IT314 Software Engineering
Lab 6 - Domain Analysis Model
Project Name: **AthleteTalk**
Group 18

Project members :

Name	ID
PATEL KHUSHKUMAR GIRISHBHAI	202001127
ACHARYA HARSH MAHAVIRBHAI	202001131
UPADHYAY DHVANI VIPULKUMAR	202001136
PORIYA NEEL DHARMESHBHAI	202001139
PATEL PARTHKUMAR KANTIBHAI	202001147
VEGAD RAJ MAHESHBHAI	202001160
CHIRAG CHAVDA	202001164
SUVAGIYA ABHAY JAGDISHBHAI	202001171
VIRADIYA ABHAY	202001174

● Create the domain analysis model for your course project.

Our application is about helping athletes manage stress, depression, anxiety and other mental problems via suggesting plans from authorized coaches and trainers, and providing video based exercises for different days of the ongoing plans.

Boundary Objects:

Login/Signup Interface: The page on which users log in/signup to access the system. Once the user provides the credentials, the system will check that the user is authenticated using the email and password that the user provided.

Web Interface: Here the user interacts with the system, once the user is successfully logged in. Here, there are many options for users to interact with the system, some of them are listed below:

- Accessing Dashboard
- Browse exercise videos
- Community chat
- Browsing and taking plans

Entities objects:

Athletes: Users who utilize the system to access recommended plans and videos and connect with other athletes and coaches through the community chat feature.

Coaches: Users having administrative access to the system and are able to create and assign recommended plans and videos for athletes based on their needs.

Administrators: Users having absolute access to the system and are able to manage user accounts, data, and system settings.

Videos: These are the videos that will help users to perform the mental health exercises.

Plans: These are the collection of videos for various days which the coaches will make.

Controller Objects:

User management controller: This controller object manages user authentication and authorization, as well as user account creation and deletion.

Messaging controller: This controller object manages the messaging service, including sending and receiving messages, moderating chats and forums, and managing group discussions.

Wellness plan controller: This controller object manages the creation, modification, and deletion of wellness plans, as well as the questions and prompts within them.

Wellness video controller: This controller object manages the distribution and accessibility of wellness videos from sports behavioral coaches.

Domain Concepts:

Mental Health: The idea of mental health, including stress, anxiety, and depression, and how to manage them with suggested plans and community support.

Recommended Plans: The treatment programmes which may include breathing exercises, meditation, cognitive-behavioral therapy, and relaxation methods, that coaches and therapists suggest to athletes depending on their requirements.

Community Support: The ability for athletes to communicate with one another and exchange advice and support through the community chat function.

User Data Management: The management of how user data is stored, secured, and backed up, including personal data, progress reports, and activity logs.

User Interface:

Login Screen: The page on which users log in to access the system.

Dashboard: The display where athletes may see their recommendations, progress, and community chat.

Plan Selection Screen: The page where athletes choose a suggested plan to follow.

Plan Execution Screen: The screen where athletes actually carry out the instructions and tasks specified in the suggested plan.

Community Chat Screen: The screen where athletes communicate and exchange thoughts.

Browse Videos Screen: All the available training videos, which have been uploaded by different coaches, will be displayed on this screen, along with their categories and

descriptions. We will offer on the spot video streaming for any videos users want to view.

Plan Creation Screen: The coaches may design the plans on this screen by choosing various video workouts for the different days of the plans, and they can then validate the plan's specifics on the summary page.

User needs:

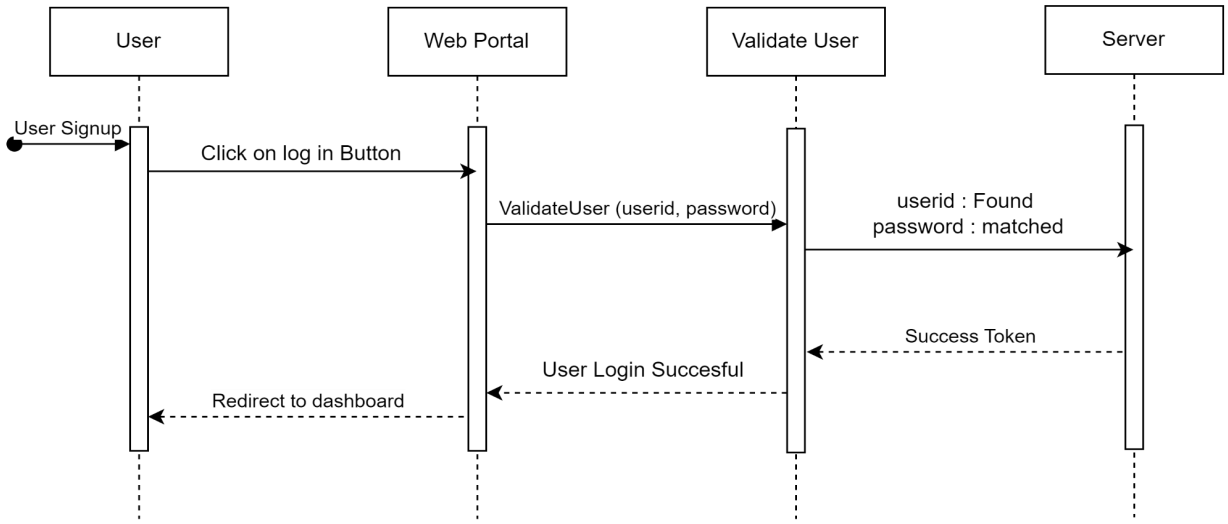
For the athletes:
Students should have access to the materials (videos and plans) offered on the site by their preferred trainers. They should have no trouble using the Community discussion area. The athlete should be able to update their well-wishers on their present development. Maximum privacy and secrecy must be maintained, and the athlete must feel at ease using the website.

For the coaches:
The coaches should be qualified and have significant knowledge of mental health. They must be able to communicate well and be familiar with the culture of the sports world today. It should be simple for the coaches to distribute their information (via videos, lesson plans, or talks in the Community discussion area).

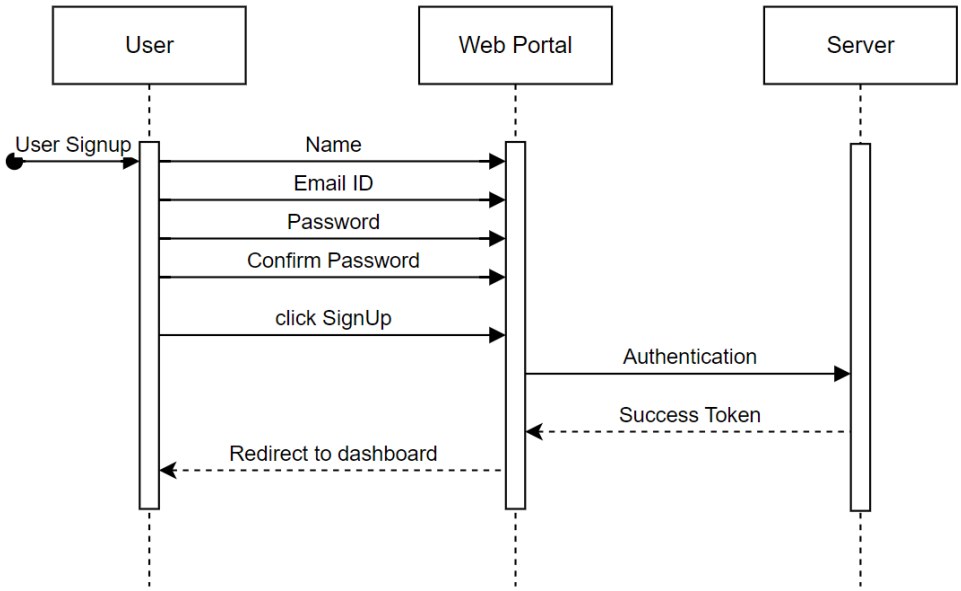
For the administrators:
The admin should be able to manage the data, ensuring it is up-to-date and accurate. They should be able to manage user accounts, including signing up, logging in and account deletion. They should also be able to monitor user activity. They should be able to ensure security of the data stored.

Sequence Diagrams:

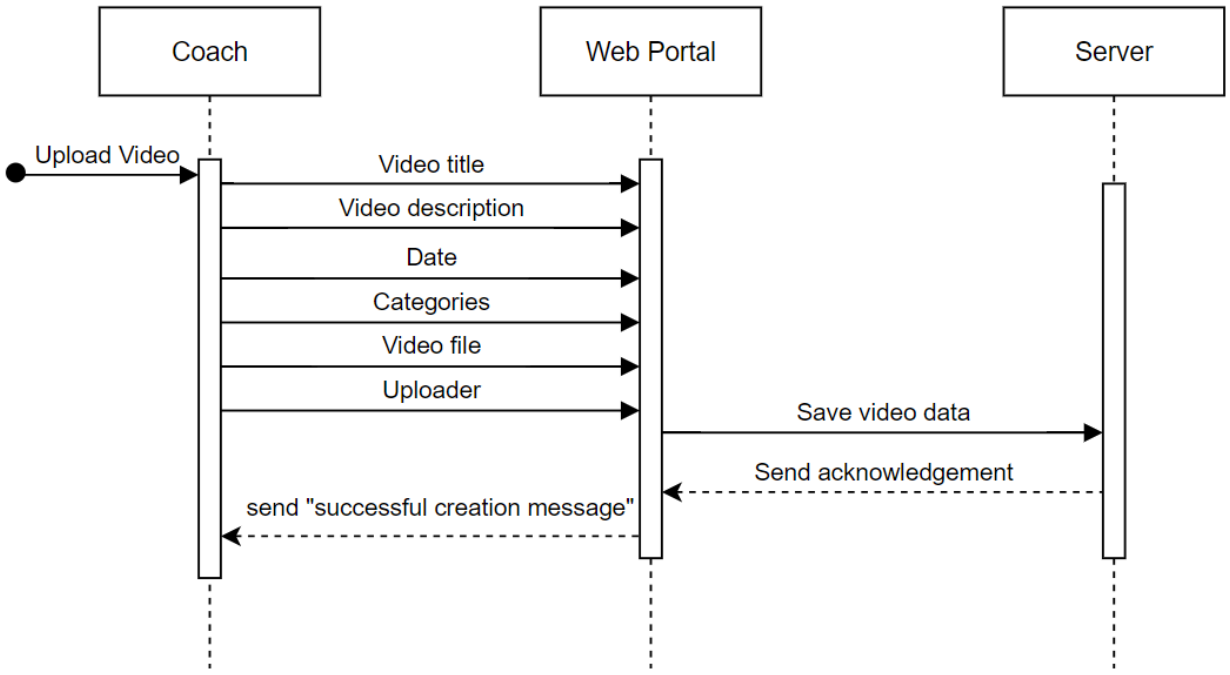
User login:



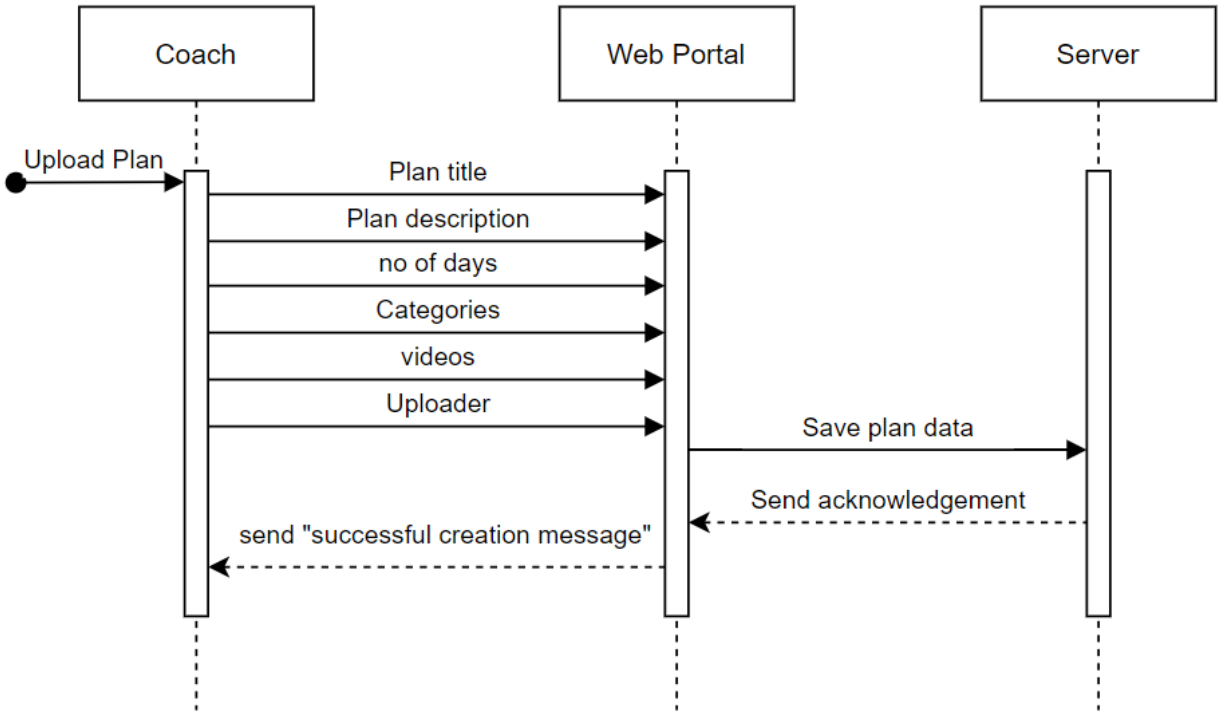
User signup:



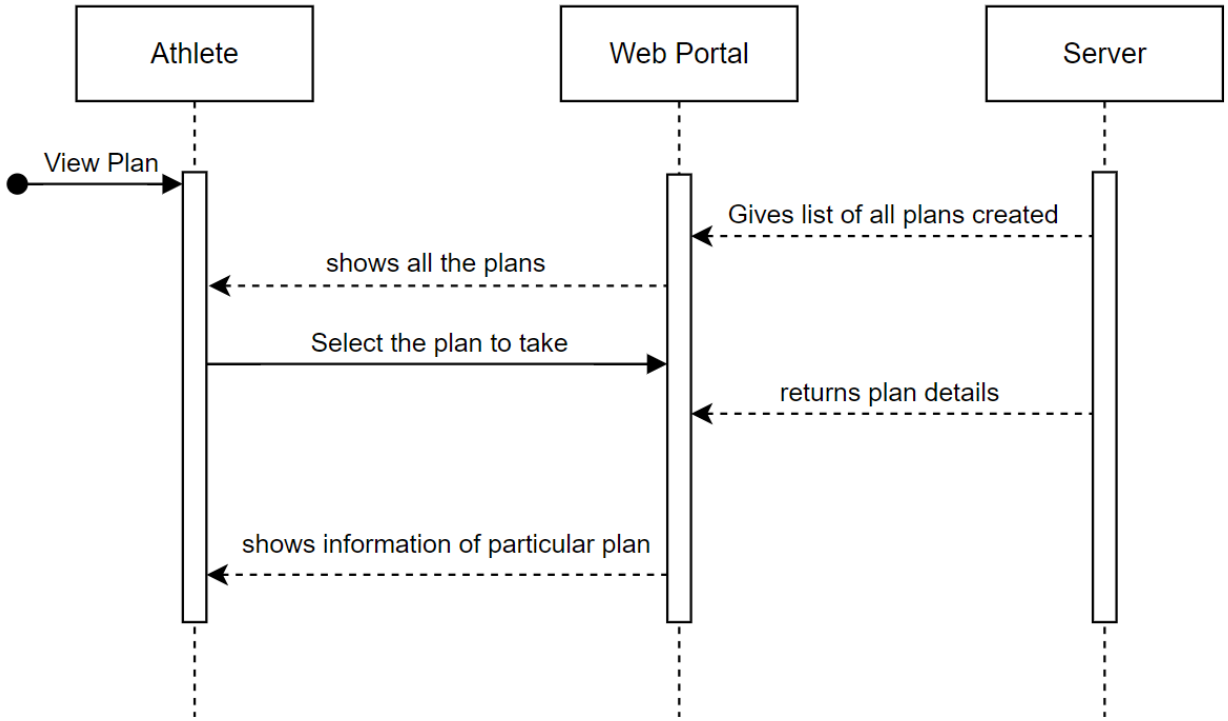
Upload Video:



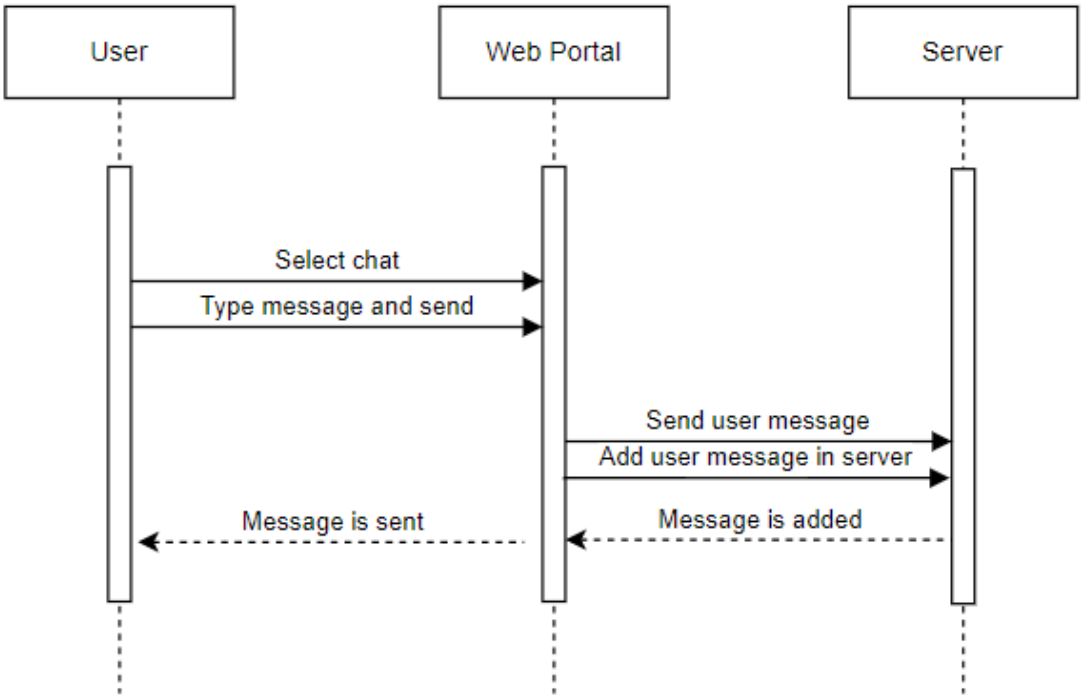
Upload Plan:



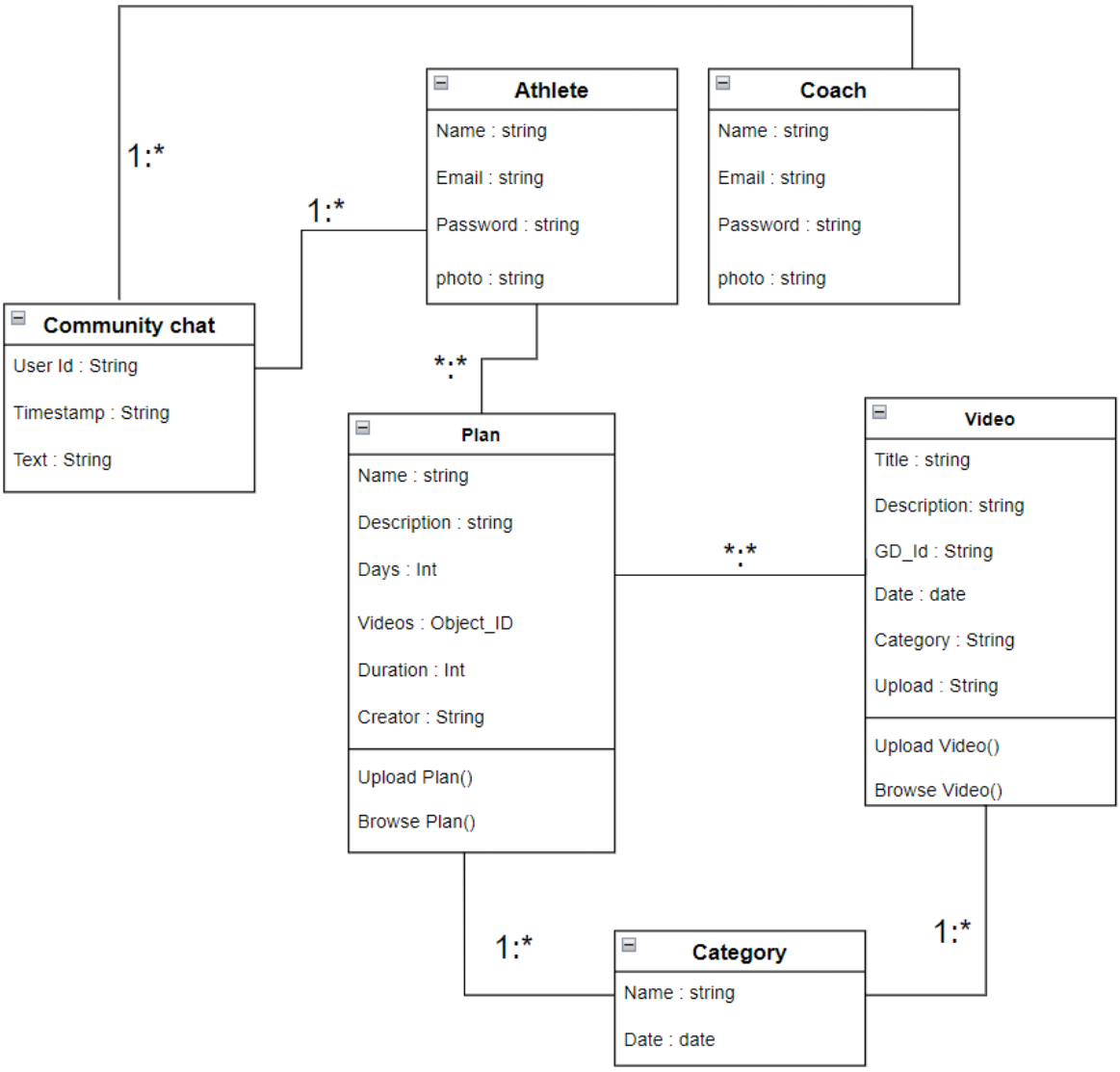
View Plan



Community Chatting:



Class Diagram:



Identify the design goals.

1. **User-friendly interface**: The application should have a simple and intuitive user interface that makes it easy for athletes to navigate and use.

2. **Discussion Section** : The application should allow athletes and other users to send and receive messages from others in a discussion panel.

3. **Notification system**: The application should have a notification system that alerts users of new messages, reminders of their plans or updates to their profile.

4. **Profile customization**: The application should allow athletes to customize their profile with information such as their bio, photos, and sports stats.

5. **Privacy and security**: The application should prioritize user privacy and security, with features such as two-factor authentication, encryption, and user control over their data.

6. **Exercise tracking and Health report :** Score will be calculated based on how efficiently users have followed their plans and health report will be generated periodically.

- **Create a high level system design.**

  - Choose architecture, identify subsystems (you can use package diagrams)

  Architecture:

  Framework for developing web-application:

  Next.js is a flexible **full stack framework** that gives you building blocks to create fast **web applications**.

  Next.js, on the inside, uses **React** as a front-end framework.

  React.js:

  ReactJS, is an open-source **JavaScript library** used for building user interfaces (UIs) and UI components. It was developed by **Facebook** and is now maintained by Facebook and a community of individual developers and companies.

  Reasons:

  React components: React allows code reusability by using components which can be used multiple times by referencing the same component.

  Fast Rendering: React uses a virtual DOM, which allows for faster rendering and better performance.

  Large Community Support: Being developed and maintained by Facebook, React has a strong backing and is constantly being improved and updated.

JSX: JSX is a syntax extension for JavaScript that allows developers to write HTML-like code directly in their JavaScript files. This makes it easy to create complex UIs without having to switch between different languages or file types.

Hooks: React introduced Hooks in newer versions, which allows developers to use state and other React features in functional components, making them easier to write and more reusable.

Next.js uses express.js inside its backend stack.

Express.js:

ExpressJS, or simply Express, is a popular open-source web application framework for Node.js. It provides a set of features and tools for building web and mobile applications, including robust routing, middleware support, and a flexible plugin architecture.

Reasons:
Minimalist and easy of use: Express is a straightforward and adaptable framework that offers programmers a selection of fundamental features and resources. This enables developers to create unique solutions that are suited to their unique demands.

Robust routing: Express provides a robust routing system that allows developers to handle different HTTP requests and responses easily.

Scalability: Express is designed to be scalable and can handle a large number of requests simultaneously.

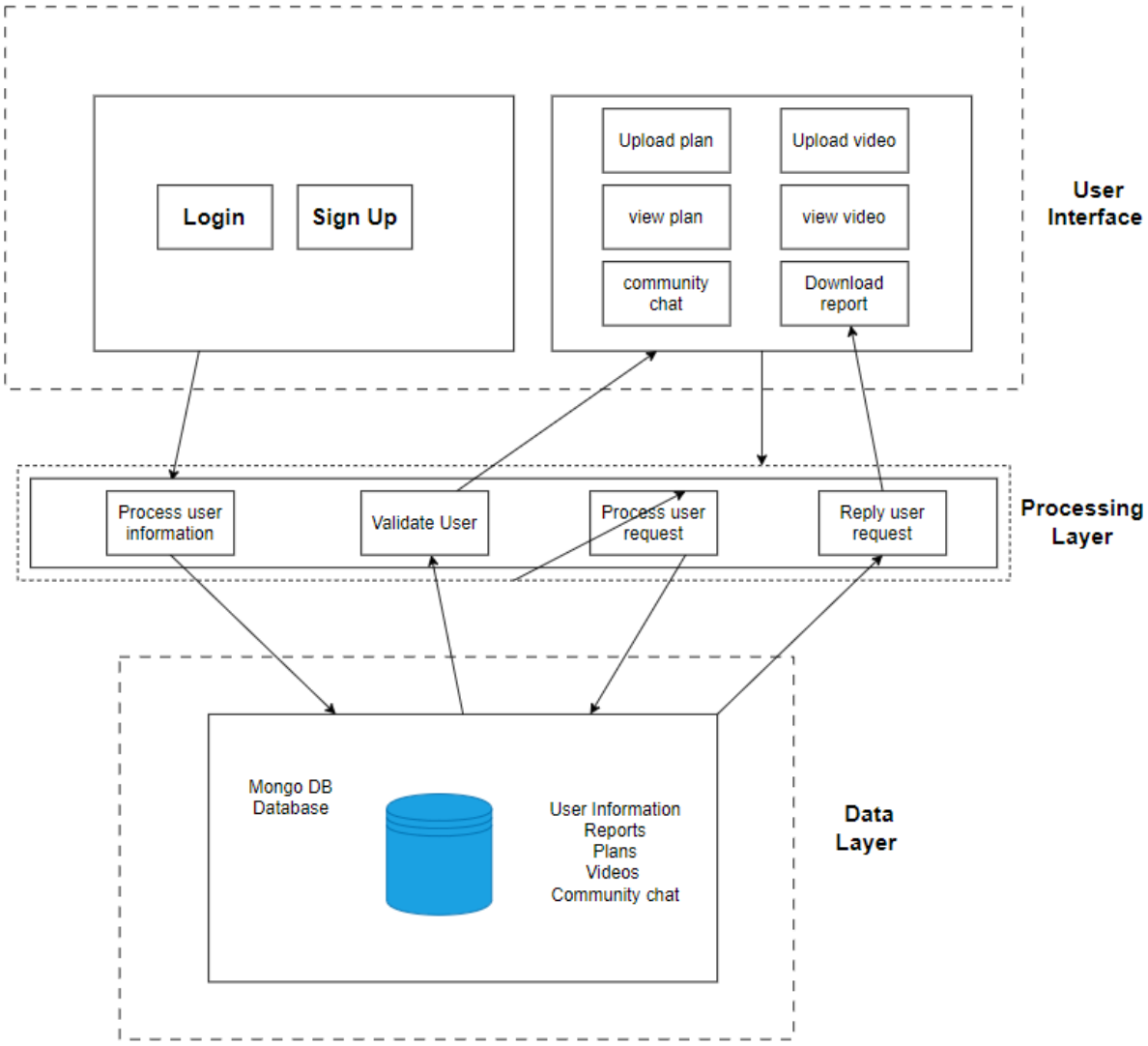
Compatible with other Node.js modules: Express is compatible with other Node.js modules, making it easy to integrate with other tools and technologies.

Middleware functions: Express provides middleware functions, which are functions that can access the request and response objects and can modify them or perform additional operations before passing the control to the next middleware function.

Error handling: Express provides a robust error-handling system that allows developers to catch and handle errors that occur during the execution of middleware functions or routes.

The system architecture will follow a three-tier architecture model, consisting of the following layers:

High Level Design



Architecture: Layered Architecture

- **Presentation Layer:** This layer will be built using Next.js and React.js and will be responsible for the user interface and user interaction. It will enable users to interact with the system through web browsers.

- **Application Layer:** This layer will manage the application's business logic and process user requests. It will be implemented using Express.js, a Node.js framework, and will interact with the database.

- **Data Layer:** This layer will be responsible for storing and retrieving data from the database. It will be implemented using MongoDB, a NoSQL database.

Subsystems:

User Management System: This subsystem handles user registration, authentication, and authorization. It also manages user profiles and preferences.

Authentication Subsystem: This subsystem is responsible for handling user authentication and authorization. It includes features such as user registration, login, password management, and role-based access control.

Track Management System: This subsystem manages the creation, management, and promotion of Tracks(challenges). It allows organizers to create and manage Track pages, manage submissions, and judge submissions. It also allows participants to view and join tracks, view challenges, and submit their solutions.

Analytics and Reporting System: This subsystem provides data analytics and reporting tools for Track organizers. It tracks and reports on submission quality, and other important metrics.

Infrastructure and Security System: This subsystem ensures the website's reliability, scalability, and security. It manages hosting, backup, and disaster recovery.

Reminder and Notification Subsystem: This subsystem is responsible for sending reminders and notifications to users regarding their health data. It includes features such as notification scheduling, notification templates, and notification delivery.