MVVM Architecture

During Sprint 1, our team utilized the Model-View-ViewModel (MVVM) architecture to structure our Android app in a flexible and accessible manner. We carefully separated our codebase into the three distinct layers: Model, View, and ViewModel. The "Model" layer consisted of our data logic, including classes representing entities such as users, recipes, ingredients, pantry items, and shopping lists. These model classes contained data structures and logic to manipulate and manage our application's data. The "View" layer consisted of XML layout files that defined the visual components and UI structure for each screen of our app. Meanwhile, the "ViewModel" layer contained Java classes responsible for interacting with the model layer and preparing data for display in the UI. By sticking to MVVM principles, we were able to address separation of concerns, improve code maintainability, and make testing of our application components easier. Additionally, the MVVM architecture allowed for smoother collaboration among our team by providing clear guidelines on code organization and responsibilities. Overall, the usage of MVVM in Sprint 1 laid a solid foundation for the succeeding development phases of our project.