

## **Team logistics and evaluation**

### **Team name**

- 2Cool 2Cook

### **Team member names and roles**

- Ruisi Jian (Expert coder/Full-stack)
- Yueting Liao (Team leader)
- Yuchen Wang (Front-end/UI design)
- Yifan Wang (Back-end/Database design)
- Xiaoyu Zhang (Back-end/Search algorithm)

### **A summary of how you will run each meeting and where minutes will be recorded and stored (No changes from Team Deliverable 2)**

- Summarize and evaluate what we did in the last meeting and conclude our inadequacy.
- Each member will be reporting their progress of tasks and potential problems from the last meeting
- Discuss the progress made so far and evaluate for potential improvements.
- Provide some constructive thinking and set a list of tasks for the next meeting.
- Assign appropriate works for each team member.
- Use Google Doc to record the minutes from the meeting (One Doc for all meetings)
- [https://docs.google.com/document/d/1mJW8UWMUnrlh8FBEff\\_GY9MLaYU6fGM3n-BjViskeRc/edit?usp=sharing](https://docs.google.com/document/d/1mJW8UWMUnrlh8FBEff_GY9MLaYU6fGM3n-BjViskeRc/edit?usp=sharing) (updates every meeting)

**Of the success criteria you identified in Team Deliverable 2, describe the one criterion that you view as being the most difficult to achieve (and why)**

Success criteria 2 is the most difficult to achieve. The team members all have other obligations in competing for the time that we can assign for this project. The time that we needed to complete some tasks were difficult to estimate. The majority of the team members are not experienced enough with front-end developments and using tools/languages such as HTML and javascript. Also, it is not very efficient to debug in node-red so it takes longer for each member to complete their tasks.

1. Attending each meeting on time.
  - Ensure the progress moving forward within the team plan.
2. Finishing the assigned task before the assigned deadline.
  - Ensure that the work schedule is followed.
3. Asking current problems and sharing feedback to team members.
  - Ensure the project is building in maximum efficiency and all team members understand the issue.
4. Planning out the unified process/brief schedule for the project.
  - Ensure team members could keep track of each milestone of the project.
5. Helping teammates on their tasks if they feel the risk of passing the deadline.
  - Ensure that the team is cohesive and each teammate takes responsibility not only for their separate part but for the whole project.

## Project status reports

**Write a project status report that summarizes the current state of your project. The project status report describes an up-to-date list of issues encountered. And for each issue, include a brief summary, the person responsible for resolving the issue, and how the problem is being solved or was solved**

The current state of the project:

- The general structure of the website page is completed

The team would like to set up the basic database to frontend connection for each section of the basic structure and also get their feet wet with node-red. The team decided to split the general structure of the website into digestible parts and assign it to each team member. An issue was how to deal with the synchronous update of code on node-red, we encountered this by transferring files through the Github repository. Xiaoyu was in charge of the main page search bar. Yuchen was responsible for the Recipe display. Yifan took charge of the user profile. Yueting was responsible for the farmers-market page. Ruisi was in charge of the Menu display.
- Databases are successfully and correctly created

To move forward with the organization of recipe data, the team needed to design a database schema that would fit well with the need for the search engine and front-end display. All the members are responsible for designing the schema and transferring data from the JSON file to the database. Then each team member was responsible for generating the CSV file for one table in the database using python programs. The CSV files were shared in the meeting and all team members set up their local instance of the database based on the CSV files.
- Page layout and display tables on the menu page

The split of the instructions didn't align with the numbering of the instructions due to an incorrect split of the instructions in a list format on node-red. To solve this problem, Ruisi replaced all the instances of the comma in the content with a rare symbol and then split by a comma when creating a list in node-red. Finally, replace the rare symbols with the commas to retrieve the original message.
- Page layout and display tables on the farmer markets page

The issue of solving how to display database table information on the page is done by using SQL sentences in javascript. There were some undefined issues with deploying some rows from the table, the bug was fixed after changing the parsing in javascript. Page layout and search buttons were set up and waiting for the implementation of search functions. Yueting is responsible for design/developing this page.

**Your project status report also must include a summary of how each team member contributed to the team deliverable 3. This helps ensure that each team member is taking an active role in each aspect of the project.**

Xiaoyu Zhang: Responsible for recipe searching and showing results by displaying table according to the users' inputs

Yuchen Wang: Search recipe data by id. Responsible for recipe data display with a specific recipe.

Yifan Wang: Created the User Profile page of the project. Managed both page design and database connection.

Ruisi Jian: Generated the CSV file for the instruction table. Responsible for the design and implementation of the Menu page.

Yueting Liao: Created farmers market page of the project, implemented general layout, and displayed tables from the database.

All members designed and implemented tables in the database by using written python programs to generate them.

## **Project deliverables**

**A brief (one paragraph?) high-level description of your system's design, which must be object-oriented; write this in such a way that your stakeholders should generally be able to understand it.**

The project is developing a website that allows users to search for recipes, add recipe datas to personal accounts, and search for farmers markets. The system's design contains 3 main objects, User, Recipe, and Farmers Market. Users have their own daily menu and set of recipes in their Favorite. Users can search among the Recipes given keyword and view the Recipes. Users can add Recipes to the daily menu and Favorites. Recipes have ratings and comments that User can write for the Recipes. Farmers Market contains markets. Users can search markets among Farmers Market using keywords.

**Use cases from Team Deliverable 2 with any revisions, additions, etc.**

Deleted 3 Use Case

**Use Case Title: Search for a Recipe**

Primary Actor: User

Stakeholders and Interests: User would like to find a recipe they want to cook

Preconditions: N/A

Main Success Scenario:

1. User navigates to the search bar
2. User types in the keywords of a recipe
3. User get a list of potential recipes through keywords;  
if displaying nothing, return to step 2
4. User chooses one of the recipes from the displayed list
5. User gets a detailed view of the chosen recipe

Extensions:

- 2a. If the user leaves the search bar fields blank, then show blank.
- 2b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: Users able to get all recipes

Use case references by which you reference another use case by underlining the given task:

N/A

## **Use Case Title: User Creates a Login Account**

Primary Actor: User

Stakeholders and Interests: User would like to upload their favorite recipes

Preconditions: N/A

Main Success Scenario:

1. User navigates to “create account” button
2. Fill in the personal account information
  - a. Username (required)
  - b. Date of Birth
  - c. Password (required)
  - d. Confirmed password (required)
  - e. Email address (required)
  - f. Allergies
  - g. Dietary preferences
3. User automatically logs into the new account

Extensions:

- 2a. If User leaves a required blank or empty, then shows errors
- 2b. If the username has already existed, then shows errors
- 2c. If the confirmed password does not match the password, then shows errors

Minimal Guarantees: N/A

Success Guarantees: Users own a new account

Use case references by which you reference another use case by underlining the given task:  
N/A

## **Use Case Title: User Creates a Daily Menu**

Primary Actor: User

Stakeholders and Interests: User would like to plan out what they cook for meals on a daily basis

Preconditions: User must be authenticated and logged in

Main Success Scenario:

4. User navigates to today's menu
5. User select which meal (breakfast, lunch, dinner) to add
6. User search a recipe for that meal
7. User add recipes to that meal on the daily menu
8. User clicks save button to finish create a menu

Extensions:

3a. If the user couldn't get a recipe from add, user can choose to upload a new recipe

Minimal Guarantees: N/A

Success Guarantees: A new daily menu is created

Use case references by which you reference another use case by underlining the given task

The users search for a recipe

The users upload a new recipe (optional)



## **Use Case Title: Apply Filters by Allergy**

Primary Actor: User

Stakeholders and Interests: User would like eliminate recipes based their dietary allergy

Preconditions: Log into the account; Enter recipe information in the search bar

Main Success Scenario:

9. User login to the account
10. User updates the their dietary allergy in their profile
11. User navigates to the search bar
12. User enters the keywords for a recipe
13. Allergy filter will be applied to the recipes returned
14. Recipes with the keywords will be displayed if it does not contain allergy ingredient

Extensions:

- 2a. If users already have their dietary allergy entered in their profile, skip this step.
- 4a. If the user leaves the search bar fields blank, then show blank.
- 4b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: Normal search result will be displayed if the filter failed in an unexpected manner

Success Guarantees: A list of recipes with no allergic ingredients will be displayed after the filter.

Use case references by which you reference another use case by underlining the given task

The users create an account and log in

The users search for a recipe

**Use Case Title: User feedback**

Primary Actor: User

Stakeholders and Interests: User would like to give some feedback on their experience with this website

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the feedback page
3. User check from a list of frequent issues (required)
4. User types in more detailed comments in textbox

Extensions:

3a. If the user did not select any of the check from the list, the submit button is disabled

Minimal Guarantees: N/A

Success Guarantees: A success submission page will be displayed to the user

Use case references by which you reference another use case by underlining the given task  
The users create an account and log in

## **Use Case Title: Favourite Recipes**

Primary Actor: User

Stakeholders and Interests: User would like to save recipes that they like for future reference

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the search bar
3. User enters the keywords for a recipe
4. Recipes with the keywords will be displayed
5. User would select the “add to favourite” button

Extensions:

- 3a. If the user leaves the search bar fields blank, then show blank.
- 3b. If the user types in unidentified keywords, then show blank.
- 5a. If the recipe is already added to favourite, click again to remove from favourite

Minimal Guarantees: N/A

Success Guarantees: Recipe saved to their list of favourite recipes

Use case references by which you reference another use case by underlining the given task

The users create an account and log in

The users search for a recipe

## **User Case: Rate the Recipe**

Primary Actor: User

Stakeholders and Interests: User would like to rate each recipe

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the search bar
3. User enters the keywords for a recipe
4. User navigates to the recipe
5. User click rate the recipe button and give a score

Extensions:

- 3a. If the user leaves the search bar fields blank, then show blank.
- 3b. If the user types in unidentified keywords, then show blank.
- 5a. If user has already rated, rating button disabled

Minimal Guarantees: N/A

Success Guarantees: Recipe score is updated with the rating input

Use case references by which you reference another use case by underlining the given task

The users create an account and log in

The users search for a recipe

## **User Case: Sort the Search Result by Rating**

Primary Actor: User

Stakeholders and Interests: User would like to see recipes with higher rating up front

Preconditions:

Main Success Scenario:

1. User navigates to the search bar
2. User enters the keywords for a recipe
3. Recipes with the keywords will be displayed in order of sorting

Extensions:

- 2a. If the user leaves the search bar fields blank, then show blank.
- 2b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: Recipes will automatically be ordered based on the rating from highest to lowest

Use case references by which you reference another use case by underlining the given task

The users search for a recipe

The users rate the recipe

## **User Case: Ranking List for Top 10 Recipes for Each Week**

Primary Actor: User

Stakeholders and Interests: User would like to see the best rating recipes

Preconditions: N/A

Main Success Scenario:

1. User navigates to the ranking page
2. Ranking is updated based on current recipe rating at the beginning of each week

Extensions: N/A

Minimal Guarantees: N/A

Success Guarantees: Display top 10 recipes weekly.

Use case references by which you reference another use case by underlining the given task

The users rate the recipe

## **User Case: Search for Nearby Farmers Market for to buy Ingredients**

Primary Actor: User

Stakeholders and Interests: User would like to see nearby farmer market to buy ingredients

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to search farmers market page
3. User enters information in the text box
4. The farmers market result will be displayed

Extensions:

- 3a. If the user leaves all text box fields blank, then show all.
- 3b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: All the farmers market will be displayed depend on user input

Use case references by which you reference another use case by underlining the given task  
The users create an account and log in

**Use Case Title: Set timer for certain steps that require timing**

Primary Actor: User

Stakeholders and Interests: User would like to time for certain steps according to the recipe instruction to improve the quality/taste of the recipe

Preconditions: User must be authenticated and logged in, user must add the recipe into his/her menu

Main Success Scenario:

1. User navigates to “record time” button
2. Fill in the value in the timer bar
3. User clicks the start time button
4. A web label will be displayed to notify the user once the time is up

Extensions:

2a. If User leaves a required blank or empty, then start button is disabled

Minimal Guarantees: A timer is set up

Success Guarantees: User correctly times the step according to the instruction

Use case references by which you reference another use case by underlining the given task:

User creates a daily menu

User creates a login account



## **A summary (a few paragraphs?) of how your analysis classes have changed since you submitted Team Deliverable 2**

Since our team decided to improve object relationships and reduce complexity of the website. We deleted the use case of recipe upload, and deactivated the user account, the system no longer needed the admin staff to verify the uploaded recipes or to execute the deactivation of the user account. Therefore, the team removed the admin class from the analysis diagram as well.

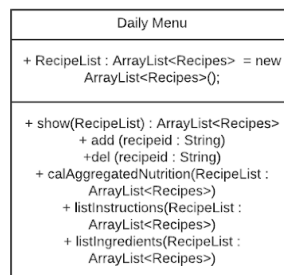
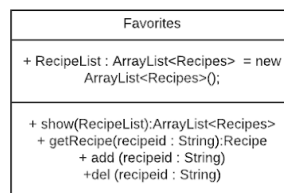
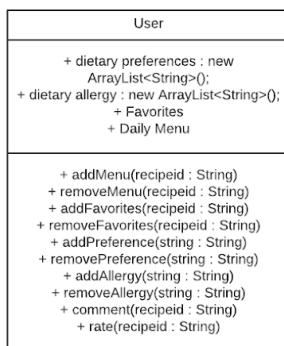
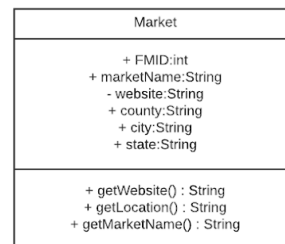
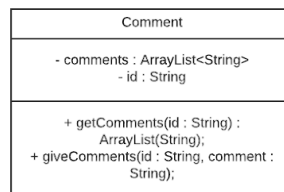
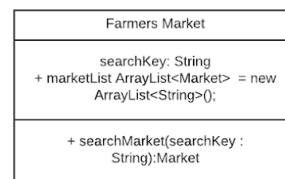
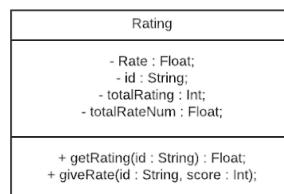
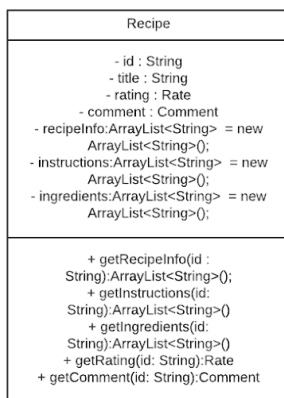
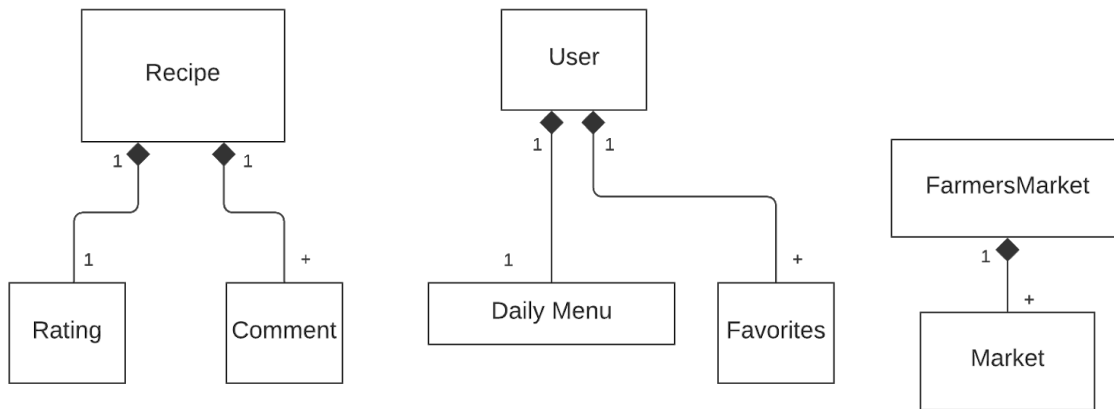
More specifically for the User class, we decide to remove the upload and deactivate function. This change reduced flexibility of users, but the system becomes more light weighted and users can focus more on the main utilities for the site. Additionally, staff members require less effort to manage/ update databases and user accounts.

In terms of class objects, for the Daily Menu class, we decided to remove the showPast7D function. Similar to removing the upload function, this change improves the focus on the site. ShowPast7D displays all visited recipes in the last week, but this function does not have high priority compared to others.

For the Recipe class, we decided to add the getNutrition function. Since we had added the nutrition information of the recipes to the database, we decided to add this information to the display of recipes. This change would give more information for the users to know more about the recipe.

For the Market class, we decided to add getMarketName and change the getStreet to getLocation. This change better gives the users a more detailed description of where the market locates, not just street information.

**Design class diagrams; using proper UML notation, present your design classes, associations, relationships, attributes, operations, etc.**

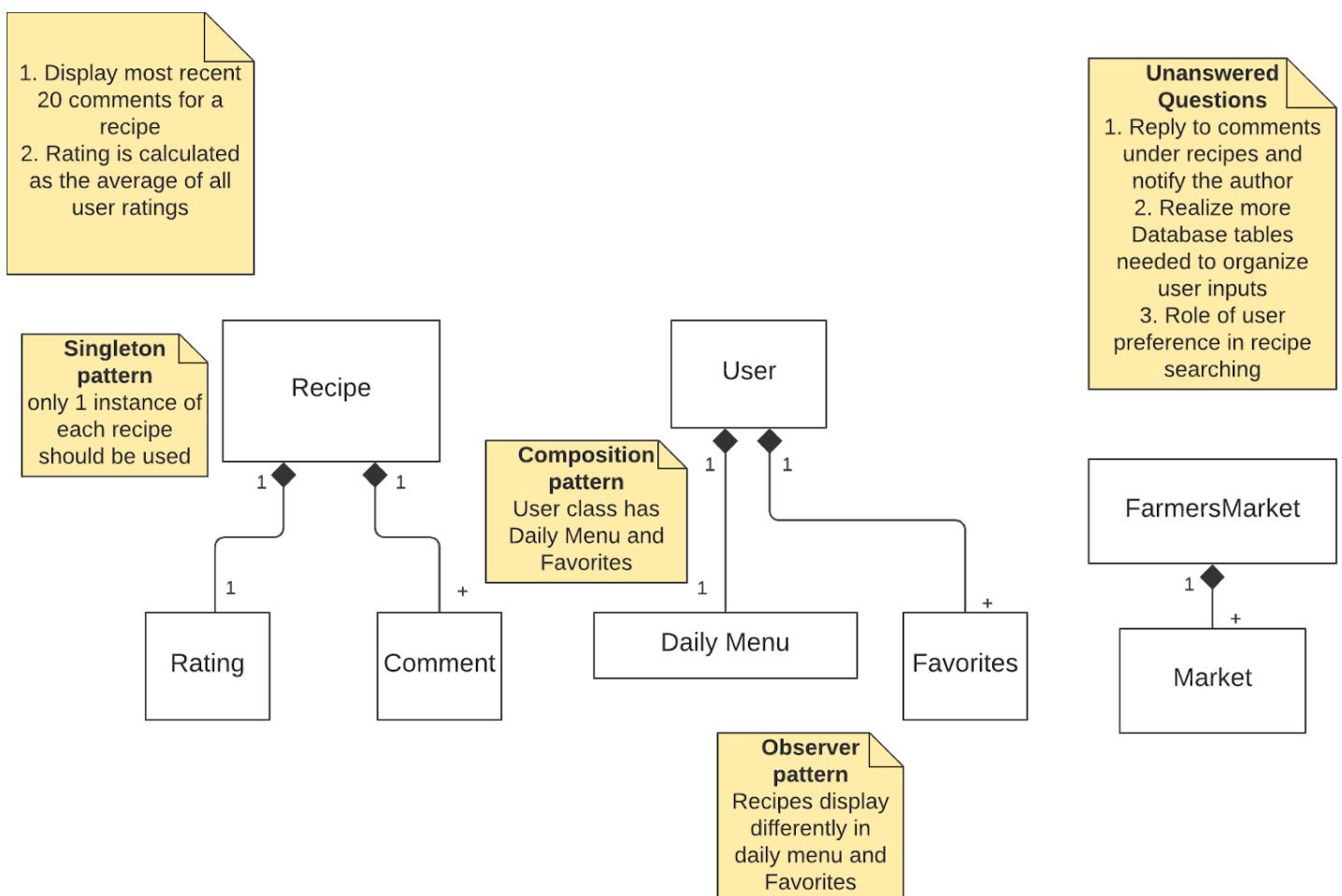


Within your class diagrams, add a UML note to briefly describe how your design classes map back to analysis classes, i.e., add a UML note to the corner of each diagram as described in lecture

Within your class diagrams, add at least three UML notes that indicate where you make use of specific design patterns

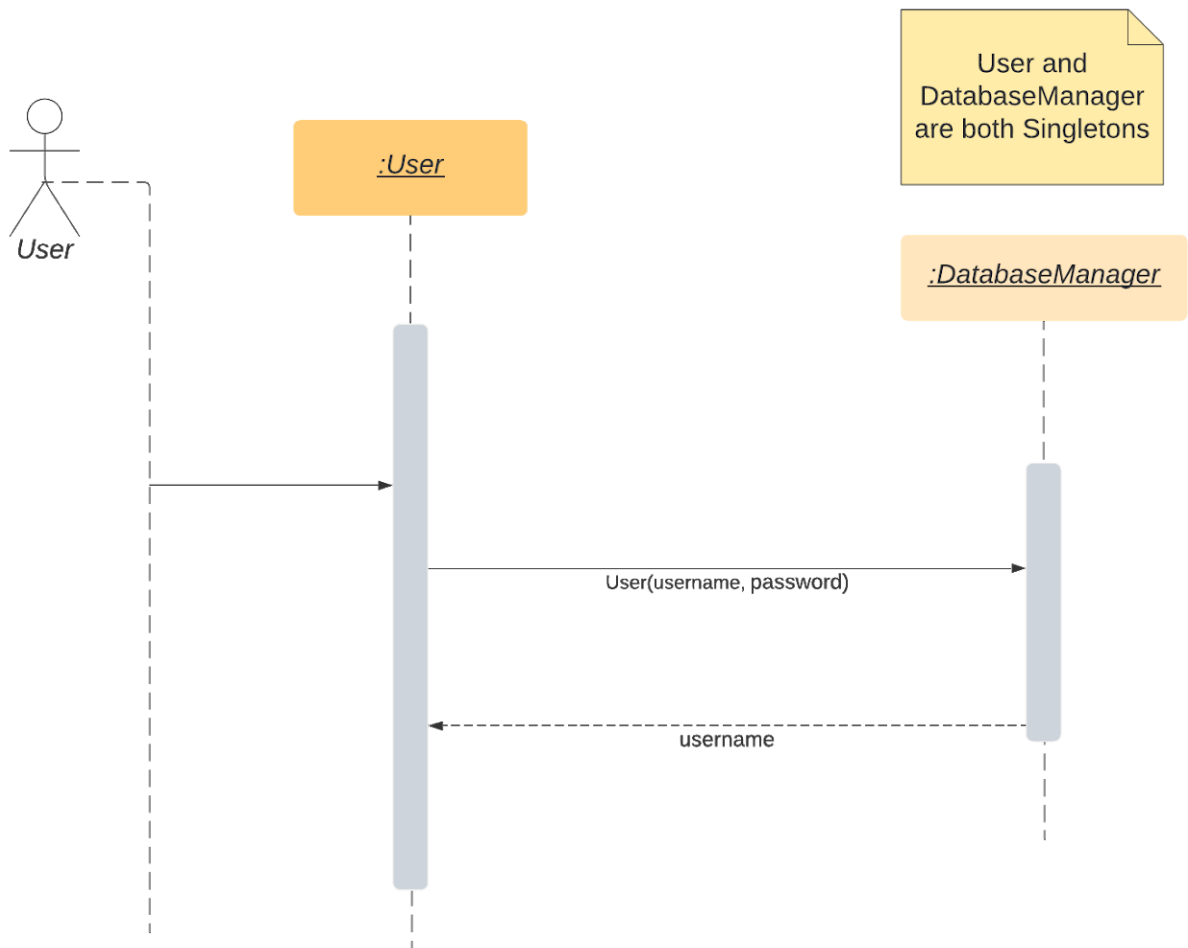
Also within your class diagrams, include at least three UML notes with unanswered questions (e.g., a design decision yet to be made, a gap in the requirements that the design reveals, etc.)

**Note: Composition refer to “Composite Design Pattern”**

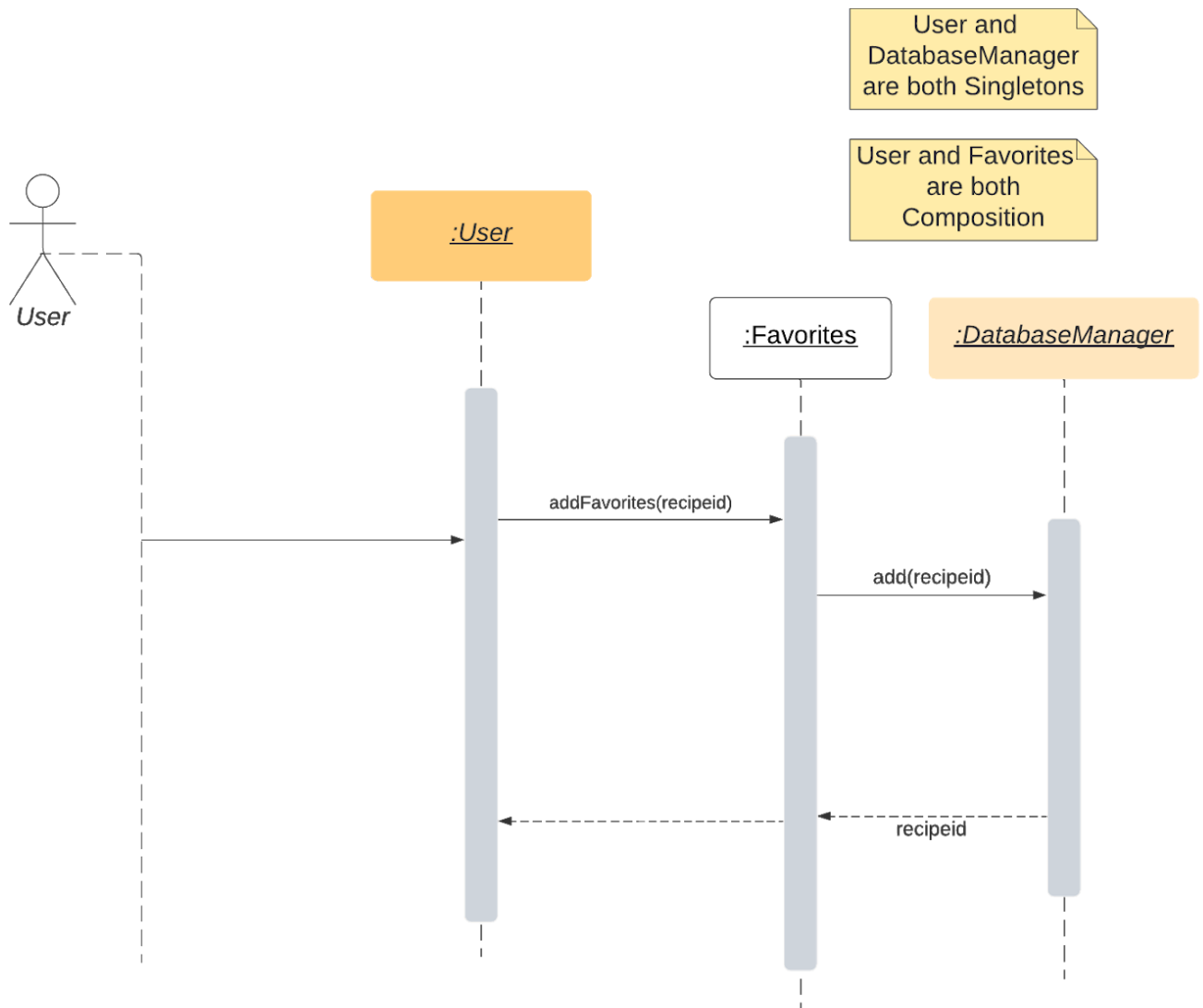


Sequence diagrams; a sequence diagram shows how a specific use case is realized within the design, i.e., how objects interact with one another to implement specific behavior; using proper UML notation, provide at least six sequence diagrams originating with your use cases

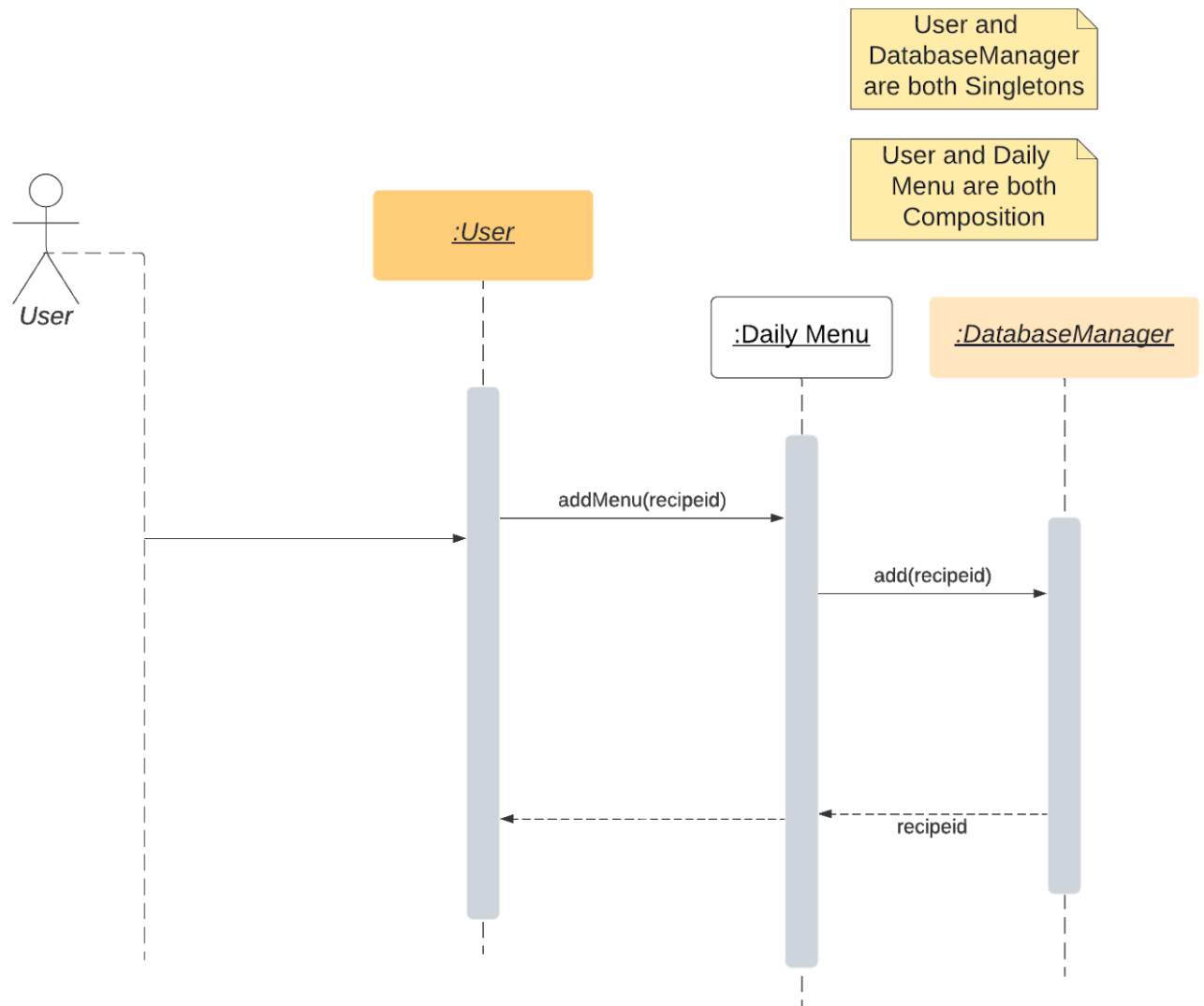
Use case: Create user



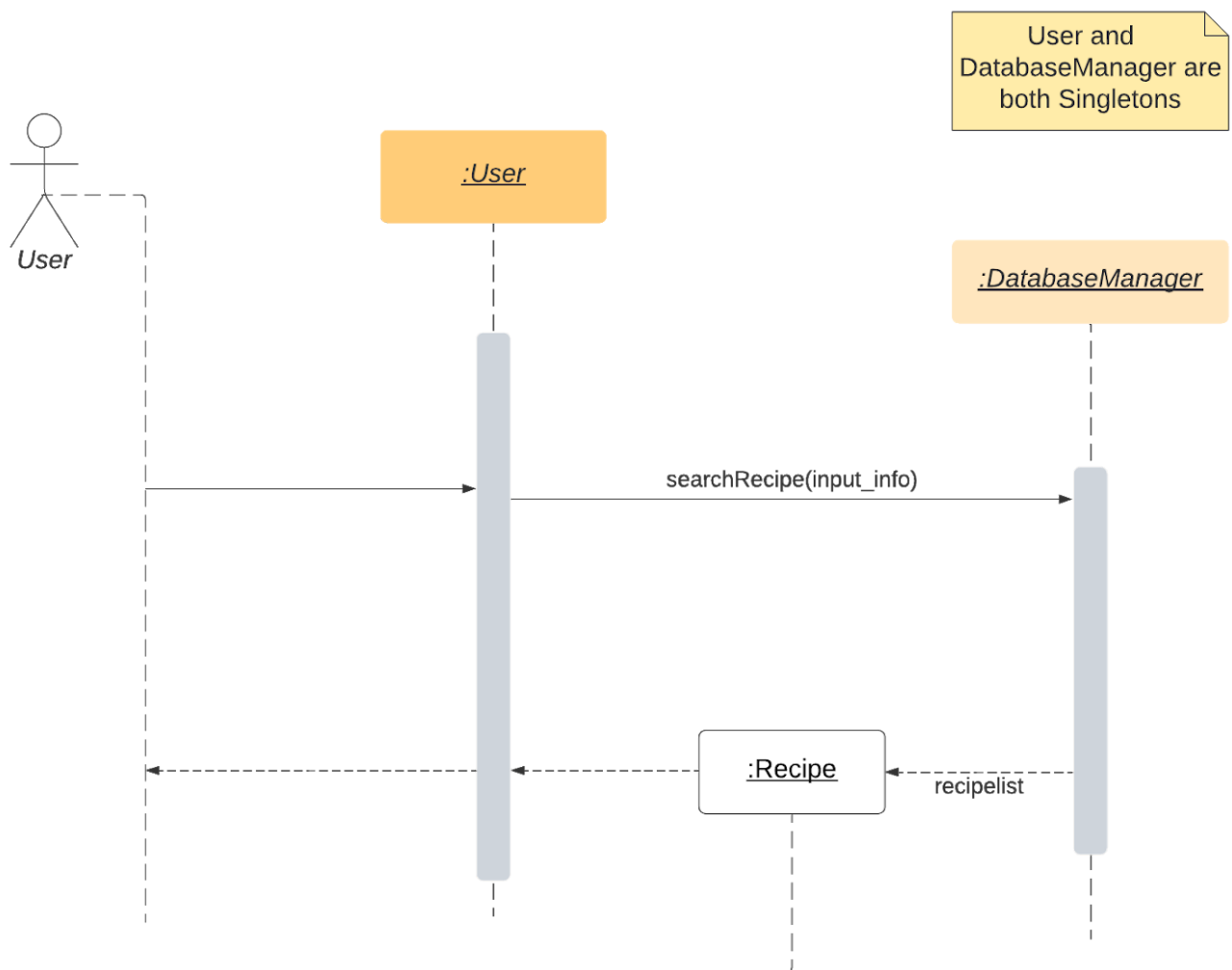
## Use case: Add favorite



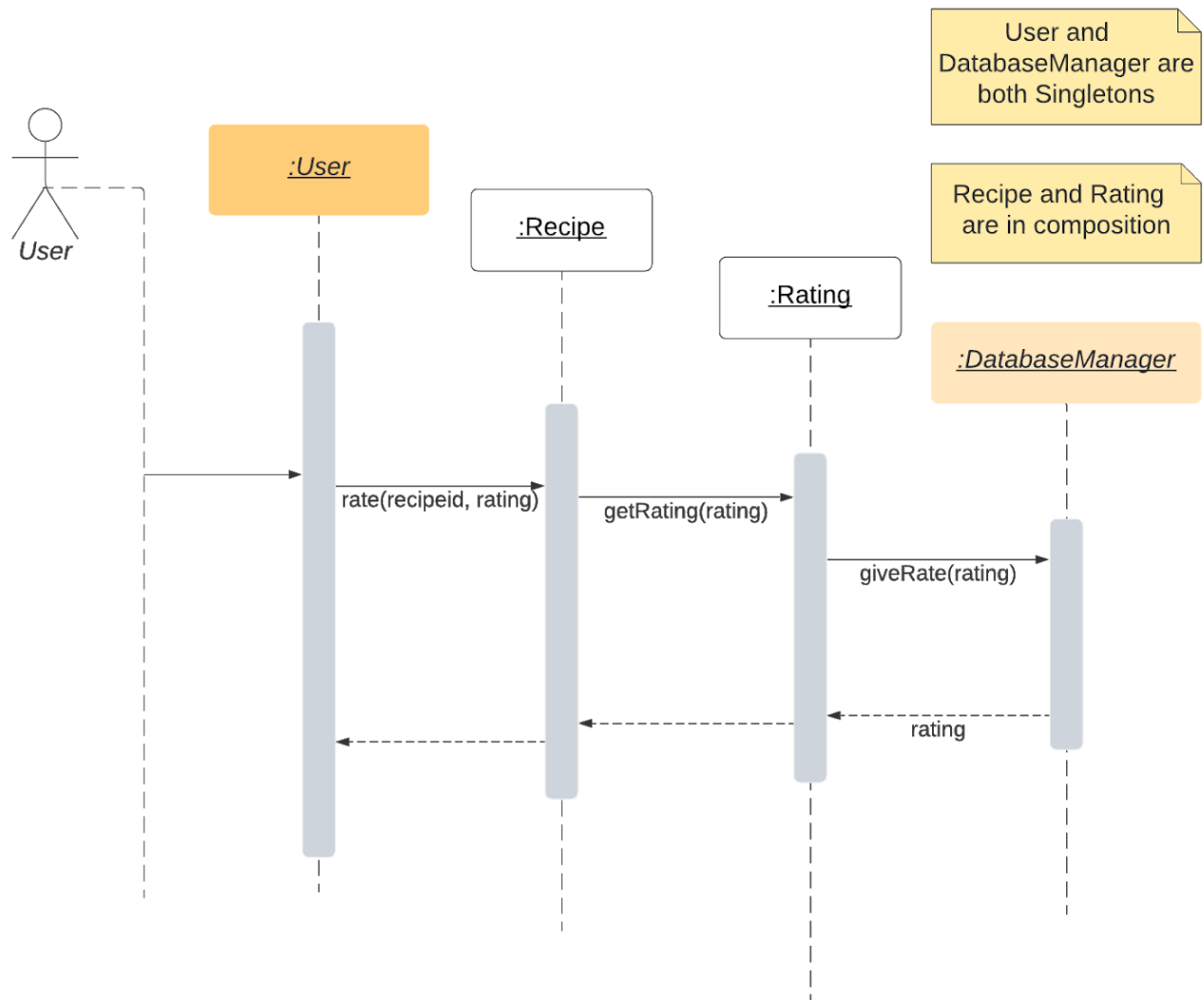
### Use case: Add Menu



### Use case: Search Recipe



### Use case: Rating





### Use case: Comment

