



# **SOFTWARE DESIGN DOCUMENT**

**Presentation by Cogitater Sigauke,  
Merry Mekonnen,  
Redi Negash,  
Pyungkang Hong**



# MyRecipes

A recipe sharing web application

A recipe search web application

An application develop your cooking skills

An application place to meet new friends

A place to be inspired



# Problem

A recipe sharing web application

People want to learn how to cook

An application develop your cooking skills

An application place to meet new friends

A place to be inspired



# and Technologies

Spring Boot

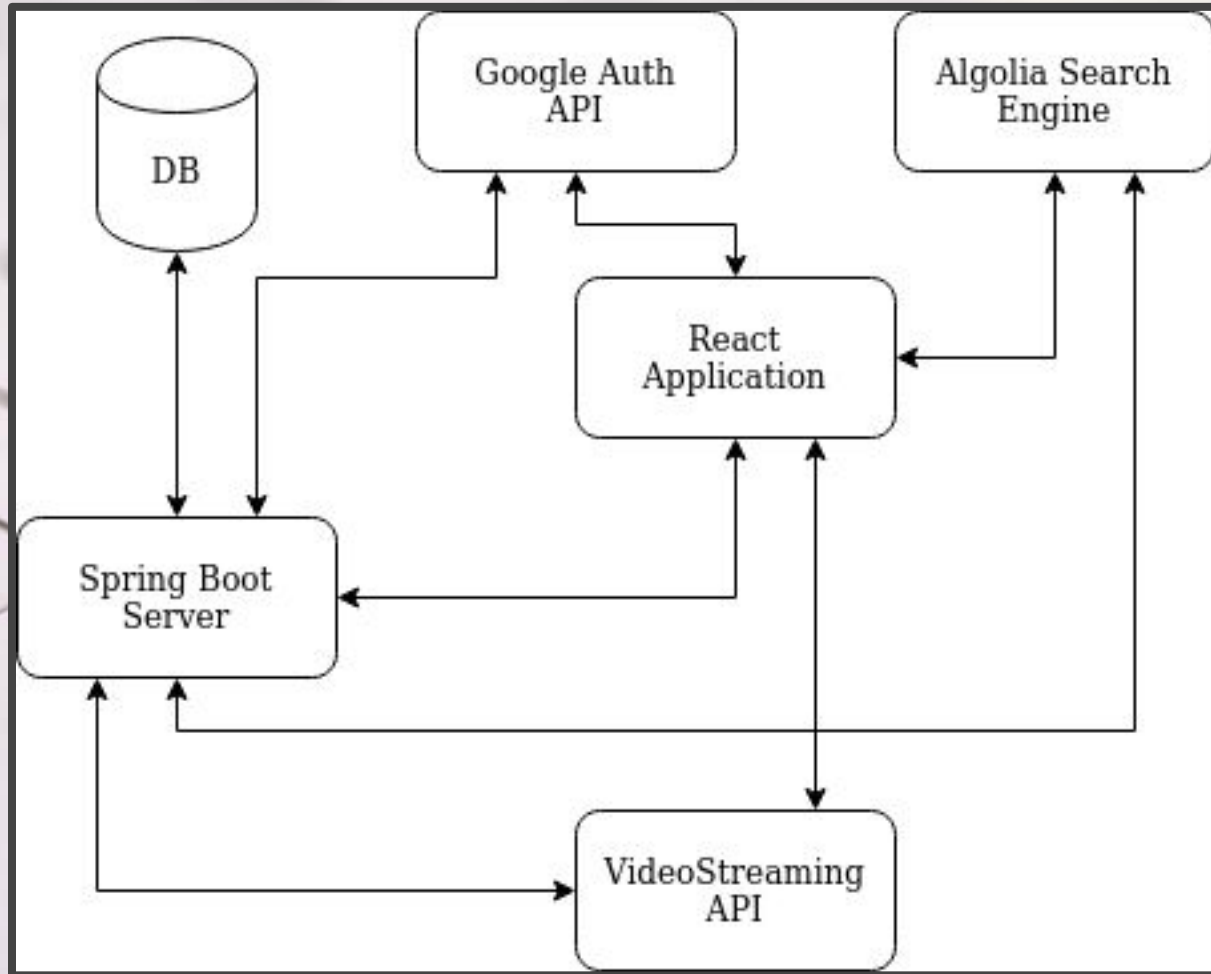
React.js

Algolia Search Engine

WebRCT

Google Auth

# General Architecture View





# Major Features

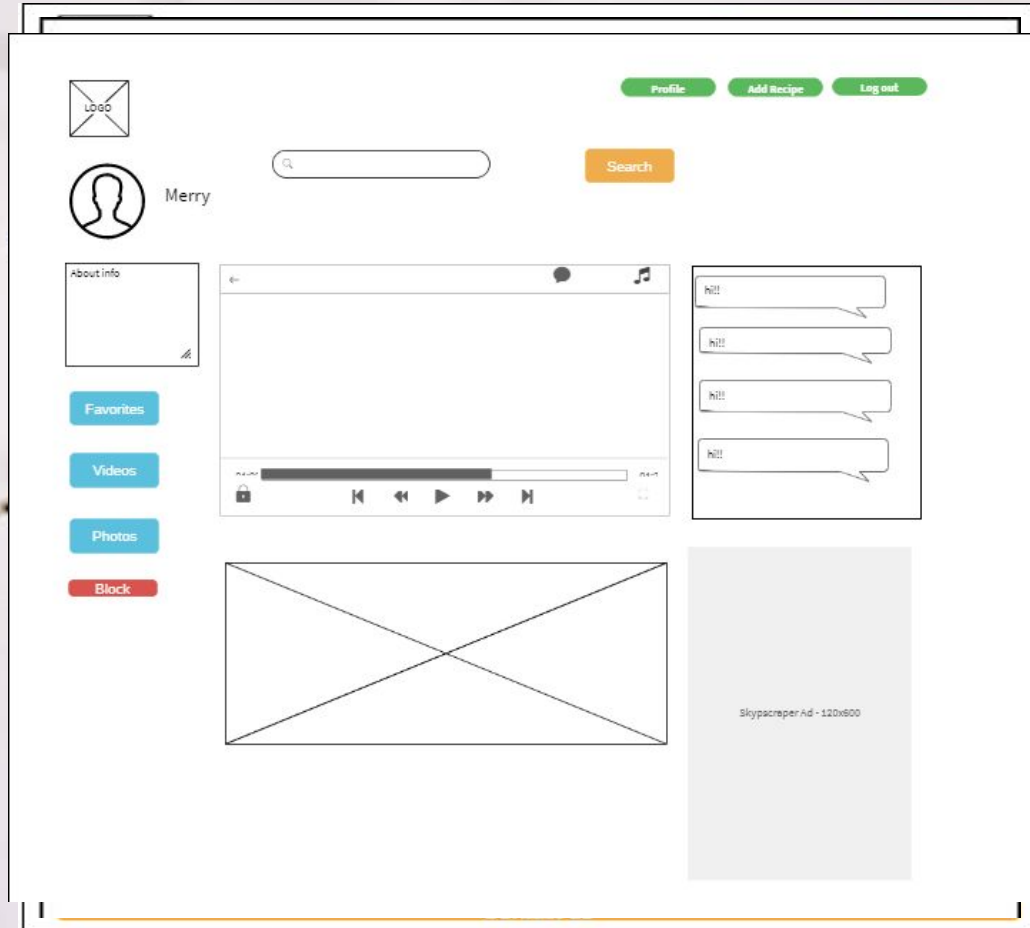
General Text Based Search with different filters

Display of uploaded recipes in different categories

Live streaming

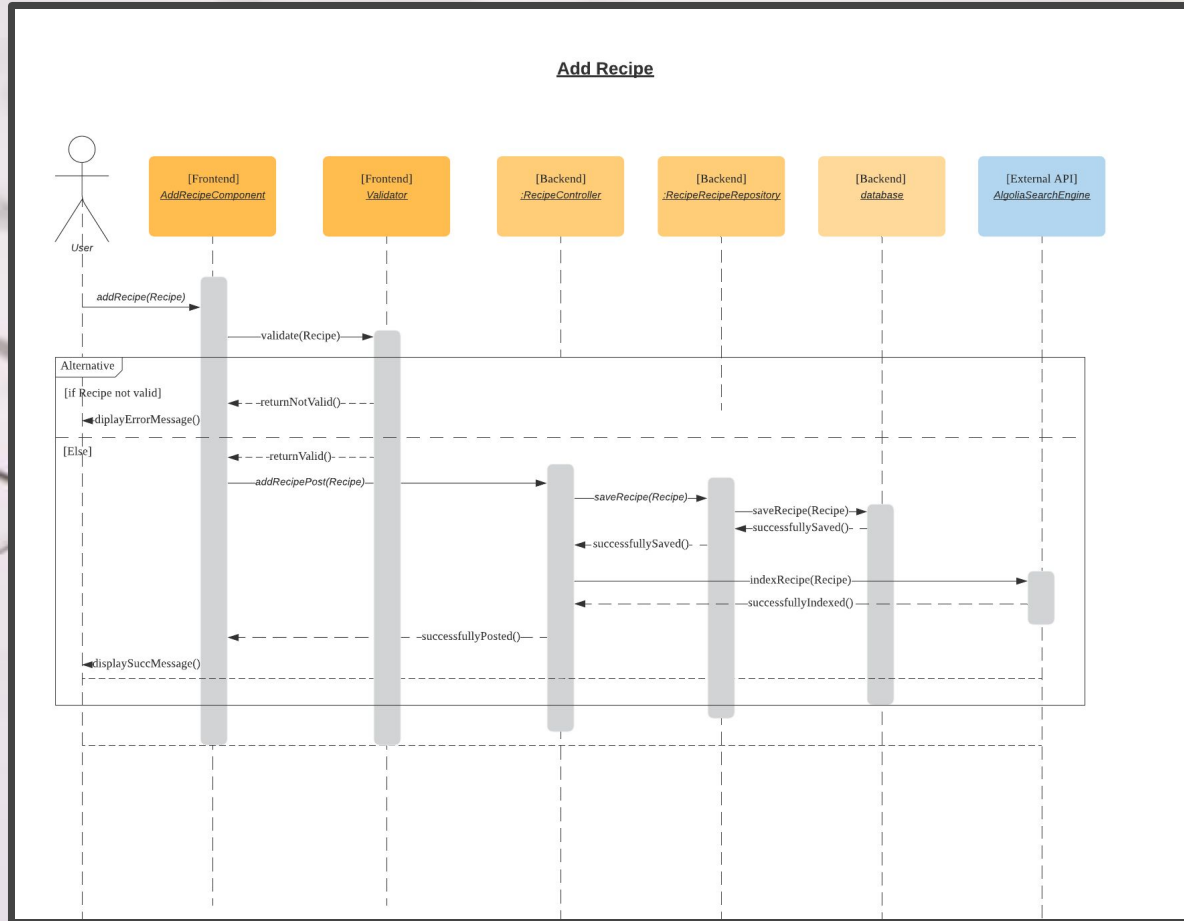
Messages and inbox

# User Interface Diagram





# Sequence Diagram





## CLASS DIAGRAM

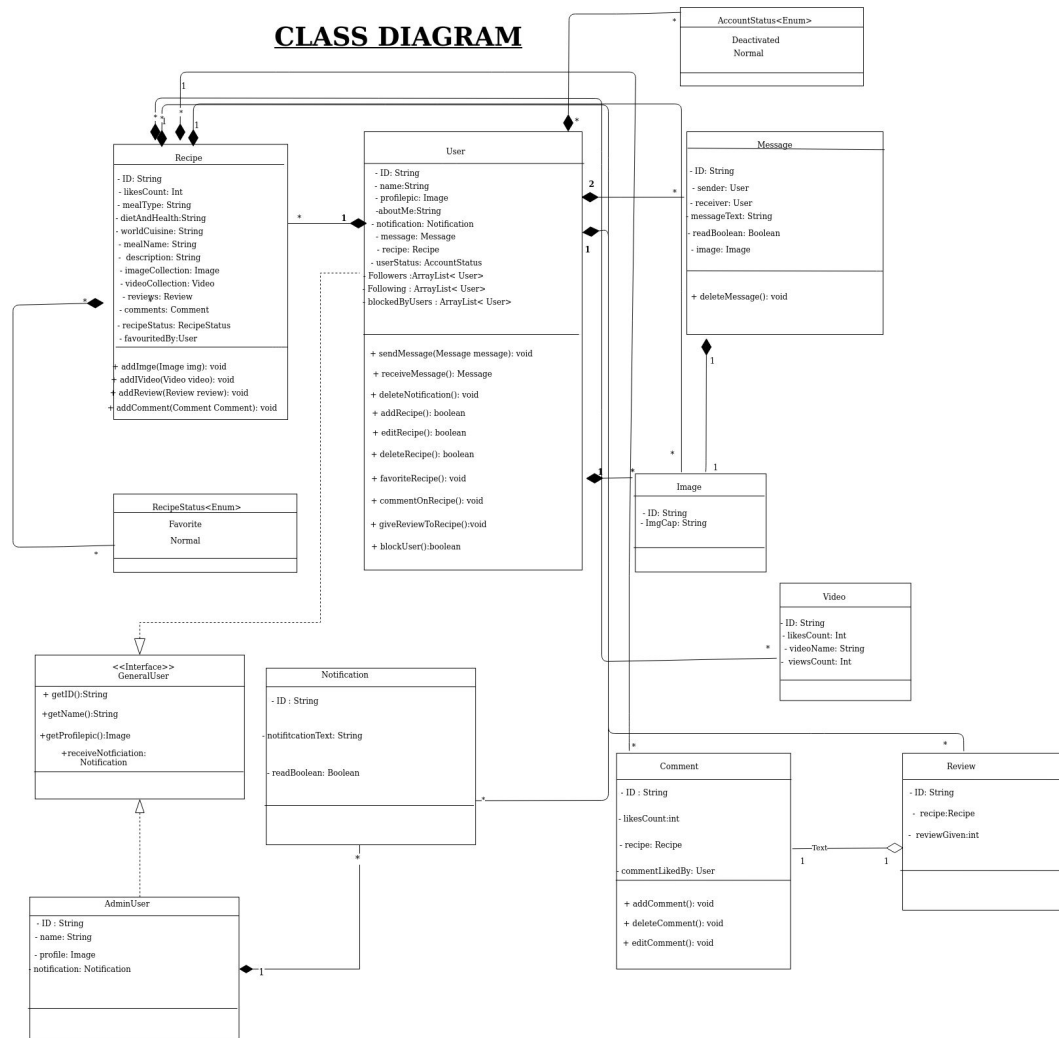
```

classDiagram
    class Recipe {
        -ID: String
        -likesCount: Int
        -mealType: String
        -dietAndHealth: String
        -worldCuisine: String
        -mealName: String
        -description: String
        -imageCollection: Image
        -videoCollection: Video
        -reviews: Review
        -comments: Comment
        -recipeStatus: RecipeStatus
        -favouriteBy: User
        +addImage(Image img): void
        +addVideo(Video video): void
        +addReview(Review review): void
        +addComment(Comment Comment): void
    }
    class User {
        -ID: String
        -name: String
        -profilePic: Image
        -aboutMe: String
        -notification: Notification
        -recipe: Recipe
        -userStatus: AccountStatus
        -followers: ArrayList<User>
        -following: ArrayList<User>
        -blockedByUsers: ArrayList<User>
        +sendMessage(Message message): void
        +receiveMessage(): Message
        +deleteNotifications(): void
        +addRecipe(): boolean
        +editRecipe(): boolean
        +deleteRecipe(): boolean
        +favoriteRecipe(): void
        +commentOnRecipe(): void
        +giveReviewToRecipe(): void
        +blockUser(): boolean
    }
    class Message {
        -ID: String
        -sender: User
        -receiver: User
        -messageText: String
        -readBoolean: Boolean
        -image: Image
        +deleteMessage(): void
    }
    class Image {
        -ID: String
        -imgCap: String
    }
    class Video {
        -ID: String
        -likesCount: Int
        -videoName: String
        -viewsCount: Int
    }
    class Comment {
        -ID: String
        -likesCount: Int
        -recipe: Recipe
        -commentLikedBy: User
        +addComment(): void
        +deleteComment(): void
        +editComment(): void
    }
    class Review {
        -ID: String
        -recipe: Recipe
        -reviewGiven: Int
    }
    class Notification {
        -ID: String
        -notificationText: String
        -readBoolean: Boolean
    }
    class AdminUser {
        -ID: String
        -name: String
        -profile: Image
        -notification: Notification
    }
    class GeneralUser {
        <<Interface>>
        +getID():String
        +getName():String
        +getProfilePic():Image
        +receiveNotification: Notification
    }
    class AccountStatus {
        <<Enum>>
        Deactivated
        Normal
    }
    class RecipeStatus {
        <<Enum>>
        Favorite
        Normal
    }

    Recipe "1" -- "*" User
    Recipe "1" -- "*" Message
    Recipe "1" -- "*" Image
    Recipe "1" -- "*" Video
    User "1" -- "*" Message
    User "1" -- "*" Image
    User "1" -- "*" Video
    User "1" -- "*" Notification
    Comment "1" -- "*" Review
    AdminUser <|-- GeneralUser
    GeneralUser <|-- User
    GeneralUser <|-- AdminUser
    AccountStatus <|-- User
    RecipeStatus <|-- Recipe
    
```

The diagram illustrates the relationships between various components of a recipe application. Key elements include:

- Recipe**: The central entity, associated with **User**, **Message**, **Image**, **Video**, **Comment**, **Review**, and **Notification**. It includes attributes like `-ID`, `-likesCount`, and `-recipeStatus`.
- User**: Manages **Message**, **Image**, **Video**, and **Notification**. It includes attributes like `-name`, `-profilePic`, and `-followers`.
- Message**: Represents communication between users, associated with **User** and **Image**.
- Image** and **Video**: Media uploads associated with **Recipe** and **User**.
- Comment** and **Review**: User feedback on recipes, with **Review** specifically linked to a **Recipe**.
- Notification**: System-generated alerts associated with **User**.
- Interfaces and Enums**: **GeneralUser** defines methods for user interaction. **AccountStatus** and **RecipeStatus** define possible states for users and recipes.



# Design Decisions

Choosing Google Auth over implementing our own login system

Using algolia search engine rather than writing our own

Choosing video streaming API over coding our own from scratch

Choosing mongoDb over MySql

# Schedule and Deliverables

The screenshot displays a Trello board titled "Schedule" in a green-themed interface. The board is organized into four columns, each representing a project milestone. The top navigation bar includes icons for home, boards, and search, along with a notification banner that reads: "Hey cogitater.sigauke, we need you to confirm your email address. [Confirm email](#) · [Remind me later](#)".

Below the navigation bar, the board is categorized by "Personal" and "Private" tabs, with a filter set to "C MM A P RY". A "Butler" button and a "Show Menu" option are also visible.

The four milestones and their associated tasks are as follows:

- Project Milestone1**
  - Create the Models for the DB (Due: Apr 24, Assignees: C, P)
  - Create React Components (Due: Apr 24, Progress: 0/3, Assignees: MM, RY)
  - Prepare the APIs and Libraries we expect to use (Due: Apr 25, Assignees: MM, P)
  - Implement the HomePage (Due: Apr 25, Progress: 0/3, Assignees: C, RY)
- Project Milestone2**
  - Implement Recipe - Backend (Due: May 6, Progress: 0/5, Assignees: C, RY)
  - Implement Recipe - Frontend (Assignees: C, P)
  - Implement Comment - Backend (Due: May 11, Progress: 0/4, Assignees: MM, P)
  - Implement Reviews - Backend (Due: May 8, Progress: 0/1, Assignees: C, P)
  - Implement Comment: Frontend (Assignees: MM, P)
- Project Milestone3**
  - Implement Chat Box - Backend (Assignees: C, MM)
  - Implement Notification - Frontend (Assignees: P, RY)
  - Implement Chat Box - Frontend (Due: May 18, Assignees: MM, RY)
  - Implement Live Stream - Backend (Due: May 20, Progress: 0/2, Assignees: C, RY)
  - Implement Live Stream - Frontend (Assignees: MM, P)
- Project Milestone4**
  - Testing (Due: May 29, Progress: 0/2, Assignees: C, MM, RY)
  - Ads (Due: May 28, Progress: 0/1, Assignees: C, MM, P)

Each milestone column has a "+ Add another card" button at the bottom. A "+ Add another list" button is located on the right side of the board.



**Thank You !!!**