Alisha Chowhan, Jay Ma, Jyothi Shankar
CS 151 Sec 05
Professor Nidhi Zare
12 May 2024

Final Deliverable: Online Recipe-Sharing Platform

Project Introduction:

Our project aims to develop an online recipe-sharing platform. Users can seamlessly share and browse their favorite recipes from a wide selection of cuisines. Through the incorporation of encapsulation, inheritance, polymorphism, and abstraction, our application can create an active online community centered around a shared love for food. Incorporating these oriented programming principles into the recipe-sharing platform, the project would be a robust and user-friendly application appealing to various users and building an online community of food lovers. Each OOP principle would help contribute to the quick and easy functionality of the program.

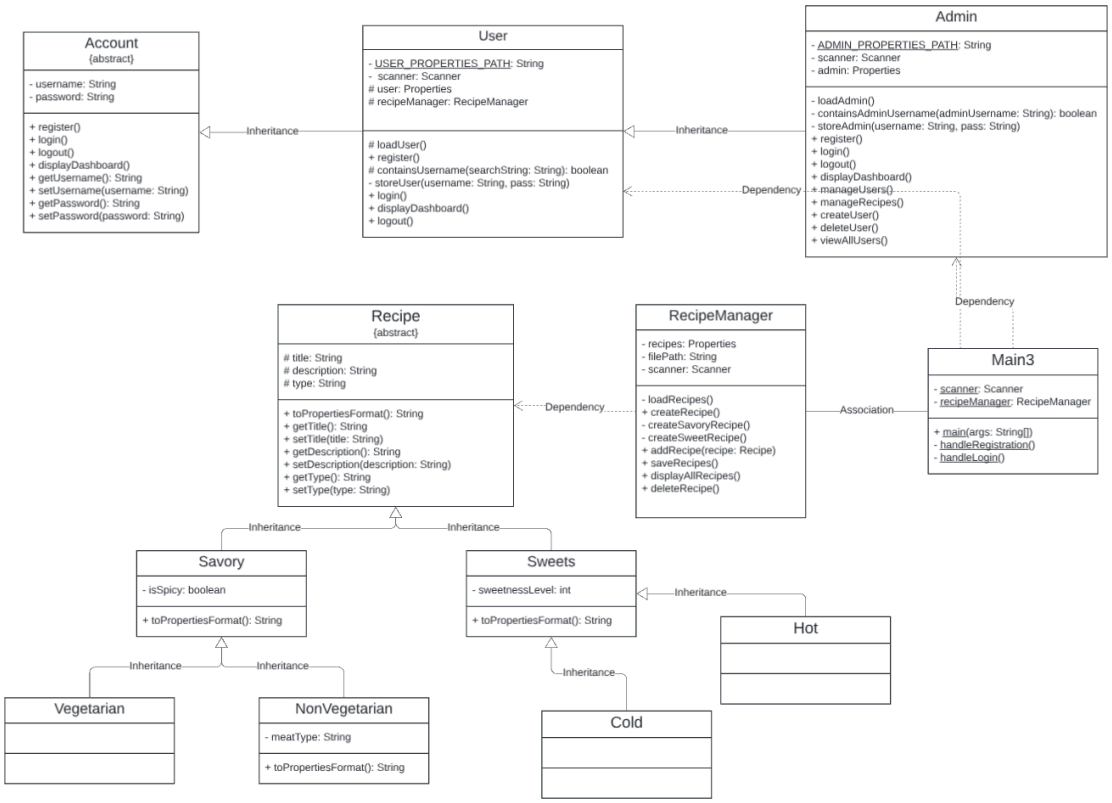
Project Objective/Deliverable:

This project aims to create a fully functional online recipe-sharing application in which users can share and explore recipes. We will be leveraging encapsulation, inheritance, polymorphism, and abstraction to create user sign-up and sign-in functionalities, recipe uploading, and recipe browsing to allow people to virtually connect over their shared love for food.

Project High-Level Design:

The purpose of this project is to allow users to share and explore recipes online. This will be done through a command-line application on Eclipse using Java. Users can register as regular users or admins, log into their accounts, and then add and browse recipes, with the admin having some extra functionalities on top of that to manage recipes and users. Inheritance will be used to make the process more efficient and allow for code reusability, and we use it in our Account, User, and Admin classes as well as our recipe, sweet, and savory classes. Our Account and Recipe classes are abstract, the Account class gives structure to the User and Admin class and the Recipe class gives structure for the Sweet and Savory classes. Encapsulation was used in our Account and Recipe class, as both have getter and setter methods which allow for the private methods to be accessed. Polymorphism is also used throughout our code, for example: the default and non-default constructors for the Account, User, and Admin classes, as well as the register(), login(), logout(), and displayDashboard() methods within these classes which are overridden.

Class Diagram

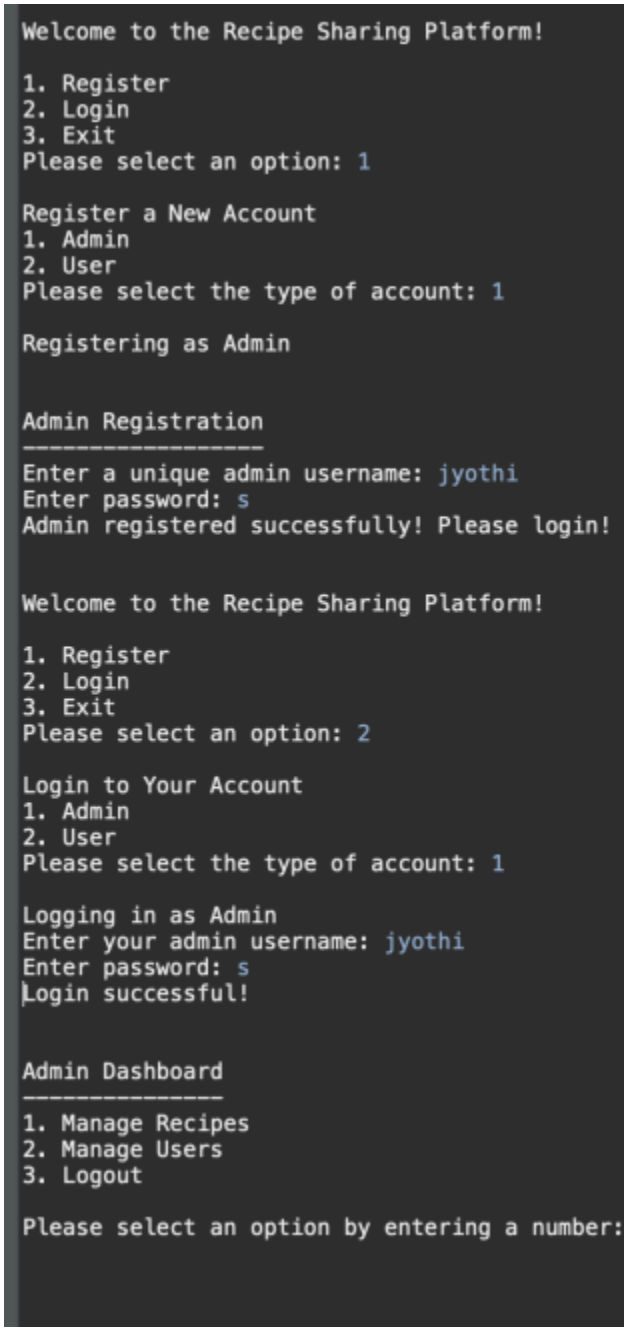


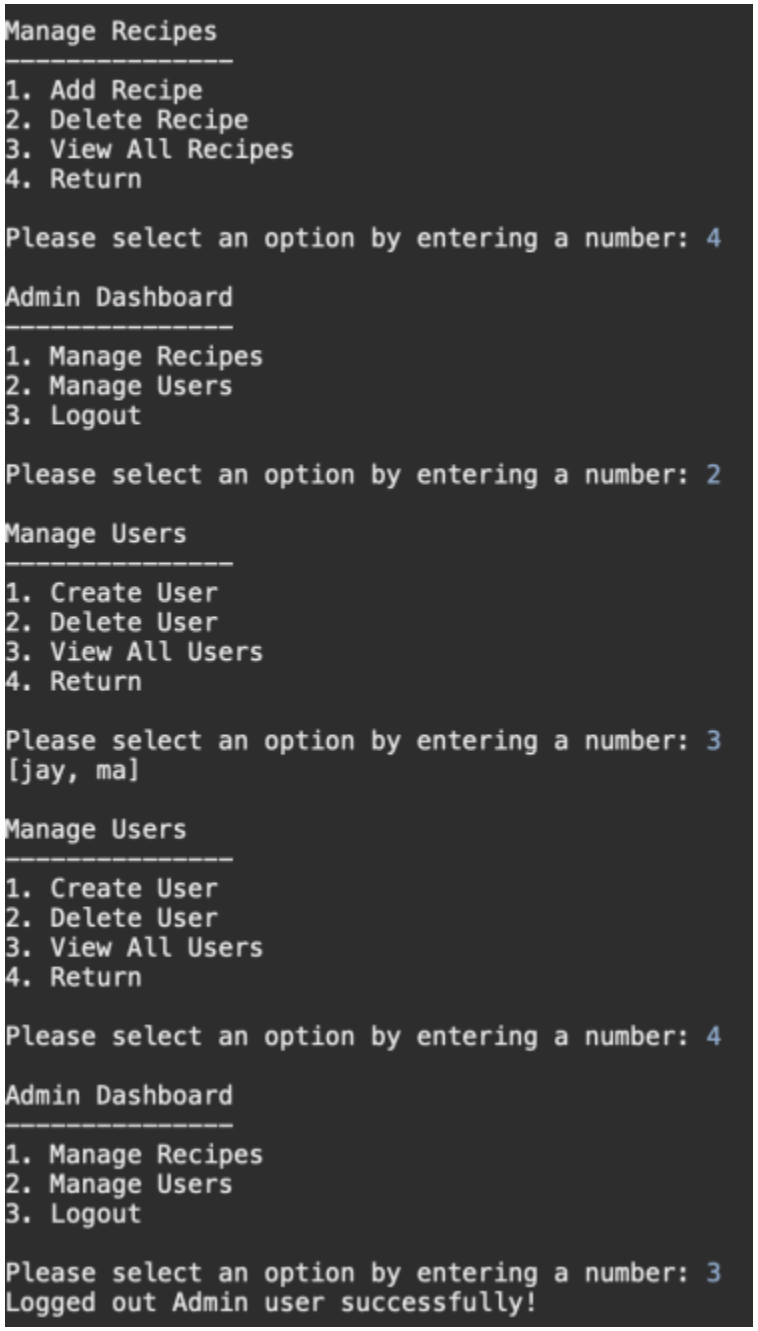
Sequence Diagrams

https://cdn.discordapp.com/attachments/1160053259302617139/1238621501721411624/image.png?ex=663ff3aa&is=663ea22a&hm=3b5d1a7891e15c65fcba19fc383adddf568970e4c9a7a1464a1bb41a5c2c056b&

(The picture of the diagram is too long, please click on the link to view it.)

Screenshots of Key Functionalities:





Contribution of Members:

Alisha: Led team meetings, kept us on track with our timeline for the project, worked on code and slides and created the GitHub repository

Jay: Worked on code, debugged issues in our code, worked on the UML diagram

Jyothi: Worked on the code and final deliverable report, and worked on slides and GitHub repository.

GitHub:
https://github.com/alishachowhan89/CS151ProjectFinalGroup11.git

LINK TO SLIDES(in this)