Health Monitoring

Group #1

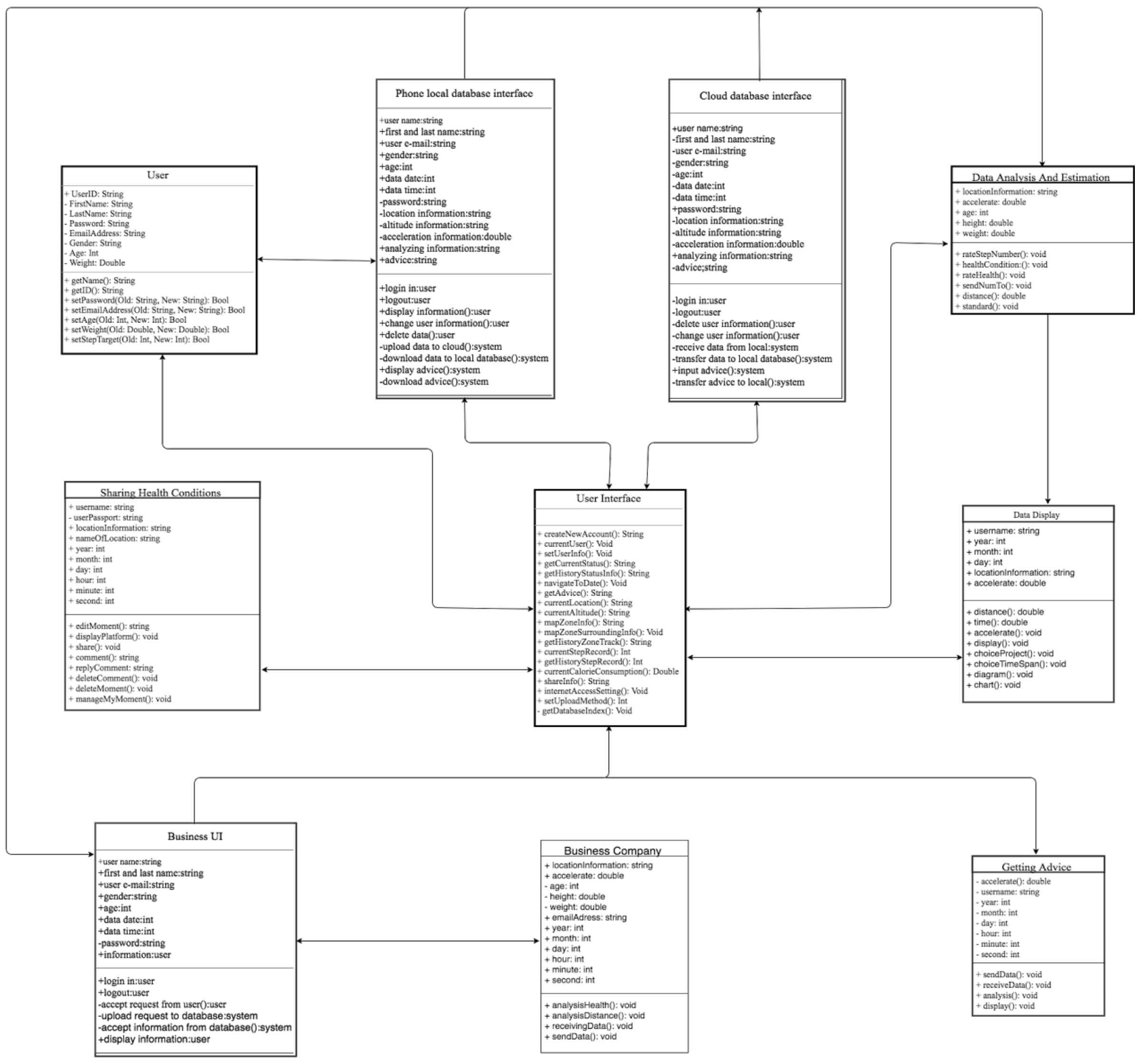
Report 2

Team Members:

- Ai, Ziqi
  - Email: ziqi.ai@rutgers.edu
  - Background: Experienced with C++, Familiar with Python
- Guo, Kai
  - Email: kai.guo@rutgers.edu
  - Background: Experienced with C, Familiar with Java
- Pielacki, Kevin
  - Email: kpielack@scarletmail.rutgers.edu
  - Background: Experienced with Python, C++, and MySQL
- Shen, Hongyuan
  - Email: hongyuan.shen@rutgers.edu
  - Background: Experienced with C, Familiar with Java
- Xing, Xiang
  - Email: xx52@scarletmail.rutgers.edu
  - Background: Experienced with C++ and JDE, Familiar with Java and MySQL
- Yang, Guanjiang
  - Email: gy101@rutgers.edu
  - Background: Experienced with C and Matlab

Class Diagram and Interface Specification

Class Diagram



Text

Data Types and Operation Signatures

Sharing Health Condition

- +editMoment(): string

Compile moments that users want to share with others in the platform. Such as words, images or videos.

- +displayPlatform(): void

A platform where users share moments with each other.

- +share(): void

A function that sending moments to the platform.

- +comment(): string

Every user can freely comment others' moments.

- +replyComment: string

Reply comments of moments.

- +manageMyMoment(): void

A function that can help users manage their own moments, such as reply others' comment, delete comment and delete their own moments.

Data Analysis And Estimation

- +standard(): void

Defining the standard that what health condition is better.

- +rateStepNumber(): void

According to the data of all users' step number, comparing the number of the user and his/her friend. Showing user the result.

- +healthCondition:(): void

According to user's everyday data and health standard defined, showing users their health condition.

- +rateHealth(): void

The app has already know everyone's health condition, then rate it and showing the result to users.

- +sendNumTo(): void

Sending health data to business company. After analyzing, the company will give results back to users.

- +distance(): double

According to location information, distance is easy to figure out.

Data Display

- +display(): void

If user A wants to check his/her health data in a diagram or chart. Just selecting year/month/week then health data and the corresponding chart will display on screen.

- +chooseProject(): void

Choose what form user want, such as chart or diagram.

- +chooseTimeSpan(): void

Choose time span. for example, data of 5 days or a month that user want to know.

- +diagram(): void

Showing diagram.

- +chart(): void

Showing chart.

Getting Advice

- +sendData(): void

Sending data to business company.

- +receiveData(): void

Receiving data from business company.

- +display(): void

Displaying data on the screen.

Business Company

- analysisHealth(): void

According to users’ everyday data, analyzing their health condition

- analysisDistance(): void

According to users’ everyday data, analyzing movement condition.

- receivingData(): void

Receiving data from users.

- sendData(): void

Sending data to users.

Phone local database interface

- -Upload data to cloud(): system

Upload all collection data to the cloud system.

- -Download data to local database(): system

Download all collection data to the local system.

Cloud database interface

- -Transfer data to database(): system

Transfer selection data to the local database.

- -Receive data from local(): user

Accept all data from local database.

User

- + getName(): String

A function that users can get usernames with IDs.

- + getID():

StringA function that users can get IDs with usernames.

- + setPassword(Old: String, New: String): Bool

A function that users can set and change their passwords.

- + setEmailAddress(Old: String, New: String): Bool

A function that users can set and change their Email addresses.

- + setAge(Old: Int, New: Int): Bool

A function that users can set and change their age information.

- + setWeight(Old: Double, New: Double): Bool

A function that users can set and change their weight information.

- + setStepTarget(Old: Int, New: Int): Bool

A function that users can set and change their step record targets (goals).

User Interface

- + createNewAccount(): String

To create a new account for new users.

- + currentUser(): Void

A function that let users to login with their account.

- + getCurrentStatus(): String

Providing the user with his/her current health condition information.
(Excellent/fine/ordinary/weak/dangerous)

- + navigateToDate(): Void

Navigation to a specific date in user's status history list.

- + getAdvice(): String

Providing advice for the user, based on his/her health condition.

- + currentLocation(): String

Showing user's current location.

- + mapZoneInfo(): String

Showing the name of location.

- + mapZoneSurroundingInfo(): Void

Providing the geography information around the current location. (Weather/humidity etc.)

- + getHistoryZoneTrack(): String

Tracking user's location history.

- + currentStepRecord(): Int

Showing current step record of the user.

- + currentCalorieConsumption(): Double

Showing current Calorie Consumption of the user.

- + shareInfo(): String

A function that user can choose to share his/her health condition with friends.

- + setUploadMethod(): Int

Setting whether to upload user info via a data or wifi connection.

Traceability Matrix

User: The actor that operate on the app
User Interface: The space where system interacts with Users
Data Display: Portal view of user information and advice
Data Analysis and Estimation: Studying health condition of users by their health data
Phone local database interface: Runs checks for local data upload in temporary local database
Cloud database interface:The place for Administrators or Users to interact with cloud database
Sharing Health Condition: Allow users to post health condition to social communication software
Getting Advice: Providing the build in Advice tag to User Interface
Business UI: The space with interacts with Business Company
Business Company: A company that give professional advice to users.

Domain concept	User Interface	System	Business Company	Database
User	X			
User Interface	X	X		

Data Display	X	X		X
Data Analysis and Estimation		X		X
Cloud database Interface				X
Phone Local Database Interface				X
Sharing Health Conditions	X	X		
Getting Advice		X	X	
Business UI			X	
Business Company			X	

System Architecture and System Design

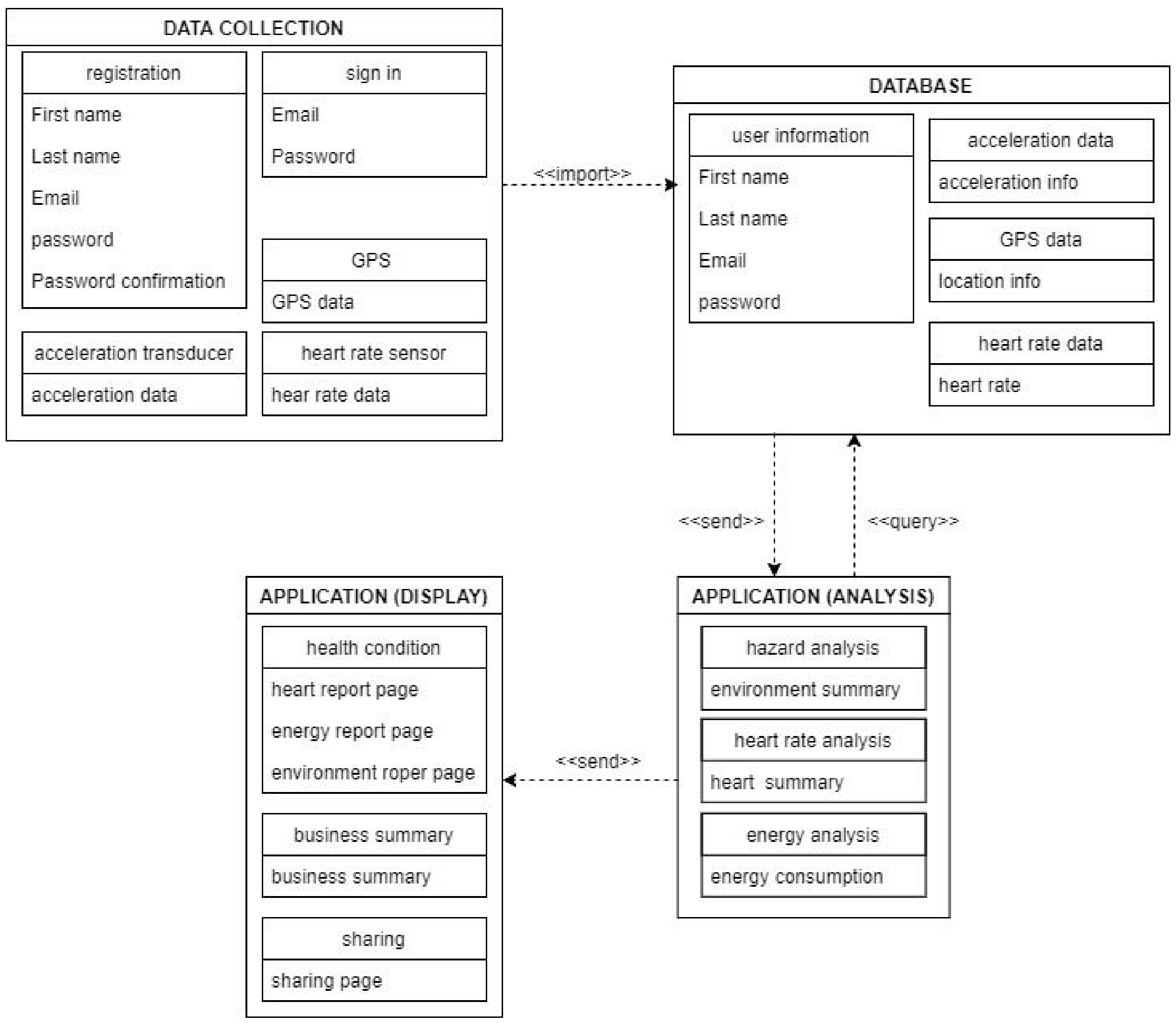
Architectural Styles

The architectural style of this health monitor system include principle from a variety of styles. Our system incorporate the Client-server model in business summary part. Business company can access the data through the internet via a website then dashboard. Service-oriented architecture also plays an important part in our app, as the system provides a number of self-contained functions including registering an account , providing user's health condition and history, providing health rating report to business company. What's more, this app is base on an Database-centric architecture, each sub-system fetch and analysis the data from the database and pass the result back to the database. We may also have a local database that store the contemporary user's info and then update to the cloud database.

Identifying Subsystems

Mapping Subsystems to Hardware

Reason to choose this architecture:



The System Architecture mainly composed of 4 parts.
The first part is Data collection. The subsystem of the part are:
(1) Registration
(2) Sign In
(3) GPS
(4) Acceleration transducer
(5) Heart Rate sensor
Users enter their personal information in Registration to create a new account.
When the app activate the three sensor collects data and send them to database.

The second part is Database
(1) User information
(2) Heart rate data
(3) Acceleration data
(4) GPS data
User information save the data the user enter, and the other three part save the data collected from sensors.
After receiving the requirement from the application, it sends the corresponding data to the next part.

The third part is Application(Analysis)
(1) Hazard analysis
(2) Heart rate analysis
(3) Energy analysis
It receives the data from the database and analyze them through the model.

The fourth part is Application(Display)
(1) Health condition
(2) Business summary
(3) Sharing
This part can display the result of the analysis in several page. The sharing and business summary function is also in this part.

Smartphone(Sensor)
-GPS
-Acceleration Transducer
-Heart rate Sensor

Server(Database)
-User Info
-GPS Data
-Acceleration Data
-Heart rate Data

System(Application)
-Data Analysis

-Health Data Page
-Business Summary
-Sharing Function
Persistent Data Storage

Persistent Data Storage is an important facet of the health monitor program. The main reason is that the data collected from the sensor will be consistently updated to the database to establish a statistic model to analysis user's health condition. The persistent Data will be stored in 2 kind of databases, user's local database and the cloud database. The cloud database will storage all metadata for this app. And the local database is actually a buffer that stores the user's data and prepare for uploading.

The local database format: The database will contain these set of fields.:
User's ID, password(if already exist), email address, accelerate sensor data,
GPS data.

The cloud database format:
User ID, password, email address, accelerate sensor data, GPS info,

Network Protocol

Due to the system uploading data to a remote server the primary network protocol will be HTTP. This protocol provides a few benefits that can simplify the data upload tremendously. The main benefit is the existing website can be used to listen to requests directed to a specific mobile URL to handle the mobile user's requests. For example, if a user want's to login to their mobile app the form would be submitted using a POST request to a mobile URL for handling login credentials. The simplest way to format the POST request would be a JSON text containing the form's ID as a dictionary key and string value assigned to the key as the form's value. Once the user has been signed in they can retrieve the remote data required for each mobile view through an HTTP GET request. The server should provide a content response with JSON as well containing the needed data to render to a mobile view. An example of this would be if a user want's to view their heart rate trend the HTTP response will have a list of values pertaining to the user's logged heart rate. From there the mobile application will create a graph showing the user's heart rate trend.

The other benefit of using HTTP is the fact that it's mainly used over TCP. TCP is a transport layer protocol that provides reliable transfer or data as well as congestion control. Overall that means the data integrity of the content will be handled mostly by the underlying network protocols. This also means should the server be overloaded with a high amount of requests the user's accessing the content will be throttled to better serve their content. Many higher level programming languages will already have an HTTP implementation library the generate proper request headers to achieve the project's desired result. This will ultimately make implementation a lot simpler and reliable. The idea is to utilize a flexible, well tested, and proven dependable implementation on an existing protocol (HTTP) and build our network needs on top of it. This way if the project demand shifts it will not require a large effort to change direction in the future with a customer's demand.

The final benefit of using HTTP is we can easily provide a bit of network security. This would be done through HTTPS which provides application layer security. This is necessary since anyone sniffing the network while the user is running the POST request with their login details mentioned earlier could potentially learn the login credentials of that user. It's important to not solely depend on a single encryption protocol since they can be cracked later in the future exposing all or your user's data. Most mobile devices will access the internet using WiFi to save on mobile data. Recently as of October 2017 it has been shown that the primary WiFi encryption protocol WPA2 has a fundamental security flaw that allows a hacker to easily decrypt WiFi packets. This attack has been shown to be especially dangerous for Android devices. HTTPS solves this by utilizing a key exchange so a potential hacker listening into the key negotiation will not learn the user's login details. It should be a noted to obtain a valid HTTPS certificate we would need a registered domain name for the remote site which can be done upon production deployment.

Global Control Flow

Execution Orderliness:

Our app is mostly procedure-driven and each customer may go through the same steps. Generally, most user's will go through the same registration and login procedure. For instance, once customer has created the account and try to check his health status, he/she have to follow the build-in procedure to get the final result and advice.

Time Dependency:

Our system use several timer to control the data collection procedure and automatic logoff procedure. For example, we collect user's info such as position or energy consumption index once a hour, thus the time will be set to 60 minute and then the corresponding data will be transferred to the local database and then the cloud database.
Furthermore, we may apply a timer for log-off. If the customer do not make any action in several hours, the program will automatically logoff. During logoff time, the user's info will be collected into the local database but user cannot check the history log. This is one way of protecting user's privacy.

Concurrency:
Our system may use multiple threads. There are some sensors, which requires multiple thread in order for them to work all at the same time. For the database part, we are applying http and there may not me multiple thread method to be applied.

Hardware Requirements

- User Requirements
    - Android Mobile Device
        - Version 4 or Higher
        - 200 MHz Processor
        - 2 GB of RAM
        - 1080x1920 Resolution
        - 50 MB Available Storage
        - Mobile Sensors (Optional per Feature)
            - Heart Rate Sensor
                - Heart Rate Predictor Feature
            - GPS Sensor
                - Hazard Area Notification
        - Mobile Network or WiFi Access
- Service System Requirements (Demo Purpose Only)
    - Remote Application Server
        - Ubuntu 16.0.4.1 x64
        - 2.40 GHz Intel CPU
        - 1 GB RAM
        - 30 GB Storage Space
        - MySQL 5.7
- Service System Requirements (Deployment)
    - Remote Application Server
        - Ubuntu 16.0.4.1 x64
        - 2.40 GHz Intel CPU
        - 128 GB RAM
        - 1 TB Storage Space
        - MySQL 5.7

Algorithms and Data Structures

GPS

Based on the observation of the distance (or distance difference) between GPS satellites and the antenna of the user receiver, the position corresponding to the receiver antenna, is the position of the observing station, is determined on the basis of known instantaneous satellite coordinates. The essence of GPS absolute positioning method is the space intersection in surveying. In principle, the observing station is located at the intersection of the sphere with the three satellites as the center of the sphere and the radius corresponding to the intersection with the plane where the observing station is located. Because GPS uses the principle of one-way ranging, the actual observed distance from the station to the satellite contains the influence of the satellite clock and the receiver's clock synchronization difference. The satellite clock error can be corrected according to the clock error parameter given in the navigation message The clock error of the receiver is generally unpredictable. It is usually treated as an unknown parameter in the data processing and coordinates with the observatory.A station solves four unknowns in real time and requires at least four simultaneous pseudo-range observations. Absolute positioning can be based on the state of the antenna is divided into dynamic absolute positioning and static absolute positioning. Whether dynamic or static, the observations are based on the measured satellite pseudoranges.

Rating

In rating part, since the Machine Learning algorithm was not used at this time, there are mainly two kinds of approximate mathematical algorithms used so far. One is the algorithms of Counting Steps and the other is the algorithms of Calculating the Calorie Consumption.

Step counting part:
For the step counting part, the x, y, z value extracted from the accelerator meter will appear as regularly vibrating graphs while the user is walking--no matter what kind of position the smartphone is, which could be taken into considering via different threshold values for one count. For example, the previous four counts beyond the threshold could be taken as the start signal, and the continue six counts within the threshold could be taken as the ending of counting. Then the rest is relatively easier part to count the coming steps.

Calorie Consumption calculating part:
For the calculating of the Calorie Consumption part, because of cannot use wearable devices other than mobile phones, the data obtained in this project are mainly calculated for showing calorie consumption while walking or running. The consumption rates of the different status of

the user are not the same. The system requires users to input their weight, height, gender, and age for further calculating. For example, the Calorie consumption of running can be put as a factor multiplies the weight and the distance of running; the consumption could also be calculated as two factors multiplies the weight and the time spent on running/walking (the factors can be calculated via the speed during this time from map track information). Additionally, the factors are also influenced by age and gender of one person.

One of the more standard and most accurate ways to calculate the equation is to use the calorie expenditure formula below. It comes from the Journal of Sports Sciences and provides a formula for each gender.
Men use the following formula:
Calories Burned = [(Age x 0.2017) — (Weight x 0.09036) + (Heart Rate x 0.6309) — 55.0969] x Time / 4.184.
Women use the following formula:
Calories Burned = [(Age x 0.074) — (Weight x 0.05741) + (Heart Rate x 0.4472) — 20.4022] x Time / 4.184.

Health Advice

Getting advice is a part to remind users about their bad behavior, helping them notice the problem and then doing better.

The US Centers for Disease Control says "300,000 deaths each year in the U.S. likely are the results of physical inactivity and poor eating habits." Actually, physical inactivity raises the risk of death from heart disease, stroke, colon cancer and diabetes. In order not to be inactive and to reduce health risks, health authorities such as the American Heart Association recommends moderate intensity exercise for either 30 minutes a day for 5 days a week or a total of 2 hours and 30 minutes per week.

In part Energy Consumption and Rating, the system has already analysis motion data of users and got conclusions such as step number and calories burned. Our application will alerts and remind users to get up and move when they have been inactive for a period of time. When users click the icon of Professional Advice, they will get a precise advice that can help them being health.

For users who exercise acute than it should be, our application will also reminds users risks. It is useful for people who spend long hours at sedentary jobs or sedentary recreational activities and people who can not endure the exercise intensity. Helping them moving health may help reduce the health risks.

Hazard Detection

Algorithm

In the hazard detection part of our health monitor app, the system will apply two linear process:

1. Fetch the hazard that is close to the user's living area from the database. The system will require nearby hazard location and the hazard level. Base on the hazard location and level, the corresponding " Health threaten zone" will be give out and stored in the database
2. Find out if user's location is close to the hazard location and give out alert. The system will compare user's location with the hazard zone in the database.

The alert will be give if catch the within bound data.

Data Structures

The hash table will be applied to link each user's data to the corresponding hazard zone.
 This will be a great way of facilitating our speed of searching process. One the other hand, this will also save a lot of space for data storage and leave more space for the local database for analysing purpose.

User Interface Implementation

<u>My Profile</u>
The My Profile part will come up and show user's personal information. When user wants to see his personal information, he needs to click the "My Profile" button, and the App will download user's information data from database and display with some instruction in the screen.

<u>Hazard Detection:</u>
The hazard detection page is designed for already login user. Within 1 click, a user is able to jump to this page and check nearby hazards in a map view. The corresponding distanced toward the hazard will be display and marked. Furthermore, there will be a text alert displayed for detailed explanation for the hazard type and thread level.

<u>Health Rating:</u>
The page displays the user's rating score with a range of 0 to 10 and will add the ranking among user's friends who use the same application in the future. The variable result will just display in the center of the page. What is more, on this page user can using the bottom button to jump to another page--Get Advice. This Rating page is reached by using main page's navigation and so is the Get Advice page.

Energy Consumption
This screen is a data display of the data about the user's energy consumption, and it is designed to show two types of data: recent data and long term statistics. The recent data includes "Today's Steps" and "Energy Consumption", and separately shows the row step data collected by acceleration sensor and energy consumption calculated by step data. Then, the long term statistics including "Total consumption this week" and "Daily average consumption this month" are the direct feedback on the user's exercise saturation.

Advice:
When the icon of Professional Advice clicked, screen will jump to another interface and display the content of advice which based on their motion data.
When the icon of Diagnose clicked, screen should be able to jump to an interface and display many kinds of specific disease. Users should be able to select the disease that suit for their symptom and know these diseases better and getting advice simultaneously.

Design of Tests

My Profile
User has already existed in the system.
User enter his personal information into the system and these information has already been stored in the database.
Testing, after login, click the "My Profile", then the App will turn out and display this user's information in the screen. And for this step, compare the screen's display data with database and make sure that there is no malformed format.

Health Rating:
The analysis process will be arranged in the system using Machine Learning algorithm, so the Tests will mainly focus on the result of this process via comparing with some already known results, which are generated by approximate mathematical methods.
Firstly, for the unit test part, the test result will show whether the score can represent the user's health condition.
Secondly, the approximate mathematical methods to rating scores will be tested by using some known status and corresponding results (probably from authoritative articles or studies).
Then the ML algorithm will be tested to see whether it could get the right approximate score of a user's health rating.
Additionally, the generated ranking scores will be tested to judge whether they work right.
Moreover, the weight of hazard site's influence in the rating process would be paid more attention to.

Hazard Detection:

Test cases:

1. Check if the user can be redirected to the hazard page upon click the button
2. Check if the correct user's info will be displayed.
3. Check if the nearby hazard is properly displayed with introduction
4. Check if the alert will be given if the user has stepped around the hazard
5. Check if the parameter for hazard level is properly passed to the rating section

Test coverage:

The test cases will cover all the code within this section of health monitor and involve the initialization of the health rating part.

Integration test :

The hazard test will also test the interface between database and health rating system. For the database part, test if the corresponding data is passed successfully and properly, and for the health rating part, check if the consequences and parameter derived from the hazard detection is properly passed.

Energy Consumption

Test cases:

- When user touch the "Energy Consumption" icon on the main menu, it can correctly jump to this page.
-It can display the correct data on the top of the page.
-The title of every text views are shown in the designed location of the page.
-Data can display following the corresponding title.
-Data is correctly sent from database to this page.

Test coverage

-the "Energy consumption" user interface page.
-the link between main menu and this page
-the data transmission with database

Advice:

Test cases

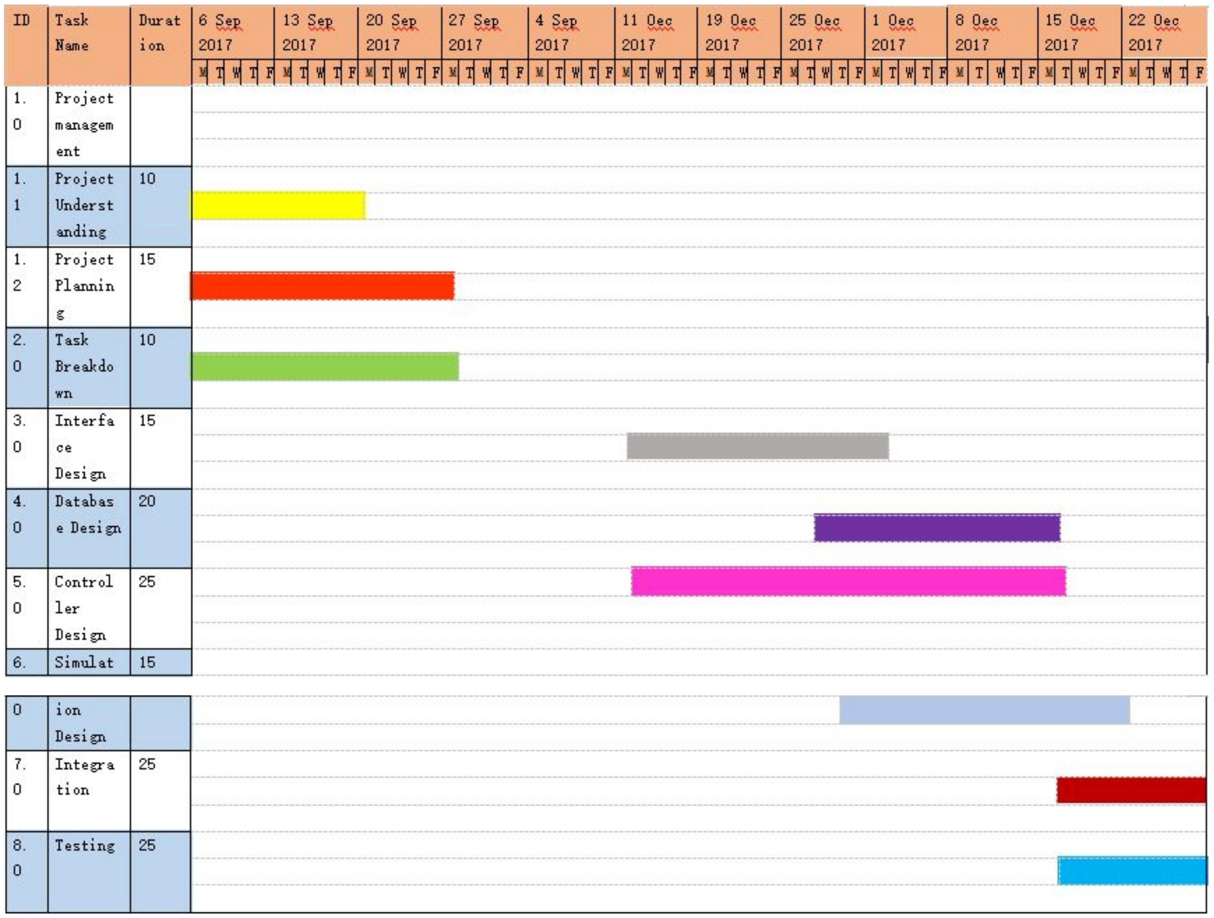
-This part can access database successfully
-It should getting the result of health rating, step counting and calories burned.
-When the icon of Professional Advice clicked, screen will jump to another interface and display the content of advice
-When the icon of Diagnose clicked, screen will display many kinds of specific disease
-Users can find almost all normal diseases that suit for their symptom

-Whichever icon of disease clicked, the screen will display specific information
-Advice will be displayed simultaneously.

Project Management and Plan of Work

Everyone contributes equally in the report content.

Plan of Work



ID	Task Name	Duration
1.0	Project management	
1.1	Project Understanding	10
1.2	Project Planning	15
2.0	Task Breakdown	10
3.0	Interface Design	15
4.0	Database Design	20
5.0	Controller Design	25
6.0	Simulation Design	15
7.0	Integration	25
8.0	Testing	25

Breakdown of Responsibilities

Group Member Name	Xing Xiang
Work Done So Far	Responsible for team organization and communication. I have been constructing the structure and breakdown of each report and submitting after organization.

	Design the main menu for our android UI and the homepage of our website. Integrated all the android UI page.
Future Work	I will continue to facilitate the division of work to ensure that everyone contributes evenly and meets deadlines. Code wisely, I will develope the data analyzing part and develop the machine learning algorithm. And I will keep doing the coding integration in android design.

Group Member Name	Guanjiang Yang
Work Done So Far	Design the “Energy consumption”page of UI. participate in database format design design the layout of homepage develop the gps data collection with Hongyuan
Future Work	develop the step and heart rate data collection with Hongyuan

Group Member Name	Hongyuan Shen
Work Done So Far	Design and finish the “My Profile” page of UI. Participate in database format design. Develop the GPS data collection with Guanjiang.
Future Work	Develop the step, heart rate and temperature data collection with Guanjiang.

Group Member Name	Ziqi Ai
Work Done So Far	Designing and coding implementation of “Health Rating” page of Android UI. Participating in database format design. Developing and tentatively testing the Step Counter function with Kai Guo.

	Working with other team members on the documents. Finding some solutions for algorithms used in the Pedometer function and Rating function.
Future Work	Developing both the analyzing and the first-level processing part for the data collected. And I will applying the solutions for testing and checking the algorithms that will be used in the system with Kai Guo.

Group Member Name	Kai GUO
Work Done So Far	Accomplish “Advice” part of Android UI. Participating in database format design. Developing and tentatively testing the Step Counter function with Ziqi Ai. Working some parts of documents. Making advertisements.
Future Work	Improving design of my part of UI. Integrated step rating program with our application with Ziqi AI. Design an appropriate algorithm to manipulate data from database.

Group Member Name	Kevin Pielacki
Work Done So Far	Web server framework. Web server user portal. SQL table design. HTTP request remote server processing for login and registration.
Future Work	Data upload insertion. Post data analysis. Feedback notification generation to mobile devices.