Recipe Management System

CS162- Final Project Requirement Document



Project Supervisor

Mr. Samyan Qayyum Wahla

Project Members(Group Id Here)

|  |  |
| --- | --- |
| Member 1 Esha Tur Razia | Member 1 2020-CS-71. |
| Member 2 Sana Shafique | Member 2 2020-CS-84. |

Department of Computer Science

University of Engineering and Technology, Lahore

Pakistan

Table of Contents

[Project Description: 3](#_Toc71624246)

[Project Features: 4](#_Toc71624247)

[Technology Stack: 5](#_Toc71624248)

[Project Actors: 6](#_Toc71624249)

[Use Cases: 7](#_Toc71624250)

[Use Case 1(Write the name of the use case): 7](#_Toc71624251)

[User Interface Details 8](#_Toc71624252)

[Classes: 9](#_Toc71624253)

[Object Oriented Features: 10](#_Toc71624254)

[Composition: 10](#_Toc71624255)

[Inheritance: 10](#_Toc71624256)

[Multiple Inheritance: 10](#_Toc71624257)

[Multi-Level Inheritance: 10](#_Toc71624258)

[Polymorphism: 10](#_Toc71624259)

[Collections: 11](#_Toc71624260)

[Exceptions: 12](#_Toc71624261)

[Data Storage: 13](#_Toc71624262)

[Email Sending: 14](#_Toc71624263)

[Project Plan 15](#_Toc71624264)

# Project Description:

This section contains the comprehensive description of the project(minimum 800 words). The description should be written in your own words and should not be plagiarized from any source.

When you have people working in shifts, different locations or in ghost kitchens you need to be sure everybody is on the same page at all times.  
  
Customers expect the same great experience at every one of your hotel restaurants, food outlets or online ordering channels.  
  
Thanks to our recipe management software all your kitchen teams have access to ingredients lists, recipes, prep overviews, plating instructions and menus.  
  
You can attribute different properties to each team depending on, for example, location or target audience. And compare the performance of all outlets.

That is easy and efficient.

A recipe management system multiplies that business value.  
  
Software for better recipe management is the steering wheel of your ship. It’s the GPS-system that keeps your restaurant business on target

# Project Features:

In this section, write down the project features clearly. These are the features that will be available to the customer of your project directly.

Firstly , it shows the home page where one can choose admin ,user and contact number .

In admin option, first he will login by using password and name .Then he will be able to add recipe ,edit recipe, search recipe and log out.he will be able to edit , add or delete igredients.

In user option, he will login by using password and name and then he will be able to to fill the form to register to access the recipies.he will view the recipies and log out.

In contact us section, contact number and few information is displayed to the user.

# Technology Stack:

|  |  |
| --- | --- |
| Language(C#/Java) | C# |
| Platform(Web/Desktop) | Desktop |
| Frontend Technology(Simple HTML/Bootstrap/any other library) | VS forms |
| IDEs | Visual Studio |

# Project Actors:

In this sections, write the name of actors and brief description, who will be using the system.

It consists of two actors , user and admin.

Admin can add recipie sech recipie while user can access the recipies after being resister and then view recipie which displays ingredients.

# Use Cases:

All the use cases should be written in the following format. The name of use should be start with a word e.g. Add Student. It should not be as Student/ Student Add. Add Student is separate use case and Edit Student is separate use case:

## Use Case 1(Admin login):

|  |  |
| --- | --- |
| Use Case ID | U01 |
| Name | login |
| Actor | Admin |
| Description | Brief description of the use case.  Admin after login can access the program .validation is used for name and password .afte entering correct password or name he will be able to access this system. |
| Layout in pencil tool | E:\Sana\untitled_page.png |
| Validators | Add here the name of validators that will be applied on this page  Email validator  Password validator |

## Use Case 2(Add recipie):

|  |  |
| --- | --- |
| Use Case ID | U02 |
| Name | Add recipie |
| Actor | Admin |
| Description | Brief description of the use case. With at least 50 words.  After admins successful attemptof entering password and name and after validating he will be able to add the details of recipie such as name , quanity , ingredients and description of the recipie. |
| Layout in pencil tool | E:\Sana\untitled_page.png |
| Validators | Name validator |

## Use Case 3(Delete recipie):

|  |  |
| --- | --- |
| Use Case ID | U03 |
| Name | Delete recipie |
| Actor | admin |
| Description | Brief description of the use case. With at least 50 words.  After admins successful attemptof entering password and name and after validating he will be able to delete the details of recipie such as name , quanity , ingredients and description of the recipie |
| Layout in pencil tool | E:\Sana\untitled_page.png |
| Validators | Add here the name of validators that will be applied on this page  Name validator |

## Use Case 4(Edit recipie):

|  |  |
| --- | --- |
| Use Case ID | U04 |
| Name | Edit recipie |
| Actor | admin |
| Description | Brief description of the use case. With at least 50 words.  After admins successful attemptof entering password and name and after validating he will be able to edit the details of recipie such as name , quanity , ingredients and description of the recipie |
|  |  |
| Layout in pencil tool | E:\Sana\untitled_page.png |
| Validators | Add here the name of validators that will be applied on this page  Name validator |

|  |  |
| --- | --- |
| Use Case ID | U05 |
| Name | User login |
| Actor | User |
| Description | This use case is used to enter the user to the system. First the user will enter the email and the password and the then it can access to the system. |
| Layout in pencil tool |  |
| Validators | Email validation will occur  Password validation will occur |

## Use Case 6(Search recipes by user):

|  |  |
| --- | --- |
| Use Case ID | U06 |
| Name | Search recipes |
| Actor | User |
| Description | This use case is used to search recipes from the system . First the user will enter the email and the password and the then it can access to the system and will have two options either to search recipes or view all the recipes. |
| Layout in pencil tool |  |
| Validators | Validation for recipe name will occur only alphabets can be entered. |

## Use Case 7(View recipes by user):

|  |  |
| --- | --- |
| Use Case ID | U07 |
| Name | view |
| Actor | User |
| Description | This use case is used to view recipe from the the system. First the user will enter the email and the password and the then it can access to the system then after searching for the recipe user can find the recipe and the view it. |
| Layout in pencil tool |  |
| Validators | No validation in this use case |

## Use Case 8(Search recipe by admin):

|  |  |
| --- | --- |
| Use Case ID | U08 |
| Name | Search recipe |
| Actor | Admin |
| Description | This use case is used to search recipe from the system. First the admin will enter the email and the password and the then it can access to the system after which he can search for the recipes. |
| Layout in pencil tool |  |
| Validators | Only alphabets in the recipe will be accepted |

## Use Case 9(Homepage):

|  |  |
| --- | --- |
| Use Case ID | U09 |
| Name | homepage |
| Actor |  |
| Description | It displays the homepage |
| Layout in pencil tool |  |
| Validators |  |

## Use Case 10(About us):

|  |  |
| --- | --- |
| Use Case ID | U010 |
| Name | About us |
| Actor |  |
| Description | This use case is used to know about the system |

# User Interface Details

In this section, fill the table for summary that which use case will have the required component. Inside each box, write the counts for each component. If component is not used, write zero.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | labels | CheckBox | Menu | Text Area | ProgressBar |
| U01 | 1 |  | 1 |  |  | 1 |  | 5 |  |  | 3 |  |
| U02 | 4 |  |  |  |  | 1 |  | 8 |  |  |  |  |
| U03 | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 8 | 0 | 0 | 0 | 0 |
| U04 | 4 |  |  |  |  | 1 |  | 8 |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | Radio Button | CheckBox | Menu | Text Area | ProgressBar |
| U05 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| U06 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 4 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U07 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| U08 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |  |
| U09 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 0 |
| U10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

# Classes:

In this section, we do not require detailed design diagram. But identify the tentative classes with the requirement that you should have at least 10 domain classes and 5 software classes, 3 abstract classes, 2 singelton classes. Fill the following table for details. Note that class name should follow naming conventions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class Name** | **Software/ Domain** | **Is Abstract (Yes/No)** | **Is Singleton (Yes/No)** | **Is the class will has parametrized constructor(Yes/No)** |
| ingredient | Domain | No | No | Yes |
| recipe | Domain | No | No | Yes |
| instruction | Domain | Yes | No | No |
| Shopping list | Software | No | No | No |
| Recipe list | Software | No | No | No |
| quantity | Domain | No | Yes | Yes |

# Object Oriented Features:

## Composition:

In this section, Identify the at least three solid examples where composition can be perform. Add UML diagram of each example.

## Inheritance:

At least 2 examples where inheritance will take place with UML diagrams.

## Multiple Inheritance:

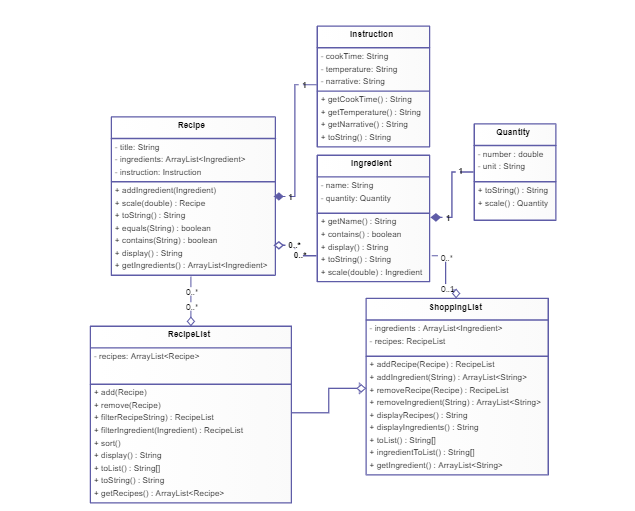
Two example with UML diagrams of multiple inheritance.

## Multi-Level Inheritance:

Two examples with UML diagrams of multi-level inheritance.

## Polymorphism:

At least 3 examples with UML diagram for polymorphism.





# Collections:

In this section, describe how and where you will use the following collections and why you are forced to use these collections.

ArrayList, LinkedList, Queue, Stack, HashSet and TreeSet

* We will use array list and linked list to store data.
* As using list is the more efficient way to store data.
* Lists can grow or shrink when we allocate or deallocate memory.
* They provide fast iteration of elements using indexing.
* Linked list is better to add any element.
* Array list is better for getting any random element.

# Exceptions:

In this sections, identify at high level which type of exceptions you can face in your code and what are the solutions. Add more rows in the table as per requirements.

|  |  |  |
| --- | --- | --- |
| Type of Exception | Why this exception will occur | How you will handle the exception |
| |  |  | | --- | --- | | **Exception Class** |  | | NullReferenceException |  | |  |  | |  |  | | Raised when program access members of null object. | By using GetValueOrDefault() |
| |  |  | | --- | --- | | OverflowException | . | | Raised when an arithmetic, casting, or conversion operation results in an overflow | if an **overflow** happens, you need to use the checked { ... } code block. |
| |  |  | | --- | --- | | OutOfMemoryException | . | | Raised when a program does not get enough memory to execute the code | By using try catch |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Data Storage:

In this sections, describe the five files with their format from where you will read or store data. In case of database table, write down the names of columns for tables. At least 5 files/ tables are required.

1<Recipie :

Which contains the recipies names.

2<Procedure:

Which contains description of recipies.

3<Ingredients:

Which contains ingredients of recipies.

# Email Sending:

In this section, describe the points where you will be required to send the email from the code. Additionally, write down the sample subject and email content.

When user fill the form he will receive an email:

Dear user , you have successfully registered now you can view the recipie by accessing the system.

Thank you.

# Project Plan

This section should include the implementation plan and work division among the members. All the estimated dates should be before June 15, 2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Id** | **Use Case Name** | **Member Name** | **Estimated Completion Date** |
| 1 | Admin login | Sana shafique | 3rd june |
| 2 | Add recipe | Sana shafique | 9th june |
| 3 | Edit recipe | Sana shafique | 1st june |
| 4 | Delete recipe | Esha tur razia | 31st may |
| 5 | User login | Esha tur Razia | 8th june |
| 6 | Search recipe by user | Esha tur Razia | 2nd june |
| 7 | View recipe | Esha tur Razia | 4th june |
| 8 | Search recipe by admin | Sana shafique | 5th june |
| 9 | homepage | Esha tur Razia | 6th june |
| 10 | About us | Sana shafique | 7th june |