Contents

[1. INTRODUCTION 2](#_Toc68697112)

[1.1 Background 2](#_Toc68697113)

[1.2 Objectives 2](#_Toc68697114)

[1.3 Scope 2](#_Toc68697115)

[1.4 Limitation 2](#_Toc68697116)

[2. SYSTEM ANALYSIS 3](#_Toc68697117)

[2.1 Use case 3](#_Toc68697118)

[2.2 ER Model 7](#_Toc68697119)

[2.3 Normalization up to BCNF 8](#_Toc68697120)

[3. SYSTEM DESIGN 9](#_Toc68697121)

[3.1 Relational Model 9](#_Toc68697122)

[3.2 Physical Model 10](#_Toc68697123)

[4. IMPLEMENTATION 11](#_Toc68697124)

[4.1 Tools Used 11](#_Toc68697125)

[4.2 Programming Language 11](#_Toc68697126)

[4.3 Data Dictionary 11](#_Toc68697127)

[5. CONCLUSION 11](#_Toc68697128)

# INTRODUCTION

## Background

This Cookbook System was basically designed to provide a platform for people to view, prepare and share the cooking ideas.

The ‘Cook-It’ is an application that lets people all over the world to view the recipes shared by other people and lets you share your own recipe ideas too. The Cook-It application provides user an easy to respond User interface to view Cooking ideas, create your own food recipe and modify and delete them accordingly. It also provides an option to report the inappropriate and duplicate food recipes and also comment on the Recipes.

## Objectives

The main objective of this system is to develop a web application for creating and viewing recipes.

Other objectives include:

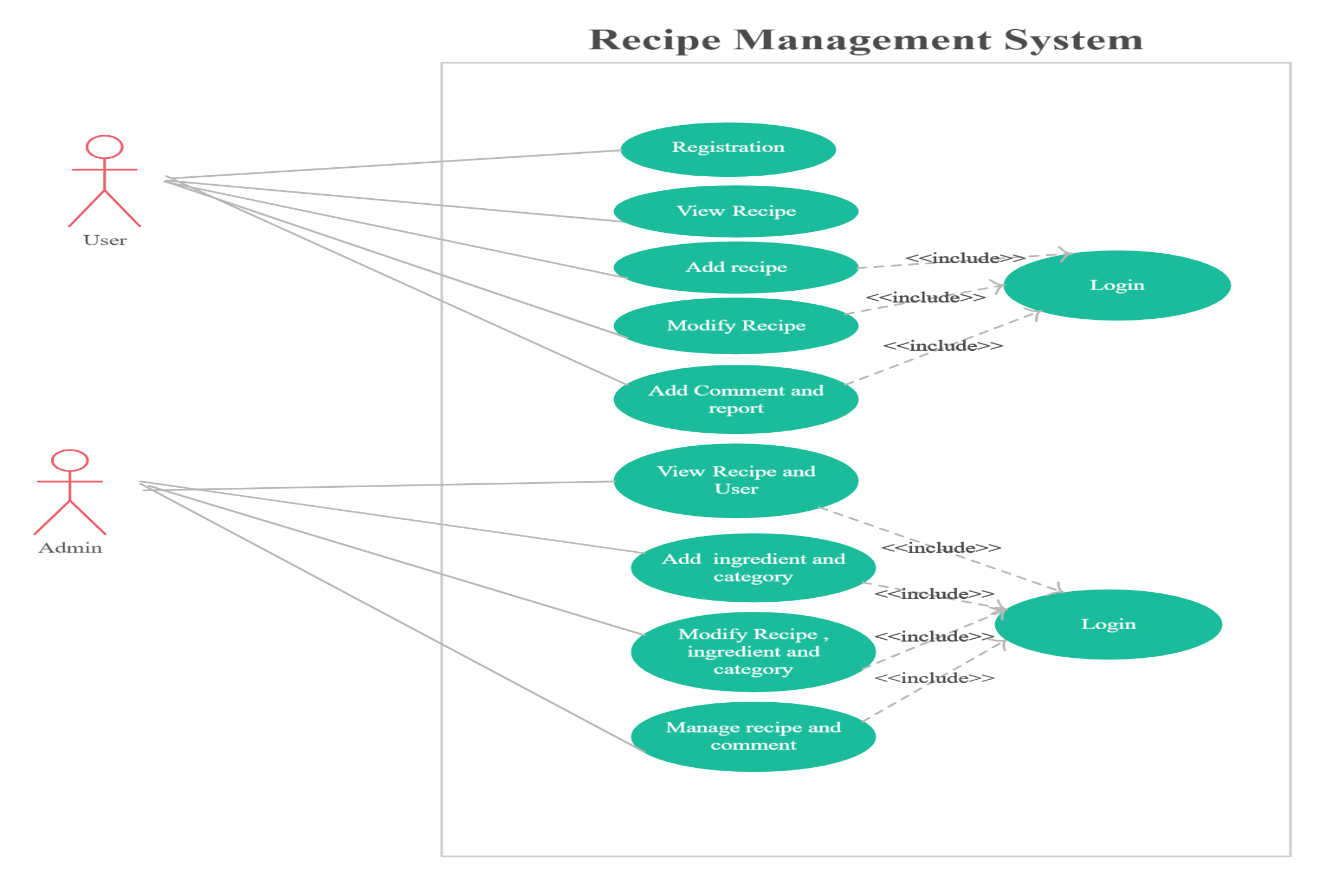
1. View all the recipes available.
2. Login to the system.
3. Create new and unique recipes and share them.
4. Modify their own previously created recipes.
5. Delete their own previously created recipes.
6. Comment & Report on the recipes.

## Scope

## Limitation

# SYSTEM ANALYSIS

## Use case



### **Use case Description**

Table 2.1 Registration

|  |  |
| --- | --- |
| Use-case identifier | UC1: Registration |
| Primary Actor | User |
| Secondary Actor | None |
| Description | New user should to register. |
| Pre-condition | User should register to login. |
| Post-condition | The database must be updated. |
| Success Scenario | Registration Confirm. |
| Failure Scenario | Registration Fail. |

Table 2.2. View Recipe

|  |  |
| --- | --- |
| Use-case identifier | UC2: View Recipe |
| Primary Actor | Admin |
| Secondary Actor | User |
| Description | The Admin and user should be able to view recipe. |
| Pre-condition | The admin and user should login to view recipe. |
| Post-condition | The database must be updated. |
| Success Scenario | Latest recipe made by user is viewed. |
| Failure Scenario | The database is not updated. |

Table 2.3. Add Recipe

|  |  |
| --- | --- |
| Use-case identifier | UC3: Add Recipe |
| Primary Actor | User |
| Secondary Actor | None |
| Description | User should be able to add recipe. |
| Pre-condition | User should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Recipe is added and can be viewed. |
| Failure Scenario | Error in adding. |

Table 2.4. : Modify recipe

|  |  |
| --- | --- |
| Use-case identifier | UC4: Modify recipe |
| Primary Actor | Admin |
| Secondary Actor | User |
| Description | User and admin should be able to modify recipe. |
| Pre-condition | User and admin should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Modification of recipe. |
| Failure Scenario | Error in to edit recipe. |

Table 2.5. Add comment and report

|  |  |
| --- | --- |
| Use-case identifier | UC5: Add comment and report |
| Primary Actor | User |
| Secondary Actor | None |
| Description | User should be able to give their comment and report to recipe. |
| Pre-condition | User should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Able to put comment and report. |
| Failure Scenario | Unable to give comment and report. |

Table 2.6. View recipe and user

|  |  |
| --- | --- |
| Use-case identifier | UC6: View recipe and user |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | Admin should be able to view the recipe and user. |
| Pre-condition | Admin should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Able to view recipe and user. |
| Failure Scenario | Unable to view recipe and user. |

Table 2.7. Add Ingredient and category.

|  |  |
| --- | --- |
| Use-case identifier | UC7: Add Ingredient and category. |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | Admin should be able to add ingredient and category. |
| Pre-condition | Admin should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Able to add ingredient and category. |
| Failure Scenario | Unable to add ingredient and category. |

Table 2.8. Modify recipe and ingredient

|  |  |
| --- | --- |
| Use-case identifier | UC8:Modify recipe and ingredient |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | Admin should be able to modify recipe and ingredient. |
| Pre-condition | Admin should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Able to modify recipe and ingredient. |
| Failure Scenario | Unable to modify recipe and ingredient. |

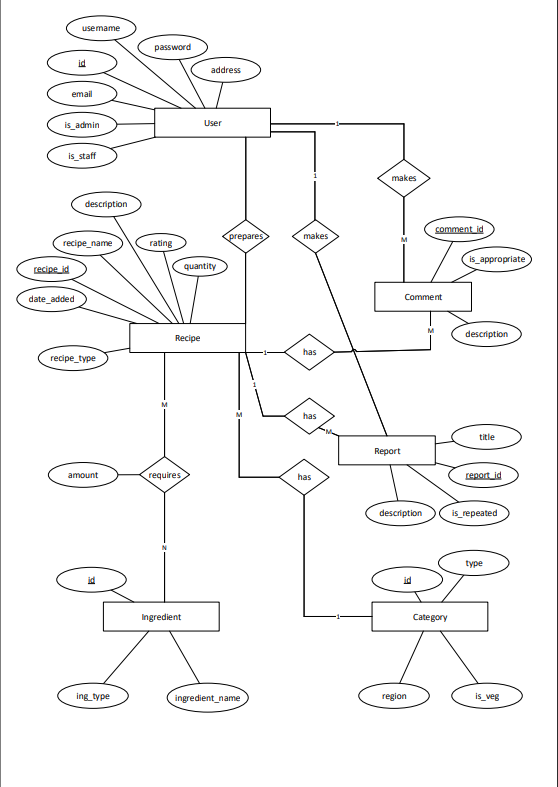
Table 2.9. Manage report and comment

|  |  |
| --- | --- |
| Use-case identifier | UC9: Manage report and comment |
| Primary Actor | Admin |
| Secondary Actor | None |
| Description | Admin should be able to manage report and comment. |
| Pre-condition | Admin should be login. |
| Post-condition | The database must be updated. |
| Success Scenario | Able to manage report and comment. |
| Failure Scenario | Unable to manage report and comment. |

## ER Model

An Entity-Relationship Diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. The following E-R Model shows the relation between User, Recipes, Ingredients, Category and Feedback:

The Entity Relational Model for our project is shown below:



## Normalization up to BCNF

We’ve done following steps to normalize upto Boyce Codd Normalization form:

Step-I: Normalization into 1NF (First Order Normal Form).

1. We’ve made sure that each column’s value into single values.
2. For the same table, there is no exact same tuples. (As primary key is changed from 1 tuple to another if all other data are same.)

Step-II-: Normalization into 2NF (Second Order Normal Form).

1. In this Step, the previous condition of the normalization into 1NF is satisfied.
2. There is no any partial dependency between. (i.e. All non-prime attributes are independent upon the part of the primary key).

Step-III: Normalization into 3NF (Third Order Normal Form).

1. We’ve made all the relations into 3NF.
2. We’ve made sure that there is no any transitive dependency. (i.e. There is no dependency of non-prime attribute with other non-prime attribute.)

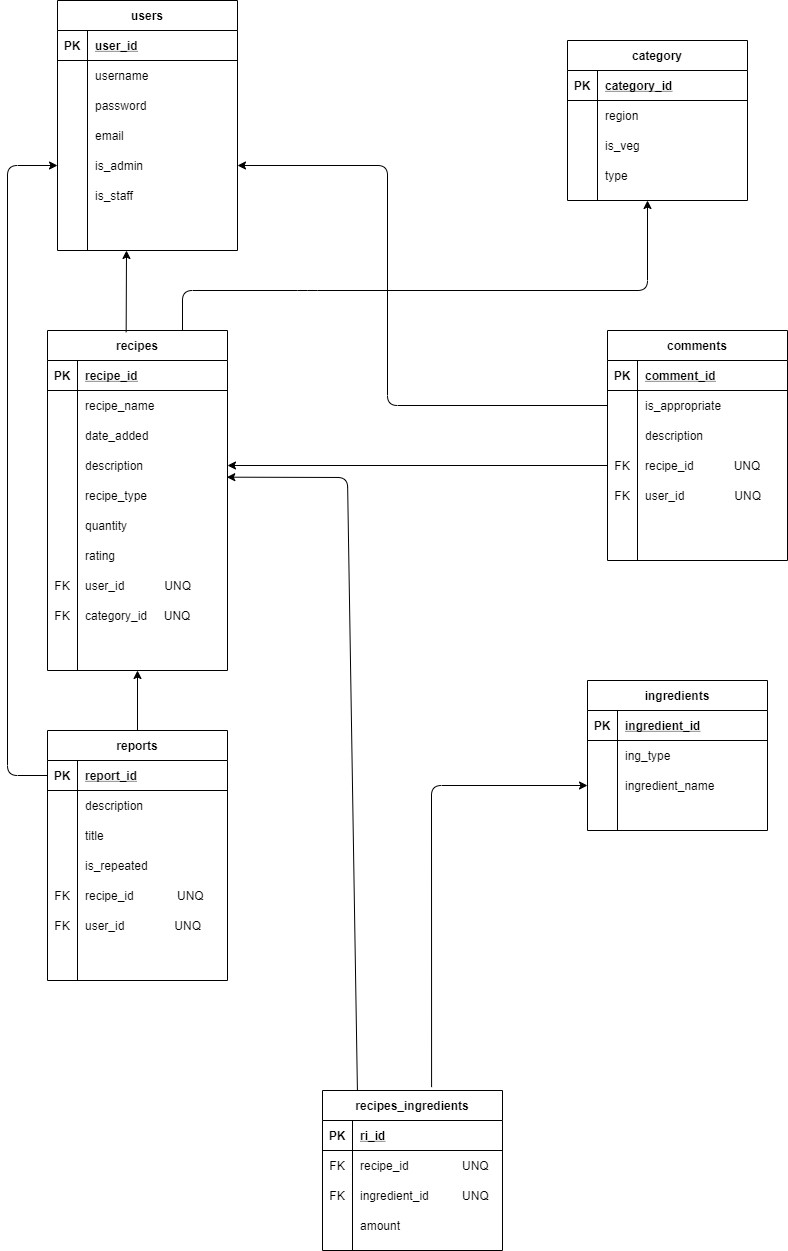
Step-IV: Normalization into BCNF.

1. We’ve made all the relations into 3NF.
2. There is no multiple column primary key.

# SYSTEM DESIGN

## Relational Model

The Relational Model of our Cookbook System is shown below:



## Physical Model

# IMPLEMENTATION

## Tools Used

The tools used in designing the system are:

1. MS Visio
2. Draw.io
3. MS Word
4. Internet Browser (Brave, Internet Explorer, Chrome)
5. Visual Studio Code.

## Programming Language

The Programming languages used in designing the system are:

1. SQL
2. HTML/CSS
3. JavaScript
4. Python (Django Web Framework)

## Data Dictionary

# CONCLUSION

This project has been a great opportunity through which various professional knowledge of developing database system in real life scenario has increased. This project gave a lot of knowledge about how to build a good Web based Database management System(DBMS). It is uniquely developed system that contains unique design. This project has been very helpful to demonstrate how software is developed.