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, 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 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 3 in order for us to be able to use Glyphicon library.

The back end of our website will use Spring Boot 2.2 and Microsoft Azure Cosmos MongoDB. 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.

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 - it allows user to search for a meal either by meal name or ingredient it contains- (through ingredients filter). It will also have auto-complete list

NavBar - It has logo(link to home), search bar and some important buttons like- Login, Logout, profile, 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 footer that has a contact information including aboutUs, social media links, phone number, email address and some other info

Recipe- it lets the user add their own recipes

addRecipe:

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

Ingredients: this has a list of ingredients where the user can add or remove from the recipe that they are adding. It has division under it to make it easier for the user to quickly select the ingredients. Some of the divisions are dairy, meat, vegetables, fruits, spices, seasonings etc…

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

editRecipe: edit recipe
favoriteRecipe: like a recipe
deleteRecipe: delete added recipe

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 shown as a star

myCategories: inherited from Recipe component to let the user filter their recipes by different categories
editRecipe: it helps edit the recipe that the user added
myReviews: it has lists of reviews given to each recipes with their ratings
myFavorites: it will have the list of favorite recipes that the user liked or added as favorite

Account:

Login - lets the user login with their google account
Logout - lets the user logout from their account
Followers - lets the user see his/her followers and also the user can click on a block button to block the users following him/her
Following - let the user look at the list of the people that he/she is following.
editProfile- lets the user change password, username, or aboutMe description including their profile picture
Notifications- it tracks news feeds and notifications and lets the user look at the list of notifications he/she has.
Settings: allows the user to deactivate their account or delete their account

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.

Detailed description for object oriented classes

(1) Recipe

a. Recipe - The model class of the recipe object that will be stored in database

- 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
imageCollection	Image	Has Image objects
videoCollection	Video	Has video objects
reviews	Review	Has reviews given to that recipe
comments	Comment	Has comments
favoritedBy	User	This contains a list of users that liked or added this recipe to their favorite list.

- Methods

Name	Parameters	Return values	Description
addImage	Recipe	void	Deletes Recipe from the Database
addVideo	Recipe	void	Saves Recipe to the Database
addReview	recipeId	Recipe	Fetches Recipe with the given ID
addComment	recipeId	Recipe	Updates the 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) The User

- Attributes

Name	Type	Description
ID	String	Describes the user’s Id
name	String	Has user's username
profilePic	String	Has user's profile image which is an Image object
aboutMe	String	Describes user's 'about me' information
notification	Notification	Has all the notifications the user receives
message	Message	Has all the conversations the user has with other users
recipe	Recipe	Has all the recipes the user has posted on his page
userStatus	AccountStatus	Shows the account status if it is either deactivated or normal

Followers	ArrayList<User>	Has a list of users who are following the user
Following	ArrayList<User>	Has a list of users whom the user follow
BlockedByUsers	ArrayList<User>	Has a list of users who are blocked by the user

Methods

Name	Parameters	Return values	Description
sendMessage	Message: message	void	Block method should be defined as the RecipeReposito ry
receiveMessage	-	ArrayList<Message>	AddRecipe the user's information. This will be frequently called from the User use case
receiveNotification	-	ArrayList<Notification>	Update the other user's recipe information. This method will be updated when the other user's add something.
addRecipe	-	boolean	addRecipe the user's information, This will be add recipe to the user
editRecipe	-	boolean	editRecipe the user's information of the current recipe. This will be frequently called from the User use case
deleteRecipe	-	boolean	Delete method should be defined as the Recipe
favoriteRecipe	-	void	Save the user's information of favoriteRecipe. This will be frequently called from the Recipe
commentOnRecipe	-	void	Update the comment user's information. This will be frequently called from the Recipe.
giveReviewToRecipe	-	void	This method helps

b, The User Repository - The class for the interaction with the user model and performing database.
Methods

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

(3) 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
readBoolean	Boolean	shows if the message is read by the receiver or not
image	Image	has the image to be sent to the receiver

-Methods

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	is used to send message to the receiver user

(4) Notification

- Attributes

Name	Type	Description
ID	String	An id that uniquely identifies a notification object
notificationText	String	The attribute holds the notification text description
readBoolean	Boolean	This boolean would help to track notifications that are already read and the ones that are not visited yet

(5) Comment

- Attributes

Name	Type	Description

ID	String	Describes the user's Id for comment
likesCount	Int	The number of likeCount that the comment received
recipe	Recipe	Has all the recipes the user has posted on page
commentLikedBy	User	The user comment like the comment from other user

-Methods

Name	Parameters	Return	Description
addComment	-	void	addComment the user's information, This will be add comment to the other user
deleteComment	-	void	Delete method is used to delete a text comment which is already sent
editComment	-	void	editComment method is used to edit the text Comment to user after the comment being posted

6) 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 could track the reviews that the recipe received

7) Image

- Attributes

Name	Type	Description
ID	String	The id uniquely identifies the image object
imgCap	String	This has a caption of the image

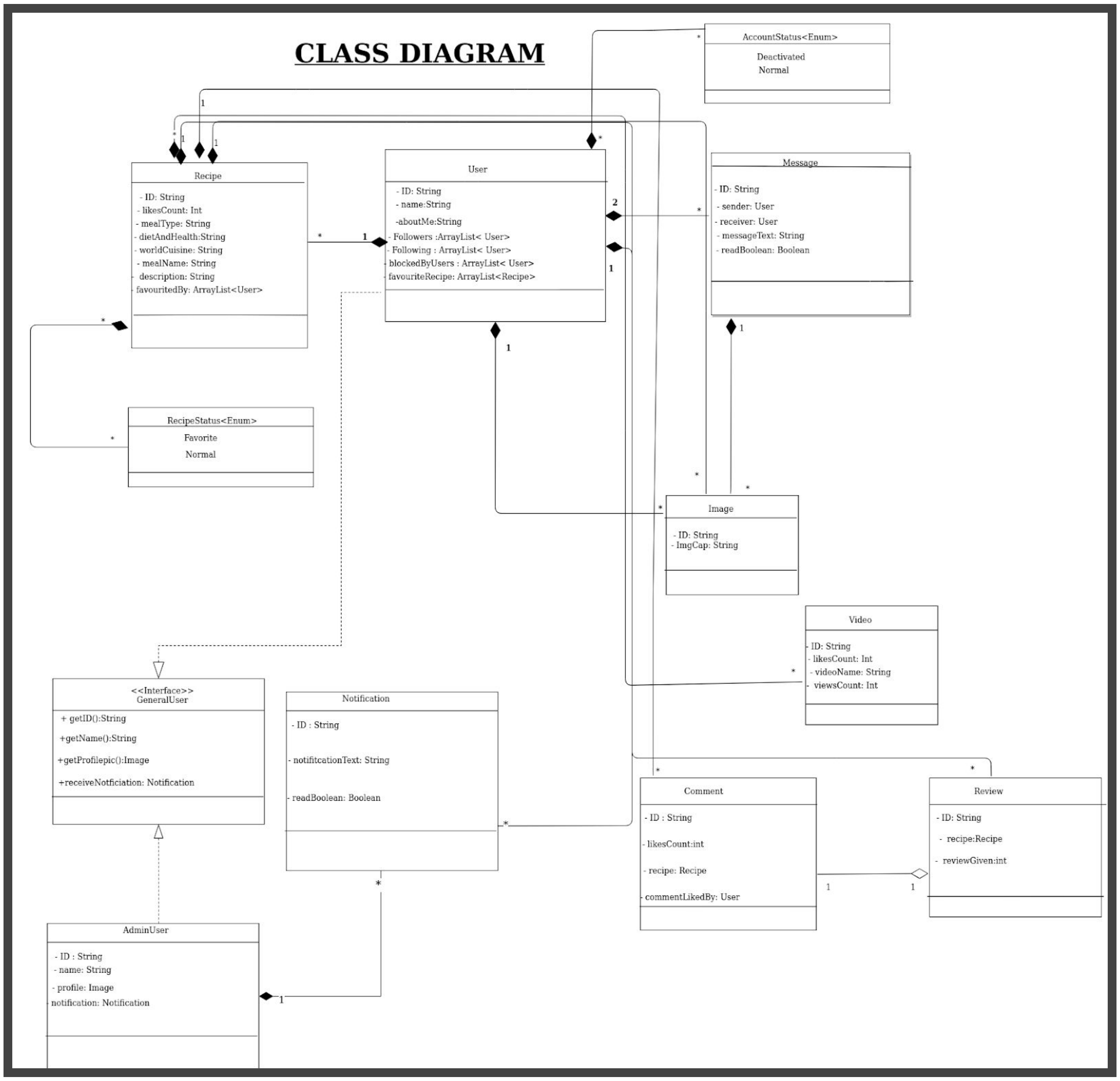
8) Video

- Attributes

Name	Type	Description
ID	String	The ID value will be the key that defines a user
likesCount	String	The number of like that the video received
videoName	String	The user input of the videoName from the User use case
videoCount	Int	The number of videoCount that the video received

1.3 UML Class Diagram

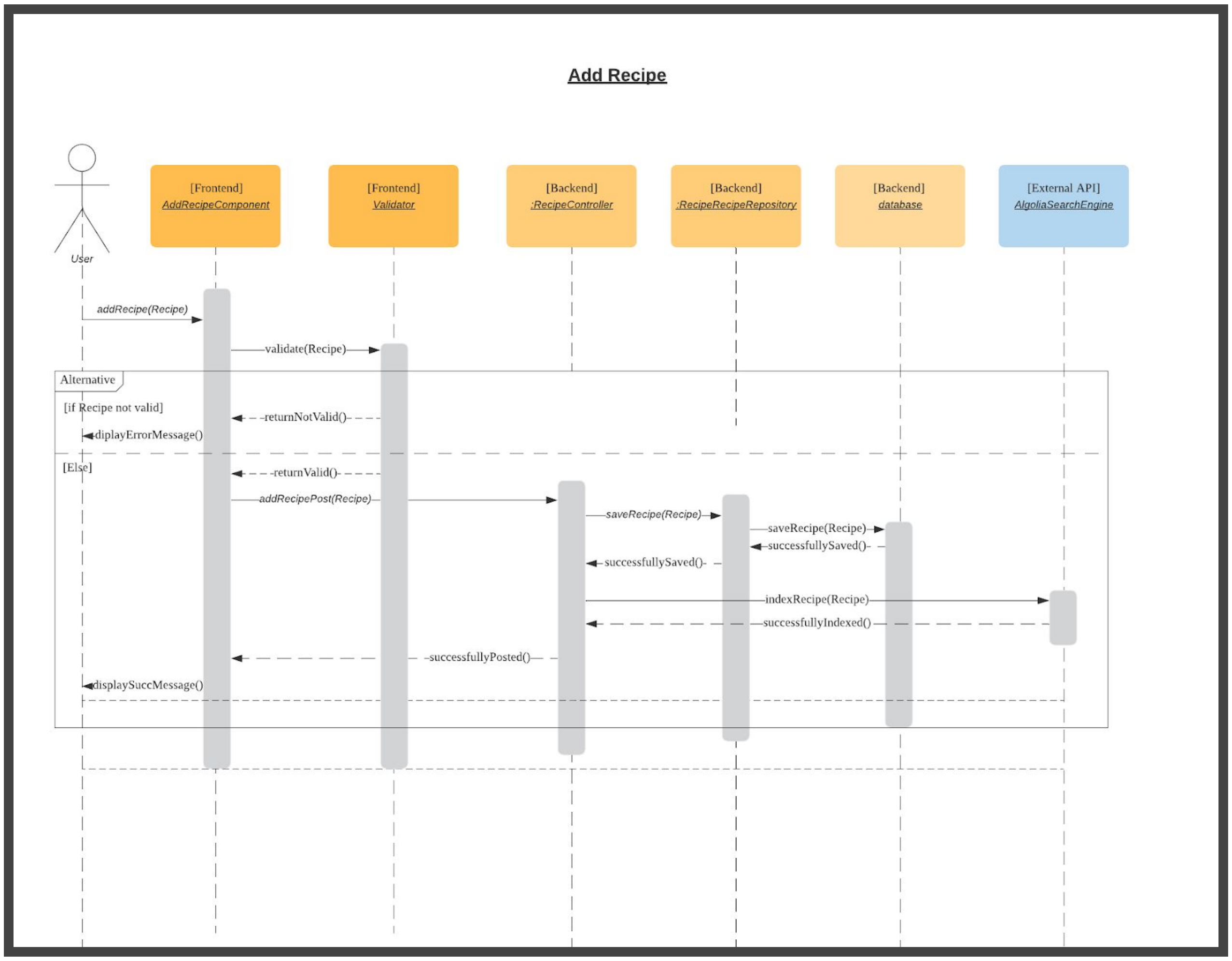
https://app.diagrams.net/#G1r1xh5914B5bmROgwju_OvgGK0r-5If2y



1.4 UML Sequence Diagrams

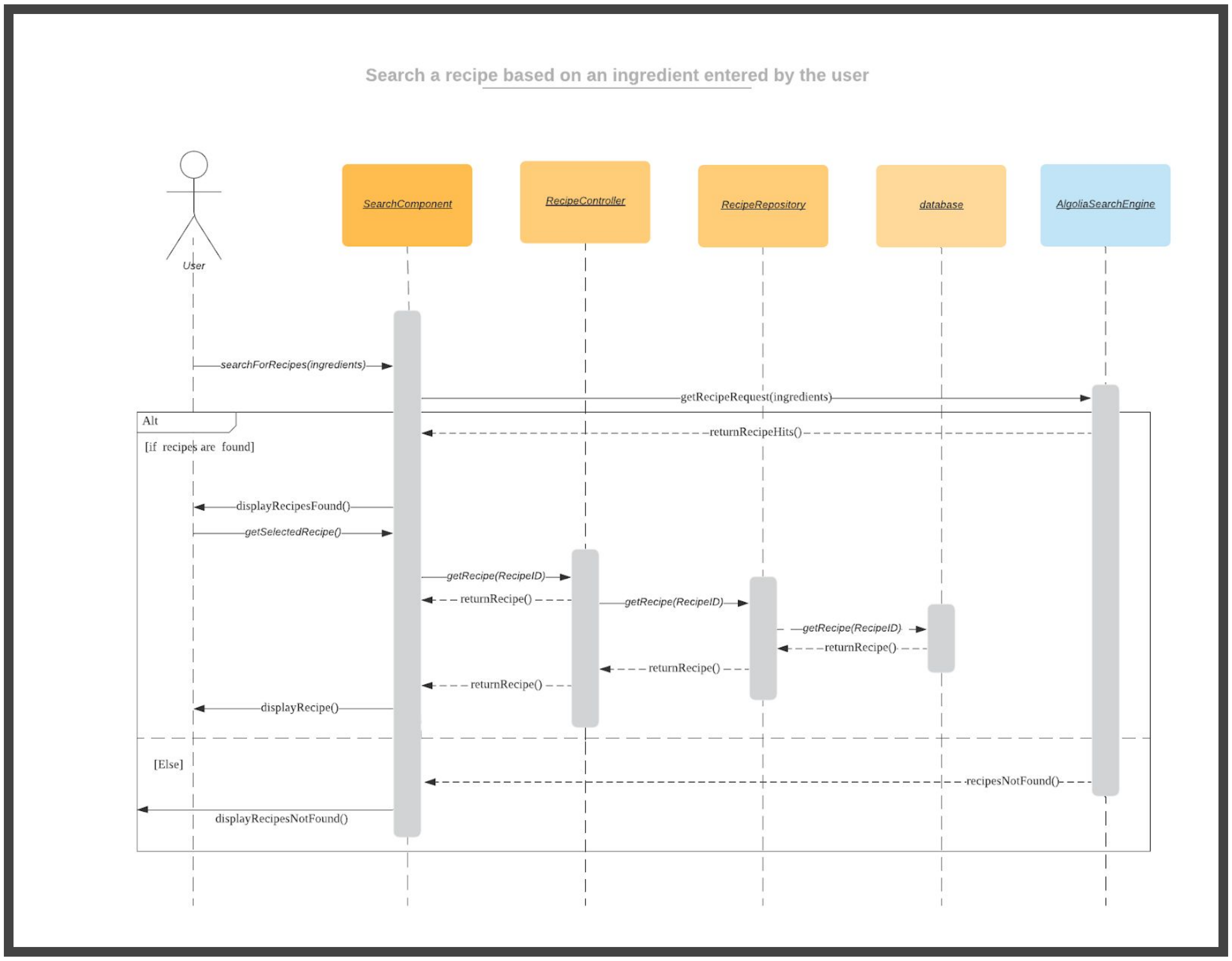
1.4.1 ADD Recipe

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:

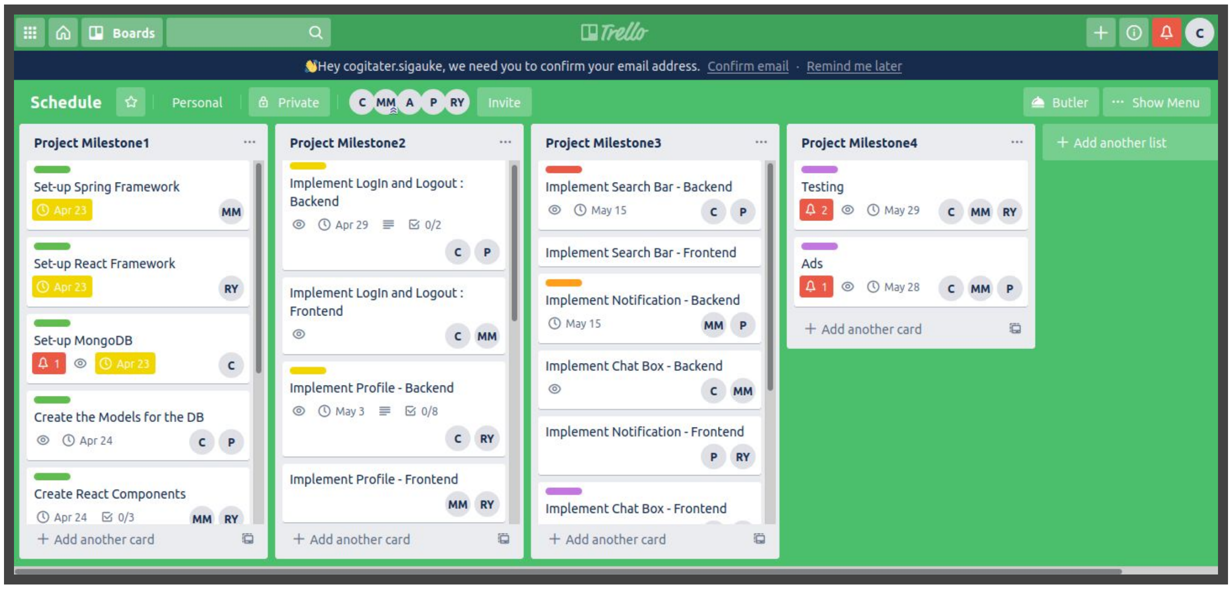
We will be using Microsoft Azure for the full development of our application. For this, we will be using the Azure Cosmos DB for MongoDB API as our database. In addition, both the backend spring boot application and the react front end of our application will be deployed to Microsoft Azure.

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

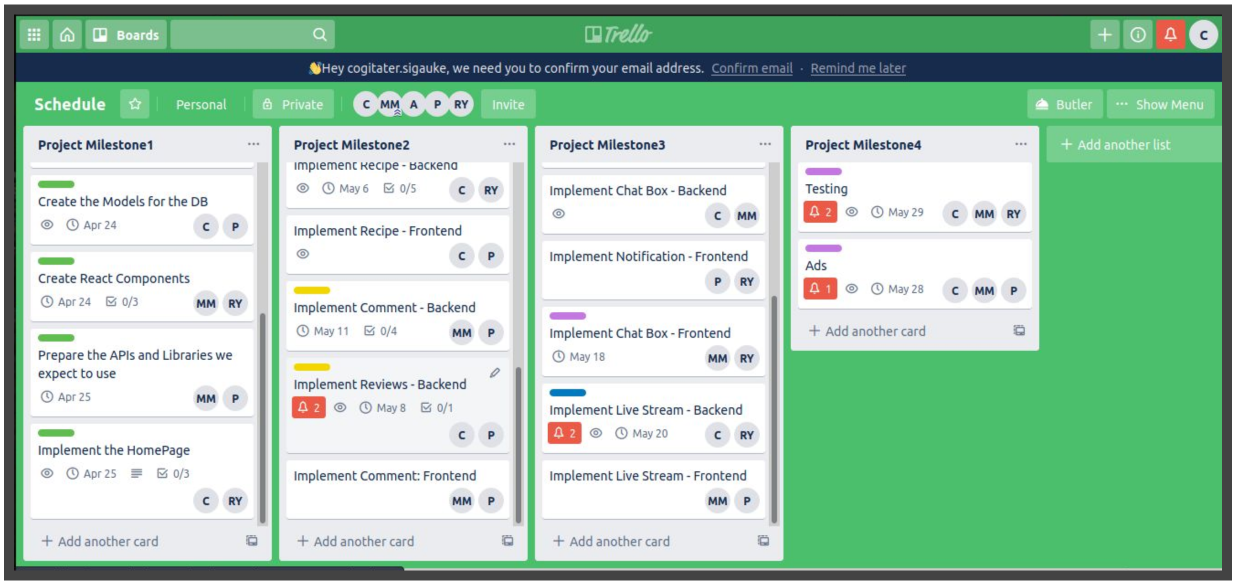
Project Milestone1
- Set-up Spring Framework — Apr 23 — MM
- Set-up React Framework — Apr 23 — RY
- Set-up MongoDB — 1 — Apr 23 — C
- Create the Models for the DB — Apr 24 — C P
- Create React Components — Apr 24 — 0/3 — MM RY

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

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

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

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