DailyApple
Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
29/11/2023	1.0	Initial Software Architecture Document for the project	Phạm Vũ Minh Giang
29/11/2023	1.0	Initial Software Architecture Document for the project	Võ Hoàng Phúc Khang
29/11/2023	1.0	Initial Software Architecture Document for the project	Vương Quốc Phong
29/11/2023	1.0	Initial Software Architecture Document for the project	Nguyễn Trịnh Duy

Table of Contents

Software Architecture Document

1. Introduction

We will present the whole architecture design of our Healthcare mobile app in this paper. It includes the use case model, which is a graphic that describes the functions of users (Administrator and End-User). In the next part (Architecture Goals and Constraints), we will provide a summary of crucial information about our project, such as Goals, Constraints, tools employed, and methods that we will employ to construct this project. The use-cases model will represent all of the key use-cases in our software. The last Section (Logical View) will display all of the software architecture information.

2. Architectural Goals and Constraints

1. Safety:

Goal: Develop a mobile application with a user-friendly interface, attractive UI design, and easy navigation.

Impact on Architecture: Prioritize the implementation of smooth and basic user interactions, ensuring effortless functions like clicking, scrolling, and typing.

2. Security:

Data Privacy: User meal data must be securely stored and encrypted to protect sensitive information.

Impact on Architecture: Ensure data safety with encryption and secure server practices, shielding sensitive health information from cyber threats in the app's architecture.

3. Privacy:

Goal: Ensure the absolute privacy of user data, preventing any leakage or unauthorized access to sensitive information.

Impact on Architecture: Implement strict access controls to safeguard user privacy and maintain the confidentiality of personal information.

4. Design and Implementation Strategy:

Goal: Follow RUP and SCRUM methodologies to enhance development efficiency and team collaboration.

Impact on Architecture: Structure the development workflow to align with RUP and SCRUM, promoting a collaborative environment where team members support each other to maximize overall program performance.

5. Stability:

Goal: Ensure the server's efficiency, particularly during high traffic situations, with a maximum one-minute response time for all requests.

Impact on Architecture: Design a scalable server architecture, incorporating load balancing to handle peak usage. Implement a notification system to inform users promptly upon task completion.

Development tools:

- We will use web draw.io to design the component diagram

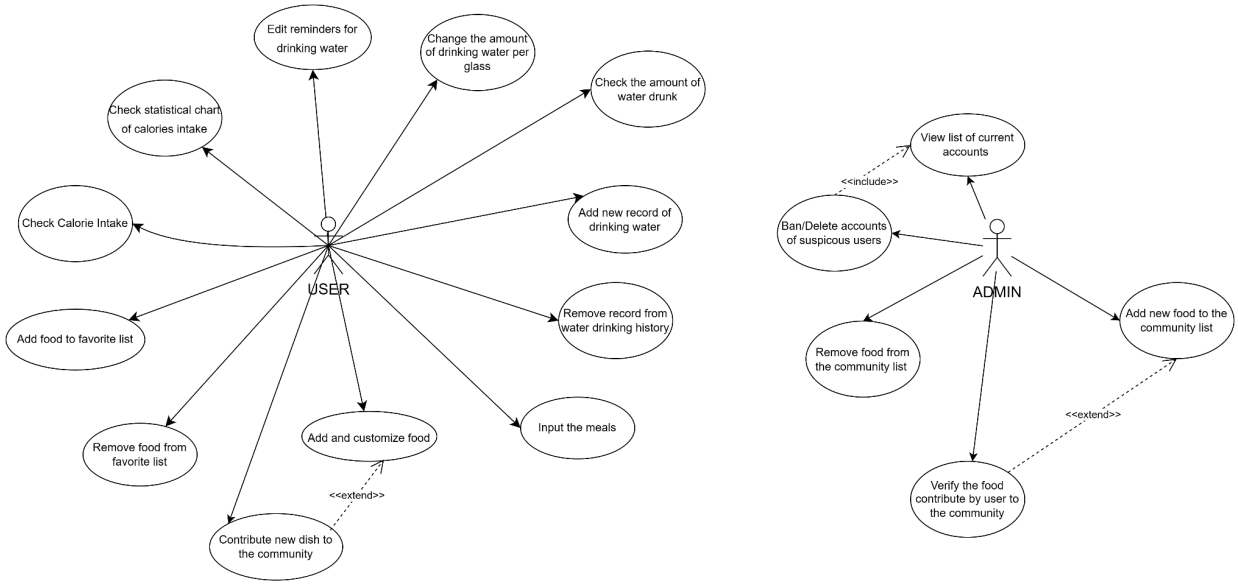
- For designing the UI of our application, we will use Figma.

- For designing the database diagram, we will use Firebase to construct the database diagram.

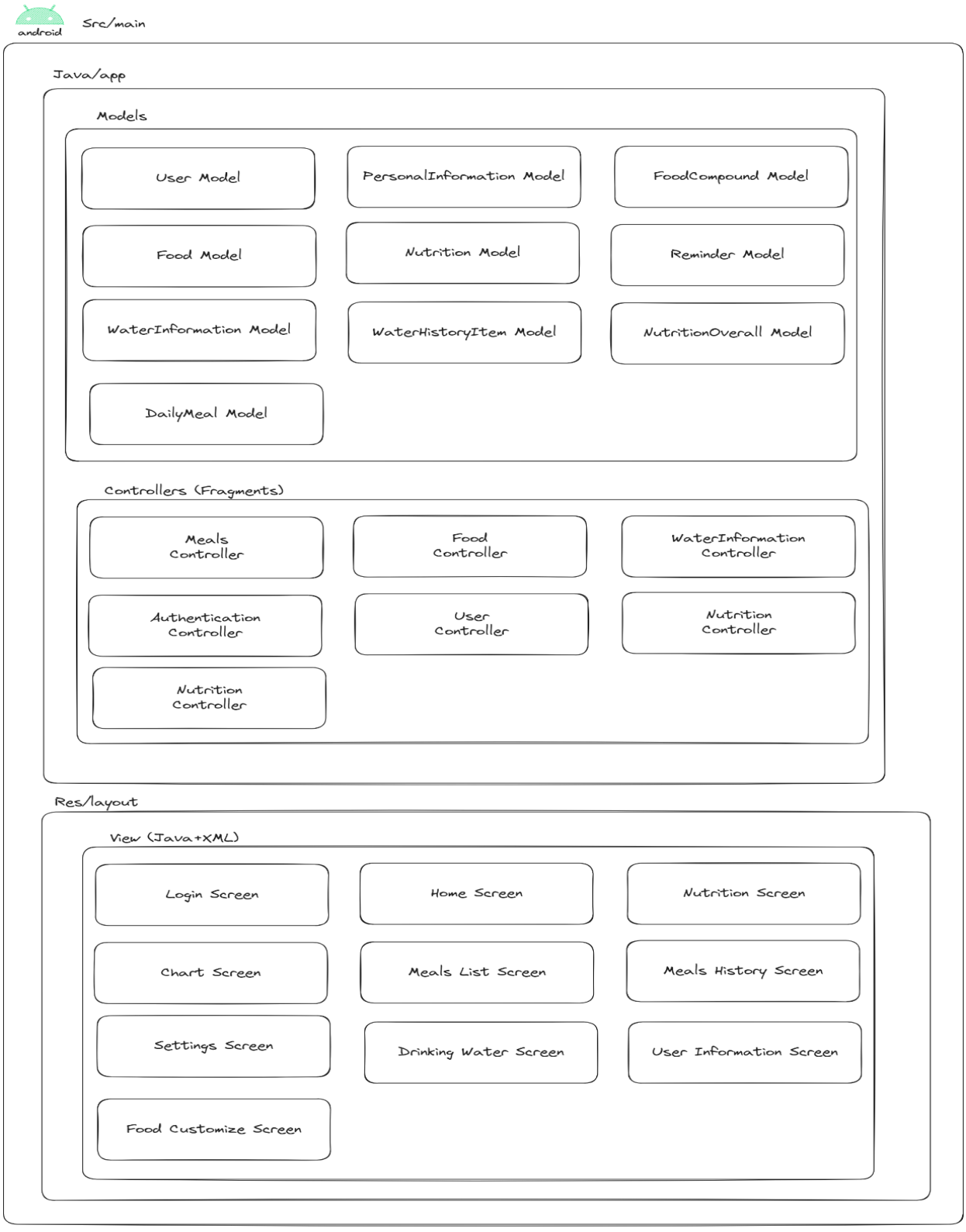
- We use Excalidraw for designing the package diagram following the MVC architecture.

- We use Java language to develop our application

3. Use-Case Model

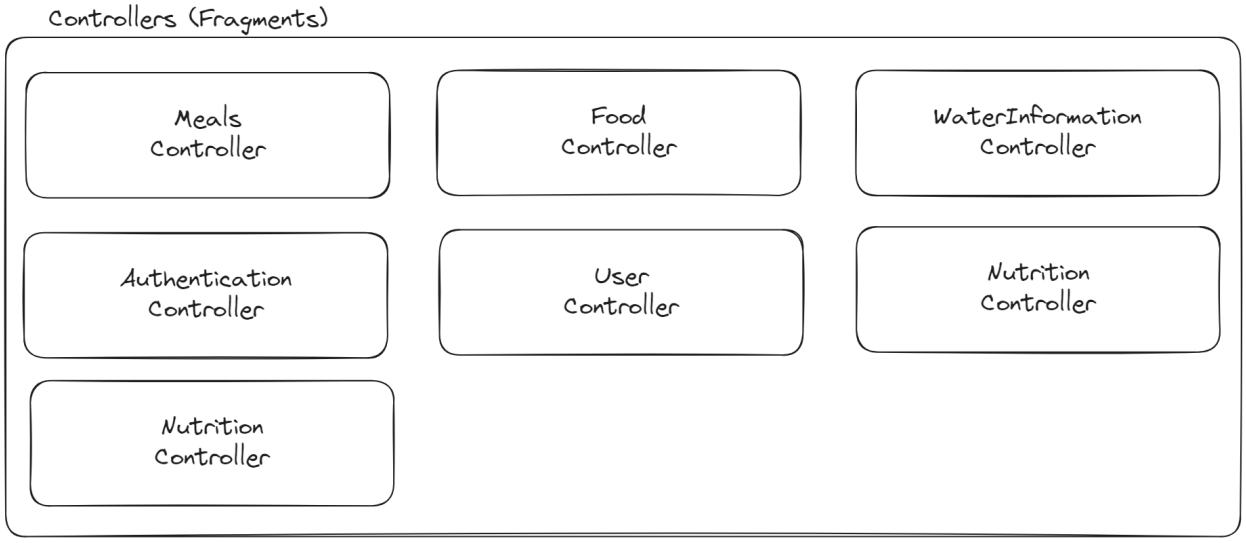


4. Logical View



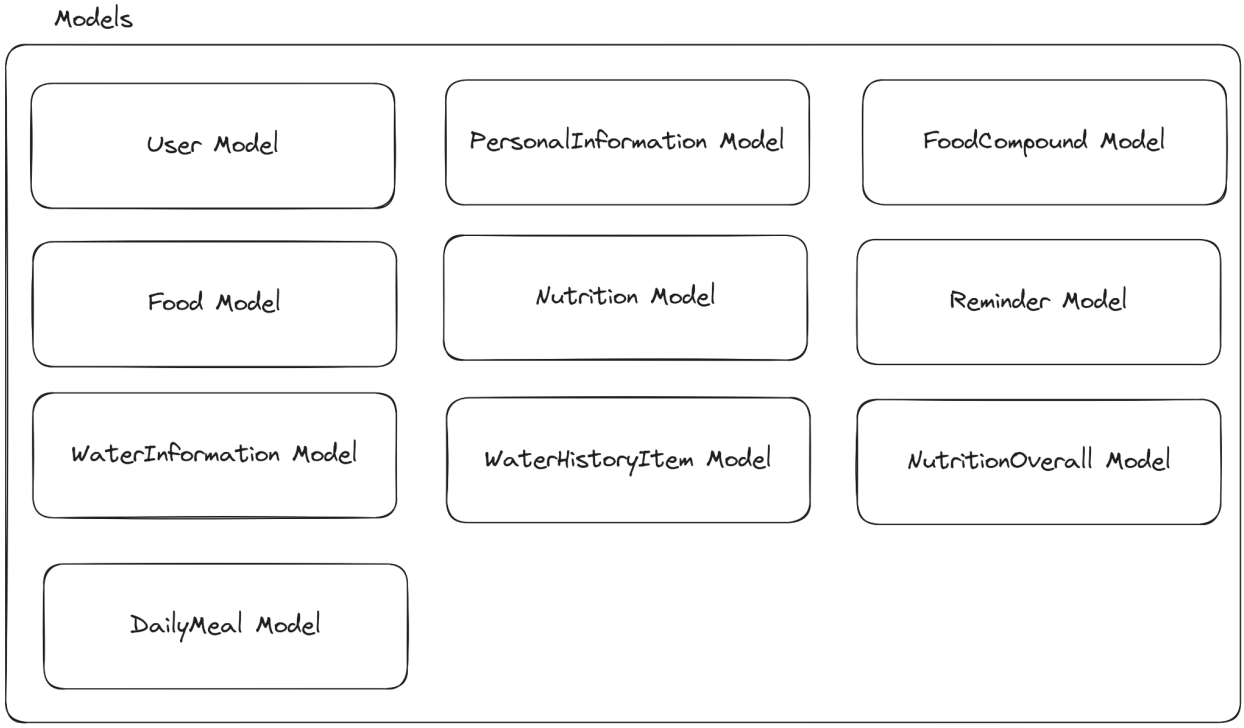
4.1 Component: Controller

Using Fragments to control events



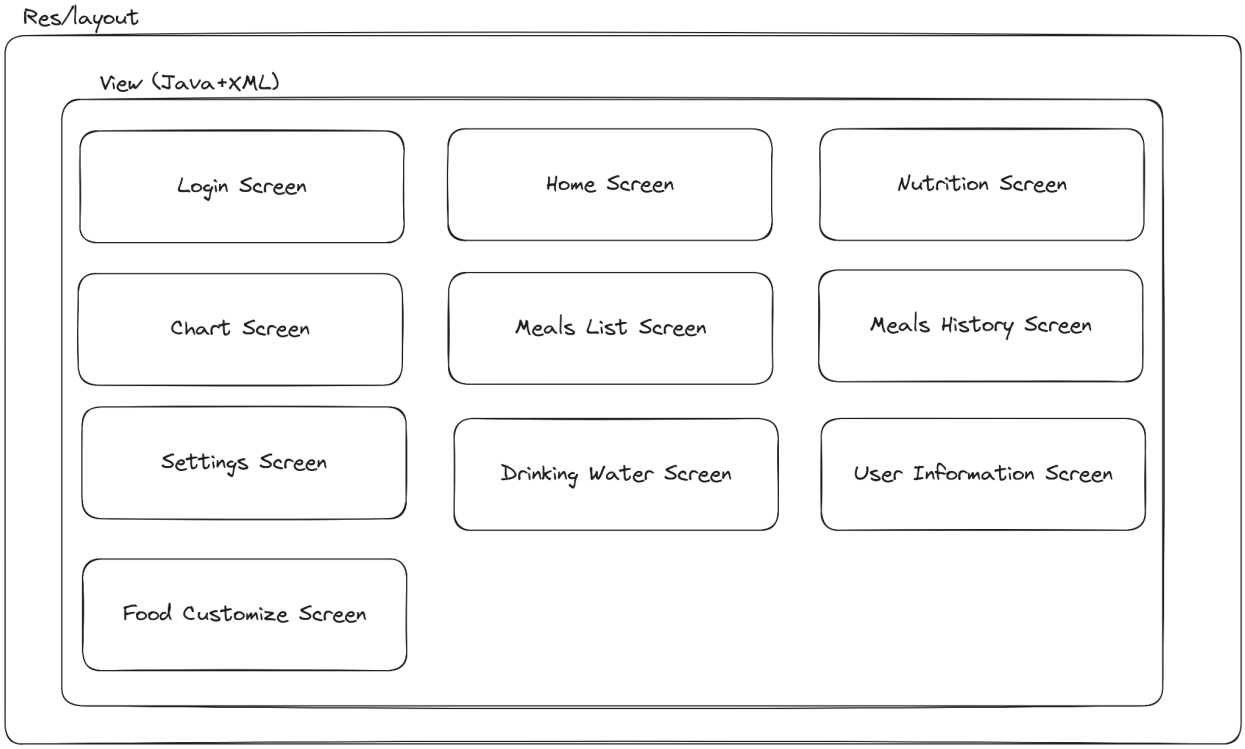
- Implement using Java language
- Authentication controller handles authentication requests (login, register, reset password)
- Food controller handles requests for updating and retrieving data from Food collections
- WaterInformation controller handles requests for updating and retrieving data related to the amount of drinking water
- Meal controller handles requests for mutating and retrieving data related to daily meals
- User controller handles requests for updating and retrieving data related to user information

4.2 Component: Models



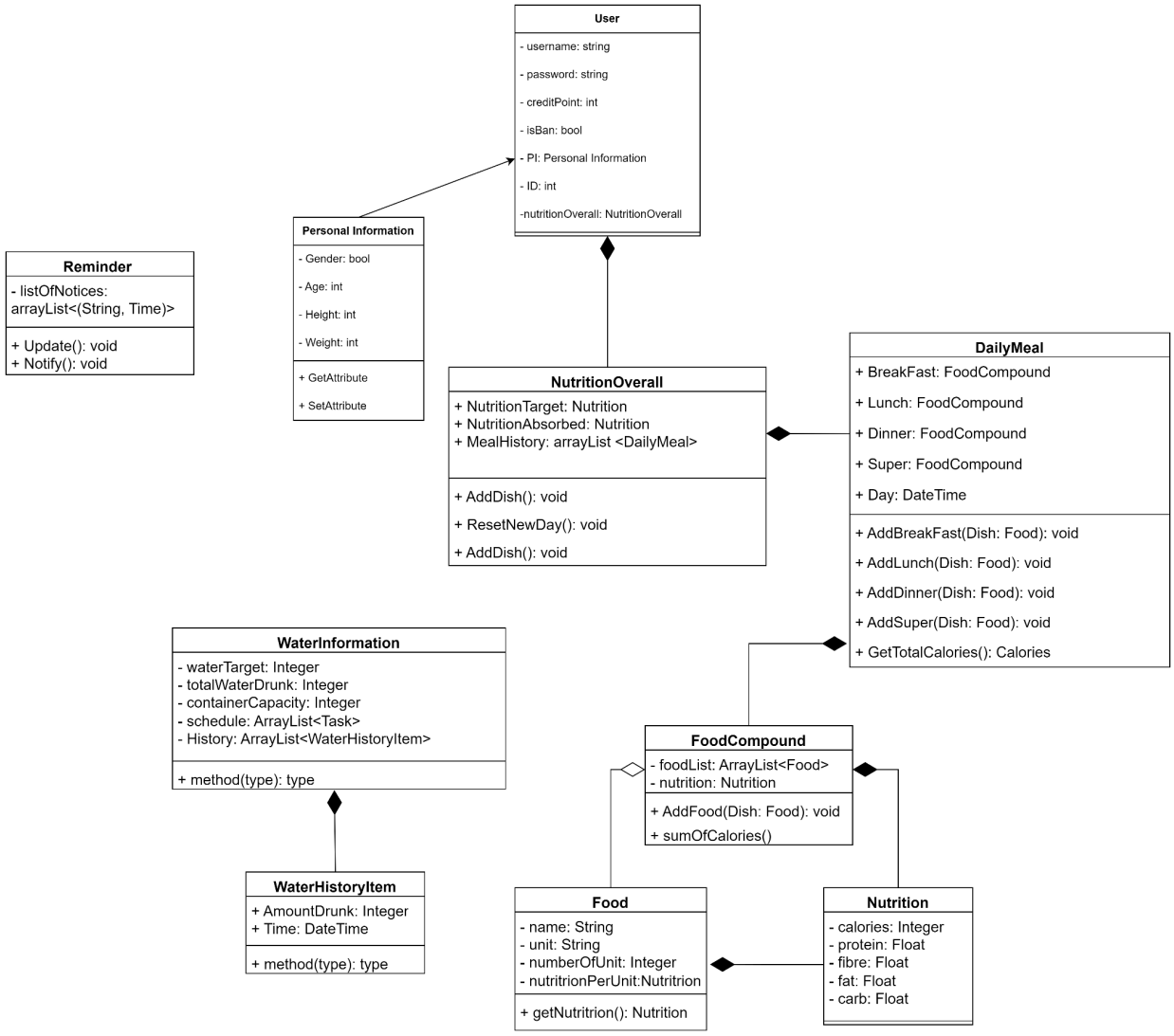
- Using Firebase to store data
- There are 10 models in total .
- They are implemented using Java language
- User, PersonalInformation, NutritionOverall contains user information, as well as meal information.
- Reminder model notices the user to drink water on time.
- Water Information, water history contains and stores drink schedules for users.
- FoodCompound, Food, Nutrition hold information about meals such as name, nutritional ingredients…
- DailyMeal stores information of meal in one day

4.3 Component: Views



- Each screen will display the UI which respects its corresponding view.
- XML is used to design the layout of the screen, and Java is used to implement the display.

4.4 Components: Schema



5. Deployment

6. Implementation View