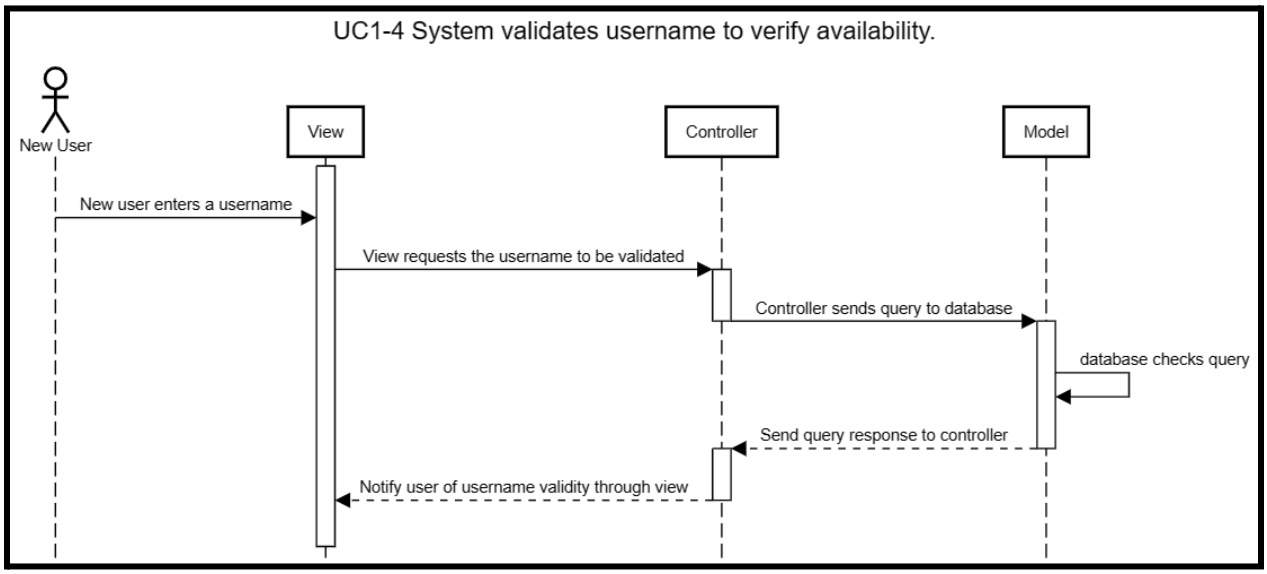
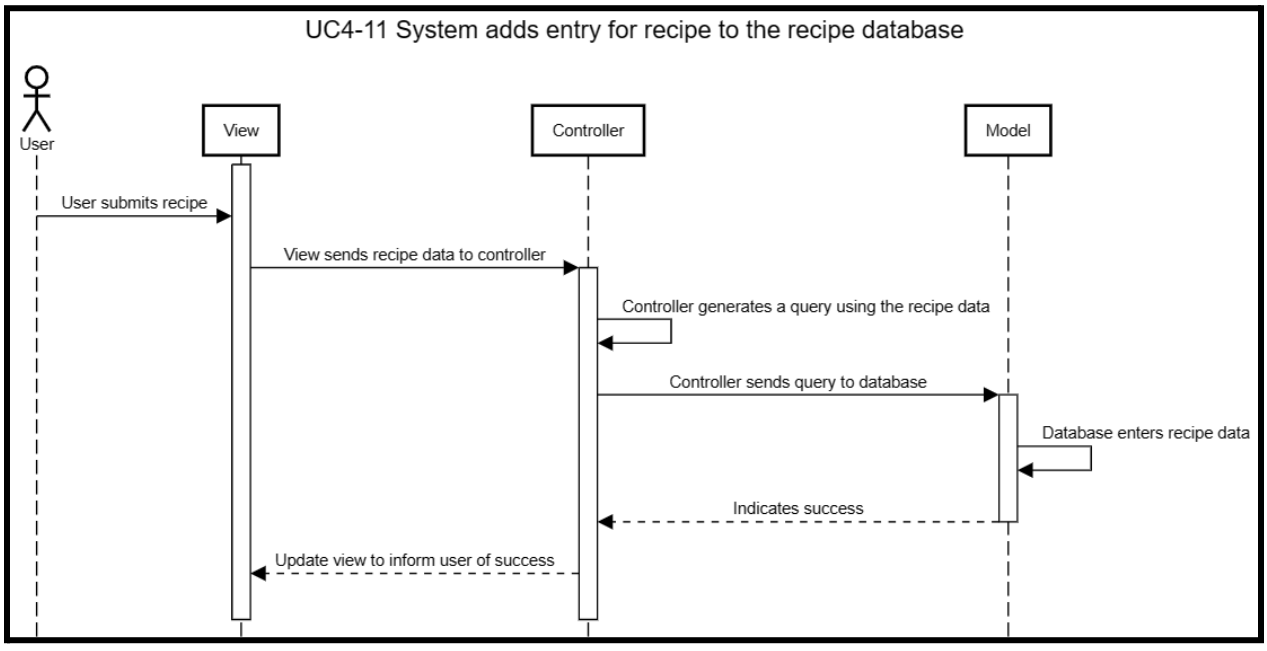
Team Grumio

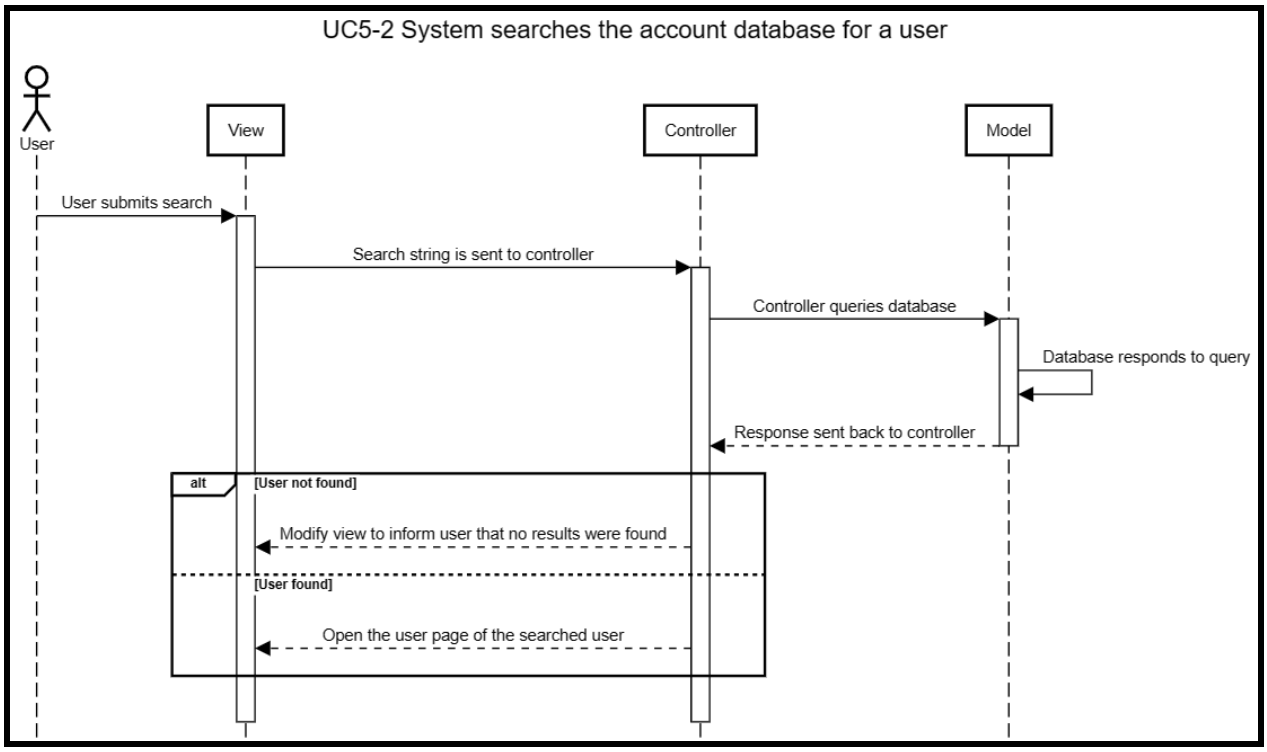
Dylan Andrews, Joe Binette, Oz Cordes, Raist Cotroneo, Odin York

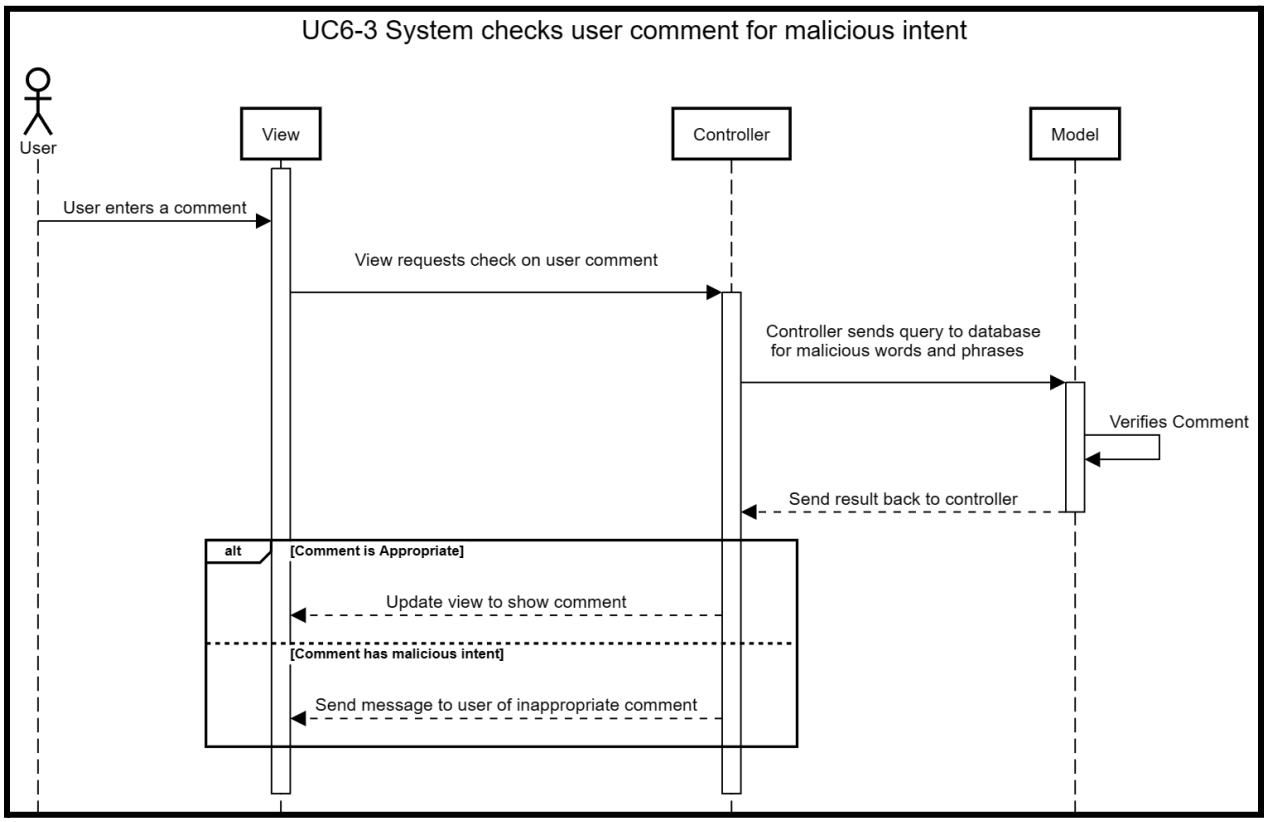
Grumio's List: Recipe-Sharing Web Application (Lifestyle/Social Media)

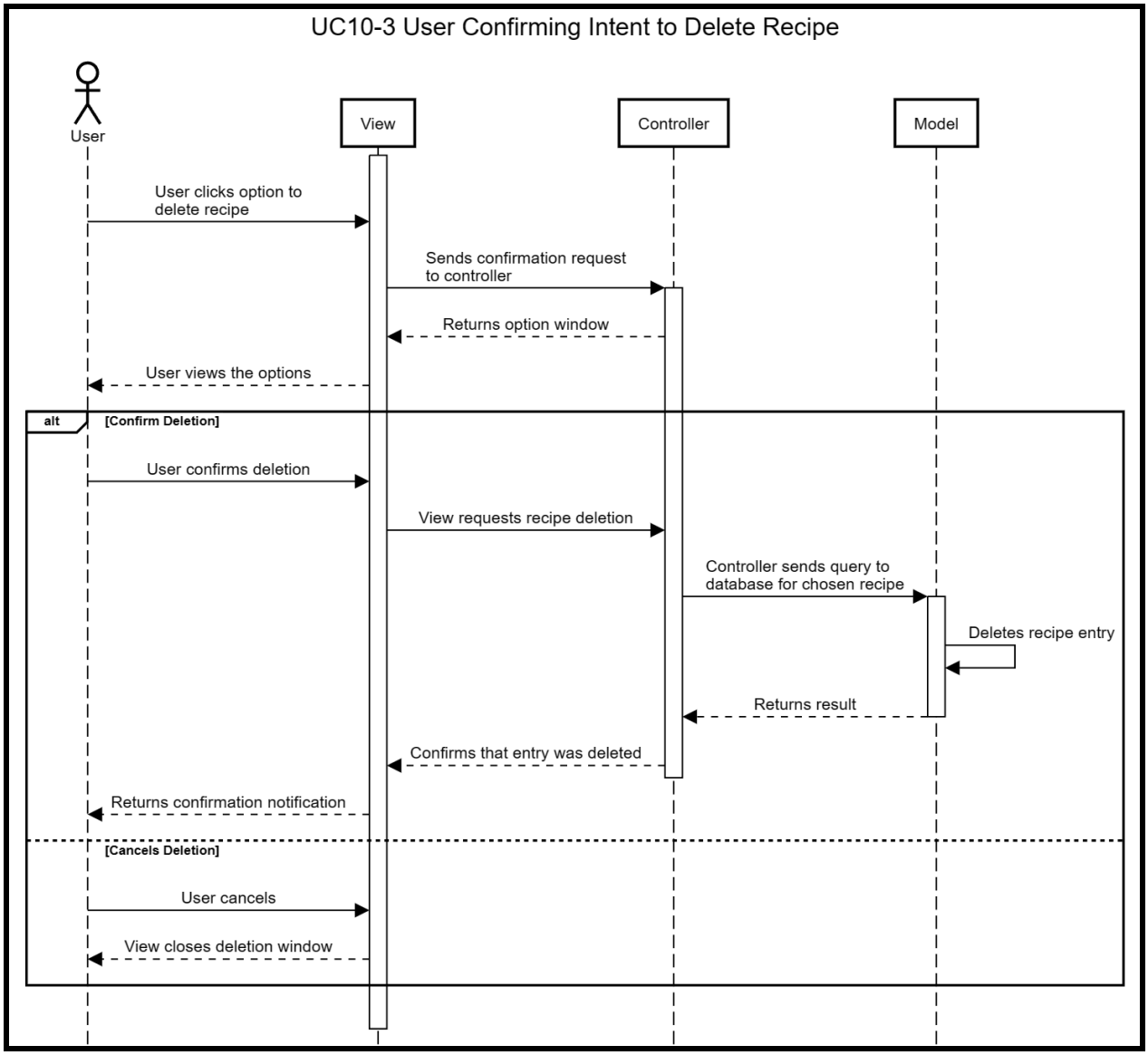
Analysis Sequence Diagrams:

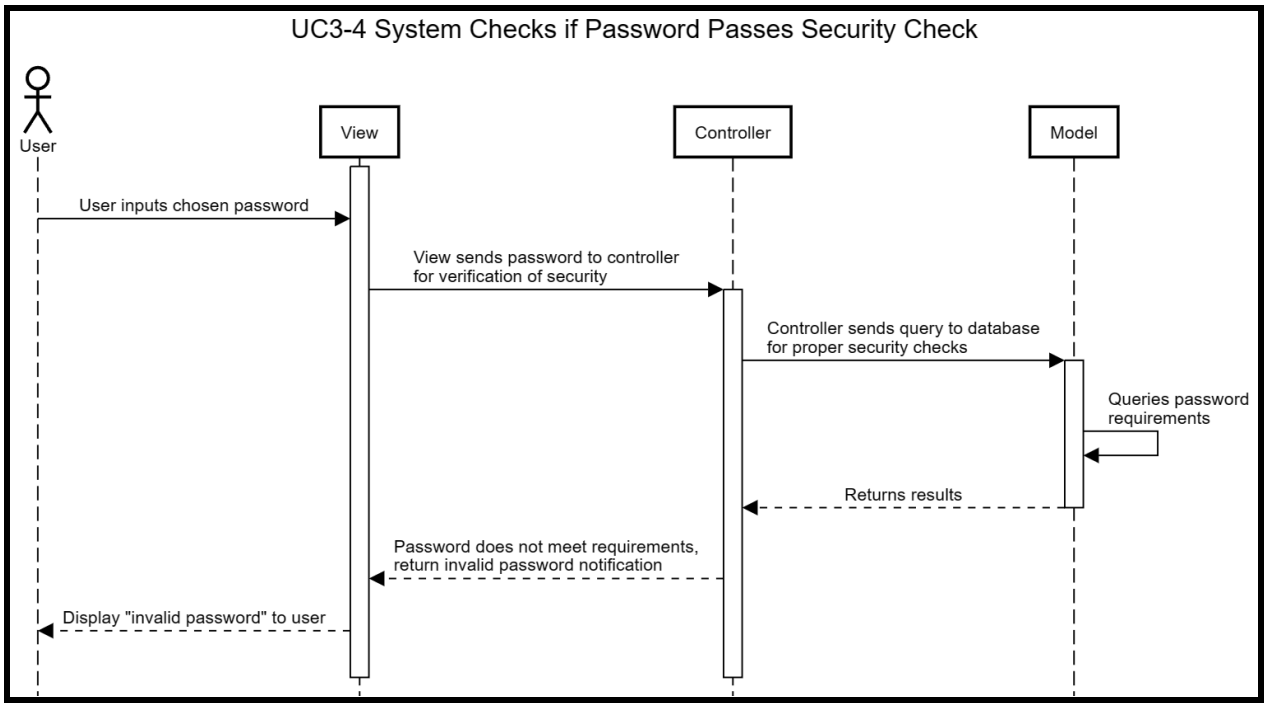


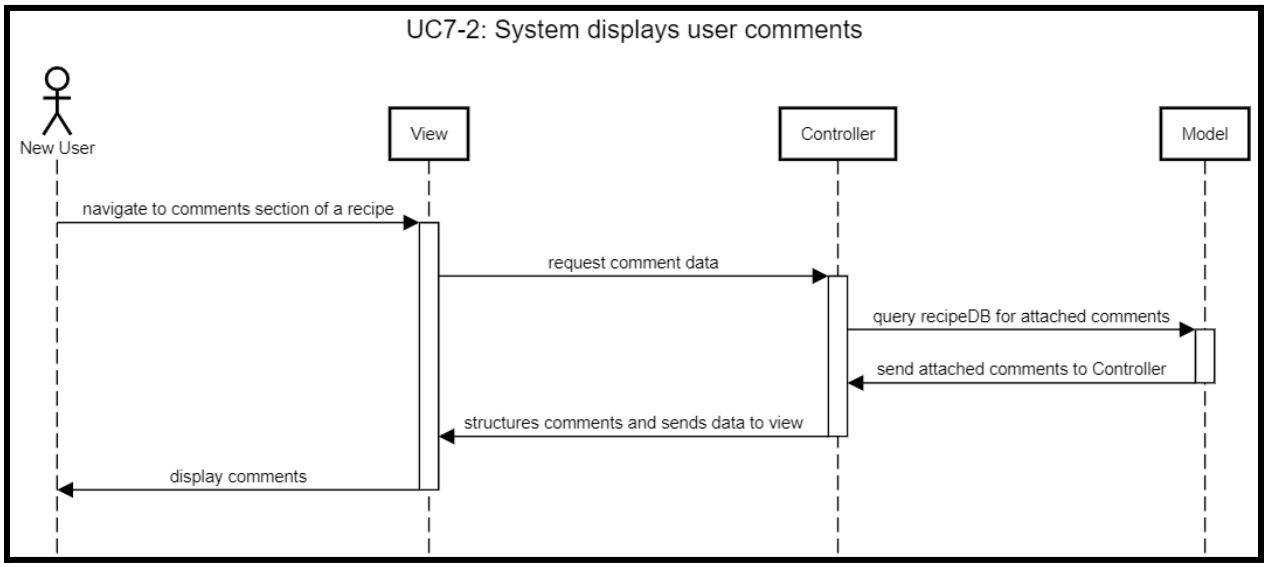



UC5-2 System searches the account database for a user
User
View
Controller
Model
User submits search
Search string is sent to controller
Controller queries database
Database responds to query
Response sent back to controller
alt
[User not found]
Modify view to inform user that no results were found
[User found]
Open the user page of the searched user

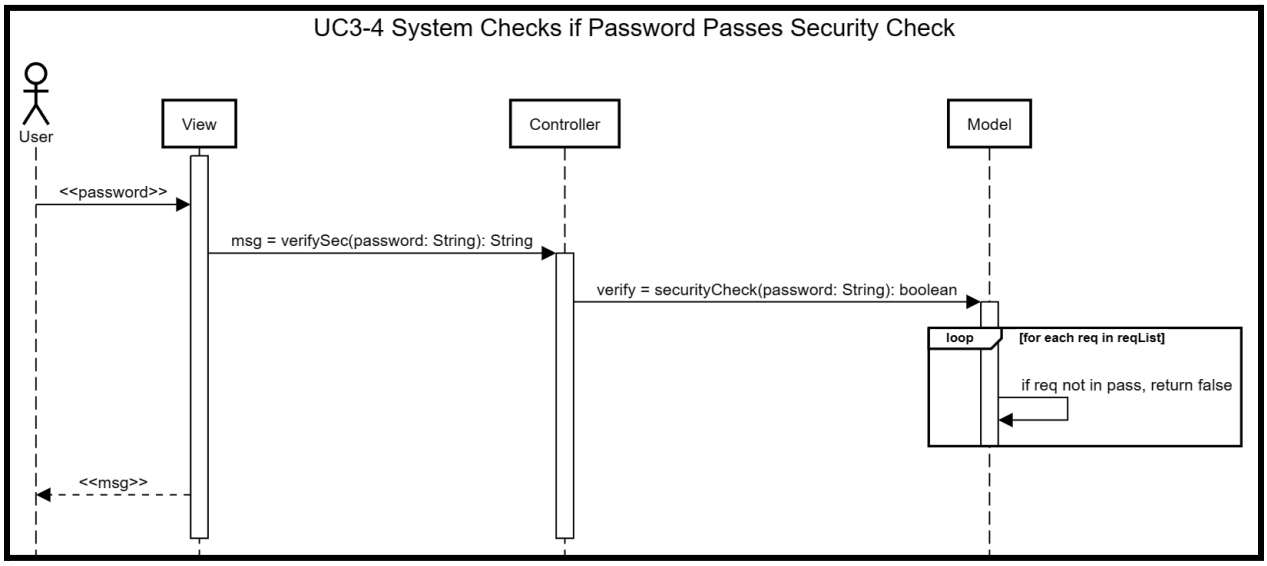

UC6-3 System checks user comment for malicious intent
User
View
Controller
Model
User enters a comment
View requests check on user comment
Controller sends query to database
for malicious words and phrases
Verifies Comment
Send result back to controller
alt
[Comment is Appropriate]
Update view to show comment
[Comment has malicious intent]
Send message to user of inappropriate comment

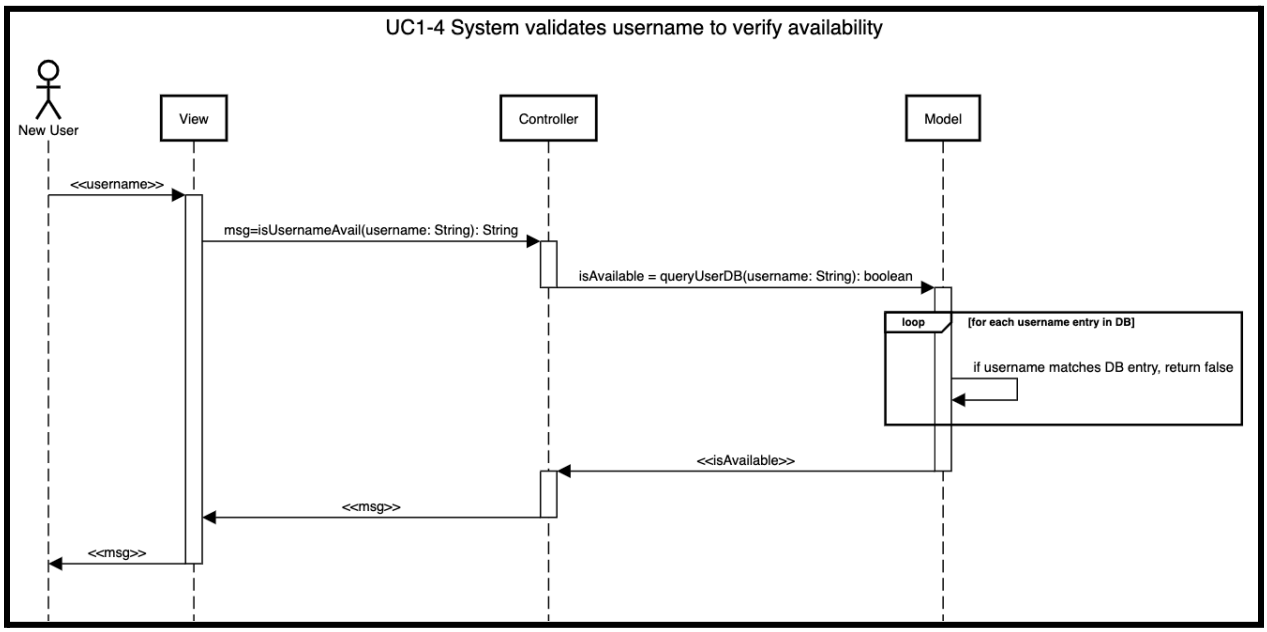

UC10-3 User Confirming Intent to Delete Recipe
User
View
Controller
Model
User clicks option to delete recipe
Sends confirmation request to controller
Returns option window
User views the options
alt
[Confirm Deletion]
User confirms deletion
View requests recipe deletion
Controller sends query to database for chosen recipe
Deletes recipe entry
Returns result
Confirms that entry was deleted
Returns confirmation notification
[Cancels Deletion]
User cancels
View closes deletion window

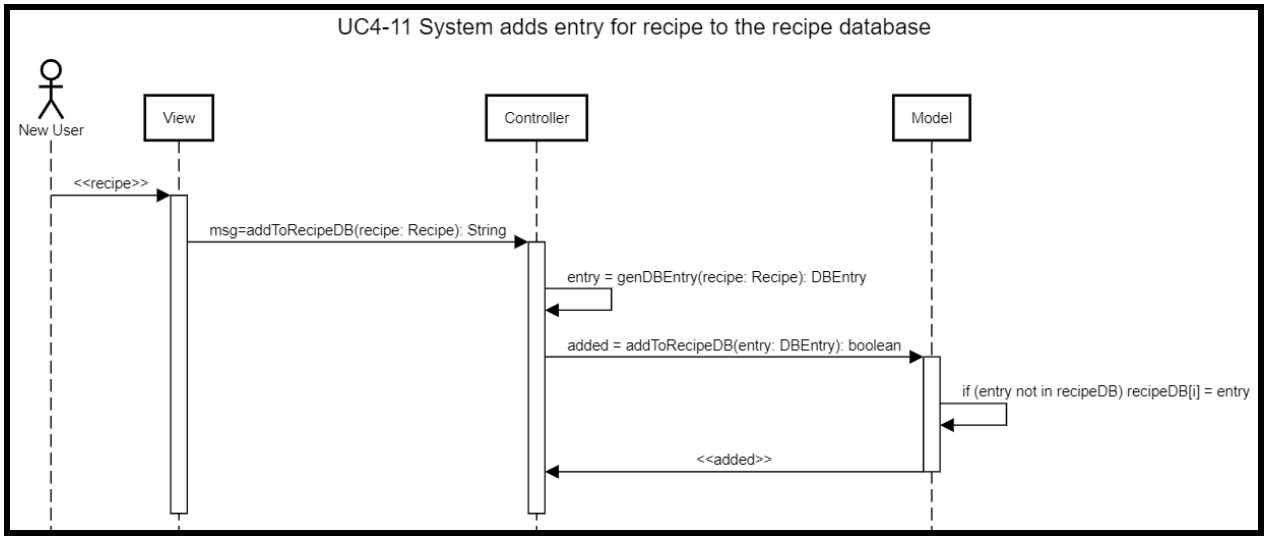

UC3-4 System Checks if Password Passes Security Check
User
View
Controller
Model
User inputs chosen password
View sends password to controller for verification of security
Controller sends query to database for proper security checks
Queries password requirements
Returns results
Password does not meet requirements, return invalid password notification
Display "invalid password" to user


UC7-2: System displays user comments
New User
View
Controller
Model
navigate to comments section of a recipe
request comment data
query recipeDB for attached comments
send attached comments to Controller
structures comments and sends data to view
display comments

Design Sequence Diagrams:

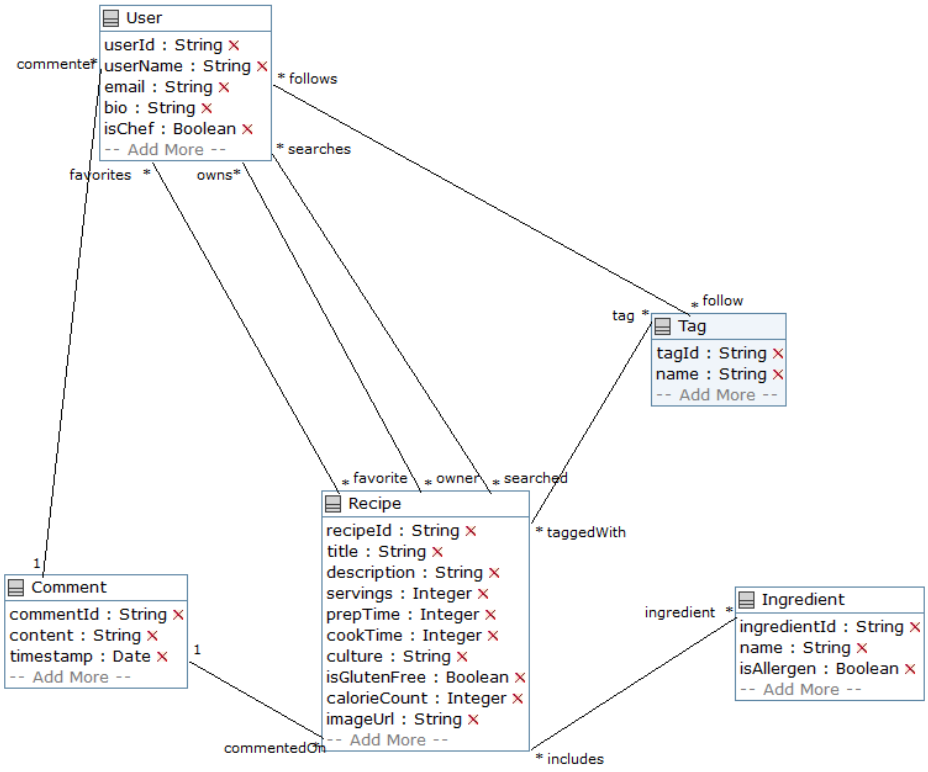




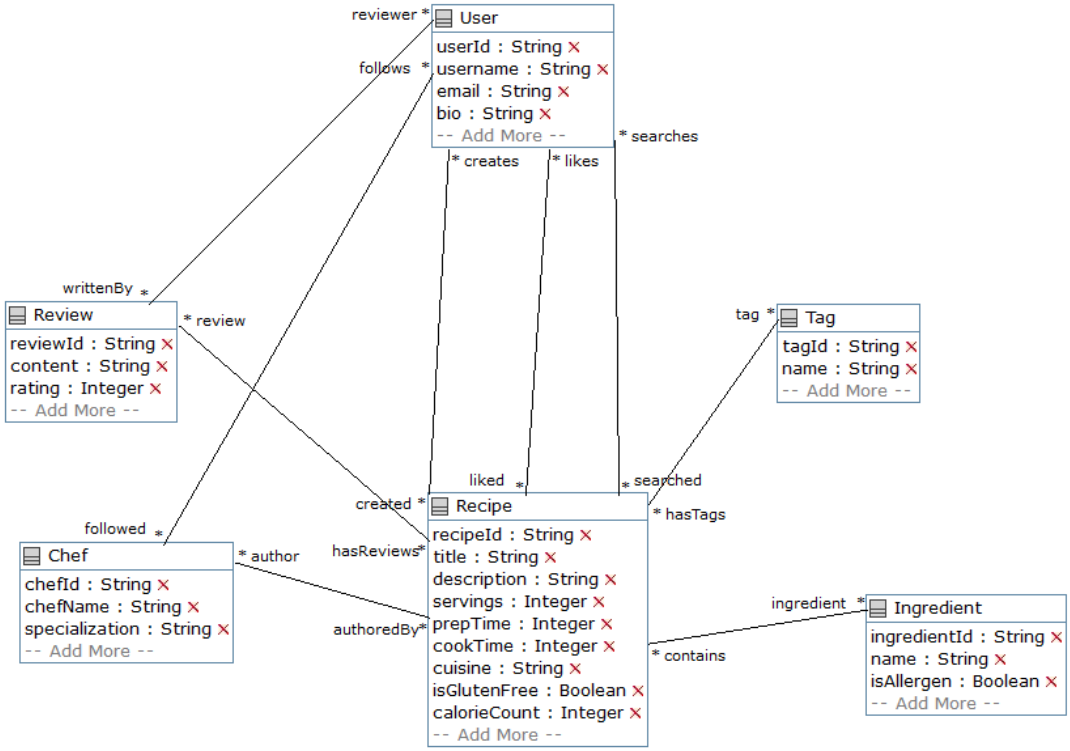

UC4-11 System adds entry for recipe to the recipe database
New User
View
Controller
Model
<<recipe>>
msg=addToRecipeDB(recipe: Recipe): String
entry = genDBEntry(recipe: Recipe): DBEntry
added = addToRecipeDB(entry: DBEntry): boolean
if (entry not in recipeDB) recipeDB[i] = entry
<<added>>

Domain Models:

Model #1



Model #2


reviewer *
User
userId : String
username : String
email : String
bio : String
-- Add More --
follows *
* searches
* creates
* likes
writtenBy *
Review
reviewId : String
content : String
rating : Integer
-- Add More --
* review
tag *
Tag
tagId : String
name : String
-- Add More --
liked *
* searched
created *
Recipe
recipeId : String
title : String
description : String
servings : Integer
prepTime : Integer
cookTime : Integer
cuisine : String
isGlutenFree : Boolean
calorieCount : Integer
-- Add More --
* hasTags
followed *
Chef
chefId : String
chefName : String
specialization : String
-- Add More --
* author
hasReviews*
authoredBy*
ingredient *
Ingredient
ingredientId : String
name : String
isAllergen : Boolean
-- Add More --
* contains

Descriptions:

Both models were created using UMPLE

Domain Model 1 Description:
For the first Grumio's list domain model, various classes intricately connect to facilitate user interaction. Recipes, Ingredients, Tags, Comments, and users help to ultimately form the system where each serves a purpose. Recipes pertain to details like the title, description, and preparation instructions, while ingredients represent the building blocks of each recipe. Tags provide a categorization method that enables users to discover recipes based on whatever criteria they want. The associations between these classes enable the user to do all sorts of things like creating, sharing, and favoriting recipes.

Domain Model 2 Description:
For the second Grumio's list domain model, once again we have several classes that help to assist user interaction with the application. Recipes, Ingredients, Tags, reviews, Chefs, and Users all form the applications architecture. Recipes once again encapsulate information regarding the title, description, and instructions pertaining to a meal. The ingredient class is also utilized here to serve as an element of each recipe. Tags also serve as a way to categorize recipes for east discovery online. Reviews help to foster community engagement by allowing users to provide feedback on recipes. Finally the "Chef" class is used to represent individuals who contribute their culinary knowledge by authoring recipes. This model is similar to the first but has some added functionality with classes like review and chef.