**Software Engineering II Project**

**Digital Cookbook**

Part task: Video Instructions Upload

**Group H**

Yang Chen

Ziheng Miao

Jingxuan Liu

**Content**

[**1.** **Specification** 3](#_Toc103773995)

[**1.1.** **Description** 3](#_Toc103773996)

[**1.1.1.** **Digital cookbook** 3](#_Toc103773997)

[**1.1.2.** **Additional task** 3](#_Toc103773998)

[**1.2.** **User characteristics** 3](#_Toc103773999)

[**1.3.** **Data** 3](#_Toc103774000)

[**1.4.** **Functional requirements** 3](#_Toc103774001)

[**1.5.** **Non-functional requirements** 3](#_Toc103774002)

[**2.** **Architecture** 4](#_Toc103774003)

[**2.1.** **Use case diagrams** 4](#_Toc103774004)

[**2.2.** **Structure/MVC** 4](#_Toc103774005)

[**2.3.** **Class diagrams** 4](#_Toc103774006)

[**3.** **GUI Design** 5](#_Toc103774007)

[**3.1.** **Structure/wireframes** 5](#_Toc103774008)

[**3.2.** **Application of usability heuristics with screenshots** 5](#_Toc103774009)

[**4.** **Test** 6](#_Toc103774010)

[**4.1.** **EC/BVA** 6](#_Toc103774011)

[**4.1.1.** **Description** 6](#_Toc103774012)

[**4.1.2.** **Results** 6](#_Toc103774013)

[**4.2.** **Usability test** 6](#_Toc103774014)

[**4.2.1.** **Description** 6](#_Toc103774015)

[**4.2.2.** **Evaluation/results** 6](#_Toc103774016)

[**5.** **Evaluation** 7](#_Toc103774017)

[**5.1.** **Group work** 7](#_Toc103774018)

[**5.2.** **Task responsibilities** 7](#_Toc103774019)

# **Specification**

# **Description**

# **Digital cookbook(Mod.)**

Introducing the ultimate culinary companion - our digital cookbook! With our digital cookbook you may discover hundreds of tantalizing recipes from around the world, meticulously curated for your taste buds. Immerse yourself in an interactive cooking experience with step-by-step instructions, stunning visuals, and expert tips. Personalize your profile, save favourite recipes. Whether you're a novice or a seasoned chef, this digital cookbook is your key to unlocking a world of flavours. Don't miss out on this incredible offer – try our digital cookbook now and elevate your cooking skills to new heights.

# **Additional task(Mod.)**

**Video Instructions:**

In addition to images and text, our digital cookbook also offers a special feature of video tutorials uploading. Reference videos can help users learn a wider variety of delicious dishes more quickly.

(Example picture)

# **User characteristics(Mod.)**

1. Tech-savvy: Digital cookbook users are comfortable using technology, such as smartphones, tablets, and computer applications, to access and navigate digital content.

2. Food enthusiasts: Users of digital cookbooks are typically passionate about food and cooking. They enjoy experimenting with recipes, discovering new flavours, and expanding their culinary skills.

3. Convenience seekers: Digital cookbook users appreciate the convenience of having a vast collection of recipes readily accessible on their devices. They value the ability to search for specific recipes, save favourites, and access them anytime, anywhere.

4. Visual learners: Many digital cookbook users are visual learners who prefer instructional content presented through images, videos, and step-by-step visuals. They find it easier to follow recipes when they can see the process visually.

# **Data(New!)**

1.Ingredient Table structures are as follows:

|  |  |
| --- | --- |
| field | var\_type |
| name | varchar(255) |
| unit | varchar(255) |

SQL file: cookbookdb.sql, Table: tbl\_ingredient

Default records of this table are as follows:

|  |  |
| --- | --- |
| name | unit |
| fish | g |
| green vegetable | g |
| meat | g |
| salt | g |
| soy sauce | ml |
| sugar | ml |
| water | ml |

2.Recipe Table structures are as follows:

|  |  |
| --- | --- |
| field | var\_type |
| id | char(32) |
| name | varchar(255) |
| steps | varchar(2000) |
| people\_number | int |
| video\_URL | varchar(255) |
| img\_URL | varchar(255) |

SQL file: cookbookdb.sql, Table:tbl\_recipe

Default records of this table are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | name | steps | People\_  number | Video\_  URL | img\_  URL |
| 21f2a340a5944923853a2b371ea4335b | Sweet-sour Pork | Put all ingredients into the pot and cook | 1 | /// | /// |
| 9f3a5bbfe72d4a0e910367986d25e30a | Sweet Pork | Put all ingredients into the pot and cook | 5 | /a/ | /b/ |
| a86104ac399742dd9d848eaecd13cbc0 | Red-cooked Pork | Put all ingredients into the pot and cook | 2 | /// | /// |
| c3473e65eebf4dc29816b0bc5a0e1f9f | Red-cooked Pork | Put all ingredients into the pot and cook | 2 | /// | /// |
| c6ea62fce487444b957add1e3fad8f57 | Red-cooked Pork | Put all ingredients into the pot and cook | 2 | /// | /// |
| cacc0c75085a485eb677a7b562a71ead | Sweet-sour Pork | Put all ingredients into the pot and cook | 1 | /// | /// |
| df543b8a2fe54032915d4fc21b72546a | Sweet-sour Pork | Put all ingredients into the pot and cook | 1 | /// | /// |
| e0e1fd1fa47247798c36cc8fafe9d4fb | Red-cooked Pork | Put all ingredients into the pot and cook | 2 | /// | /// |

3.Recipe-Ingradient Relation Table structures are as follows:

|  |  |
| --- | --- |
| field | var\_type |
| recipe\_id | char(32) |
| ingredient\_name | varchar(255) |
| ingredient\_unit | varchar(255) |
| ingredient\_quantity | float |

SQL file: cookbookdb.sql, Table:tbl\_recipe\_ingredient\_relation

# **Functional requirements**

1.Recipe search:

Users can search for recipes based on keywords, dish names, or ingredients.

2.Recipe details:

Each recipe should have a dedicated page displaying ingredients, quantities, preparation steps, and cooking instructions.

3.Recipe creating:

Users can create personalized recipes in the Digital Cookbook.

4.Recipe deleting:

Users can delete specific recipes they don’t need.

5.Recipe modifying:

Users can modify specific recipes which need improving.

6.Dish size adjustment:

Users can modify the recipe quantities to match their desired number of people.

7.Favorite recipes:

Users can save their favorite recipes for future reference.

# **Non-functional requirements**

1.Usability:

The user interface should be intuitive, visually appealing, and easy to navigate.

2.Performance:

The application should have fast response times and handle a large recipe database and user base efficiently. Page response will not exceed 0.15 seconds.

3.Scalability:

The software architecture should be scalable to accommodate future growth and increased user demand.

4.Reliability:

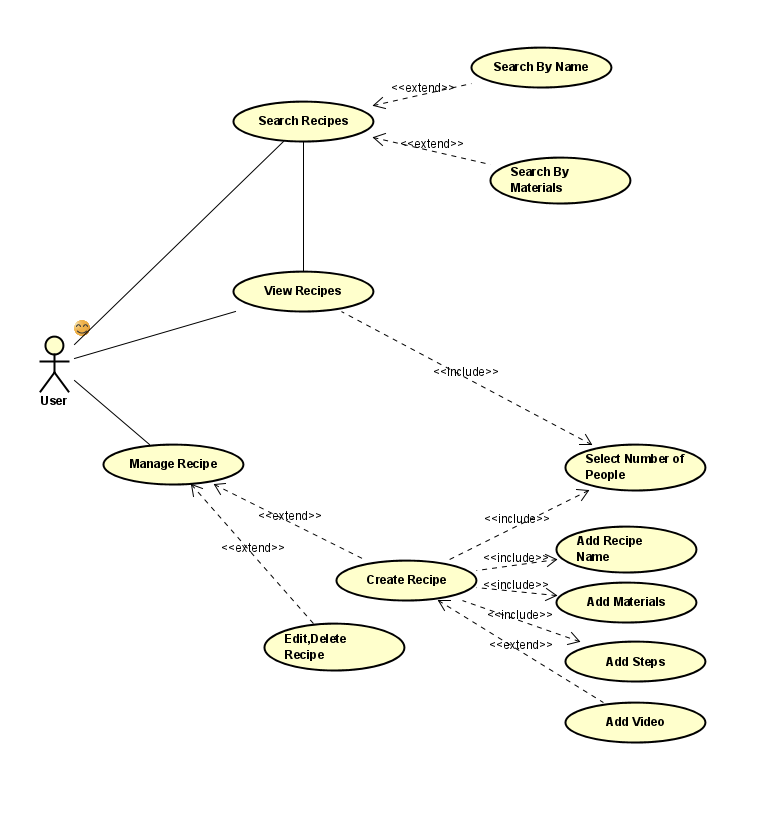
The Digital Cookbook should be reliable, ensuring that recipes, user data are stored securely and accurately. It should minimize the risk of data loss or corruption.

5.Availability:

The application should have a high level of availability, ensuring that users can access the Cookbook and its features at all times, with minimal downtime for maintenance or upgrades.

# **Architecture**

# **Use case diagrams**



# **Structure/MVC(New!)**

## 2.2.1 Intro of MVC

Model-View-Controller is applied in this digital cookbook project as a software architectural pattern. It divides an application into three interconnected components

|  |  |
| --- | --- |
| Model | The data and business logic of the application. It encapsulates the data and defines how it is accessed and manipulated. The model component is responsible for handling data validation, storage, and retrieval. |
| View | The user interface of the application. It defines how the data is presented to the user. The view component focuses on displaying information and gathering user input. It communicates with the model to fetch data and updates the user interface accordingly. |
| Controller | The intermediary between the model and the view. It receives user input from the view and processes it. It interacts with the model to update or retrieve data and updates the view to reflect any changes. The controller component coordinates the flow of data between the model and the view. |

The MVC pattern promotes separation of concerns, making the application code more modular, maintainable, and scalable. It allows developers to work on different components independently, facilitates code reuse, and enhances testability by enabling unit testing for each component separately.

## 2.2.2 MVC Structure in this Digital Cookbook

A major part in this MVC structures is the recipe part, that contains RecipeModel(Model), EditrecipePage(Major View), IngredientAddPane(Component View), CreateRecipePageController(Controller), ModifyRecipePageController(Controller). HomePage, ManagePage, SearchPage fill up the remained simple MVC parts.

MVC structures in details are as follows:

|  |  |
| --- | --- |
| Model | RecipeModel.java  SearchPageModel.java |
| View | **MajorViews:**  CreateRecipePageView.java  HomePageView.java  ManagePageView.java  ModifyRecipePageView.java  SearchPageView.java  ViewRecipePageView.java  **Components:**  TitlePane.java  IngredientAddPane.java  RecipeSummaryPane.java  ManageRecipeSummaryPane.java |
| Controller | ManagePageController.java  CreateRecipePageController.java  ModifyRecipePageController.java  SearchPageController.java |

And Ingredient.java and Recipe.java are the core entities of the MVC.

## 2.2.3 Communication of MVC in this Digital Cookbook

### 2.2.3.1 Communication between Views and Controllers

The communication between views and controllers is typically facilitated through a combination of event handling and data binding mechanisms as follows:

User Interactions: When a user interacts with the view (e.g., clicks a button, enters text in an input field), an event is triggered by the view component.

Event Handling in Controllers: The controller associated with that view component listens for the event. It has registered event listeners or callbacks for specific events within the view.

Controller Updates the Model: Upon receiving the event notification, the controller performs the necessary logic or operations based on the user interaction. This may involve querying or updating the model(s) to reflect the changes.

Data Binding: The updated model(s) notify the view of any changes through a data binding mechanism. This can be achieved using observable patterns or frameworks that facilitate automatic synchronization between the model and view.

View Updates: Upon receiving the update notification from the model(s), the view updates its display to reflect the new data or state. It retrieves the updated data from the model(s) and renders it accordingly.

View-State Changes: Sometimes, the controller may need to directly manipulate the view's state or trigger specific view-related actions. In such cases, the controller may call methods or set properties exposed by the view to perform these operations.

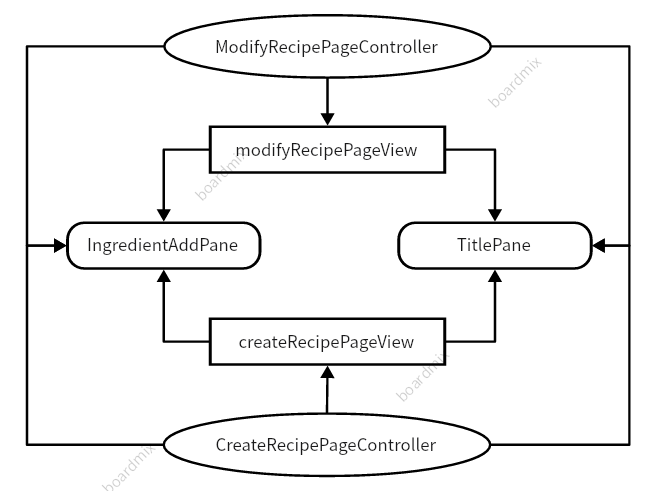
Through this communication process, the view remains decoupled from the business logic and data management, as it delegates those responsibilities to the controller and model(s). The controller acts as an intermediary, facilitating the communication between the view and the model(s), ensuring that the view remains responsive to user interactions and reflects the current state of the application's data.

The View- Controller interactions in Digital Cookbook are listed as follows:

|  |  |
| --- | --- |
| View | Controller |
| ManagePageView.java | ManagePageController.java |
| ModifyRecipePageView.java | ModifyRecipePageController.java |
| CreateRecipePageView.java | CreateRecipePageController.java |
| SearchPageView.java | SearchPageController.java |

In the traditional MVC architecture, a view is typically associated with a single controller. However, in this Digital Cookbook, conditions exists that multiple controllers associated with a single view. This is because some of Views are Composite Views that consist of Reusable Components. A complex relationship is established in this condition. Each components that are designed to be reusable across different Major Views and each of them have their own dedicated controllers to manage their specific behavior, so a single Major View may have more than one controller. On the other hand, sub controllers of Components can only work in conjunction with a higher-level controller that coordinates the overall behavior of the Major view, so a components may have more than one controller too.

It is important to note that having multiple controllers for a single view should be done judiciously. It increases the complexity of the code and requires careful coordination between controllers to avoid conflicts and ensure consistency. The following diagram illustrated one of the multi controller conditions in Digital Cookbook:



### 2.2.3.1 Communication between Models and Controllers

The communication between models and controllers is fundamental for implementing business logic and data management. Here's how the communication typically occurs:

Model Initialization: The controller creates or retrieves the model instance and initializes it with the necessary data or state.

Data Manipulation in Controllers: The controller interacts with the model to perform various operations, such as reading, updating, or deleting data. It invokes methods on the model to handle these operations.

Responding to User Input: When the user interacts with the view, such as entering data or triggering an action, the associated controller receives the input. It then communicates with the model to update its state or trigger appropriate actions based on the user input.

Updating the Model: The controller modifies the data or state of the model based on the user input, business logic, or other factors. It calls appropriate methods on the model to update its properties, trigger events, or invoke relevant operations.

Retrieving Data from the Model: Controllers can query the model for data required to update the view or to perform specific operations. They can retrieve the requested data through appropriate methods provided by the model.

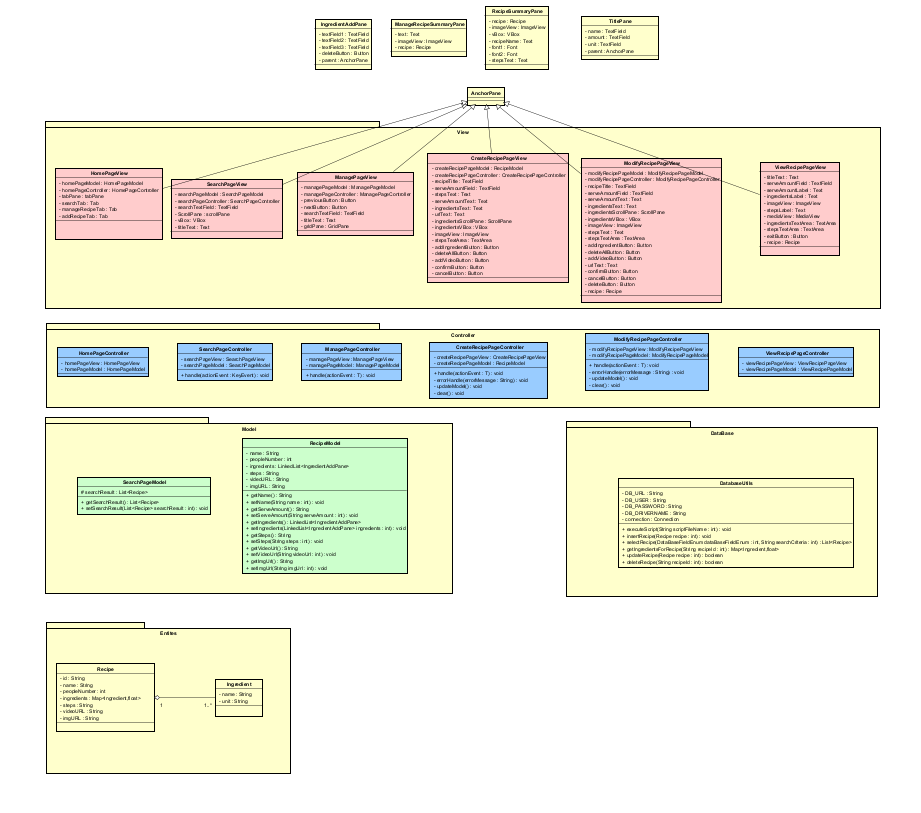
By facilitating communication between models and controllers, the MVC pattern ensures separation of concerns. The models encapsulate data and business logic, while controllers manage the flow of information and coordinate the interaction between the view and the model. This separation allows for modularity, flexibility, and easier maintenance of the application's codebase.

The Model - Controller interactions are listed as follows:

|  |  |
| --- | --- |
| Model | Controller |
| RecipeModel.java | CreateRecipePageController.java  ModifyRecipePageController.java |
| SearchPageModel.java | SearchPageController.java |

RecipeModel is the most vital Model in this MVC structure so it is related to more than one controllers.

# **Class diagrams(Mod.)**



5 main part is illustrated in the diagram: Model, View, Pane Component, Controller, and Entity Class:

Model: contains two models: RecipeModel, SearchPageModel, . Each model includes the constructors, getters, setters.

View: contains six files: CreateRecipePageView, HomePageView, ManagePageView, ModifyRecipePageView, SearchPageView, ViewRecipePageView. Each will be loaded in the Main class functioning as the GUIs.

Pane Component: contains 3 panes: ShowRecipeListPane, ShowRecipePane, ShowRecipeEditPane. Each can be loaded in Major Views with corresponding controller.

Controller: contains four controllers: CreateRecipePageController, ManagePageController, ModifyRecipePageController, SearchPageController. Each implement the logic layer of their corresponding interfaces, composed of  
commands that take inputs and convert them to a model or view.

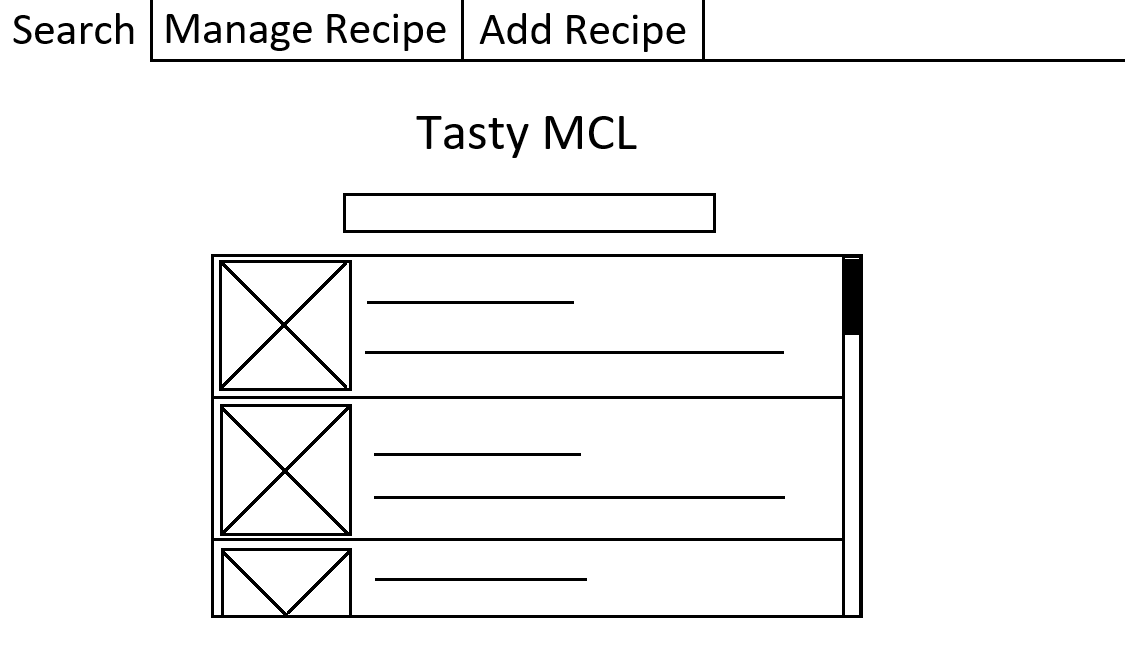
Entity Class: contains 2 classes: Recipe and Ingradient. These 2 classes are the core entities of the whole MVC, defining the data structures of this system.

# **GUI Design**

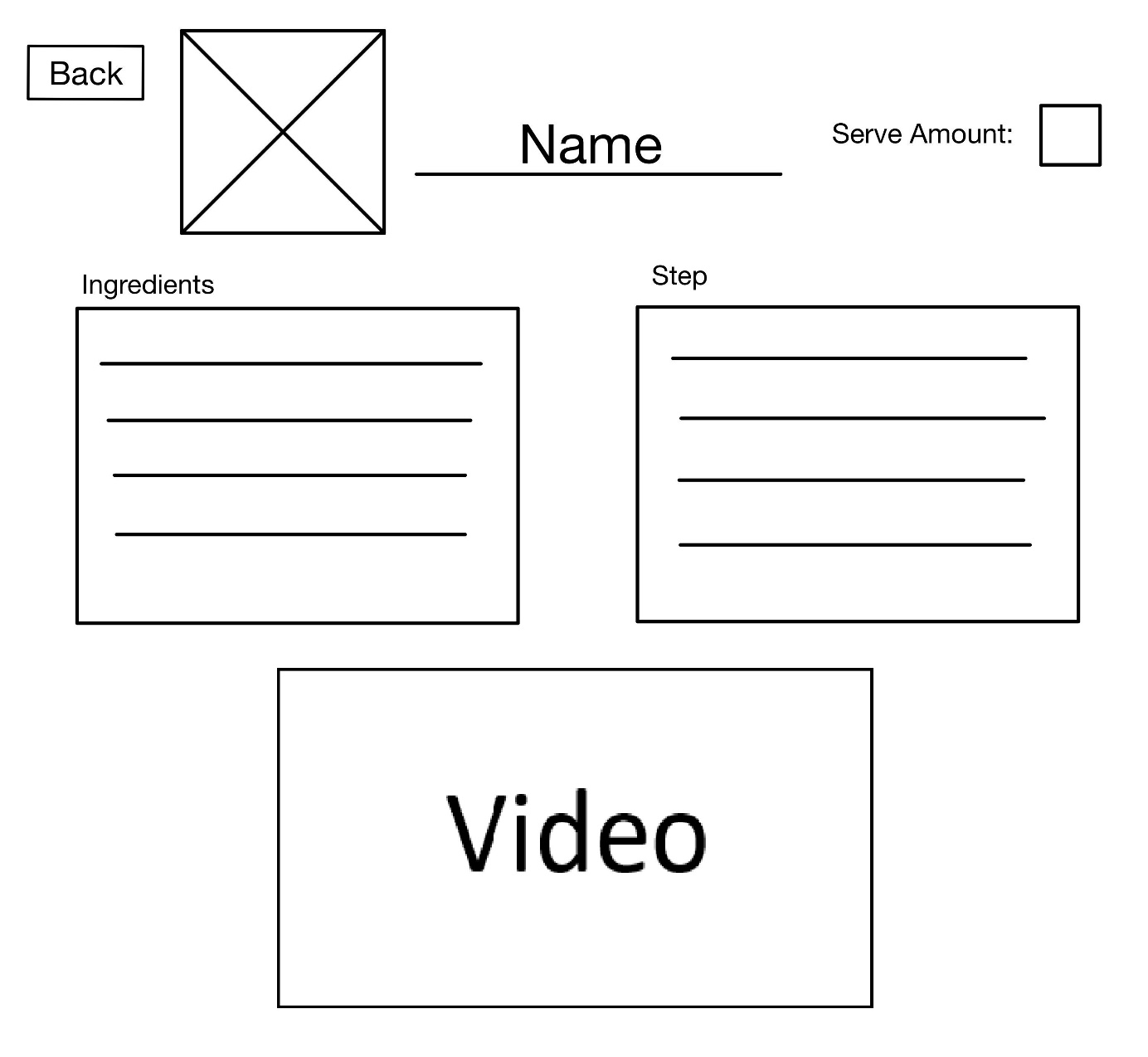
# **Structure/wireframes**

We design to use the pattern of Tabpage to manage our pages. When the user opens this application, he will be on the search page by default. We have three basic pages that can be switched through the tabs on the top of the application which are the “Search”, “Manage Recipe” and “Create Recipe”. When he wants to switch between the three basic pages, he just need to click the these tabs.

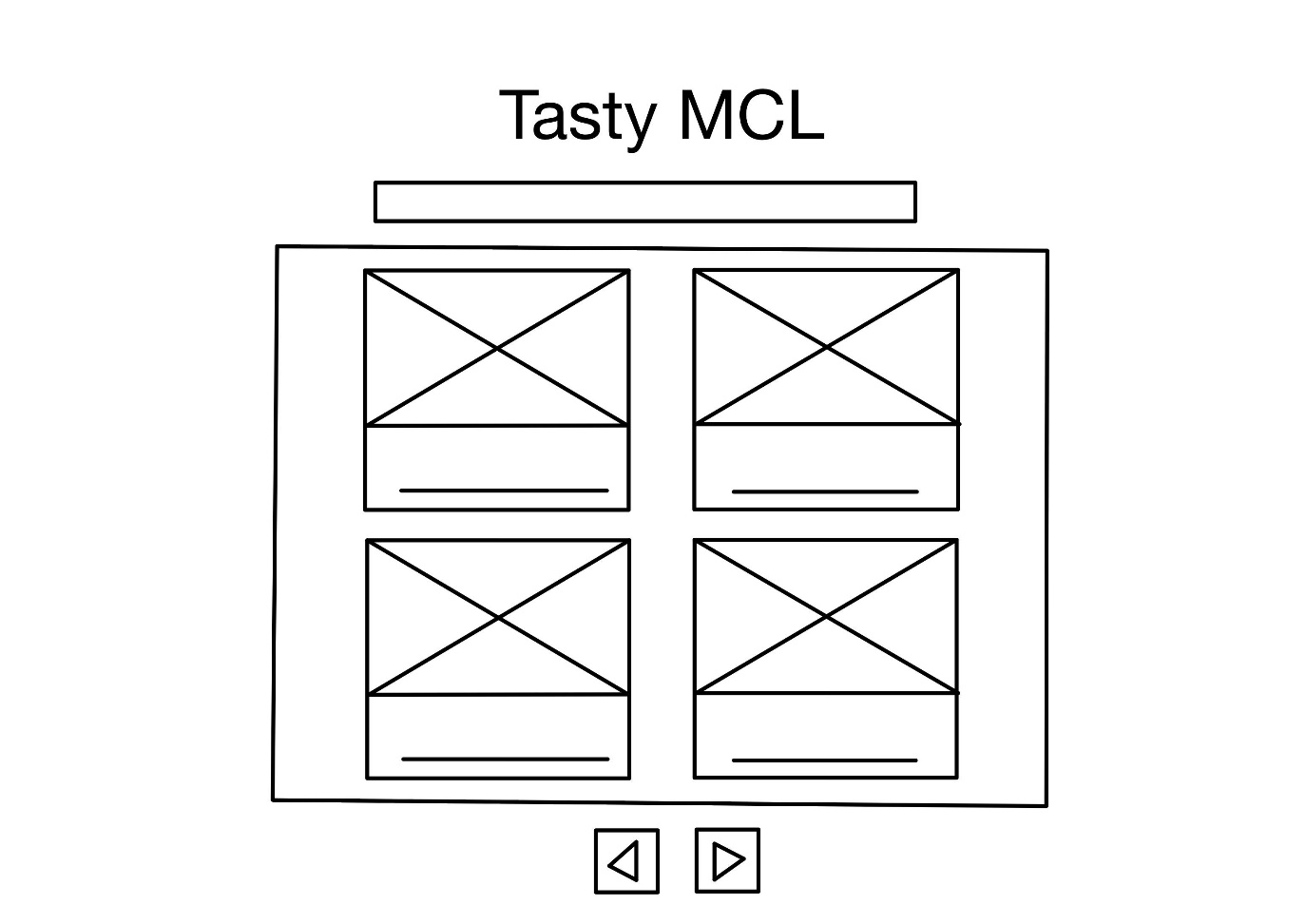
Search Page



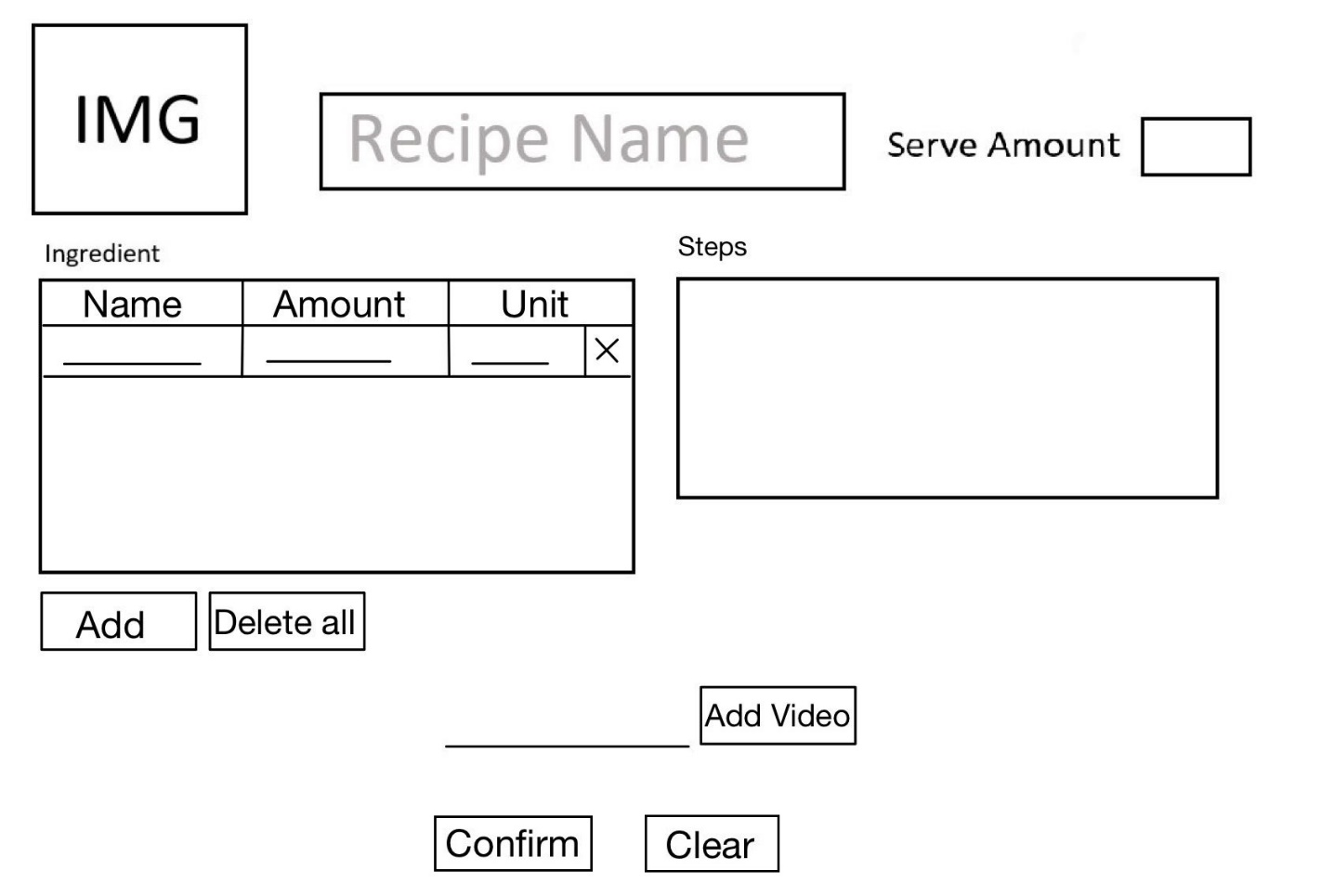
View Recipe Page:



Manage Recipe Page:



Create and Modify Recipe Page:

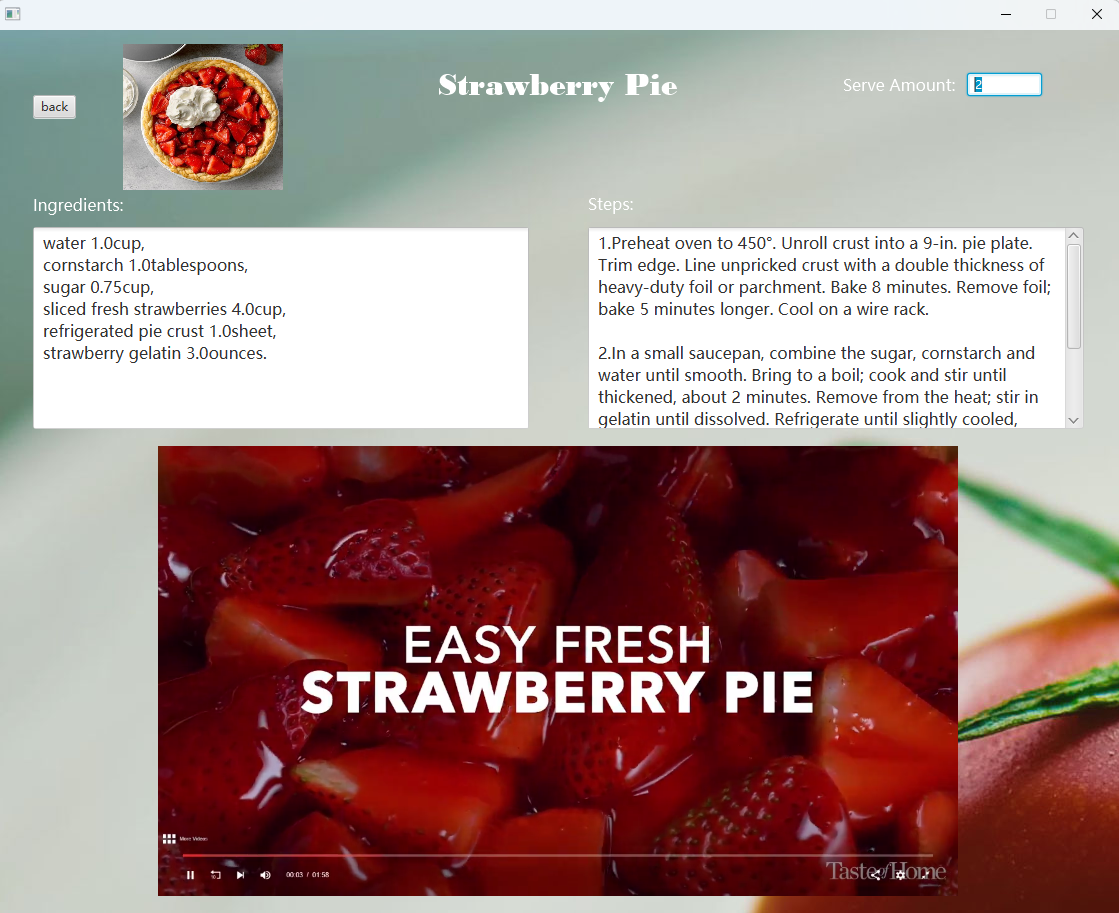


# **Application of usability heuristics with screenshots**

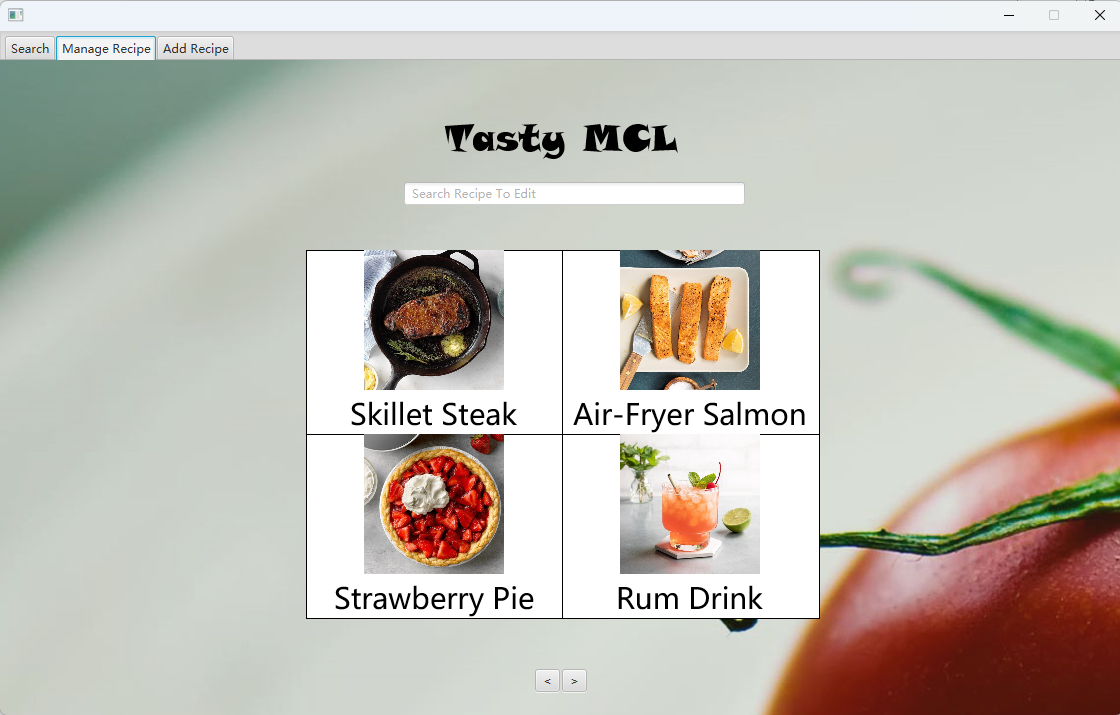
Search Page:



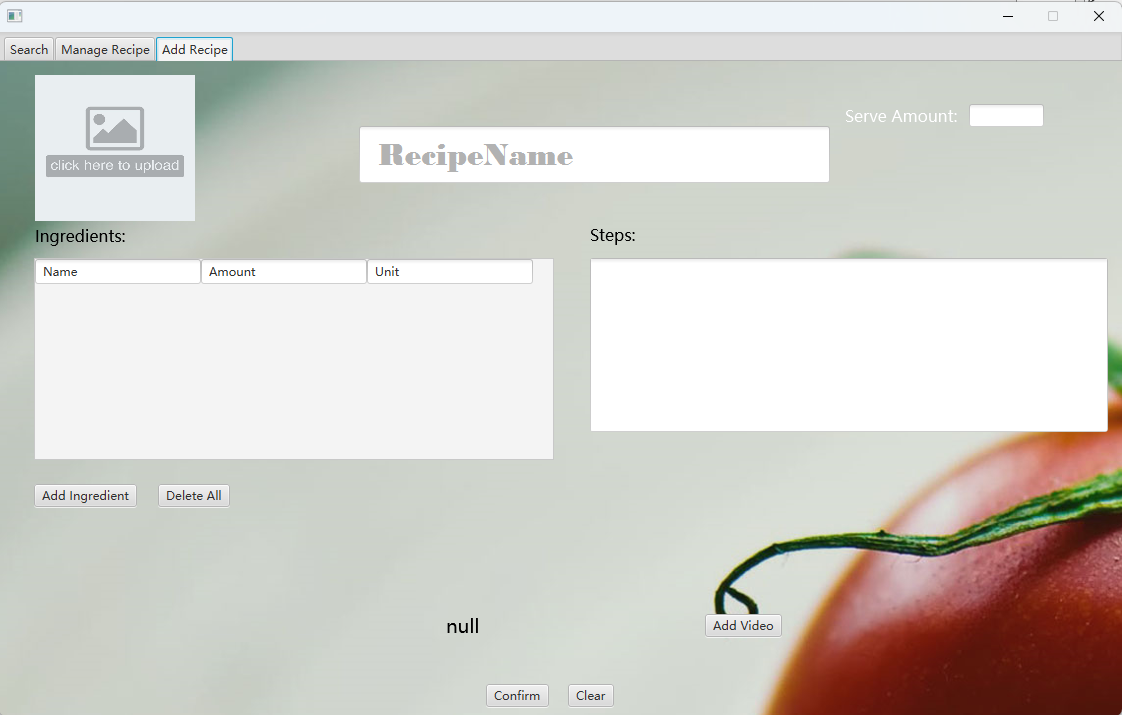
View Recipe Page:



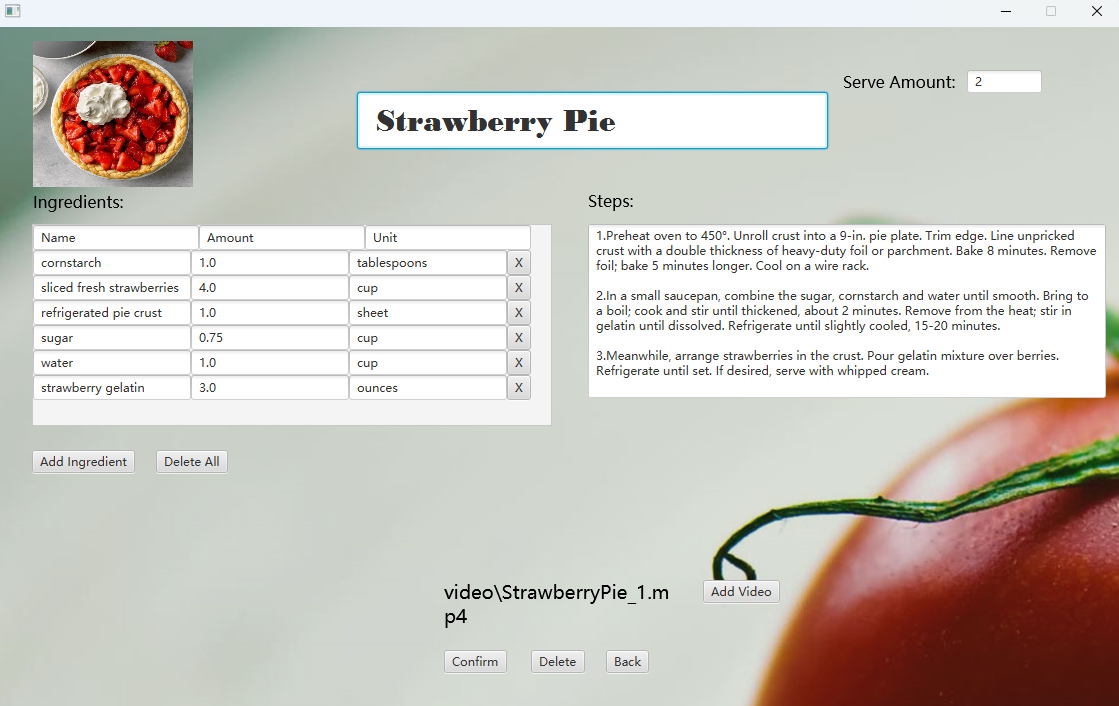
Manage Recipe Page:



Create Recipe Page:



Modify Recipe Page

****

# **Test**

# **EC/BVA(New!)**

Equivalence Class (EC) and Boundary Value Analysis (BVA) testing are two techniques used to design effective and efficient test cases.

Equivalence Class Testing: Equivalence Class Testing is based on the concept that inputs within the same equivalence class should exhibit similar behaviour or produce the same output. This technique helps reduce redundant test cases while ensuring sufficient coverage. It involves identifying groups of input values that are expected to be processed similarly by the system. Test cases are then designed to represent each equivalence class.

Boundary Value Analysis: Boundary Value Analysis focuses on testing values that lie on or near the boundaries of input domains. Since defects often occur at the boundaries, this technique helps identify potential issues effectively. The goal is to test both the lower and upper boundaries, as well as values just inside and outside those boundaries.

# **EC Test Cases and Results**

## 4.1.1.1 EC Class For Create Recipe Page

**1. Recipe Name**

Valid EC for Recipe Name ECRecipeName1 = {value | value∈not null String}

Invalid EC for Recipe Name ECRecipeName2 = {value | value=”” (empty String)}

**2. Serve Amount**

Valid EC for Serve Amount ECServeAmount1 = {value | 0<value< 2^31 – 1, value ∈ Z}

Invalid EC for Serve Amount ECServeAmount2 = {value | value<=0, value ∈ Z }

Invalid EC for Serve Amount ECServeAmount3 = {value | value ∉ Z}

Invalid EC for Serve Amount ECServeAmount4 = {value | value is not a number}

Invalid EC for Serve Amount ECServeAmount5 = {value | value is null}

**3. ImageURL Format**

Valid EC for ImageURL Format ECImageURLFormat1 = {value | value∈["jpg", "jpeg", "png", "gif", "bmp", "wbmp", "tiff", "tif" ,"ico=",null(no image)]}

Invalid EC for ImageURL Format ECImageURLFormat2 = {value | value∉ ["jpg","jpeg","png","gif","bmp","wbmp","tiff","tif","ico",null(no image)]}

**4. Ingredient Name**

Valid EC for Ingredient Name ECIngredientName1 = {value | value∈not null String}

Invalid EC for Ingredient Name ECIngredientName2 = {value | value=”” (empty String)}

**5. Ingredient Amount**

Valid EC for Ingredient Amount ECIngredientAmount1 = {value | value>0}

Invalid EC for Ingredient Amount ECIngredientAmount2 = {value | value<=0}

Invalid EC for Ingredient Amount ECIngredientAmount3 = {value | value is not a number}

**6. Ingredient Unit**

Valid EC for Ingredient Unit ECIngredientUnit1 = {value | value∈not null String}

Invalid EC for Ingredient Unit ECIngredientUnit2 = {value | value=”” (empty String)}

**7. Ingredient List Size**

Valid EC for Ingredient List Size ECIngredientListSize1 = {value | value > 0, value∈Z}

Invalid EC for Ingredient List Size ECIngredientListSize2 = {value | value = 0}

**8. Duplicated (Same Ingredient Name and Unit) Ingredient Numbers in List**

Valid EC for Duplicated Ingredient Numbers ECDuplicatedIngredientNumber1 = {value | value=0}

Invalid EC for Duplicated Ingredient Numbers ECDuplicatedIngredientNumber2 = {value | value > 0}

**9. VideoURL Format**

Valid EC for VideoURL Format ECVideoURLFormat1 = {value | value∈["mp4", null(no video)]}

Invalid EC for Video Size ECVideoSize2 = {value | value ∉ ["mp4", null(no video)]}

**10. Video Size**

Valid EC for Video Size ECVideoSize1 = {value | 0(No Video)<=value <= 25MB}

Invalid EC for Video Size ECVideoSize2 = {value | value > 25MB }

## 4.1.1.2 EC Test for Create Recipe Page

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **1** | **2** |
| **Recipe Name** | “” | Rum Drink |
| **Serve Amount** | 1 | 0 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | Salt | rum, sugar |
| **Ingredient Amount** | 10 | 1,1.5 |
| **Ingredient Unit** | g | cup, teaspoon |
| **Ingredient List Size** | 1 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName2, ECServeAmount1, ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1, ECDuplicatedIngredientNumber1, ECVideoURLFormat1, ECVideoSize1 | ECRecipeName1,  ECServeAmount2,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **3** | **4** |
| **Recipe Name** | Rum Drink | Rum Drink |
| **Serve Amount** | 1.11 | aaa |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | rum, sugar | rum, sugar |
| **Ingredient Amount** | 1,1.5 | 1,1.5 |
| **Ingredient Unit** | cup, teaspoon | cup, teaspoon |
| **Ingredient List Size** | 2 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount3,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount4,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **5** | **6** |
| **Recipe Name** | Rum Drink | Salad |
| **Serve Amount** | null | 4 |
| **ImageURL Format** | jpg | pdf |
| **Ingredient Name** | rum, sugar | vegetable, sugar, ketchup |
| **Ingredient Amount** | 1,1.5 | 1,1.5,100 |
| **Ingredient Unit** | cup, teaspoon | kg, teaspoon,g |
| **Ingredient List Size** | 2 | 3 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount5,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat2,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **7** | **8** |
| **Recipe Name** | Steak | Steak |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | “” | steak |
| **Ingredient Amount** | 1 | -1.11 |
| **Ingredient Unit** | kg | kg |
| **Ingredient List Size** | 1 | 1 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName2,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount2,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **9** | **10** |
| **Recipe Name** | Steak | Steak |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | steak | steak |
| **Ingredient Amount** | bbb | 1 |
| **Ingredient Unit** | kg | “” |
| **Ingredient List Size** | 1 | 1 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount3,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit2,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **11** | **12** |
| **Recipe Name** | Egg Roll | Egg Roll |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** |  | flour, flour |
| **Ingredient Amount** |  | 1,0.5 |
| **Ingredient Unit** |  | kg, kg |
| **Ingredient List Size** | 0 | 1 |
| **Duplicated Ingredient Numbers** |  | 1 |
| **VideoURL Format** | mp4 | mp4 |
| **Video Size** | 19.2MB | 19.2MB |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientListSize2,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber2,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **13** | **14** |
| **Recipe Name** | Cookies | Egg Roll |
| **Serve Amount** | 5 | 5 |
| **ImageURL Format** | png | png |
| **Ingredient Name** | flour, butter | flour, flour |
| **Ingredient Amount** | 1,200 | 1,0.5 |
| **Ingredient Unit** | kg, g | kg, kg |
| **Ingredient List Size** | 2 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | png | mp4 |
| **Video Size** |  | 130MB |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat2 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber2,  ECVideoURLFormat1,  ECVideoSize2 |

## 4.2.1.1 EC Class For View Recipe Page

**1. Serve Amount**

Valid EC for Serve Amount ECServeAmount1 = {value | 0<value< 2^31 – 1, value ∈ Z}

Invalid EC for Serve Amount ECServeAmount2 = {value | value<=0, value ∈ Z }

Invalid EC for Serve Amount ECServeAmount3 = {value | value ∉ Z}

Invalid EC for Serve Amount ECServeAmount4 = {value | value is not a number}

Invalid EC for Serve Amount ECServeAmount5 = {value | value is null}

## 4.2.1.2 EC Test for Create Recipe Page

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **1** | **2** |
| **Serve Amount** | 2 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECServeAmount1 | ECServeAmount2 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **3** | **4** |
| **Serve Amount** | 2.222 | a |
| **ScreenShots** |  |  |
| **Covered EC** | ECServeAmount3 | ECServeAmount4 |

|  |  |
| --- | --- |
| **Inputs/Test No.** | **5** |
| **Serve Amount** | null |
| **ScreenShots** |  |
| **Covered EC** | ECServeAmount5 |

## 4.3.1.1 EC Class For Modify Recipe Page

**1. Recipe Name**

Valid EC for Recipe Name ECRecipeName1 = {value | value∈not null String}

Invalid EC for Recipe Name ECRecipeName2 = {value | value=”” (empty String)}

**2. Serve Amount**

Valid EC for Serve Amount ECServeAmount1 = {value | 0<value< 2^31 – 1, value ∈ Z}

Invalid EC for Serve Amount ECServeAmount2 = {value | value<=0, value ∈ Z }

Invalid EC for Serve Amount ECServeAmount3 = {value | value ∉ Z}

Invalid EC for Serve Amount ECServeAmount4 = {value | value is not a number}

Invalid EC for Serve Amount ECServeAmount5 = {value | value is null}

**3. ImageURL Format**

Valid EC for ImageURL Format ECImageURLFormat1 = {value | value∈["jpg", "jpeg", "png", "gif", "bmp", "wbmp", "tiff", "tif" ,"ico=",null(no image)]}

Invalid EC for ImageURL Format ECImageURLFormat2 = {value | value∉ ["jpg","jpeg","png","gif","bmp","wbmp","tiff","tif","ico",null(no image)]}

**4. Ingredient Name**

Valid EC for Ingredient Name ECIngredientName1 = {value | value∈not null String}

Invalid EC for Ingredient Name ECIngredientName2 = {value | value=”” (empty String)}

**5. Ingredient Amount**

Valid EC for Ingredient Amount ECIngredientAmount1 = {value | value>0}

Invalid EC for Ingredient Amount ECIngredientAmount2 = {value | value<=0}

Invalid EC for Ingredient Amount ECIngredientAmount3 = {value | value is not a number}

**6. Ingredient Unit**

Valid EC for Ingredient Unit ECIngredientUnit1 = {value | value∈not null String}

Invalid EC for Ingredient Unit ECIngredientUnit2 = {value | value=”” (empty String)}

**7. Ingredient List Size**

Valid EC for Ingredient List Size ECIngredientListSize1 = {value | value > 0, value∈Z}

Invalid EC for Ingredient List Size ECIngredientListSize2 = {value | value = 0}

**8. Duplicated (Same Ingredient Name and Unit) Ingredient Numbers in List**

Valid EC for Duplicated Ingredient Numbers ECDuplicatedIngredientNumber1 = {value | value=0}

Invalid EC for Duplicated Ingredient Numbers ECDuplicatedIngredientNumber2 = {value | value > 0}

**9. VideoURL Format**

Valid EC for VideoURL Format ECVideoURLFormat1 = {value | value∈["mp4", null(no video)]}

Invalid EC for Video Size ECVideoSize2 = {value | value ∉ ["mp4", null(no video)]}

**10. Video Size**

Valid EC for Video Size ECVideoSize1 = {value | 0(No Video)<=value <= 25MB}

Invalid EC for Video Size ECVideoSize2 = {value | value > 25MB }

## 4.3.1.2 EC Test for Modify Recipe Page

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **1** | **2** |
| **Recipe Name** | “” | Rum Drink |
| **Serve Amount** | 1 | 0 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | Salt | rum, sugar |
| **Ingredient Amount** | 10 | 1,1.5 |
| **Ingredient Unit** | g | cup, teaspoon |
| **Ingredient List Size** | 1 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName2, ECServeAmount1, ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1, ECDuplicatedIngredientNumber1, ECVideoURLFormat1, ECVideoSize1 | ECRecipeName1,  ECServeAmount2,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **3** | **4** |
| **Recipe Name** | Rum Drink | Rum Drink |
| **Serve Amount** | 1.11 | aaa |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | rum, sugar | rum, sugar |
| **Ingredient Amount** | 1,1.5 | 1,1.5 |
| **Ingredient Unit** | cup, teaspoon | cup, teaspoon |
| **Ingredient List Size** | 2 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount3,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount4,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **5** | **6** |
| **Recipe Name** | Rum Drink | Salad |
| **Serve Amount** | null | 4 |
| **ImageURL Format** | jpg | pdf |
| **Ingredient Name** | rum, sugar | vegetable, sugar, ketchup |
| **Ingredient Amount** | 1,1.5 | 1,1.5,100 |
| **Ingredient Unit** | cup, teaspoon | kg, teaspoon,g |
| **Ingredient List Size** | 2 | 3 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount5,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat2,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **7** | **8** |
| **Recipe Name** | Steak | Steak |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | “” | steak |
| **Ingredient Amount** | 1 | -1.11 |
| **Ingredient Unit** | kg | kg |
| **Ingredient List Size** | 1 | 1 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName2,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount2,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **9** | **10** |
| **Recipe Name** | Steak | Steak |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | steak | steak |
| **Ingredient Amount** | bbb | 1 |
| **Ingredient Unit** | kg | “” |
| **Ingredient List Size** | 1 | 1 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount3,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit2,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **11** | **12** |
| **Recipe Name** | Egg Roll | Egg Roll |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** |  | flour, flour |
| **Ingredient Amount** |  | 1,0.5 |
| **Ingredient Unit** |  | kg, kg |
| **Ingredient List Size** | 0 | 1 |
| **Duplicated Ingredient Numbers** |  | 1 |
| **VideoURL Format** | mp4 | mp4 |
| **Video Size** | 19.2MB | 19.2MB |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientListSize2,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat1,  ECVideoSize1 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber2,  ECVideoURLFormat1,  ECVideoSize1 |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **13** | **14** |
| **Recipe Name** | Cookies | Egg Roll |
| **Serve Amount** | 5 | 5 |
| **ImageURL Format** | png | png |
| **Ingredient Name** | flour, butter | flour, flour |
| **Ingredient Amount** | 1,200 | 1,0.5 |
| **Ingredient Unit** | kg, g | kg, kg |
| **Ingredient List Size** | 2 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | png | mp4 |
| **Video Size** |  | 130MB |
| **ScreenShots** |  |  |
| **Covered EC** | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber1,  ECVideoURLFormat2 | ECRecipeName1,  ECServeAmount1,  ECImageURLFormat1,  ECIngredientName1,  ECIngredientAmount1,  ECIngredientUnit1,  ECIngredientListSize1,  ECDuplicatedIngredientNumber2,  ECVideoURLFormat1,  ECVideoSize2 |

# **BVA Test Cases and Results**

## 4.1.2.1 Serve Amount in Create Page and View Page and Modify Page

**Serve Amount**

**Boundary value analysis for suited values**

|  |  |  |
| --- | --- | --- |
| **0** | **1 2^31-1** | **2^31** |
| **Invalid** | **Valid** | **Invalid** |

**BVA Test**

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **1** | **1** |
| **Recipe Name** | Rum Drink | Rum Drink |
| **Serve Amount** | **0** | **1** |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | rum, sugar | rum, sugar |
| **Ingredient Amount** | 1,1.5 | 1,1.5 |
| **Ingredient Unit** | cup, teaspoon | cup, teaspoon |
| **Ingredient List Size** | 2 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **3** | **4** |
| **Recipe Name** | Steak | Rum Drink |
| **Serve Amount** | **100** | **2^31-1(2147483647)** |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | steak | rum, sugar |
| **Ingredient Amount** | 100 | 1,1.5 |
| **Ingredient Unit** | kg | cup, teaspoon |
| **Ingredient List Size** | 1 | 2 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |

|  |  |
| --- | --- |
| **Inputs/Test No.** | **5** |
| **Recipe Name** | Steak |
| **Serve Amount** | **2^31(2147483648)** |
| **ImageURL Format** | jpg |
| **Ingredient Name** | steak |
| **Ingredient Amount** | 100 |
| **Ingredient Unit** | kg |
| **Ingredient List Size** | 1 |
| **Duplicated Ingredient Numbers** | 0 |
| **VideoURL Format** | null |
| **Video Size** | 0 |
| **ScreenShots** |  |

## 4.1.2.2 Ingredient Amount in Create Page and Modify Page????????????

**Ingredient Amount**

**Boundary value analysis for suited values**

|  |  |  |
| --- | --- | --- |
| **0** | **1 2^31-1** | **2^31** |
| **Invalid** | **Valid** | **Invalid** |

**BVA Tests**

|  |  |  |
| --- | --- | --- |
| **Inputs/Test No.** | **7** | **8** |
| **Recipe Name** | Steak | Steak |
| **Serve Amount** | 2 | 2 |
| **ImageURL Format** | jpg | jpg |
| **Ingredient Name** | steak | steak |
| **Ingredient Amount** | **0** | 1 |
| **Ingredient Unit** | kg | kg |
| **Ingredient List Size** | 1 | 1 |
| **Duplicated Ingredient Numbers** | 0 | 0 |
| **VideoURL Format** | null | null |
| **Video Size** | 0 | 0 |
| **ScreenShots** |  |  |

# **Usability test**

# **4.2.1 Description**

1. **Install and setup the application**
   1. We require a windows OS computer for usability test.
   2. We require the system that supports idea and MySQL database.
   3. Test location is in student dormitory in Germany.

**B). Roles of test makers**

* 1. Observer: Ziheng Miao
  2. Technical Support: Yang Chen

**C). Participants**

1. students from IT class, 4 from EE class.

**D). Test Tasks**

**Scenario:** The user wants to view each recipe to learn the cooking steps.

**Usability records:**

**1.** Whether the video correctly displays and all relating information correctly displays.

**2.** When user change serve amount, whether the ingredient amount changes correctly.

**Scenario:** The user wants to change the information about existing recipes because he find the ingredients of recipe is not right.

**Usability records:**

1. Whether the image can be successfully uploaded.
2. Whether the video can be successfully uploaded.
3. Whether the ingredients and steps and recipe name and serve amount can be successfully modified.
4. Error check of user input correctly works.
5. After confirmation, whether the change is committed to database, and the page successfully updated.

**Scenario:** User learned a new recipe he wants to add the recipe to the cookbook.

**Usability records:**

1. Whether the image can be successfully uploaded.
2. Whether the video can be successfully uploaded.
3. Whether the ingredients and steps and recipe name and serve amount can be successfully modified.
4. Error check of user input correctly works.
5. After confirmation, whether the change is committed to database, and the page successfully updated.

**Scenario:** The user finds that there are too many recipes, user wants to use search functionality to search specific recipe.

**Usability records;**

1. Whether the search function works properly.

2. Whether the result is correctly shown.

3. If user click for more information, whether he can get the correct recipe.

**Scenario:** The user no longer wants this recipe, he wants to delete this recipe from the cookbook.

**Usability records:**

1. If the deletion in the database is correct.
2. Whether the deleted recipe is removed from the View.

# **Evaluation/results**

**Main qualities to be evaluated**

1. **Usability**

Most users can easily finish all the tasks, first time they use the software, they may find it weird not to have search button, but after a while they fully understand it, and start to feel comfortable about the design. But it’s difficult to realize where to manage (delete, edit) the message. And one major drawback is that user cannot control the play or pause of the video, this is the future improvement of our software.

1. **Repeatability**

During the second time our test taker manipulate our software, they can all manipulate our software proficiently, there are less instructions given during the test.

1. **Errors**

For error part, 3 people forget to insert a ingredient to the recipe, but after the reminder, they realize their mistakes and correct it. For some experienced IT students, we ask them to deliberately insert false data to test the error detection part of code. It turns out the error detection works pretty well, and when the alert pop up, most people can realize their mistakes.

**Bugs during test**

However, there are still some bugs during the test.

1. **Problem 1 (Solved):** When people are editing their recipes, if they don’t want to edit this recipe and click the back button, the changes they make will be preserved, although the change is not committed to the database.

**Solution:** We realize that every time the user exits a page, we still have to refresh the content of previous page, so we add the code to refresh on exiting page.

1. **Problem 2 (Solved):** During the view recipe period, there is a specific way that user can enter a negative number and somehow avoid triggering the error detection part.

**Solution:** We locate where this bug come from, and add in our error detection part the corresponding code

1. **Problem 3 (Unsure):** When user view recipe, the video will not be shown each time, and sometimes it takes a long time to open the view scene.

**Solution:** We checked our code, the lagging issue is caused by the improper video URL, and this is because during the test, we insert some illegal URL into the database and forget to delete it, after we correct the database, this issue is solved.

As for the video playing part, we checked the code of video playing, our result is that the video do starts playing, but the video just didn’t show properly, we don’t know where this issue come from, we think this is the problem of media player. But this is a random issue, and if user exit and reenter, the media can play again.

1. **Problem 4 (Solved):** In the modify recipe page, if the recipe doesn’t have a image selected, the default image may not be correctly displayed.

**Solution:** This is caused by the URL of imageview, it will return a URL in form like “file:path/to/file”, so the prefix “file:” is added to database which should not be added, our solution is to check the URL every time and then add it to database to avoid illegal URL.

1. **Problem 5 (Solved):** During the test part, about 4 users point out that the software is much easier to use if we put the switching tab on the top of our window.

**Solution:** We modified our code, and put all the tabs on the top of the window.

1. **Problem 6 (Solved):** When adding ingredients, during both modify recipe or create recipe part, if we add two or more ingredients with the same name and same unit, the database will not update and will throw an exception.

**Solution:** We discussed within the group that two or more ingredients with the same name and unit should not appear in the ingredients part, so we show an alarm to avoid this. And also in addition, we think it reasonable to have ingredients with the same name but their units are different.

1. **Problem 7 (Solved):** When return from modify page to manage page, we notice that we need to click change page button twice to change page, the first click will not work.

**Solution:** After analyzing the code, we find out that we forget to reset the current page of manage page, so this means user have to “sync” page manually with button click, so we reset current page each time we open manage page.

# **5.** **Evaluation**

# **Group work**

# **Task responsibilities**