جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

Recipe Management System
Report

Course : OOP 1

Dr.Haya Aldawsri

Level: 3

Name of students	ID number
Asayel Albuainain	2240007410
Laila Albadri	2240007284
Lwlwah Alyusif	2240007777
Jory Ibraheem	2240005101
Waad Aloomar	2240007034

Abstract:

The Recipe Management System is a Java-based application that enables users to manage and interact with recipes. The system distinguishes between two types of users: **Regular users**, who can only view predefined recipes, and **Premium users**, who can add, edit, and interact with all recipes. The project demonstrates the principles of object-oriented programming (OOP), including encapsulation, inheritance, and polymorphism. The system allows for role-based access, commenting on recipes, and community engagement.

Introduction:

Background

The Recipe Management System is designed to simplify how users manage and share cooking recipes. Many cooking enthusiasts and chefs face challenges in organizing recipes and accessing unique ones. By introducing role-based access, the system provides premium functionality to enhance user experience and engagement.

Objective

The primary goal of this project is to create a structured recipe management application with:

- A role-based access system for Regular and Premium users.
- Predefined recipes available for Regular users.
- Full recipe creation, editing, and interaction for Premium users.
- Community features like commenting on recipes.

Scope

The system focuses on managing recipes with the following functionalities:

- Viewing recipes.

- Adding comments to recipes.

- Differentiating between Regular and Premium users.

System Design

Architecture

The system is based on object-oriented programming principles and includes:

1. **Encapsulation**: Protects data fields using private access modifiers and provides controlled access through getters and setters.

2. **Inheritance**: The User class is a base class, extended by RegularUser and PremiumUser. Similarly, Recipe is a base class for MainDishRecipe, DessertRecipe, and BeverageRecipe.

3. **Polymorphism**: Subclasses like `PremiumUser` and `RegularUser` behave differently when interacting with recipes.

Class Descriptions

1. User Class:

- Base class for managing users.

- Subclasses: `RegularUser` (view-only) and `PremiumUser` (full access).

2. Recipe Class:

- Base class for managing recipes.

- Subclasses: `MainDishRecipe`, `DessertRecipe`, `BeverageRecipe`.

3. Key Responsibilities:

- `RegularUser`: Can only view predefined recipes.

- `PremiumUser`: Can add, edit, and mark recipes as special.

- `Recipe`: Manages recipe data like title, ingredients, instructions, and comments.

Code Structure

- Main Class: Manages user interaction (login, register, view recipes).

- Supporting Classes:

- `User` and its subclasses handle user roles and permissions.

- `Recipe` and its subclasses manage different types of recipes.

Difficulties and Solutions:

The challenges faced during development and how they were solved:

- **Role-Based Access:**

Challenge: Managing role-based access (Regular vs. Premium) so that Regular users couldn't add recipes but could comment.

Solution: Used the instanceof operator to check the user's role and restrict access to functionality like adding recipes.

Handling Invalid Inputs

Challenge:

- Users might enter invalid data (Ex: selecting a non-existent menu option)

Solution:

- Added **input validation**:

  For menu options,we used a condition to check and ensure the choice was valid , so that if the user enterd incorrect number th is massege will exist " Invalid choice. Please try again."

Requirements Fulfilled:

1. Classes Management

Fulfillment:

- The program is structured around multiple classes to encapsulate functionality:
    - **Superclass Recipe**: Handles shared properties and behaviors of all recipes (title, ingredients, instructions, comments, etc.).
    - **Subclasses (MainDishRecipe, DessertRecipe, BeverageRecipe)**: Represent specific types of recipes, inheriting from Recipe.
    - **Superclass User**: Provides a base structure for users (username and password).
    - **Subclasses (RegularUser, PremiumUser)**: Represent user roles with specific permissions.
    -

2. Inheritance & Polymorphism

Fulfillment:

- **Inheritance**:
    - Recipe is the superclass for MainDishRecipe, DessertRecipe, and BeverageRecipe.
    - User is the superclass for RegularUser and PremiumUser.
- **Polymorphism**:
    - Methods in the Recipe and User classes are overridden in subclasses where needed.

3. Array of Objects

Fulfillment:

- Recipes and users are stored in dynamic **ArrayList** objects:

- **Dynamic and Editable**: New recipes are added to the recipes list during runtime, and users are added to the users list.
- **Hierarchy**:

  The recipes list stores objects of the Recipe type, which can be any of its subclasses (MainDishRecipe, DessertRecipe, BeverageRecipe).

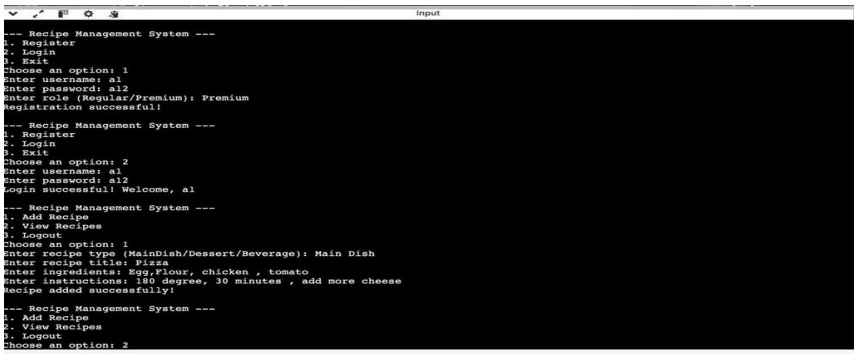
Work Division

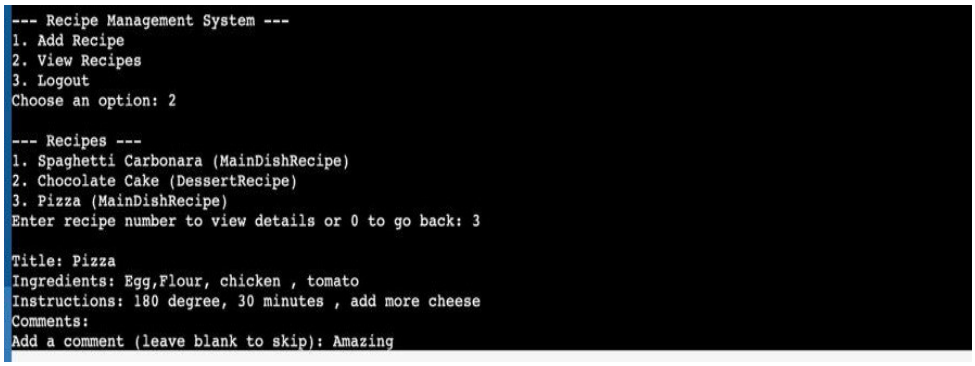
Divide the work among team members and describe what each person contributed:

- **Member 1 (Jory) :** Responsible for the User class , user login , Regular user subclass , Premium user subclass .
- **Member (Waad) :** Responsible for the User registration , User Class .
- **Member 3 (Asayel) :** Responsible for the Recipe class, Main dish subclass, add recipe and view recipe methods .
- **Member 4 (Laila):** Responsible for Recipe class , Desert recipe subclass , add recipe and view recipe methods .
- **Member 5 (Lolwah):** Responsible for Recipe class , Beverge recipe subclass, add recipe and view recipe methods .

Screen Shots of interface:

Premium user interface





Regular user interface

```
-- Recipes ---
. Spaghetti Carbonara (MainDishRecipe)
. Chocolate Cake (DessertRecipe)
nter recipe number to view details or 0 to go back: 1

itle: Spaghetti Carbonara
ngredients: Spaghetti, Eggs, Bacon, Parmesan
nstructions: Boil pasta, cook bacon, mix all with eggs and cheese.
omments:
dd a comment (leave blank to skip): Good
omment added!

-- Recipe Management System ---
. Add Recipe
. View Recipes
. Logout
hoose an option: 3
ogged out.

-- Recipe Management System ---
. Register
. Login
. Exit
hoose an option:
```

```
--- Recipe Management System ---
1. Register
2. Login
3. Exit
Choose an option: 4
Invalid choice. Try again.

--- Recipe Management System ---
1. Register
2. Login
3. Exit
Choose an option:
```

Conclusion:

The Recipe Management System is a successful implementation of a Java-based application that leverages object-oriented programming principles to provide a role-based platform for managing recipes. Regular users can view predefined recipes, while premium users enjoy full access to add, edit, and mark recipes as special. The system fosters community engagement by allowing users to comment on recipes, creating a collaborative environment. Despite its achievements, such as effective role management and encapsulated class design, the system has some limitations, including the lack of persistent storage and a graphical user interface. These limitations provide avenues for future enhancements, such as integrating a database, implementing a payment system for Premium users, and developing a GUI to improve usability. Overall, the project demonstrates the practical application of OOP concepts like inheritance, polymorphism, and encapsulation, while showcasing a clear separation of functionality between user roles.