**System Design**

I want my application design to be in a way that separates responsibilities into clear layers. This makes the application easier to understand and update, and it also ensures that any changes in the future will not affect the whole system. By using layers, I can fix or replace one part without breaking the others, which makes the system more reliable and maintainable.

First, I will manage how the user moves between different screens. For example, if I press a button on the Home Screen, it should directly go to my Detail Screen using a navigation route. I chose this design because separating navigation into its own layer makes it easy to add new screens later without rewriting the whole app.

Next, for storing data, I will use Firestore database to store all the data, called directly using Firestore’s built-in functions such as addDoc() to create new records, updateDoc() to change existing ones, and deleteDoc() to remove them. I chose Firestore because it updates data in real time, which means that changes made by one user will immediately show up for others. This ensures that the data is handled consistently, securely, and in a way that can grow as the number of users increases.

For managing a JSON file, I will load the data at the start of the application and store it in memory so it can be displayed quickly. I will also make sure that any updates to the data are saved back to the JSON file, so the information stays consistent for the next time the app is opened. For example, if my JSON file contains a list of recipes such as Pasta Carbonara or Ratatouille with their ingredients and cooking steps, I can show this information on the screen as soon as the app starts. If a new recipe is added, it will be written into the JSON file and appear again the next time the app is opened.

I chose JSON for this part because it is lightweight, easy to read, and very useful for storing sample data like recipes. I designed the system to focus on scalability, maintainability, and performance. Scalability is achieved because Firestore can handle small apps and still grows to support thousands of users without redesigning the system. Maintainability is achieved by separating the application into layers, which makes it easier to update one part without affecting the others. For example, I can switch from JSON to Firestore without changing the navigation system. Performance is improved because JSON is lightweight and loads quickly with static data, while Firestore only loads the data requested and provides real-time updates. JSON is also a good choice for managing fixed or sample data, such as recipes, because it is easy to read, fast to parse, and does not require an internet connection.

This design also gives flexibility for future growth. JSON can continue to be used for storing small static data such as sample lists or recipes, while Firestore manages larger dynamic data that changes in real time. By combining both options, the system can balance speed and reliability, making the application stronger as it develops further.