A Requirement Analysis Report on:

**Cookbook Network**

Created By:

Aqib Shah

Erico Oyarzun

Anais Hernandez

CEN 4010: Software Engineering I

Spring 2015

Table of Contents

Section 1: Introduction 3

1.1 Objective 3

Section 2: Proposed System 4

2.1 Overview 4

2.2 Functional requirements 4

2.3 Nonfunctional requirements 5

2.4 Constraints (“Pseudo requirements”) 5

2.5 System models 6

2.5.1 Use case Model 6

2.5.2 Object Model 24

2.5.3 Dynamic Models 25

# Section 1: Introduction

This report aims to discuss all the features of the Cookbook Network. The first part of this report will discuss the objective of the Cookbook Network. The second section of this report, Proposed System, will discuss the functionalities of the system. The functional requirements will list what the system can do, the non-functional requirements will demonstrate the usability, reliability, and performance of the system, and the constraints of the system will be discussed. This report will provide system models, such as the use case model, the class diagram, and the sequence diagram, showing the sequence of events that can be performed for each use case.

## 1.1 Objective

The Cookbook Network is a virtual recipe book where users will be allowed to share recipes with one another and create their own cookbooks with their favorite recipes. The objective of this application is to have users post recipes, and allow users to search and find recipes they are interested in a quick and simple manner. Each recipe will consist of an image, ingredients, steps, ratings or flags, and short tags. These short tags will describe the type of food that is being made in the recipe, making searching for recipes easier and precise for the user. The Cookbook Network will help users keep track of their favorite recipes by adding it to their cookbook, instead of having to search for it each time. A diverse group of users will be able to enjoy the site, sharing and discovering new recipes from different parts of the world.

# Section 2: Proposed System

## 2.1 Overview

The Cookbook Network will consist of three different levels of users: Guest Users, Registered Users, and Administrators. Each recipe will have short tags that describe the recipe, making searching easier and accurate for the user. All users can search through recipes by the title of recipe, tags, or ingredients. When searching, they will be given a list of tags from which they can select and will display the recipes that most comply with the user’s search. Users can select a recipe and view the ingredients, steps, images, and its ratings. A registered user can rate a recipe or flag it, if it does not comply with system policies by a registered user. Multiple recipes can be grouped together by a registered user to create a cookbook. Registered users can set the privacy of each cookbook, as well as each recipe, restricting its access to other users. Guest users will only be allowed to view recipes and cookbooks that are set as public. The system will invite all guest users to register an account. Registered users can upload recipes to the system and cookbooks. Registered users can view their own recipes, and recipes marked as public and registered. Recipes set as friendly will be restricted to those accounts that the user lists as viewable. Administrators are a type of registered user having the same capabilities, in addition to a few others. Administrators are able to view all recipes, delete users from the system, search and browse through accounts, and delete other users’ recipes.

## 2.2 Functional requirements

The Cookbook Network will be designed to fulfill the following functional requirements:

1. System allows all guest users to register and create an account.
2. System allows registered users to create, delete, and edit their recipes.
3. System allows registered users to create, delete, and edit cookbooks.
4. System allows registered users to add recipes to their cookbooks.
5. System allows all users to search for and access public recipes based on title, ingredients, and tags.
6. System allows all users to search for and access public cookbooks based on title and tags.
7. Registered users can make their cookbooks and recipes to public, registered, friendly, or private.
8. Registered users may rate recipes available to them.
9. Registered users may flag recipes that do not comply with system policies.
10. System allows administrators to delete recipes and remove irrelevant tags.
11. System allows administrators to search and browse through accounts and delete users.
12. System preserves the registered users' preferences, information, and recipes.

## 2.3 Nonfunctional requirements

The following is a list of nonfunctional requirements that the Cookbook Network will fulfill:

1. Searching for recipes returns a results page within 2 seconds.
2. Newly created recipes should be available to all users, including the creator, within 1 second.
3. Recipes should not be available to any user immediately after it has been deleted.
4. Users should be able to upload recipes at any time of the day.
5. The upload time should be relatively the same throughout the entire day.
6. System should be able to handle multiple recipes being uploaded at the same time.
7. Must be compatible with multiple web browsers.

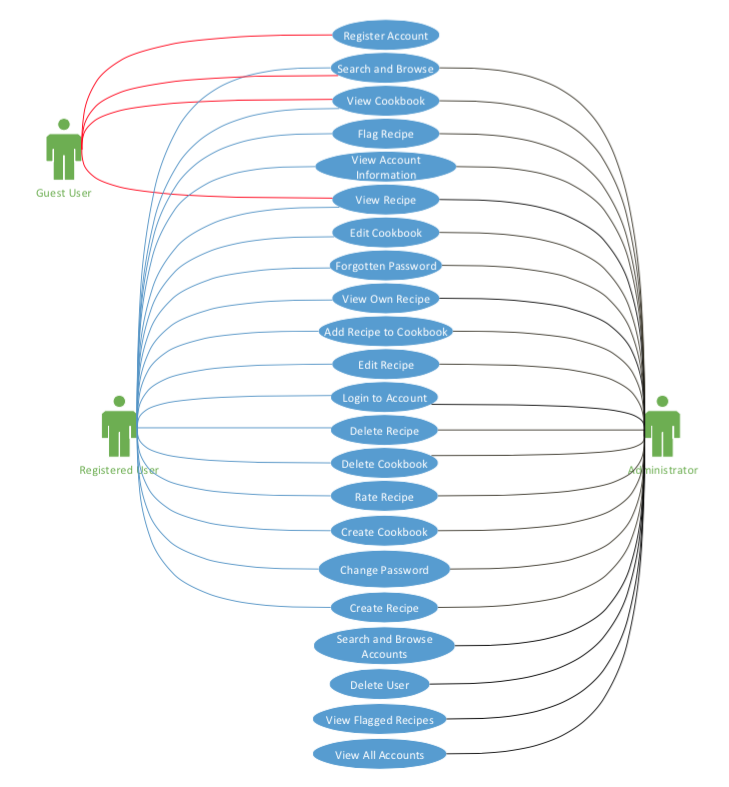
## 2.4 Constraints (“Pseudo requirements”)

1. The Cookbook Network shall be completed by April 17th, 2015.

## 2.5 System Models

The following system will show the use case model, the different use cases of the system, object model, such as class diagram, and object models, such as sequence diagrams. These diagrams will demonstrate the activity of the proposed system.

### 2.5.1 Use Case Model

A use case explains the interaction between the user and the system. The following model shows all the use cases of the system and what each user (Guest, Registered, and Administrator) has access to.

#### 2.5.1.1 Use Cases

The following are a description of events of each use case shown in the use case model.

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Register An Account | **ID**: UC-1 | **Priority**: High |
| **Actor**: Guest User | | |
| **Description**: This use case describes a guest user that is creating an account. | | |
| **Trigger**: Guest user selects sign up link | | |
| **Preconditions**:   1. User is not logged in | | |
| **Normal Course**:   1. Select sign up link    1. System displays registration page    2. Guest user enters an email, password, and submits    3. System displays success page | | |
| **Post conditions**:   1. User is logged in 2. User is taken to their personalized account page 3. Email message is sent to the user | | |
| **Exceptions**:  E1.0: User enters an email that is already in use (occurs at step Normal Course 1.2)   1. System displays message that the email is already used 2. Return to Normal Course 1.2   E2.0: User does not enter valid email or password (occurs at Normal Course step 1.2)   1. System displays message that the email/password are invalid 2. Return to Normal Course 1.2   E3.0: User cancels registration   1. System displays home page   E4.0: User leaves email or password blank (occurs at step Normal Course 1.2)   1. System displays message that email and password cannot be blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Login to Account | **ID**: UC-2 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a user logging into their account. | | |
| **Trigger**: Registered user selects Login link | | |
| **Preconditions**:   1. User is not logged in | | |
| **Normal Course**:   1. Select login link. 2. System displays login page 3. Registered user enters email and password. 4. System displays success page | | |
| **Post conditions**:   1. User is logged in 2. User is taken to their personalized account page | | |
| **Exceptions**:  E1.0: User enters invalid email or password combination (occurs at Normal Course 1.2)   1. System displays message that the login information is incorrect 2. Return to Normal Course 1.2   E2.0: User enters email that is not linked to an account (occurs at Normal course 1.2)   1. System displays message that account does not exist with given email 2. Return to Normal Course 1.2   E3.0: User leaves email or password blank (occurs at Normal Course 1.2)   1. System displays that email or password cannot be blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Change Password | **ID**: UC-3 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is changing their password. | | |
| **Trigger**: Registered user selects change password link | | |
| **Preconditions**:   1. User is logged in | | |
| **Normal Course**:   1. Select change password link 2. System displays change password page 3. User enters password information (e.g., current password and new password) 4. System displays confirmation page 5. User selects confirm 6. System displays success page | | |
| **Post conditions**:   1. User is taken to their personalized account page | | |
| **Exceptions**:  E1.0: User enters invalid current password (occurs at Normal Course 1.2)   1. System displays invalid password message 2. Return to Normal Course 1.2   E2.0: User enters invalid new password (occurs at Normal Course 1.2)   1. System displays invalid new password message 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Forgotten Password | **ID**: UC-4 | **Priority**: Medium |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered using that forgot their password. | | |
| **Trigger**: User selects forgot password link | | |
| **Preconditions**:   1. User already has a registered account | | |
| **Normal Course**:   1. Select forgotten password link 2. System displays forgotten password page 3. User enters email 4. System displays message sent notification | | |
| **Post conditions**:   1. Email message is sent to the user with a randomized password | | |
| **Exceptions**:  E1.0: User enters an email that is not registered (occurs at Normal Course 1.2)   1. System displays a message notifying that no account exists with the given email address 2. Return to Normal Course 1.2   E2.0: User enters invalid email address (e.g., random text) (occurs at Normal Course 1.2)   1. System displays a message that the email is invalid 2. System request user to re-enter email address | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Create recipe | **ID**: UC-5 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user creating a new recipe. | | |
| **Trigger**: User selects add new recipe link | | |
| **Preconditions**:   1. User is logged in | | |
| **Normal Course**:   1. User selects link to add new recipe. 2. System displays New Recipe page 3. User enters recipe information 4. System displays recipe (Use case – View Recipe) | | |
| **Post conditions**:   1. Recipe is added to database. 2. User is taken to the webpage containing the new added recipe. | | |
| **Exceptions**:  E1.0: User clicks cancel (occurs at Normal Course 1.2)   1. System displays a message asking the user if they are sure they want to leave page. 2. System takes user back to home page.   E2.0: Recipe information is left blank (occurs at Normal Course 1.2)   1. System displays message that a recipe information cannot be blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Flag Recipe | **ID**: UC-6 | **Priority**: Medium |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is flagging a recipe | | |
| **Trigger**: Registered user selects flag recipe link | | |
| **Preconditions**:   1. User is logged in 2. User is viewing a recipe | | |
| **Normal Course**:   1. Select flag recipe link 2. System displays flag recipe page 3. User enters flag information (reason and comment) 4. System displays success page | | |
| **Post conditions**:   1. Recipe is set as flagged | | |
| **Exceptions**:  E1.0: User cancels flag (occurs at Normal Course 1.2)   1. System displays previously visited page   E2.0: User enters blank flag information (occurs at Normal Course 1.2)   1. System displays empty flag information message 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Delete Recipe | **ID**: UC-7 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is deleting a recipe. | | |
| **Trigger**: User selects delete recipe link | | |
| **Preconditions**:   1. User is logged in 2. User has selected one or more recipes 3. User is the owner of the selected recipes | | |
| **Normal Course**:   1. Select delete recipe link 2. System displays deletion confirmation page 3. User selects confirm 4. System displays success page | | |
| **Post conditions**:   1. Recipe is removed from user’s account and all cookbooks it is a part of | | |
| **Exceptions**:  E1.0: User cancels deletion of recipe (occurs at Normal Course 1.2)   1. User selects cancel on confirmation page 2. System displays previous page | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Create Cookbook | **ID**: UC-8 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is creating a cookbook. | | |
| **Trigger**: User selects create a cookbook link | | |
| **Preconditions**:   1. User is logged into their account | | |
| **Normal Course**:   1. User selects create cookbook link 2. System displays New Cookbook page 3. User enters cookbook information 4. Systems displays new cookbook | | |
| **Alternate Course:**   1. User selects recipe to add to a cookbook. 2. User clicks create new cookbook link 3. User enters cookbook information 4. System displays success message | | |
| **Post conditions**:   1. New cookbook is added to user’s account. 2. System takes user to their new cookbook. | | |
| **Exceptions**:  E1.0: User leaves cookbook information blank (occurs at Normal Course 1.2)   1. System displays that cookbook information cannot be left blank 2. Return to Normal Course 1.2   E2.0: User selects cancel (occurs at Normal Course 1.2)   1. System displays cancel confirmation page 2. User selects confirm 3. System displays previously visited page | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Edit Cookbook | **ID**: UC-9 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is editing a cookbook. | | |
| **Trigger**: User selects edit cookbook link | | |
| **Preconditions**:   1. User is logged in 2. User has selected one cookbook 3. User is the owner of the selected cookbook | | |
| **Normal Course**:   1. Select edit cookbook link 2. System displays edit cookbook page 3. User makes desired changes 4. System displays success page | | |
| **Post conditions**:   1. Cookbook is updated | | |
| **Exceptions**:  E1.0: User cancels editing of cookbook (occurs at Normal Course 1.2)   1. User selects cancel on confirmation page 2. System displays previously visited page   E2.0: Cookbook information is left blank (occurs at Normal Course 1.2)   1. System displays cookbook information cannot be left blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Delete Cookbook | **ID**: UC-10 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is deleting a cookbook. | | |
| **Trigger**: User selects delete cookbook link | | |
| **Preconditions**:   1. User is logged in 2. User has selected one or more cookbooks 3. User is the owner of the selected cookbooks | | |
| **Normal Course**:   1. Select delete cookbook link 2. System displays deletion confirmation page 3. User selects confirm 4. System displays success page | | |
| **Post conditions**:   1. Cookbook is removed from user’s account | | |
| **Exceptions**:  E1.0: User cancels deletion of cookbook (occurs at Normal Course 1.2)   1. User selects cancel on confirmation page 2. System displays previously visited page | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Add Recipe to Cookbook | **ID**: UC-11 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is adding a recipe to cookbook | | |
| **Trigger**: User selects add recipe to cookbook link | | |
| **Preconditions**:   1. User is logged in 2. User has selected one or more recipes to add | | |
| **Normal Course**:   1. User selects add recipe to cookbook link on a recipe 2. System displays cookbooks owned 3. User selects cookbook where recipe(s) will be added 4. System displays success message | | |
| **Post conditions**:   1. Recipe is added to user’s cookbook | | |
| **Exceptions**:  E1.0: User cancels deletion of cookbook (occurs at step Normal Course 1.2)   1. User selects cancel on confirmation page 2. System displays previously visited page. | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Search and Browse | **ID**: UC-12 | **Priority**: High |
| **Actor**: Guest Users, Registered Users | | |
| **Description**: This use case describes a registered user that is searching or browsing through recipes and cookbooks. | | |
| **Trigger**: User selects search link | | |
| **Normal Course**:   1. User selects search link 2. System displays search page 3. User enters search criteria 4. System displays matching recipes or cookbooks | | |
| **Exceptions:**  E1.0: No recipes match search criteria (occurs at Normal Course 1.2)   1. System displays no results found message 2. Return to Normal Course 1.2   E2.0: User enters blank search (occurs at Normal Course 1.2)   1. System displays message that search criteria must not be left blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Edit Recipe | **ID**: UC-13 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is editing a recipe. | | |
| **Trigger**: User selects edit recipe link | | |
| **Preconditions**:   1. User is logged in 2. User has selected one recipe 3. User is the owner of one or more recipe | | |
| **Normal Course**:   1. Select edit recipe link 2. System display recipe editing screen 3. User makes desired changes 4. System displays updated recipe | | |
| **Post conditions**:   1. Recipe information is updated 2. User is taken back to the recipe page | | |
| **Exceptions**:  E1.0: User cancels editing the recipe (occurs at Normal Course 1.2)   1. User selects cancel 2. System asks user to confirm cancellation 3. All changes are discarded if user selects confirm; otherwise return to Normal Course 1.2   E2.0: Recipe information is left blank (occurs at Normal Course 1.2)   1. System displays message that a recipe information cannot be blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Delete User | **ID**: UC-14 | **Priority**: Medium |
| **Actor**: Administrator | | |
| **Description**: This use case describes an administrator that will delete a user. | | |
| **Trigger**: Administrator selects delete user link | | |
| **Preconditions**:   1. User is logged in as an administrator 2. User is viewing a user’s account information | | |
| **Normal Course**:   1. Administrator selects delete user link 2. System displays deletion user page 3. Administrator enters email of user to be deleted. 4. System displays deletion confirmation page. 5. User selects confirm 6. System displays success message | | |
| **Post conditions**:   1. User’s account is deleted, as well as all recipes and cookbooks created by user. | | |
| **Exceptions**:  E1: Administrator cancels deletion (occurs at step Normal Course 1.1 or 1.4)   1. Administrator selects cancel. 2. System displays previously visited page | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Rate Recipe | **ID**: UC-15 | **Priority**: Medium |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is rating a recipe | | |
| **Trigger**: Registered user selects rate recipe link | | |
| **Preconditions**:   1. User is logged in 2. User is viewing a recipe | | |
| **Normal Course**:   1. Select rate recipe link 2. System displays rate recipe page 3. User enters desired rating 4. System displays success message | | |
| **Post conditions**:   1. System displays recipe page | | |
| **Exceptions**:  E1.0: User cancels rating (occurs at Normal Course 1.2)   1. System displays previous page   E2.0: User enters blank rating (occurs at Normal Course 1.2)   1. System displays message that rating cannot be blank 2. Return to Normal Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View Recipe | **ID**: UC-16 | **Priority**: High |
| **Actor**: Guest User/Registered User | | |
| **Description**: This use case describes a user who is viewing a recipe. | | |
| **Trigger**: User selects recipe | | |
| **Normal Course**:   1. Select recipe 2. System displays recipe information | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View Cookbook | **ID**: UC-17 | **Priority**: High |
| **Actor**: Guest User/Registered User | | |
| **Description**: This use case describes a user who is viewing a cookbook. | | |
| **Trigger**: User selects cookbook | | |
| **Normal Course**:   1. Select cookbook 2. System displays cookbook | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View Own Recipe | **ID**: UC-18 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is viewing their own recipes | | |
| **Trigger**: User selects “My Cookbook” | | |
| **Preconditions**:   1. User is logged in | | |
| **Normal Course**:   1. Select “My Cookbook” 2. System display all owned recipes 3. User selects specific recipe 4. System displays recipe (Use case – View Recipe) | | |
| **Alternate Course**:   1. User searches/browses for recipes 2. User selects own recipe 3. System displays recipe | | |
| **Exceptions**:  E1.0: User does not own any recipes (occurs at Normal Course step 1.0)   1. System displays message that no results were found | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View All Accounts | **ID**: UC-19 | **Priority**: High |
| **Actor**: Administrator | | |
| **Description**: This use case describes an administrator is viewing all registered users | | |
| **Trigger**: User selects “View All Accounts” | | |
| **Preconditions**:   1. User is logged in 2. User is an administrator | | |
| **Normal Course**:   1. Select “View All Accounts” 2. System displays all registered users | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View Account Information | **ID**: UC-20 | **Priority**: High |
| **Actor**: Registered User | | |
| **Description**: This use case describes a registered user that is viewing an account’s information | | |
| **Trigger**: User selects “View Profile” | | |
| **Preconditions**:   1. User is logged in | | |
| **Normal Course**:   1. Select “View Profile” 2. System displays account information | | |
| **Alternate Course**:   1. User is an administrator 2. User performs View All Accounts 3. User selects an account 4. Return to Normal Course 1.1 5. User is an administrator 6. User performs Search/Browse Accounts 7. Continue to Alternate Course 1.2 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name**: View Flagged Recipe | **ID**: UC-21 | **Priority**: High |
| **Actor**: Administrator | | |
| **Description**: This use case describes an administrator that is viewing flagged recipes | | |
| **Trigger**: User selects “View Flagged Recipes” | | |
| **Preconditions**:   1. User is logged in 2. User is an administrator | | |
| **Normal Course**:   1. Select View Flagged Recipes 2. System displays flagged recipes page | | |

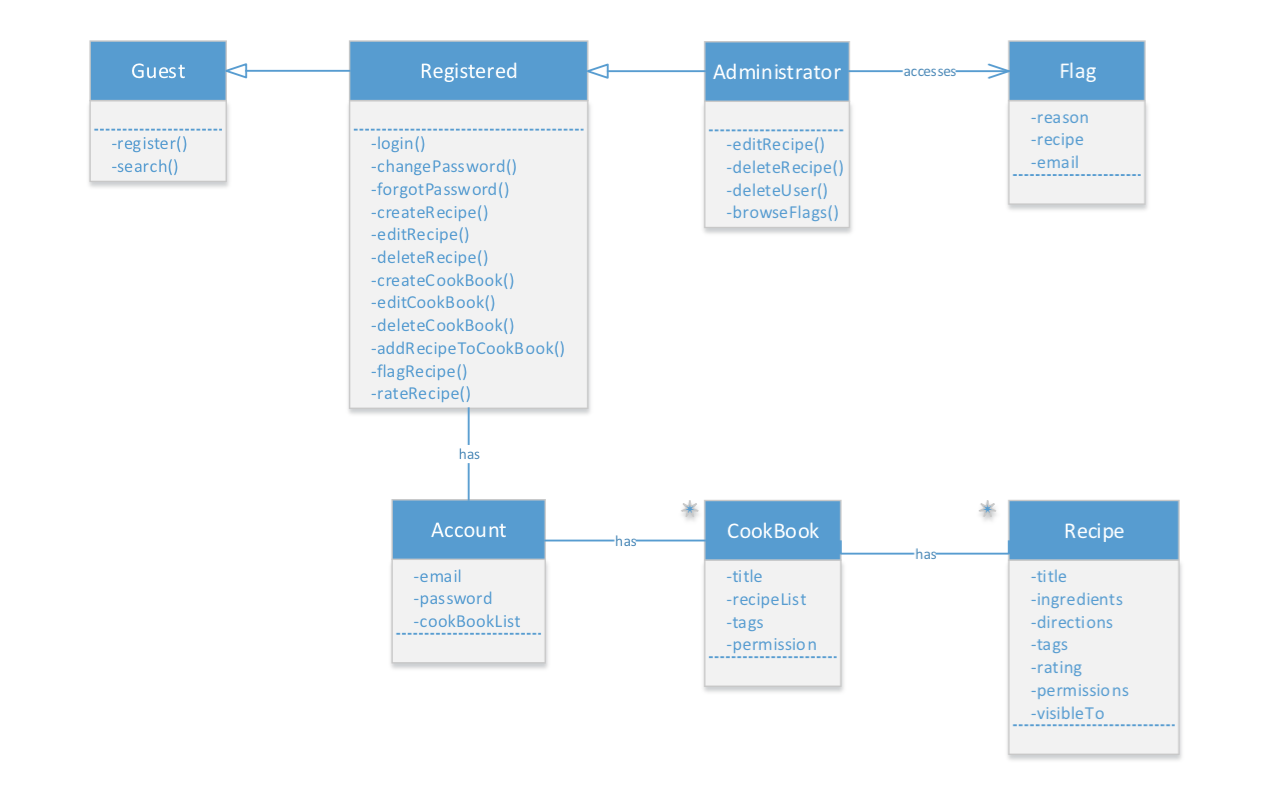
|  |  |  |
| --- | --- | --- |
| **Use Case Name**: Search/Browse Accounts | **ID**: UC-22 | **Priority**: High |
| **Actor**: Administrator | | |
| **Description**: This use case describes an administrator that is searching for a user | | |
| **Trigger**: User select search accounts link | | |
| **Preconditions**:   1. User is logged in 2. User is an administrator | | |
| **Normal Course**:   1. Select “Search Accounts” 2. System displays search accounts page 3. User enters an email 4. System displays user’s information | | |
| **Alternate Course**:   1. User enters invalid email 2. System displays invalid email message 3. Return to Normal Course 1.2 | | |

### 2.5.2 Object Model

Object models show the classes, attributes, and methods of a system, and their relationships.

#### 2.5.2.2 Class Diagram

The following is the class diagram of the Cookbook Network, which shows seven classes and their relationships.

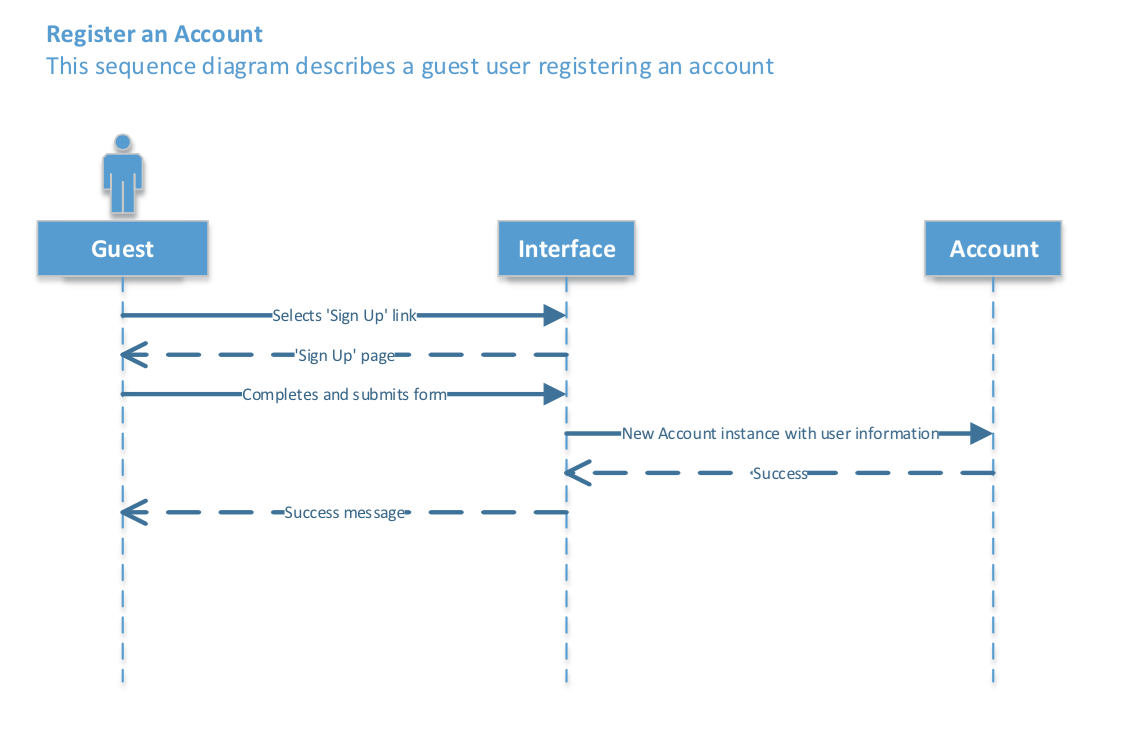


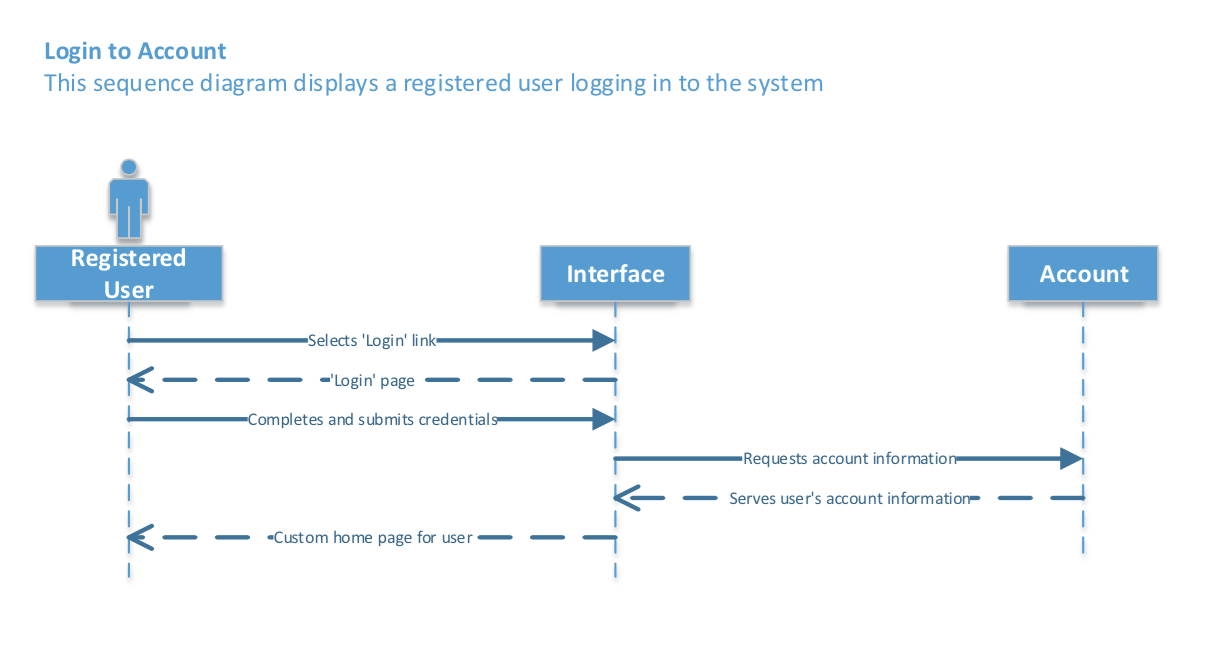
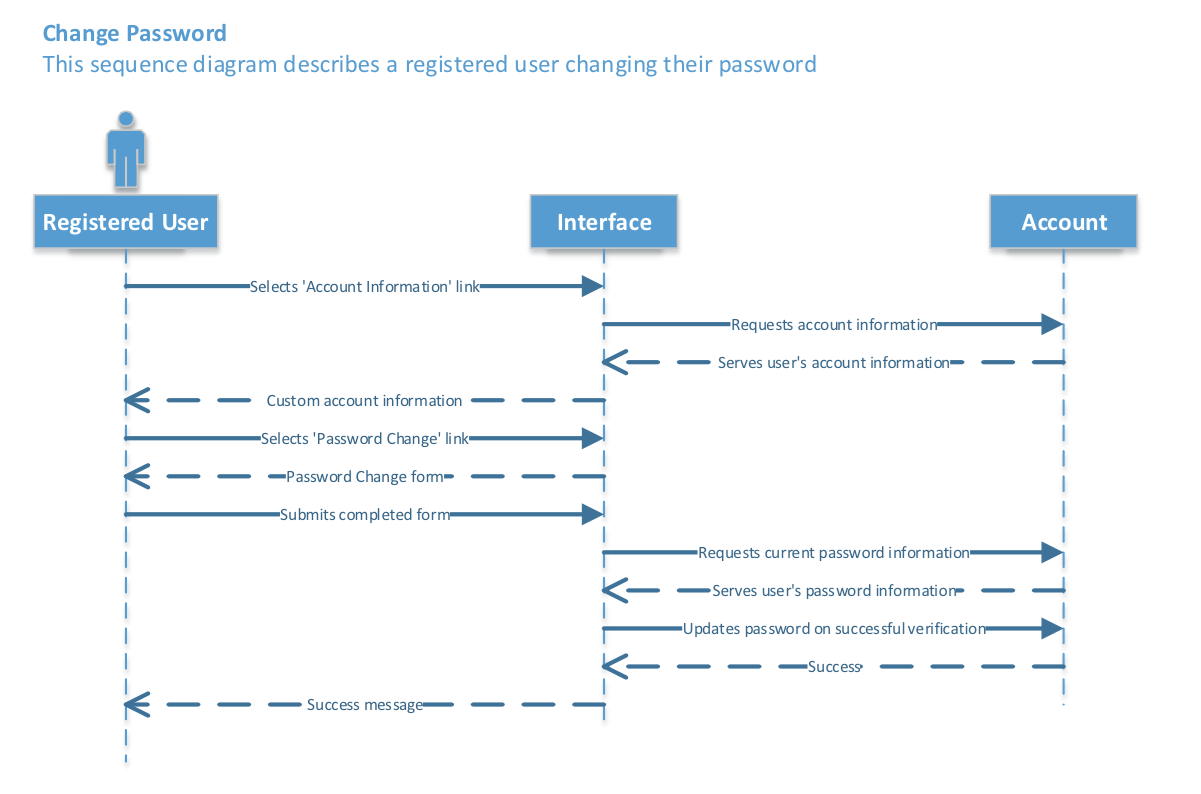
### 2.5.3 Dynamic Models

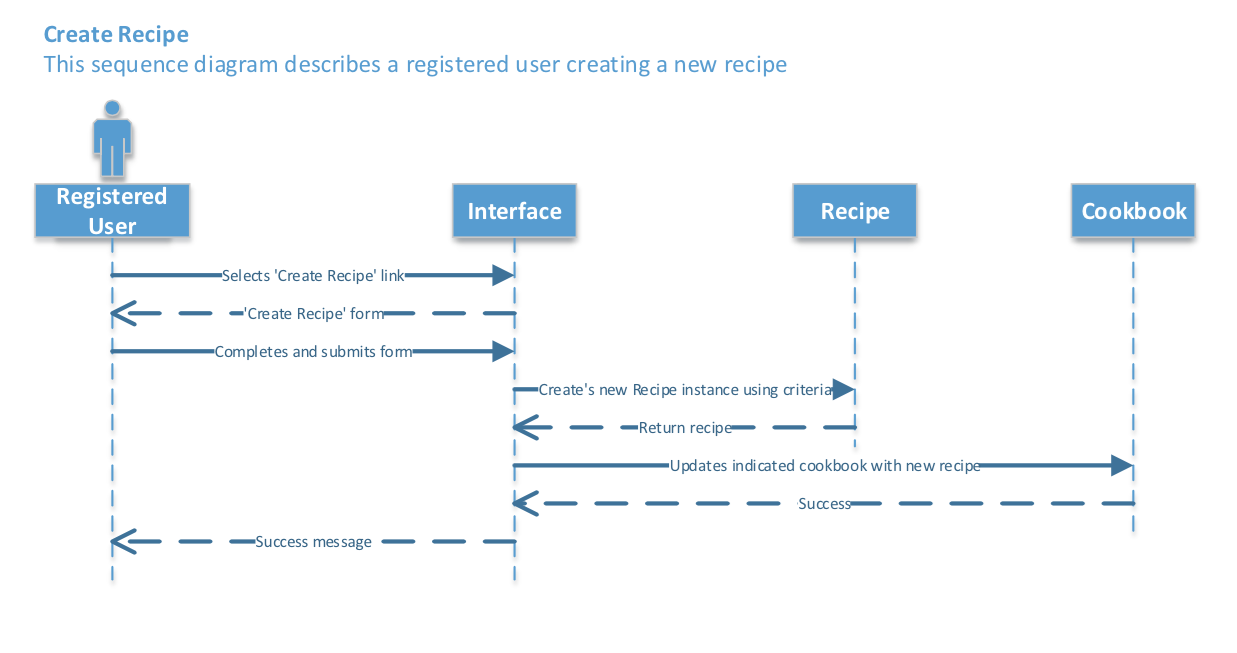
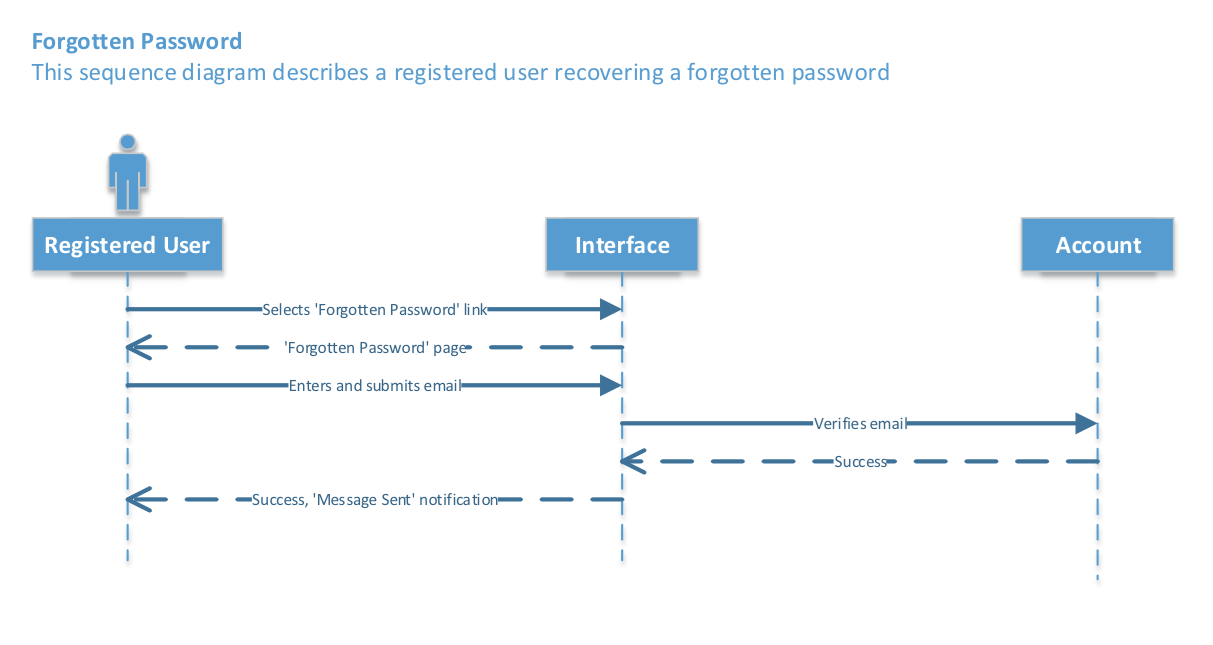
Dynamic models demonstrate the behavior of the system. One type of dynamic model is a sequence diagram, which shows the interaction between objects in each use case.

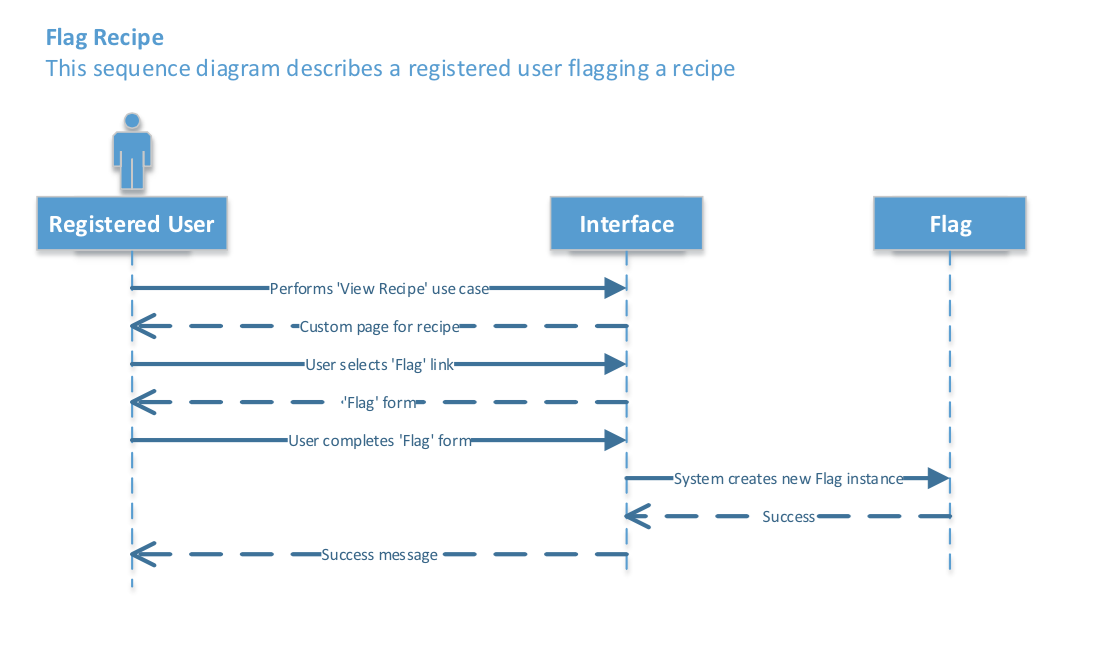
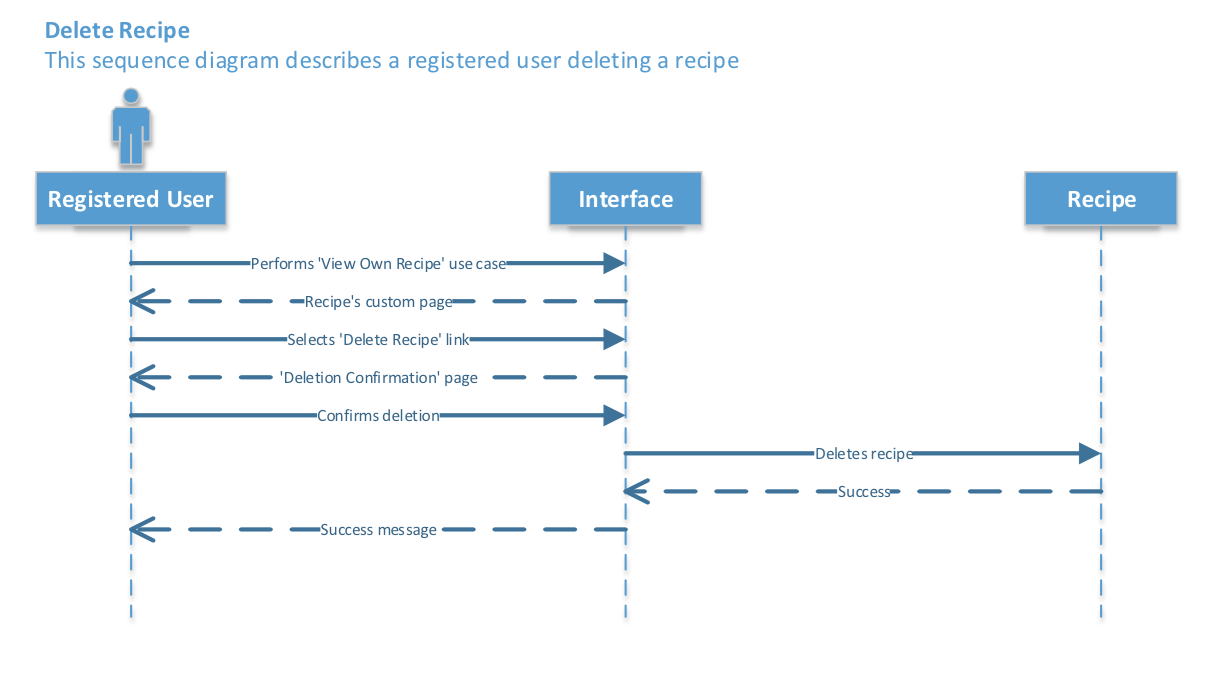
#### 2.5.3.1 Sequence Diagrams

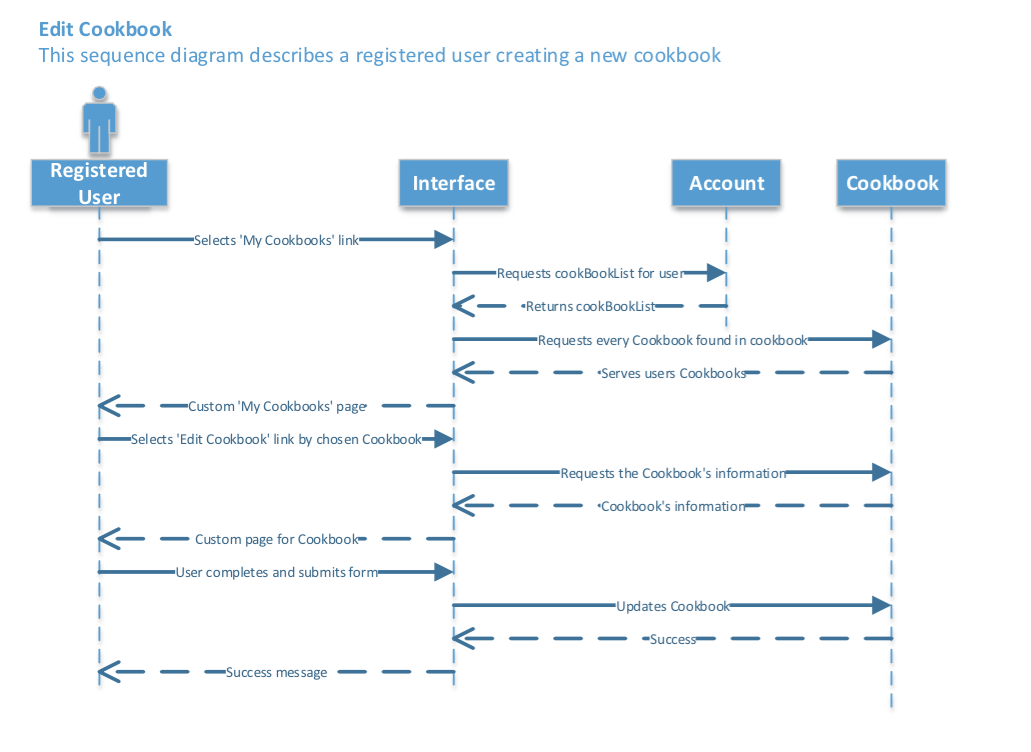
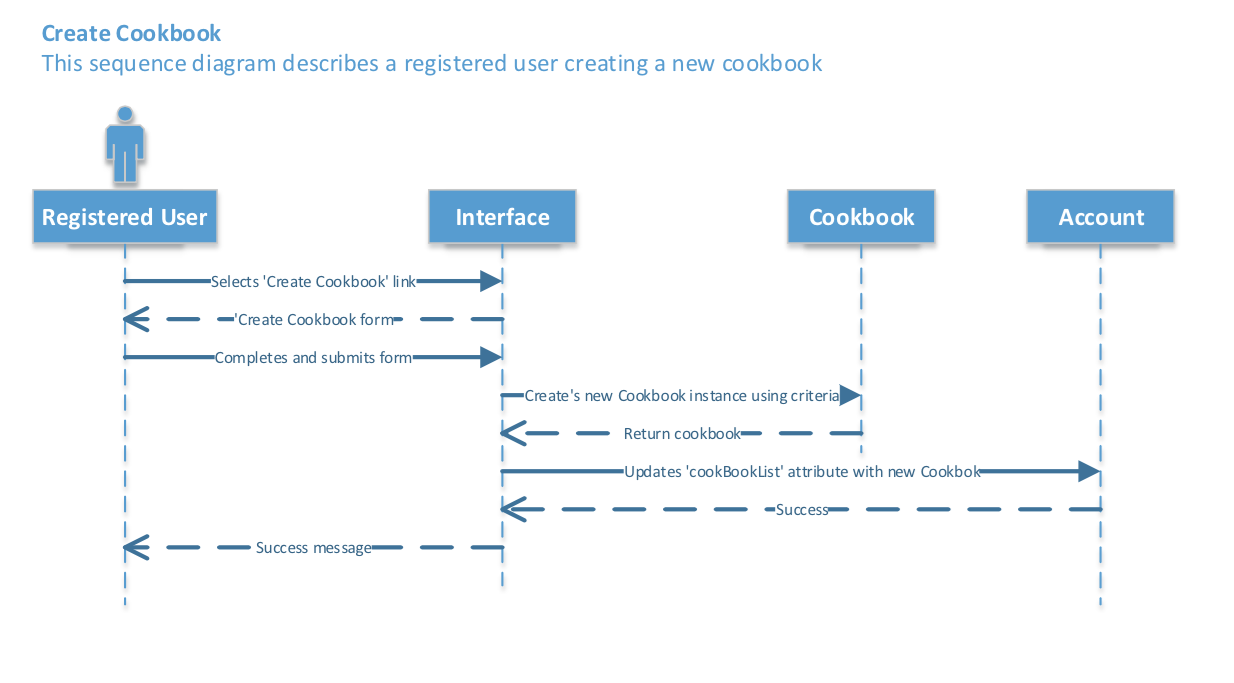
The following are the sequence diagrams of the system. Each sequence diagram corresponds to its use case.

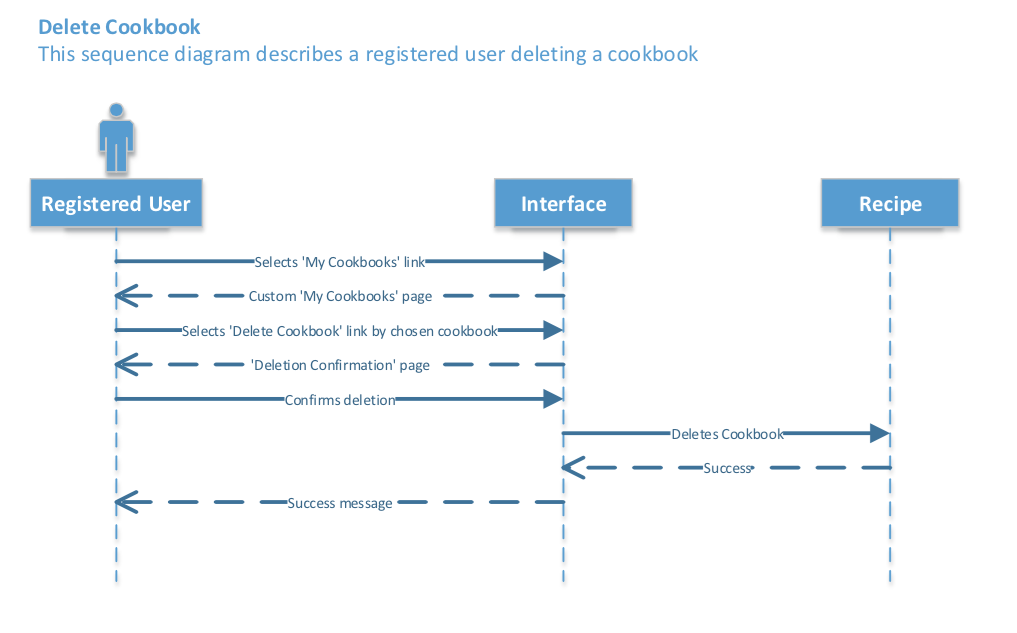
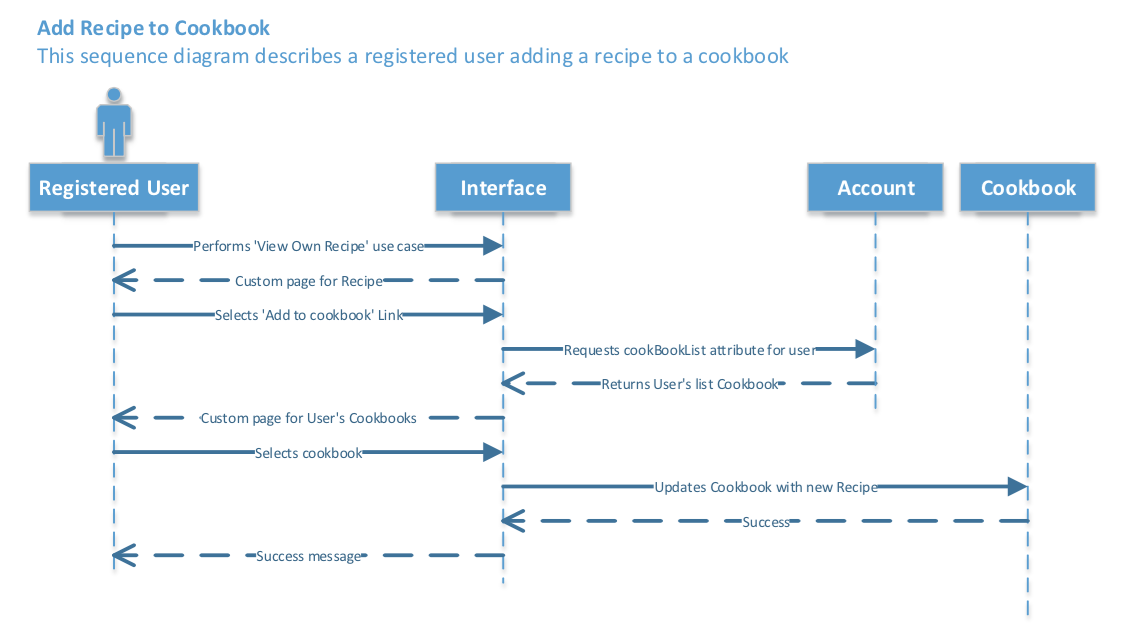


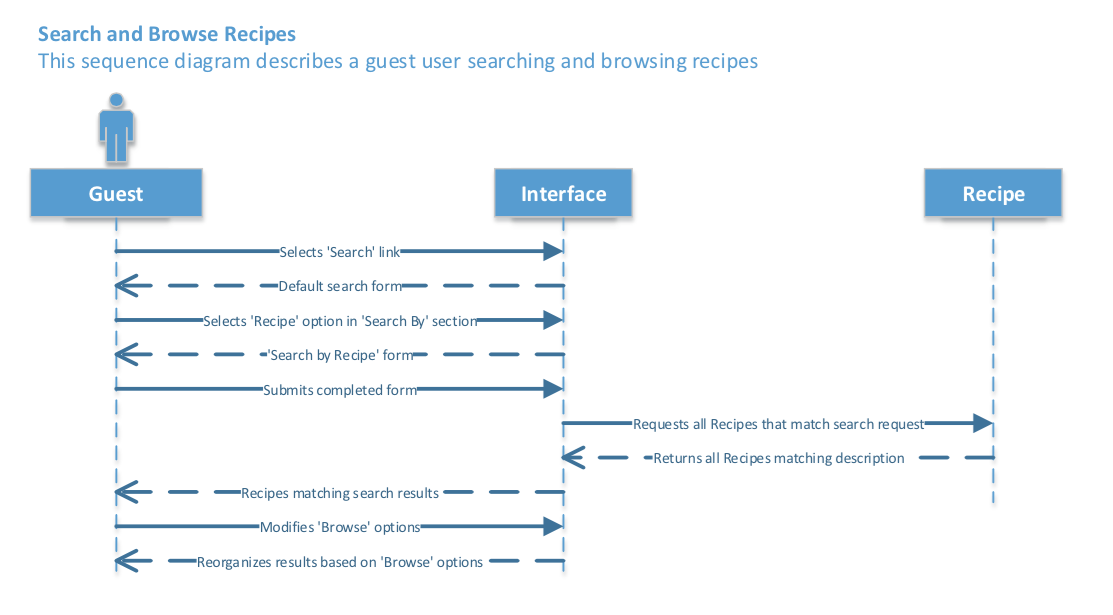
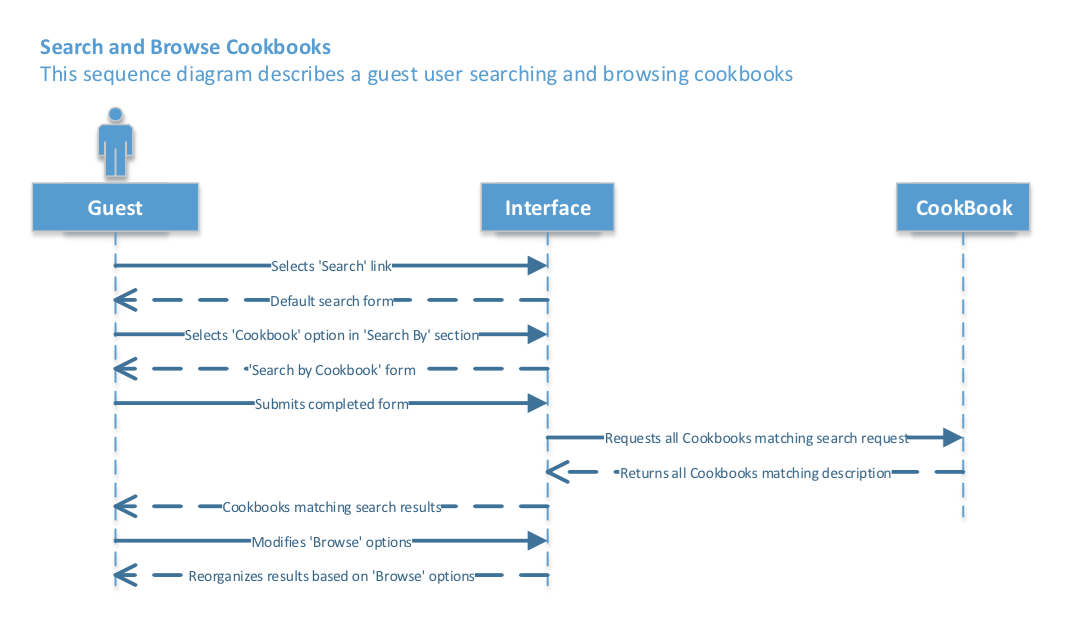


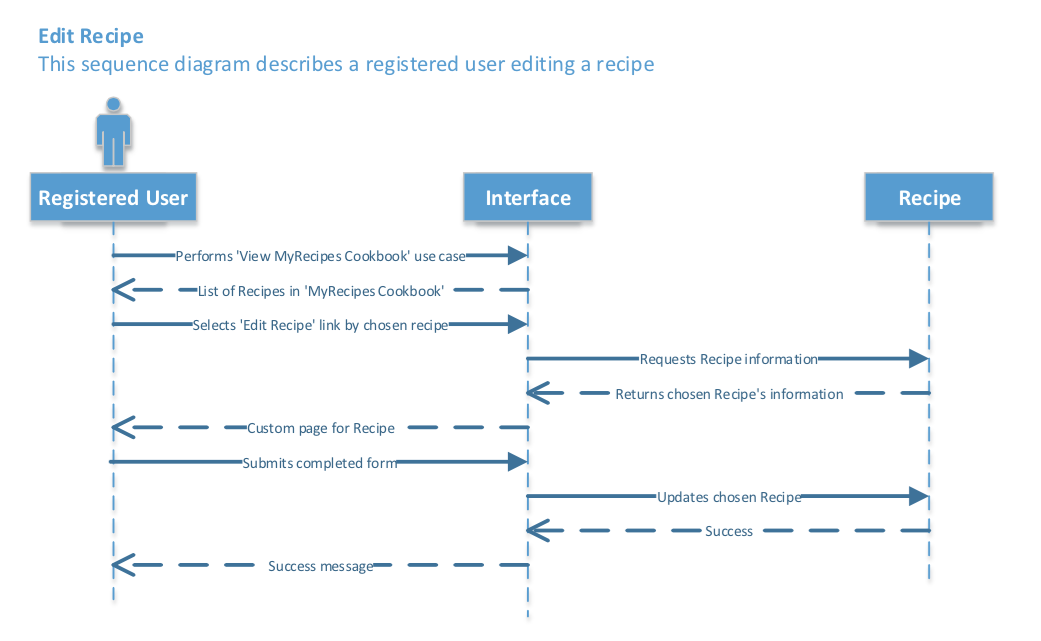
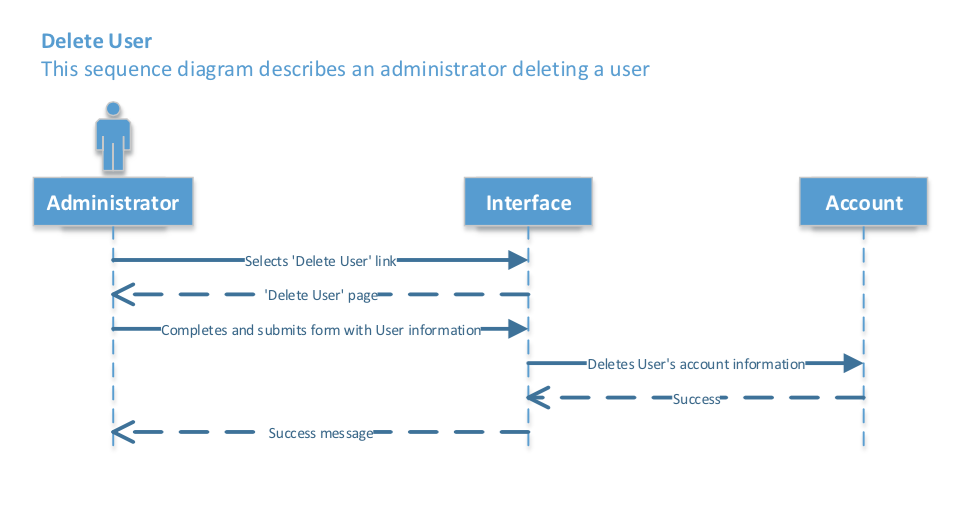


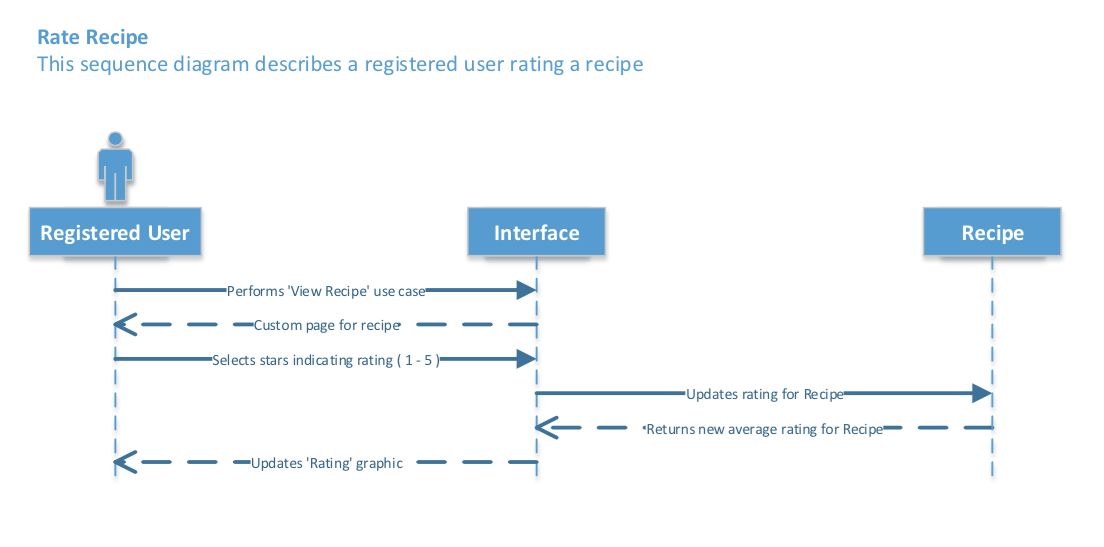
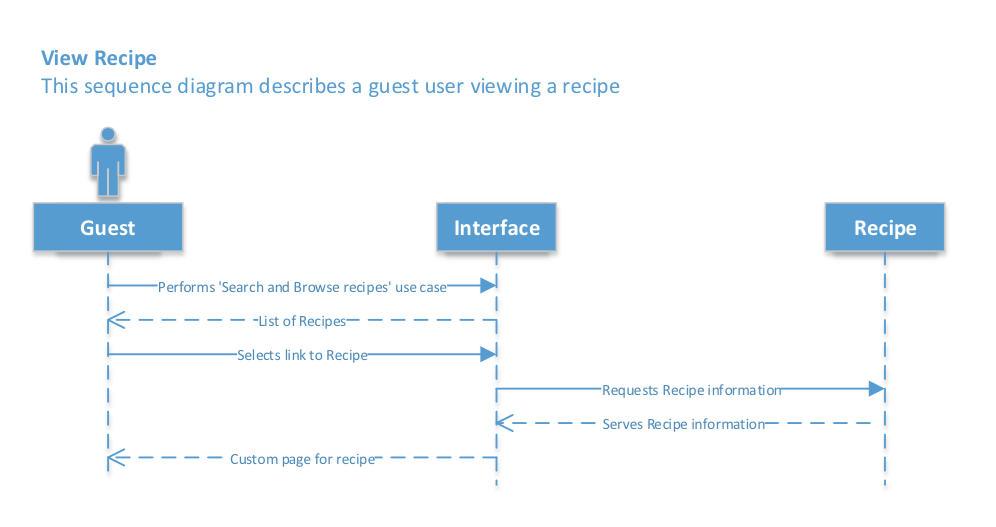
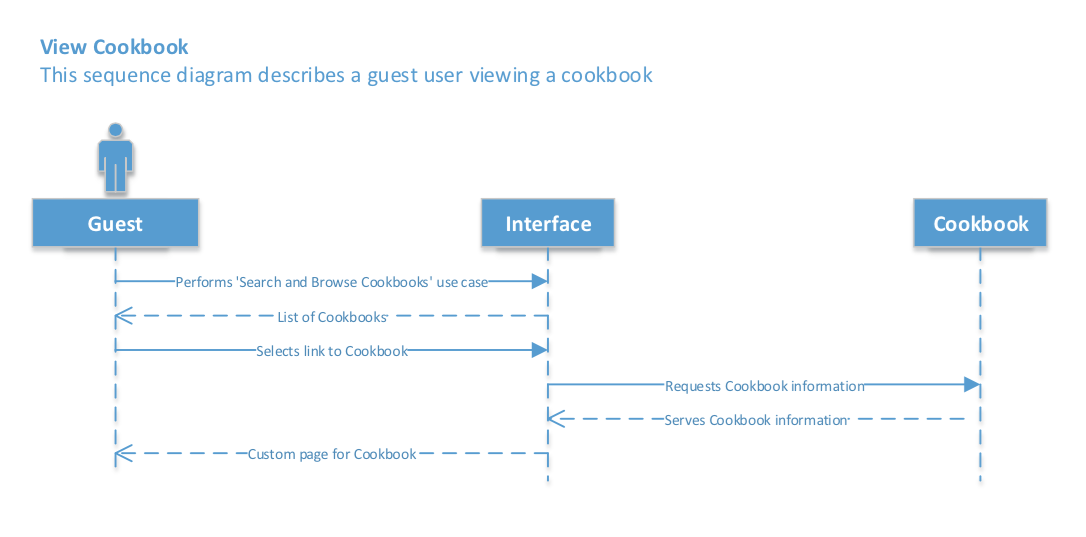


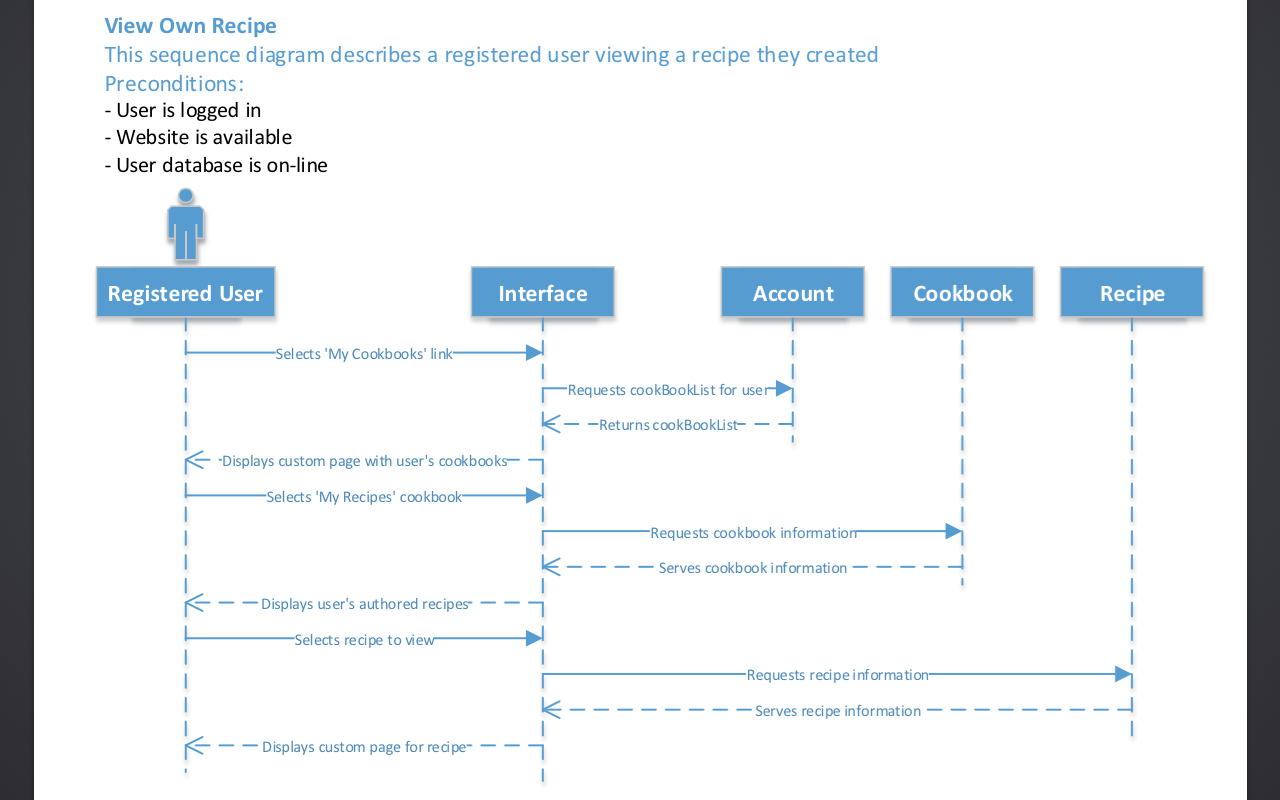


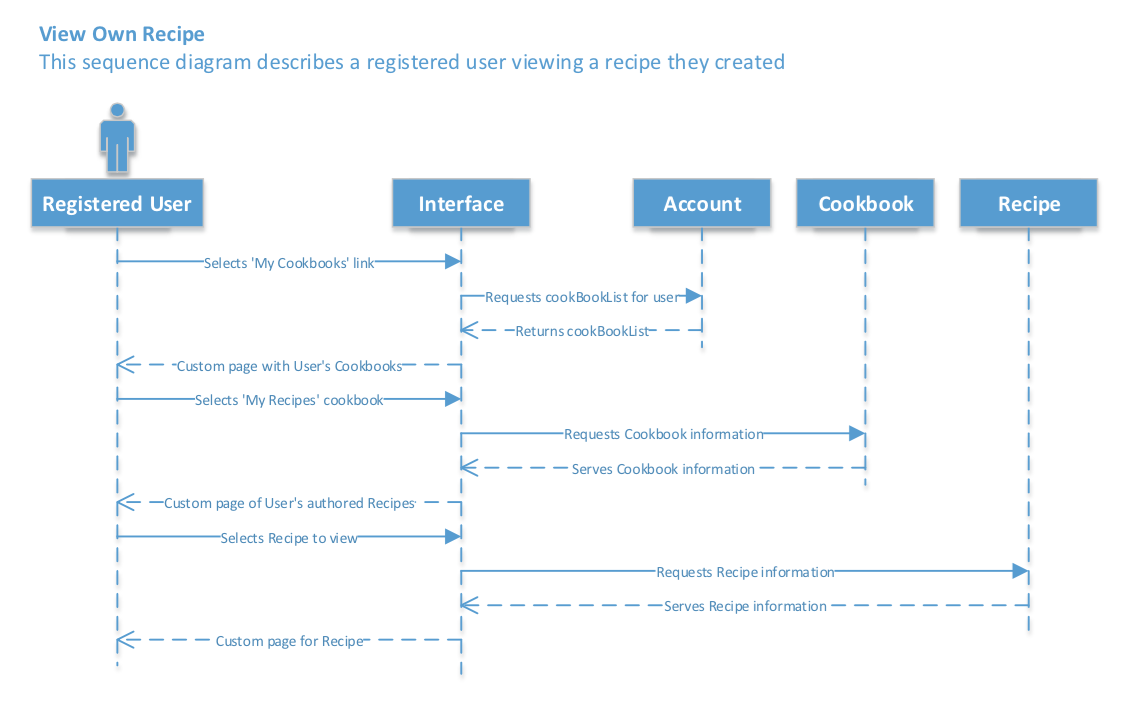
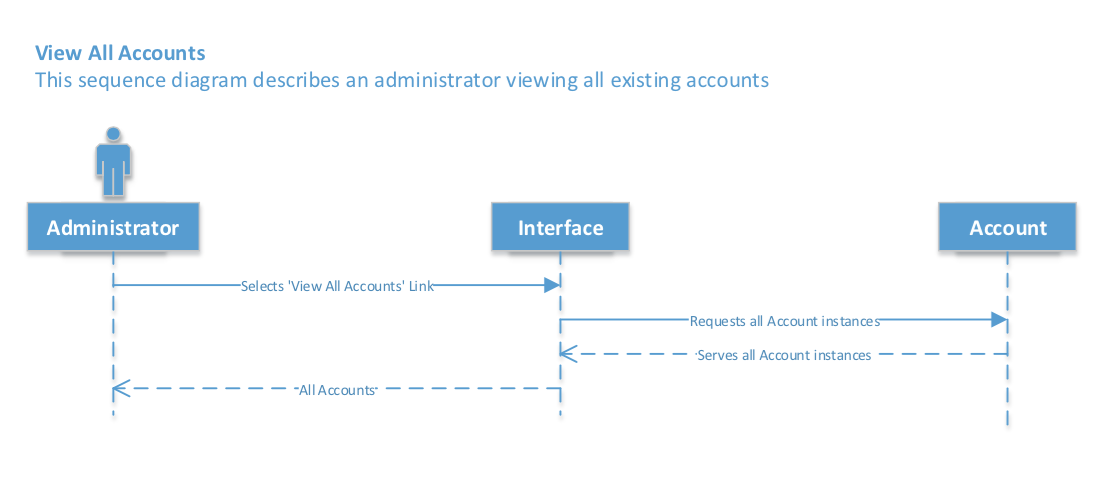


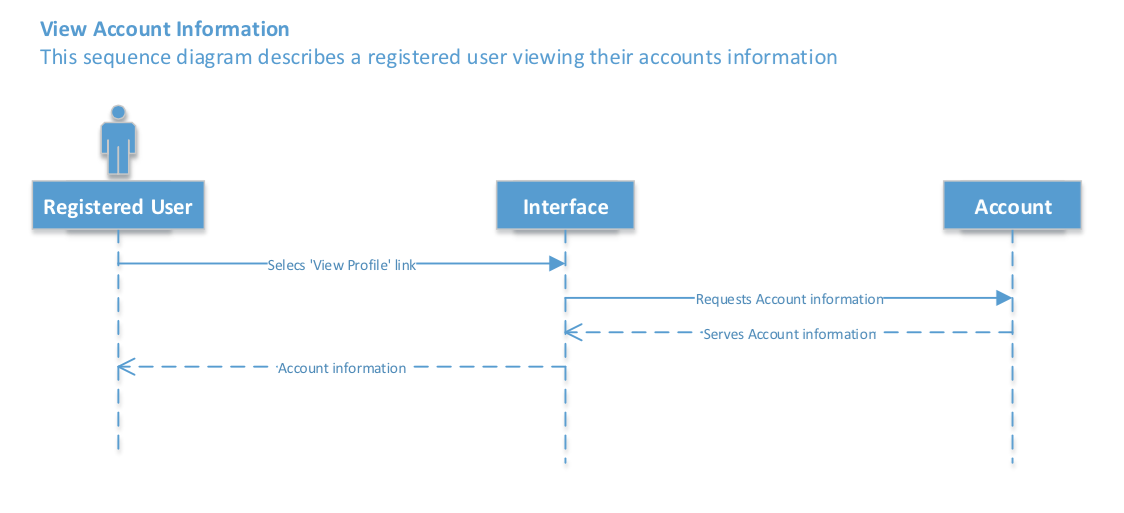
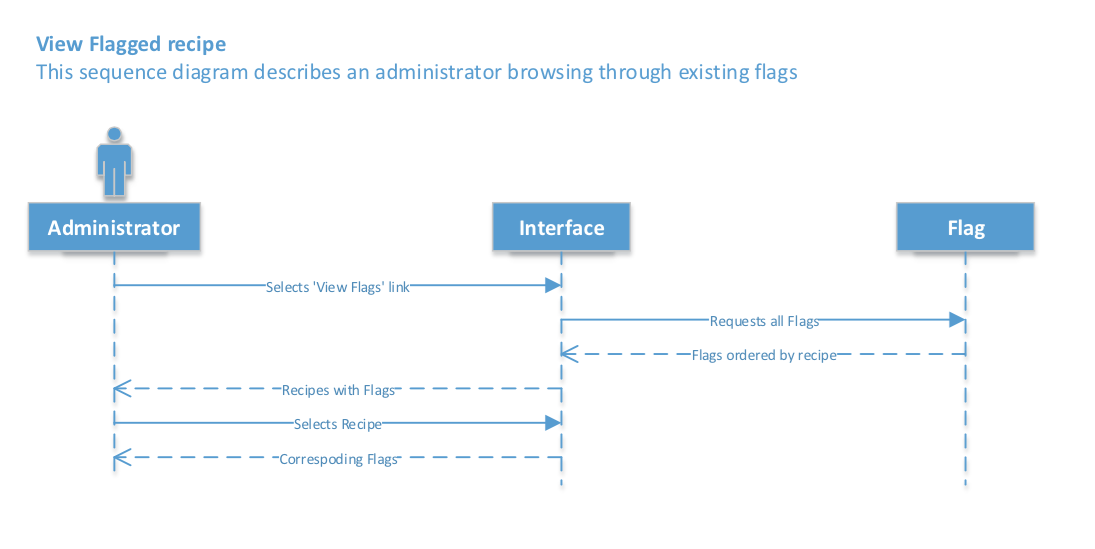












# 