

# Team Grumio

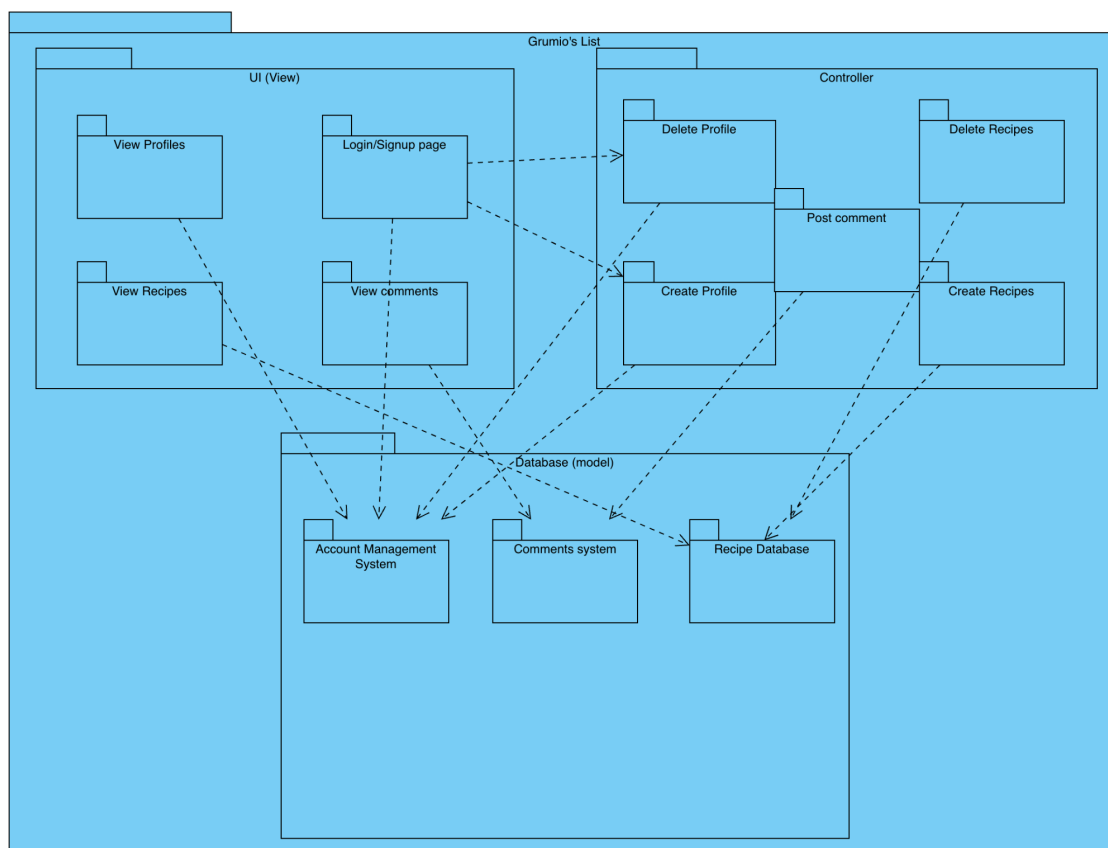
Dylan Andrews, Joe Binette, Oz Cordes, Raist Cotroneo, Odin York

Grumio's List: Recipe-Sharing Web Application (Lifestyle/Social Media)

## Architecture:

Our app will employ the model-view-controller (MVC) architecture. This architecture is the best one that will suit our web application. The model of the app will be done through JavaScript and Spring.io using Java. This will take care of the actual changes being made by a user and store information based on the changes in the database. Our view is all of our frontend using React, Typescript, and CSS which will give the user a UI to be able to log in, view profiles, view recipes, etc. Our controller will be the web browser where the users will be able to make changes like creating profiles, creating/modifying recipes, as well as sharing them across a network.

## UML Package Diagram:



## **Use Case Descriptions:**

### Use Case 1: Create Profile

Description: User creates an account on the application

Related Use Cases: Extends to UC2, Extends to UC3

System under development: Grumio's List

Actors: user. System

Goals: User is able to create an account when supplying valid input.

Steps:

User clicks the "Sign Up" button on the homepage.

System prompts the user to input a valid username and password.

User inputs requested information.

System validates username to verify availability.

System validates password to match security requirements (length, characters).

System notifies the user that their choices are valid.

User clicks the "create account" button to finalize account creation.

System creates an account entry in the associated database.

### Use Case 2: Attempting to Create a Profile with Taken Username

Description: User attempts to create an account using a username that is already in the DB

Related Use Cases: Extension of UC1

System under development: Grumio's List

Actors: user, system

Goals: user is not able to create an account when username is invalid.

Steps:

User clicks the "Sign Up" button on the homepage.

System prompts the user to input a valid username and password.

User inputs requested information.

System checks account management system database for availability.

System finds that the username has already been taken.

System notifies the user that their chosen username is unavailable.

### Use Case 3: Creating Invalid Password

Description: User supplies a password that does not meet security requirements.

Related Use Cases: Extension of UC1

System under development: Grumio's List

Actors: user. System

Goals: user is not able to create an account when password is invalid.

Steps:

User clicks the "Sign Up" button on the homepage.

System prompts the user to input a valid username and password.

User inputs requested information.

System validates username to verify availability.

System checks password against security requirements (length, characters).

System finds that password does not meet requirements.

System notifies user that their chosen password does not meet security requirements.

#### Use Case 4: Create Recipe

Description: User creates a recipe.

System under development: Grumio's List

Actors: user, system

Goals: user is able to create a recipe.

Steps:

User logs in to their account.

System brings up user homepage.

User clicks "Create Recipe" button.

System brings up recipe creation page.

User supplies recipe title.

User supplies ingredients list.

User supplies instructions for recipe preparation and cooking.

User supplies pictures of recipe.

User clicks "Post Recipe" button.

System validates that required fields have been filled (title, ingredients, steps).

System adds entry for recipe to the recipe database.

System notifies user that their recipe has been uploaded successfully.

#### Use Case 5: View Profile

Description: User views another user's profile.

System under development: Grumio's List

Actors: user (generalization; userA, userB), system

Goals: User is able to view the profile of another user.

Steps:

userA searches the username of userB.

System searches the account database for matching entries.

A) System returns a matching entry for userA's search.

B) System fails to find a matching entry and notifies userA.

If A:

System renders profile view of userB profile to userA.

If B:

System returns empty search results page.

#### Use Case 6: Comment on Recipe

Description: User comments on a recipe

System under development: Grumio's List

Actors: user (generalization; userA, userB), system

Goals: User is able to leave a comment on a recipe, given their input is not malicious.

Steps:

userA navigates to a recipe created by userB.

userA writes a comment and clicks post.

System checks userA comment for malicious input (spam links, harassment, etc.)

A) System validates userA comment.

B) System fails to validate userA comment.

If A:

System stores userA comment in the comments database.

System notifies userB of a new comment on their recipe.

If B:

System notifies userA that their comment could not be posted.

#### Use Case 7: View Comment

Description: User checks comments on their recipe

System under development: Grumio's List

Actors: user, system

Goals: User is able to view comments on their recipe.

Steps:

User navigates to a recipe they created.

System displays user comments in the comment section on the recipe page.

#### Use Case 8: Delete Comment

Description: User deletes a comment on their recipe

System under development: Grumio's List

Actors: user (generalization; userA, userB), system

Goals: userA is able to delete an unwanted comment from userB on their recipe.

Steps:

userA navigates to a recipe they created.

System displays user comments in the comment section on the recipe page.

userA clicks the dropdown menu for an unwanted comment by userB.

userA selects the Delete Comment button.

System prompts userA for intention to delete the comment.

A) userA confirms intent to delete the comment.

B) userA cancels the Delete Comment process.

If A:

System prompts userA to ask if they would like to block userB.

1) userA confirms they want to block userB.

2) userA denies they want to block userB.

If 1:

UC9 process takes place.

If 2:

UC9 process is canceled.

System deletes userB's comment from userA's recipe.

If B:

UC8 process is canceled.

## Use Case 9: Block User

Description: User blocks another user

System under development: Grumio's List

Actors: user (generalization; userA, userB), system

Goals: User is able to block another user from viewing their profile and recipes, as well as commenting on their recipes.

Steps:

UserA navigates to the profile of UserB.

UserA opens the dropdown menu on UserB's profile.

UserA selects the Block button.

System prompts userA to confirm their selection.

A) UserA confirms intent to block.

B) UserA cancels Block process.

If A:

System adds userB to userA's block list.

System notifies userA that userB was blocked successfully.

If B:

System returns to the profile page of userB.

#### Use Case 10: Delete Recipe

Description: User deletes recipe

System under development: Grumio's List

Actors: user, system

Goals: User is able to delete any of their recipes.

Steps:

User navigates to a recipe they created.

User opens the dropdown menu for the recipe and clicks the Delete button.

System prompts user to confirm intent to delete:

A) User confirms intent.

B) User cancels Delete process.

If A:

System removes recipe entry from recipe database.

System notifies user that the recipe was deleted successfully.

If B:

System returns to recipe page.

## Use Case Models:

