1. **Architectural Styles**

As a web application, our system is based on the Client/ Server model, which benefit the system for centralized data storage, and a scalable amount of users accessing and providing data. The clients of the system are different applications run on different user machines, which allow users to login the system, upload their monitoring data, and get or give the data analysis and interactive with social network. While on the other hand, the server which runs on a separate machine is in charge of the system database and all the backstage requests process. The server can be further partitioned into several basic functional units, which in charge of data management, data transit, and user application communication.

By using this client/ server model, the system reduced the procedure done on the client side to a minimum degree, and leave all the process work to the server. In this way, we can provide a light user application, also a better coherence with all the user data. We could have a light front-end application for user to run fast on most device. But on the other hand, it requires a powerful server to deal with and store lots of data effectively. The centralized architecture make the server act as both the database and the central data processor in whole system.

1. **Identifying Subsystems**

We can basically divide the system into two subsystems: the Client side and the Server side. The Client subsystem is consisted of Web Brower and other User Interfaces, such PHP, HTML and XML files. As for Server side, there is a central controller unit to control the whole works inside server side. User authentication unit can validate user information and send user request to central controller. Then the controller forwards the request to Server Logic unit to parse and process it. After analyzing request, the central controller will connect to Database. The Database package is responsible for data storage and providing external data such as Social Network information.

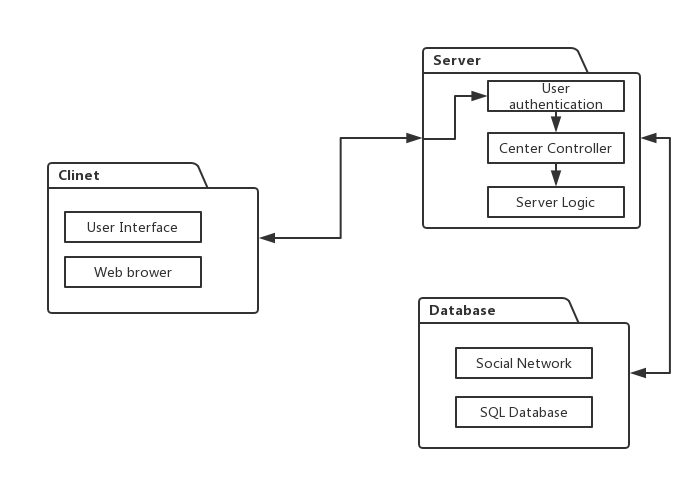


Figure UML Package Diagram

1. Mapping Subsystems to Hardware

As mentioned above, we can map the subsystems to the following hardware.

Client Subsystem: Allocated in user’s device, such as PC and intelligent mobile phone. The web browser can be run on different user machines.

Server Subsystem: Allocated in one powerful server. We will use a PC as server (For current stage)

1. Persistent Data Storage

We use MySQL database for the persistent data storage. MySQL Database is an open-source document database and one of the best Relational Database Management System in WEB application. It gives great support to the PHP. It has multiple storage engines, allowing one to choose the one that is most effective for each table in the application.

We will store health data, custom diet plan data and social information data in MySQL database. Different type of data has different schema. We use PHP to retrieve and save data in Server. The health data keeps healthy records which get from personal health monitor. It can be read both by user and analyst. The custom diet plan contains a food list which chosen by user. At last, the social network factors are saved in social information data, such as rank among all friends.

Below are schemas of persistent data:

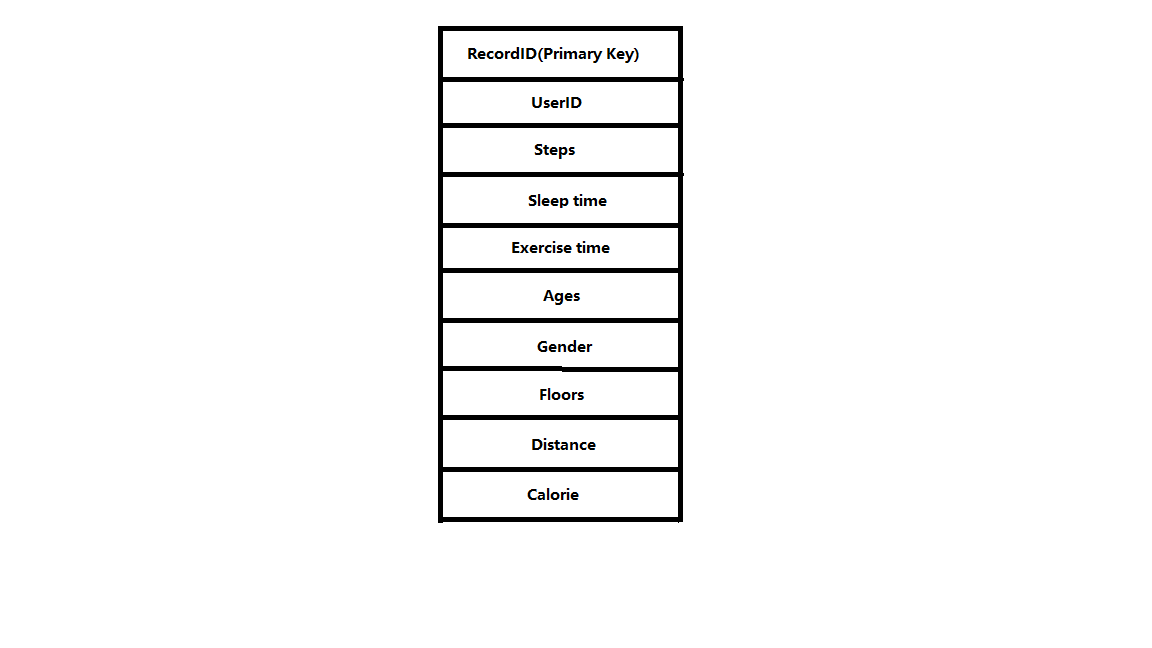


Figure HealthData

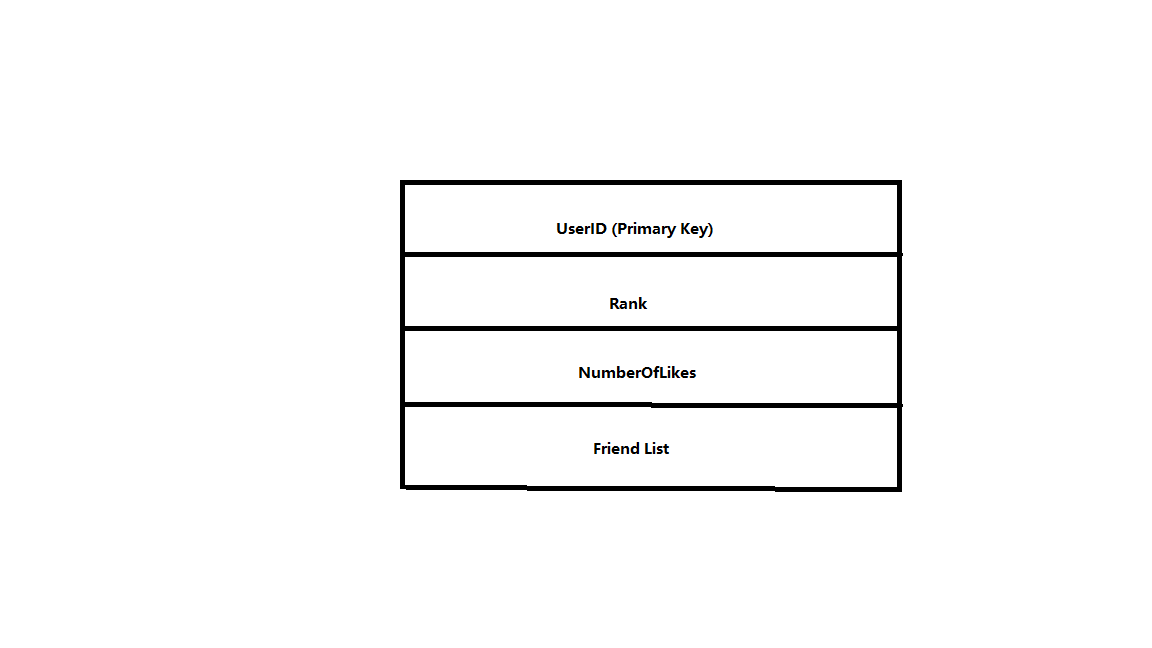


Figure CustomDietPlanData

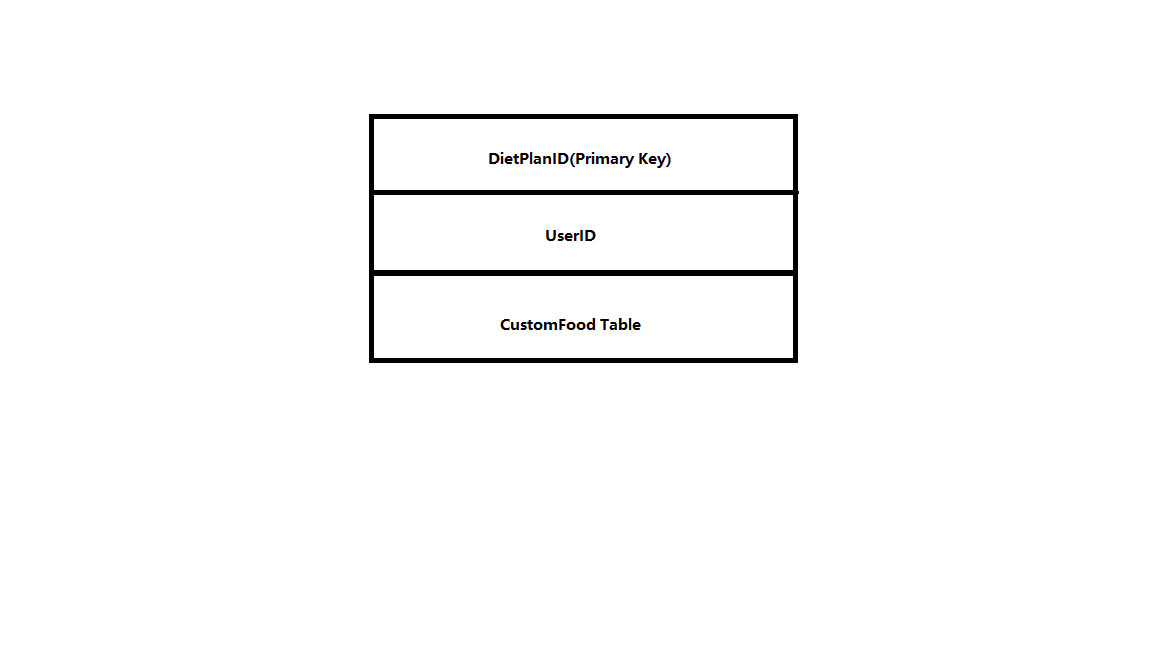


Figure SocialData

1. Network Protocols

There are many network protocols such as FTP (File Transfer Protocol), HTTP (Hyper Text transfer Protocol) and SSH (Secure Shell). In this system, a user uses the website to log into his/her personal account and access the software user interface. Since we have a web design, we need to use HTTP so users can navigate to the website from their personal computer.

Also, we will use SQL through PHP to exchange data from database

1. Global Control Flow

This system is an event-driven, which means that the system must respond to different events at any time. When user log in, he/she can send request by clicking or typing at any time, and the server will respond to his/her request, accept or decline, and display corresponding page to user.

1. Hardware Requirement

We require Fitbit health monitor to get healthy data. The operating system of our system can be Window XP, Windows 7, Windows 8 or Mac. The hard drive storage needed should bigger than 3 Gbytes. The backend will based on the PHP using the web server xampp and Matlab, which make our computer as both the client and server. Fitbit Database and MySQL Database are the databases for our system.