

CSCI 5448: Object-Oriented Analysis & Design

Team members: Saumya Bansal & Mehmood Ali

Project 7: Semester Project – Final submission

1. Name of project and names of all team members

Title: Recipe Recommender

Team Members: Saumya Bansal & Mehmood Ali

2. Final State of System Statement

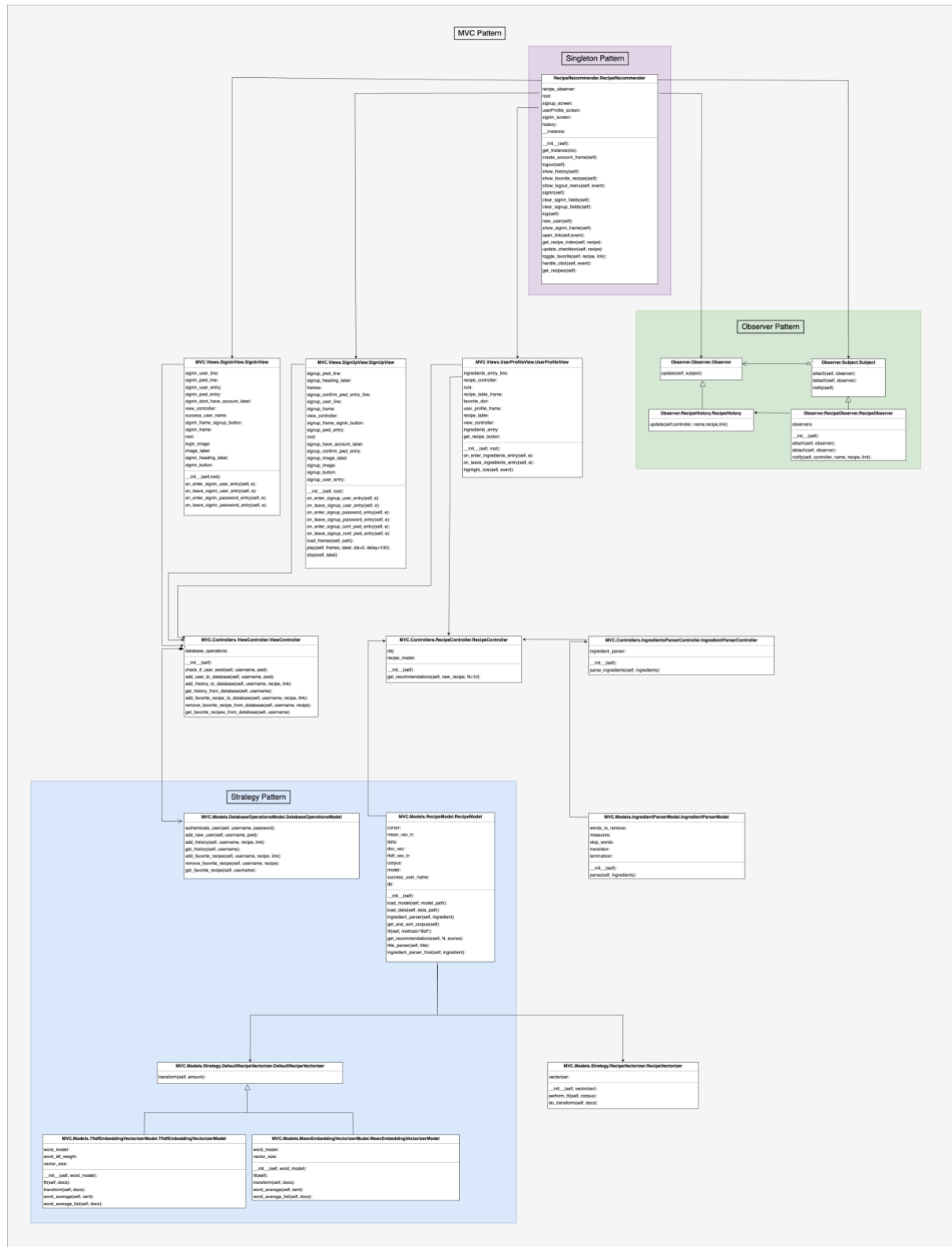
The system's final version has successfully implemented all the required features, offering users a seamless and user-friendly experience. User authentication is ensured through secure login and logout procedures, while user data is stored safely in the database. The user favorites page allows for easy access and removal of favorite recipes, while the user history page displays previously viewed recipes. The ingredient management system enables users to input and manage their available ingredients, allowing the system to suggest recipes based on those ingredients. Real-time recipe recommendations are provided through the system's recommendation engine.

Our desktop recipe recommendation application utilizes several design patterns to enhance organization, efficiency, and user experience. The Model-View-Controller (MVC) pattern is used to improve code readability, while the Strategy pattern embeds vectorizers and handles multiple techniques seamlessly. The Observer pattern updates recipe recommendations based on user preferences and ingredients, and the Singleton pattern ensures only one instance of the RecipeRecommender class exists, enhancing overall performance and efficiency. These patterns contribute significantly to our application's robustness, efficiency, and user-friendliness.

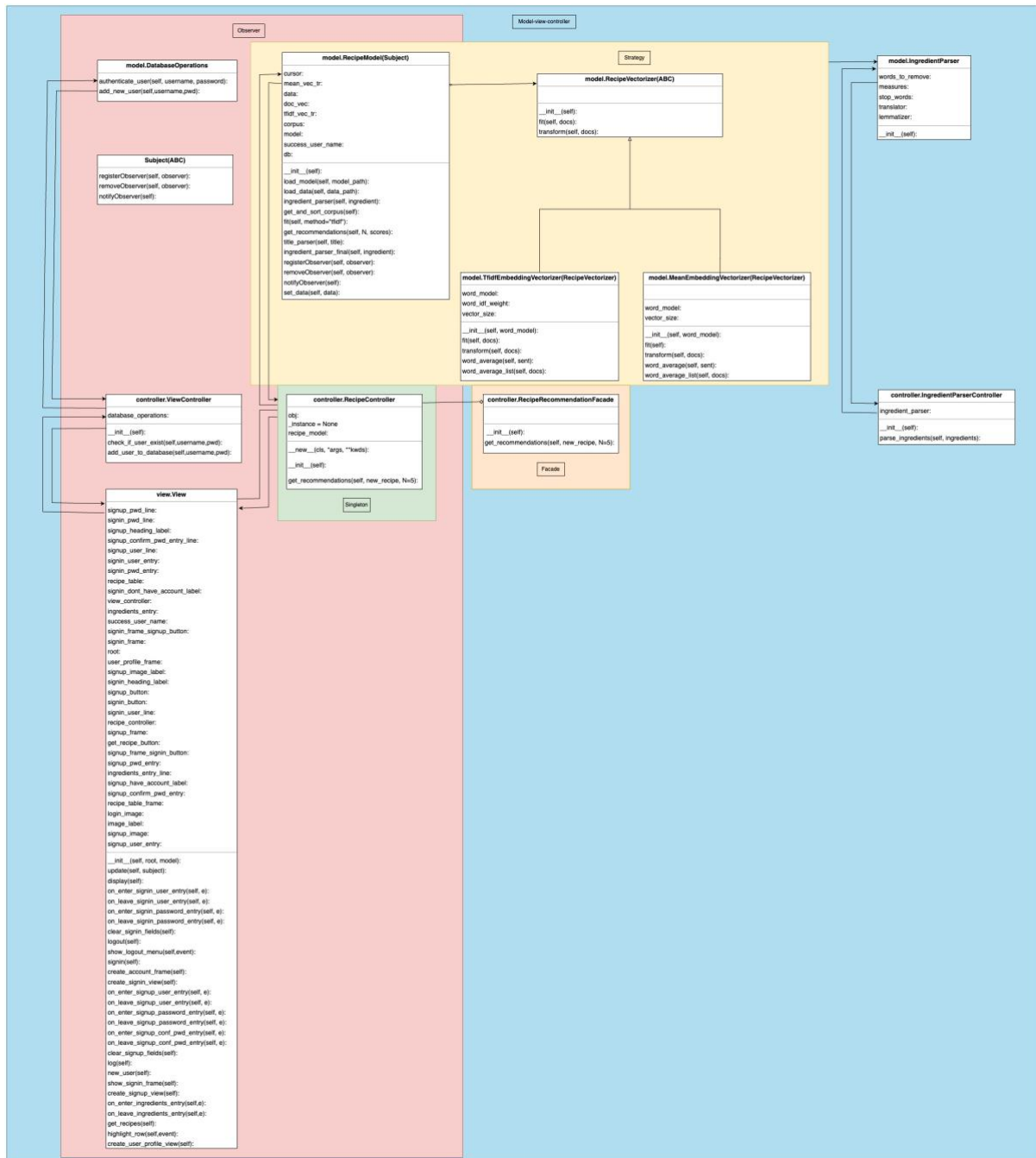
However, the Dietary Preference Management feature, which would have allowed users to input and manage their dietary preferences such as vegetarian, vegan, and gluten-free, was not included in the prototype as a part of Project 5 and 6 features. This was because the dataset did not have enough dietary specification details, which prevented the system from providing accurate and relevant recommendations based on users' dietary preferences.

3. Final Class Diagram and Comparison Statement

Final Class Diagram



Project 5 Class Diagram



Since the submission of the design/work in Projects 5 and 6, several key changes have been made to the system. Firstly, the views were split into SignInView, SignUpView and UserProfileView in the MVC pattern. This helped to further decouple the presentation layer from the business logic and allowed for greater flexibility in the development process.

Secondly, the observer pattern is being used to update RecipeHistory instead of being applied to the view controllers. This change improved the flexibility of the codebase by allowing observers to be notified of changes in the model and react accordingly.

Thirdly, the singleton pattern was previously applied to RecipeController, but it is now being applied to the RecipeRecommender class. This change improved the scalability of the

system by allowing the RecipeRecommender class to be accessed globally without creating multiple instances.

Lastly, the facade pattern was not implemented in the system as previously designed.

4. Third-Party code vs. Original code Statement

All the code in the project was written by our team, and no third-party code or frameworks were used. Therefore, there are no sources (URLs) for third-party elements.

5. Statement on the OOAD process for your overall Semester Project

During our OOAD process for the Recipe Recommender project, our team faced several design process elements and issues that had a significant impact on our analysis and design:

- One of the most notable challenges was defining the project's scope and determining the appropriate use cases, features, and functionalities that would satisfy our users' requirements while also being achievable within the given time frame.
- Another essential element of our design process was choosing suitable design patterns to ensure that our codebase was well-organized, maintainable, and scalable. We carefully evaluated different design patterns such as MVC, observer, and singleton and adopted those that were most appropriate for our project's specific needs.
- Finally, manual testing played a crucial role in our design process. We conducted extensive testing of each feature to ensure that it functioned as intended, and we promptly resolved any bugs or issues that emerged during testing. This helped to guarantee that our final product was both functional and user-friendly.