OBJECT ORIENTED SOFTWARE DEVELOPMENT PROJECT

PROJECT TITLE

COOKING RECIPE SYSTEM

NAME: RAFIKA SULTANA NAME: MAHDI HASAN		ID:151-35-1050 ID:151-35-1024
Supervisor Name	:	Md Alamgir Kabir
Department:	:	Software Engineering
Faculty of Science and Information Technology		

April 2018

Contents

List of figure

List of Tables

Chapter 1

1. Introduction

1.1 About the system

Cooking Recipe is an application based project. It is design for store record like recipe name, ingredients. It holds a collection of recipe picture, recipe name, recipe ingredients. In this system, it suggest recipe name when as user input ingredients name.

1.2 Purpose

Cooking Recipe will have following key goals:

- Provide an easy interface for user.
- Easy to store record.
- Easy to add a new recipe to cooking recipe.
- Easy to edit exiting ingredients about the recipe.
- Easy to delete recipe.

1.3 Scope

Cooking Recipe provides the following functionalities:

- Record recipe
- Add new recipe
- Delete recipe
- Edit recipe
- Suggest recipe

1.4 Vision

In this system, system suggest to user or visitor about the recipe. User or visitor input ingredients item, then the system suggest how many recipe they made.

1.5 Why this system necessary

This is necessary because it suggest recipe when user input ingredients they have. It record many recipe & suggest recipe. If we use this system it record many recipe in a short time.

1.6 Proposed solution

Chapter 2

2. System Analysis

2.1 Actor goal list

2.1.1 Assumptions

In this system, it work recipe information. It store recipe ingredients. It holds a collection of recipe, each recipe recording recipe name, ingredients, pictures.

2.2 Use Case Model

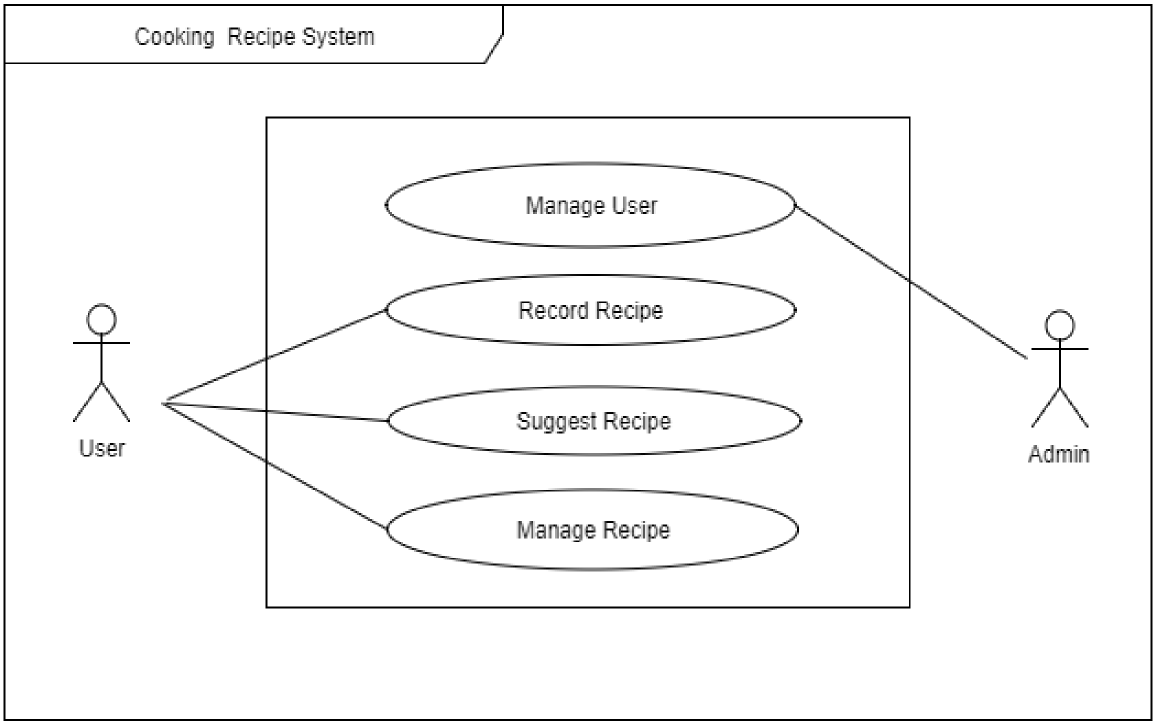

Figure 1: Use case diagram for Cooking Recipe System

2.3 Use Case Description (Brief)

2.3.1 Manage User

In manage user, admin add new person information, edit their information, delete their information and save this information.

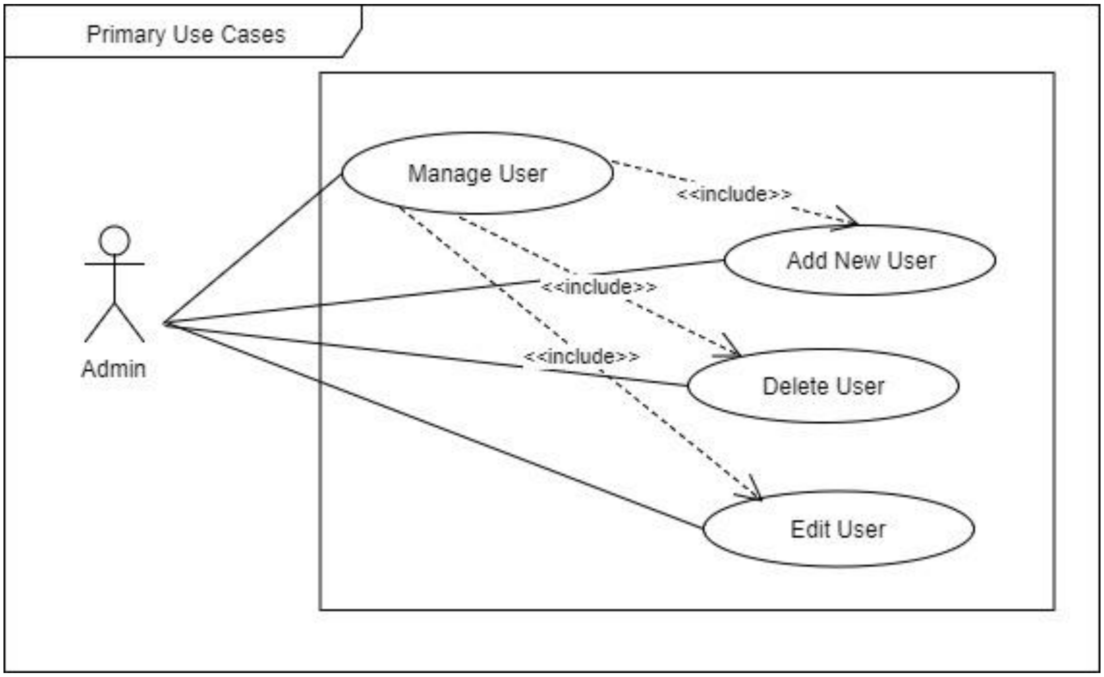

Figure 2: Primary use case for manage user

2.3.2 Record Recipe

In record recipe, at first user log in the system. Then user entry their information. They entry their first name, last name, address, city, state and phone number and save this information

2.3.3 Manage Recipe

In manage recipe, user add new recipe, edit recipe ingredients, delete recipe and save this recipe.

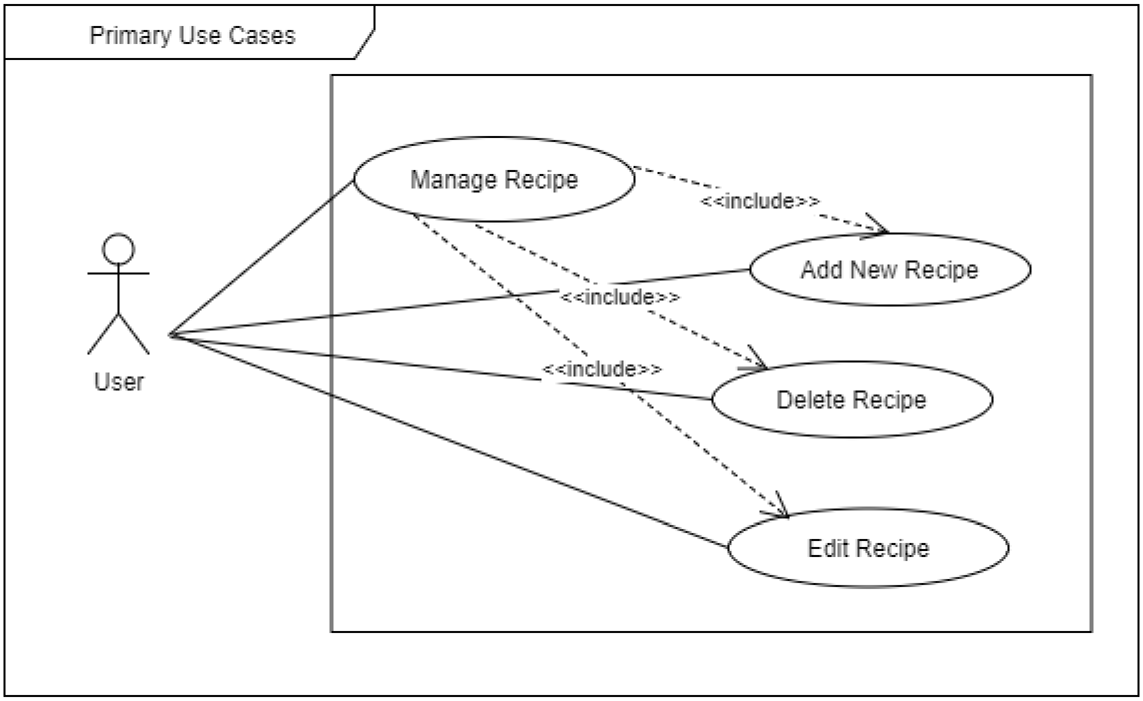

Figure 3: Primary use case for manage recipe

2.3.4 Suggest Recipe

When user input recipe ingredients then its suggest the recipe name those made by that's inputs ingredients.

2.4 Use Case Description

2.4.1 Manage User

Table 1: Description of Use Case 1: Manage User

<table>
<tr><td>Scope</td><td>New user create</td></tr>
<tr><td>Level</td><td>Admin level</td></tr>
<tr><td>Primary Actor</td><td>Admin</td></tr>
<tr><td>Precondition</td><td>Admin must login first</td></tr>
<tr><td>Post condition</td><td>Add new user must be created</td></tr>
<tr><td>Main flow</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Get log in after registration. Get (username, password) for log in in the system.
2. Get add new user after log in. userInfo (first and last name, address, city, phone number).
3. editUserInfo(first and last name, address, city, phone number).
4. deleteUser() from the information.</td><td>1.1. System returns log in successfully.
1.2. System returns entry add new user successfully.
2.1 The system return entry user information successfully.
3.1 Delete user successfully.</td></tr>
</table>
</td></tr>
<tr><td>Alternative scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. editUserInfo(first and last name, address, city, phone number).</td><td>1.1 System returns edit unsuccessfully.</td></tr>
</table>
</td></tr>
</table>

2.4.2 Record Recipe

Table 2: Description of Use Case 2: Record Recipe

<table>
<tr><td>Scope</td><td>Recipe Ingredients Recording</td></tr>
<tr><td>Level</td><td>User level</td></tr>
<tr><td>Primary Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User must login first</td></tr>
<tr><td>Post condition</td><td>Recipe name, ingredients must be recorded</td></tr>
<tr><td>Main flow</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Get log in after registration. Get (user name, password) for log in in system.</td><td>1.3. System returns log in successfully.</td></tr>
<tr><td>2. Get entry recipe name, ingredients after collection or cooked recipe information. recipeItem(items).</td><td>2.1. The system return entry recipe name, ingredients information successfully.</td></tr>
</table>
</td></tr>
<tr><td>Alternative scenario</td><td>
<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Get log in after registration. Get (user name, password) for log in in system.</td><td>1.1. System returns wrong user name.</td></tr>
</table>
</td></tr>
</table>

2.4.3 Manage Recipe

Table 3: Description of Use Case 3: Manage Recipe

<table>
<tr><td>Scope</td><td>Add new cooking recipe</td></tr>
<tr><td>Level</td><td>User level</td></tr>
<tr><td>Primary Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User must login first</td></tr>
<tr><td>Post condition</td><td>Add new recipe must be created</td></tr>
<tr><td>Main flow</td><td><table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Get add new user after log in. recipeInfo (items).
2. editRecipe(items).
3. deleteRecipe() from the information.</td><td>1.1. System returns entry new recipe successfully.
2.2 The system return edit recipe successfully.
3.1 Delete recipe successfully.</td></tr>
</table></td></tr>
<tr><td>Alternative scenario</td><td><table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>2. editRecipe(items).</td><td>2.1 System returns edit unsuccessfully.</td></tr>
</table></td></tr>
</table>

2.4.4 Suggest Recipe

Table 4: Description of Use Case 4: Suggest Recipe

<table>
<tr><td>Scope</td><td>System suggest recipe</td></tr>
<tr><td>Level</td><td>User level</td></tr>
<tr><td>Primary Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User must login first</td></tr>
<tr><td>Post condition</td><td>System must be suggest recipe to user</td></tr>
<tr><td>Main flow</td><td><table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>3. Get log in after registration. Get (user name, password) for log in in system.
4. Get recipeItem(items) for suggest recipes.</td><td>1.2. System returns log in successfully.
4.1. The system return recipe name successfully.</td></tr>
</table></td></tr>
<tr><td>Alternative scenario</td><td><table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>2. Get log in after registration. Get (user name, password) for log in in system.</td><td>2.1. System returns wrong user name.</td></tr>
</table></td></tr>
</table>

2.5 System Sequence Diagram

2.5.1 Manage User

2.5.1.1 Manage User (success scenario)

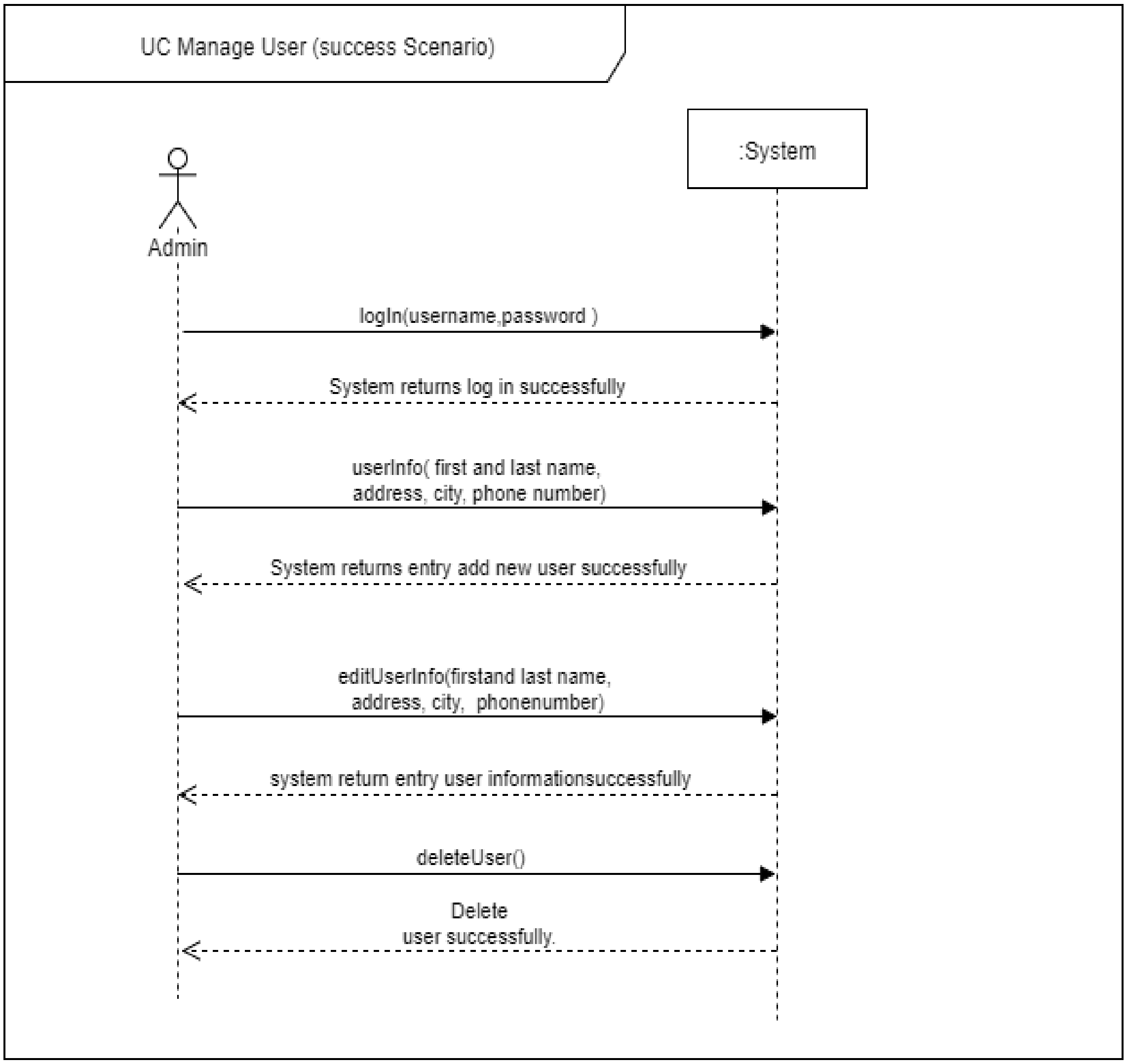
UC Manage User (success Scenario)
:System
Admin
logIn(username,password)
System returns log in successfully
userInfo(first and last name,
address, city, phone number)
System returns entry add new user successfully
editUserInfo(firstand last name,
address, city, phonenumber)
system return entry user informationsuccessfully
deleteUser()
Delete
user successfully.

Figure 4: System sequence diagram for use case Manage User success scenario

2.5.1.2 Manage User (Alternative scenario)

2.5.2 Record Recipe

2.5.2.1 Record Recipe (success scenario)

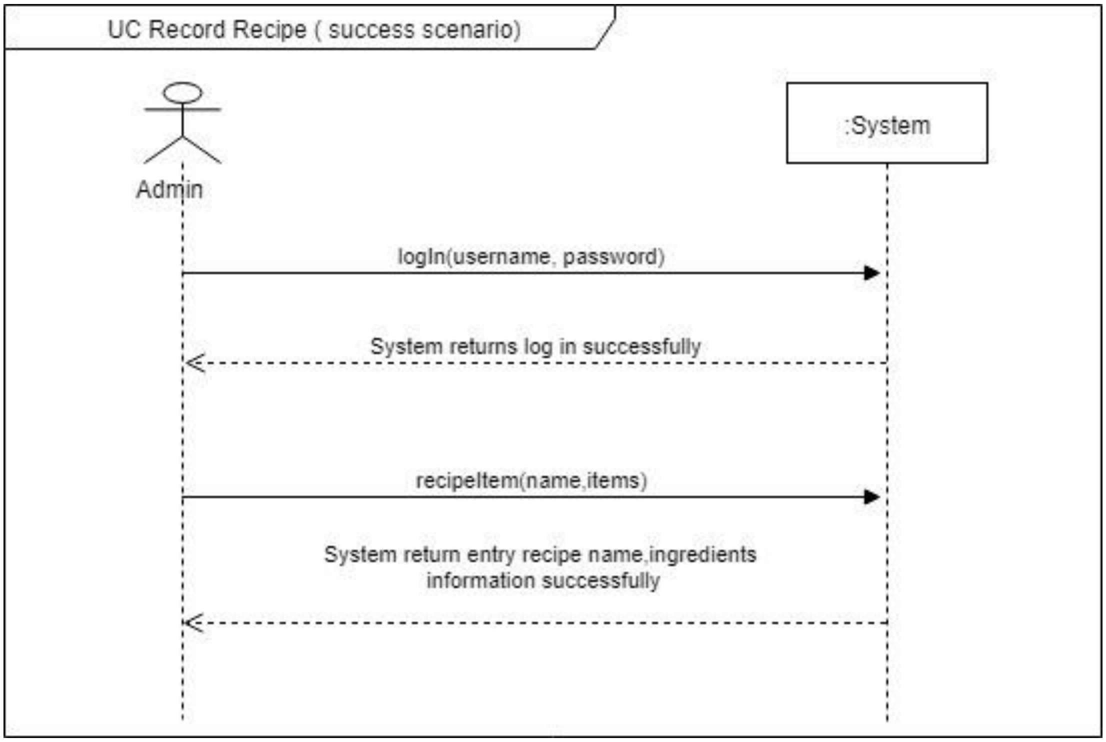
UC Record Recipe (success scenario)
Admin
:System
logIn(username, password)
System returns log in successfully
recipeItem(name,items)
System return entry recipe name,ingredients information successfully

Figure 5: System sequence diagram for use case Record Recipe success scenario

2.5.2.2 Record Recipe (Alternative scenario)

2.5.3 Manage Recipe

2.5.3.1 Manage Recipe (success scenario)

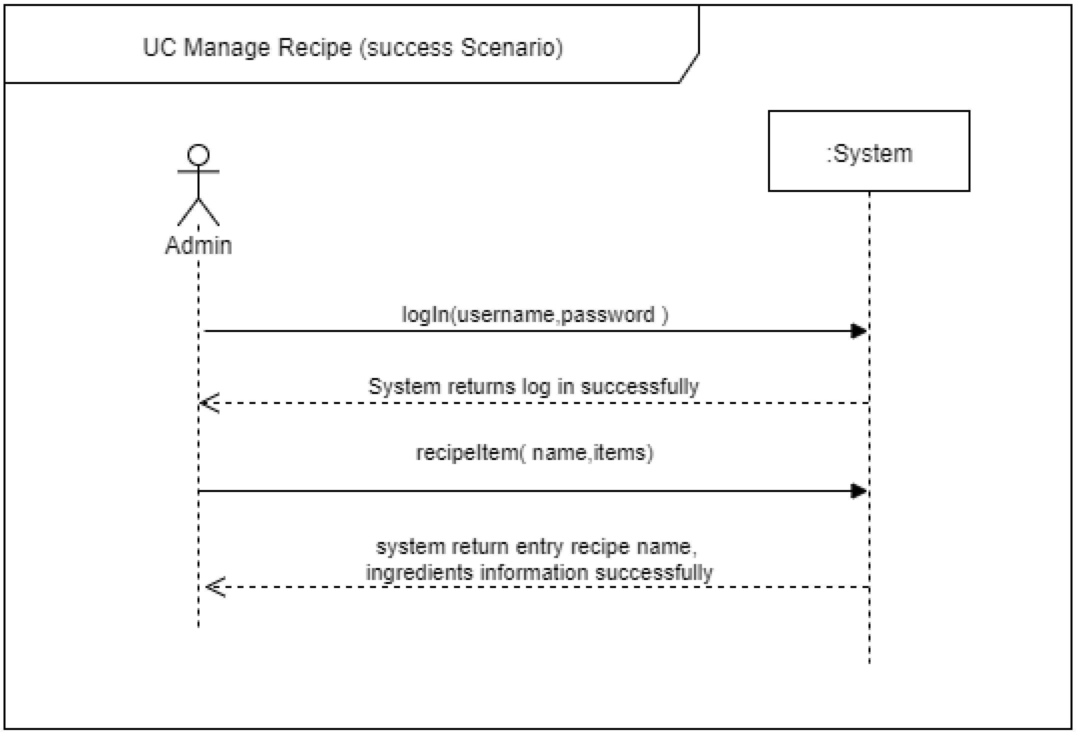
UC Manage Recipe (success Scenario)
:System
Admin
logIn(username,password)
System returns log in successfully
recipeItem(name,items)
system return entry recipe name,
ingredients information successfully

Figure 6: System sequence diagram for use case Manage Recipe success scenario

2.5.3.2 Manage Recipe (Alternative scenario)

2.5.4 Suggest Recipe

2.5.4.1 Suggest Recipe (success scenario)

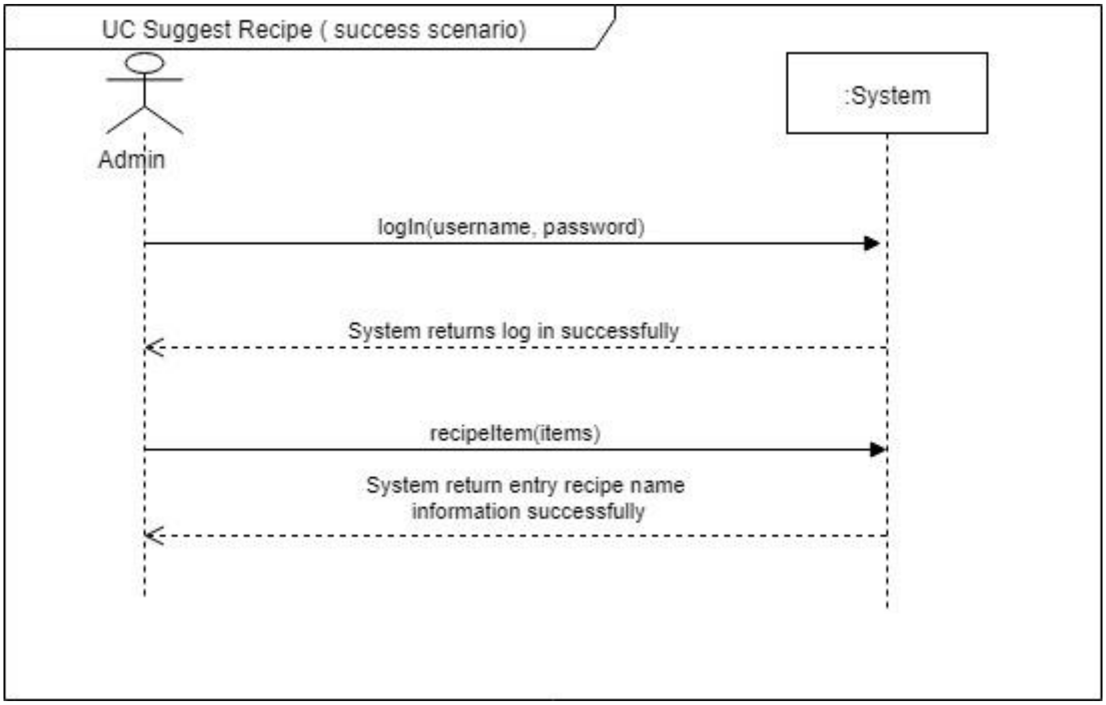
UC Suggest Recipe (success scenario)
Admin
:System
logIn(username, password)
System returns log in successfully
recipeItem(items)
System return entry recipe name information successfully

Figure 7: System sequence diagram for use case Suggest Recipe success scenario

2.5.4.2 Suggest Recipe (Alternative scenario)

2.6 Domain Model

2.7 Activity Diagram

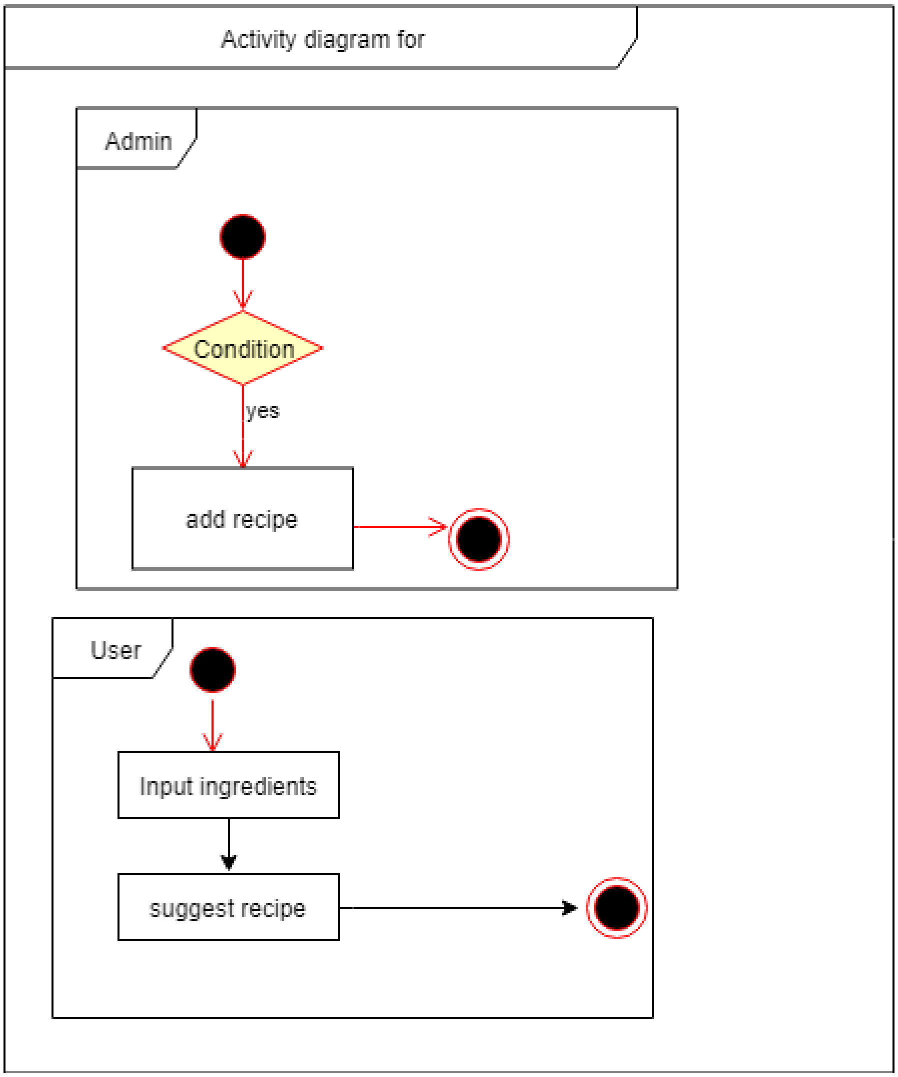

Figure 9: Activity Diagram

Chapter 3

3. System Design

3.1 Sequence Diagram

3.1.1 Manage User

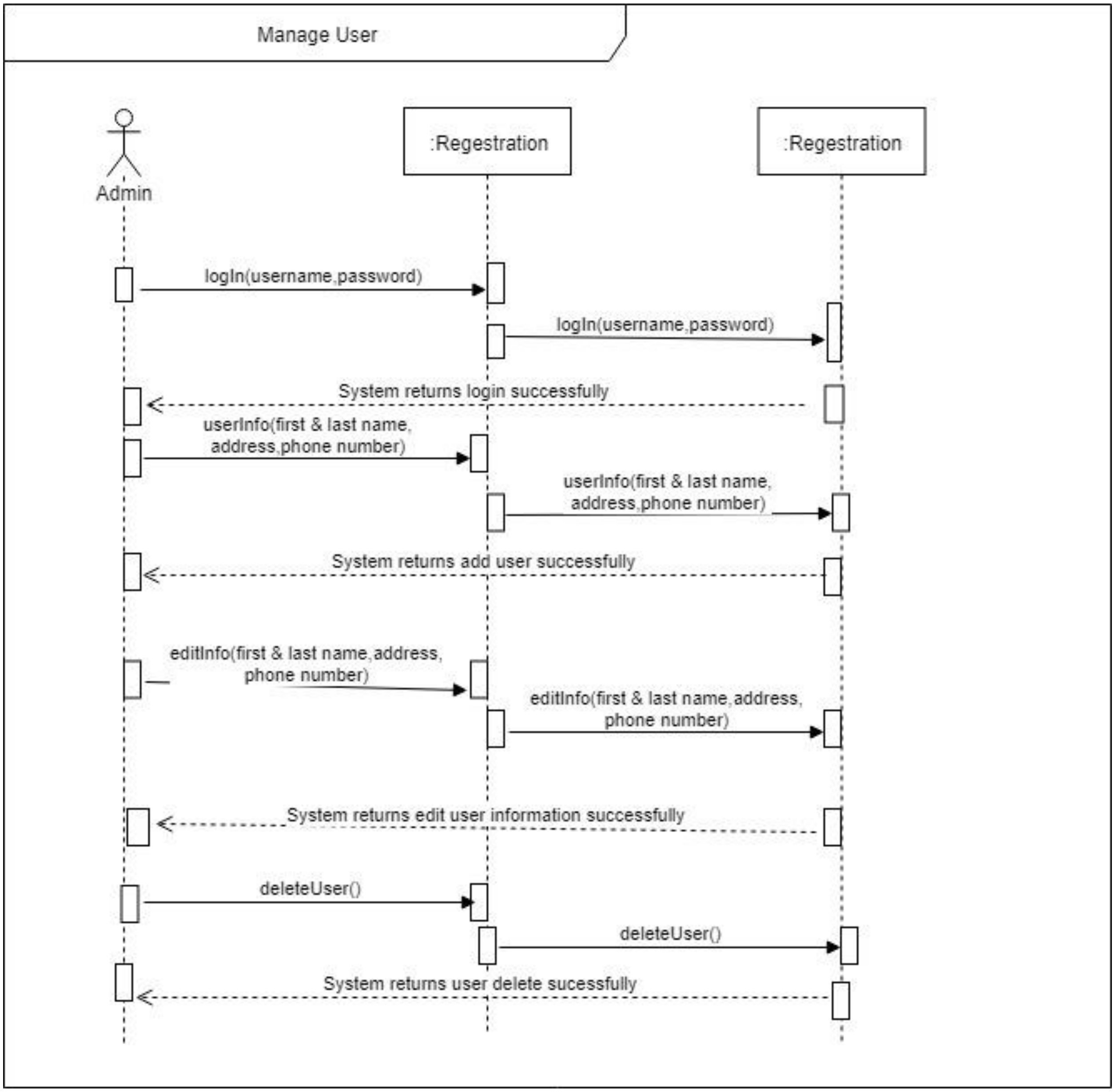
Manage User
Admin
:Regestration
:Regestration
logIn(username,password)
logIn(username,password)
System returns login successfully
userInfo(first & last name, address,phone number)
userInfo(first & last name, address,phone number)
System returns add user successfully
editInfo(first & last name,address, phone number)
editInfo(first & last name,address, phone number)
System returns edit user information successfully
deleteUser()
deleteUser()
System returns user delete sucessfully

Figure 8: Sequence diagram for Manage User

3.1.2 Record Recipe

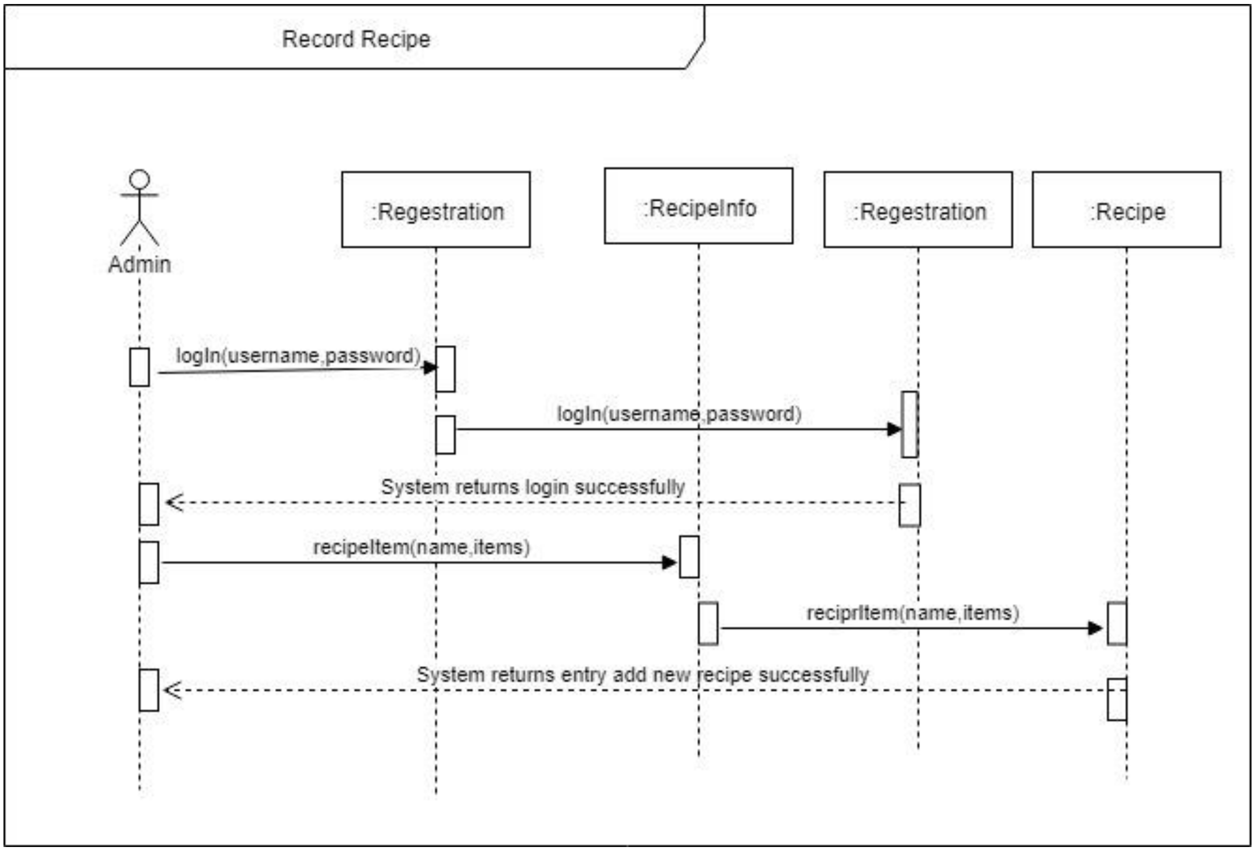

Figure 9: Sequence diagram for Record Recipe

3.1.3 Manage Recipe

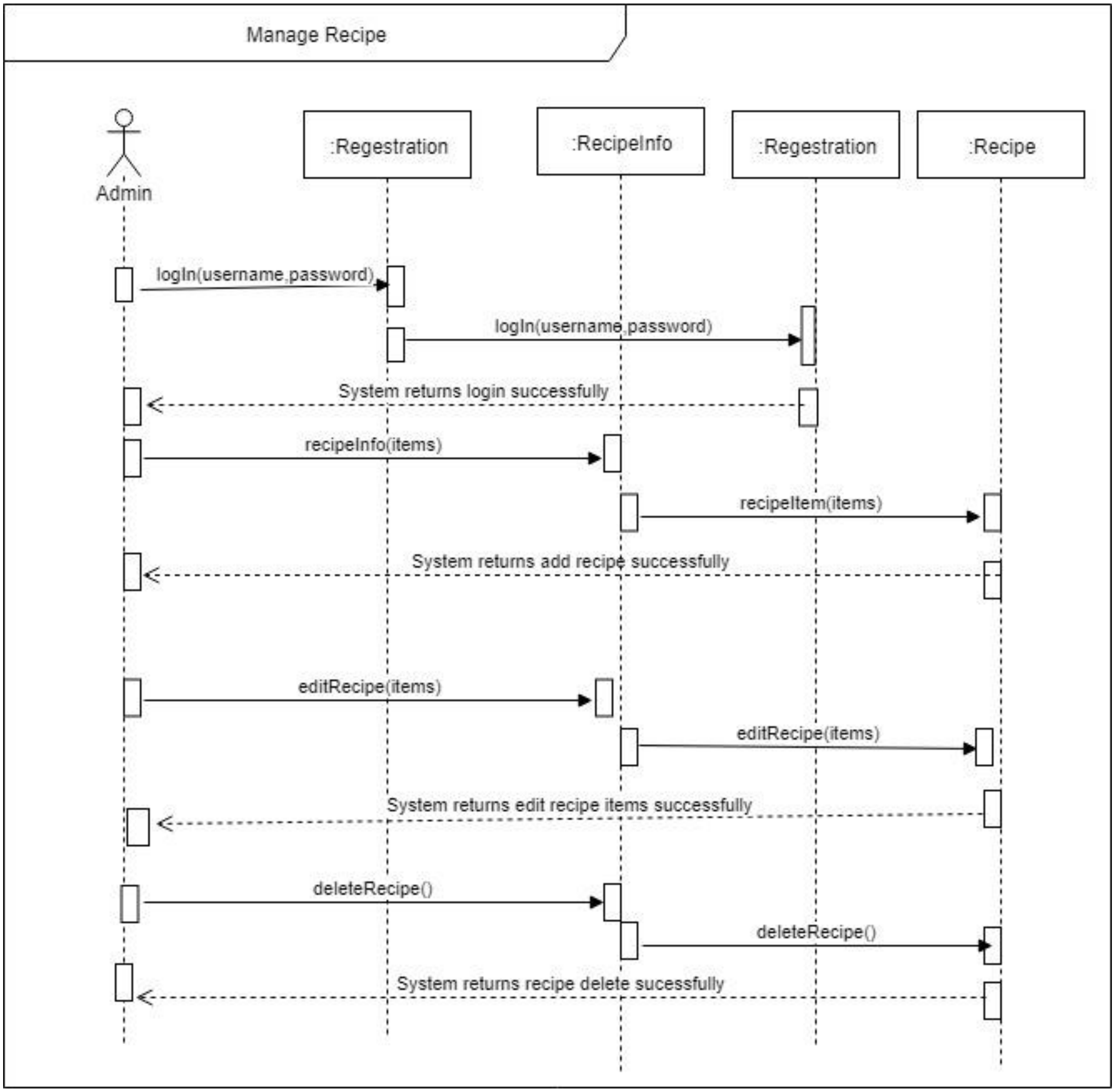

Figure 10: Sequence diagram for Manage Recipe

3.1.4 Suggest Recipe

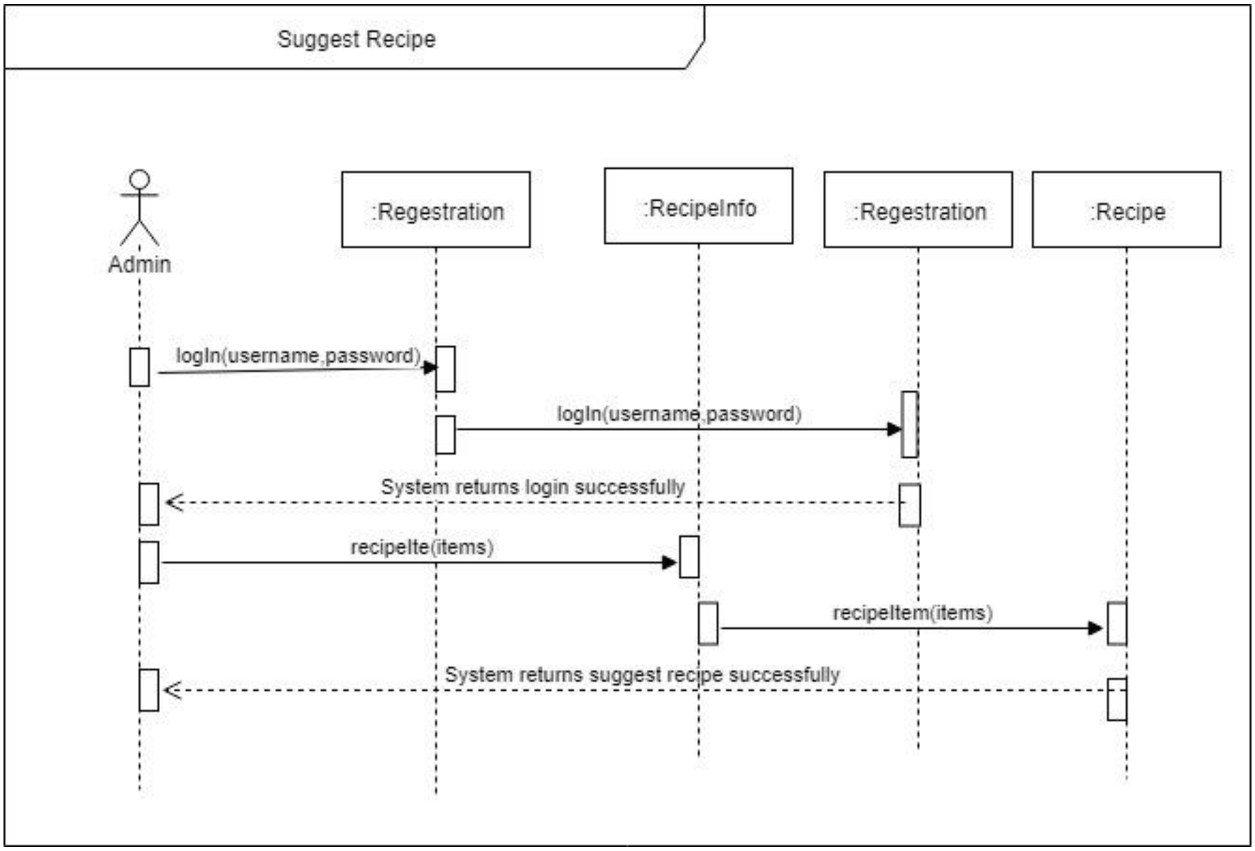

Figure 11: Sequence diagram for Suggest Recipe

3.2 Class Diagram

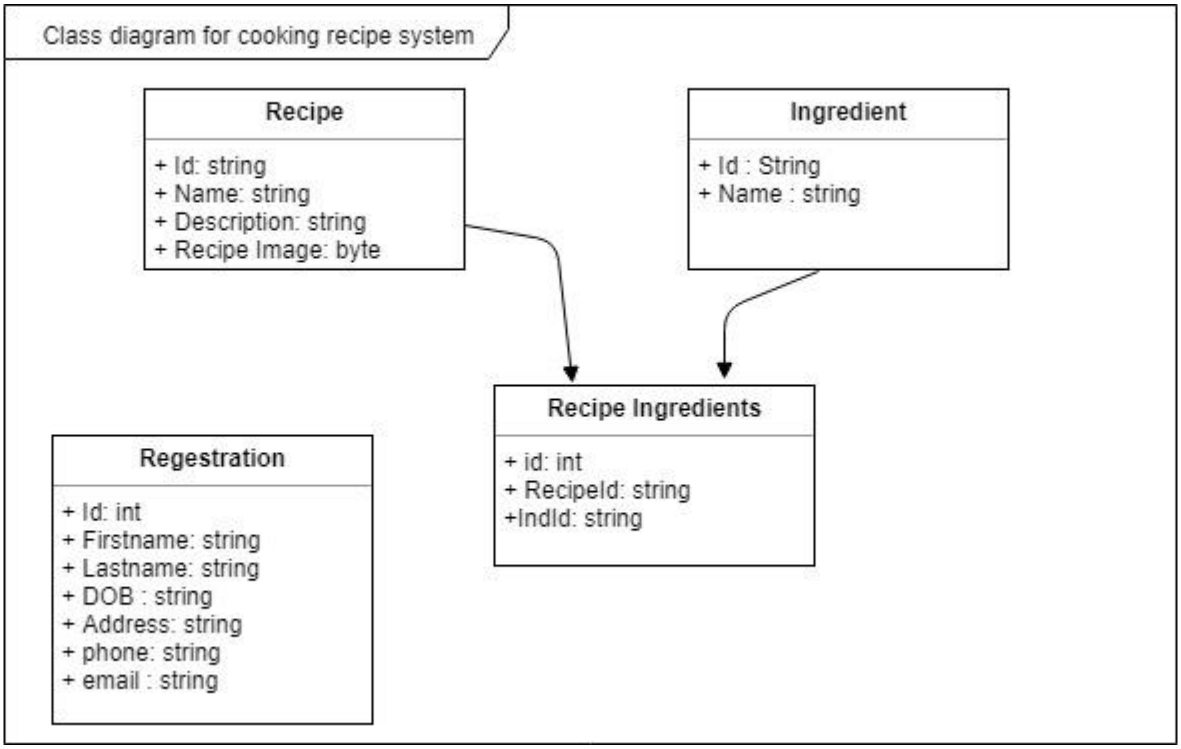

Figure 12: Class diagram for Cooking Recipe system

3.3 Entity Relationship Diagram

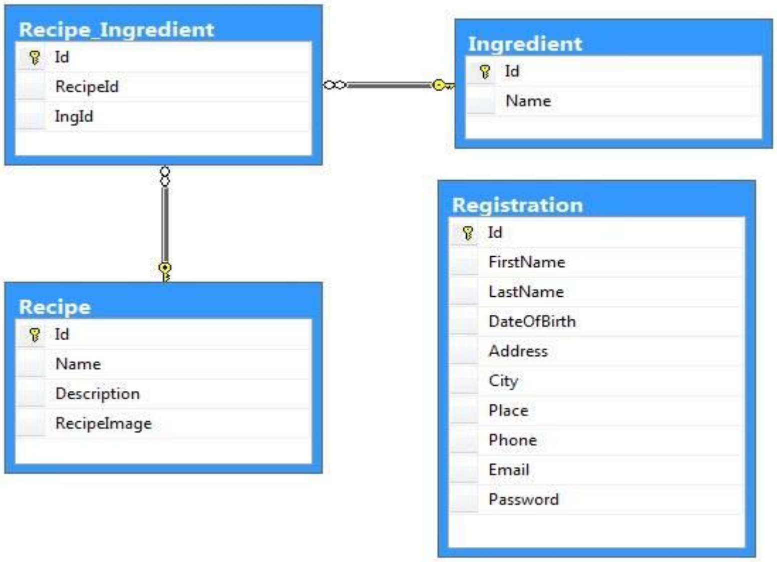

Figure 13: Entity Relationship diagram for Cooking Recipe System

Chapter 4

4. Implementation

Chapter 5

5. System Testing

Chapter 6

6. Conclusion

Appendix

Reference