Sri Lanka Institute of Information Technology

Programming Applications and Frameworks (IT3030)

Continuous Assignment – 2024, Semester 1

Initial Document

GROUP ID: DS_08

Group Details:

Name	**IT Number**
Mapa M.M.N.D.	IT 21269516
Deshapriya H.B.M.T	IT 21291432
Samarakoon S.M.R.S.B	IT 21276200
Premathilake M.N.	IT 21347894

Table of Contents

1. Project Description

Gymly: Your Premier Fitness Platform

Gymly presents itself as the best option for people who want to keep up their exercise routine but frequently struggle with desire to go to the gym. Customised just for fitness enthusiasts, Gymly provides a platform for social media web applications.

Gymly is a platform that is meant to transform your fitness journey and act as a source of encouragement and inspiration for people who want to lead healthier lives. Gymly strives to encourage passion and dedication among its customers by utilising innovative technology and ensuring consistent participation in physical activities.

Gymly is unique because of its active community of trainers and fitness influencers that provide insightful advice, exercise plans, status updates, and lifestyle advice. With the help of Gymly, individuals may set up individualised accounts to visually represent their fitness journey through posts, videos, and eye-catching images.

Gymly makes it easy for users to participate and interact by letting them like, comment on, and follow their favourite profiles. Notifications are tailored to provide encouragement and appreciation, fostering a supportive environment for all members while prioritizing their privacy and security.

Gymly uses modern frontend development tools like React.js to create a dynamic and responsive user experience behind the scenes. Java Spring Boot powers the backend in the meantime, enabling effective database communication and guaranteeing the smooth operation of a strong RESTful API.

2. Functional Requirements

2.1. Functional requirements for the Client Web Application

1. **User Profile Handling:**
   - Allow users to sign up and login using forms
   - Employ React Router for navigating between authentication-related pages, utilizing React hooks. [1]
   - Use Axios to make HTTP requests for user authentication, encompassing sign-up and login. [2]
   - Allowing users to sign up using google

2. **Media Upload and Posting:**
   Enable users to upload up to 3 photos or videos, each with a maximum length of 30 seconds, showcasing fitness activities, workouts, healthy meals, and progress.

3. **Workout Status Updates:**
   Authorize users to create and share updates on their current workout status, including distance ran, number of pushups completed, weight lifted, etc.

4. **Workout Plan Sharing:**
   Enable users to share their workout plans, including routines, exercises, sets, and repetitions.

5. **Meal Plan Sharing:**
   Enable users to share their meal plans, including recipes, nutritional information, and portion sizes.

6. **Interactions and Notifications:**
   - Allow users to interact with posts by liking and commenting.
   - Enable post owners to delete comments on their posts.
   - Users should receive notifications for likes and comments on their posts.

7. **Post Management:**
   Allow users to edit or delete their posts as needed.

2.2. Functional requirements for the REST API

1. **CRUD Operations:**
   Support Create, Read, Update, and Delete operations for user profiles, posts, workout plans, and meal plans.

2. **HTTP Methods Support:**
   Implement GET, POST, PUT, and DELETE methods for resource manipulation.

3. **Error Handling:**
   Provide clear error messages for invalid requests or server-side errors.

4. **Authentication and Authorization:**
   Implement OAuth2 or API keys for secure access control.

5. **Hypermedia Support (HATEOAS):**
   Enhance resource discoverability through linked responses.

3. Non Functional Requirements

3.1.Non-Functional Requirements for the Client Web Application

1. **Usability:**
   Ensure a user-friendly interface suitable for non-technical users.

2. **Performance:**
   Minimize latency for swift user interactions.

3. **Scalability:**
   Ensure the application can handle increased traffic without performance degradation.

4. **Security:**
   Adhere to strict security measures to prevent unauthorized access.

5. **Compatibility:**
   Ensure compatibility with various hardware, software, and browsers.

6. **Reliability:**
   Maintain operational reliability with minimal downtime.

3.2. Non-Functional Requirements for the REST API

1. **Performance:**
   Ensure efficient handling of requests and responses, with scalability as needed.

2. **Reliability:**
   Guarantee consistent responsiveness and availability.

3. **Scalability:**
   Manage increasing loads without sacrificing reliability.

4. **Usability:**
   Provide clear documentation and error handling for ease of use.

5. **Interoperability:**
   Adhere to industry standards for compatibility with different technologies.

6. **Extensibility:**
   Design the API to accommodate future modifications for mobile applications.

7. **Compliance:**
   Ensure adherence to relevant laws and standards.

4. System Architecture Diagram for the 'Gymly' web application

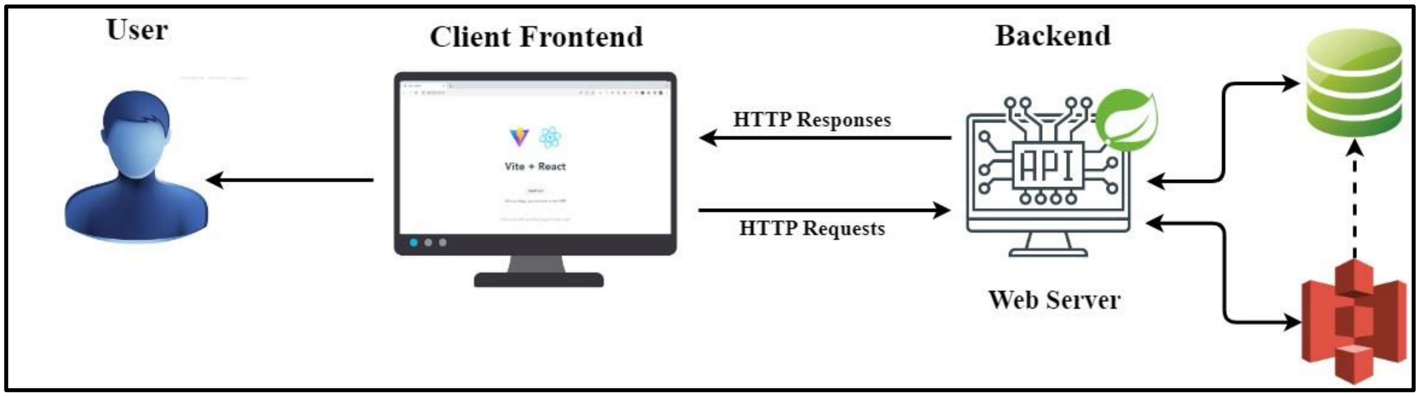

Figure 1: System Architecture Diagram [3]

5. Detailed Architecture Diagram for the REST API

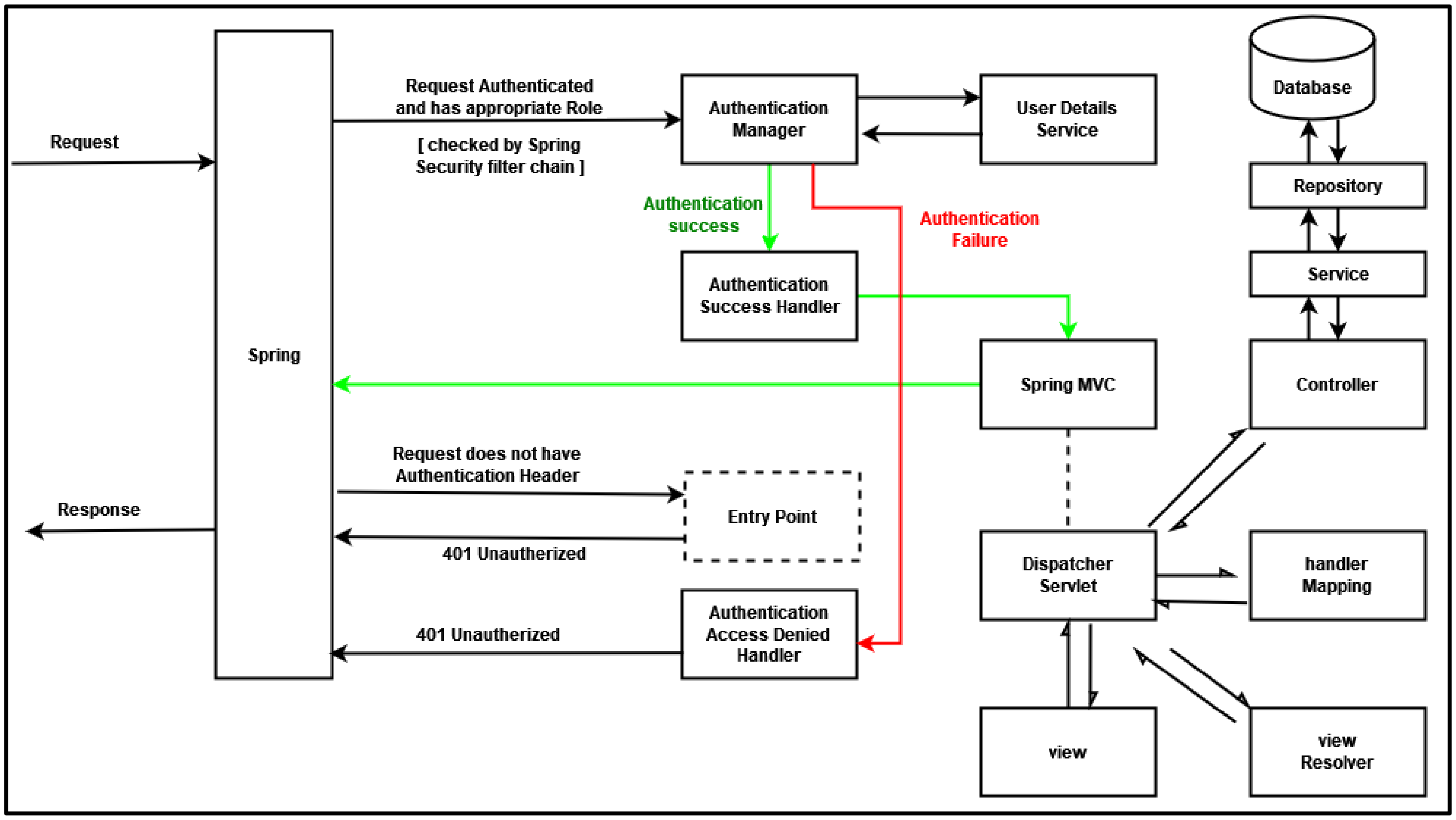

Figure 2: REST API Architecture Diagram [4]

6. Detailed architecture diagram for the client web application

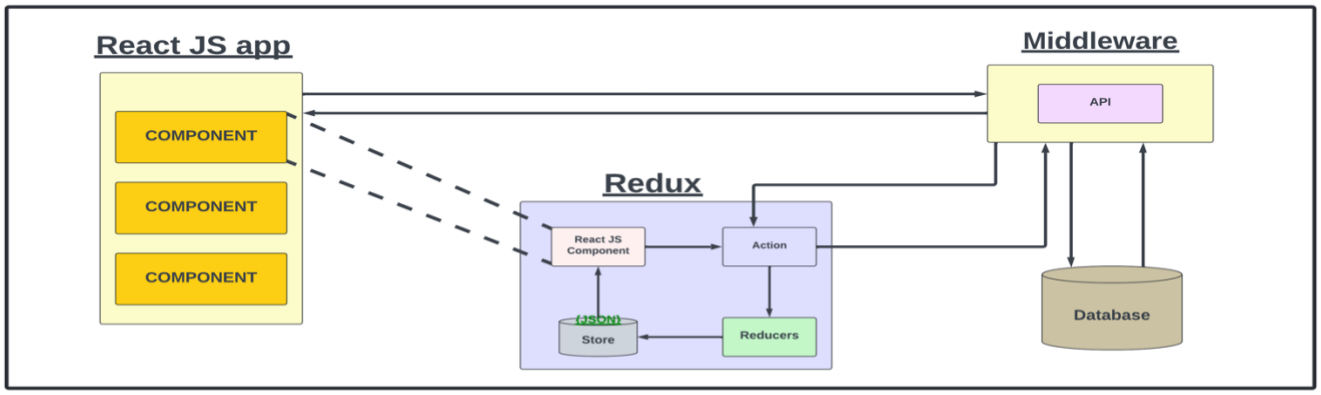

Figure 3: Detailed architecture diagram for the client web application [5]

7. References

[1] "Authentication with React Router," [Online]. Available: https://blog.logrocket.com/authentication-react-router-v6/.

[2] "How to make HTTP requests with Axios," [Online]. Available: https://blog.logrocket.com/how-to-make-http-requests-like-a-pro-with-axios/.

[3] "System Architecture – Detailed Explanation," [Online]. Available: https://www.interviewbit.com/blog/system-architecture/.

[4] "REST Architecture," [Online]. Available: https://auth0.com/blog/rest-architecture-part-1-building-api/.

[5] "Reactjs CRUD RestAPI Application Frontend Architecture," [Online]. Available: https://leloizenai.medium.com/reactjs-crud-restapi-application-frontend-architecture-diagramspringboot-react-postgresql-b608e79495ef.