SDD-01

SOFTWARE DEVELOPMENT DOCUMENT

MyRecipe

Final Project For:
CIJ3A3: SE: Design and Analysis Implementation

Eko Darwiyanto S.T., M T.

Prepared By:

Nur Afina Rahmani	1301202563
Sofyan Rinaldi	1301203311
Muhammad Fadli Ramadhan	1301203533
Shinta Dewi Lestari S.	1301203567

Bachelor of Informatics Study Program – School of Computing
Telkom University

Telkom University	**Bachelor of Informatics Telkom University**	**No**		**Number of Pages**
		SDD-0001		
		Revision	*1*	*Date: 10 October 2022*

REVISION LIST

Revision	Description
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Written by								
Checked by								
Agreed by								

List of Change

Page	Revision	Page	Revision

Table of Contents

1. Introduction

1.1. Purpose

The purpose of making this Software Design Document is to fulfil the big task of the course RPL: Design and Implementation and to document all activities or activities carried out during the development of the MyRecipe website project starting from the user requirements stage, analysis and design, implementation to testing. Besides that, the writing of this document will be used as a reference in implementation. The purpose of this MyRecipe software is that Users can see recipes that the user wants with a simple click.

1.2. Project Scope

MyRecipe is a web application that provides recipes that are made by chefs from around the world. Where users can access and see this recipe online and rate it by giving a like when the recipe fits their taste.

1.3. Definitions and Terms

Table 1.1 lists the definitions, abbreviations, and acronyms that are used in this document:

Keywords or phrases	Definitions and/or acronyms
SDD	The document to describe the detail about the plan to develop the software.
SDP	*Software Development Plan*
SRS	*Software Requirement Specification*
SE	Software Engineering Software development activity
IEEE	*Institute of Electrical and Electronics Engineers* International standard to develop and plan the food recipe
ANSI	*American Standard Institute*

1.4. Naming and Numbering Rules

In this document there is no specific numbering and naming rules.

1.5. References

Document	Date	Title
IEEE Std 830-1998	25 June 1998	IEEE Recommended Practice for Software Requirements Specifications

1.6. Document Overview

This SDD contains the description of the website application based on what was determined on the SRS document. This SDD will explain the details of the software so that the software can be implemented. This document consists of four chapters as it follows.

A. Introduction

Introduction contains the explanation about SDD documents which embrace the purpose of this document, the scope of the development of this software, definition, references, and document overview.

B. Description of Global Design

Description of global design contains about the plan of the software which will be developed involves architectural description, and component description.

C. Detail Design

Detail Design on this document contains about the realisation of the use case, the plan of the class details, the description of class diagram, algorithm/query, user interface, and the plan of the class representation.

2. Description of The Global Planning

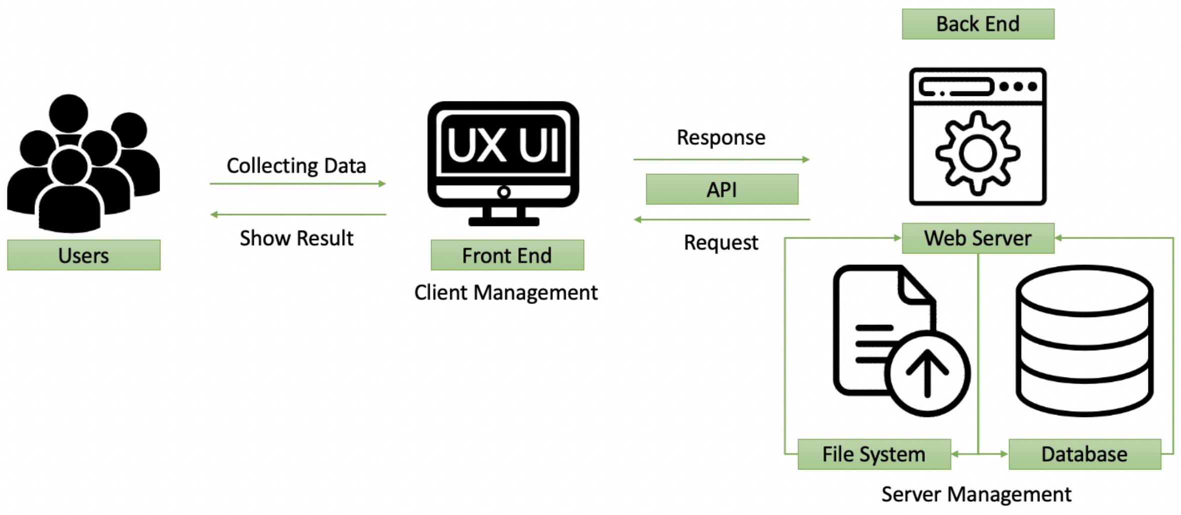
The description of the global planning involves the description of the Implementation of Environment Plan, Architectural Description, and Component Description.

2.1. Implementation of Environment Plan

This system is will be implemented as in:

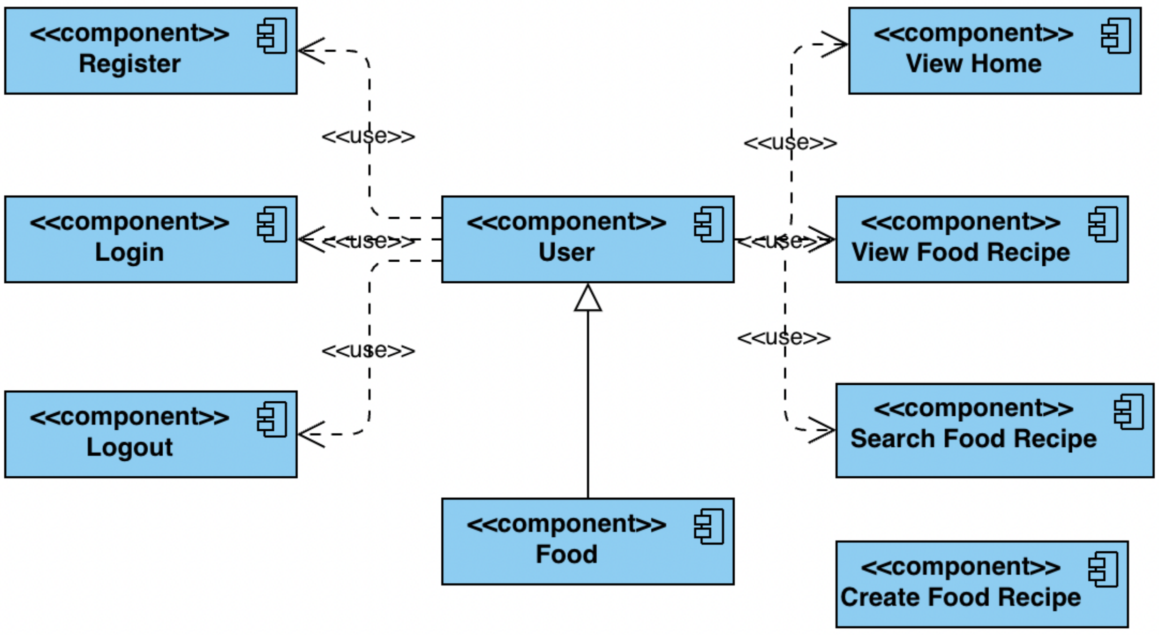
1. Operation system:
2. Programming Language: Java
3. DBMS: MySQL
4. Development Tool: PHPmyAdmin (XAMPP), Netbeans

2.2. Architectural Description



Above is an architecture or component that will be implemented on our software “MyRecipe” to simplify the development process.

2.3. Component Description

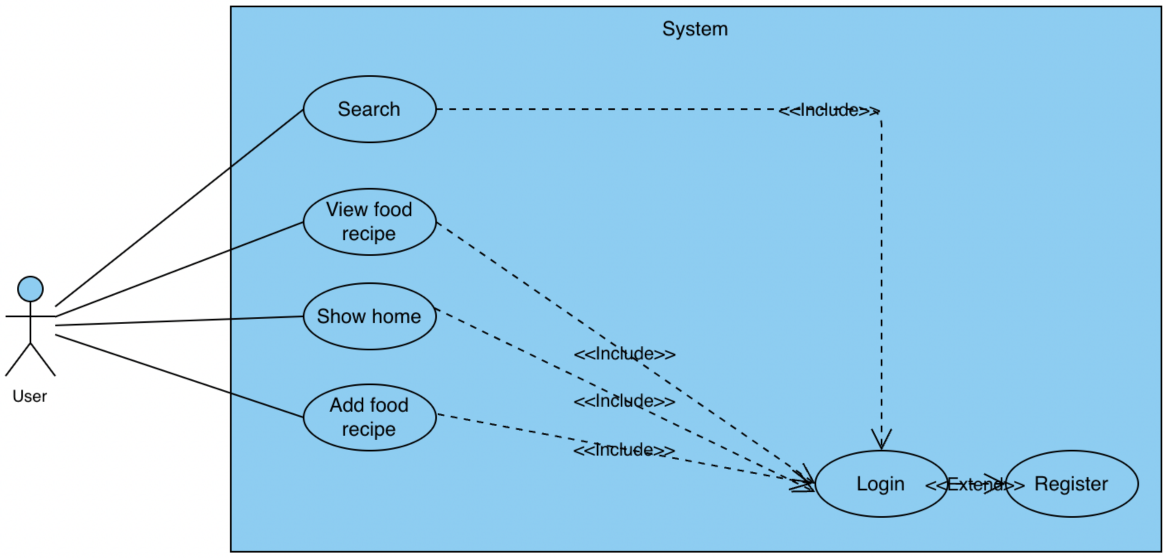


No	Component Name	Description
1	User	User on MyRecipe
2	Food	Component of the food that we want to show
3	Register	Menu to register the account
4	Login	Menu to log the account in
5	Logout	Menu to log the account out
6	View Home	Menu to show home
7	View Food Recipe	Menu to show food description
8	Search Food Recipe	Menu to search food recipe
9	Add Food Recipe	Food Recipe will be added into the list

3. Detailed Planning

3.1. Use Case Realisation

This part is to explain the realisation of all the use case that have been developed on SRS Document.



No	Use Case	Description of Use Case
1	Register	This feature is used by all the actors and it is working to register a new account.
2	Login	User inputs username and password to log in the account and to go to the home page.
3	Search food recipe	Users can search for the item that they want.
4	View food recipe	This feature is for the user. This feature is to look at the food recipe information.
5	Add food recipe	User add food recipe
6	Show home	This feature is to take the user to the main page.

7	Logout	Users log their account out.

3.1.1.Register Use Case

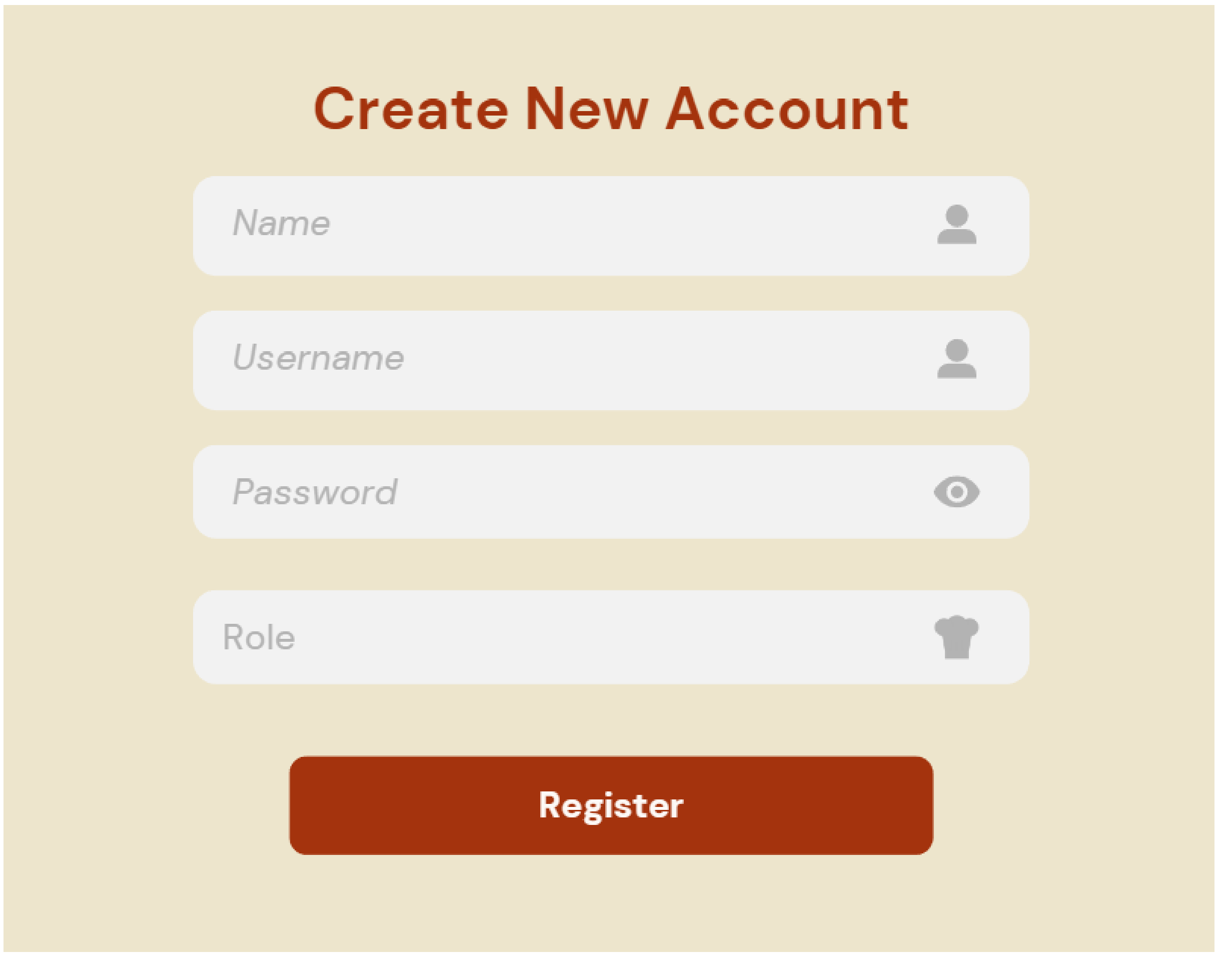
Use Case	Register
Description	This feature is used by all the actors and it is working to register a new account.
Pre-condition	This user registers the data by filling the data that is requested by the system.
Post-condition	Users successfully make an account to log in.

Flow of Event Table

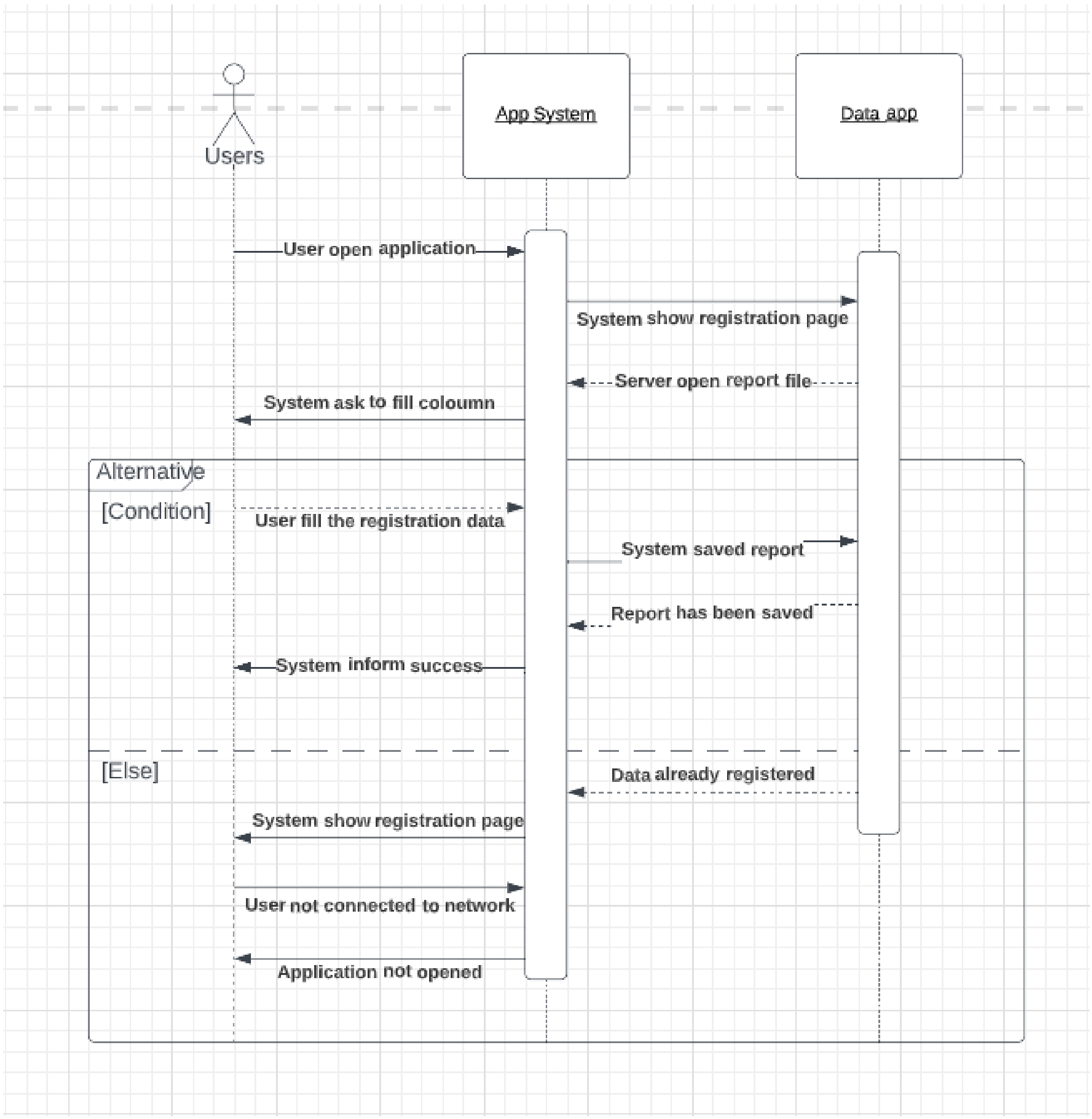
Register	
Actor Actions	System Actions
1. User open application	
	2. System shows a registration page
3. User fills name, username, role, and password and confirmation password.	
4. User pressed the register button	
	5. System records the registration data
	6. System processes and save the data registration to the database
	7. System shows the login page and pop-up “Account successfully made”
Alternate Flow	
	5a. System checks data. If the username has been registered, go back to step 2.

Exceptional Flow of Event
If the user's device is not connected to the internet connection, all actions from step 1 are rejected.

3.1.1.1 User Interface



3.1.1.2 Sequence Diagram



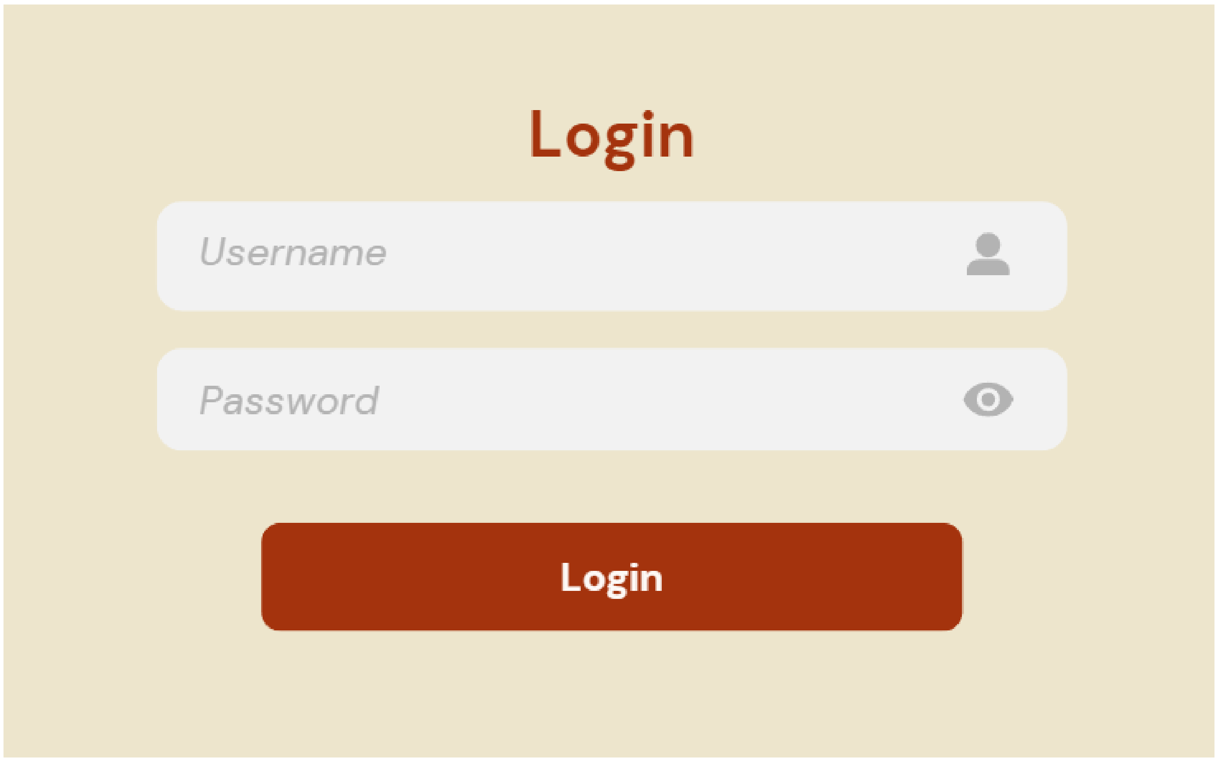
3.1.2. Login Use Case

Use Case	Log In
Description	User inputs username and password to log in the account and to go to the home page.
Pre-condition	User inputs username and password.
Post-condition	User has successfully logged in to the account and go to the home page.

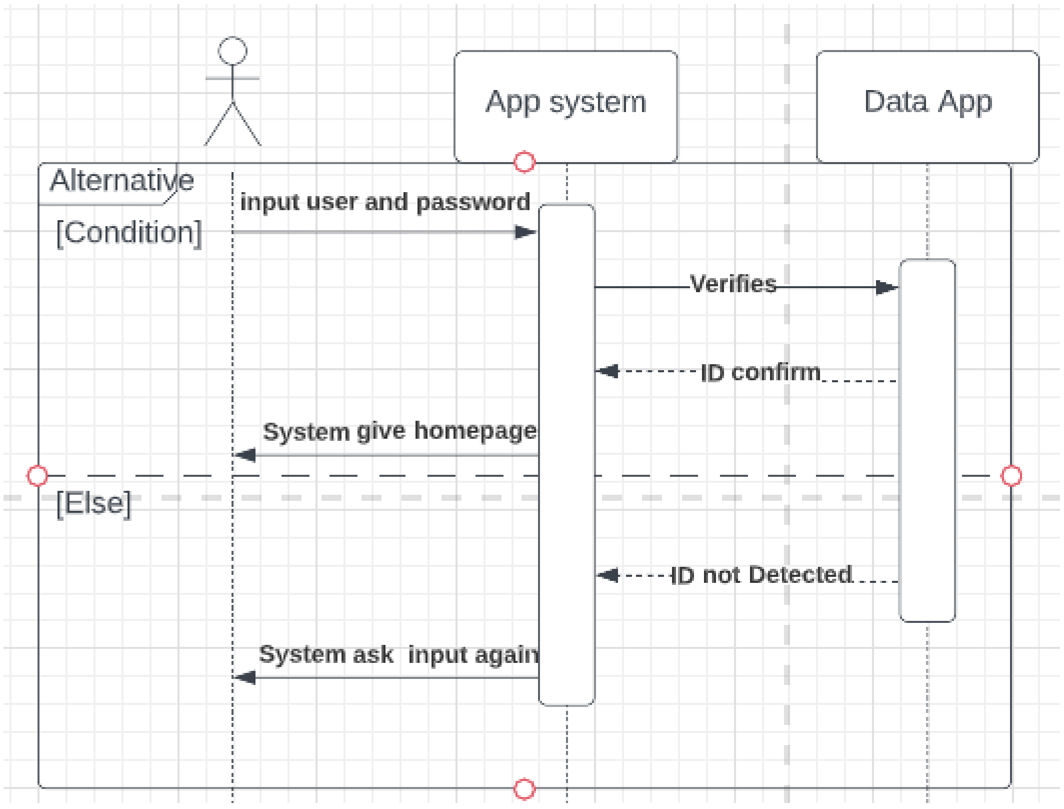
Flow of Event Table

Log In	
Actor Actions	System Actions
1. User opens the application	
	2. System shows login page
3. User fills the username and account password.	
4. User clicks the "Login" button.	
	5. System shows the login page and pop-up "Successfully login"
Alternate Flow	
	3a. If the data account that is input is wrong, repeat step 3
Exceptional Flow of Event	
If the user's device is not connected to the internet connection, all actions from step 1 are rejected.	

3.1.2.1 User Interface



3.1.2.2 Sequence Diagram



3.1.3.Search Food Recipe Use Case

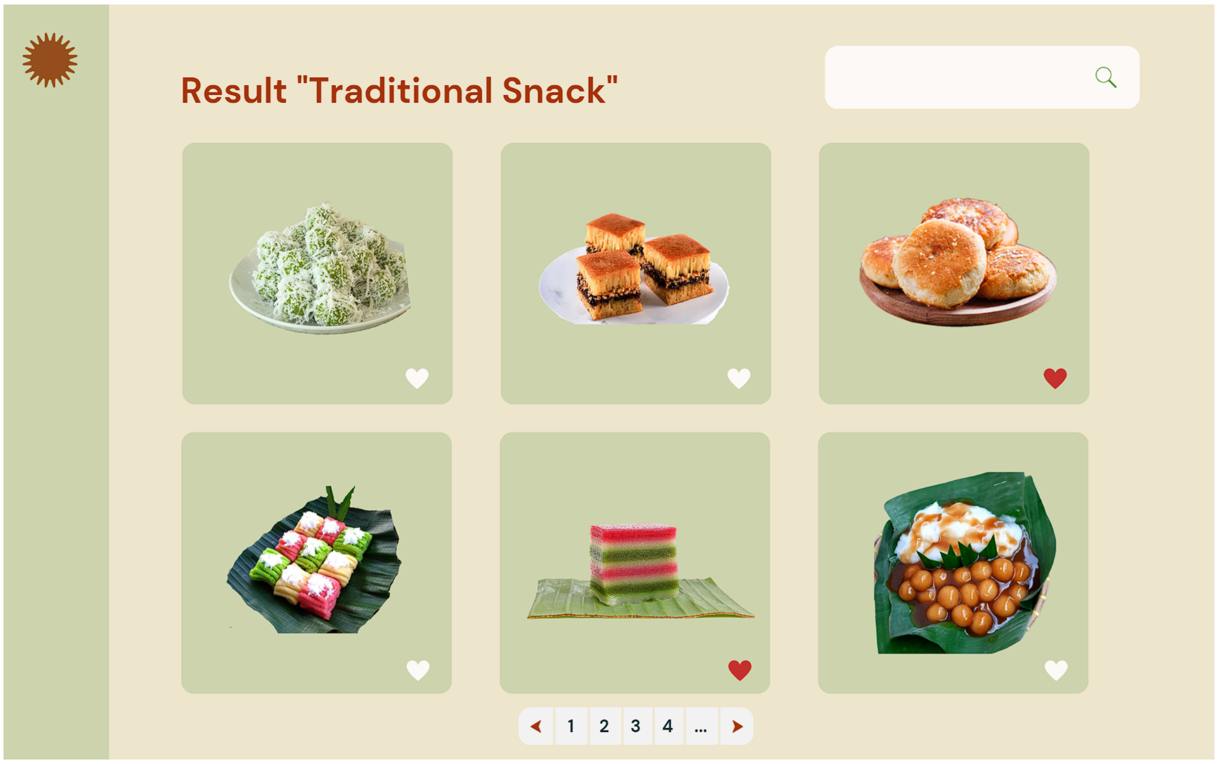
Use Case	Search
Description	This feature is for the user and admin. This feature is to search the food recipe that they wanted to search.
Pre-condition	The system successfully synchronised the food recipe that user or food recipe admin's searched for and connected to the MyRecipe' network.
Post-condition	System shall show the search result of the food recipe.

Flow of Event Table

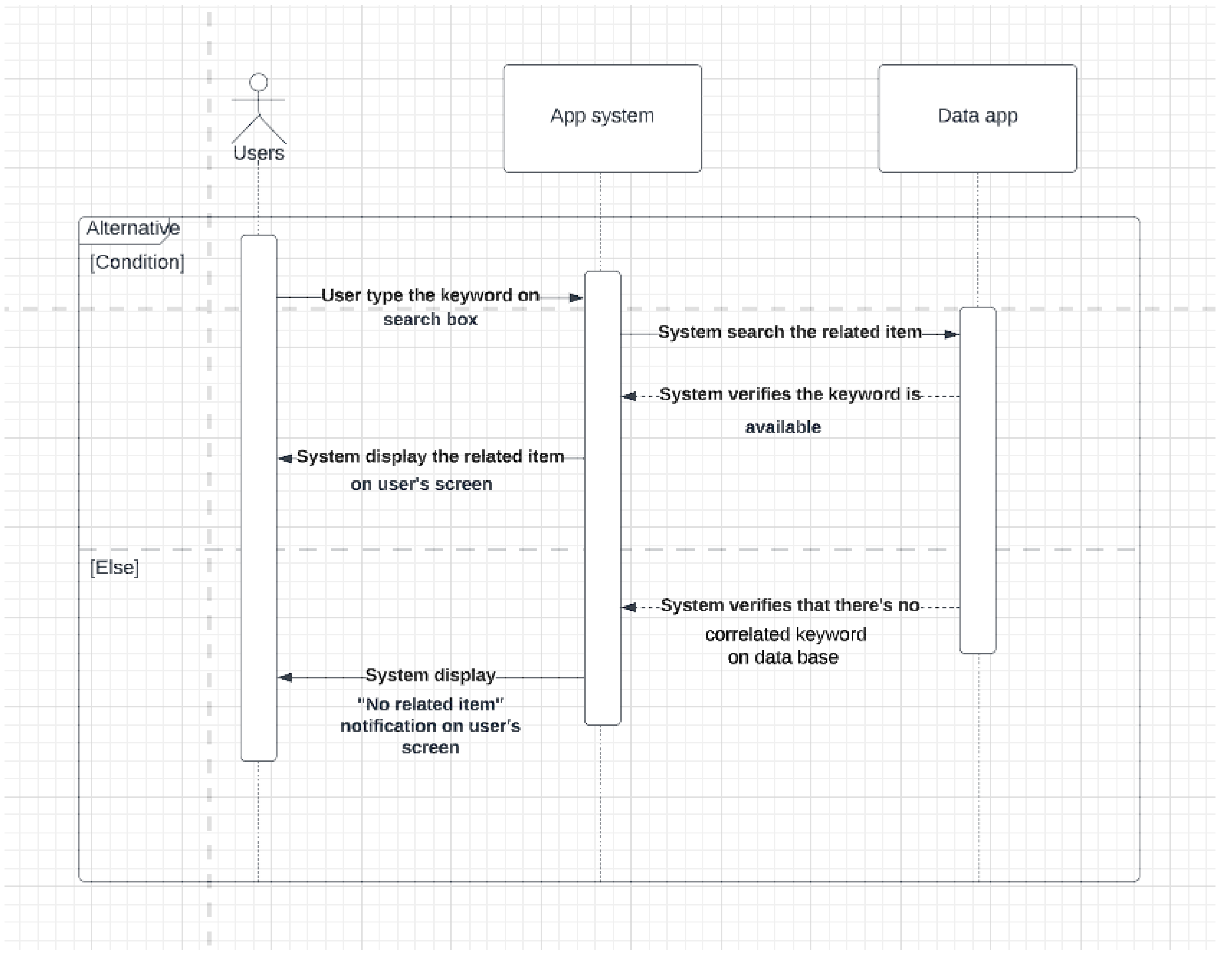
Search food recipe	
Actor Actions	System Actions

1. User input keywords into the search box	2. System search for the related items according to the keywords.
	3. System verifies that it is true there is a key word in the system database (item).
	4. System outputs the related item to the user's screen device.
Alternate Flow	
	3a. System verifies there is no correlated key word in the system database.
	4a. System outputs "No item is here" to the user's screen device.

3.1.4.1 User Interface



3.1.4.2 Sequence Diagram


Users
App system
Data app
Alternative
[Condition]
User type the keyword on search box
System search the related item
System verifies the keyword is available
System display the related item on user's screen
[Else]
System verifies that there's no correlated keyword on data base
System display "No related item" notification on user's screen

3.1.4. View Food Recipe Use Case

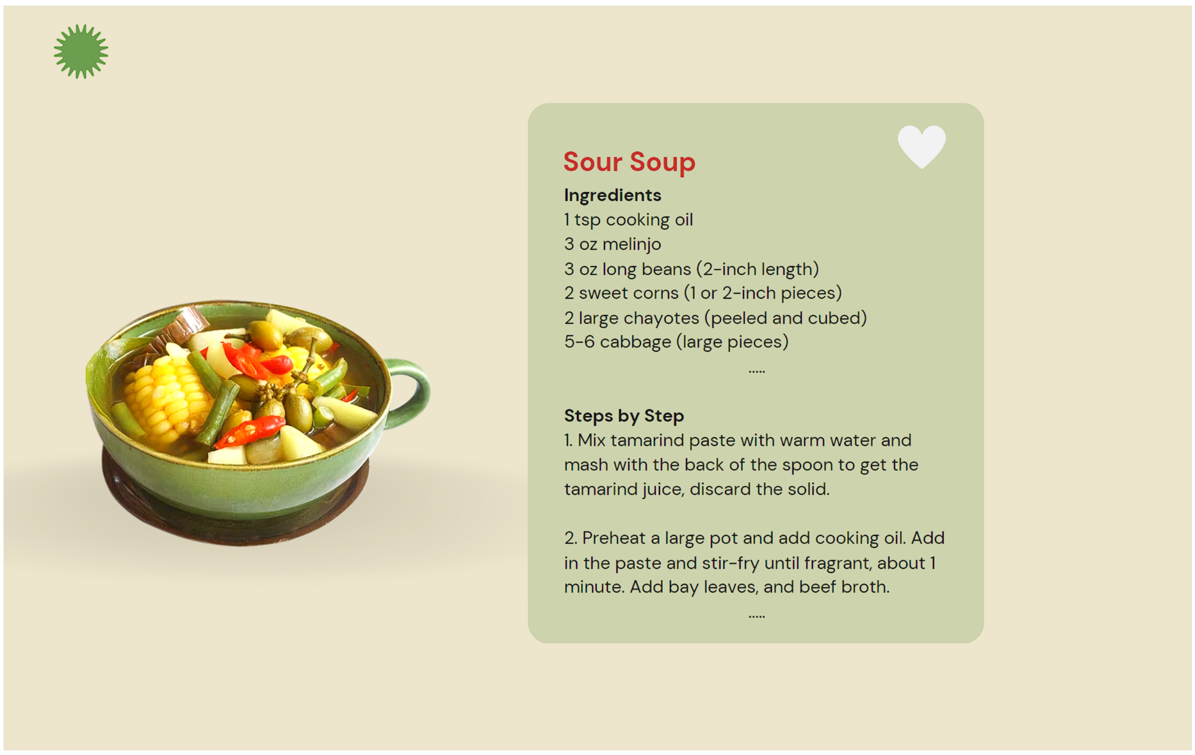
Use Case	View food recipe
Description	This feature is for the user, food recipe admin, and admin. This feature is to look at the food recipe information.
Pre-condition	The user clicked the food recipe they wanted to look at and connected to the MyRecipe' network.
Post-condition	The system shall show the food recipe view page.

Flow of Event Table

View Food Recipe	
Actor Actions	System Actions

1. Users click on the recipe intended to see.	2. System directs the user to the food recipe's page.
	3. System outputs the food recipe's page to the user's screen device.
4. User clicks the back button	5. The system brings back the user to the previous page.
	6. The system outputs the previous page to the user's screen device.
Exceptional Flow of Event	
If the user's device is disconnected from the internet, actions from step 1 are rejected.	

3.1.5.1 User Interface

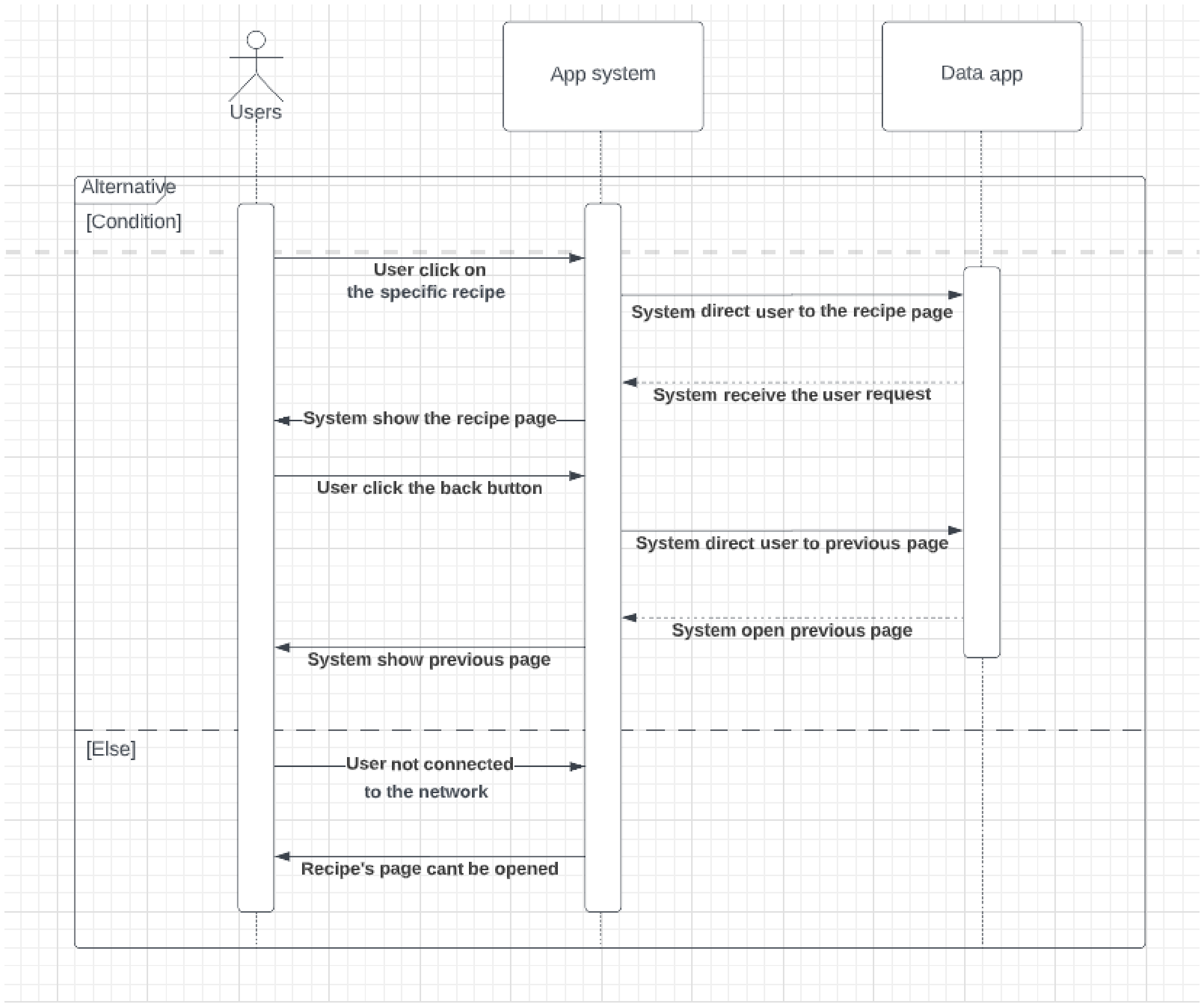

Sour Soup

Ingredients
1 tsp cooking oil
3 oz melinjo
3 oz long beans (2-inch length)
2 sweet corns (1 or 2-inch pieces)
2 large chayotes (peeled and cubed)
5-6 cabbage (large pieces)
.....

Steps by Step
1. Mix tamarind paste with warm water and mash with the back of the spoon to get the tamarind juice, discard the solid.

2. Preheat a large pot and add cooking oil. Add in the paste and stir-fry until fragrant, about 1 minute. Add bay leaves, and beef broth.
.....

3.1.5.2 Sequence Diagram


Users
App system
Data app
Alternative
[Condition]
User click on the specific recipe
System direct user to the recipe page
System receive the user request
System show the recipe page
User click the back button
System direct user to previous page
System open previous page
System show previous page
[Else]
User not connected to the network
Recipe's page cant be opened

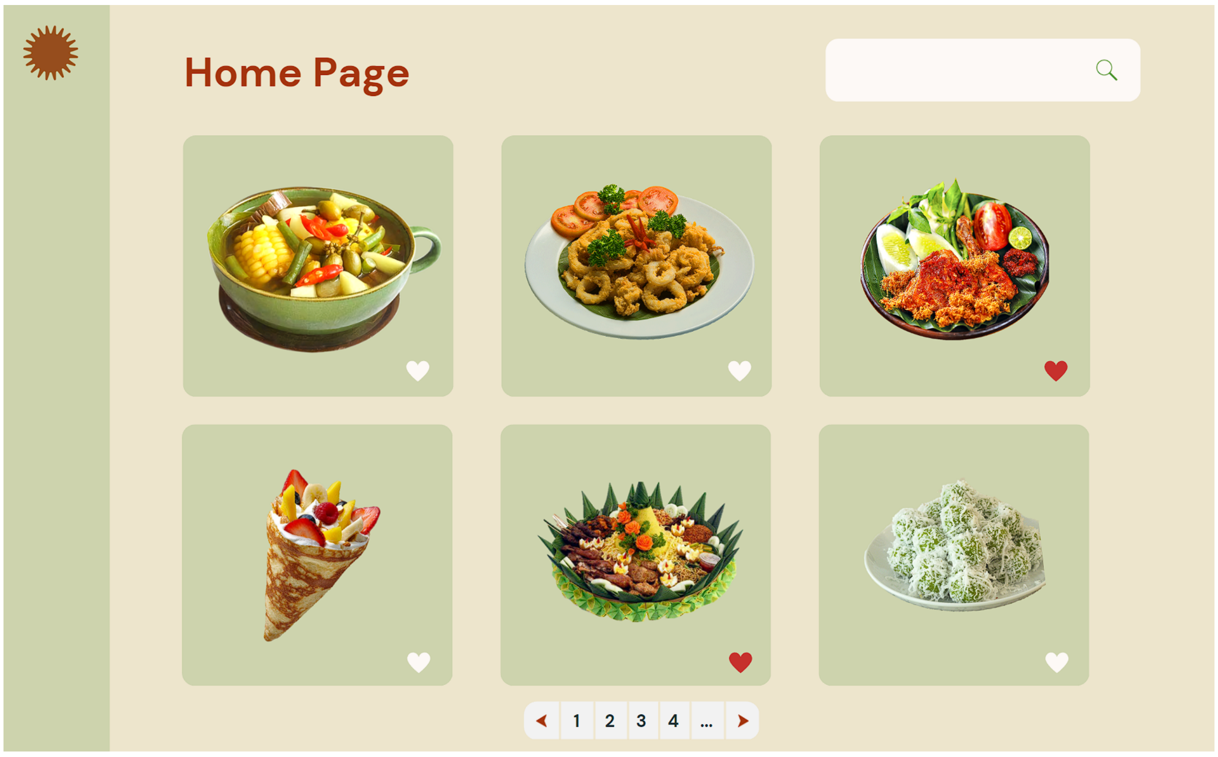
3.1.5. Show Home Use Case

Use Case	Show Home
Description	This feature is to take the user to the main page.
Pre-condition	The user clicked the home button they connected to the MyRecipe' network.
Post-condition	The user is looking at the main page.

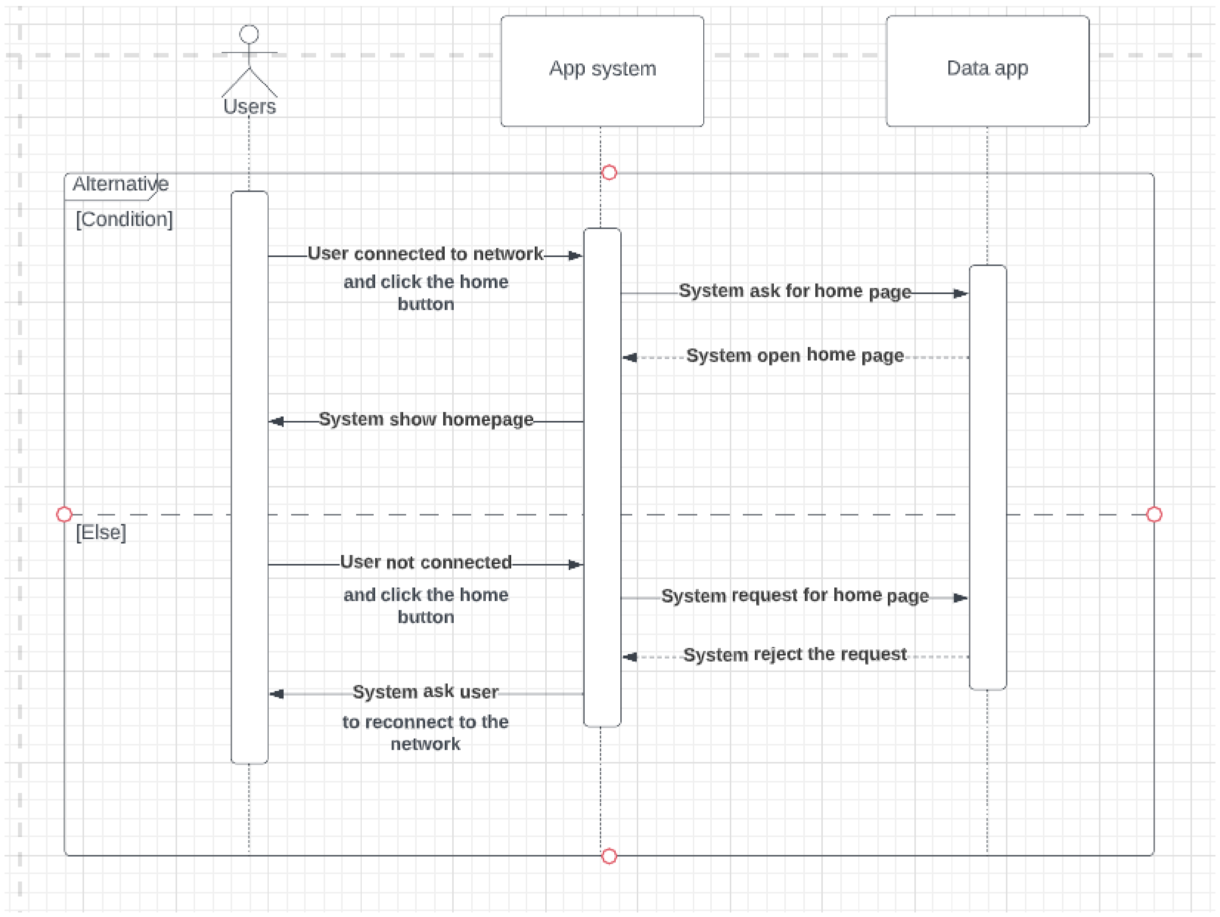
Flow of Event Table

Show Home	
Actor Actions	System Actions
1. The user is connected to the MyRecipe network and clicked the home button	2. System directs the user to the main page.
	3. System shows the main page.
Exceptional Flow of Event	
If the user's device is disconnected from the internet, action after step 1 will be rejected and the user will be given the notification to redo from step 1.	

3.1.9.1 User Interface



3.1.9.2 Sequence Diagram



Users
App system
Data app
Alternative
[Condition]
User connected to network and click the home button
System ask for home page
System open home page
System show homepage
[Else]
User not connected and click the home button
System request for home page
System reject the request
System ask user to reconnect to the network

3.1.6. Add Food Recipe

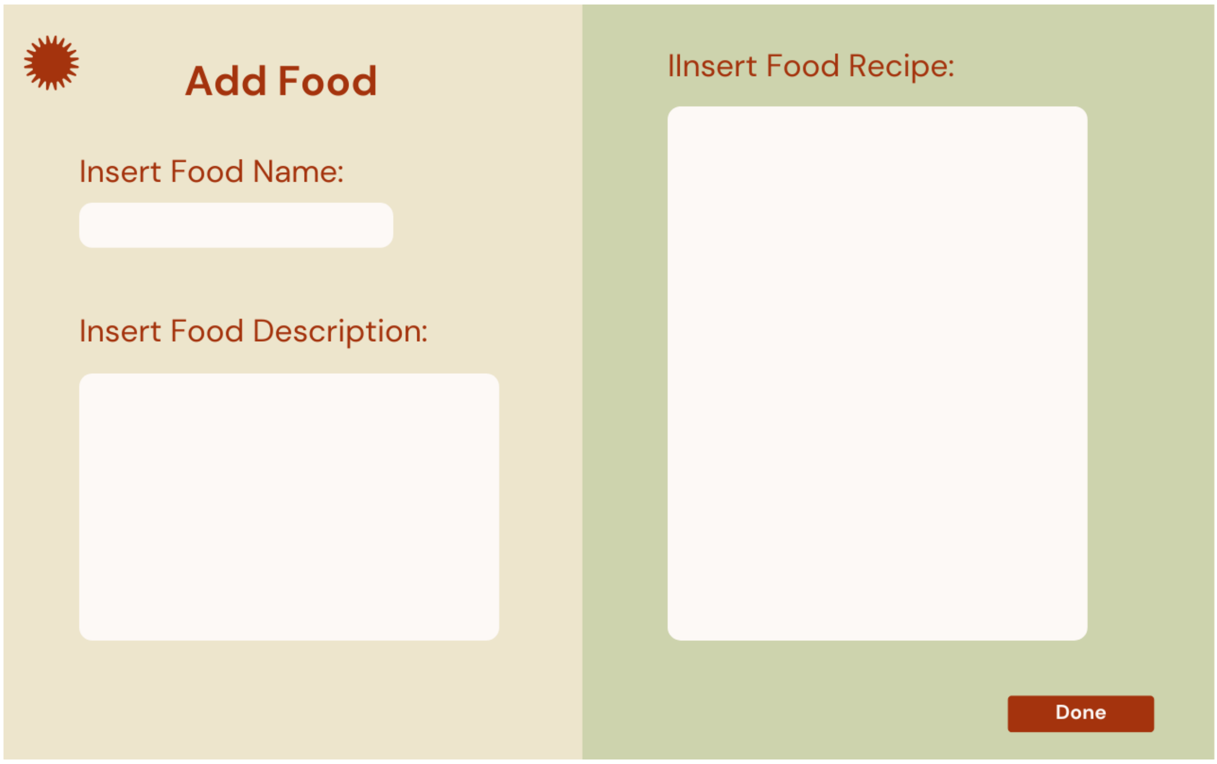
Use Case	Add Food Recipe
Description	User add food recipe.
Pre-condition	The user clicked the add button they connected to the MyRecipe' network.
Post-condition	The user is adding the food recipe.

Flow of Event Table

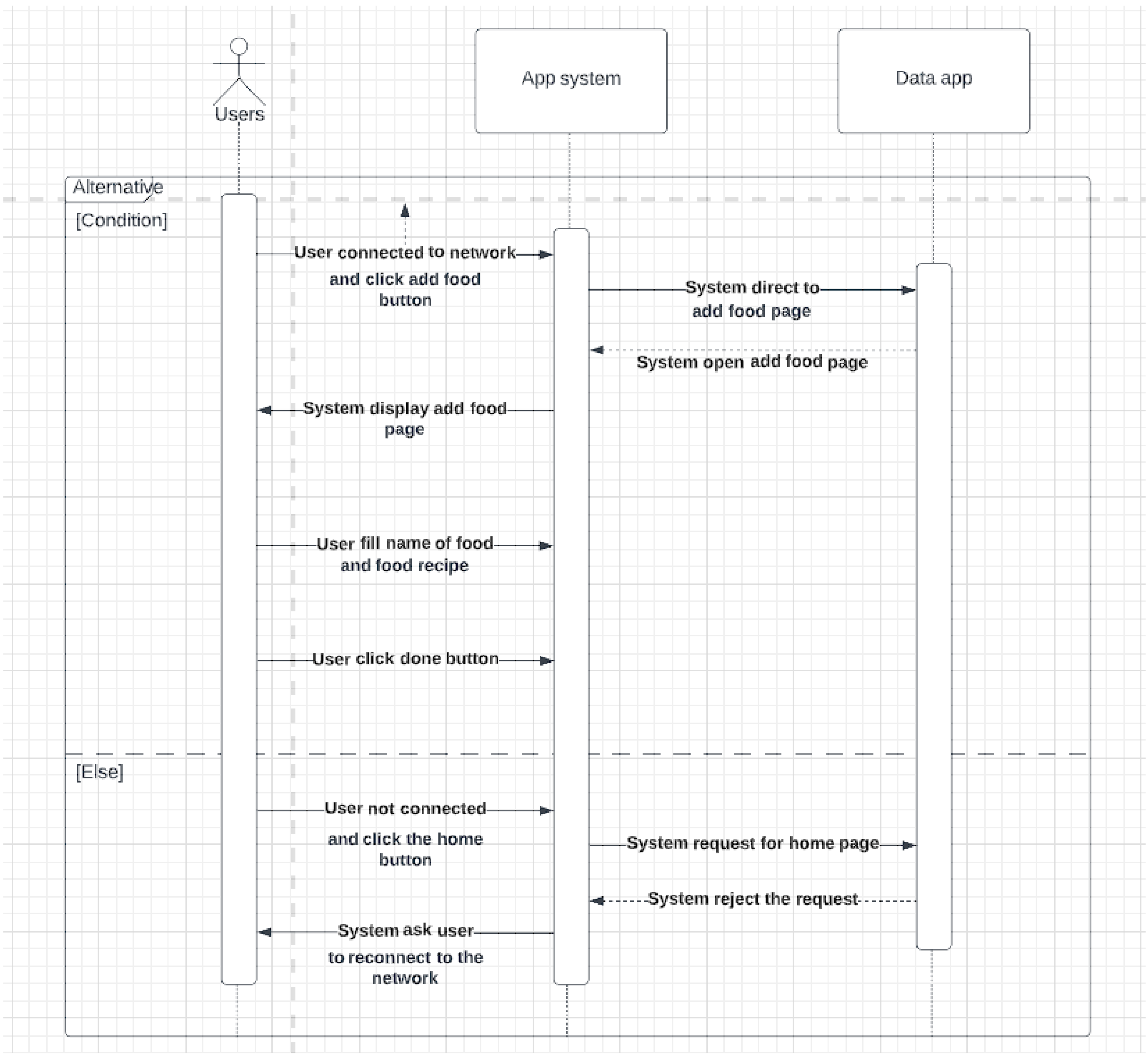
Add Food	
Actor Actions	System Actions
1. The user is connected to the MyRecipe network. The user is already in the home page and clicked the add button	2. System directs the user to the add food recipe page.

3. User fills the name of the food, the description of the food, and the recipe of the food.	
4. User clicks the done button.	5. System successfully saves the new food recipe.
Exceptional Flow of Event	
If the user's device is disconnected from the internet, action after step 1 will be rejected and the user will be given the notification to redo from step 1.	

3.1.9.1 User Interface



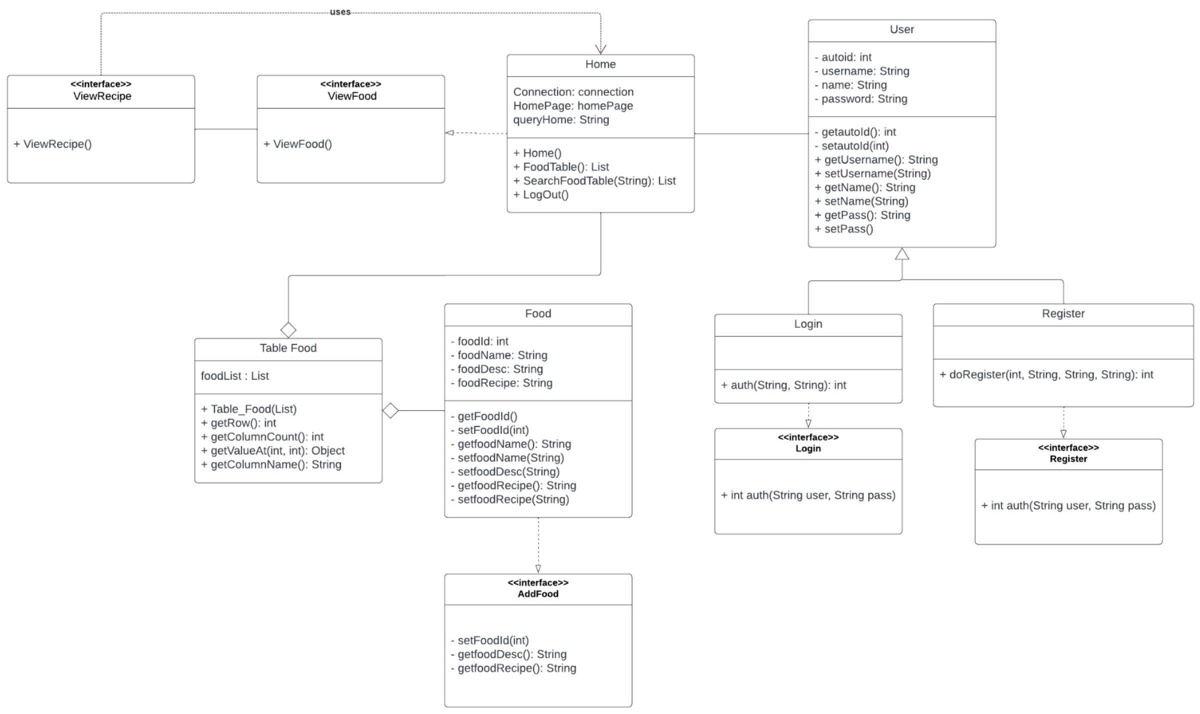
3.1.9.2 Sequence Diagram


Users
App system
Data app
Alternative
[Condition]
User connected to network and click add food button
System direct to add food page
System open add food page
System display add food page
User fill name of food and food recipe
User click done button
[Else]
User not connected and click the home button
System request for home page
System reject the request
System ask user to reconnect to the network

3.2. Class identification

No	*Nama Kelas Perancangan*	*Nama Kelas Analisis Terkait*

3.3. Class Diagram



3.4. Algorithm/Query

3.4.1. Algorithm #1

Class name: AuthController

Operation name: registration

Algorithm:

```
public int Regist() {
        String username, name, pass, conPass;
        int result = 0;
        username = frame.getjUsername().getText(); // get text username dari jtextfield yang ada
di registerform
        name = frame.getjName().getText();
        pass = frame.getjPassword().getText();
        conPass = frame.getjConPassword().getText();
          if (!frame.getjUsername().getText().isEmpty() & !frame.getjName().getText().isEmpty()
& !frame.getjPassword().getText().isEmpty() &
!frame.getjConPassword().getText().isEmpty()) { // kondisi ideal
            User user = new User();
            implRegister.doRegister(autoid, username, name, pass);
            JOptionPane.showMessageDialog(null, "Create Account Successfully");
            result = 1;
        } else if (!pass.equals(conPass)) {
            JOptionPane.showMessageDialog(null, "Password does not match!");
            result = 0;
            } else if (pass.equals("") || username.equals("") || username.equals("") ||
conPass.equals("")) {
            JOptionPane.showMessageDialog(null, "Login failed. Please fill all the blank
forms.");
        result = 0;
    }
    return result;
  }
```

3.4.2. Algorithm #2

Class name: AuthController

Operation name: login

Algorithm:

```
public int LogIn() {
    int resultAuth = 0;
    int result = 0;
                                    if    (!frame.getTfUsername().getText().isEmpty()    &
!frame.getjPasswordField().getText().isEmpty()) {
        User user = new User();
        String username = frame.getTfUsername().getText();
        String password = frame.getjPasswordField().getText();
        resultAuth = implLogin.auth(username, password);
        if (resultAuth == 1) {
          result = resultAuth;
        } else {
          result = 0;
        }
        if (result == 1) {
          JOptionPane.showMessageDialog(null, "Successfully Logged in");

        } else if (result == 0) {
          JOptionPane.showMessageDialog(null, "Login failed: Cannot find your account.");
        }

    } else {
         JOptionPane.showMessageDialog(null, "Login failed. Username and password can
not be empty.");
    }
    return result;

  }
```

3.4.3. Algorithm #3

Class name: ContentController

Operation name: viewFoodRecipe

Algorithm:

```
view.setVisible(true);

    String foodName;
    foodName = (String) jfoodName.getText();

    String foodRec = null;
    try {
      // TODO add your handling code here:
    Statement st = connection.createStatement();
      ResultSet rs = st.executeQuery("SELECT foodRecipe FROM food WHERE foodName
LIKE '%"+foodName+"%'");
    if (rs.next()){
      foodRec = rs.getString(1);

    }else {
      foodRec = "failed";
    }

    } catch (SQLException ex) {
      Logger.getLogger(ViewFood.class.getName()).log(Level.SEVERE, null, ex);
    }
    view.setVisible(true);
    view.pack();
    view.setLocationRelativeTo(null);
    view.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    view.jfoodName.setText(foodName);
    view.jviewRecipe.setText(foodRec);
```

3.4.4. Algorithm #4

Class name: ContentController

Operation name: showHome

Algorithm:

```
public class ControllerHome {
   private Food food;
   HomePage homepage;
   Homeable implHome;
   List<Food> foodList;
   public ControllerHome(HomePage hp) {
      this.food = food;
      this.homepage = hp;
      this.implHome = new Home();
      this.foodList = implHome.FoodTable();
   }

   public void showFoodTable() {
      foodList = implHome.FoodTable();

      Tabel_Food tFood = new Tabel_Food(foodList);
      homepage.getTableFood().setModel(tFood);

homepage.getTableFood().getTableHeader().getColumnModel().getColumn(0).setHeaderValu
e("Food");
   }
}
```

3.4.5. Algorithm #5

Class name: ContentController

Operation name: search

Algorithm:

```
public void showSearchFood() {
      String foodText;
      foodText = homepage.getTfSearch().getText();
      if (!foodText.isEmpty()) {
         foodList = implHome.SearchFoodTable(foodText);
         Tabel_Food tFood = new Tabel_Food(foodList);
         homepage.getTableFood().setModel(tFood);

homepage.getTableFood().getTableHeader().getColumnModel().getColumn(0).setHeaderValu
e("Food");
      }
```

```
    else {
      showFoodTable();
    }
  }
```

3.4.6. Algorithm #6

Class name: ContentController

Operation name: add

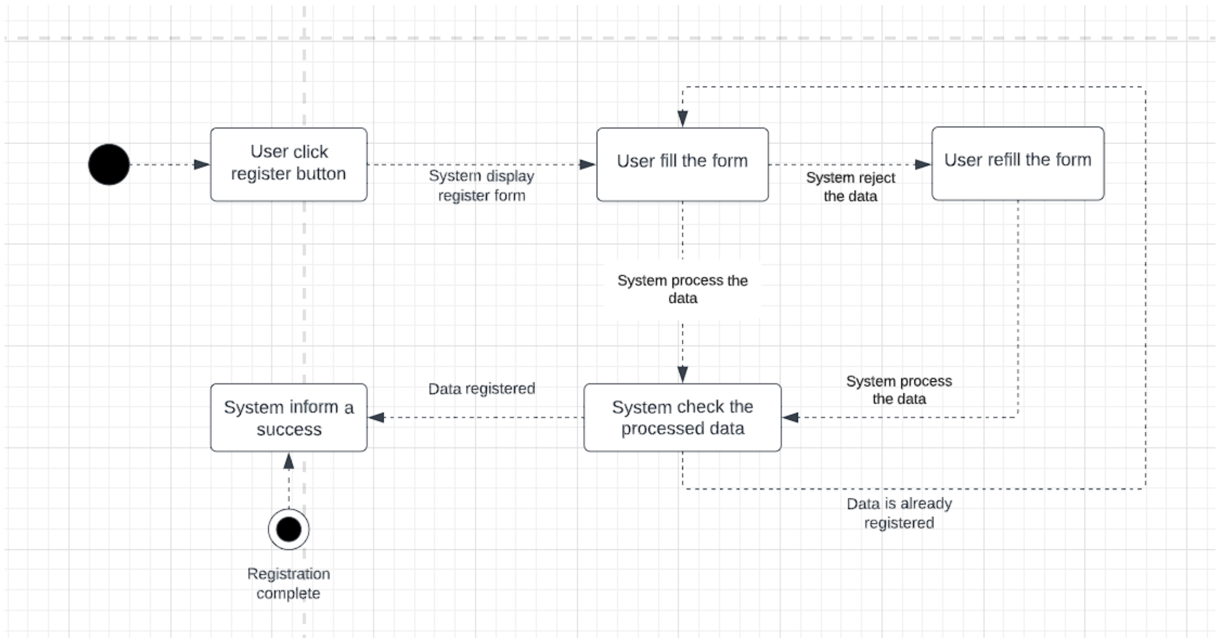
Algorithm:

```
String foodName, foodDesc, foodRec;



        //final String sql = "INSERT INTO food (foodName,foodDesc,foodRecipe) VALUES
(?,?,?)";
    foodName = getjfoodName().getText();
    foodDesc = getjfoodDesc().getText();
    foodRec = getjfoodRec().getText();

    try {
      st = connection.createStatement();
            String sql = "INSERT INTO food (foodName,foodDesc,foodRecipe) VALUES
('"+foodName+"','"+foodDesc+"','"+foodRec+"')";
      st.executeUpdate(sql);
      st.close();
      JOptionPane.showMessageDialog(null,"Fill succed");
    } catch (SQLException ex) {
      Logger.getLogger(AddFood.class.getName()).log(Level.SEVERE, null, ex);

    }

    pack();
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setVisible(false);
    HomePage hp = new HomePage();
    hp.setVisible(true);
```

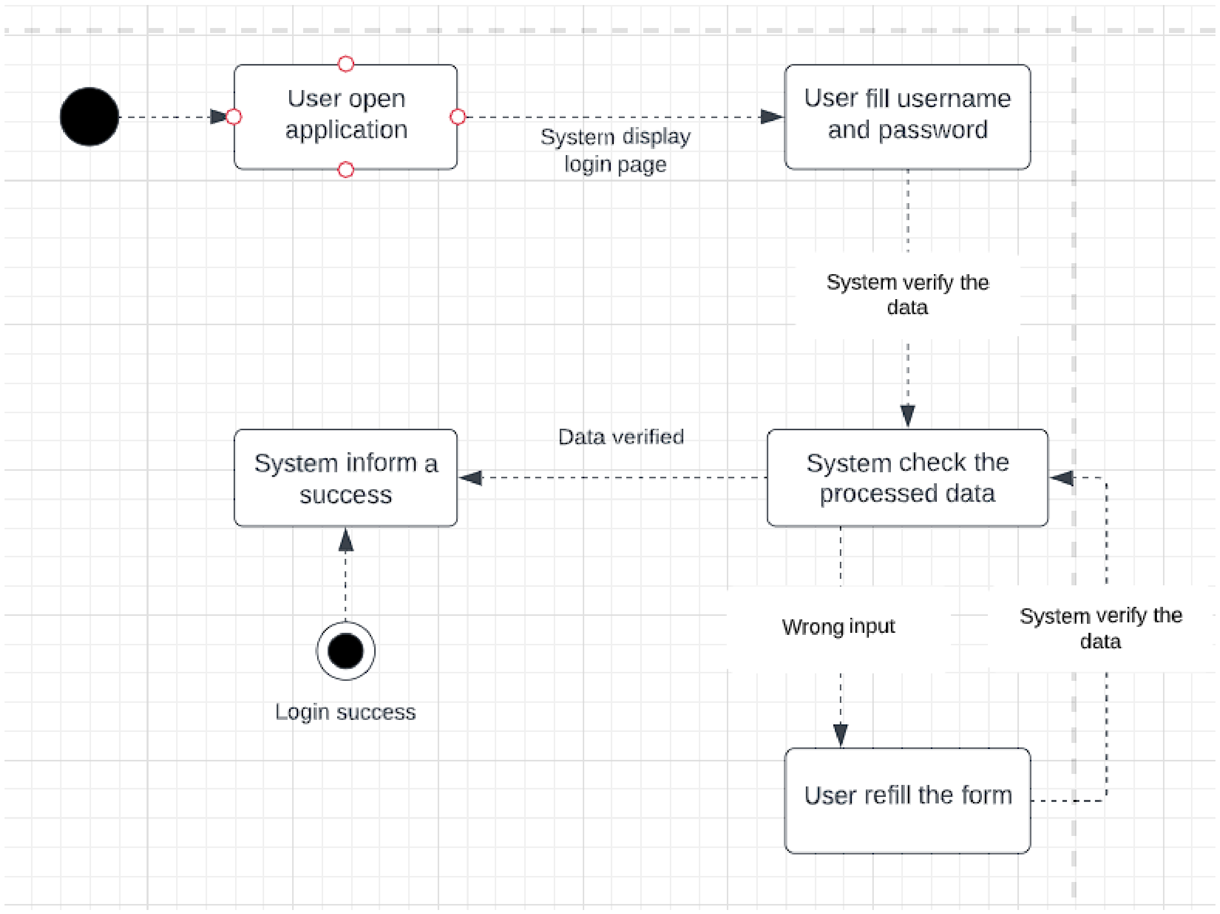
3.5. Diagram Statechart

Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subba per kelas.

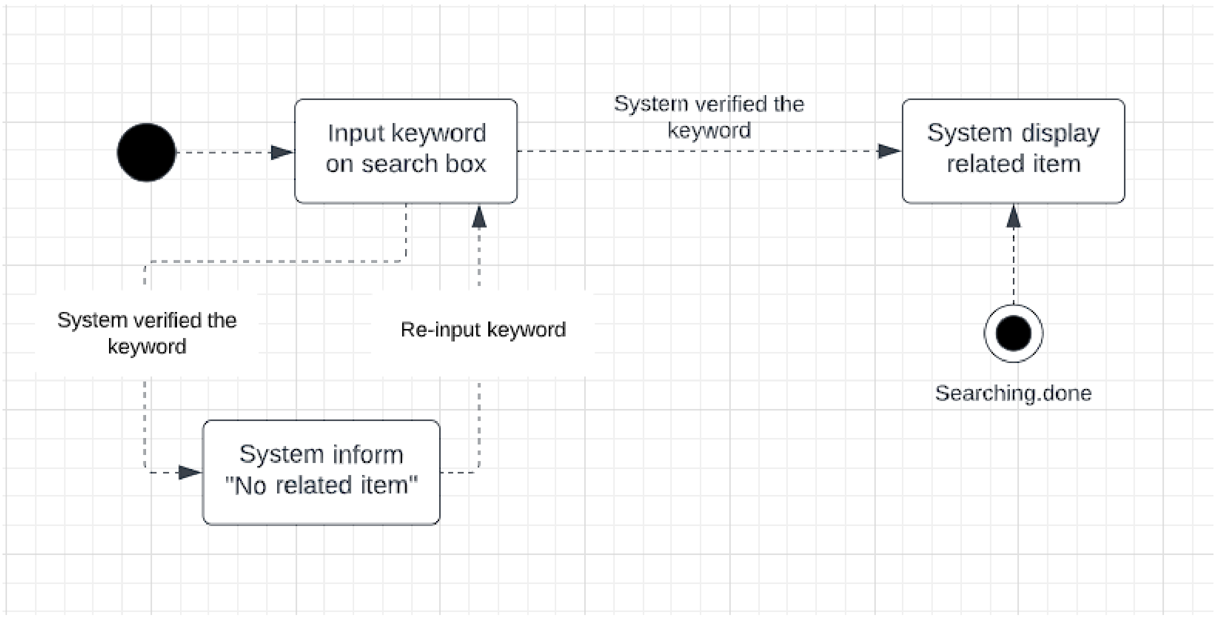
3.5.1 Register Statechart



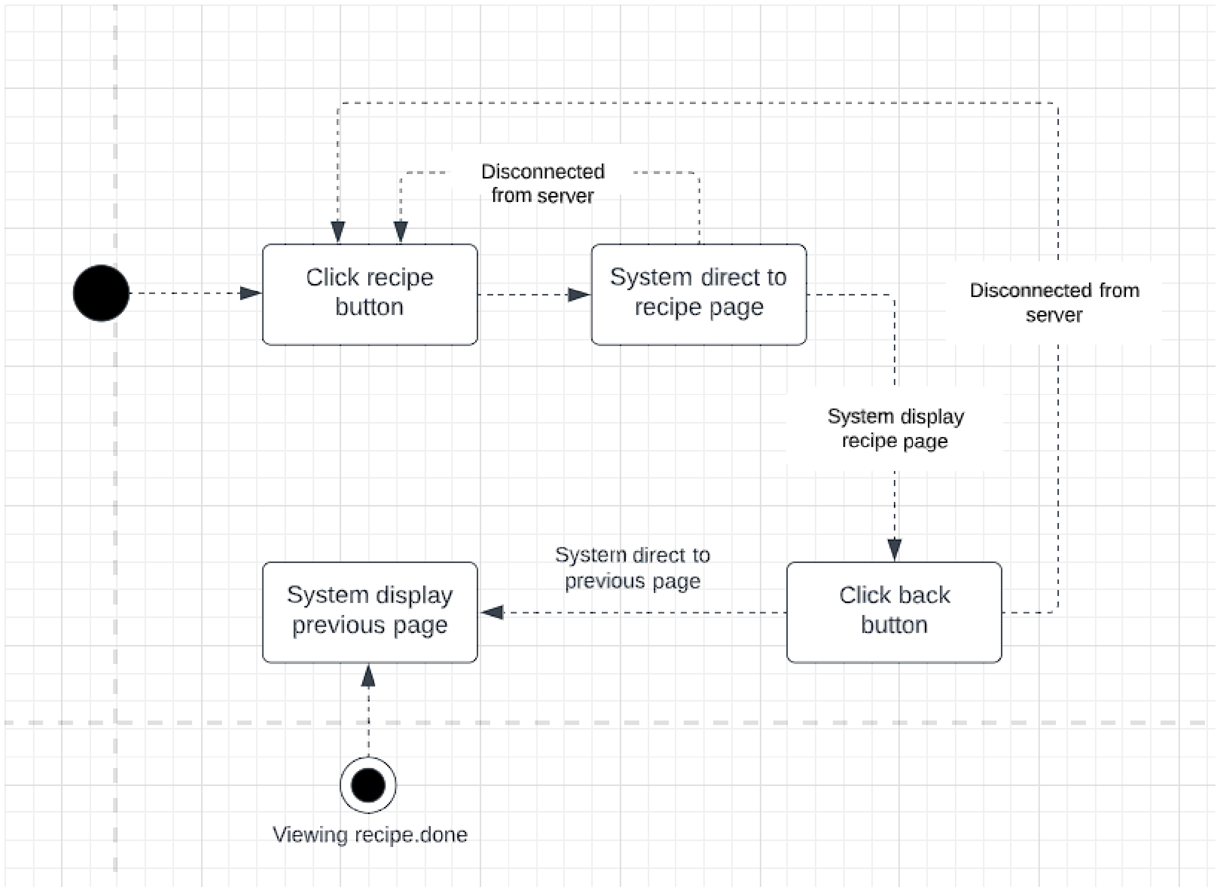
3.5.2 Login Statechart



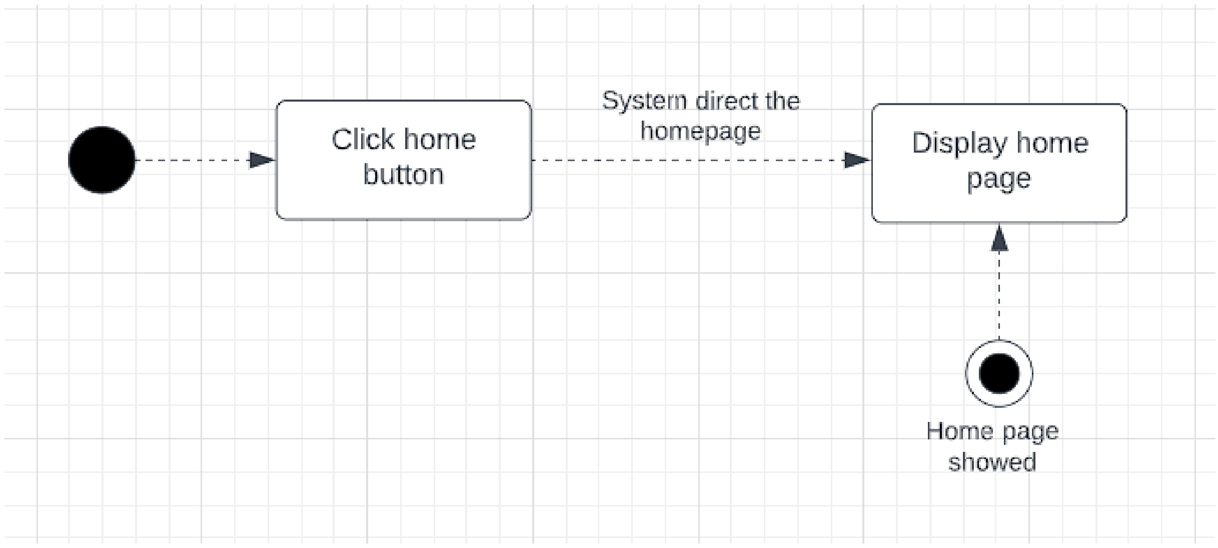
3.5.3 Search Food Statechart


Input keyword on search box
System verified the keyword
System display related item
System verified the keyword
Re-input keyword
System inform "No related item"
Searching.done

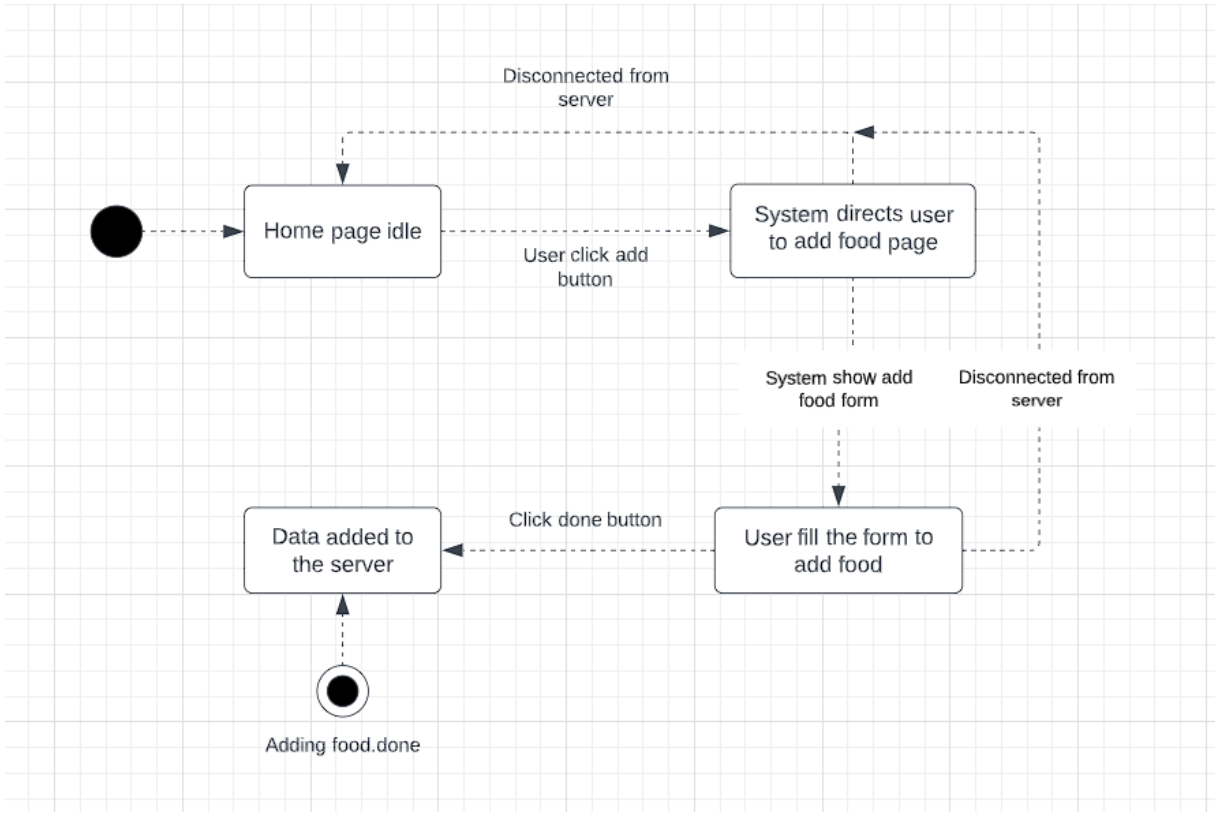
3.5.4 View Food Recipe Statechart


Disconnected from server
Click recipe button
System direct to recipe page
Disconnected from server
System display recipe page
System direct to previous page
System display previous page
Click back button
Viewing recipe.done

3.5.5 Show Home Statechart



3.5.6 Add Food Statechart



4. Trace Matrix

Class	**Use Case Related**
User	Registrasi
User	Login
User	Logout
User	View Food Recipe
User	Search Food Recipe
User	Show Home