub main page.
  2. The user chooses the option "Sign Up".
  3. The user has to indicate if he/she is a 'Trainee' or a 'Coach'.
  4. The user provides the requested information (for example: their name, age, weight, role).
  5. In the event, the user opts for 'Coach', the registration is subjected to admin approval.
  6. The user presses "Submit".
  7. The system captures the user details and takes the user to their corresponding dashboard (trainee or coach).



3.1.2 Use case - Trainee Selecting a Coach

3.1.2.1 Textual use case

- **Description**:

  - This use case is concerned with how the trainee is able to select and interact with a coach to develop or enhance their workout and diet plans.

  - The trainee seeks a coach who specializes in their field, can chat privately, and have the coach adjust the plan for them.

- **Actors**: Trainee, Coach, System.
- **Preconditions**:

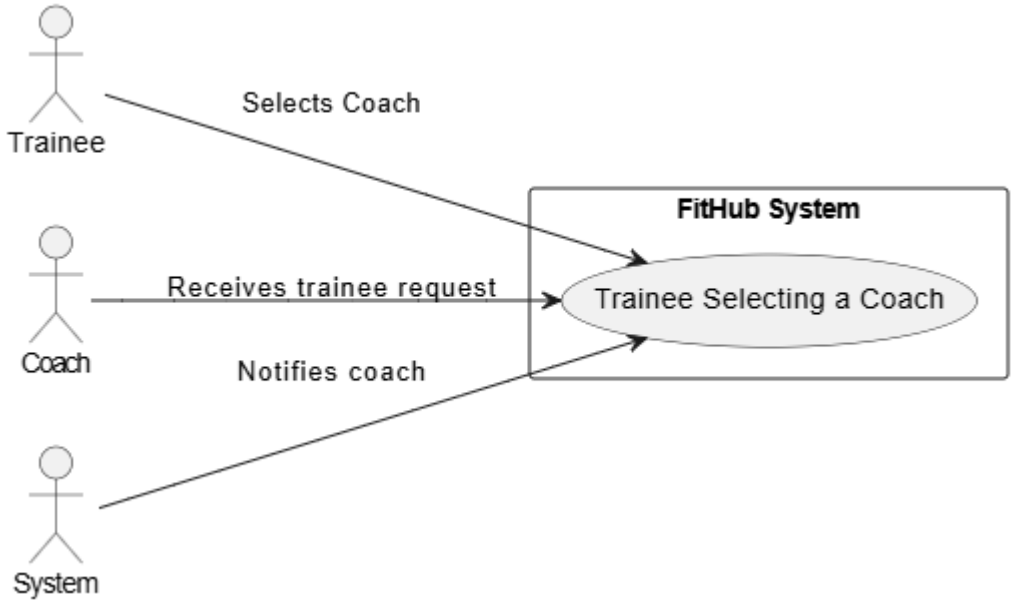
  - The system enforces the trainee to log in before accessing the application
  - The coach is supposed to be vetted by the admin and makes it to the list of available coaches.

- **Postconditions**:

  - The trainee selects a coach from the list of available ones and is able to chat and see his/her profile information.
  - The system alerts the coach who can then start modifying the plan for the trainee.

o **Steps**:

1. The trainee visits the page titled “Coach Catalog”.
2. The trainee looks for a coach based on the specific area of specialization or preferences.
3. The trainee selects the coach from a list of available coaches.
4. The system makes a notification to the designated coach.
5. The coach gets the notification and accepts.
6. The trainee and the coach share their messages and the coach starts modifying the trainee’s program upon approval.



3.1.3 Use Case - Posting in the Community Forum

3.1.3.1 Textual use case

o **Description**: This use case outlines the process of posting a message in the community forum whereTrailsmen and Patrol leaders can give feedbacks, also ask and give tips on various issues. o **Actors**: Trainee, Coach, System. o **Preconditions**:

- The user must be logged in.
- The user must have access to the community forum section.

- **Postconditions**:
  - The post is visible to all users in the forum. - Other users can comment on the post.

- **Steps**:

1. The user goes to the ‘Community Forum’ page.
2. The created post button is clicked by the user.
3. The user enters a post (for example: his experience, a question, recommendation, etc.).
4. The user chooses a category or a topic for the post.
5. The user presses the button “Submit”.
6. This post is shown by the system in the appropriate forum section for other users to see it.

3.1.4 Use Case - Monitoring Progress of a Trainee

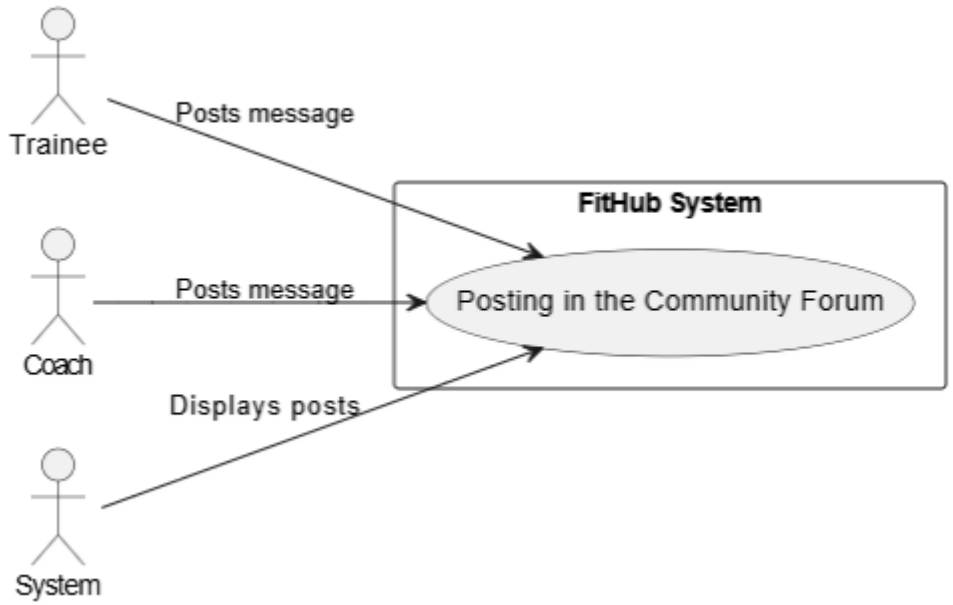
3.1.4.1 Textual use case

- **Description**:
  - This use case examines how the trainee logs both exercise and diet achievements into the system while it builds and updates charts to show progress over time.

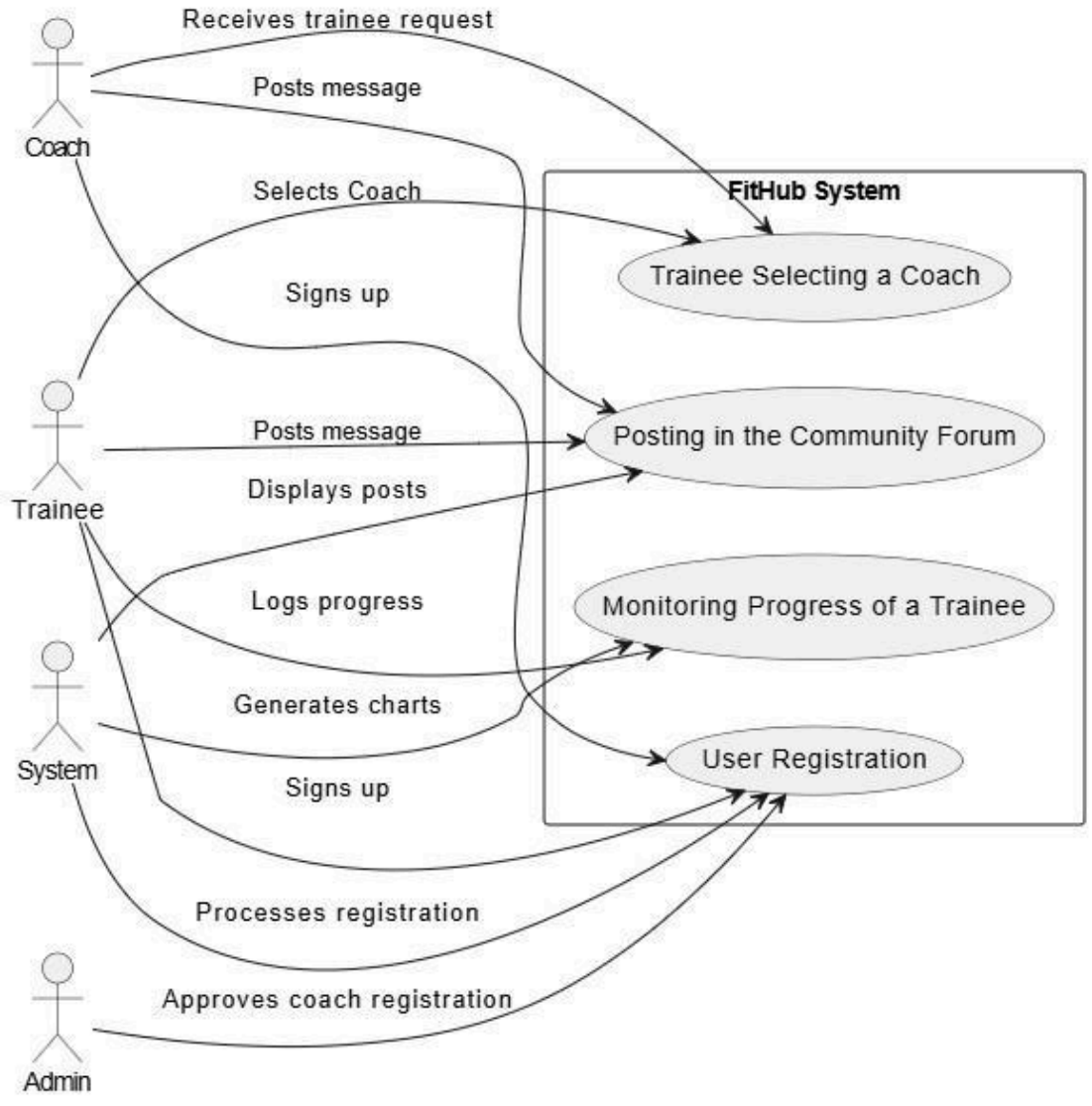
- **Actors**: Trainee, System.
- **Preconditions**:
  - The trainee should be signed in.
  - The trainee should have access to the training plan assigned to him/her
- **Postconditions**:
  - The system records the progress of the trainee.
  - Graphs are refreshed and displayed to the trainee

- **Steps**:

1. The trainee signs in and clicks on the “Progress Tracking” page.
2. The trainee selects to track either exercise or nutrition.
3. The trainee Enters statistics (i.e. completed exercises, meals eaten, etc.).
4. The system records the entry and generates a chart illustrating the progress of the trainee.
5. The trainee can check the progress updated and how it stands against the set target.



Combined diagrams:



4. Non-Functional Requirements

4.1 Usability:

User-Friendly Interface: The system should provide a clear interface as well as provide interaction management that makes it easy for every user regardless of the user being a trainee, a coach

Simplicity: The design of the interface should require the least effort from the user to reach all functions

4.2 Maintainability:

The project will be well-organized and commented for easy readability and updates and clear explanations for understanding and will be designed in separate files to allow updating without impact whole system

4.3 Availability:

The system should be available , and no other downtime should be allowed apart from the periodic maintenance.

4.4 Compatibility:

The system should be compatible with all web browsers (Microsoft Edge, Google Chrome, Firefox, Safari, etc)

---

5. External Interface Requirements

5.1 User Interfaces

The user interface of the FitHub platform must be friendly and easy to use and should comprise the following major elements:

1. **Home Screen:** Displays posts and enables easy navigation to important areas including personal profile ,recipes,exercises,messages
2. **Profile Management:** Allows the clients to make changes in their personal detail (name, age, weight etc) and make modifications in the account settings such as the password.
3. **Forums:** Provide a space for users to interact by allowing them to post topics and comments to posts on the forum. Trainees and coaches can give tips and share their stories.
4. **Chat:** A feature available for trainees to send messages to their respective coaches in a safe environment.

5.2 API Interfaces

The FitHub system will provide REST APIs for the following functional areas:

1. **User Management:** Encompasses users registering, logging in, updating their profiles and managing passwords.
2. **Progress Tracking:** Includes logging and retrieving of progress data, including but not limited to exercise logs, food intake logs and progress contour charts.
3. **Chat:** Exchange of messages between trainees and coaches is facilitated.

Data exchange with all APIs will be done using the REST Architecture and will make use of the standard HTTP methods (GET, POST, PUT, DELETE). The system will also add support for authentication and authorization by using JWT (JSON Web Tokens) functionalities

---

6. Conclusion

Fithub gives users the ability to take charge of their fitness and nutrition in a healthy manner. It presents functional requirements, high level architecture, and users of the system interact with each other and with the system to give guidance on aspects such as user signing up, creating and updating user profile, providing users with a coach verified guidance, forums, and tracking progress. Non-functional aspects along with the functional ones require that FitHub is user-friendly, maintainable, compatible, and available.This document will serve as template for the development of Fithub by the team.