RUTGERS

School of Engineering

# Software Engineering

# Personal Health Monitor

## Report I


Group 4

Chenfan Xiao, Chengyao Wen, Jianing Xu, Xinyu Li, Yuwei Jiang

# Content

# 1. Customer Statement of Requirements

## a. Problem Statement

Personal health monitoring can be useful for individuals: providing them with professional and highly-customized food and sports suggestions, motivating them to keep active and have a good habit. Also it gives a lot of potential important information by analyzing the health data collected. It can be used by public health specialists to provide more effective policies in a big way and by manufactures to make products that could really help.

From the customers' views, the system can help them track and store their activity data and recommend activities or diet for them. They can also share their data with friends once they log in. They will also be able to read analysis and find the most important attributes to their personal health.

As a customer, I do care about if my daily data, including activity data and diet data, are stored to the server correctly. I also want to see a plot showing the change of my personal data. What else is I may want to know what kind of diet plan should I have according to my activity data and personal goals. Plus, I will care about how much activity my friends do.

As an analyst, I want to know if certain variable has anything to do with another one. I also want to know if users from different regions or have different activity data will have any change in their general information in the future. I do not and can not know their sensitive information individually.

Based on the analysis above, we will have these problems as follows:

Problem 1: Collecting data from wearable devices or health applications with small time interval
Problem 2: Users information or data should be kept safe
Problem 3: Users will be given diet / activity recommends according to their personal data
Problem 4: Users will interact with their friends
Problem 5: Users will request visualized plot for their own data
Problem 6: Analyzer will apply certain algorithms on the data collected

## b. System Statement

In our project, we will design a system to keep track of users' health data and provide them with valuable suggestions to achieve their goal. In the background we will design a powerful analyzing system to give feedback about any difference so that several main factors resulting in the difference can be found. Typical suggestions for users including food diet based on their active status. One of the information by analyzing them is whether or what kind of wearable devices can really help. Generally speaking we are going to design a system which will be useful for both users and the ones not using it. Figure 1 shows the overview of our designed system.

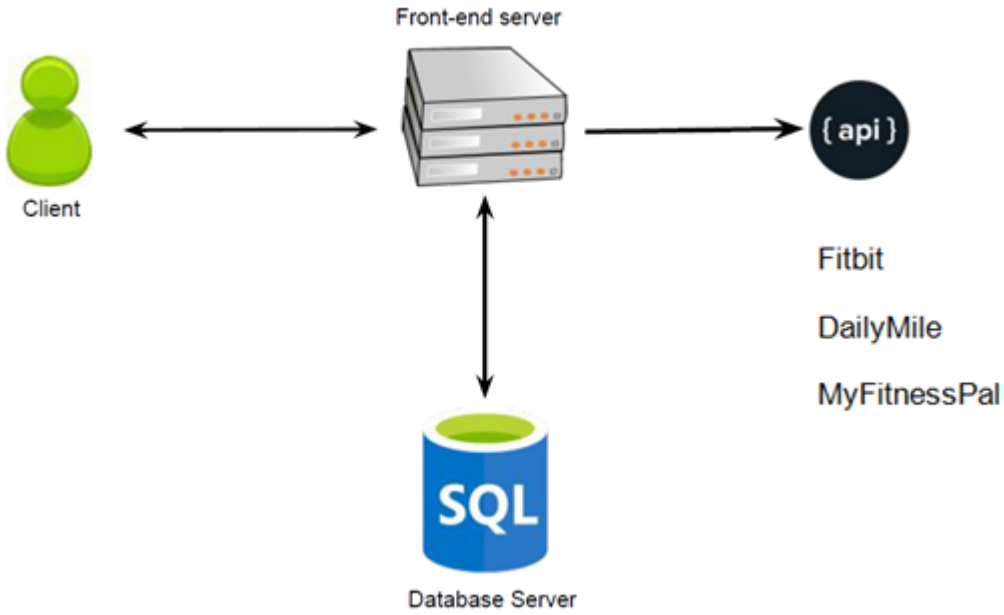



Figure 1

The system will collect the data from wearable health devices - fitbit - using the API provided by the manufacture. Also it will be collecting data from integrated health interface like Apple Health. The data will be uploaded or synced to the server and shared to the client's friends if permitted. The clients will get visualized graphs of their progress and target daily, weekly and monthly.

We are going to imply APIs from the internet as the specialist to give recommendation to the users. We are still considering using single or multiple API to generate recommendations, because sometimes, there is no definite solution to the healthy diet, not every API would get identical result. So what we are trying to do is to add a function in our system: the system would keep a value to represent the weight of each API we utilized, and the users are presented with different recommendations generated by different API. As a result, certain API would get a increased weight when chosen by the users, and the most weighted API would be labeled as 'Preferred' which would make it easier for the user to decide among various options.

The fitbit API will mainly provide the following data: Activity & Exercise, Body & Weight, Friends, Heart Rate and Sleep information. It can be used to track users personal activities and body information. Data will be stored into our databased for further analysis. For recommendation, we will use the API from DailyMile, which will provide us the following functions: Track running route and GPS information and Social functions. With the API from MyFitnessPal, we will be able to access over 20k kind of foods' information, which can be used for our customized food recommendation system. When the APIs have overlaps, we will use the system above to decide which one is most recommended to clients.

It will be possible for our users to share pictures with others who are not their friends in real life and can share their working out data like steps, running time or active hours with their real friends. Because according to our research, it works better if strangers share pix of themselves while acquaintances share data with each other. This will contribute them to use our system more often and keep track of their health information and be of great help for them to achieve their goal. Also it can provide us more accurate data for foods and sports suggestions as well as background health information search for specialists.

We intend to use PHP and MySQL for our website and we may also develop an Android app to collect users' health data more efficiently.

## c. Glossary of Terms

Activity: what kind of activities we can monitor, including but not limited to, running, walking, hiking, swimming.

Dailymile: Track running route and GPS information and Social functions.

Fitbit :provide the following data: Activity & Exercise, Body & Weight, Friends, Heart Rate and Sleep information.

MyFitnessPal: be able to access over 20k kind of foods' information, which can be used for our customized food recommendation system.

Diet: what kind of diet customer can get. For example low-calories food for customer expecting losing weight or high-protein food for those wanting to gain muscle.

Customer/Clients: synonym here, referring to personnel who upload their activity data via wearable devices or applications from their phones.

Analyst/Analyzer or Specialist: referring to those who have access to anonymous data collected by system and run filters or clusters to make some conclusion from it.

# 2. System Requirements

## a. Functional Requirements

| Identifier | Priority | Description |
|---|---|---|
| Req-1 | 3 | The system should attain raw data from the Internet and save it in a local/online database. |
| Req-2 | 5 | The system should process raw data from the local/online database and obtain the required analysis. |
| Req-3 | 2 | The system should allow users to register and provide personalized services. |
| Req-4 | 4 | The system should show the recommendation to the user about the healthy diet. |
| Req-5 | 2 | The system should obtain information from other social service platform. |
| Req-6 | 1 | The system should allow analyze give suggestions about how helpful it is of certain behavior which is commonly believed as helpful. |
| Req-7 | 3 | The system should allow analyst provide analysis of health conditions of users from different areas. |
| Req-8 | 2 | The system should show the rank of a certain user about his health among his or her friends as well as among the entire users. |
| Req-9 | 2 | The system should enable the user to customize his or her diet plan. |
| Req-10 | 1 | The system should provide user's friend health goal with permission. |
| Req-12 | 2 | The system shall allow users to invite friends to the system by sending emails. |
| Req-13 | 3 | The system shall allow users to share information of their exercise activities to social networking websites such as Facebook. |

| | | |
|---|---|---|
| Req-14 | 3 | The system should show the rank of user in his or her area(can be state,city or even around the house). |
| Req-15 | 5 | The system should allow analyst provide analysis about the major contribution to health. |
| Req-16 | 4 | The system should provide the graph which satisfy the analyst requirement to the analyst. |
| Req-17 | 4 | The system could classify the data by the index(like region, health index) which analyst has input. |
| Req-18 | 3 | The system should output the table that contains the data that analyst need. |

## b. Non-functional Requirements

| | |
|---|---|
| Req-1 | The database should keep all the information rather than user's device. Direct modification from user should be forbidden |
| Req-2 | The system should be simple and easy to use. The webpage should be easy for the user to get the information they want |
| Req-3 | The system should be able to use when related API is updating |
| Req-4 | The response time should be quick |
| Req-5 | The recover time from failure should not be too long |
| Req-6 | The system should enable multiple users to use without overload |
| Req-7 | Minimum maintenance is required, at least once a week |

## c. User Stories

| Identifier | User Stories |
|---|---|
| ST_1 | As a User, I can get a look at the average ratings(0-10 pts) and reviews of the software. |
| ST_2 | As a User, I can get a look at the amount of the registered users. |
| ST_3 | As a User, I can read the detail instruction of the software. |

| | |
|---|---|
| ST_4 | As a Registered user, I can write reviews and rating. |
| ST_5 | As a Registered user, I can contact with the Software maintainer. |
| ST_6 | As a Registered user, I can get a look at the rank among all the Registered users( among Registered users in my State / among my friends), and get my ranking(%). |
| ST_7 | As a Registered user, I can add friends(or follow other users) and press “like” |
| ST_8 | As a Registered user, I can get a look at friends’ exercise data (daily/weekly) and get comparison data. |
| ST_9 | As a Registered user, I can check my healthy condition (healthy or not). |
| ST_10 | As a Registered user, I can set(change) my health goal. |
| ST_11 | As a Registered user, I can get a recommended exercise plan (based on current health condition and goal). |
| ST_12 | As a Registered user, I can get a recommended wearable equipment. |
| ST_13 | As a Registered user, I can check my daily (weekly average) exercise data. |
| ST_14 | As a Registered user, I can get a costumed meal plan (based on current health condition, daily exercise data and goal). |
| ST_15 | AS a Registered user, I can get a look at my daily health data analysis, (get a score based on the time-distribution and duration of exercise) |
| ST_16 | As a Registered user, I can share information of my exercise activities to social networking websites such as Facebook. |
| ST_17 | As a Software maintainer, I can get feedback from users. |
| ST_19 | As a Software maintainer, I can updates database and modify UI. |
| ST_20 | As an Analyst, I can analyze users’ activity data, general information |
| ST_21 | As an Analyst, I can filter users’ activity data, general information |
| ST_22 | As an Analyst, I can plot graph as my demand to analyze if two factors are related |

# 3. Functional Requirements Specification

## a. Stakeholders

Users, analyzers

## b. Actors and Goals

Wearable devices, health apps: initiating the system with data collected from users
system maintainers, frontend and backend system: participating the system to generalize the data and keep it functional

## c. Use cases

Casual description
The summary use cases are as follows:

UC-1 : The user get access to the website either from mobile device or PC , and get the health information provided by the fitbit APF. The information are collected by a wearable device on the users.

UC-3 : After log in, the user could get a random daily recommendation of food or chose to get a customized recommendation of diet through the external API.

UC-6 : Users could also share the health information through a social platform like facebook. The system provide such a platform for users to see other friends health or daily activities information.

UC-10 : Once the recommended diet is provided from the website, the users have the capability to maintain or to modify the diet based on personal preference.

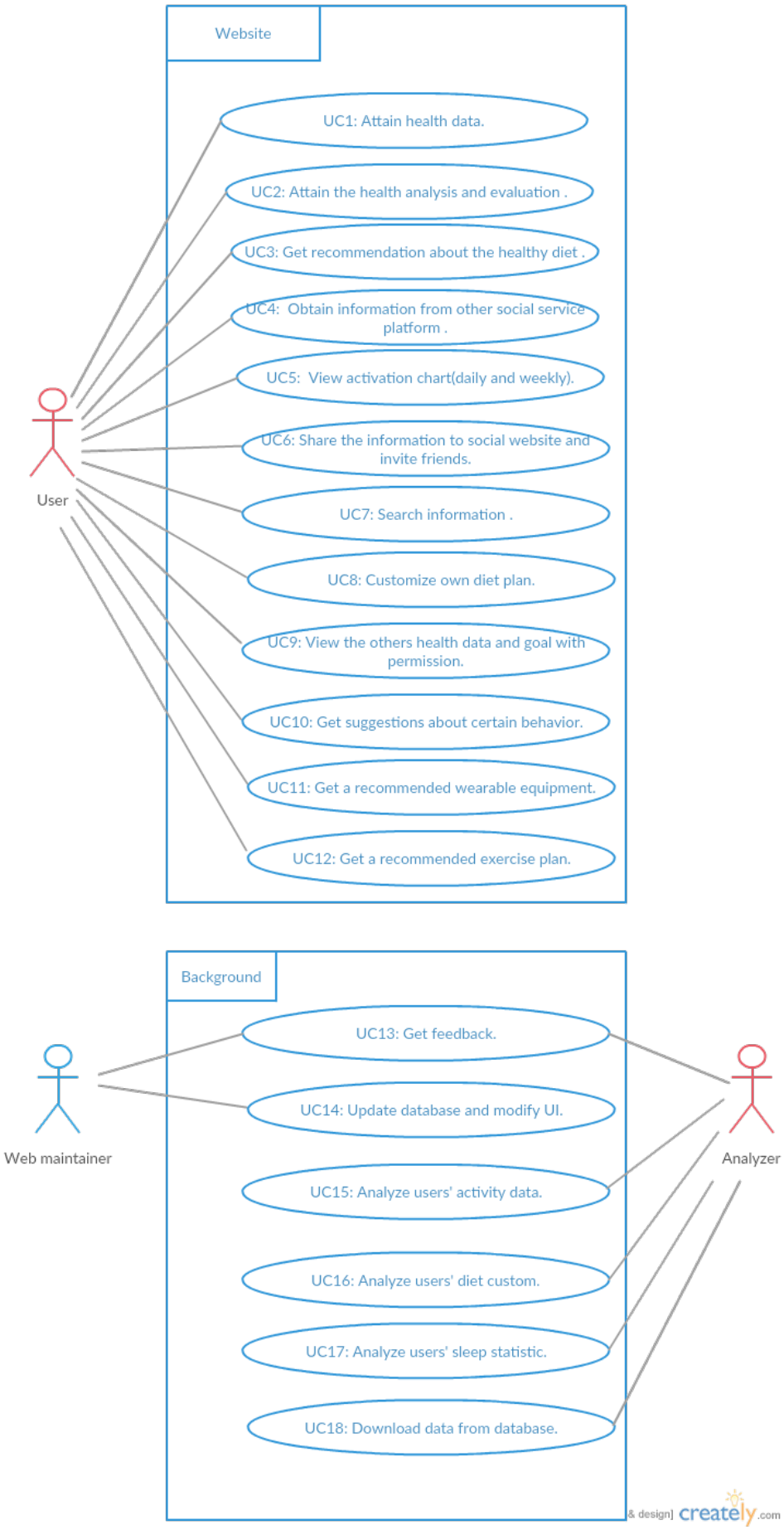
| | ST 1 | ST 2 | ST 3 | ST 4 | ST 5 | ST 6 | ST 7 | ST 8 | ST 9 | ST 10 | ST 11 | ST 12 | ST 13 | ST 14 | ST 15 | ST 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UC 1 | | | | | | | | | X | | | | X | | X | |
| UC 2 | | | | | | | | | X | | X | X | | | | |
| UC | | | | | | | | | | | X | | | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | | | | | | | | | | | | | | | | |
| UC 4 | | | | | | | X | X | | | | | | | | |
| UC 5 | | | | | | X | | X | | | | | | | | |
| UC 6 | | | | | | | | X | | | | | | | | X |
| UC 7 | | | | | | | | | | | | | | X | | |
| UC 8 | | | | | | | | | | X | | | | | | |
| UC 9 | | | | | | X | X | X | | | | | | | | |
| UC 10 | | | | | | | | | | | | | | | X | |
| UC 11 | | | | | | | | | | | | X | | | | |
| UC 12 | | | | | | | | | | | X | | | | | |
| UC 13 | X | X | X | X | X | | | | | | | | | | | |

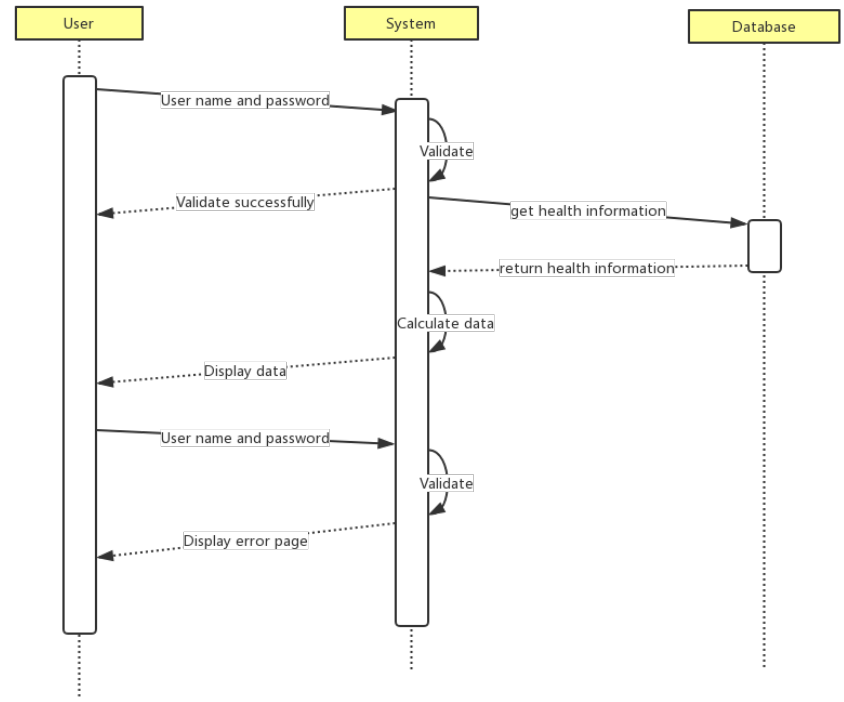
| | |
|---|---|
| Use Case UC-1 | Attain health data |
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br>Postconditions: | ST_9, ST_13, ST_15<br>Users<br>Collecting the user's data, analysis and evaluations<br>Database, health information API<br>user is logged in with proper authorization from the system<br>health information API is updated with user's modification |
| Flow of events for main successful scenario | |
| | 1. user successfully log in to the system<br>2. system requires information from health API<br>3. system display the required information<br>4. user reviews the information and modify the information<br>5. system notify the API and updates the information |
| Use Case UC-3 | Get Recommendation from the healthy diet |

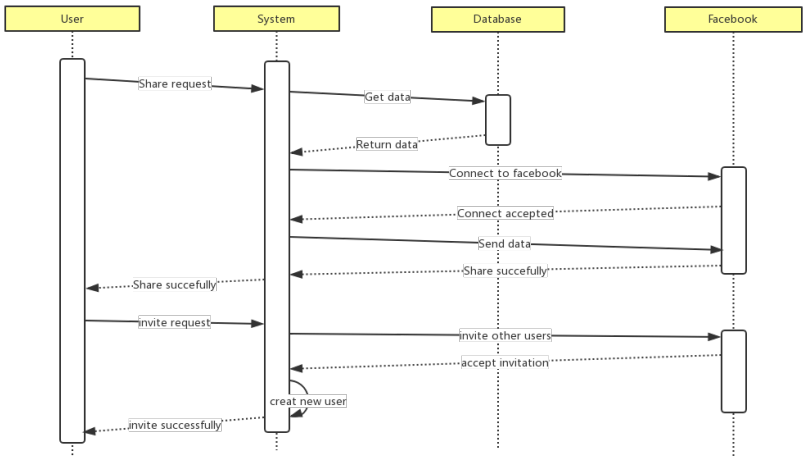
| | |
|---|---|
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br>Postconditions: | ST_11<br>Users<br>To hear suggestions from the system<br>diet recommendation API, Database<br>user is logged in with proper authorization from the system<br>Recommendations are kept in the system |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system<br>2 . user asks for recommendation<br>3. system requires diet API to provide recommendation<br>4. system display the recommendation<br>5. user determine whether to follow the suggestions<br>6 . system collect the reflection from the user<br>7. system determine whether to keep the recommendation based on the user's decision |
| Use Case UC-6 | Share the information to social website and invited friends |
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br>Postconditions: | ST_16<br>Users<br>To share the information to others<br>Sharing API, other platform, Database<br>user is logged in and the internet access is available<br>Information from others is kept in the system |
| Flow of events for main successful scenario | |
|  | 1. user successfully log in to the system<br>2. system requires information from health API<br>3. system display the required information<br>4. user reviews the information and modify the information<br>5. system notify the API and updates the information |
| Use Case UC-8 | Customize own diet plan |
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br>Postconditions: | ST_10<br>Users<br>Modify the recommended diet plan<br>Database, diet recommend API<br>user has at least ask the system to recommend healthy diet<br>diet API is updated with user's modification |
| Flow of events for main successful scenario | |

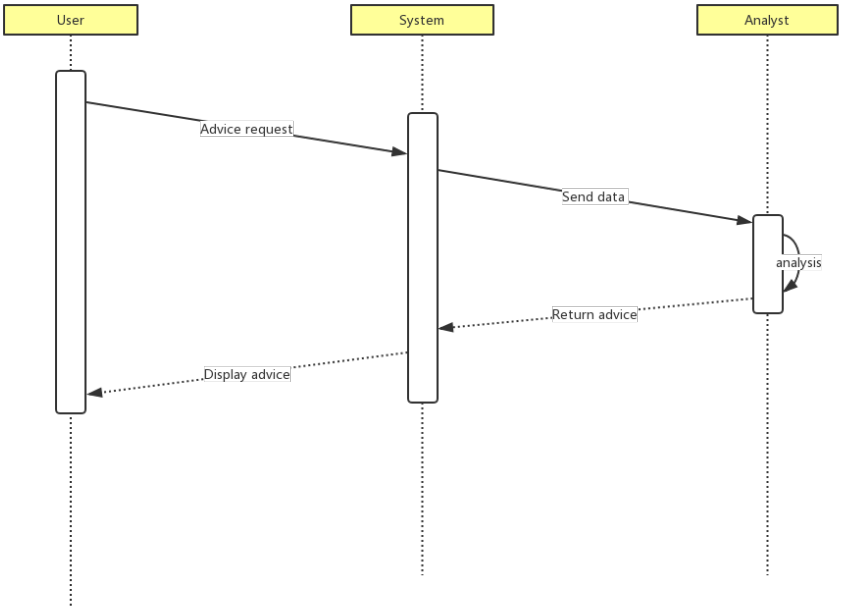
| | |
|---|---|
| | 1. user successfully log in to the system<br>2. user require the recommendation diet from system<br>3. system display the required diet<br>4. user reviews the diet and modify the diet<br>5. system notify the API and updates the diet |
| Use Case UC-9 | View others health data and goal with permission |
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br>Postconditions: | ST_6, ST_7, ST_8<br>Users<br>Compare with friends to share experience<br>Database, social communication API<br>both the user and user's friend are using the system<br>the database would keep user's friend's health information |
| Flow of events for main successful scenario | |
| | 1. user successfully log in to the system<br>2. user send a request to his friend to get health information<br>3. user get permission<br>4. system attain required information through network connection<br>5. system display the attained information<br>6. database keep the health information of the user's friend |
| Use Case UC-11 | Get a recommended wearable equipment |
| Related requirements :<br>Initiating actor:<br>Actor's Goal:<br>Partitioning Actors:<br>Preconditions:<br><br>Postconditions: | ST_12<br>Users<br>Get recommendation of wearable equipment<br>Database, wearable equipment recommend API<br>User logged in and database has health information about the user<br>wearable equipment API is updated, database keep the information |
| | 1. user successfully log in to the system<br>2. user requires recommendation of wearable equipment<br>3. system look into the database<br>4. based on the information stored in the database, the recommendation API make suggestions<br>5. system display the recommendation<br>6. user determine whether to follow the suggestion |

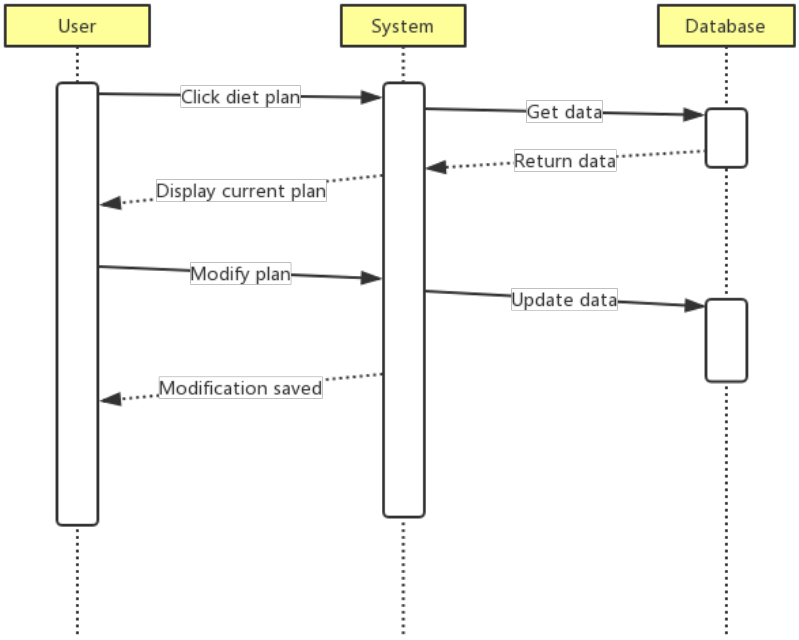

Website
UC1: Attain health data.
UC2: Attain the health analysis and evaluation .
UC3: Get recommendation about the healthy diet .
UC4: Obtain information from other social service platform .
UC5: View activation chart(daily and weekly).
UC6: Share the information to social website and invite friends.
UC7: Search information .
UC8: Customize own diet plan.
UC9: View the others health data and goal with permission.
UC10: Get suggestions about certain behavior.
UC11: Get a recommended wearable equipment.
UC12: Get a recommended exercise plan.
User
Background
UC13: Get feedback.
UC14: Update database and modify UI.
UC15: Analyze users' activity data.
UC16: Analyze users' diet custom.
UC17: Analyze users' sleep statistic.
UC18: Download data from database.
Web maintainer
Analyzer
& design] creately.com

# D. System Sequence Diagram



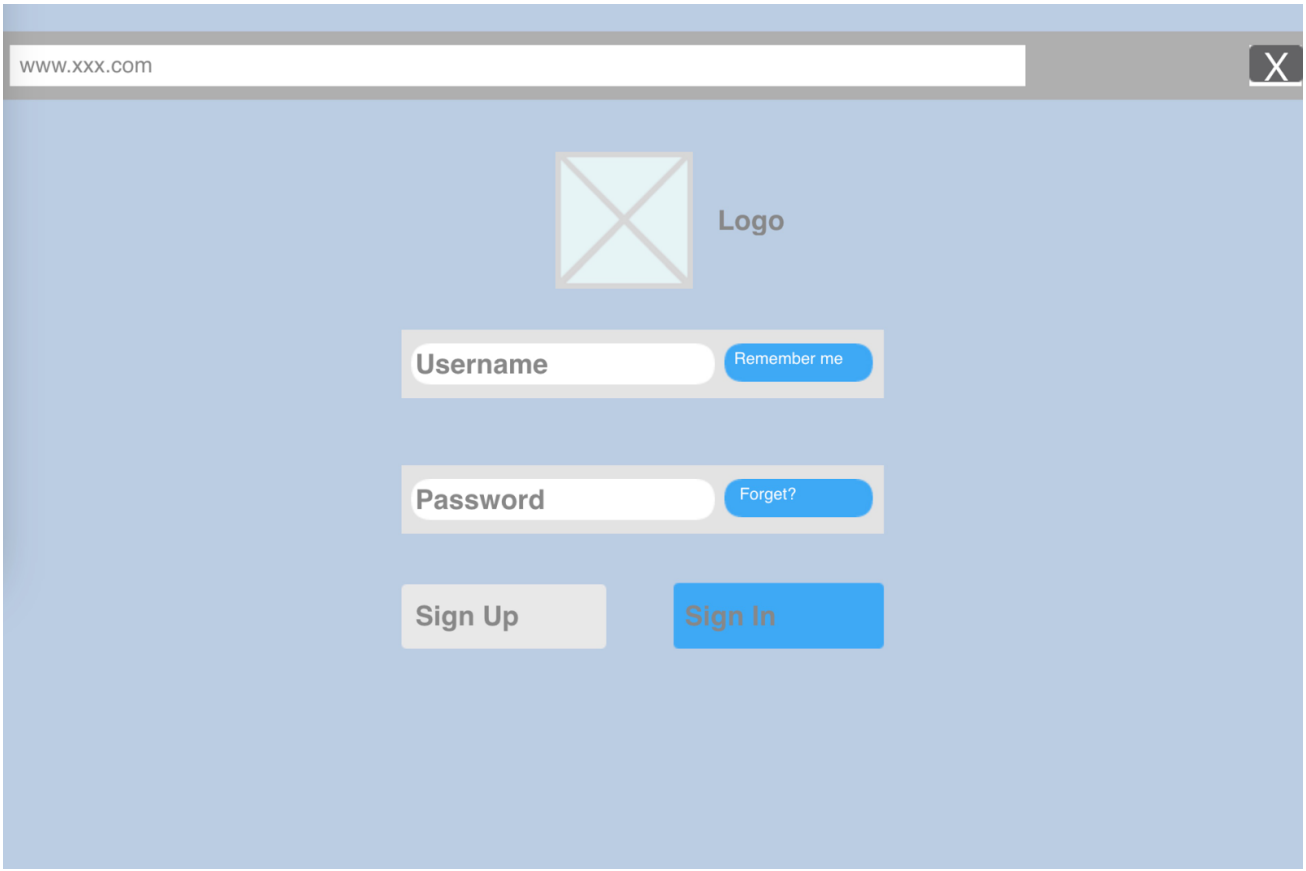


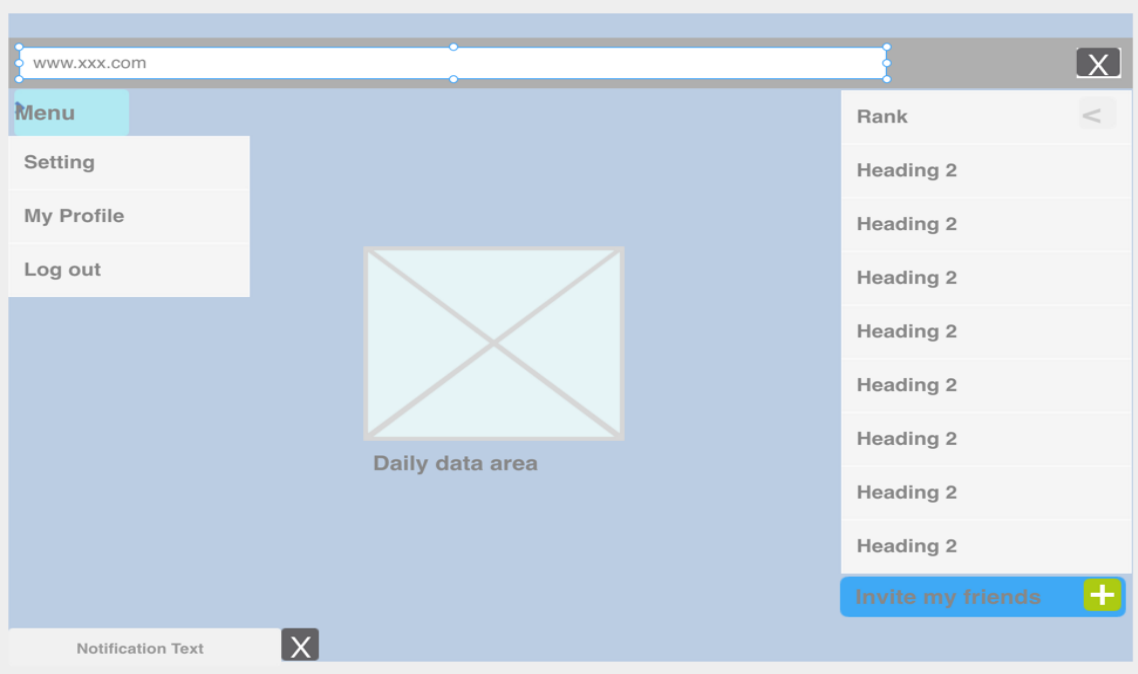

# 4. User Interface Specification

## a. Preliminary Design

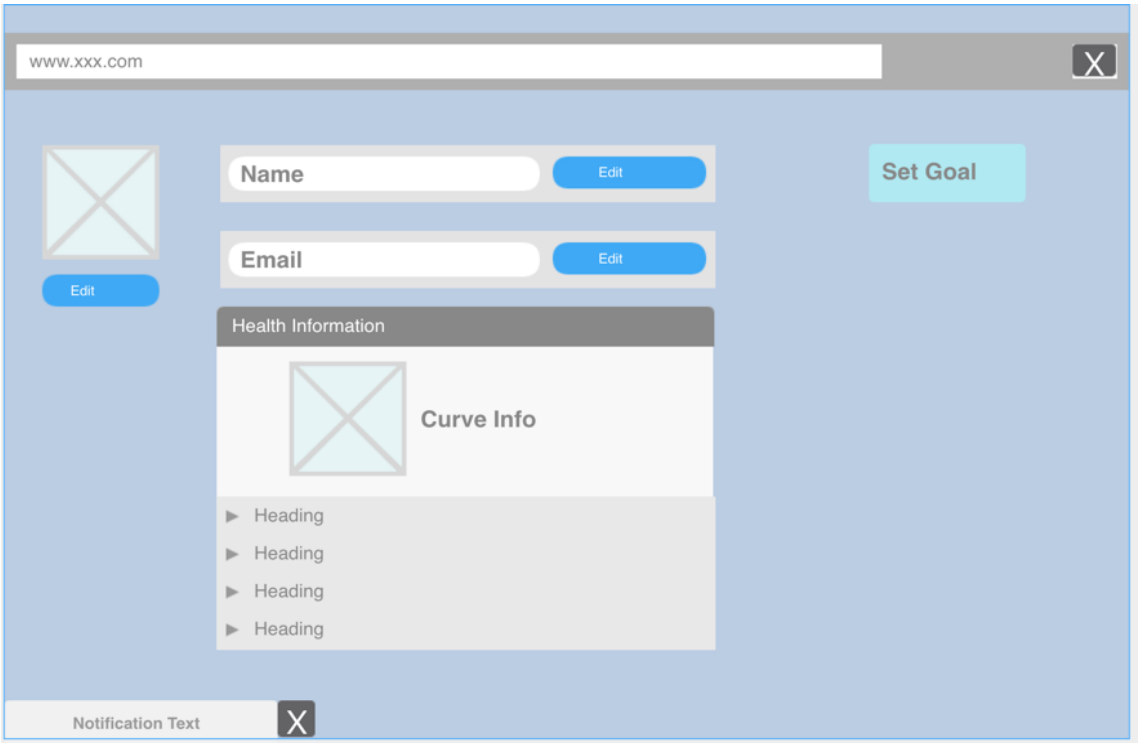
For Users:



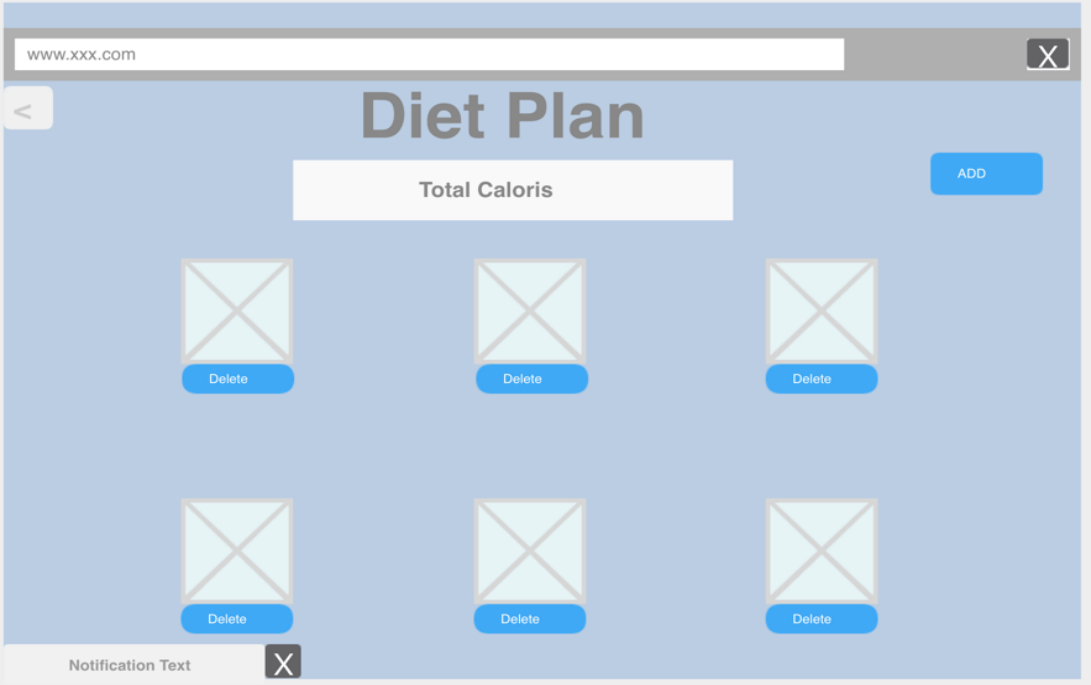

Log in page




Main UI

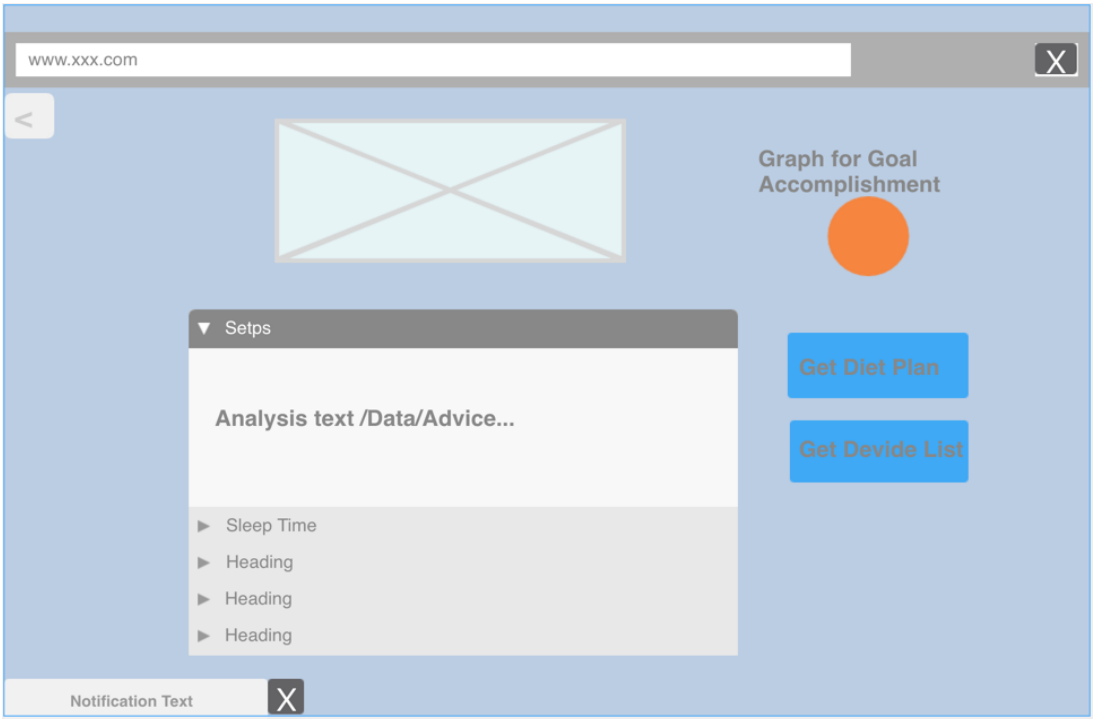



Personal Profiles




Diet Plan

For Data Analysts:




Analysis

## b. User effort estimation

For users:

Log in

1. Click on account settings.
2. Click on sign in button.
3. Input username.
4. Input password.
5. Click sign in button.

Invite friends

1. Click on invite friends button.
2. Input your friends email address.
3. Click on send button.

Set Goals

1. Click on Set Goals button.
2. Select one item in the task list.
3. Set up the target.
4. Click on the Enter button.

Share on Facebook

1. Click on the Share on Facebook button.
2. Input some description.
3. Click on the share button;

Custom diet plan.

1. Click on custom diet plan button.
2. Input the expected calories.
3. Select the food in the list.
4. Click on the save button.

Get suggestion

1. Click on the Get suggestion button.

For the data analyst:

Log in

1. Click on account settings
2. Click on sign in button.

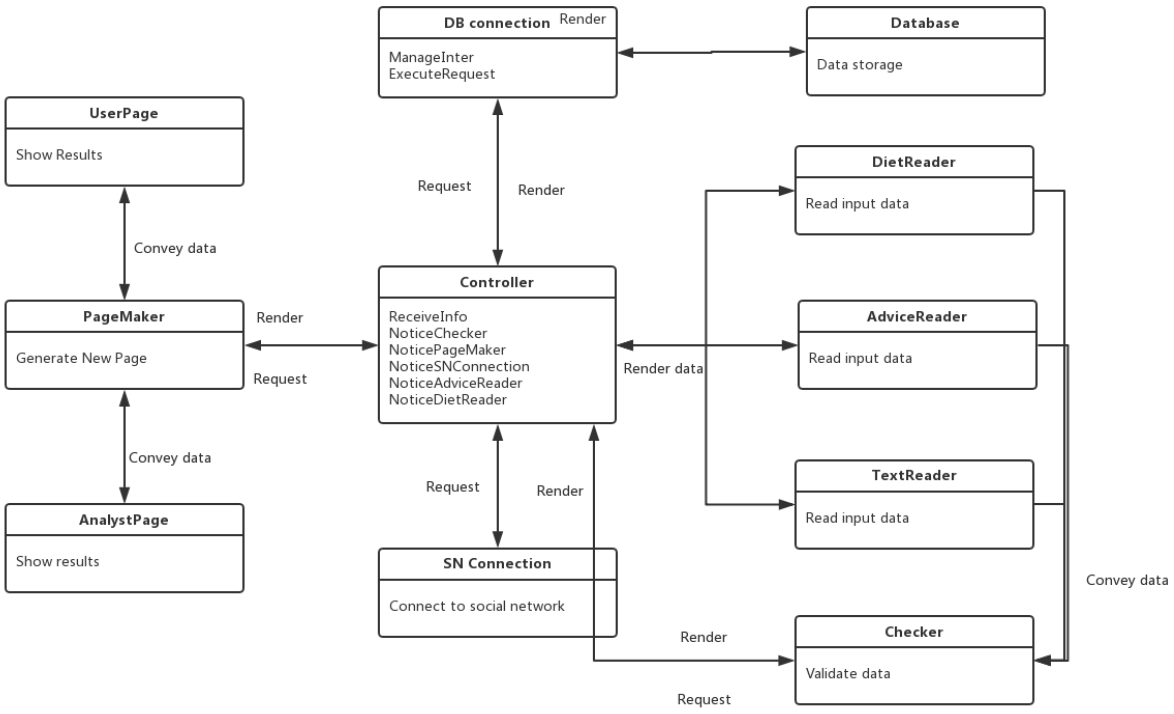
3. Input username.
4. Input password.
5. Click sign in button.

View and get statistic charts

1. Select the item in the categories of data.
2. Input the filter to search.
3. Click on the enter button.
4. Click on the Save data button.

# 5. Domain Analysis

## a. Domain Model




i. Concept Definitions

To analyze the domain model, we first derive the domain model concepts and corresponding responsibilities from the formerly defined system use cases. Table lists all the domain model concepts and corresponding responsibilities

| Responsibility Description | Type | Concept Name |
|---|---|---|

| | | |
|---|---|---|
| R1:Store the data about healthy information | K | Database |
| R2:Read user's custom diet plan | D | DietReader |
| R3:Read data that user types in | D | TextReader |
| R4:Container for user's custom diet plan | K | DietPlan |
| R5:Controller receives the operation "Custom Diet Plan" and controls DietReader to get DietPlan and send to DB Connection to write into the Database. | D | Controller |
| R6: HTML document that shows the user's healthy data | K | UserPage |
| R7:HTML document that shows the user's healthy data such as number of steps , sleeping time and activity hours to analyst. | K | AnalystPage |
| R8: Prepare a database query that best matches the actor's input and retrieve the records from the database | D | DB connection |
| R9:Connect to Social Network and retrieve information | D | SN connection |
| R10: Render the retrieved records into an HTML document for sending to actor's Web browser for display. | D | Page Maker |
| R11: Controller receives the operation "Share on Facebook" and controls SN connection to connect to social network and retrieve information | D | Controller |
| R12: Controller receives the operation "Get advice" and controls DB connection to get Advice from Database | D | Controller |
| R13:Container for the advice that analyst gives | K | Advice |
| R14:Checker receives the username and requests Controller to use DB connection for using datatbase(InfoStorage) to find data in database to compare. | D | Checker |
| R15:Read advice from analysts | K | AdviceReader |
| R16: Controller receives the operation "Input advice" and controls AdviceReader to get Advice and send to DB Connection to write into the Database. | D | Controller |
| R17: Checker gets information from TextReader and check if the information is valid | D | Checker |

ii. Association definitions

Some of the concepts defined above as domain concepts have to work in certain patterns to finish some target requirements. Table below gives the corresponding association definitions based on the defined domain concepts.

| Concept Pair | Association Description | Association Name |
|---|---|---|
| Controller<->Checker | Controller calls checker to check if information is valid and Checker returns results to Controller or checker requests to use database and controller returns results | Generate requests<br>Convey data |
| TextReader<->Controller | TextReader sends signals to Controller or Controller sends signals to TextReader to receive data | Generate Requests |
| Controller<->Page Maker | Controller renders its results and generate requests to Page Maker to display | Generate requests<br>Convey data |
| Checker<->TextReader | Checker checks information from TextReader | Check |
| Checker<->Database | Check checks information from Database | Check |
| DB Connection<->Database | DB Connection gets access into Database and saves data in Database | Save data |
| Controller<->DB Connection | Controller generates requests to use DB connection | Render requests |
| Controller<->Advice Reader | Controller calls Advice Reader to read Advice from analyst | Generate requests<br>Convey data |
| Checker<->DietReader | Checker checks information from DietReader | Check |
| Controller<->SN Connection | Controller generates requests to use SN connection | Render requests<br>Convey data |

iii. Attribute Definition

Among the defined concepts, some concepts share the same attribute, and only different from each other as they have different operands. These concepts are listed in below.

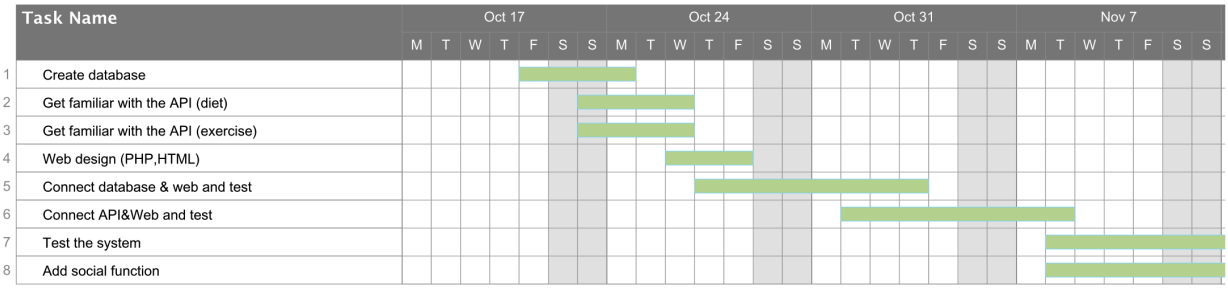
| Concept | Attributes | Attribute Description |
|---|---|---|
| TextReader | Read input data | Allow actor inputs data and put the external data into system |
| DietReader | | |
| AdviceReader | | |
| Database | Data storage | Store all the data that the system needs |
| Controller | ReceiveInfo<br>NoticeChecker | Control the check to validate data |
| | NoticePageMaker | Control the page maker to generate page to display data |
| | NoticeSNConnection | Control the SN connection to connect social network and get information from it |
| | NoticeAdviceReader | Control the advice reader to read advice and store in database |
| | NoticeDietReader | Control the diet plan reader to read diet plan and store in database |
| UserPage | Display Result | Interface for user and analyst to interact with system |
| AnalystPage | | |
| Checker | Validate data | Check whether the data is legal |
| PageMaker | Generate New Page | Generate Userpage or Analystpage to show the result or get input |
| SN Connection | Connect to Social Network | Connect to social network to share information or retrieve information |

iv. Traceability Matrix

| Domain Model | UC-1 | UC-3 | UC-6 | UC-8 | UC-9 | UC-11 |
|---|---|---|---|---|---|---|
| Database | ✓ | ✓ | ✓ | ✓ | ✓ | |

| | | | | | | |
|---|---|---|---|---|---|---|
| DietReader | | | | ✓ | | |
| TextReader | ✓ | | | | | |
| AdviceReader | | ✓ | | | | ✓ |
| Controller | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| UserPage | ✓ | | | | ✓ | ✓ |
| AnalystPage | | ✓ | | | | ✓ |
| PageMaker | ✓ | ✓ | | | ✓ | ✓ |
| Checker | ✓ | ✓ | | ✓ | | |
| DBConnection | ✓ | ✓ | ✓ | ✓ | ✓ | |
| SNConnection | | | ✓ | | ✓ | |

## 6. Plan of Work




## Individual members' contribution

| Contribution | Yuwei Jiang | Xingyu Li | Chenfan Xiao | Jianing Xu | Chengyao Wen |
|---|---|---|---|---|---|
| User Interface | 60% | | | | 40% |
| System Architecture | 60% | | | 40% | |
| Database Architecture | processing | processing | | | |

| | | | | | |
|---|---|---|---|---|---|
| User Management | 60% | | 40% | | |
| Social Network API(construction, test) | | | | processing | processing |
| Fitbit API (construction, test) | | 60% | 40％ | | |
| DietPlan API (construction, test) | | processing | processing | | |
| AWS Construction | 60% | 40% | | | |
| Documentary Work | | | | 60％ | 40% |
| Project Management | | | | 40% | 60% |
| Peer Review | 20% | 20% | 20% | 20% | 20% |
| Subsystem Integration | | | | processing | processing |
| System Test & Debugging | | | | processing | processing |

## 7. References


a. Database(MySQL): http://php.net/manual/en/book.mysql.php
b. Database(MySQL): https://www.tutorialspoint.com/mysql/
c. DietPlan API: http://diet-agents.sourceforge.net/SoftwareDownload.html
d. Food API: http://www.programmableweb.com/category/food/apis?category=20048
e. MySQL Databases Management: https://sourceforge.net/projects/phpmyadmin/
f. MySQL Databases Management: https://www.phpmyadmin.net/
g. MySQL management tool: https://github.com/phpmyadmin/
h. HTML: http://www.w3school.com.cn/html5/
i. API: http://www.webdesignerdepot.com/2011/07/40-useful-apis-for-web-designers-and-developers/
j. API: http://www.gregreda.com/2015/02/15/web-scraping-finding-the-api/
k. XML API: http://itransact.com/support/toolkit/xml-connection/api/