Recipe Finder

Aaron Liauw, Jane Libby,
Kayla Terzioğlu, Afnan Alobaid, and Esther
Attar

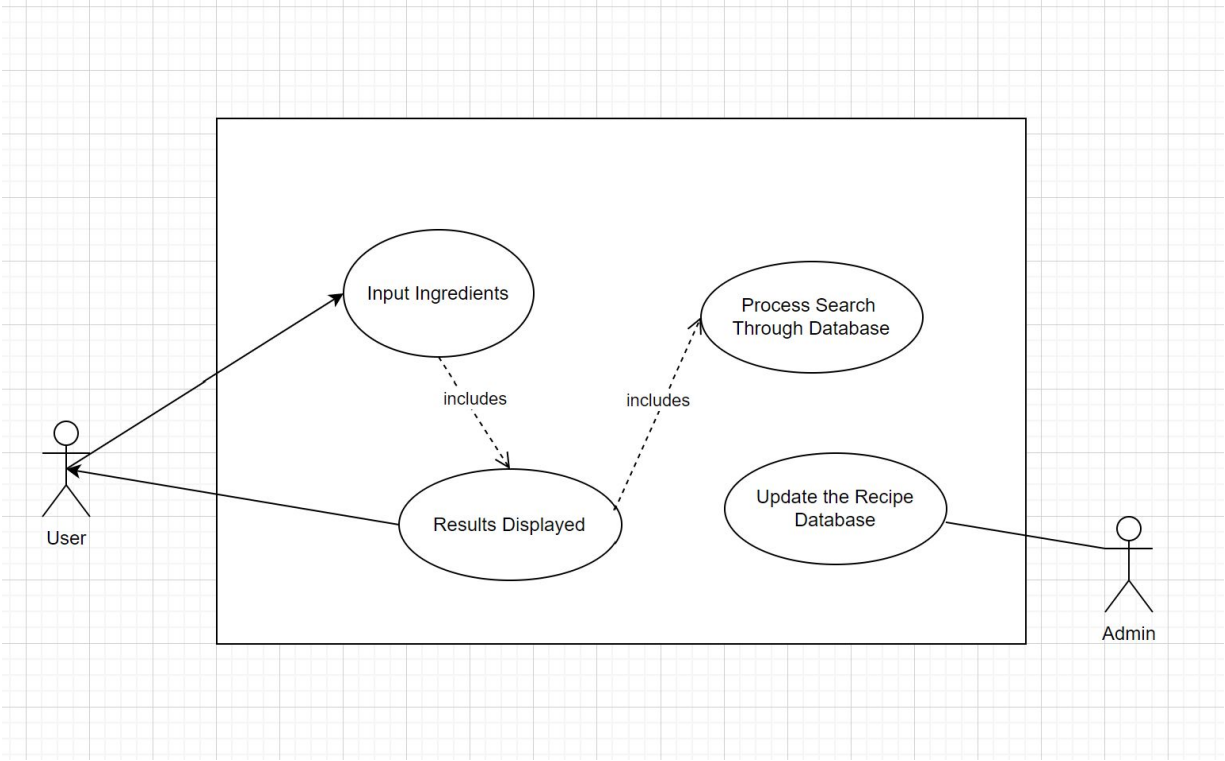
Problem Definition:

The Recipe Finder is a Java Application that allows its users to search through a database of recipes by ingredients. Recipe Finder is designed to make the lives of home cooks easier by optimizing the process of finding a recipe. Whether the homecook is a college student looking to make an easy meal, or a full time chef who doesn't want to waste any remaining food in their fridge, Recipe Finder allows them to use up the ingredients they have in their home without needing to go buy more. This will save the user time, money, and will help reduce their daily food waste.

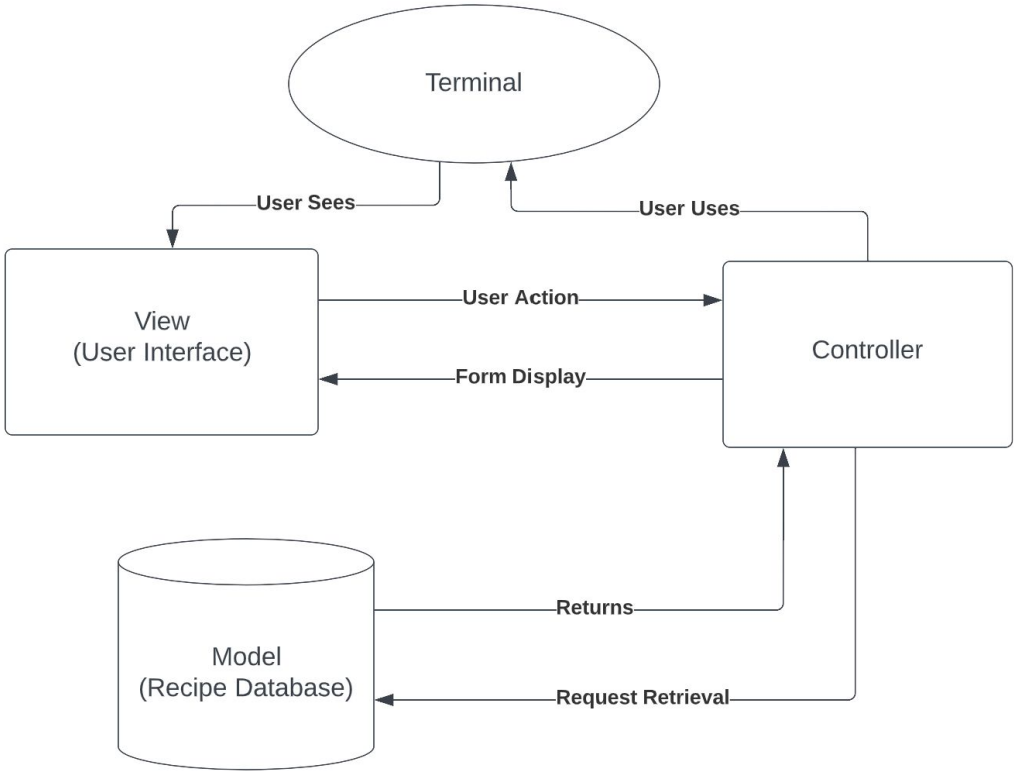
Task Breakdown

Task:	Task Owner:	Support:
Problem Definition	Kayla	
Problem Objective	Aaron	
Stakeholders List	Kayla	Aaron
Success / Acceptance Criteria for each Stakeholder	Aaron	Jane
Use Case Diagram	Aaron	Jane
Detailed Use Case Description #1	Esther	Afnan, Kayla, Jane, Aaron
Detailed Use Case Description #2	Jane	Afnan, Kayla, Aaron, Esther
Sequence Diagram #1	Jane	
Sequence Diagram #2	Aaron	
Architecture	Aaron	
Class Diagram	Jane	
State Machine Diagram	Esther	
ER Diagram	Kayla	
Task Assignment Matrix	Kayla	
Website Code	Afnan	
SQL Code	Kayla	Afnan
Work Breakdown Structure	Esther	
Conclusion	Jane	

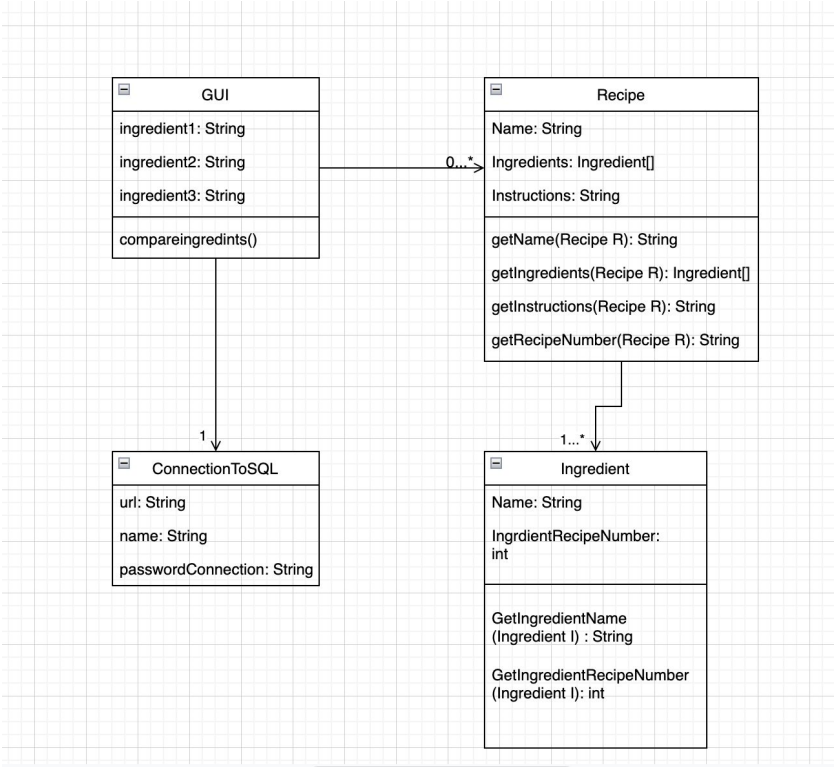
Capabilities: Use Case Diagram



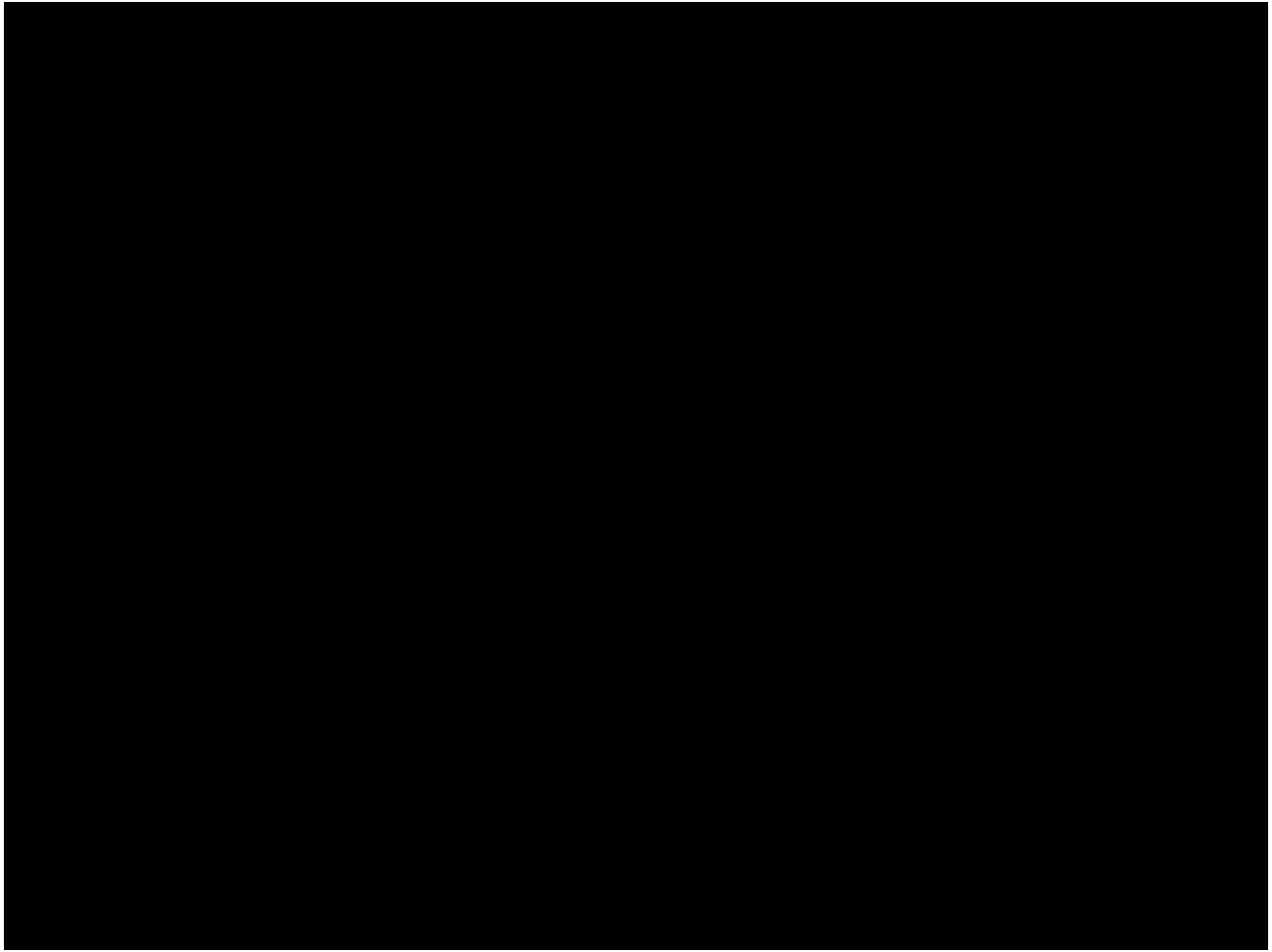
System Architecture: Model View Controller Pattern



Application Structure: Class Diagram



Demonstration:



Conclusion:

This was a valuable experience in learning how to properly plan a software project. Most of our team had little experience approaching software development in such an organized and structured fashion, which meant that the planning process was rather chaotic, yet fluid, as we were all still learning how to approach the project in an orderly way. However, the effort we put into the planning phase certainly paid off when we finally moved into the coding phase, as it gave us a much clearer understanding of what we were trying to build.

Additionally, we learned important lessons about how to plan a software development project, though many of those lessons came in the form of mistakes we made. These mistakes taught us that future projects would benefit more from a clearer, more ordered approach, as well as a willingness to engage with and understand the software we'd be using. We were forced to make frequent revisions to our planning documents and diagrams, in part because we had a somewhat unclear understanding of how to incorporate the database elements of our software. While this obviously presented us with some with this current project, this experience will allow us to avoid such pitfalls in any future projects that may come about, whether in school or in our careers.