COOKBAIT FINAL REPORT
Group-13

Details of Group Members

Name	Enrollment Number	Phone Number	Contributions
Manashree Kalode	21114057	+918080930624	Overall UI Design, developed Individual-Recipe screen, wrote README file, making the ppt
Mehak Sharma	21114060	+918121020227	Python Web Scripting and Scraping, writing Report, developed UI of Display-Recipe screen, designed the sequence Diagram
Neha Gujar	21114039	+918329717688	Developed Display-Recipe Screen, writing Report, designing of database
Nishita Singh	21114068	+918826468735	Developed Launch Screen, designed the Use-case Diagram and Class Diagram, Python Web Scripting and Scraping, writing Report
Raiwat Bapat	21114078	+917666191528	Developed Comments-Page, Integrated all screens, making the ppt
Sanidhya Bhatia	21114090	+918817471350	Developed Home-Screen, designed the Use-case Diagram and Class Diagram

Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. It is a type of Behavior diagram which draws distinction between actors and the system.

In our application, i.e. Cookbait, we have only a single user(actor) which is the app user. The actor first enters into the application and is greeted by the welcome screen. The user then enters the set of ingredients available with him\her into the search bar. Following this he/she gets redirected to a screen which displays various recipes within a range of 60-100% match, having separate categories for each of >=90%, >=80%, >=70% and >=60% match. In case such a set of recipes does not exist the user is presented with basic recipes that can be prepared with the most common set of ingredients available in an Indian household. On clicking on the recipe, the user is redirected to the recipe page which displays cuisine, recipe name, ingredients,an image of the prepared dish, total time taken to prepare the dish and comments of previous users. There's an option to add comments to the chosen recipe by entering details like name, email address and comment. A user also has the option to add multiple comments to the same recipe.

A brief description of each module of the use case diagram is as follows:

1. Choose available Ingredients:

    - After clicking on the Search bar User has to choose certain ingredients from the checklist. This is a compulsory action. Hence, "Choose from Checklist" sub-module i**ncludes** "Choose Ingredient" module.
    - The user may search for an ingredient of their choice in the checklist from the search bar. Since this action is optional, The "Search an ingredient" sub-module **extends** "Choose Ingredient" module.
    - After performing the search, the user is redirected to the page with the resultant recipes

2. Choose from Shortlisted Recipes:

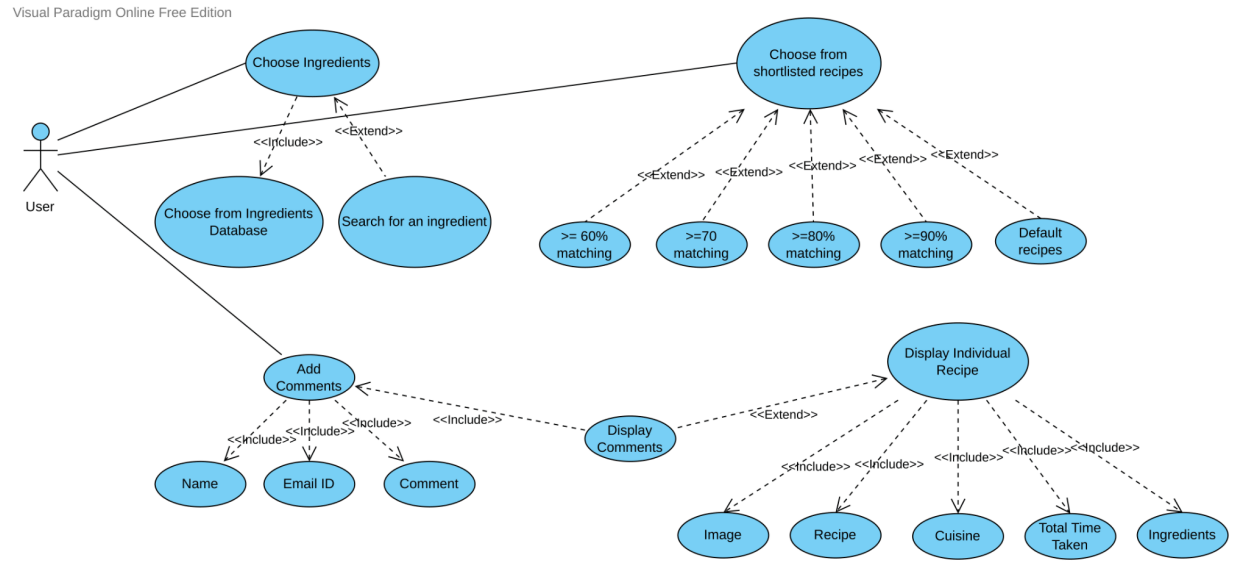
    - Depending upon how many ingredients from the entered list matched with the ingredient list of each recipe, and vice-versa, four categories of recipes are returned.
    - These are recipes with >=90%, >=80%, >=70% and >=60% match percentages respectively.
    - If there are no recipes lying within a certain percentage range, then the particular category is not shown. If none of the four categories have recipes, then a set of

default recipes are returned. Hence, all of the sub-modules **extend** “Choose from Shortlisted Recipe”

3. Display Individual Recipe:
    - Every individual recipe page has the Cuisine name, Ingredients listed, Time taken to cook the dish, Detailed recipe, and image of the Dish.
    - All these are mandatorily displayed for each recipe in the database, hence the usage of includes.
    - A recipe may have comments from previous users.So the comment submodule extends “Display individual recipe” module.

4. Add Comments:
    - A user may choose to comment on a particular recipe(multiple comments possible).
    - For this, they have to necessarily provide their Name, email ID and Comment.

Use Case diagram:



Class Diagram

A class diagram describes the static structure of a system. It shows how a system is structured rather than how it behaves. It is also known as a structural diagram.
Classes are represented as solid outline rectangles with compartments and have optional attributes and operations compartments. Attributes are named properties of a class. Operations are the functions executed within the classes.
In the Class diagram of Cookbait, the classes are:

1. Recipes:
    a. Attributes:
        i. Name
        ii. RecipeID
        iii. Cuisine
        iv. Ingredients
        v. Procedure
        vi. Total time taken
        vii. Image

2. Ingredients:
    a. Attributes:
        i. IngredientID
        ii. Name
3. Comments:
    a. Attributes:
        i. Name
        ii. Email-id
        iii. Comment
    b. Operations:
        i. Add a new Comment
4. Choose ingredients:
    a. Attributes
        i. Selected ingredients list
5. Display recipes:
    a. Operations
        i. Get recipes with >= 90% ingredient match
        ii. Get recipes with >= 80% ingredient match
        iii. Get recipes with >= 70% ingredient match
        iv. Get recipes with >= 60% ingredient match
        v. Get default recipes
6. Individual recipe:

a. Attributes
    i. All comments

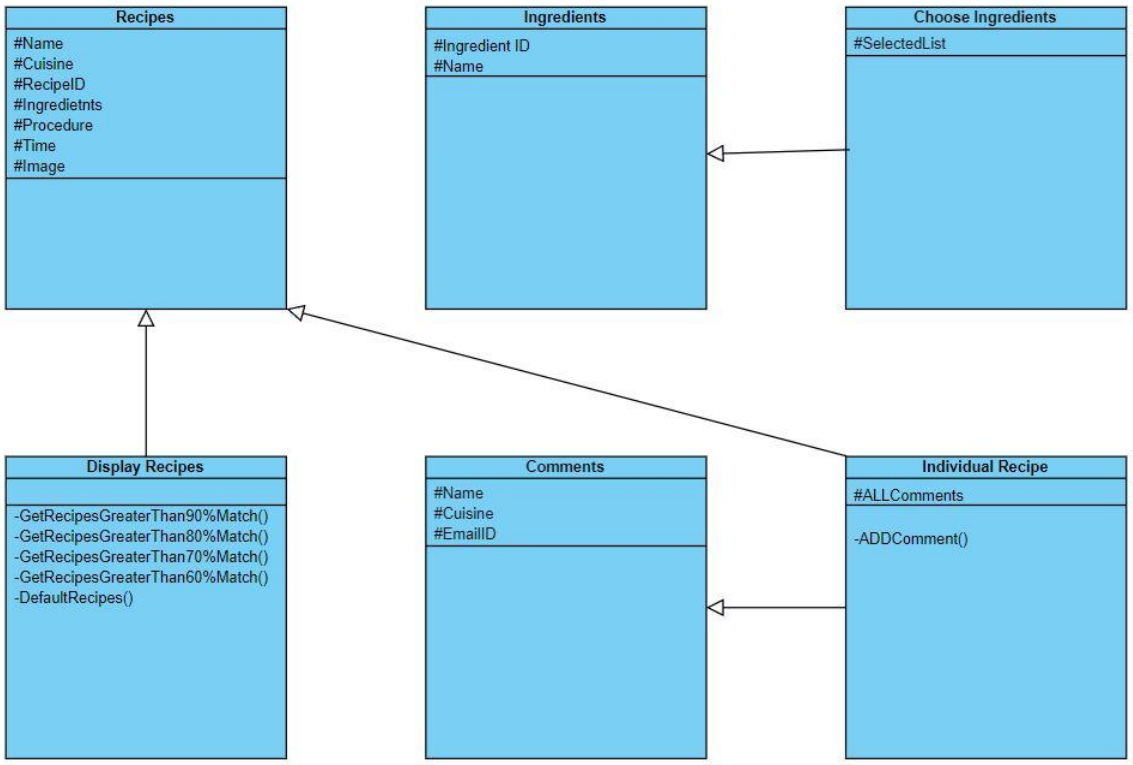
Generalization:

The generalization relationship occurs between two entities or objects, such that one entity is the parent, and the other one is the child. The child inherits the functionality of its parent and can access as well as update it.

Generalization relationship used in class diagram:

Here, Display Recipes, Individual recipe and comments class inherits recipes class and is thus able to access the attributes of each recipe such as name, ingredients etc.
Choose ingredients class inherits the ingredients class and is able to access the names of the ingredients.
Comments class also inherits the individual recipe class since it requires data for 'all comments' attribute.

Class Diagram:



Sequence Diagram

A sequence diagram emphasizes on the sequence in which the actions occur along with the requests made, responses received during the course of action.

The sequence diagram of Cookbait has one single actor ie the user.In accordance with the app flow, the components of the sequence diagram are as follows:

1. Launch Screen: User opens the app and is greeted by the launch screen

2. Home Screen and Recipe Database : User chooses the available ingredients from the checklist. The query is made to the database to fetch the recipes with the matching set of ingredients.Return response contains the list of all matching recipes which has sub-sections on the basis of percentage matched and in-case no matches are greater than 60% it returns a basic recipes list.

3. Full Page Recipe: When the user clicks on a specific recipe, the specifics of that recipe are fetched from the database i.e. both the recipes and comments by other users.

4. Comment Database: If the user chooses to comment on the recipe, the comment is added to the comments database.