Software Design Document

MyRecipe webApp

Submitted to: Professor Alex Kuhn

April 23, 2020

Group Members

1, Rediet Negash: rediet.negash@stonybrook.edu

2, Cogitater Sigauke: cogitater.sigauke@stonybrook.edu

3, Merry Mekonnen: merry.mekonnen@stonybrook.edu

4, Pyungkang Hong: pyungkang.hong@stonybrook.edu

Content

Software Design Document

1. System Architecture

1.1 Overview

We are using Spring Boot 2.2, Azure Cosmos DB for MongoDB API, Firebase, React 16.8, Bootstrap 3, Java as a programming tool, WebRCT(API), and libraries such as, Messaging libraries(JMS), Logging libraries, Unit-testing libraries, JSON parsing libraries. We will be using Spring Boot Framework and Firebase on the backend and follow the MVC(Model View Controller) pattern.

The front end of our website will be purely built on React 16.8. React supports many core features which will be of great use for our website, such as component-based approach, backward compatibility, flexibility. At this point, we have a clear picture of the components we need for our website and using React will allow us to design the UI components with a minimal effort as it follows a component-based approach. React also provides backward compatibility and flexibility which we believe our website will greatly benefit from. We have decided to use the latest version of React since many useful APIs are included such as Hooks. It is highly likely that we use the newly included APIs in React. In addition to React, we are using Bootstrap for the front end of our website for styling the pages. As Bootstrap has excellent documentation, it will save our time on styling the front-end. We will be using Bootstrap 4.

The back end of our website will use Spring Boot 2.2, Microsoft Azure Cosmos MongoDB and Firebase. We decided to use Spring Boot because it is auto configured and does not need any manual configuration. It also manages Rest points easily. Similarly, MongoDB is simple to install and implement. It also provides high performance and automatic scaling. The libraries we decided to use may change later in the process, depending on the implementation of our code.We will use Firebase to store images.

1.2 Components and interfaces

Major React Components for the frontend

App - root component

Home - extends from app and is parent for the following components

searchBar - allows user to search for a meal either by meal name or Ingredient that it contains- (through ingredients filter).

NavBar - It has logo-link to home-, search bar and some important buttons like- Login, Logout, profile and addRecipe

Categories: it has the meal categories including the following

meal type:- it could be either breakfast, lunch, dinner or supper
dish type:- starter, main dish, sweets, salad, fruits, juice
diet and health:- healthy foods, diet based dishes
world Cuisine:- it has cultural foods from different countries

ContactUs - is a footer that has contact information including aboutUs, social media links, phone numbers, email address and other contact information.

Recipe- it lets the user add their own recipes

addRecipe:

Categories: allows the user add a recipe to a specific category by providing an option where the user can select from list of categories

descriptionForm: It contains meal name, description box, steps with an option to upload images for recipes and submit button to complete the form and register the recipe

editRecipe: lets the owner of the recipe edit the recipe they already have.

deleteRecipe: lets the owner delete the recipe that he/she added.

profile - this refers to the personal pages of a specific user

myRecipes - it has the recipes that are uploaded by the user so far with their ratings

Account:

Login - lets the user login with their google account using google auth.

Logout - lets the user logout from their account

editProfile- lets the user change password, username, or aboutMe description including their profile picture

chatBox - root component

Title- contains the user name

messagesList- it contains the list of messages the people chatted

sendMessageForm- this refers to the text box where the user types the message

We will have one index.html file where it is used to link to the generated JS components, and has that one single parent node for the React content to attach itself to.

1. 3 Third party Libraries and APIs

Spring Boot- this would basically serve as our backend server

React.js - we will use react for developing our frontend application

WebRCT - this is the API we will use for live streaming

Algolia Search Engine - This is a search engine that we will use for searching our ingredients and meals. It allows human language sentence searches as the algorithm so the user doesn’t need to worry about entering a specific key for searching.

Google Auth- this is what we will use to authenticate a user when logging in. This would avoid any problem with account insecurity as google does it for us.

1.4 Detailed description for object oriented classes

(1) Recipe

- Attributes

Name	Type	Description
ID	String	The ID value will be the key that defines a single recipe.
LikeCount	Int	The number of likes that the recipe received
MealType	String	Is a category that describes the type of the meal which could be breakfast, lunch, dinner or supper
DietAndHealth	String	It is another category of a meal that contains specific type of meals that are healthy and diet specific
WorldCuisine	String	This category refers to cultural foods from different parts of the world
MealName	String	Refers to the name of the meal that is entered when the Recipe is added to the database
Description	String	Refers to the description entered by the user when the recipe is created and iis added to the database

-RecipeRepository: An interface to store,edit,and delete a recipe.

Name	Parameters	Return values	Description
delete	Recipe	void	Deletes Recipe from the Database
save	Recipe	void	Saves Recipe to the Database
find	recipeId	Recipe	Fetches Recipe with the given ID
update	recipeId	Recipe	Updates the Recipe

(2) GeneralUser<Interface>

- Methods

Name	Type	Description
getID	String	Describes the user’s Id
getName	String	Has user's username
getProfilePic	Image	Has user's profile image which is an Image object

(3) AdminUser

- Attributes

Name	Type	Description
ID	String	Describes the user’s Id
name	String	Has user's username
profilePic	Image	Has user's profile image which is an Image object

(4) User

- Attributes

Name	Type	Description
ID	String	Describes the user’s Id
name	String	Has user's username
profilePic	Image	Has user's profile image which comes from google account
aboutMe	String	Describes user's 'about me' information

- The User Repository - An interface for the interaction with the user model and performing database.

Name	Parameters	Return values	Description
delete	user	void	Deletes user from the Database
save	user	void	Saves user to the Database
find	userId	User	Fetches user with the given ID
update	userId	User	Updates the user

(5) Message

- Attributes

Name	Type	Description
ID	String	ID value that uniquely identifies the message object.
sender	User	the user by whom the message is sent
receiver	User	the user to whom the message is sent
messageText	String	has the message text to be sent to the receiver

- The Message Repository

Name	Parameters	Return	Description
editMessage	User	void	is used to edit the text message to user after the message being sent
deleteMessage	User	void	is used to delete a text message which is already sent
sendMessage	User	void	used to send a message to the receiver

(6) Comment

- Attributes

Name	Type	Description
ID	String	The comment id helps to uniquely identify comments
recipe	Recipe	Refers to the recipe that the comment is given to

-The Comment Repository

Name	Parameters	Return	Description
save	Comment	void	addComment adds the user’s comment given to a specific recipe
delete	Comment	void	Delete method is used to delete a text comment which is already sent
update	Comment	void	editComment method is used to edit the text Comment to user after the comment is posted

7)Review

- Attributes

Name	Type	Description
ID	String	The Id would uniquely identify the review object
recipe	Recipe	This is the recipe that is being reviewed or rated
reviewGiven	Int	This refers to the review that the recipe has received. It is the number of starts that the commenter gave while rating

-The Review Repository

Name	Parameters	Return	Description
save	Review	void	Save review in the database
Find	reviewId	void	Fetches a review from the database given the ID
update	Review	void	Updates a review in database

8) Image

- Attributes

Name	Type	Description
ID	String	The id uniquely identifies the image object

-Image Repository

Name	Parameters	Return values	Description
Delete	image	void	Deletes image from the Database
Save	image	void	Saves image to the Database
Find	imageId	Image	Fetches image with the given ID
update	imageId	Image	Updates the image in the database

1.3 UML Class Diagram

https://app.diagrams.net/#G1r1xh5914B5bmROgwju_OvgGK0r-5If2y

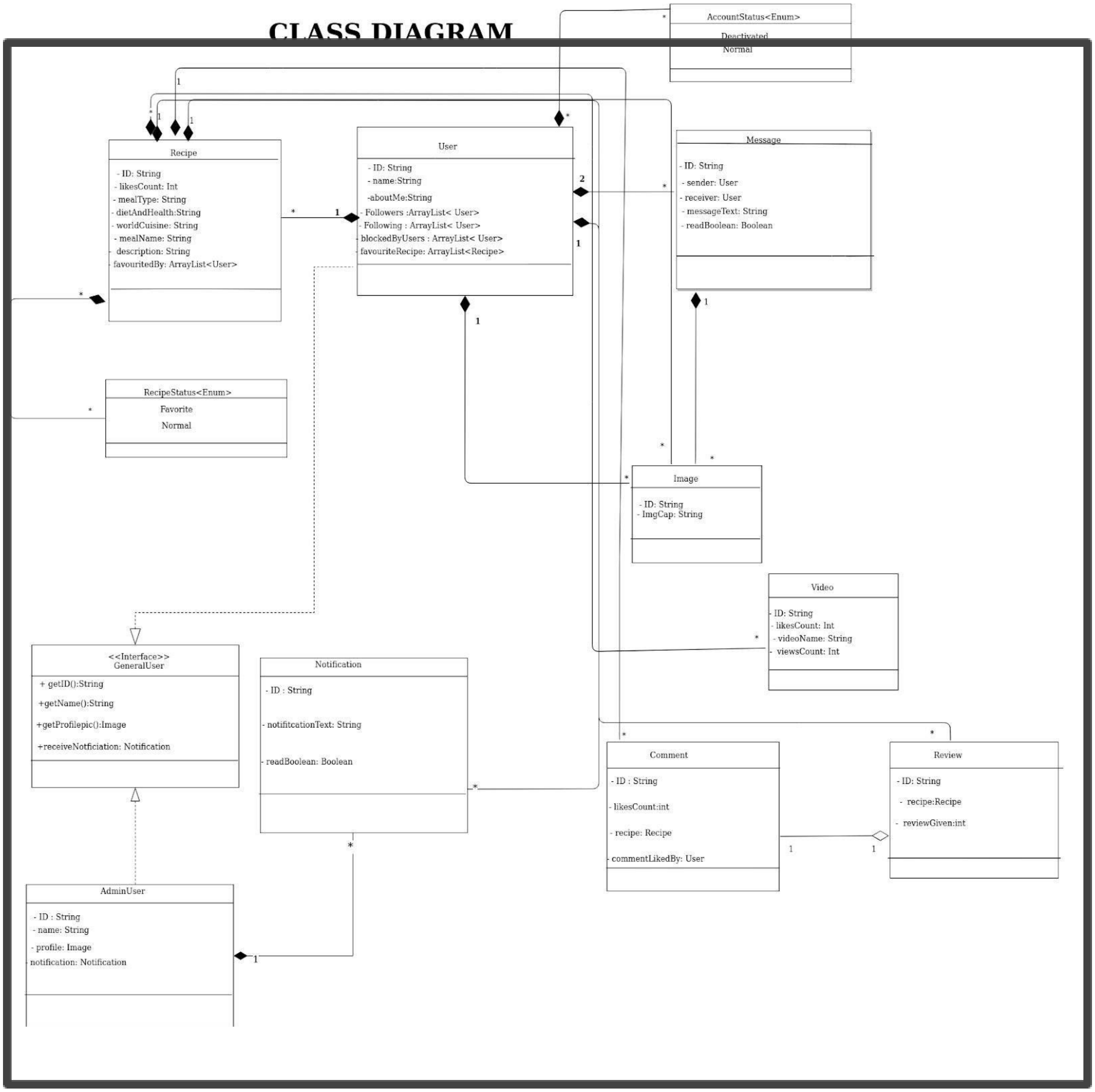


Figure 1.1

The following class diagrams are zoomed version of the above to help with readability

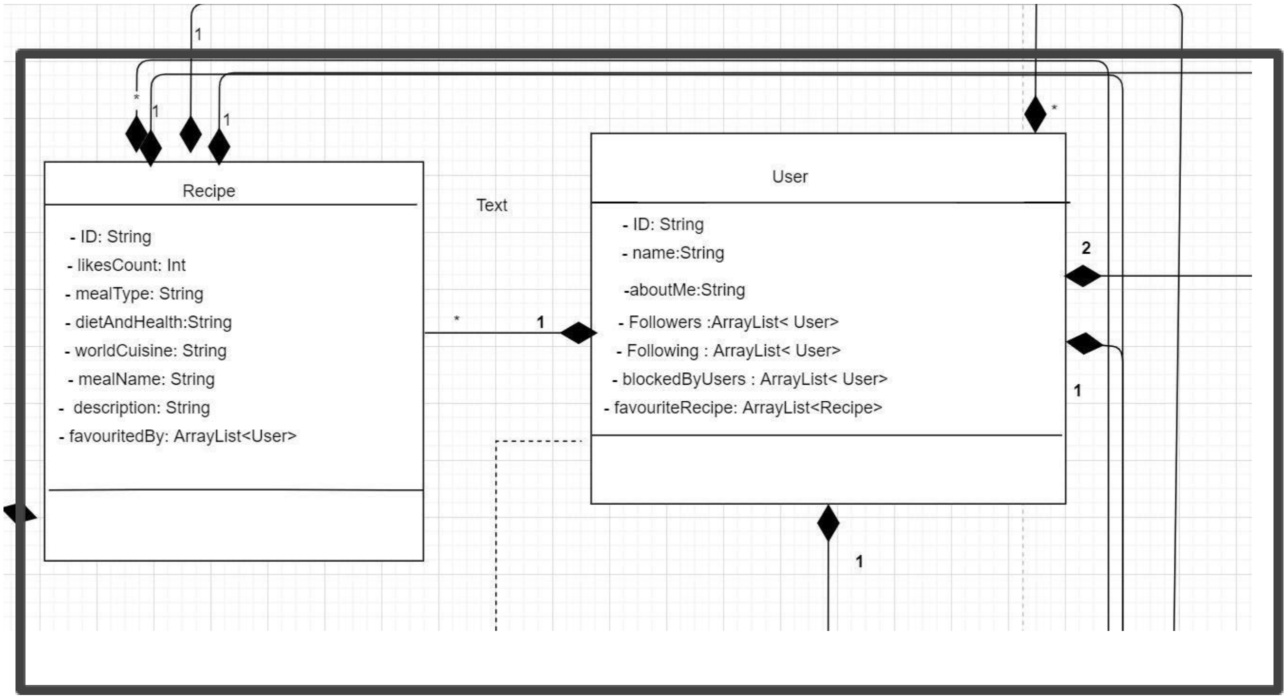
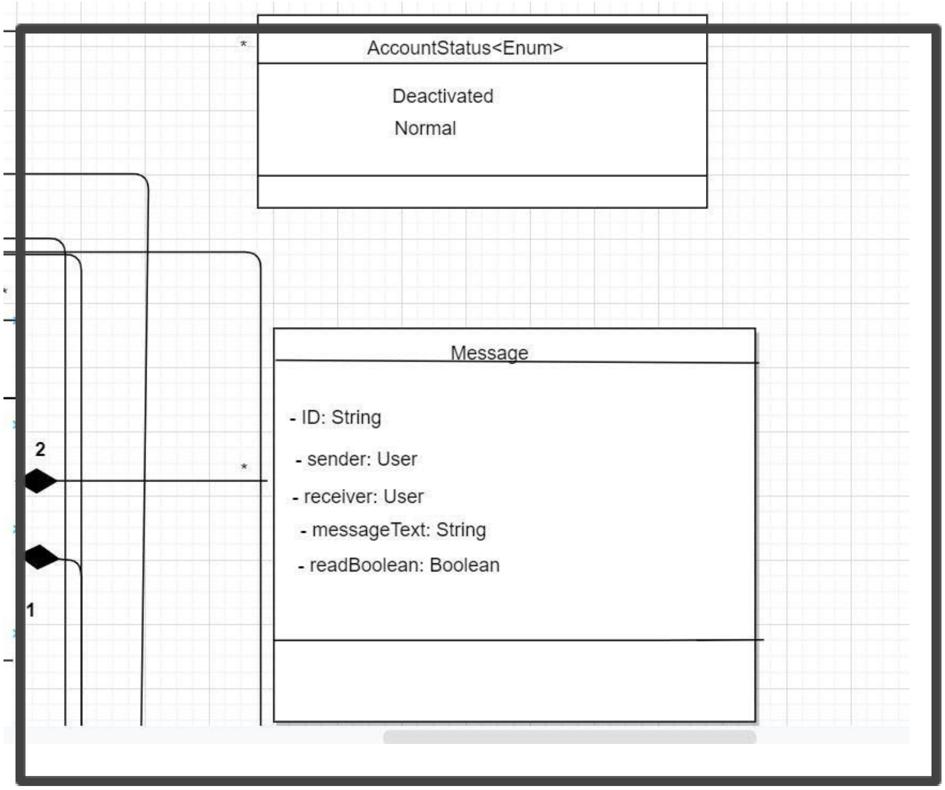


Figure 1.2

Figure 1.3



*
AccountStatus<Enum>
Deactivated
Normal
Message
- ID: String
- sender: User
- receiver: User
- messageText: String
- readBoolean: Boolean
2
*
1

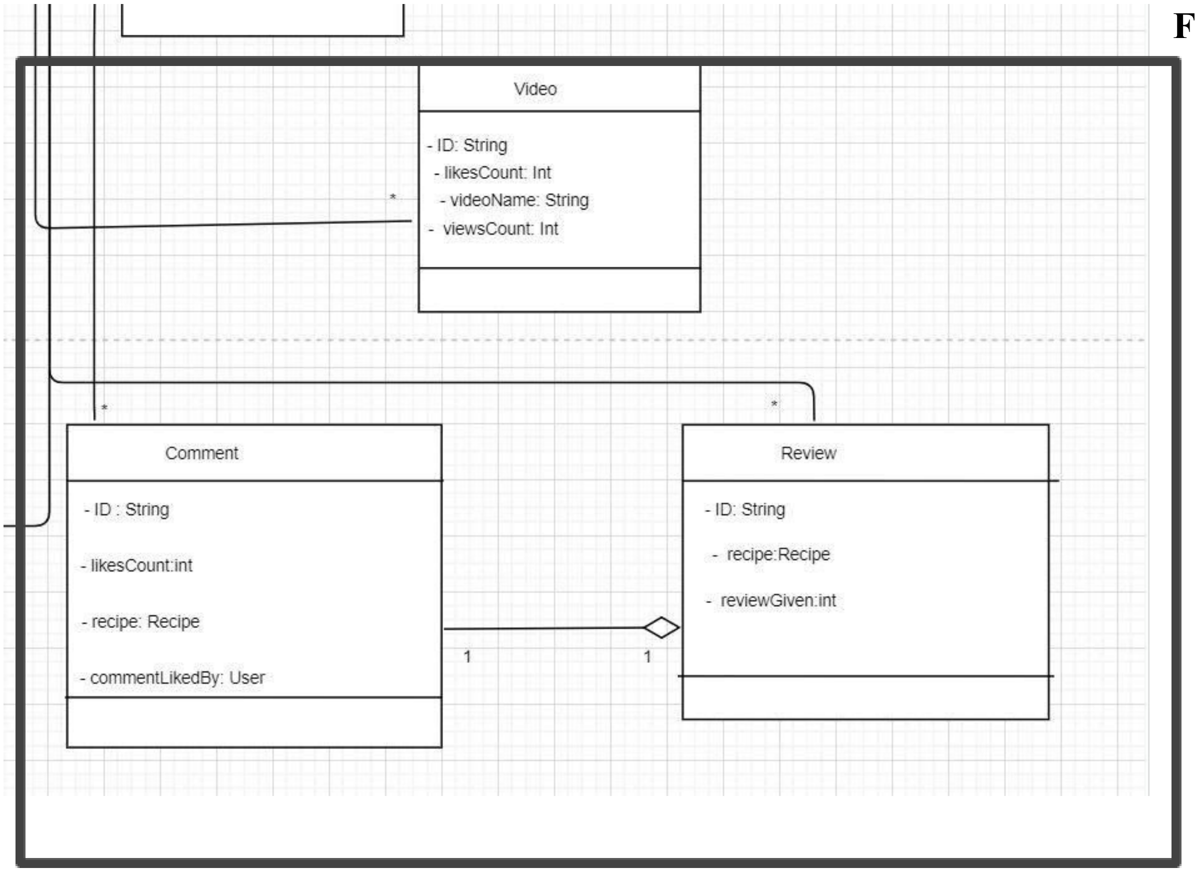


Figure 1.4

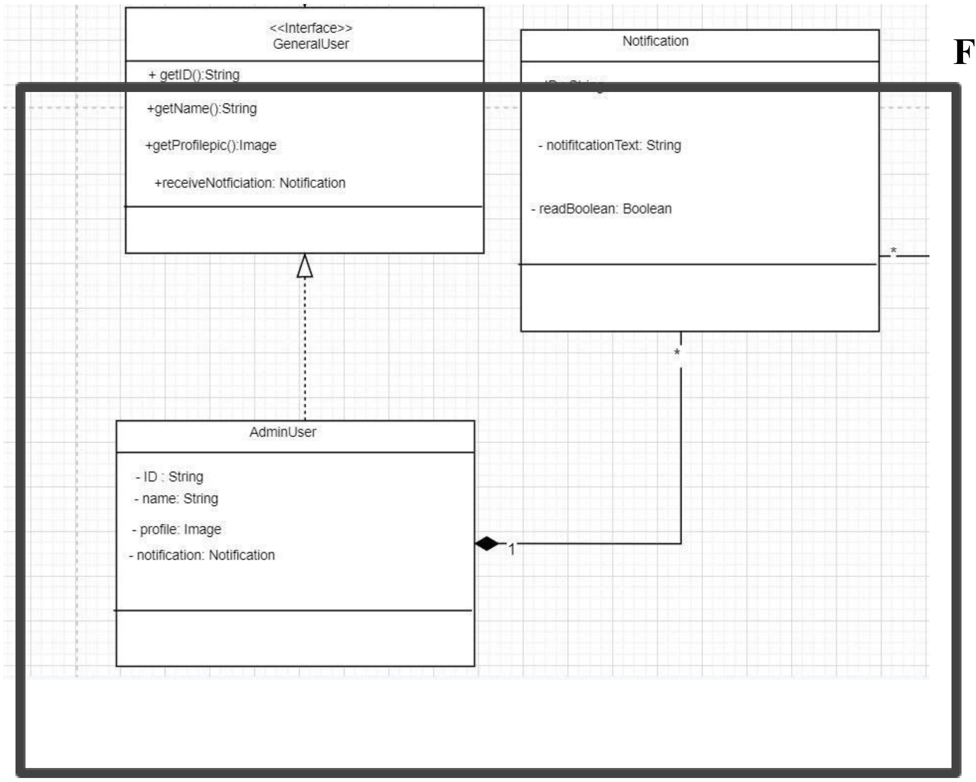
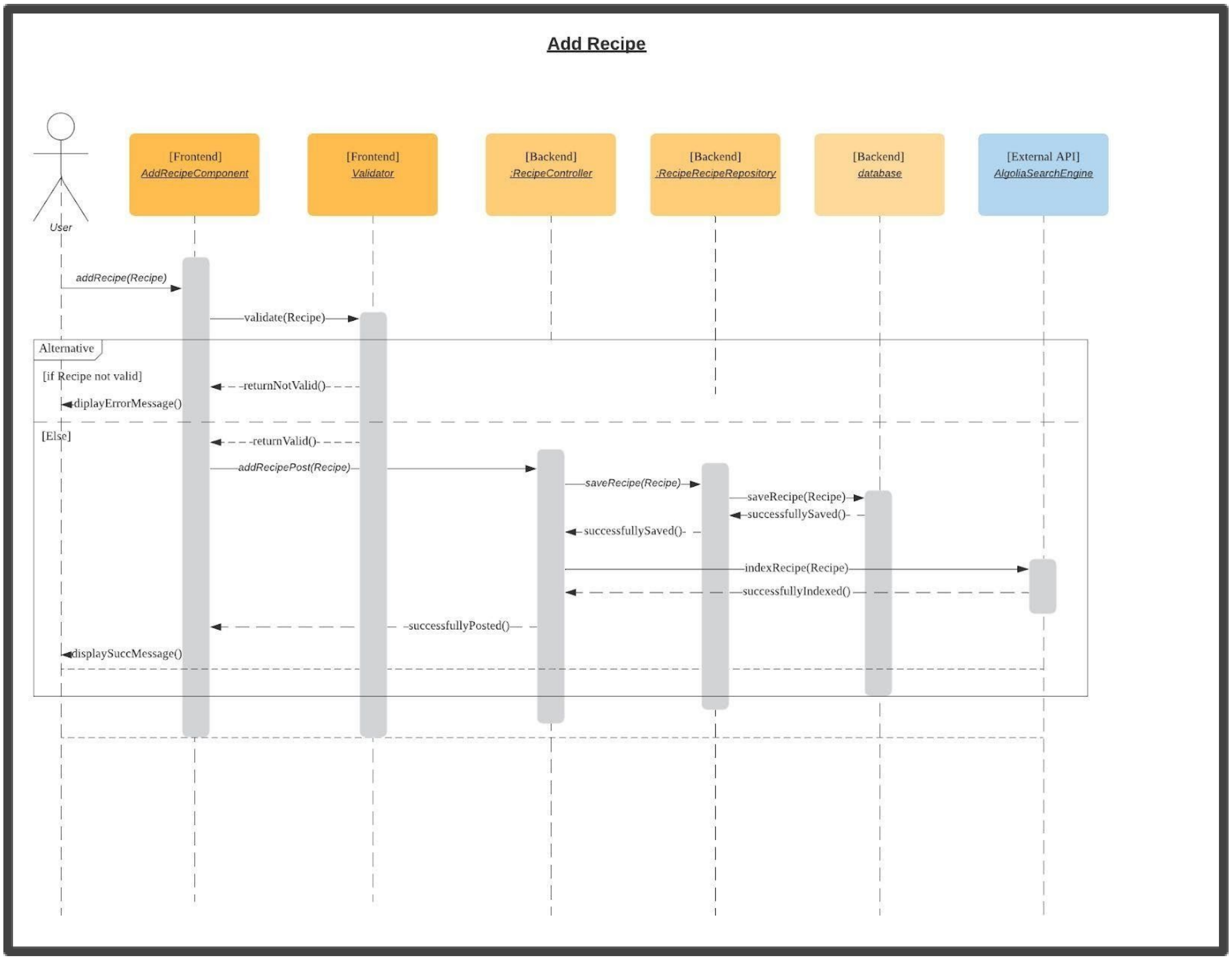


Figure 1.5

1.4 UML Sequence Diagrams

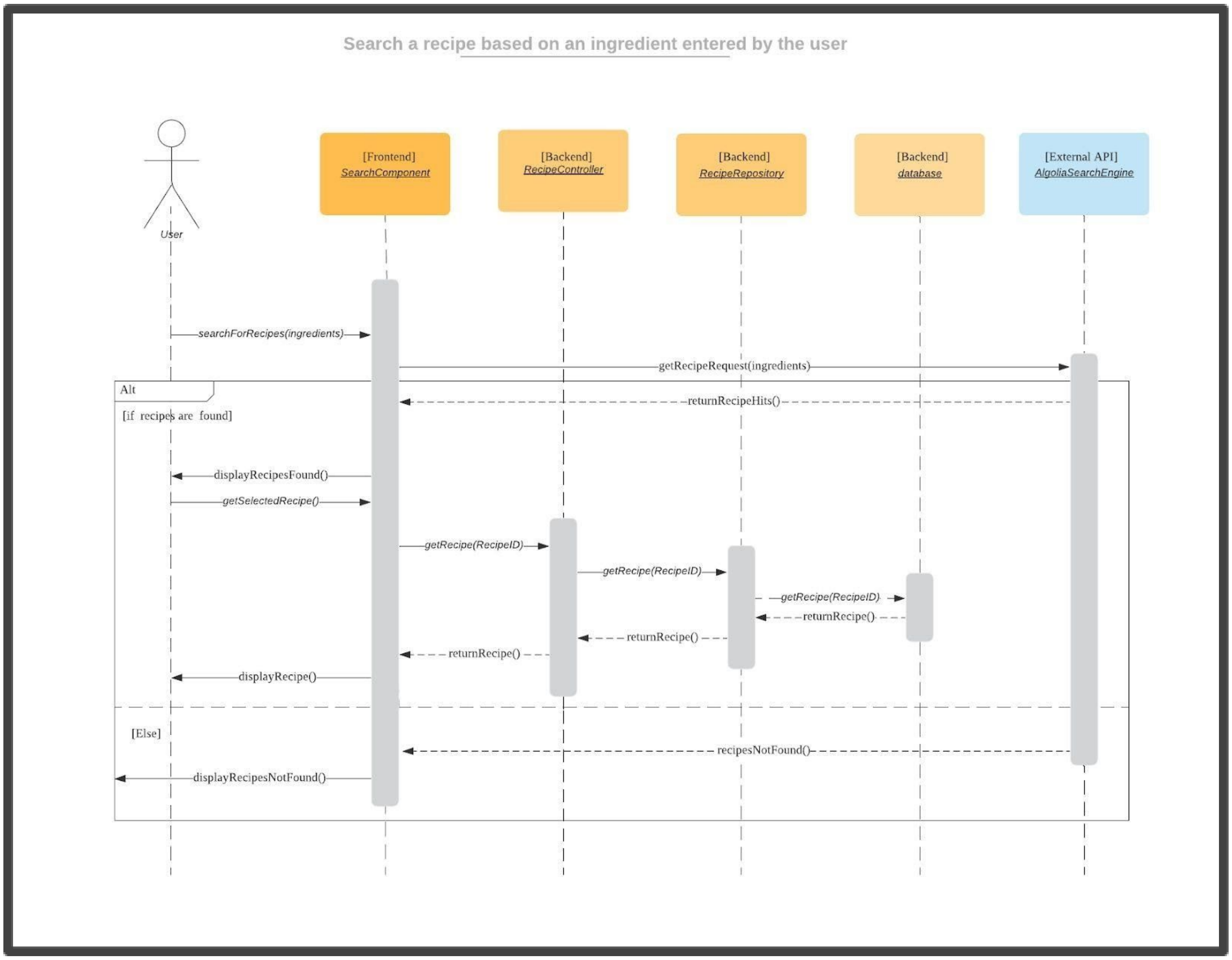
1.4.1 ADD Recipe

https://www.lucidchart.com/documents/edit/daa98cb7-cfb7-4c9d-8ab7-2d4912b9c1ca/0_0



1.4.2 Search a recipe based on an ingredient entered by the user

https://www.lucidchart.com/documents/edit/ef294250-de7b-4e97-9052-439695208515/0_0



1.5 Deployment:

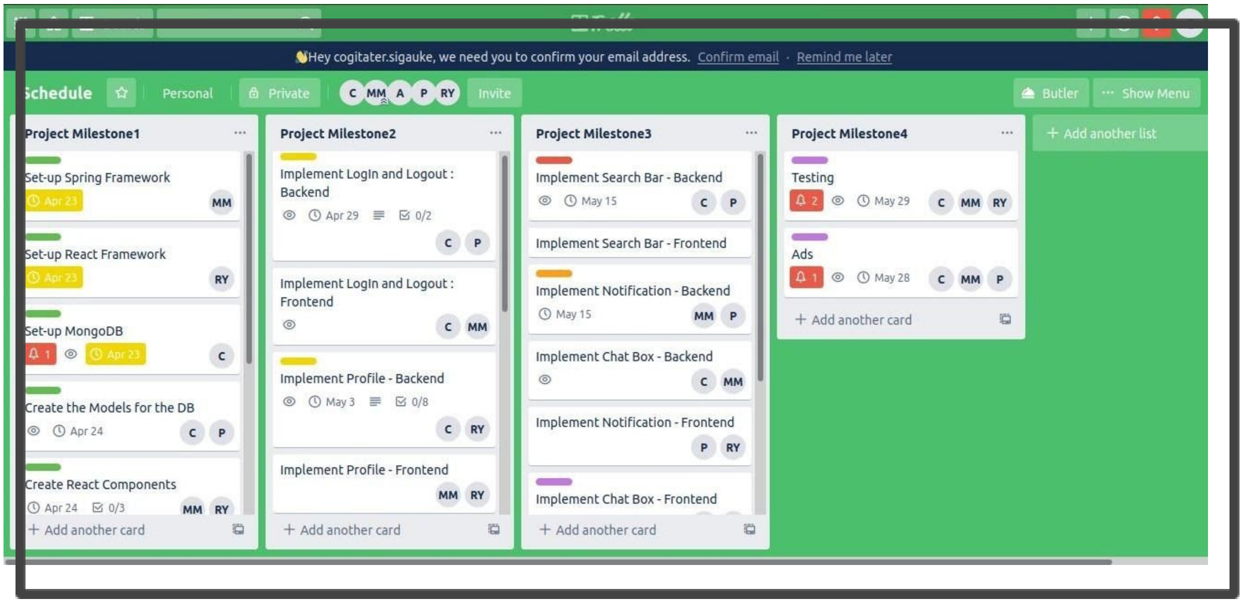
We will be using Microsoft Azure for the full development of our application. For this, we will be using the Azure Cosmos DB for MongoDB API as our database. In addition, both the backend spring boot application and the react front end of our application will be deployed to Microsoft Azure.

1.6 Alternatives:

We will be using MongoDB for our database with MySQL as a second option. Similarly, We plan to use Google Auth for signing in(authentication and verification). However, depending upon the users' experience, we might include account creation features rather than simply using Google Auth.

2. Schedule

https://trello.com/b/OUJ2p6AO/schedule



👋Hey cogitater.sigauke, we need you to confirm your email address. Confirm email · Remind me later

Schedule ☆ Personal Private C MM A P RY Invite Butler ··· Show Menu

Project Milestone1

- Set-up Spring Framework — Apr 23 — MM
- Set-up React Framework — Apr 23 — RY
- Set-up MongoDB — 1 — Apr 23 — C
- Create the Models for the DB — Apr 24 — C P
- Create React Components — Apr 24 — 0/3 — MM RY

+ Add another card

Project Milestone2

- Implement Login and Logout : Backend — Apr 29 — 0/2 — C P
- Implement Login and Logout : Frontend — C MM
- Implement Profile - Backend — May 3 — 0/8 — C RY
- Implement Profile - Frontend — MM RY

+ Add another card

Project Milestone3

- Implement Search Bar - Backend — May 15 — C P
- Implement Search Bar - Frontend
- Implement Notification - Backend — May 15 — MM P
- Implement Chat Box - Backend — C MM
- Implement Notification - Frontend — P RY
- Implement Chat Box - Frontend

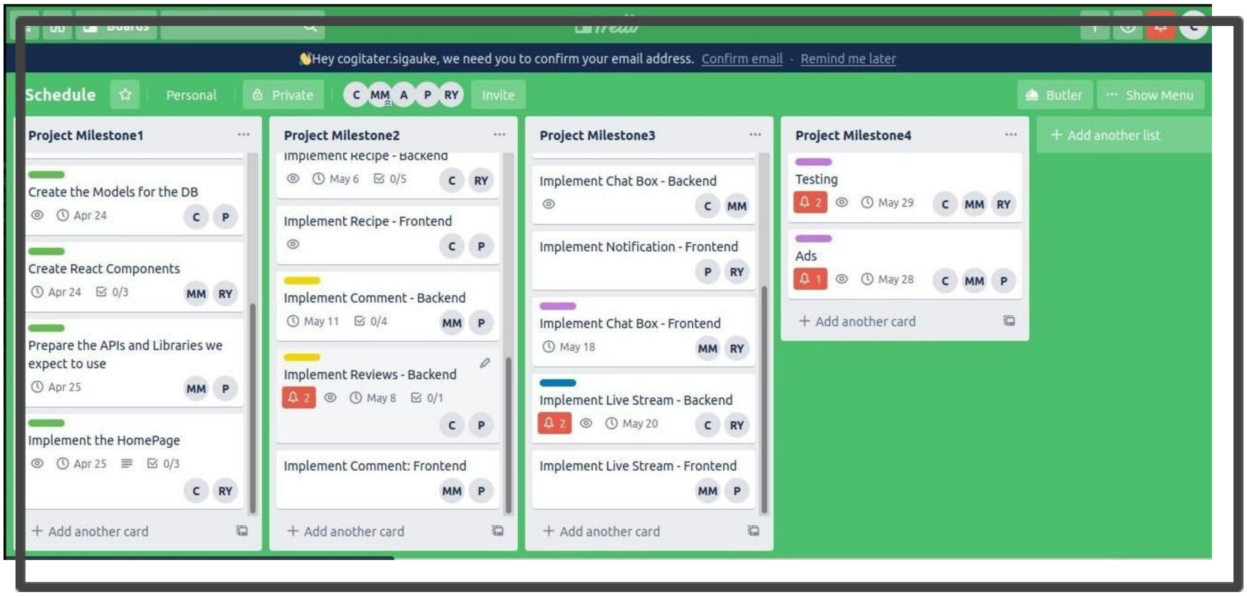
+ Add another card

Project Milestone4

- Testing — 2 — May 29 — C MM RY
- Ads — 1 — May 28 — C MM P

+ Add another card

+ Add another list



👋Hey cogitater.sigauke, we need you to confirm your email address. Confirm email · Remind me later

Schedule ☆ Personal Private C MM A P RY Invite Butler ··· Show Menu

Project Milestone1

- Create the Models for the DB — Apr 24 — C P
- Create React Components — Apr 24 — 0/3 — MM RY
- Prepare the APIs and Libraries we expect to use — Apr 25 — MM P
- Implement the HomePage — Apr 25 — 0/3 — C RY

+ Add another card

Project Milestone2

- Implement Recipe - Backend — May 6 — 0/5 — C RY
- Implement Recipe - Frontend — C P
- Implement Comment - Backend — May 11 — 0/4 — MM P
- Implement Reviews - Backend — 2 — May 8 — 0/1 — C P
- Implement Comment: Frontend — MM P

+ Add another card

Project Milestone3

- Implement Chat Box - Backend — C MM
- Implement Notification - Frontend — P RY
- Implement Chat Box - Frontend — May 18 — MM RY
- Implement Live Stream - Backend — 2 — May 20 — C RY
- Implement Live Stream - Frontend — MM P

+ Add another card

Project Milestone4

- Testing — 2 — May 29 — C MM RY
- Ads — 1 — May 28 — C MM P

+ Add another card

+ Add another list