Spotifood

Software Requirement Specification

By

Mr. Kittinut Saengsri 592115003

Mr. Sirisak Wongwal 592115031

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

Project Advisor

Dr. Passakorn Phannachitta

Document History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Document name | Version | Status | Date | Viewable | Editable | Responsible |
| Spotifood\_SRS\_v0.1 | - Add chapter one  - Add chapter two  - Add chapter three  - Add chapter four | Draft | 07/10/2019 | SW, KS, PP | SW, KS | SW, KS |
| Spotifood\_SRS\_v1.0 | - Update chapter one  - Update chapter two  - Update chapter three  - Update chapter four | Draft | 07/29/2019 | SW, KS, PP | SW, KS | SW, KS |

\*SW = Sirisak Wongwal

\*KS = Kittinut Saengsri

\*PP = Dr.Passakorn Phannachitta

Table of Contents

[Chapter 1 | Introduction 5](#_Toc15338042)

[1.1 Identification 5](#_Toc15338043)

[1.2 Project Overview 5](#_Toc15338044)

[1.2.1 Purpose 5](#_Toc15338045)

[1.2.2 Scope 5](#_Toc15338046)

[1.3 Work Product to be developed 6](#_Toc15338047)

[1.4 Acronyms and Definitions 6](#_Toc15338048)

[1.4.1 Acronyms 6](#_Toc15338049)

[CHAPTER 2 | Overall Description 7](#_Toc15338050)

[2.1 Product Perspective 7](#_Toc15338051)

[2.2 Product Function 7](#_Toc15338052)

[2.3 User Characteristics 8](#_Toc15338053)

[CHAPTER 3 | USER REQUIREMENT SPECIFICATION 9](#_Toc15338054)

[3.1 URS - Progress 1 9](#_Toc15338055)

[Feature#1: Authentication system 9](#_Toc15338056)

[Feature#2: Content CRUD 9](#_Toc15338057)

[Feature#3: User management 9](#_Toc15338058)

[CHAPTER 4 | SOFTWARE REQUIREMENT SPECIFICATION 10](#_Toc15338059)

[4.1 URS-SRS 10](#_Toc15338060)

[Feature#1: Authentication system 10](#_Toc15338061)

[Feature #2: Content CRUD 11](#_Toc15338062)

[Feature#3: User management 13](#_Toc15338063)

[4.2 Requirements Specifications 15](#_Toc15338064)

[4.3 Use case Description 18](#_Toc15338065)

[4.3 Activity diagram 43](#_Toc15338066)

# Chapter 1 | Introduction

## 1.1 Identification

This Project Management Plan is the document for planning, scheduling activities and evaluating overall of the project so that it could complete successfully despite all the risks. It provides detailed scheduled, assigned task and identified risks of the project.

## 1.2 Project Overview

The Spotifood application will guide people find their new dish and help people who are interested in cooking and people who have cooking skill by suggesting the dishes from a word that people search. The AI system will suggest a dish from the personality of people. For example, The people regularly search steamed fish with herbs dish. The AI system will analyze the ingredient tag of dish that is related to the healthy tag. Then, the AI will suggest the people in the type of healthy dishes.

The suggest dish of AI will help those people found new dishes and challenging in cooking every day that makes people not disconnected and bored to cooking. Moreover, we will be a middleman that brings trader of ingredient to people who can’t find the ingredients in any food. The Spotifood help people to buy an ingredient. And the Spotifood will suggest a dish that is similar to personality or suggests from the behavior of people.

### 1.2.1 Purpose

The main purpose of developing the Spotifood is to help the people who are interested in cooking and who already have skills not bored with cooking or disconnected to prepare the food.

### 1.2.2 Scope

There are six major parts in Spotifood.

Feature #1: Authentication

Feature #2: Content CRUD

Feature #3: User management

Feature #4: Similar dishes identification

Feature #5: Personalization

Feature #6: ADS ingredient

## 1.3 Work Product to be developed

1.3.1 Deliverable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Deriverable/Release | Media | Copies | Date |
| 1 | Project proposal  - Spotifood\_proposal3 | Hard copy | 3 | 19/06/2019 |
| 2 | Progress report I  - Project management plan version 1.0  - Software requirement specification version 1.0  - Software design document version 1.0  - Test plan version 1.0  - Traceability record version 1.0  - Software version 0.1 | Hard copy | 3 | 29/07/2019 |

## 1.4 Acronyms and Definitions

### 1.4.1 Acronyms

UC Use case

URS User Requirement Specification

SRS Software Requirement Specification

CRUD Create-Read-Update-Delete

AI Artificial Intelligence

ADS Advertising

ID Identification Data

# CHAPTER 2 | Overall Description

## 2.1 Product Perspective

Spotifood contain two parts. First is web application and second is AI. Web application use for interaction. Then AI will retrieve data from web application and send back to web application again.

## 2.2 Product Function

Spotifood have five major parts.

* Authentication system

This feature is used for preventing the unwanted user to access some feature in the system.

* Content CRUD

User input ingredients and how-to-cook step in this form. Then the user shares their dish into the app page. And Feature is used to pin dishes(Dish that user interest and pin it) and user’s history dishes(Dish that user view it already).

* User management

This feature is used to show the dish details including ingredients and hot-to-cook step. And used to edit their personal information.

* Similar dish identification

Registered user search by name of the dish and show the dishes that similar to registered user searched.

* Personalization

This feature is used for suggestion a dish by type of food and word of dish that frequently searched.

* ADS ingredient

This function is used to help a registered user find ingredients that registered user can’t cook or main ingredients to use in the dishes. By using a sponsored ranking example, the shop that sold the ingredients is most popular or pay a lot to the system will show first. The shop info comes from website that we can contact shop allow to show.

## 2.3 User Characteristics

There are three types of users in Spotifood application: general, registered user, admin.

General users are the users of the web application. Which can use a few functions. That is register and sign in.

Registered users are the users of the web application. Which can use most of the functions. That is sign in, create post, edit profile, delete post, delete profile, and view dish.

Admins are the administrator of the web application. Admin can manage post and manage user.

# CHAPTER 3 | USER REQUIREMENT SPECIFICATION

## 3.1 URS - Progress 1

### Feature#1: Authentication system

USER: General user, Registered user

URS01- User can register to the system by input email, password, name, date of birth, and gender.

URS02 - User can sign in to the system by input email and password.

URS03 - User can sign out of the system by sign out nav bar.

USER: Admin

URS04 - Admin can sign in into the system by input email and password.

URS05 - Admin can sign out of the system by sign out nav bar.

### Feature#2: Content CRUD

USER: Registered user

URS06 - User can post the dish by input name of dishes, ingredients, how-to-cook of the dish, and thumbnail photos of the dish.

URS07 - User can delete the post of the dish.

URS08 - User can view a pin dish.

URS09 - User can view a history dish.

URS10 - User can select topic or type of food that user interested.

URS15 - User can edit their post.

### Feature#3: User management

USER: Registered user

URS11- User can view a dish that user want.

URS12 - User can update their name, about, email and password.

URS13- User or upload and update their profile picture.

URS14- User can update their personality of interesting food.

URS16 - User can view their profile page.

# CHAPTER 4 | SOFTWARE REQUIREMENT SPECIFICATION

**Use case**

**UC01:** User can register to the system.

**UC02:** User can sign in to the system.

**UC03:** User can sign out of the system.

**UC04:** Admin can sign in to the system.

**UC05:** Admin can sign out of the system.

**UC06:** User can update their profile.

**UC07:** User can update their personality of interesting food.

**UC08:** Edit profile picture.

**UC09:** User can post the dish.

**UC10:** User can delete the posted dish.

**UC11:** User can like dish.

**UC12:** User can unlike dish.

**UC13:** User can view like dish

**UC14:** User can view history.

**UC15:** User can select topic.

**UC16:** User can view dish.

**UC17:** User can edit their post.

**UC18:** User can view profile page.

## 4.1 URS-SRS

### Feature#1: Authentication system

**URS01** - User can register to the system by input email, password, name, date of birth, and gender.

SRS-01: The system shall provide the user interface for receiving an email, password, name, and gender.

SRS-02: The system shall provide an error message “This field can not empty” when the textbox to empty.

SRS-03: The system shall provide a button to register.

SRS-04: The system shall validate the email, password, name, and gender of the user.

SRS-05: The system shall provide an error message “This [input] is incorrect format.” if input incorrect format of email, password, name, and gender.

SRS-06: The system shall provide an error message “This email already exists.” if the input email is already in database.

SRS-07: The system shall display a message “New account is successfully created. Please Sign In”.

SRS-66: The system shall update data in the database.

**URS02** - User can sign in to the system by input email and password.

SRS-08: The system shall provide the user interface for receiving email and password.

SRS-09: The system shall provide a button to sign in.

SRS-10: The system shall validate the email and password.

SRS-11: The system shall provide an error message “Incorrect password” if email is existing but password does not match.

SRS-12: The system shall provide an error message “This email does not exist” if email does not exist in the database.

SRS-13: The system redirects user to homepage.

**URS03** - User can sign out of the system by sign out button.

SRS-14: The system shall provide a button to sign out.

SRS-15: The system shall be logged out.

SRS-16: The system redirects user to the home page.

**URS04** - Admin can sign in into the system by input email and password.

SRS-17: The system shall provide a user interface for receiving email and password.

SRS-18: The system shall validate the email and password.

SRS-19: The system shall provide an error message “Incorrect password” if email is existing but password does not match.

SRS-20: The system shall provide an error message “This email does not exist” if email does not exist in the database.

SRS-21: The system redirects admin to homepage.

**URS05** - Admin can sign out of the system by sign out nav bar.

SRS-22: The system shall provide a button to sign out.

SRS-23: The system shall be logged out.

SRS-24: The system redirects admin to the home page.

### Feature #2: Content CRUD

**URS06** - User can post the dish by input name of dishes, ingredients, how-to-cook of the dish, and thumbnail photos of the dish.

SRS-25: The system shall provide the user interface for receiving data of the dish including:

- Name of dish

- Ingredients

- How-to-cook of the dish

- Upload thumbnail photos of food

SRS-26: The system shall provide a button to post.

SRS-27: The system shall validate the name of the dish, ingredients, and how-to-cook of the dish, and thumbnail photos of the dish.

SRS-28: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-29: The system shall provide an error message “File size should be less than 150kb” when file size is more than 150kb.

SRS-30: The system redirects registered user to the user’s profile.

SRS-66: The system shall update data in the database.

**URS07** - User can delete the post of the dish.

SRS-31: The system shall provide the delete button for the user.

SRS-32: The system shall provide the confirm button for the user after click delete button.

SRS-33: The system shall delete the dish when the user clicks the delete button and confirm to delete the dish.

SRS-34: The system redirects registered user to homepage.

SRS-35: The system shall alert message “Are you sure to delete the post?”.

SRS-66: The system shall update data in the database.

**URS08** - User can view a pin dish.

SRS-36: The system shall provide the pin button for the user.

SRS-37: The system shall cancel the pin dish when the user clicks a pin button again.

SRS-38: The system shall update the data into the database after the user clicks pin button.

SRS-39: The system shall provide the user interface to show a pin dish.

SRS-65: The system will retrieve data in database.

**URS09** - User can view a history dish.

SRS-40: The system shall provide the history dish button to show a history dish.

SRS-41: The system shall redirect to history dish page when the user clicks the history button.

SRS-65: The system will retrieve data in database.

**URS10** - User can select topic or type of food that user interested.

SRS-42: The system shall provide the user interface for receiving a type of food that user interested.

SRS-65: The system will retrieve data in database.

**URS11** - User can view a dish that user want.

SRS-67: The registered user clicks a name of the dish.

SRS-65: The system will retrieve data in database.

SRS-68: The system shall redirect user to view dish page.

### Feature#3: User management

**URS12** - User can update their name, about, email and password.

SRS-46: The system shall provide the user interface for editing data of the dish including:

- Name

- About

- Email

- Password

SRS-47: The system shall provide the edit button to the user.

SRS-48: The system shall validate name, about, email and password.

SRS-49: The system saves the new data of user management to the database after the user input all data validation is passed.

SRS-50: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-51: The system shall provide an error message “A valid email is required” when the textbox is invalid format. (ex. dummy123@hotmail.com)

SRS-52: The system shall provide an error message “Email is taken!”. if input email is already in database.

SRS-66: The system shall update data in the database.

SRS-53: The system redirects registered user to profile page.

**URS13** - User or upload and update their profile picture.

SRS-59: The system shall provide the user interface for upload profile picture.

SRS-60: The system shall provide the upload button to the user.

SRS-61: The system shall provide an error message “File size should be less than 150kb” when file size is more than 100kb.

SRS-66: The system shall update data in the database.

SRS-62: The system redirects user to user’s profile page.

**URS14** - User can update their personality of interesting food.

SRS-54: The system shall provide the user interface for personality of interesting food.

SRS-55: User can select tag that user wants to update on personality. And user click selected tag again to unselect.

SRS-56: The system shall provide a submit button to the user.

SRS-57: The system saves the new data of User management to the database.

SRS-58: The system redirects user to the homepage.

SRS-66: The system shall update data in the database.

**URS15** - User can edit their post.

SRS-25: The system shall provide the user interface for receiving data of the dish including:

- Name of dish

- Ingredients

- How-to-cook of the dish

- Upload thumbnail photos of food

SRS-51: The system shall provide edit button.

SRS-28: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-45: The system shall provide an error message “File size should be less than 150kb” when file size is more than 150kb.

SRS-32: The system redirects registered user to the user’s profile.

SRS-66: The system shall update data in the database.

**URS16** - User can view their profile page.

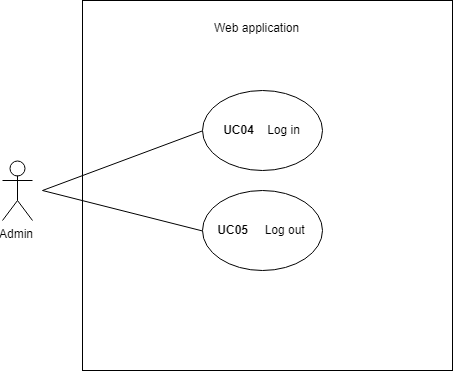
SRS-63: The system shall provide profile nav bar on the web application.

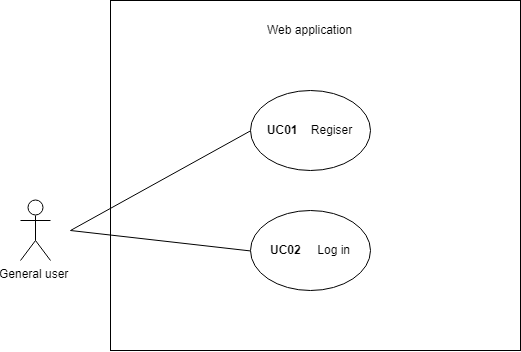
SRS-64: The system shall redirect user to their profile page.

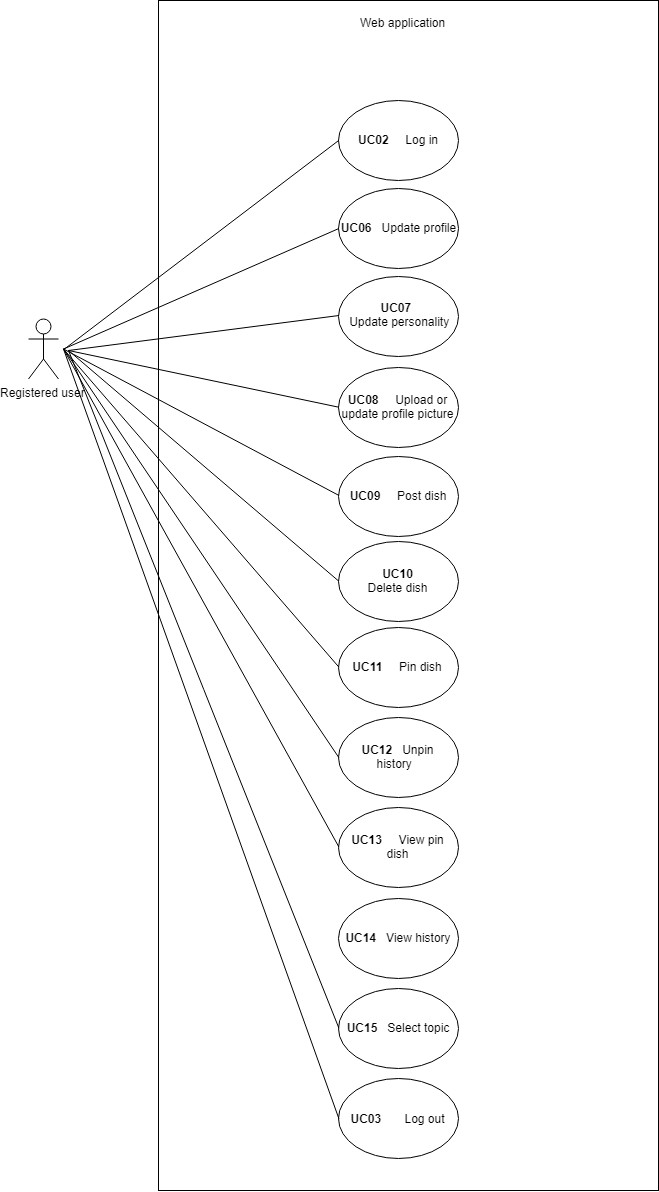
SRS-65: The system will retrieve data in database.

## 4.2 Requirements Specifications

Use case diagram







## 4.3 Use case Description

**Feature#1:** Authentication system

Description: This feature is used for preventing the unwanted user to access some feature in the system.

Actor: Admin, Registered system

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC01 | | |
| Use Case Name: | Register | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/20/2019 |
| Actor: | General user | | |
| Description: | General user can register to have their own account. | | |
| Trigger: | General user clicks the register nav bar on the web application. | | |
| Precondition: | General user open Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| Email | String | - This field can not empty.  - Have to input correct format. | temabc123@hotmail.com |
| Password | String | - This field can not empty.  - Have to input correct format. | passcode123 |
| Name | String | - This field can not empty.  - Have to input correct format. | Kittinut |
| Gender | String | - This field can not empty.  - Have to input correct format. | Male |
| DateOfBirth | integer | - This field can not empty | 01/01/1111 |
| Post Condition: | General user has an account for sign in. | | |
| Normal Flow: | 1. General user clicks the register bar. 2. The system will redirect the general user to register page. 3. General user input name. 4. General user input date of birth. 5. General user input gender. 6. General user input email. 7. General user input password. 8. General user click submit button. 9. The system validates the input. 10. The system will save account to the database. [A1][A2][A3] 11. The system shall display a message “New account is successfully created. Please Sign In”. | | |
| Alternative Flow: | [A1] If the text field is empty.   1. The system shall provide an error message “This [input] field can not empty”. 2. Back to normal flow 3.   [A2] If the email is incorrect format.   1. The system shall provide an error message “Input a valid email”. 2. Back to normal flow 6.   [A3] If the email already exists.   1. The system shall provide an error message “This email is taken!”. 2. Back to normal flow 6. | | |
| Exception Flow: | - | | |
| Assumption: | 1. General user must connect with the internet. | | |

Related Requirements

**URS01** - User can register to the system by input email, password, name, date of birth, and gender.

SRS-01: The system shall provide the user interface for receiving an email, password, name, and gender.

SRS-02: The system shall provide an error message “This field can not empty” when the textbox to empty.

SRS-03: The system shall provide a button to register.

SRS-04: The system shall validate the email, password, name, and gender of the user.

SRS-05: The system shall provide an error message “This [input] is incorrect format.” if input incorrect format of email, password, name, and gender.

SRS-06: The system shall provide an error message “This email already exists.” if the input email is already in database.

SRS-07: The system shall display a message “New account is successfully created. Please Sign In”.

SRS-66: The system shall update data in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC02 | | |
| Use Case Name: | User sign in | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/20/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can sign in into the system with their own account. | | |
| Trigger: | The registered user click the sign in nav bar on the web application. | | |
| Precondition: | The registered user open Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| Email | String | - This field can not empty.  - Have to input correct format. | temabc123@hotmail.com |
| Password | String | - This field can not empty.  - Have to input correct format. | passcode123 |
| Post Condition: | The registered user sign in into the system. | | |
| Normal Flow: | 1. Registered user click sign in nav bar. 2. The system shall redirect registered user to the sign in page. 3. The registered user input email. 4. The registered user input password. 5. The registered user input click the sign in button. 6. The system validates the account. [A1][A2] 7. The system redirect registered user to the homepage. | | |
| Alternative Flow: | [A1] If the email does not exist in the database.   1. The system shall provide a message “This email does not exist” 2. Go back to normal flow 4.   [A2] If the password does not correct.   1. The system shall provide a message “Incorrect password” 2. Go back to normal flow 4. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS02** - User can sign in to the system by input email and password.

SRS-08: The system shall provide the user interface for receiving email and password.

SRS-09: The system shall provide a button to sign in.

SRS-10: The system shall validate the email and password.

SRS-11: The system shall provide an error message “Incorrect password” if email is exist but password not match.

SRS-12: The system shall provide an error message “This email does not exist” if email does not exist in the database.

SRS-13: The system redirects user to homepage.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC03 | | |
| Use Case Name: | User sign out | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/20/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can sign out of the system. | | |
| Trigger: | The registered user click the sign out nav bar on the web application. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user sign out of the system. | | |
| Normal Flow: | 1. The registered user clicks the sign out button. 2. The system redirect registered user to the home page. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS03** - User can sign out of the system by sign out nav bar.

SRS-14: The system shall provide a button to sign out.

SRS-15: The system shall be logged out.

SRS-16: The system redirects user to the home page.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC04 | | |
| Use Case Name: | Admin sign in | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/05/2019 |
| Actor: | Admin | | |
| Description: | The admin can sign in into the system with their own account. | | |
| Trigger: | The admin click the sign in nav bar on the web application. | | |
| Precondition: | The admin open Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| Email | String | - This field can not empty.  - Have to input correct format. | temabc123@hotmail.com |
| Password | String | - This field can not empty.  - Have to input correct format. | passcode123 |
| Post Condition: | The admin sign in into the system. | | |
| Normal Flow: | 1. Admin click sign in nav bar. 2. The system shall redirect admin to the sign in page. 3. The admin input email. 4. The admin input password. 5. The admin input click the sign in button. 6. The system validates the account. [A1][A2] 7. The system redirect admin to the homepage. | | |
| Alternative Flow: | [A1] If the email does not exist in the database.   1. The system shall provide a message “This email does not exist” 2. Go back to normal flow 4.   [A2] If the password does not correct.   1. The system shall provide a message “Incorrect password” 2. Go back to normal flow 4. | | |
| Exception Flow: | - | | |
| Assumption: | 1. The admin must connect to the internet. | | |

Related Requirements

**URS04** - Admin can sign in into the system by input email and password.

SRS-17: The system shall provide a user interface for receiving email and password.

SRS-18: The system shall validate the email and password.

SRS-19: The system shall provide an error message “Incorrect password” if email is exist but password not match.

SRS-20: The system shall provide an error message “This email does not exist” if email does not exist in the database.

SRS-21: The system redirect admin to homepage.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC05 | | |
| Use Case Name: | Admin sign out | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/05/2019 |
| Actor: | Admin | | |
| Description: | The admin can sign out of the system. | | |
| Trigger: | The admin click the sign out nav bar on the web application. | | |
| Precondition: | The admin must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The admin sign out of the system. | | |
| Normal Flow: | 1. The admin click sign out nav bar. 2. The system redirect admin to the sign in page. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS05** - Admin can sign out of the system by sign out nav bar.

SRS-22: The system shall provide a button to sign out.

SRS-23: The system shall be logged out.

SRS-24: The system redirects admin to the home page.

**Feature#2**: Content CRUD

Description: User input ingredients and how-to-cook step in this feature. Then the user shares their dish into the app page. And feature is used to pin dishes (Dish that user interest and pin it) and user’s history dishes (Dish that user view it already).

Actor: Registered user

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC09 | | |
| Use Case Name: | Post dish | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/05/2019 |
| Actor: | Registered user | | |
| Description: | User can post the dish by input name of dishes, ingredients, and how-to-cook of the dish, tag of the dish, upload their insert photo of each step and thumbnail photos of the dish. | | |
| Trigger: | The registered user clicks the create post nav bar on the web application. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| nameOfDish | String | This field can not empty. | Omelet |
| ingredient | String | This field can not empty. | Egg, Olive oil, Fish sauce |
| howToCook | String | This field can not empty. | 1. Break the egg and put it into a bowl.  2. Add fish sauce 1 tablespoon.  3. Mix it together  4. Put olive oil into the pan and use medium heat.  5. Put a mixed egg into the pan.  6. Wait until the below side looks cooked then turn it back.  7. Wait until all side of omelet is looking perfect.  8. Put it into the dish and ready to eat. |
| Post Condition: | A new dish saves into the database. | | |
| Normal Flow: | 1. The registered user clicks create post nav bar. 2. The system redirect registered user to the post dish page. 3. The registered user input the name of the dish. 4. The registered user input ingredients. 5. The registered user input how-to-cook of the dish. 6. The registered user uploads thumbnail photos of food.[A1] 7. The registered user clicks create post button. 8. The system will check the text box. [A2] 9. The system updates the database. 10. The system redirect registered user to the posted dish view page. | | |
| Alternative Flow: | [A1] If the photo size is more than 150 kb.   1. The system displays a message “File size should be less than 150kb”. 2. Go to normal flow 7.   [A2] If the textbox is empty.   1. The system displays a message “This [input] field can not empty”. 2. Go to normal flow 3. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS06** - User can post the dish by input name of dishes, ingredients, how-to-cook of the dish, and thumbnail photos of the dish.

SRS-25: The system shall provide the user interface for receiving data of the dish including :

- Name of dish

- Ingredients

- How-to-cook of the dish

- Upload thumbnail photos of food

SRS-26: The system shall provide a button to post.

SRS-27: The system shall validate the name of the dish, ingredients, and how-to-cook of the dish, and thumbnail photos of the dish.

SRS-28: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-29: The system shall provide an error message “File size should be less than 150kb” when file size is more than 150kb.

SRS-30: The system redirect registered user to the user’s profile.

SRS-66: The system shall update data in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC10 | | |
| Use Case Name: | Delete dish | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user uses this function to delete their own posts. | | |
| Trigger: | The registered user clicks name of their own post. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | A selected dish delete from the database. | | |
| Normal Flow: | 1. The registered user clicks name of their own post. 2. The registered user click delete post button. 3. The system shall display a message “Are you sure to delete the post?”[A1][A2] 4. The system update data in the database. 5. The system redirect registered user to the homepage. | | |
| Alternative Flow: | [A1] If registered user clicks the Yes button.   1. Go to normal flow 3.   [A2] If registered user click No button.   1. Go to normal flow 5. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS07** - User can delete the post of the dish.

SRS-31: The system shall provide the delete button for the user.

SRS-32: The system shall provide the confirm button for the user after click delete button.

SRS-33: The system shall delete the dish when the user clicks the delete button and confirm to delete the dish.

SRS-34: The system redirect registered user to homepage.

SRS-35: The system shall alert message “Are you sure to delete the post?”.

SRS-66: The system shall update data in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC11 | | |
| Use Case Name: | Like dish | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can pin the dish that they like. | | |
| Trigger: | The registered user clicks name of dish to view dish. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user will have a new dish in their pin dish view. | | |
| Normal Flow: | 1. The registered user click name of the dish. 2. The system redirect user to view dish page. 3. The registered user clicks like button on the view dish page. 4. The system will update data in the database. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS08** - User can view a like dish.

SRS-36: The system shall provide the like button for the user.

SRS-37: The system shall unlike the dish when the user clicks a like button again.

SRS-38: The system shall update the data into the database after the user clicks pin button.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC12 | | |
| Use Case Name: | Unlike dish | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can unlike the dish that they do not like it anymore. | | |
| Trigger: | The registered user clicks name of dish to view dish. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user will remove the selected dish in their view like dish. | | |
| Normal Flow: | 1. The registered user click name of the dish. 2. The system redirect user to view dish page. 3. The registered user clicks like button to unlike on the view dish page. 4. The system will update data in the database. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS08** - User can view a like dish.

SRS-36: The system shall provide the like button for the user.

SRS-37: The system shall unlike the dish when the user clicks a like button again.

SRS-38: The system shall update the data into the database after the user clicks pin button.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC13 | | |
| Use Case Name: | View like dish | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can view like dish on the Spotifood web application. | | |
| Trigger: | The registered user clicks view like dish nav bar. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user will see all the dish that they pin. | | |
| Normal Flow: | 1. The registered user clicks view pin dish button. 2. The system retrieve data in database. 3. The system redirects the registered user to view pin dish page. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS08** - User can view a pin dish.

SRS-39: The system shall provide the user interface to show a pin dish.

SRS-65: The system will retrieve data in database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC14 | | |
| Use Case Name: | View history | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can view the history of their account. Only the dish that they ever click in the past. | | |
| Trigger: | The registered user clicks history nav bar. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user will see all the dish that they ever click on it. | | |
| Normal Flow: | 1. The registered user clicks history nav bar. 2. The system retrieve data in database. 3. The system will redirect registered user to history page. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS09** - User can view a history dish.

SRS-40: The system shall provide the history dish button to show a history dish.

SRS-41: The system shall redirect to history dish page when the user clicks the history button.

SRS-65: The system will retrieve data in database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC15 | | |
| Use Case Name: | Select topic | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/06/2019 | Last Revision Date: | 07/06/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can select the type of food that user interest. And system will show the dish that tag the same as that type of food. | | |
| Trigger: | The registered user clicks select topic nav bar. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | The registered user will see all the dishes have the same tag as a topic that they select. | | |
| Normal Flow: | 1. The registered user click topic nav bar. 2. The system redirects the user to select topic page. 3. The registered user click topic that they want. 4. The system retrieve data in the database. 5. The system shows all dishes that have the tag same as a selected topic. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS10** - User can select topic or type of food that user interested.

SRS-42: The system shall provide the user interface for receiving a type of food that user interested.

SRS-65: The system will retrieve data in database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC17 | | |
| Use Case Name: | Edit post | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/10/2019 | Last Revision Date: | 07/10/2019 |
| Actor: | Registered user | | |
| Description: | User can edit their post the dish by input name of dishes, ingredients, and how-to-cook of the dish, tag of the dish, upload their insert photo of each step and thumbnail photos of the dish. | | |
| Trigger: | The registered user clicks the edit post button on the web application. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| nameOfDish | String | This field can not empty. | Omelet |
| ingredient | String | This field can not empty. | Egg, Olive oil, Fish sauce |
| howToCook | String | This field can not empty. | 1. Break the egg and put it into a bowl.  2. Add fish sauce 1 tablespoon.  3. Mix it together  4. Put olive oil into the pan and use medium heat.  5. Put a mixed egg into the pan.  6. Wait until the below side looks cooked then turn it back.  7. Wait until all side of omelet is looking perfect.  8. Put it into the dish and ready to eat. |
| Post Condition: | A new dish saves into the database. | | |
| Normal Flow: | 1. The registered user clicks edit post button. 2. The system redirect registered user to the edit dish page. 3. The registered user input name of the dish. 4. The registered user input ingredients. 5. The registered user input how-to-cook of the dish. 6. The registered user uploads thumbnail photos of food.[A1][A2][A3] 7. The registered user clicks the edit button. 8. The system will check the text box. [A3] 9. The system updates the database. 10. The system redirect registered user to user’s profile page. | | |
| Alternative Flow: | [A1] If the photo does not exist.   1. The system displays a message “Select a file to upload photo.”. 2. Go to normal flow 7.   [A2] If the photo size is more than 150 kb.   1. The system displays a message “Photo size is more than 150 kb.”. 2. Go to normal flow 7.   [A3] If the textbox is empty.   1. The system displays a message “This [input] field can not empty”. 2. Go to normal flow 3. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS15** - User can edit their post.

SRS-25: The system shall provide the user interface for receiving data of the dish including :

- Name of dish

- Ingredients

- How-to-cook of the dish

- Upload thumbnail photos of food

SRS-51: The system shall provide edit button.

SRS-28: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-45: The system shall provide an error message “File size should be less than 150kb” when file size is more than 150kb.

SRS-32: The system redirect registered user to the user’s profile.

SRS-66: The system shall update data in the database.

**Feature#3**: User management

Description: This feature is used to post the dish details including ingredients and how-to-cook step. And used to edit their personal information.

Actor: Registered system

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC06 | | |
| Use Case Name: | Edit profile | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/21/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can update their own profile. Update email, password, name, gender, and profile picture. | | |
| Trigger: | The registered user click the edit profile button on view profile page. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| Email | String | - This field can not empty.  - Have to input correct format. | temabc123@hotmail.com |
| Password | String | - This field can not empty.  - Have to input more than 6 characters. | passcode123 |
| name | String | - This field can not empty. | Kittinut |
| About | String | - | Hi. I love sushi. |
| Post Condition: | Profile of registered user has been updated. | | |
| Normal Flow: | 1. Registered user clicks edit profile button. 2. The system redirect registered user to the edit profile page. 3. Registered user input name. 4. Registered user input about. 5. Registered user input email. 6. Registered user input password. 7. Registered user clicks the update button. 8. The system will update the profile in the database.[A1][A2][A3] 9. The system will redirect the registered user to the homepage. | | |
| Alternative Flow: | [A1] If the text field is empty.   1. The system shall provide an error message “This [input] field can not empty”. 2. Back to normal flow 4.   [A2] If the input is incorrect format.   1. The system shall provide an error message “A valid email is required”. 2. Back to normal flow 4.   [A3] If the email already exists.   1. The system shall provide an error message “Email is taken!”. 2. Back to normal flow 4. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS12** - User can update their name, about, email and password.

SRS-46: The system shall provide the user interface for editing data of the dish including :

- Name

- About

- Email

- Password

SRS-47: The system shall provide the edit button to the user.

SRS-48: The system shall validate name , about , email and password.

SRS-49: The system saves the new data of user management to the database after the user input all data validation is passed.

SRS-50: The system shall provide an error message “This [input] field can not empty” when the textbox to empty.

SRS-51: The system shall provide an error message “A valid email is required” when the textbox is invalid format. (ex. dummy123@hotmail.com)

SRS-52: The system shall provide an error message “Email is taken!”. if input email is already in database.

SRS-66: The system shall update data in the database.

SRS-53: The system redirect registered user to profile page.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC07 | | |
| Use Case Name: | Edit personality | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/07/2019 | Last Revision Date: | 07/08/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can update their personality of interesting food. | | |
| Trigger: | The registered user click the personality nav bar on the web application. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | Personality of registered user has been updated. | | |
| Normal Flow: | 1. The registered user click the personality nav bar. 2. The system shall provide user interface for personality of interesting food. 3. User click tag that user want to update on personality of food. [A1] 4. The registered user click submit button. [A2] 5. The system saves the new data of user management to the database. 6. The system redirects user to the homepage. | | |
| Alternative Flow: | [A1] If registered user click tag again.   1. Tag will unselect. 2. Go to normal flow 3.   [A2] If registered user click cancel button.   1. Go to normal flow 2. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS14** - User can update their personality of interesting food.

SRS-54: The system shall provide the user interface for personality of interesting food.

SRS-55: User can select tag that user want to update on personality. And user click selected tag again to unselect.

SRS-56: The system shall provide a submit button to the user.

SRS-57: The system saves the new data of User management to the database.

SRS-58: The system redirects user to the homepage.

SRS-66: The system shall update data in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC08 | | |
| Use Case Name: | Edit profile picture. | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/05/2019 | Last Revision Date: | 07/05/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can upload their profile picture on profile page. | | |
| Trigger: | The registered user click the profile nav bar on the web application. | | |
| Precondition: | The registered user or admin must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | Profile picture of registered user has been updated. | | |
| Normal Flow: | 1. Registered user clicks the profile nav bar. 2. Registered user clicks the edit button. 3. Registered user click select file button. 4. The system validates file. [A1] 5. Registered user click update button. 6. The system update the database. 7. The system redirects user to user’s profile page. | | |
| Alternative Flow: | [A1] If picture file is more than 100kb.   1. The system shall provide an error message “File size should be less than 100kb”. 2. Go to normal flow 2. | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS13** - User or upload and update their profile picture.

SRS-59: The system shall provide the user interface for upload profile picture.

SRS-60: The system shall provide the upload button to the user.

SRS-61: The system shall provide an error message “File size should be less than 150kb” when file size is more than 100kb.

SRS-66: The system shall update data in the database.

SRS-62: The system redirects user to user’s profile page.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC16 | | |
| Use Case Name: | View dish. | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/08/2019 | Last Revision Date: | 07/08/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can view dish. | | |
| Trigger: | The registered user click on name of dish. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | User will redirect to selected view dish. | | |
| Normal Flow: | 1. Registered user click on any dish. 2. The system will retrieve data in database. 3. The system redirects user to view page of the dish. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS11** - User can view a dish that user want.

SRS-67: The registered user click a name of the dish.

SRS-65: The system will retrieve data in database.

SRS-68: The system shall redirect user to view dish page.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC18 | | |
| Use Case Name: | View profile. | | |
| Create By: | Kittinut Saengsri | Last Update By: | Kittinut Saengsri |
| Date Create: | 07/08/2019 | Last Revision Date: | 07/22/2019 |
| Actor: | Registered user | | |
| Description: | The registered user can view their profile page. | | |
| Trigger: | The registered user click on profile nav bar. | | |
| Precondition: | The registered user must sign in to the Spotifood web application. | | |
| Use Case Input Specification | | | |
| Input | Type | Constrain | Example |
| - | - | - | - |
| Post Condition: | Registered will redirect to their profile page. | | |
| Normal Flow: | 1. Registered user click on profile nav bar. 2. The system retrieve data in database. 3. The system redirect user to their profile page. | | |
| Alternative Flow: | - | | |
| Exception Flow: | - | | |
| Assumption: | 1. Registered user must connect to the internet. | | |

Related Requirements

**URS16** - User can view their profile page.

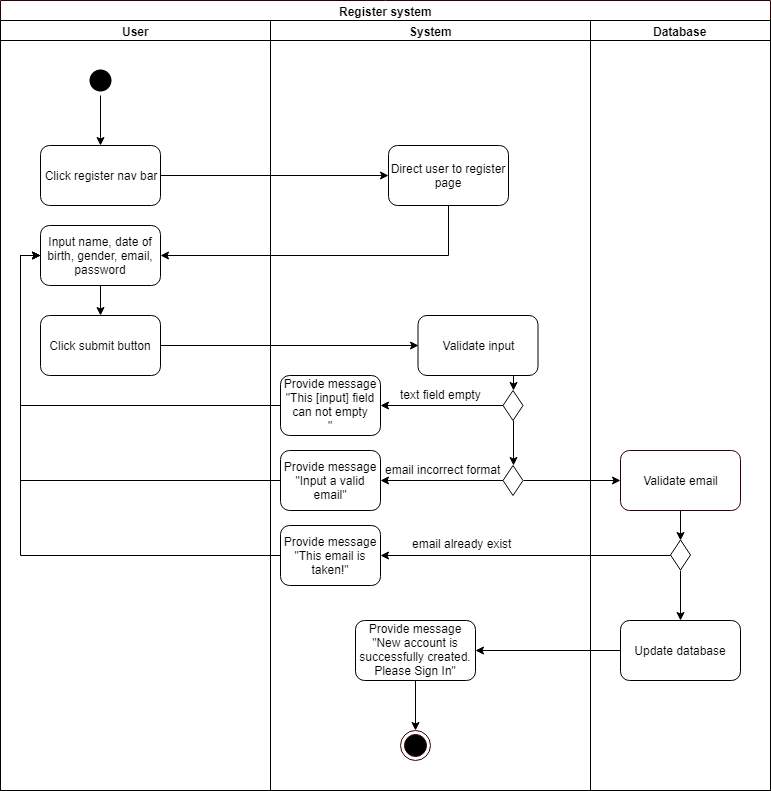
SRS-63: The system shall provide profile nav bar on the web application.

SRS-64: The system shall redirect user to their profile page.

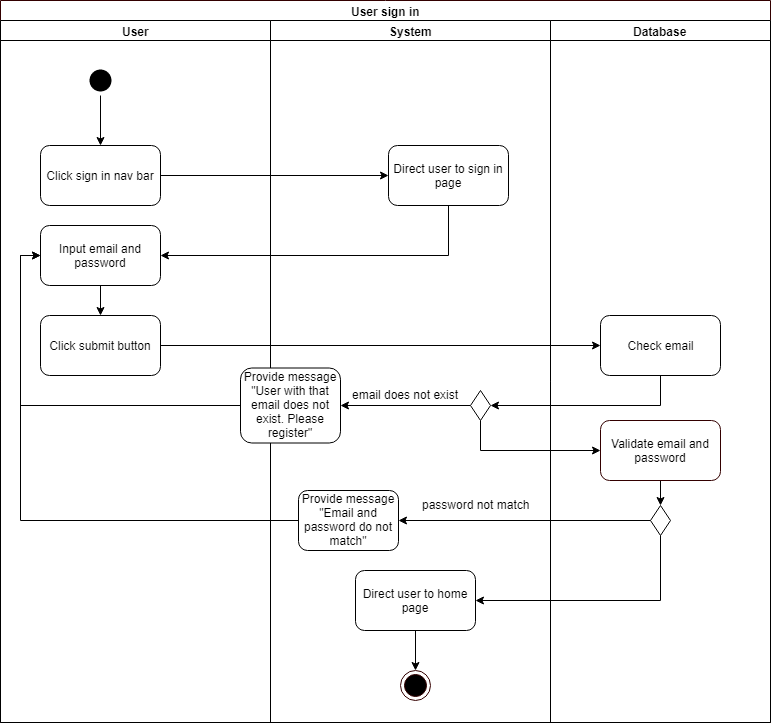
SRS-65: The system will retrieve data in database.

## 4.3 Activity diagram

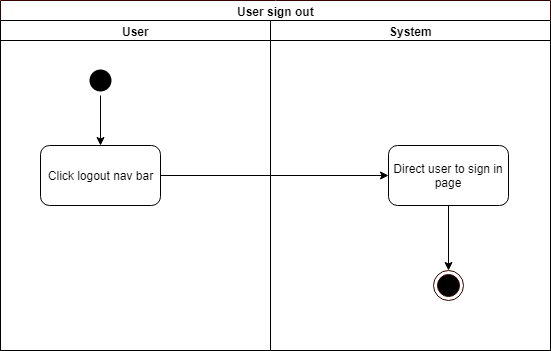
Activity diagram 1: AC01



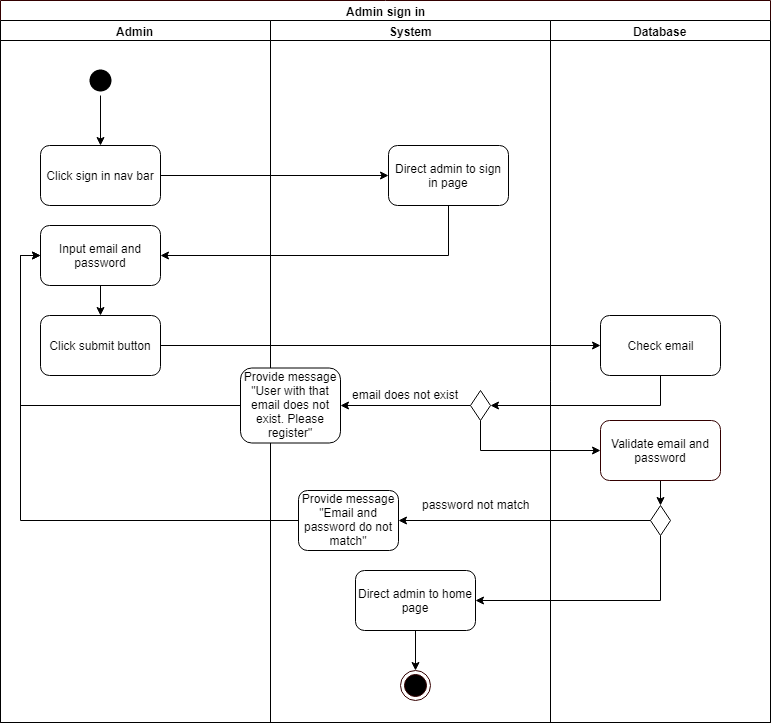
Activity diagram 2: AC02



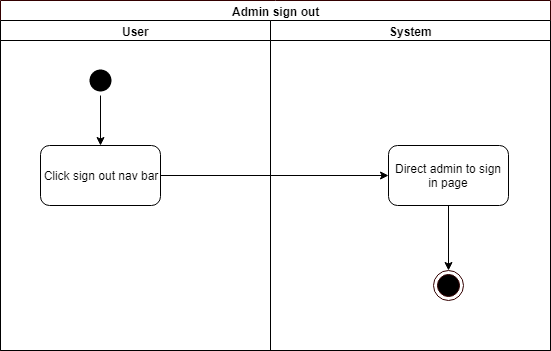
Activity diagram 3: AC03



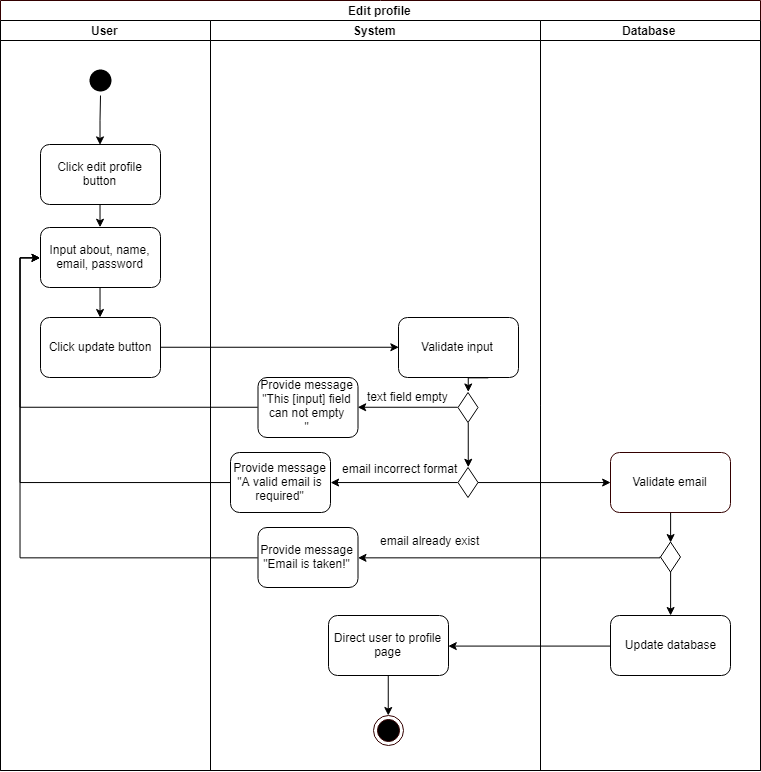
Activity diagram 4: AC04



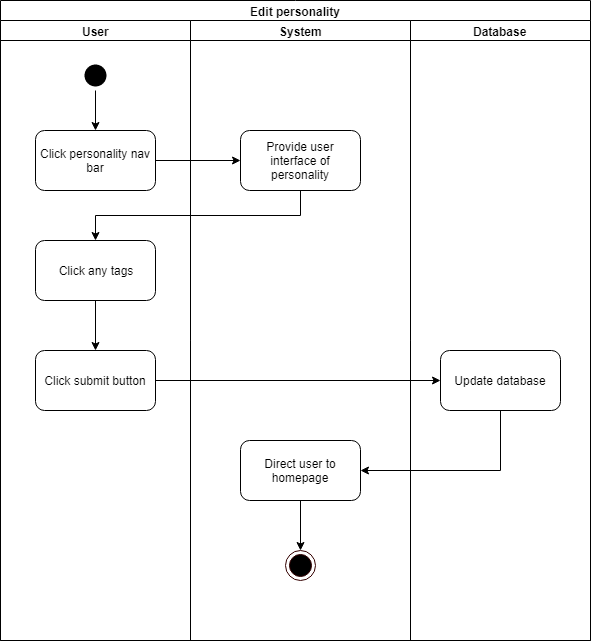
Activity diagram 5: AC05



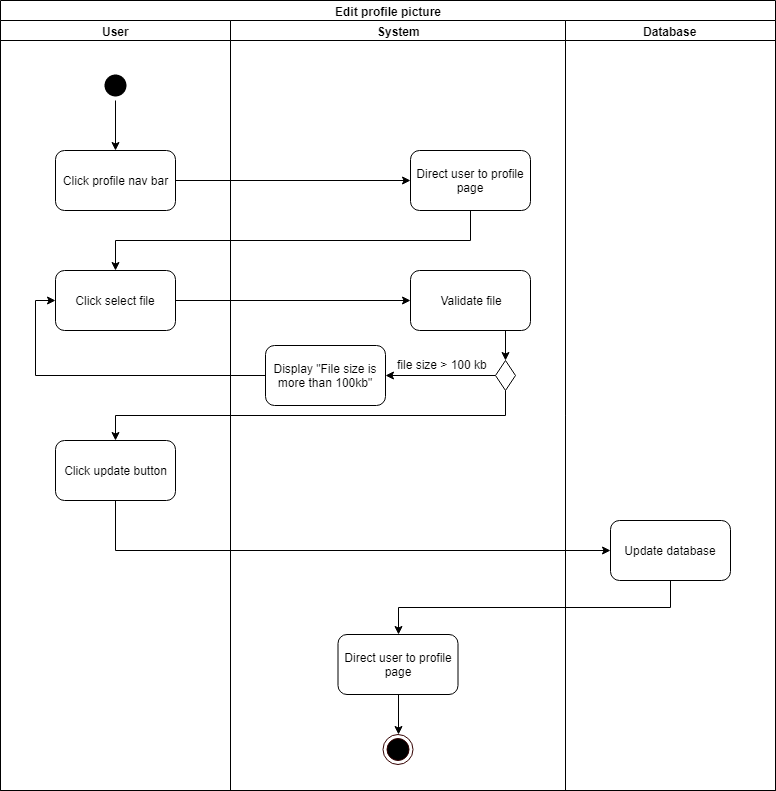
Activity diagram 6: AC06



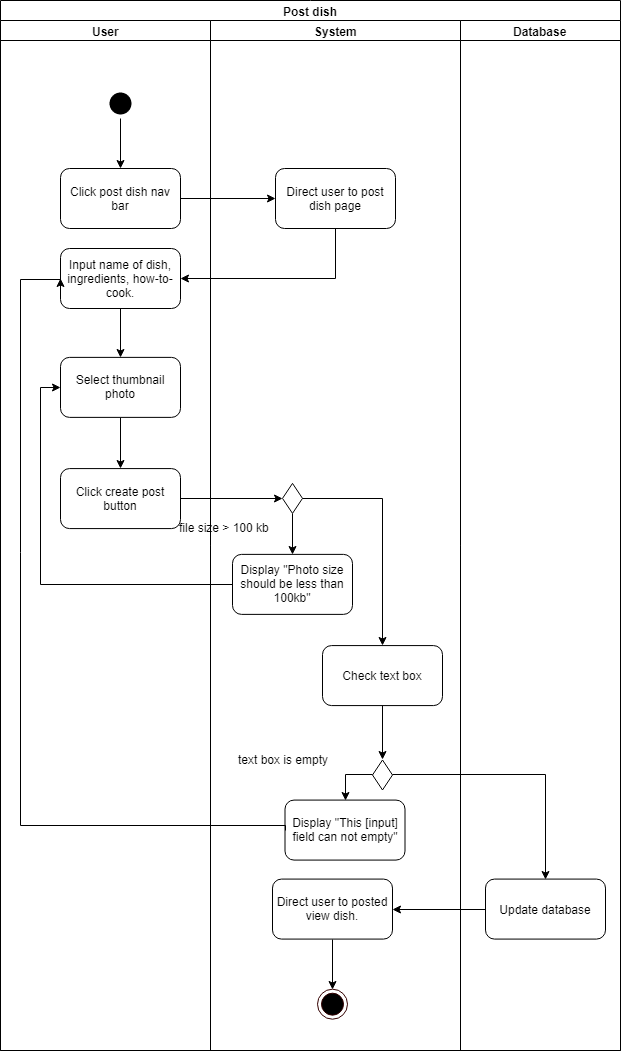
Activity diagram 7: AC07



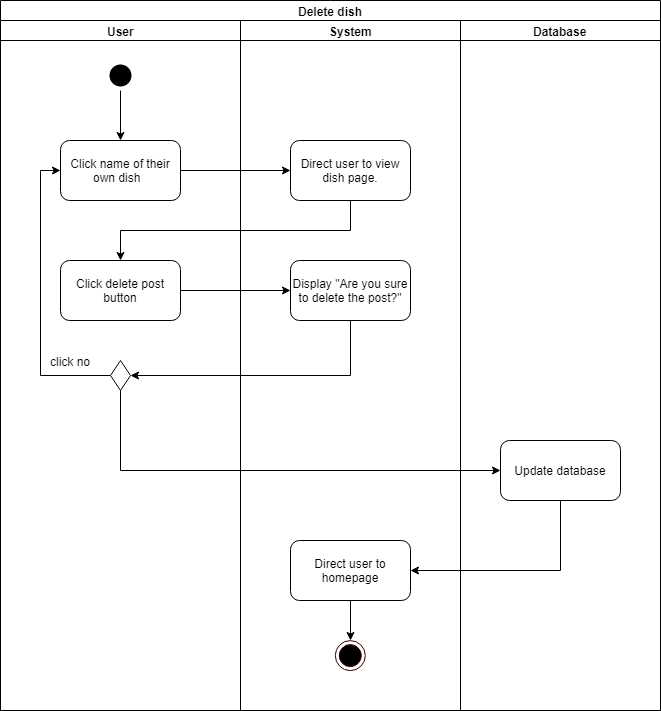
Activity diagram 8: AC08



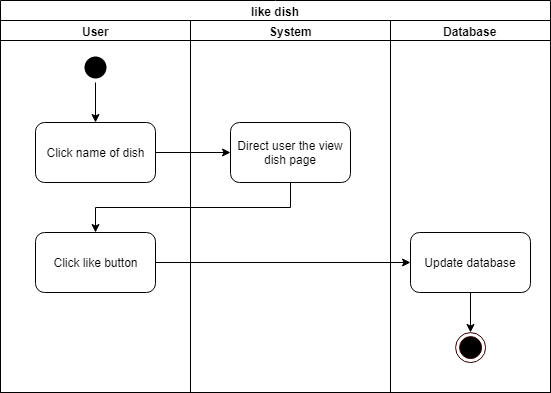
Activity diagram 9: AC09



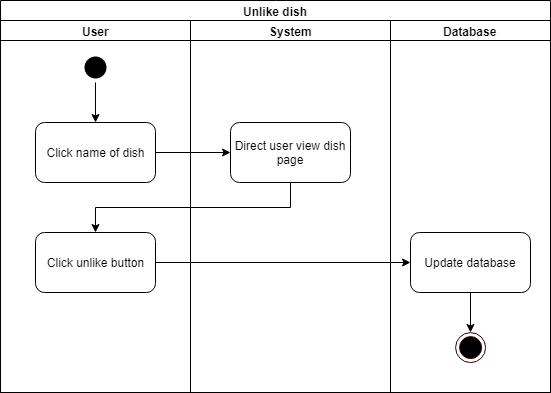
Activity diagram 10: AC10



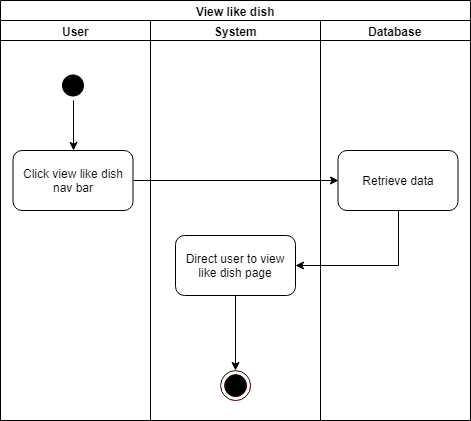
Activity diagram 11: AC11



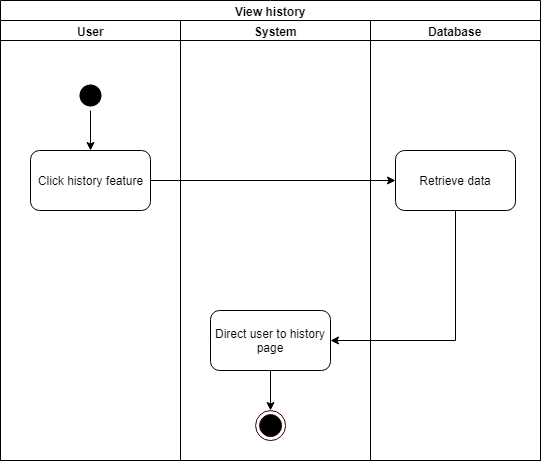
Activity diagram 12: AC12



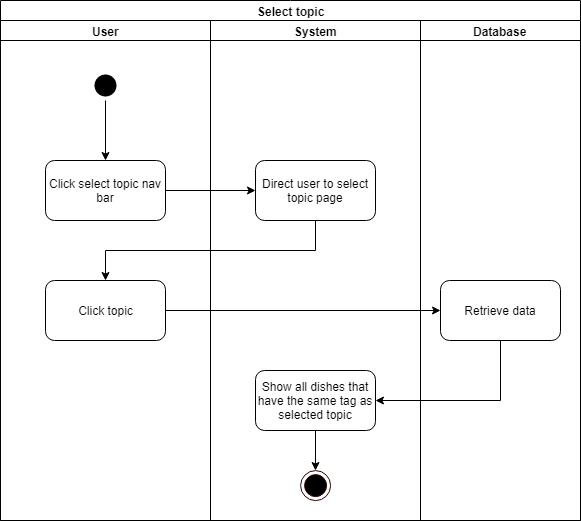
Activity diagram 13: AC13



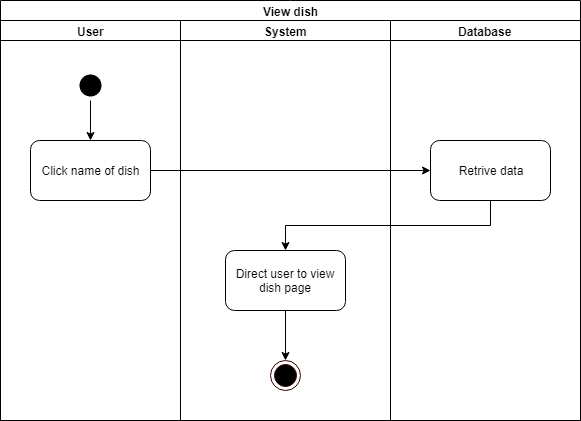
Activity diagram 14: AC14



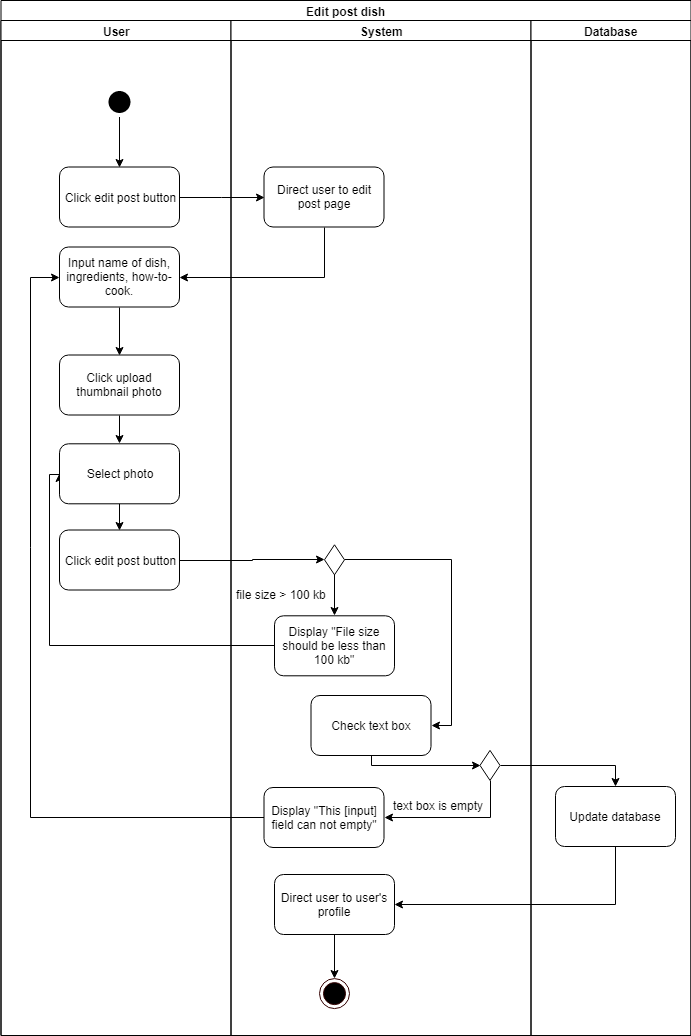
Activity diagram 15: AC15



Activity diagram 16: AC16



Activity diagram 17: AC17



Activity diagram 18: AC18

