Sri Lanka Institute of Information Technology

Programming Applications and Frameworks (IT3030)

Continuous Assignment – 2024, Semester 1

Initial Document

GROUP ID: JUN_WE_34

Gamlath W.A.V.K | IT21211928

Pathirana K.P.V | IT21244766

Bandara A.M.B.S | IT21229398

Hasanka G.S | IT21233876

Contents

Contents

Project Description

Our fitness social media platform enables users to share photos and videos showcasing their fitness activities and healthy meals, with a limit of three uploads per post. Users can also provide descriptions for each post. Additionally, they can share real-time workout status updates, inputting metrics like distance run or weights lifted using predefined templates. The platform allows users to exchange workout and meal plans, with customizable templates to suit individual preferences and goals, including categorization based on dietary choices.

1. **Post Management:**
    - Our platform offers users full control over their posts, allowing them to edit or delete them as needed. Whether it's updating a workout progress photo or refining a meal plan description, users can easily manage their content to ensure it accurately reflects their fitness journey and goals.
2. **User Management:**
    - With user-centricity at the forefront, our platform provides seamless user management features. Users can create and customize their profiles, showcasing their fitness-related activities and achievements. Additionally, the option to log in with existing social media accounts streamlines the onboarding process, enhancing user convenience and accessibility.
3. **Comment & Like Management:**
    - Interaction is key to fostering a supportive fitness community, which is why our platform prioritizes robust comment and like management. Users can engage with each other's posts through likes and comments, with the flexibility to edit or delete their own comments. Post owners retain control over their content, including the ability to manage and moderate comments on their posts.
4. **Community Group Management:**
    - Building on the foundation of connectivity and camaraderie, our platform facilitates community group management. Users can join or create groups based on shared fitness interests, allowing for deeper engagement and collaboration within specialized communities. Whether it's a running club or a nutrition-focused group, users can connect with like-minded individuals to share insights, tips, and motivation.

The Functional requirements for the REST API and the client web application separately.

Rest API

• Authorization and Authentication

Users should be able to authorize and authenticate themselves using the API, with support for a variety of authentication techniques like OAuth.

• HATEOAS

The API should be compatible with HATEOAS (Hypermedia as the Engine of Application State), which calls for the response to contain links to pertinent sites to help clients find and use the API.

• Resource Manipulation

The API should be able to create, read, update, and delete resources as per the HTTP verbs – POST, GET, PUT, and DELETE respectively.

• Cacheable

The API should be able to support caching to improve performance and reduce network traffic.

Client Web Application

• User Registration and Authentication

Let users to register for accounts and administer them using their email, phone, or social network credentials. To ensure security, authentication procedures should be in place.

• Commenting

Users should be able to like or dislike comments as well as leave feedback on comments.

• Follow and Unfollow

Users have the option to follow other users and their meal plans on our platform. This allows for easy updates on activities and posts, as well as access to diverse and personalized meal plans for inspiration and guidance.

• Social Sharing

Allow users to share their reviews and photos on other social media platforms.

The Non-functional requirements for the REST API and the client web application.

REST API

• Performance

Even under heavy demand, the API should be built to be extremely performant, with quick response times and little latency.

• Scalability

The API should be scalable to handle a large number of requests and users, without compromising performance or reliability.

• Security

The API need to be created with security in mind, with safeguards to prevent unauthorized access, guard sensitive data, and guarantee the confidentiality, integrity, and accessibility of the API.

• Maintainability

To make updates, bug repairs, and feature additions simple, the API should be easily maintained with a distinct separation of responsibilities, modularity, and high-quality code.

Client Web Application

• Usability

Even non-technical users should be able to easily utilize and navigate the site.

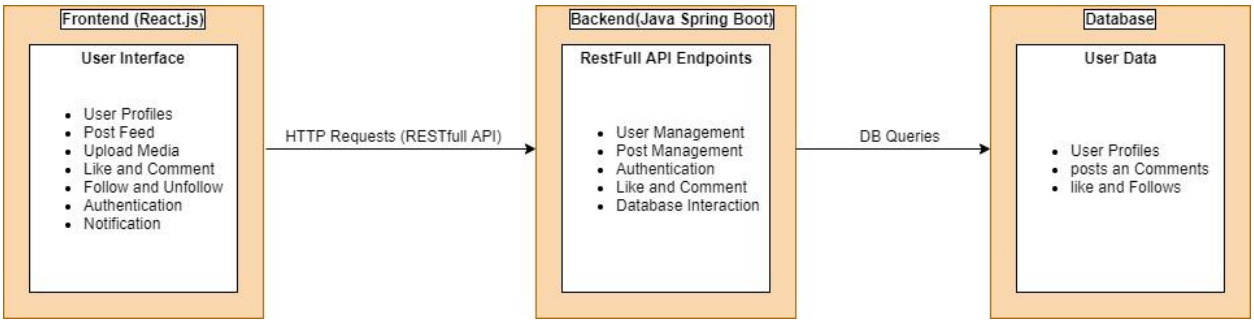
• Performance

Even during times of high demand, the platform should respond quickly and load pages quickly.

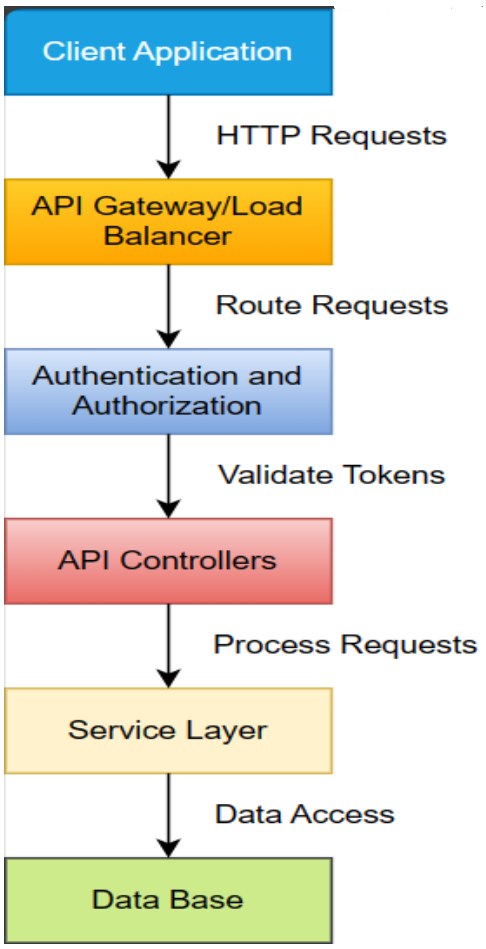
• Reliability

There should be little to no downtime and constant availability of the platform.

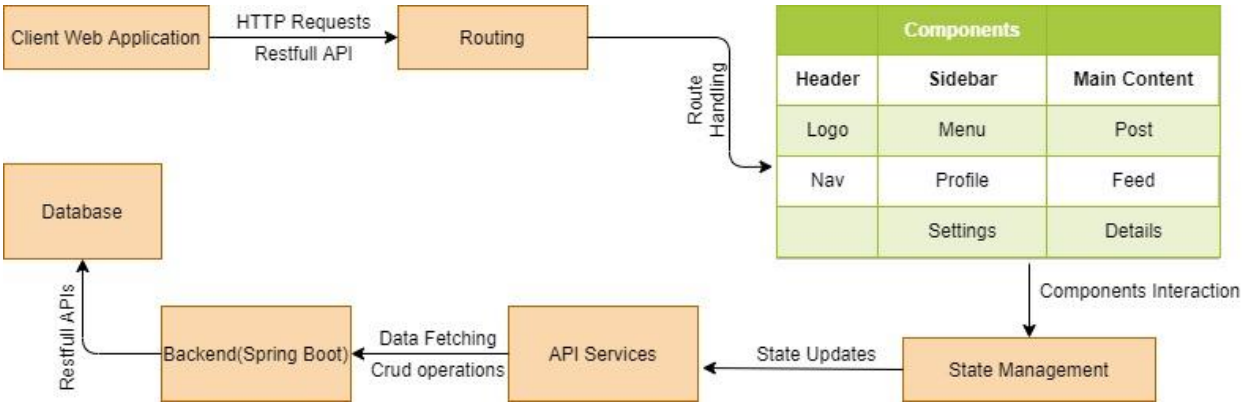
An overall architecture diagram for the entire system you are contracted to design and implement



A detailed architecture diagram(s) for the REST API



A detailed architecture diagram(s) for the client web application. You should decide upon a suitable front-end architecture.



References

What is RESTful API

https://aws.amazon.com/what-is/restful-api/#:~:text=RESTful%20API%20is%20an%20interface,applications%20to%20perform%20various%20tasks.

UML Diagrams

https://miro.com/diagramming/what-is-a-uml-diagram/