# CSE-5324-001 - SW Engineering: Analysis, Design, & Testing

# RICETTA

## Team Number: Team 11 Team Name: Ricetta

**Team Members:** Tanmay Kakade – Team Captain(1002023412)

Mrunmai Magar(1002092125) Neha More(1002060504) Utkarsh Sonawane(1002029320) Aditya Vichare (1001968015)

**Date: 04/07/2023 Iteration #02**

# Table of Contents

|  |  |
| --- | --- |
| **1.** | **Project Description… 3** |
| **3.** | **Requirements… 4** |
| **4.** | **Use Case List… 5** |
| **5.** | **High Level Use Cases… 6** |
| **6.** | **Use Case Diagram… 8** |
| **6.** | **Requirements to Use Case Traceability Matrix… 9** |
| **7.** | **Increment Matrix… 10** |
| **8.** | **Domain Model… 11** |
| **9.** | **Task List… 12** |

**Project Description**

## A meal recipe application called Ricetta allows users to prepare their preferred meals whenever and wherever they choose. When you don't know how to cook and you live away from home, this application is incredibly helpful. For instance, during Covid-19, many individuals were trapped in their houses because they lacked cooking skills and were unable to place any kind of order. In such circumstances, our app will be of assistance. In order to keep you healthy, it will recommend meals that fit your preferences and inform you of the nutritional worth of the food you are consuming.

**Functionalities**

Function 1: Display Categories

## A user can choose an option from the selection pane for this function to separate recipes into categories. The choices are divided into groups based on the different types of meals a user can choose in accordance with his tastes. Breakfast, salads, soups, etc., as an example.

## Function 2: Suggest Similar recipes

## The application will suggest recipes of similar kind based on the choice user has chosen. This feature can be useful to get a variety of similar choice.

## Function 3: Nutrition chart

## A lot of people prefer their diet to be balanced, healthy diet. One can maintain track of their consumption by using this tool, which allows users to learn the nutritional values of a chosen recipe.

Function 4: List of Ingredients

## To cook a recipe, a user must first know what ingredients he/her has on hand, so our app provides a list of ingredients based on the recipe selected.

Function 5: Time required

## Once a user selects a recipe to follow, this functionality allows him to check for how much preparation time and cooking time is required.

Function 6: Number of Servings

## According to the quantities of ingredients added using the recipe, this feature tells user about how many people can have the dish.

Function 7: Stepwise cooking process

## This gives a complete description and instructions about how the dish should be prepared, cooked, and served.

## Function 8: Feedback and Ratings

## With the use of this feature, users can rate or comment on a recipe for future users, giving them useful information about any improvements that might be made.

## Function 9: Add Favorites

## The recipes can be added to the user's favorite favorites lists, making it simple for them to access them.

## Function 10: Create Daily/ Weekly Menu

This provides users the flexibility to create a daily or weekly meal based on their preferences.

**Resources provided by the phone infrastructure will be required to implement the app functionality:**

connectivity options – internet connectivity required

database system used: Firebase/ Sql

Required OS: Android 7.0 and up

RAM: 2GB and up

Memory required: minimum 50 Mb free

**Experience level of team members:**

1. **Mrunmai Magar: Experience Level**: Beginner

**Project Name**: MyCalculator

**Project Description:** Built the MyCalculator Android app using Java. The application's functionalities include addition, subtraction, multiplication, and division as well as other arithmetic operations. This knowledge will assist us in designing the user interface for the application as well as in doing mathematical computations like cooking times, also in documentation of the project. The program was tested using unit, integration, black-box, and white-box testing.

**Programming Language used**: Java

**Testing**: Unit, Integration, Black-Box, White Box testing.

1. **Neha More: Experience Level**: Beginner

**Project Name**: Smart Complaint Box

**Project Description:** The project's goal is to create the "Smart Complaint Box" application, which will allow users to register complaints and find out information about their location and address problems online without having to visit an officer repeatedly until the issue is resolved. The people can eliminate corruption in government agencies and save time using this technique. Its main goal is to offer an intuitive and simple method using Android. This knowledge will help us to work with database and design UML diagrams in our ongoing project.

**Programming Language used:** Java

**Database:** MySQL

**Testing:** Unit, Integration, GUI, White-Box Testing

1. **Utkarsh Sonawane: Experience Level:** Beginner **Experience**: Attended a workshop on Android Studio **Project Name**: Smart Blood Bank

**Project Description**: Using this application the patient can request for Blood and the nearest donor gets a notification that someone needs blood. Using this application, it becomes easy to find a blood donor in less time. This experience will help us in various areas of our project like maintaining database, creating account etc.

**Database Used**: Firebase

**Testing**: Unit, Integration, GUI, White-Box Testing, Acceptance Testing.

1. **Tanmay Kakade: Experience Level:** Beginner **Experience**: Attended a workshop on Android Studio **Project Name**: Rent Management System

**Project Description**: The management of housing rent was the focus of this application's creation. Owners will have less work to do and will be able to manage their properties more effectively if they use this application. It keeps track of information about tenants and their rent. It determines the entire rent, which accounts for utilities like electricity, water, and gas, and divides it among the renters, making things simple for the landlord and saving them time. This experience will help us in areas like registering customers, division of recipes and ingredients, documentation of the project and designing UML diagrams.

**Database Used**: MySQL

**Testing**: Unit, Integration, GUI, White-Box Testing, Acceptance Testing.

1. **Aditya Vichare: Experience Level:** Beginner **Experience**: Undergraduate degree project **Project Name**: Medical Assistance Application

**Project Description**: Built a Medical Assistance App using programming languages like Java and XML. Using this application, the user can login to find nearby Hospital details along with the list of available doctors and their contact information, as also book an appointment with them. The user can also find information related to the medical stores and available medicines, Clinics nearby the user’s location and Blood Banks.

**Database Used**: MySQL

**Testing**: Unit, GUI, White-Box Testing, Acceptance Testing.

# Requirements

|  |  |  |
| --- | --- | --- |
| ReqID | Requirement Statement | Line refernce |
| R1 | The system shall provide registration window for new users and authenticated login for users to access Ricetta app. | 3 |
| R2 | The system shall provide a selection pane to separate recipes into categories like soup recipes, salad recipes, breakfast recipes, etc. | 4,5 |
| R3 | The system shall suggest recipes of similar kind based on the choice user has chosen. User choices can be type of meal like soup, salad,etc. | 6,7 |
| R4 | The system shall allow users to view the nutritional values of a chosen recipe. | 8,9 |
| R5 | The system shall provide a list of ingredients with their respective quantity based on the recipe selected. | 10,11 |
| R6 | The system shall provide users to view how much preparation time and cooking time is required to make the dish. | 12,13 |
| R7 | The system shall provide approximately what number of servings can be made after following the recipe. | 14,15 |
| R8 | The system shall provide a complete description and instructions about how the dish should be prepared, cooked, and served. | 16 |
| R9 | The system shall provide a feature where users can rate/ comment on a recipe, and to view previous feedback. | 17,18 |
| R10 | The system shall provide the ﬂexibility to add favorite recipes into their favorite list. | 19,20 |
| R11 | The system shall provide users the ﬂexibility to create a daily or weekly meal based on their preferences. | 21 |

**Use Case List**

|  |  |
| --- | --- |
| **Use Case#** | **Use Case Name** |
| UC1 | User Registeration/ Login |
| UC2 | Display Categories |
| UC3 | Recipe List |
| UC3.1 | Recipe Information |
| UC3.2 | Nutritional Values |
| UC3.3 | Write Feedback |
| UC3.4 | Provide Review |
| UC3.5 | Number of Servings |
| UC3.6 | Add recipe to favourites |
| UC4 | Suggest Similar Recipes |
| UC5 | Create Daily/ Weekly Menu |

* **UC1: Login**

# High Level Use Cases

-TUCBW the user enters his credentials, and clicks on login button

-TUCEW the user gets logged in and the home screen is displayed.

## UC2: Display Categories

-TUCBW the user clicks on display categories button.

-TUCEW the system provides a selection pane to separate recipes into categories.

## UC3: View Recipe

-TUCBW the user selects a recipe and clicks on its name.

-TUCEW the application opens user-selected recipe

## UC3.1: Show ingredients

-TUCBW the user scrolls down after choosing a recipe to ‘list of ingredients’’ section.

-TUCEW application shows a brief list containing all the ingredients required to prepare the dish using the selected recipe.

## UC3.2: View Time Required

-TUCBW the user scrolls down after choosing a recipe to ‘time required’ section.

-TUCEW the section informs about preparation time and cooking time required for the recipe.

## UC3.3: Show Nutrition Chart

-TUCBW the user clicks a button to see selected recipe’s nutritional values.

-TUCEW all the nutritional elements present in that dish are displayed after clicking.

## UC3.4: View Reviews/ Provide Feedback

-TUCBW the user views a button to show previously submitted reviews and to add a new one

-TUCEW all the previously submitted reviews are displayed and user also can get an option to add his own feedback/rating

## UC3.5: View Number of Servings

-TUCBW the user scrolls down after choosing a recipe to ‘number of servings’ section.

-TUCEW this section displays how many people can enjoy the dish after they follow given recipe.

## UC3.6: Add to Favorites list

-TUCBW the user sees an option along with the recipe to bookmark it for subsequent use

-TUCEW after clicking on ‘add to favorites’ the user adds the recipe to a list which later can be referred to in future, if needed.

## UC4: Suggest Similar Recipes

-TUCBW the user scrolls down after choosing a recipe to ‘Similar to this’ section.

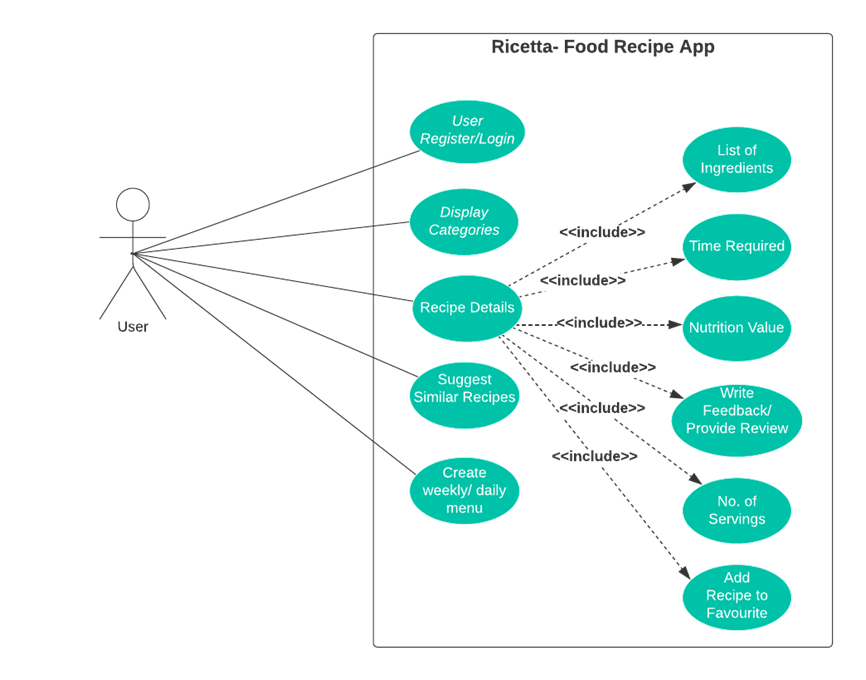
-TUCEW recipes which are of similar kind are displayed to get a variety of similar choice

## UC5: Create Daily/ Weekly Menu

-TUCBW users have an option to create a daily or weekly menu.

-TUCEW the user now has the ﬂexibility to see a daily or weekly meal based on their preferences.

# Use Case Diagram



**Requirements to Use Case Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Priority Weight** | **UC1** | **UC2** | **UC3** | **UC3.1** | **UC3.2** | **UC3.3** | **UC3.4** | **UC3.5** | **UC3.6** | **UC4** | **UC5** |
| **R1** | 1 | X |  |  |  |  |  |  |  |  |  |  |
| **R2** | 1 |  | X |  |  |  |  |  |  |  |  |  |
| **R3** | 4 |  |  |  |  |  |  |  |  | X |  |  |
| **R4** | 2 |  |  |  |  |  | X |  |  |  |  |  |
| **R5** | 2 |  |  |  | X |  |  |  |  |  |  |  |
| **R6** | 2 |  |  |  |  | X |  |  |  |  |  |  |
| **R7** | 2 |  |  |  |  |  |  |  | X |  |  |  |
| **R8** | 1 |  |  | X |  |  |  |  |  |  |  |  |
| **R9** | 3 |  |  |  |  |  |  | X |  |  |  |  |
| **R10** | 3 |  |  |  |  |  |  |  |  |  | X |  |
| **R11** | 4 |  |  |  |  |  |  |  |  |  |  | X |
|  | **SCORE** | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 2 | 3 | 3 | 4 |

**NOTE: Priority 1 is highest priority, work this ﬁrst**

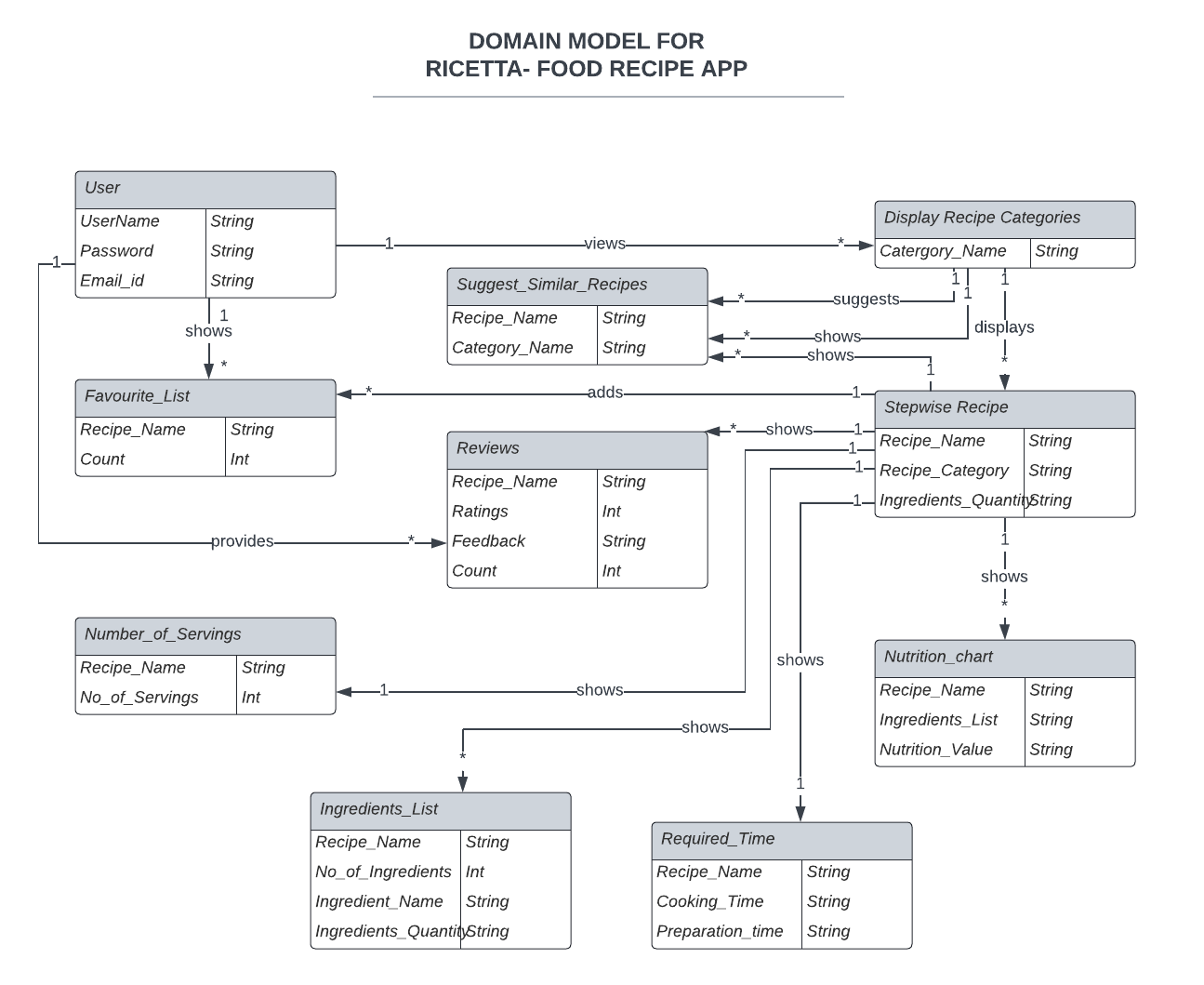
# Increment Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case** | **Priority** | **Effort(person-weeks)** | **Depends on** | **Assigned to** | **Iteration 1**  **(Due:02/27/23 )** | **Iteration 2**  **(Due:04/03/23)** | **Iteration 3**  **(Due:04/19/23 )** |
| **UC1** | 1 | 2 | None | TK, AV | 2 |  |  |
| **UC2** | 1 | 3 | UC1 | US | 2 | 1 |  |
| **UC3** | 1 | 2 | UC2 | NM |  | 2 |  |
| **UC3.1** | 2 | 1 | UC2,UC3 | MM |  |  |  |
| **UC3.2** | 2 | 1 | UC2,UC3 | MM |  | 1 |  |
| **UC3.3** | 2 | 1 | UC2,UC3 | NM |  | 1 |  |
| **UC3.4** | 3 | 2 | UC2,UC3 | US,NM |  |  |  |
| **UC3.5** | 2 | 1 | UC2,UC3 | TK |  |  |  |
| **UC3.6** | 3 | 2 | UC2,UC3 | AV |  |  |  |
| **UC4** | 3 | 3 | UC2 | TK,MM |  |  |  |
| **UC5** | 4 | 4 | UC1 | AV,NM |  |  |  |
| **Total effort** |  | **22** |  |  | **4** | **5** |  |

1 Person-Week = 5 hrs

Team Members: Tanmay K, Mrunmai M, Neha M, Utkarsh S, Aditya V

# Domain Diagram



**Task List**

We are going to work on Use Case 1 and Use Case 2 for the ﬁrst Iteration. UC1: Register/ Login

UC2: Display Categories