

Jaydon Eppinette  
Joshua Constine  
Agustin Munoz  
Marc Rodriguez

## SDS 2.0

### Software Architecture Diagram:

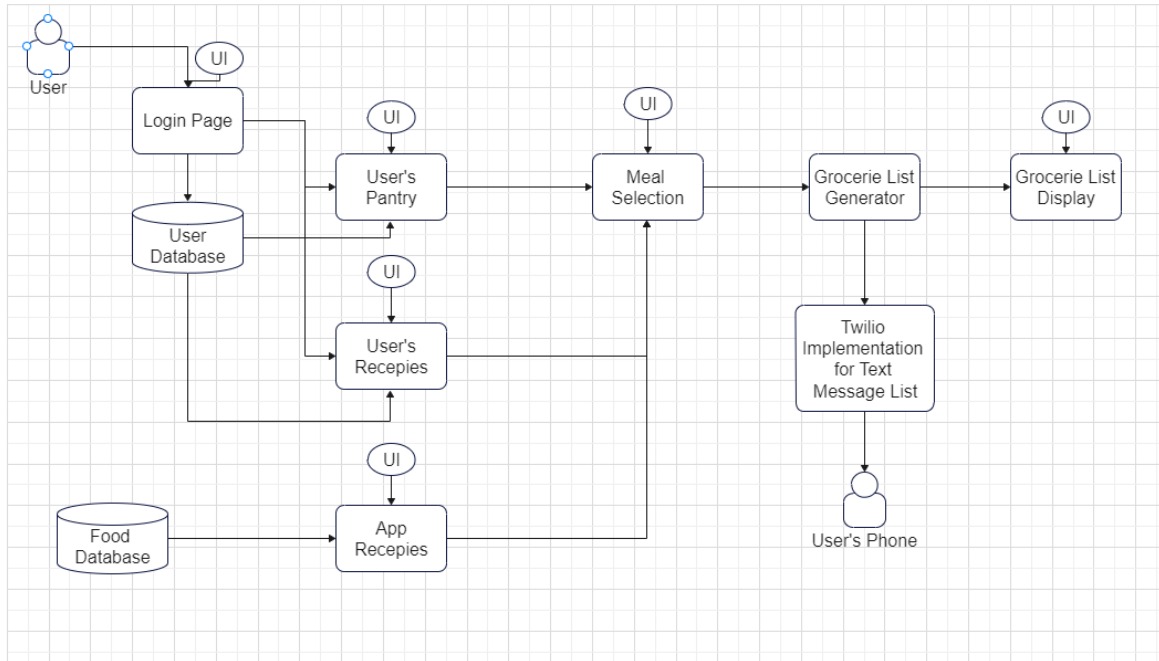
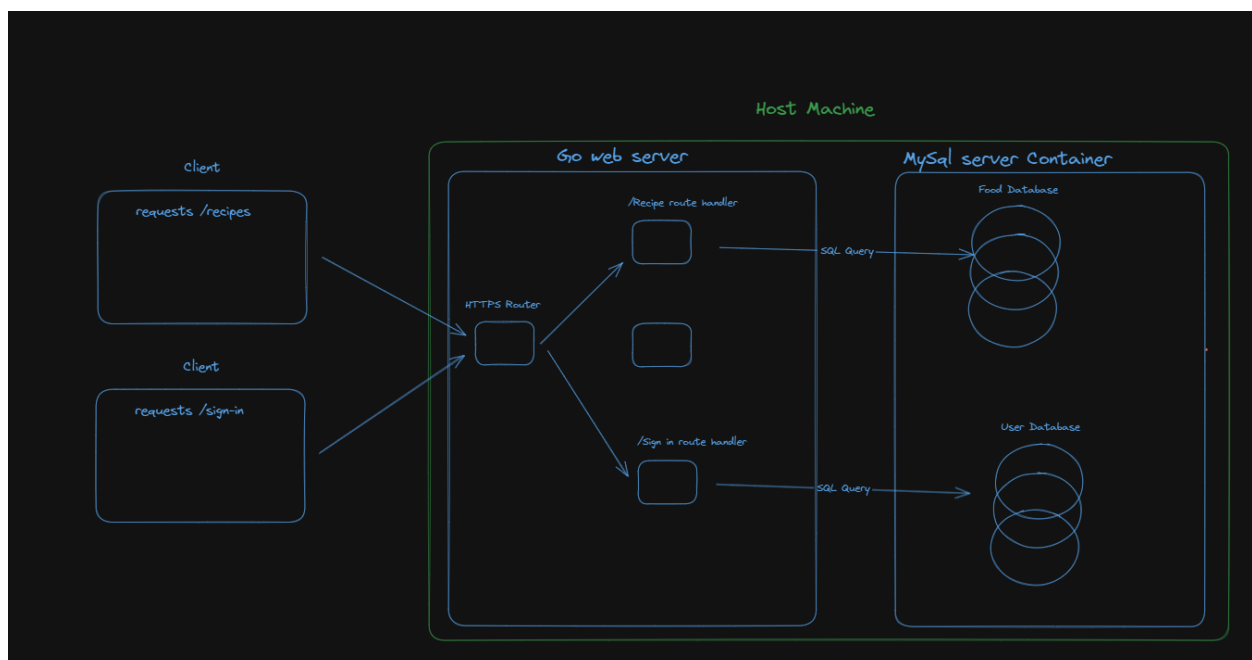


Diagram showing how users' interactions will be directed within the program and our databases:



## Data Management Strategy:

For our database we decided to use a relational database approach. For this project a relational database was a good option because we are working with a well understood schema. Our data fields relate to each other as well. For example, a recipe has many ingredients.

We will use a “User” database for the user credentials, which will have restricted access. The other tables will be in a separate database called “Food”.

In order to protect our users, PII will not be stored in plain text. All PII will be encrypted using a hashing algorithm.



SQL will be used on our project because it is highly portable, it provides good scalability, data integrity and consistency, and there are built-in security features that will help with user privacy and database

security. Although this comes at the cost of a more rigid database and partial control of the database.

In the context of the relational database at hand and the implementation of SQL, it is crucial to note that this will also allow us to join or merge certain tables within the database. Through this we are better able to understand the structure and role of relations found within the database. This is likely to lead to the prevention of certain errors and contribute to an overall better user experience for the stakeholders involved. With this being said it is crucial to note that the usage of SQL will come with a limited amount of access or flexibility within our data. However, despite this fact it is crucial to refer back to the idea of general scalability and integrity within our database.