Sri Lanka Institute of Information Technology

Initial Document

GROUP ID: *<GROUP_76>*

Table of Contents

Introduction

The goal of the project is to establish a social networking platform where users may upload photographs and text, update or remove their posts, check other users' profiles, follow other users, like and comment on posts, get notifications, and even create groups with followers.

The platform should be well-engineered with a REST API that can be accessed by both applications, be accessible as a client web application and mobile application and be simple and straightforward for non-technical users to use.

Application programming interfaces, or APIs, are a collection of guidelines that specify how devices and software can connect to and communicate with one another. An API that conforms to the representational state transfer, or REST, design principles is known as a REST API.

In our project client approaches the team with an idea to develop an a social media platform for "foodies" to share their experiences.

A user should be able to create posts, edit them, have a profile, view other users' profiles, and follow them. All profiles should be public.

The most important details in this text are that users should be able to "like" and comment on other users' posts, and the original author of the post should also be able to edit them or delete them. The platform should be user friendly and intuitive to use, and a user should get notified when another user likes/comments on their post.

The client intends to make the social media platform available as a client web application and mobile application, and a separate company is to be contracted to develop the REST API and the client web application of the platform. The REST API must be well engineered, secured with

OAuth based authentication, and version controlled using Git on GitHub, using the GitHub Workflow.

01. Functional requirement for REST API

- User authentication and authorization: Users should be able to sign up and sign in to the platform using their credentials The API should give suitable mechanisms to clarify proper authentication.
- **Post Management- T**he API should allow users to create posts, delete, and edit posts while also providing proper mechanisms to retrieve previously added posts by using their post ID or the time and date it was posted.
- **Group Management-** The API should allow users to create groups and allow join to already created groups based on different criteria while also giving full authorization for the creator of the group**.**
- **Comment Management-**The API should allow users to add comments and add like to other users' posts, allowing owners to delete them.
- Follow managements-The API should allow users to view their followers' and followers' lists in addition to following and unfollowing other people. The list of suggestions that follows should also be displayed.
- **Profile Management-** The API should enable users to create a profile on the platform while also allowing users to add any picture they want, delete or edit their profiles and biographies while also allowing users to maintain their profiles.
- **Notification Management-** The rest API should allow users to receive notifications about updates and recent subscriptions and other related materials.

02. Non-Functional requirement for REST API

- Performance: The REST API should respond to user requests quickly and with low latency. The API should be able to handle a large number of requests without becoming slow or crashing.
- scalability-The REST API should be able to handle an increasing number of users and data without degradation in performance.
- Robustness-If a user forgets his or her login information, the API should provide a method to recover the information. If a user forgets their password, the API should provide a way to reset it.
- Security: The REST API should be secure and protect user data from unauthorized access. The API should use authentication and authorization mechanisms such as OAuth, JWT, or API keys
- Reliability-The API should provide the ability with which the software performs its required functions under stated conditions for a specified period of time.

03. Functional requirement for Client Web application

- User Interface: The web application should have an impress and user-friendly interface that is easy to navigate.
- User authentication and authorization: The application should allow users to create accounts and log in, log out securely, and it should control access to certain features based on a user's permissions. Also, user should be able to reset their password using their given email address.
- User profiles: users should be able to create and customize their own profiles, including adding profile pictures, biographical information, and other relevant details.
- Sign Up and Login: Users should be able to sign up, log in and log out.
- Posts: users are allowed to create and share various types of post contents, such as text posts, photos, videos, and links. Interface should display all posts created by user and user profiles as suggestions.
- Reviews and Ratings: Users should be able to add, view, and edit their reviews and ratings for food item. The interface should provide clear guidelines on how to write a description and how many photos can be included in the post.
- Notification: The interface should notify users when there are new reviews, comments, or replies to their comments. The notification system should be user-friendly and unobtrusive.
- Groups: User should be able to create groups adding other users, view & edit group details also delete groups.
- Like & Comment: users have the option to give thumbs up and write comments on posts.
- Privacy and security: There are measures to protect user privacy and security, such as by implementing secure login processes, enabling two-factor authentication, and giving users control over who can see their content and interactions
- Search functionality: The application should include search functionality that enables users to find specific content or users, or to search for content based on certain criteria (e.g., hashtags, keywords).
- Accessibility: The application should be designed with accessibility in mind, ensuring that all users, regardless of ability, can use and navigate the application with ease.

04. Non-Functional Requirements for client web application

- Accessibility: The web application should be accessible to users with following the Web Content Accessibility Guidelines.
- Responsiveness: The web application should be responsive and able to adapt to different screen sizes and devices, such as smartphones and tablets.
- Performance: The web application should be fast and responsive, with low latency and high throughput. It should also have a caching mechanism to improve performance.
- Compatibility: The web application should be compatible with multiple web browsers and operating systems.
- User Experience: The web application should provide an excellent user experience, with an impress and user-friendly interface, clear navigation, and engaging design. It should also be easy to use and understand, with clear feedback and error messages.

5.Diagrams

5.1 Overall architecture diagram

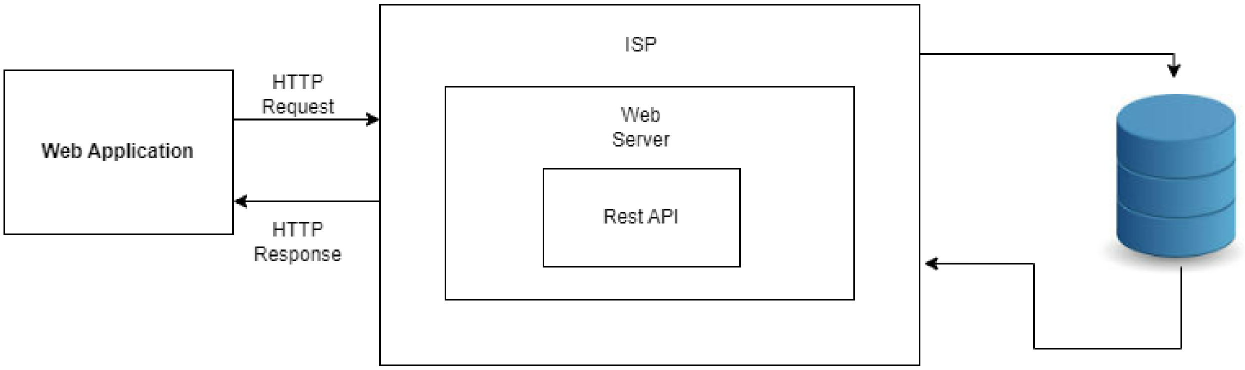
System's overall design

overall Architecture

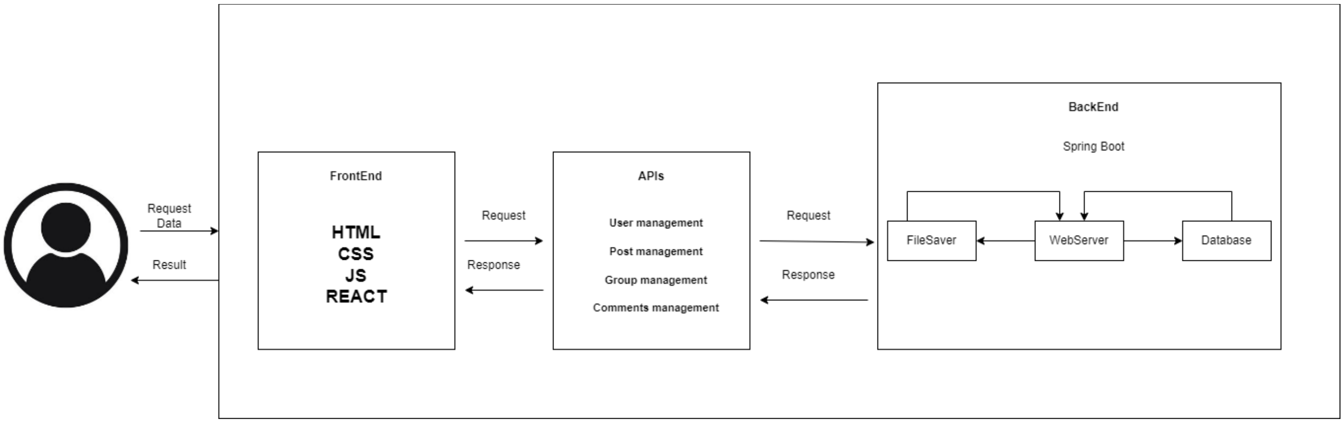
User Interface

Functions	Database	Backend
User management		spring boot
Post management		spring boot
Group management		spring boot
Comments management		spring boot

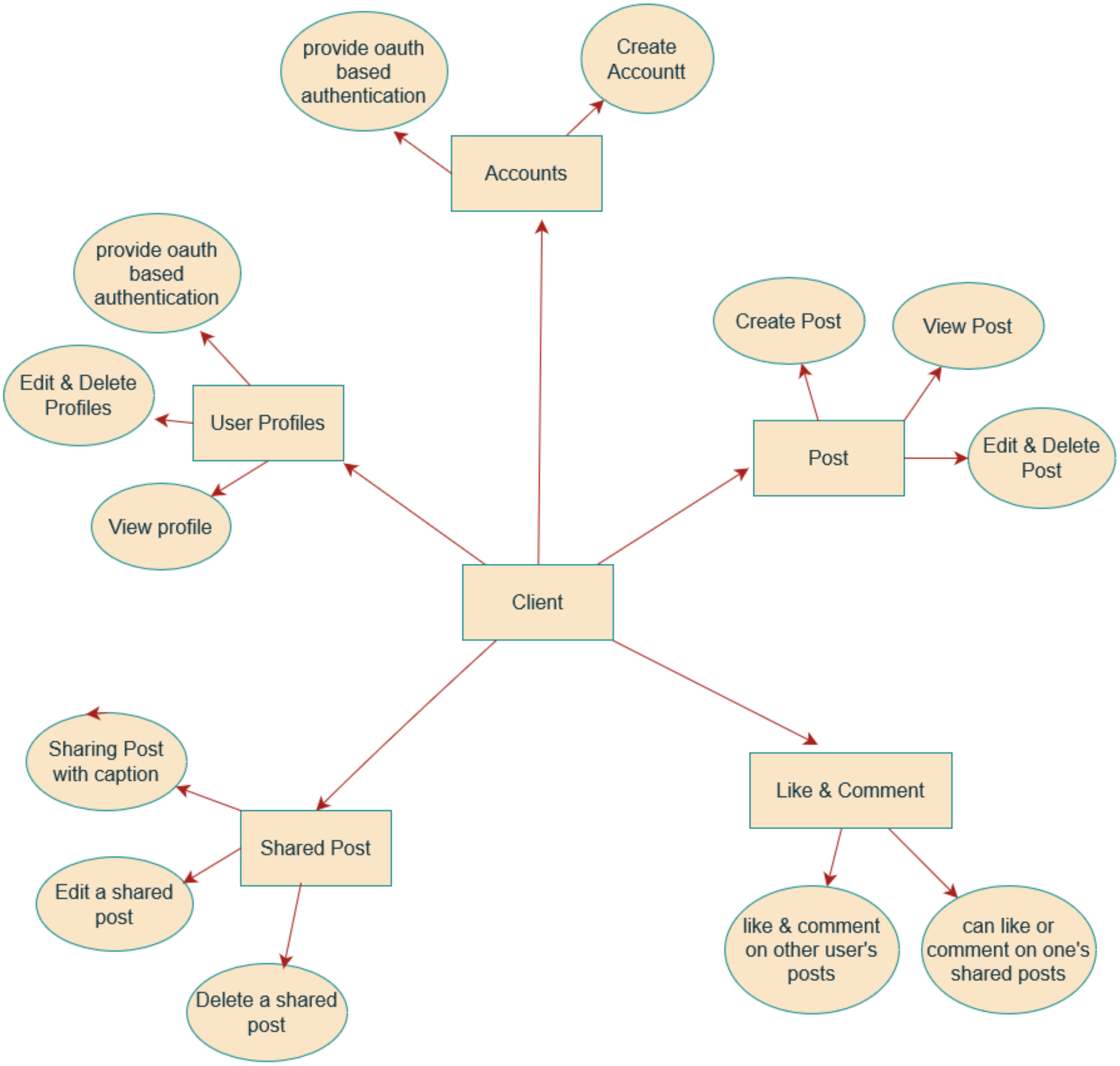
5.2 Detailed architecture diagram for the REST API



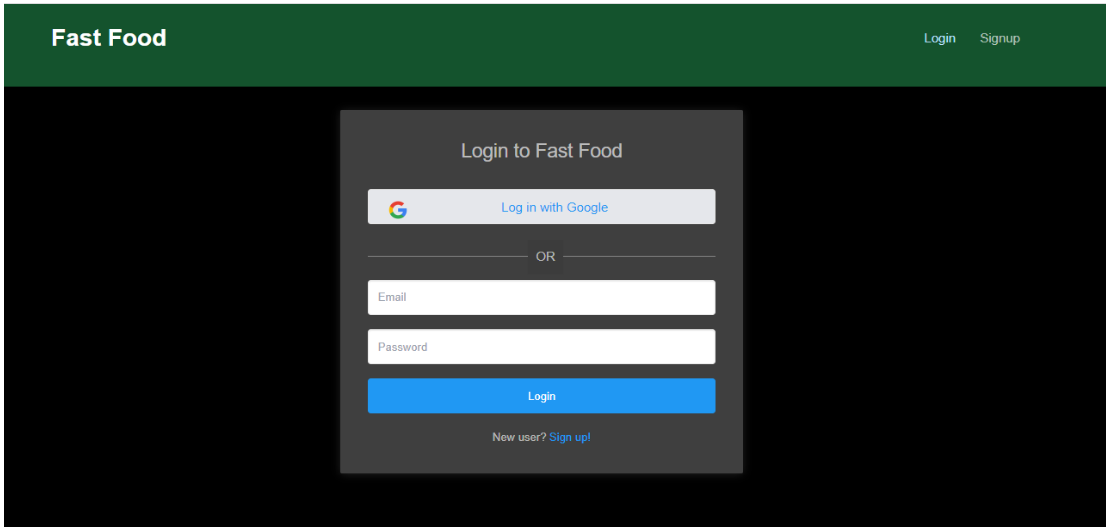
5.3 detailed architecture diagram for the client web application



5.1 Function Architecture diagram of client web application

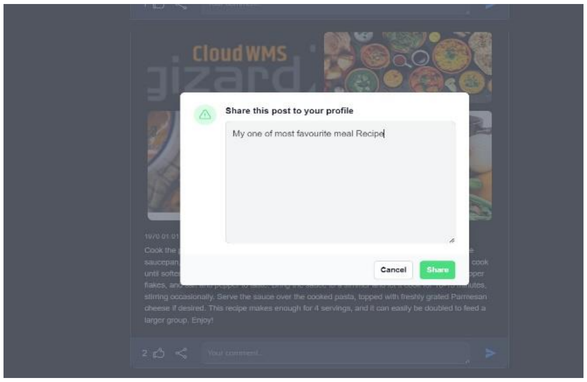


6.Individual functions



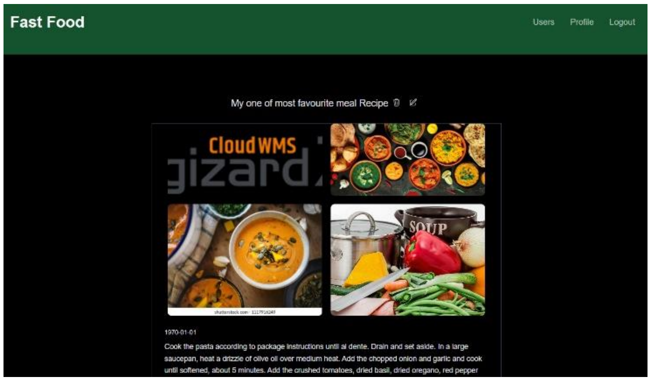
6.1User Management –

The initial screen that a user encounters necessitates logging in to access the system. One can enter their credentials or opt for signing in with Google. Each user possesses a profile, which they may modify by updating their personal information and managing details within it. Additionally, one's profile showcases all of the content created by them on this platform. As part of exploring other users' profiles, suggestions are provided while searching is also an option available to facilitate browsing through fellow members' pages on this social media site.



My one of most favourite meal Recipe

stirring occasionally. Serve the sauce over the cooked pasta, topped with freshly grated Parmesan cheese if desired. This recipe makes enough for 4 servings, and it can easily be doubled to feed a larger group. Enjoy!



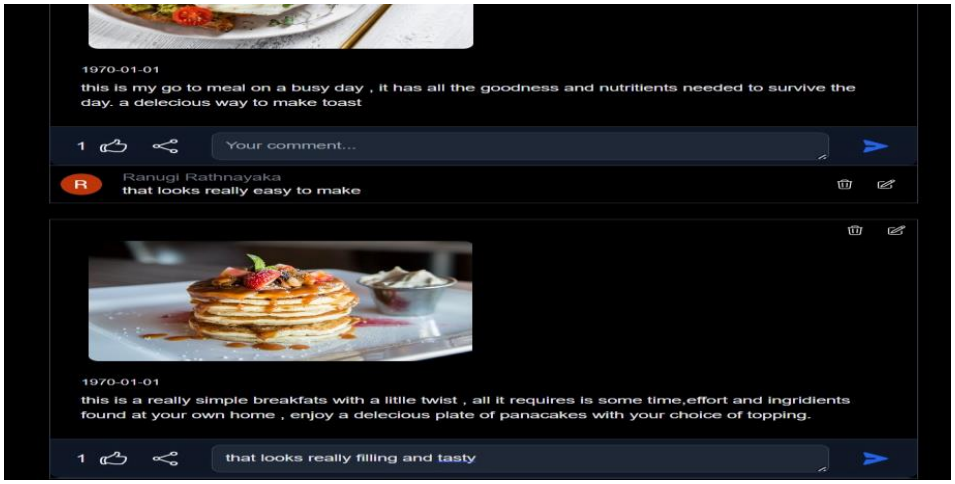
My one of most favourite meal Recipe

1970-01-01

Cook the pasta according to package instructions until al dente. Drain and set aside. In a large saucepan, heat a drizzle of olive oil over medium heat. Add the chopped onion and garlic and cook until softened, about 5 minutes. Add the crushed tomatoes, dried basil, dried oregano, red pepper

6.2 Share Post Management

In order to disseminate a post on this particular social media platform, initial login is required. Once logged in, the user can browse through various posts and select the ones that capture their interest; subsequently sharing them onto their personal wall within the application. This process involves utilizing an existing share icon present within each individual post. Through clicking said icon, users are given the opportunity to include a personalized caption with their shared content - thus initiating crud operations. The option for editing such captions is also available via selecting an edit icon while viewing previously shared material. Additionally, removal of these posts from one's own personal wall may be achieved by selecting delete icons associated with specific items or updates.



1970-01-01

this is my go to meal on a busy day , it has all the goodness and nutritients needed to survive the day. a delecious way to make toast

Ranugi Rathnayaka
that looks really easy to make

1970-01-01

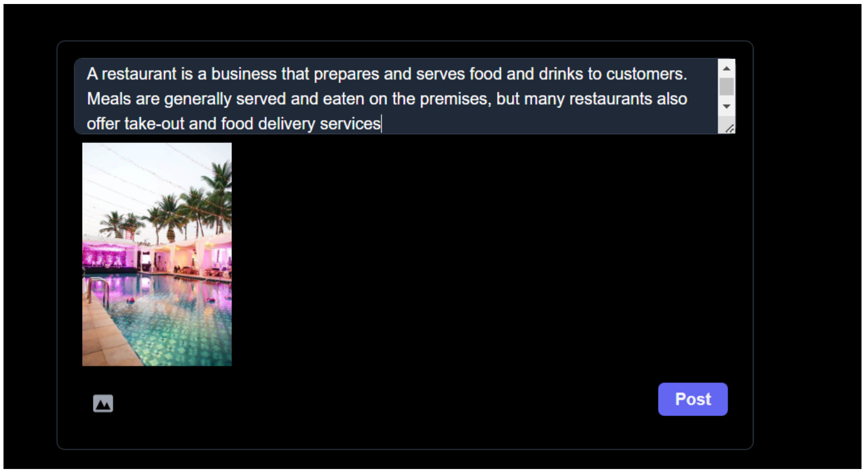
this is a really simple breakfats with a litile twist , all it requires is some time,effort and ingridients found at your own home , enjoy a delecious plate of panacakes with your choice of topping.

that looks really filling and tasty

6.3Like And Comment Management

The function I developed encompasses the management of comments and likes. Comment: Within posts and shared posts, there is an explicit icon denoting "comment." By selecting this icon, users gain the ability to create comments. Users are permitted to submit comments and pose questions within posts. This functionality incorporates CRUD operations, as users are presented with the options to edit their posted comments by selecting the "edit" icon. Additionally, unwanted comments can be easily removed by clicking the "delete" icon.

Like: When encountering a post, users have the option to express their appreciation by liking it. An icon is provided to enable users to take specific actions and make choices. Furthermore, users retain the capability to "unlike" any post that they had previously shown their approval for



A restaurant is a business that prepares and serves food and drinks to customers. Meals are generally served and eaten on the premises, but many restaurants also offer take-out and food delivery services

6.4 Post Management

Posting content is a fundamental feature of a social media platform. In order to make a post on this platform, users must first log into their accounts. Once logged in, they can navigate to the post creation page by selecting the "Fast Food" option. On this page, users have the ability to include a description, attach an image, and ultimately create a post by clicking the "Post" button. Once a post is created, it is then published and made visible to others. This is where the CRUD (Create, Read, Update, Delete) operations come into play. Based on the user's preferences, posts can be modified by selecting the "edit" icon, and the edited version can be viewed. Conversely, if a user wishes to remove unwanted posts, they can simply click the "delete" icon, leading to the deletion of the respective content.

7. Repository link

https://github.com/cham102/Food-Review-App