Software Engineering I Group II

Health Monitoring Analytics 2014

D. Yao, W. Fang, W. Zhang, Y. Sun, Y. Wu, Z. Zheng

https://sites.google.com/site/healthmonitoringanalytics2014 Dec 13th, 2014

Individual Contributions Breakdown

“All team members contributed equally, and should be scored equally.”

Assignments	D. Yao	Z. Zheng	W. Zhang	Y. Wu	W. Fang	Y. Sun
Weighted calorie	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Part of speech	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Word frequency	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Tweet sentiment	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Demography	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Geography	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Time span	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Exercise classification	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Exercise frequency	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Personal diagnosis	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Topic correlation	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Healthy food	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
User ranking	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Report writing	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Code debug	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Integration	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%

Table of Contents

Summary of Changes

We keep changing our features and goals since the first time we begin doing this project. Below is the summary of change we made. This change is the comparison between proposal and the third report. And the change covers project objectives, use case descriptions, and system design.

Project objectives:

1. In previously version, we use single hashtags and some simple keywords to collect user information and Tweets text, but this is low accuracy. In the new version, we improve the methods to collect data and obtain better result.

2. We no longer use VSM to analyze Tweet text, instead, we use if-idf methods to get result. The reason is we can hardly get word from GATE. However, we can easily obtain such word frequency by using the R language.

3. For sentimental analysis, in the previously version, we use a common mood table to analysis user mood. However, this is not very accurate. Now, we use a improve methods. First, we guess the user's gender based on their tweet text, next we have two table that corresponding to man and woman. Based on our observation, this alternative methods can acquire better result.

Use Case Descriptions

1. In the previous report, we has the log in use case that user can register to our website. However after deliberate consideration, we decided to delete this function. The reason is that we don't want to waste the user too much time on filling their personal and private information. Besides, we find we can easily to get their personal information by using the Twitter API.

2. In the previous report, we decided to add advertisement in the main website page. However, we no longer consider doing that. The reason is that adding advertisement is an effort which takes a lot of time and is not meaningful. Also, we can't make the website profitable, so we decide to cut off this function.

3. We no longer doing the facility search function, the reason is that we can't get enough information of the local gyms and some other facility, also, we think this function overlap with the google map display function. So we cut it off.

4. We no longer doing the community forum function, because it also overlap with the google map display function.

5. We cut off the score systems. One of the most important reason is the professor thought this feature can't reflect data mining algorithm. And that is the essence of the project.

6. We increase the sentimental analysis feature. This feature is important because this could help people know which exercise can bring them happier mood.

7. We add calories calculation feature. This feature is useful because we can help user know the average calories burn for each sports. So they can make optimize choose for exercise.

System design

The main change in system design is that we decided to make the Android client. The reason is that based on our analysis, we find people are using Android to send tweet texts that related to the health topic frequently (as shown in figure 0).

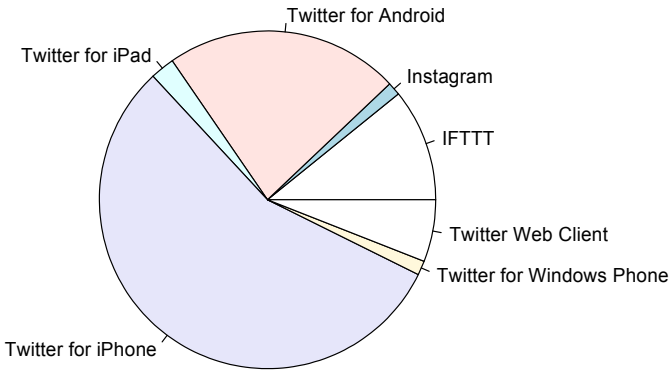


Figure 0. The number of Twitter users in different platforms

1 Customer Statement of Requirement

1.1 Problem Statement

1.1.1 Health monitoring analytics with Twitter

With the development of Social Networking Service (SNS), Health Care Analytics [1] (HCA) discovers a new world. Traditionally, HCA is based on Personal Activity Monitoring (PAM) which collects individual's activity data with sophisticated carry-on devices, e.g., Google Shoe, Nike Fuel, Zeo Sleep, Jawbone's Up, etc. In PAM, customers could adjust their excising intensity depending on the analyzing results by comparing their activity data with the standard health statistics that come from relevant health researches. Although some PAM devices with network connection could help customers share their activity data and analyzing results with others, there is still a limitation of user population. In another word, "none of these products is able to provide analytics for tracking population activities related to healthy lifestyle and gaining insight into lifestyle trends" (see ref. [2]). However, this is what the customers need – the customers are not satisfied with merely looking at their own activity data and analyzing results, and they are also eager to see that of others from all over the world. This is proved by more than one in three American adults who go online to figure out a medical condition surveyed by the Pew Research Center's Internet & American Life Project in 2013 [3].

Due to the emergence of the powerful SNSs with large amount of users and their information, for example, Facebook, Google+, Twitter, etc., it is possible for us to gain public data relating to health issues in order to meet the customers' requirement. Figure 1-1 shows the number of monthly active users in different SNSs in 2014 [4]. Although Facebook and Google+ seem to have more monthly active users than Twitter, each user's information cannot be accessed without his or her authentication because of the privacy, which means we are unable to crawl the private information of the public from Facebook and Google+ database. On the other hand, Twitter is much more friend. We could crawl the past and real-time Twitter data stream provided that we register a Twitter developer account. Thus, we chose Twitter database as our data source. The goal of our team is to mine and analyze the online information about people's exercising activity and other health related issues, to show useful public analyzing results and also customer-oriented health suggestions [2] by using Twitter Application Programming Interface (API).

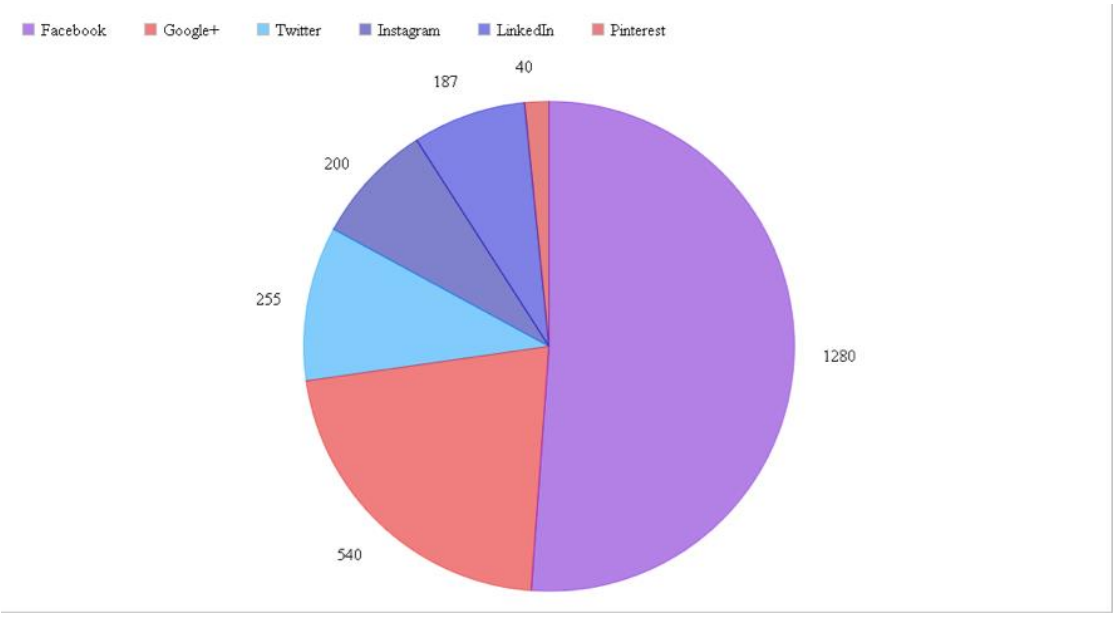


Figure 1-1. Monthly active users (million)

1.1.2 Existing works

Based on the Twitter API, three former groups have developed their own Health Care System (HCS). Although they use different third party APIs for Twitter development - group 1 takes Phirehouse [5] while group 2 and group 3 take Tweet Tracker [6], the scope of the data (always in a JSON format [7]) which they could scrawl from Twitter database is the same since the same Twitter access rules - Rest API and Streaming API [8]. The significant difference among these groups is how they use these data, i.e., what are their features. The comparison of features among the three groups is shown in the Table 1-1.

Features	Group 1	Group 2	Group 3
Tweet heat in geographical distribution (heat map)	√	√	√
Tweet heat in geographical distribution (marker map)	√		√
User ranking		√	√
User Profile			√
Find Partner		√	

Table 1-1. Comparison of features among the former groups

Tweet heat is calculated by counting the number of tweets. To improve the accuracy, group 2 employs two methods: combination keywords and keywords weight. Although they show the distribution of tweet heat in heat map and marker map, the display cannot be varied by time or exercising type. Besides, the heat map is crude, for example, in the group 3's android application, the heat map is simply displayed by implementing the circle overlays on the Google Map – no gradient. The user ranking is also based on tweets counting and the display cannot be varied by the state, time or exercising type. The user profile in group 3's android application shows the user's tweet history, however, there is no combination between the user's data and the public data. As for the finding partner, it is a special feature in group 2, but the finding operation should be in a much more intelligent way. In conclusion, all three groups obviously lack sufficient analysis from Twitter data, probably because they spent too much time on establishing the system and cannot figure out some technique problems. Thus, our team is going to pay more effort on analysis. The Table 1-2 shows the display platform in three groups.

Platform	Group 1	Group 2	Group 3
Website	√	√	√
Android		√	√

Table 1-2. Platforms of the former groups

1.1.3 Improvement and extension

Our team takes the advantage of the existing data collection infrastructure from group 1 because we prefer to use the Phirehouse. Phirehouse is the Twitter streaming API which can download the real-time data from twitter database. However, we also need past tweets which are not used by group 1, so we ourselves find a good Rest API on GitHub [9]. Our team also takes the advantage of the combination keywords and keywords weight method from group 2. But we have refined the keywords settings, for example, we make it equal among exercise type keywords, so it is more reasonable when comparing the exercise type. Besides, we add user demography information in the user table such as age, gender and type (people or company) by text analyzing [10]. Unfortunately, the accuracy of this demographic speculation is not shown in their website.

Features	Description

Tweet heat in geographical distribution	Counting the tweet number in different states shown as a state map which can be varied by time, exercising type and demography. Shown as bar charts. Shown as a heat map with gradient. Shown as a marker map with screen name and tweet text.
Tweet heat in variation tendency	Counting the tweet number in different day in a week, and different time in a day which can be varied by state, exercising type and demography. Shown as line charts.
Tweet heat in exercising classification	Counting the tweet number in different exercising type which can be varied by state, time and demography. Shown as pie charts.
Tweet heat in demography	Counting the tweet number in different age and gender which can be varied by state, time and exercising type. Shown as pie charts.
User ranking	Ranking the user by number of tweets which can be varied by state and exercising type. Shown as tables.
Weighted average calorie	Extract the exercising time in the tweet text, then translate the time into calorie by calorie burned rate in different exercises and weights which determined by the faction of different exercises.
Exercising frequency	Figure out the mean of how many days in a week that different users exercise in different state and exercising type and demography shown as bar charts, and as a state map.
Word frequency	Split the words in tweets and count to find out related topics for correlation topics analytics. Shown as a table.
Correlation topics	Draw the linear regression among different health topics in line charts to find out how close the relationship is.
Tweet sentiment	Match the tweets with different mood and count the number in different state, type, time and demography.
Personal diagnosis	Associate the registered users'

	information with the public information to estimate the health situation and make suggestions.

Table 1-3. Feature description

Besides expanding the user's information, we are also trying hardly to dig out more creative features. Some of them are from the project description in the reference 2, and the others are thought out by ourselves. The Table 1-3 shows the features that are scheduled with fully-description. The rate of progress is shown in the plan of work section. As for the display part of our project, we choose IOS and website as our platform. The drawing APIs are PNchart[11] and Google API[12]. More details about the display part can be found in the user interface (UI) design sections.

1.2 Glossary of Terms

- Social Networking Service (SNS)

  A social networking service is a platform to build social networks or social relations among people who share interests, activities, backgrounds or real-life connections (From Wikipedia). See the Figure 1-2.

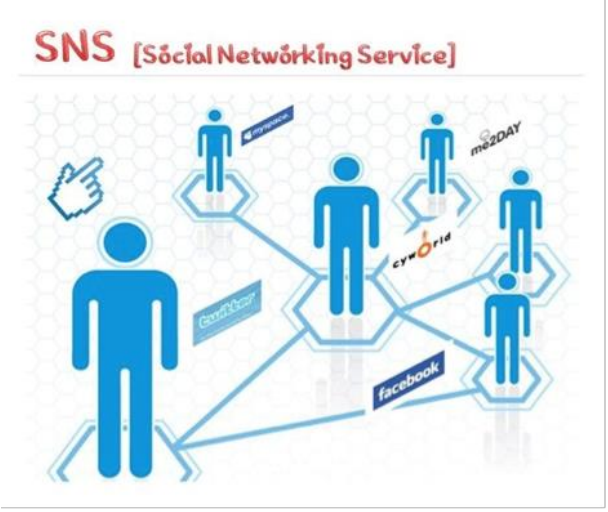


Figure 1-2. SNS

- Health Care Analytics (HCA)

Health care analytics is a product category used in the marketing of business software and consulting services. It makes extensive use of data, statistical and qualitative analysis, explanatory and predictive modeling (From Wikipedia). See the Figure 1-3.

Figure 1-3. HCA

- Server and database

A server is a running instance of an application (Software) capable of accepting requests from the client and giving responses accordingly. A database is an organized collection of data. The data are typically organized to model aspects of reality in a way that supports processes requiring information. For example, modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies (From Wikipedia). See the Figure 1-4.

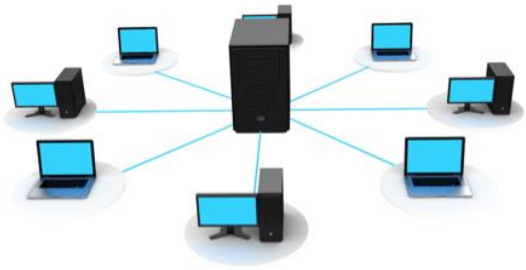


Figure 1-4. Server and database

- Application Programming Interface (API)

  In computer programming, an application programming interface (API) specifies a software component in terms of its operations, their inputs and outputs and underlying types. Its main purpose is to define a set of functionalities that are independent of their respective implementation, allowing both definition and implementation to vary without compromising each other (From Wikipedia).

- Rest API and Streaming API

  Rest API and Streaming API are both Twitter API for accessing Twitter users' data in the Twitter database. Rest API acquires historical data while Streaming API acquires real-time data. Besides, Rest API has connection limitations.

- Heat Map

  A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors (From Wikipedia). See the Figure 1-5.

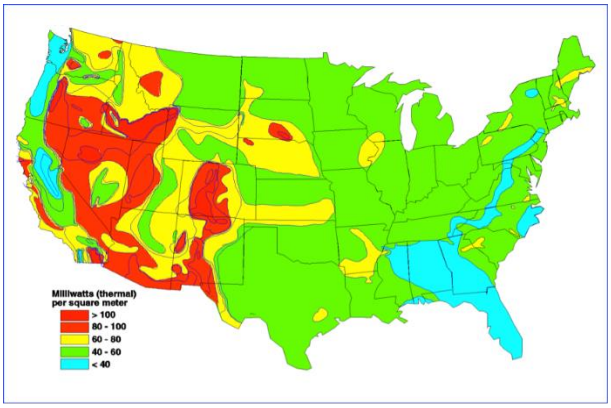


Figure 1-5. Heat map

2 System Requirements

2.1 Enumerated Functional Requirements

Identifier	PW	Requirement
REQ1a	5	The system shall attain real-time Tweets from Twitter and save them in its database.
REQ1b		The system shall acquire the historical health-related tweets the users sent before.
REQ1c		The system shall attain information of users that send health-related tweets. (Like username, location, follower numbers, Tweets Numbers, Twitter user URL, etc.)
REQ2	3	The system shall retrieve facilities' information (like location and type) from Google.
REQ3	4	The system shall acquire demographic data from demography text analytics API.
REQ4	5	The system shall obtain authorization from the third party API.
REQ5a	5	The system shall filter out unrelated tweets in the database.
REQ5b		The system shall classify relevant tweets data from the Database by different classification (like types of exercises, location, age, gender, types of food, etc.).
REQ6	4	The system shall screen out the tweets with duration of time or distance, etc.(intensity related)
REQ7	3	The system shall allocate tweets with different weight based on different types and sources.
REQ8	5	The system shall count the exercise-related numbers of tweets sent in specific area and time respectively and also calculate the percentage of exercise-related tweets.
REQ9	5	The system shall count the numbers of specific tweets and rank them. (Like running-related, swimming-related in the exercise field and like apple-related, banana-related in the diet-field, etc.)
REQ10	5	The system shall calculate the average intensity of users' exercises (including different types).
REQ11a	5	The system shall calculate the overlap between the group that post tweets about exercise and the group post tweets about wellness.
REQ11b		The system shall calculate the overlap between the group that post tweets about exercise and the group post tweets about food/diet.
REQ12	5	The system shall record same users' exercises-related tweets and calculate how regularly they do exercise.
REQ13	5	The system shall allow user knows how many calories are consumed

		in a certain time through the America. cities, devices, topic, etc.
REQ14	5	The system shall show heat map indicating active level of certain exercise in one specific area.
REQ15	5	The system shall display the amount of different kinds of tweets (based on location, types of exercise, time, etc.) in column chart.
REQ16a	5	The system shall compare the number of exercise-related tweets in different time period in a given area, calculate and show the trend.
REQ16b		The system shall show the trend of amount of exercised-related tweets in a given area by displaying two Heat Maps in the same time or sequentially.
REQ17	4	The system shall display the overlap between people who concern about wellness and who exercises and also between people who exercises and who talking about diet by pie chart.
REQ18	2	The system shall display the recent related tweets on the map (based on their location) and show what they just posted.
REQ19	5	The system shall give user suggestion on what kind of food he should take for doing a specific type of exercise.
REQ20	5	The system shall give out a public suggestion about exercise based on the average intensity and regularity of exercise posted by twitter users.
REQ21	3	The system shall allow users to register in order to obtain personal services provided by the website.
REQ22	3	The system shall allow registered users to modify their personal profiles.
REQ23	4	The system shall allow users to search for related information of interested exercise in given cities, type of exercise and date.
REQ24	5	The system shall give out an analysis of the user's health status and propose suggestions about exercises, diet and sleep for users based on the data derived from analysis (like average intensity and regularity).
REQ25	2	The system shall recommend most active users when users search for a given exercise program or login.
REQ26	2	The system shall recommend facilities' locations when users search for a given exercise program or login.
REQ27	3	The system shall allow registered users to share the analysis result on social websites such as Twitter and Facebook.
REQ28	3	The system shall allow the specific user knows how many time he / she takes exercise in a week.
REQ29	3	The system shall allow the specific user knows his / her ranking and can compared with other users.
REQ30	1	The system shall do the frequency and ranking analysis.
REQ31	1	The system shall allow registered users to invite friends for new users by sending emails or post their activities in the system on Facebook or Twitter.

REQ32	1	The system shall allow advertisers to post and change their advertisements on the website.
REQ33	3	The system shall allow the administrator to modify the database.
REQ34	2	The system shall allow the user to know the key words which represent the topic they are interested in.
REQ35	3	The system shall split the words in tweets, calculate in what frequency they appear and show the result in specific forms (like in table).
REQ36	3	The system shall analyze the correlation between different health topics and show the result in some forms (like the linear regression graph).
REQ37	3	They system shall do the sentiment analysis by evaluating corresponding mood state the tweets show. The system shall display the results.

Table 2-1. Functional requirements

2.2 Enumerated Nonfunctional Requirements

Identifier	PW	Requirement
REQ-38	5	The system should be accessible to any users through website or mobile application (IOS).
REQ-39	4	Related information of Twitter users should be updated when changes occur.
REQ-40	4	The system should remain running if there's update in the Twitter API.
REQ-41	3	Important data should have a backup in case the system goes down.
REQ-42	3	No raw data should be dependent on third parties other than Twitter.
REQ-43	3	The system should be fixed as soon as possible every time it goes down.
REQ-44	2	The user interface for both website and application should be users-friendly and easy to navigate.
REQ-45	2	All graphs related to data collected should be displayed in a simple and direct way.
REQ-46	2	All users' information should only be stored in the database of the system.
REQ-47	2	The system should support medium or high level of testing to find faults ahead of time.

Table 2-2. Non-functional requirements

2.3 On-Screen Appearance Requirements

2.3.1 Requirement descriptions

Identifier	PW	Requirement
REQ-48a	3	Registered user clicks on “Sign In” button to log in our application.
REQ-48b	3	People who want to register can click on the “Sign Up” button. They will be lead to a sign up window.
REQ-48c	4	This window shows recent tweets which about exercise.
REQ-48d	4	It offers both Android and IOS app to users in order to get a better user experience.
REQ-48e	3	User can share our website with other SNS.
REQ-49a	4	It allows the user to search for cities, communities, neighbor-hoods, relevant hash- tags, or other users.
REQ-49b	4	User scan find their partner by searching location, exercise program and time.
REQ-49c	5	Add leaderboard with more classifications that tweet about health in different states and city.
REQ-49d	5	This window shows data analysis about the amount of people in a given area exercise and the timeline of people exercise, etc.
REQ-49e	5	This window shows the frequency of exercise by different states and cities. Meanwhile, any new tweet can be found on this map.
REQ-50	3	On this page, you can find a lot of information about your entered community, like Health score, Exercise rank, Total number of twitter exercisers, Exercisers percentage, Exercise frequency, Exerciser distribution, Popular sports, devices and gyms.
REQ-51	4	On this page, you can view device introduction, positive experience, negative experience, device overall grade, and comments. Clicking on the “share” button, user will be linked to a “share your comment” page. Clicking on the name in the comments module, user will be linked to this reviewer’s twitter home page.
REQ-52	3	This page is where the user manages their personal account such as changing their login or password or location, etc.

Table 2-3. On-screen appearance requirements

2.3.2 Website appearance

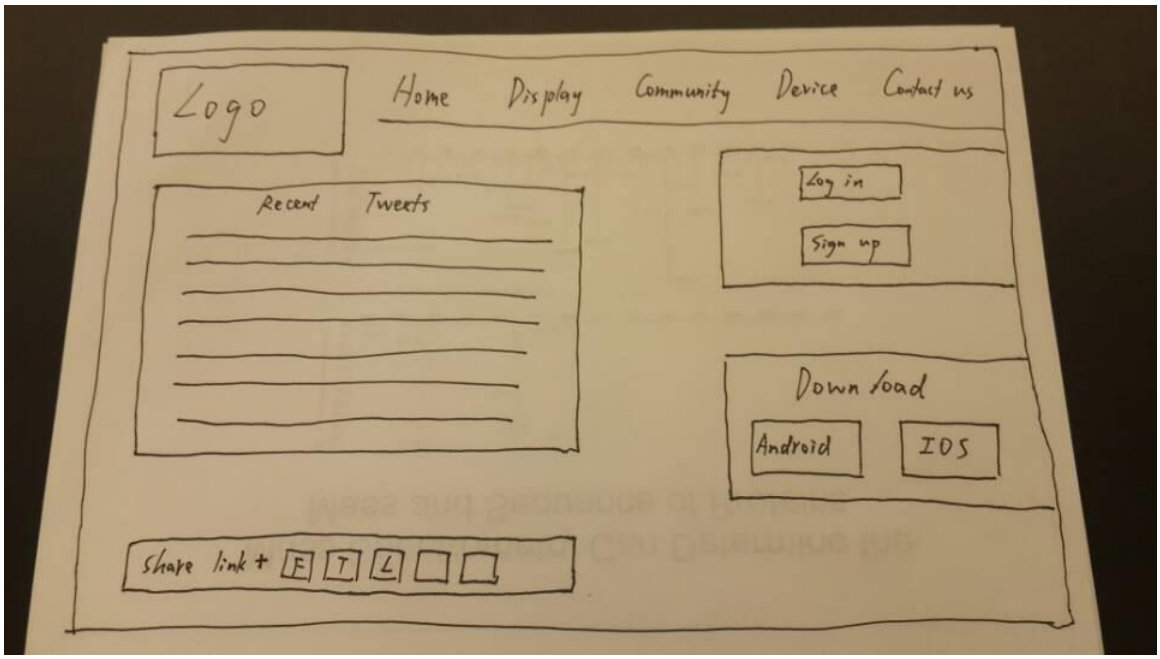


Figure 2-1. Homepage of the website

The figure 2-1 shows the homepage of our website. The users can easily see the menu: Home, Display, Community, Device and Contact Us. When clicking the Home menu, the users can see the recent tweets about health and exercise. Also, members can login to manage their personal profile. The Android and IOS apps are provided to free download. When clicking the Display menu, the users can browse several analysis charts or graphs about health and exercise. Details are shown in figure 2-2. When clicking the Community menu, members who has logged in can use the forum to contact with other members by asking or answering questions. When clicking the Device menu, the users can see the leaderboard of device recommended by members. When clicking the Contact us menu, the users can contact us by making any suggestion or improvement.

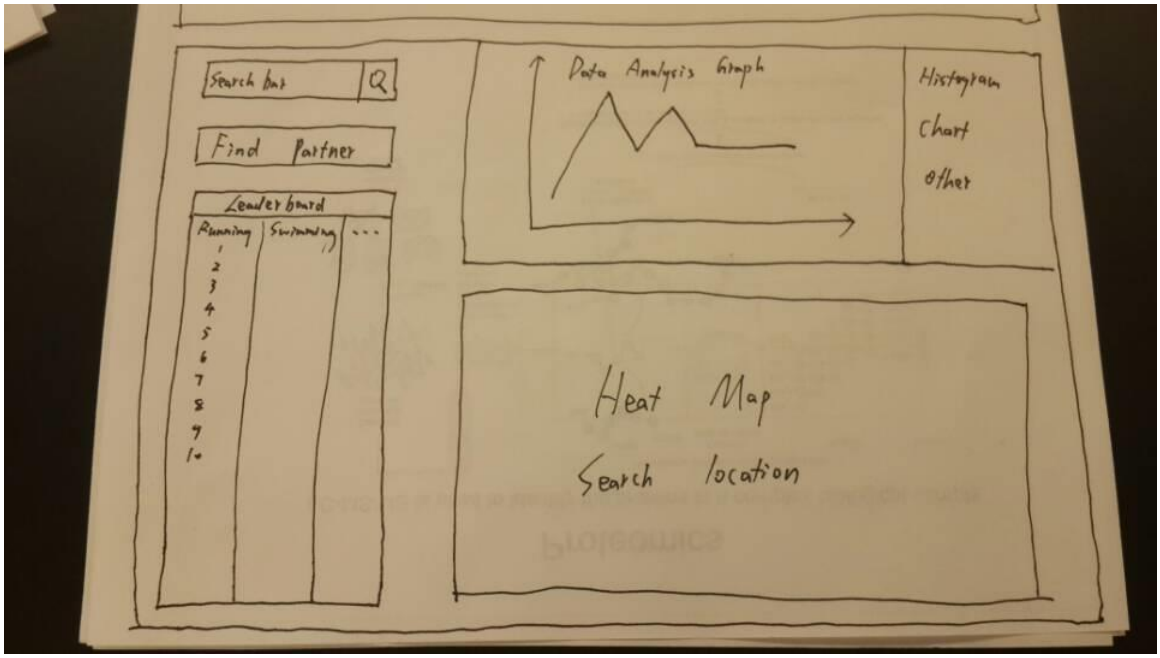


Figure 2-2. Display page

The figure 2-2 shows the details about the display menu. Firstly, the users can use the search bar to search exercise they like. Secondly, the leaderboard shows the rankings of different exercise such as running and swimming. Thirdly, the users can see the analysis of tweets about health and exercise, in order to know the amount of people divided by area, time, demographics and exercise type. Also, the state map and marker map are provided to browse.

3 Functional Requirements Specification

3.1 Stakeholders

Our system focuses on offering exercise and health field problems solutions, so the stakeholders include these people and organizes:

- System Users

  Defined as most important customers. For exercise topic, they can use our system to view exercise heat map, leaderboard, score system, to find friends and gym location, to find suitable sports for individual. For healthy related topic they can find useful information about diet and sleep on twitter via our system, can acquire suggestions on health issues through our system.

- Systems architects and developers

  Defined as system supporters. They take responsibility on building the system that fulfills the user's requirements, testing and maintaining the system, and provide technical support to other stakeholders.

- Academic Researchers

  Defined as third party, can use our system to gain information about people's health and exercise situation for academic research purpose.

- Gym and sports equipment advertisers

  Defined as our profit supporters, can pay to put advertisement in our system. Advertisement is restricted to healthy and exercise fields.

3.2 Actors and Goals

- User (Initiating type)

Goals: to interact with the system, acquire exercise and health information they need, make friends and find useful exercise places via the system.

- Administrator (Initiating type)

  Goals: has the top priority to collect data, access, manage, and maintain the database, provide service to the user.

- Advertiser: (Initiating type)

  Goals: analyze data, put and manage advertisement on the system to attract customers to buy their commodities.

- Google server and database (Participating type)

- Twitter server and database (Participating type)

- Demography text analytics server and database (Participating type)

- Our server and database (Participating type)

3.3 Use Cases

3.3.1 Casual description

The summary use cases are as follow:

- UC2 User information adding

  Allow the user to create account in the Health Monitoring System for some private health-related services.

  Derived from REQ20, 21.

- UC3 Advertiser information adding

  Allow the administrator to create account in the Health Monitoring System for advertisers.

  Derived from REQ20, 21.

- UC4 Data deleting

  Allow the administrator to delete useless, incorrect data or do some necessary adjust in system's database.

  Derived from REQ32.

- UC5 Data collecting & classifying

  Allow the administrator to retrieve Twitter's users' data from Twitter and demography text analytics database, classify them by some specific sort (like location, exercise type, etc.) and store these data in system's database.

  Derived from REQ1-3, 5-7.

- UC6 Third party API auth.

  Allow the administrator to get verification of accessing and retrieving data from Twitter and demography text analytics.

  Derived from REQ4.

- UC8 Advertisement updating

  Allow the advertiser to change the advertisement post on the website after login.

  Derived from REQ31.

- UC11 Exercise heat

  Allow the public user to have a glance at exercise heat in area, time, type, demography and also their trends. (Shown by heatmap, pie chart, column chart, etc.)

  Derived from REQ14-16.

- UC12 Tweet sentiment and part of speech

  Allow user know the mood distribution in a certain area.

  Derived from REQ20, REQ37.

- UC13 Weighted average calorie

Allow user knows how many calories are consumed in a certain time through the America.

Derived from REQ13.

- UC14 Correlation between health topics

  Allow the public user to check the overlap between people who concern about wellness and who exercises and also between people who exercises and who talking about diet by pie chart.

  Derive from REQ17.

- UC15 Word Frequency

  Allow the user to know the key words which represent the topic they are interested in.

  Derived from REQ34.

- UC16 Exercise Frequency & User Ranking

  Allow the specific user knows how many time he takes exercise in a week and know his ranking and can compared with other user.

  Derived from REQ28-30.

- UC17 Personal Diagnosis

  Allow the login user acquires suggestions about how their exercise should be (including intensity and regularity), nearest facilities, recommendation of friends (like twitter users that share the same hobby and have high activities).

  Derived from REQ24-27.

- UC18 Healthy Food

  Give user suggestion on what kind of food he should take for doing a specific type of exercise.

  Derived from REQ19.

- UC19 Login

Allow the user (including the normal user and advertiser) to login and gain specific services (Like normal user would have the system analyzed their health status and provided he/she related services, advertiser could login to change their previous advertisement).

Derived from REQ21.

- UC20 Data analyzing

Allow the advertiser to analyzing the data stored in database like count the total and exercise-related numbers of tweets, rank different types tweets by amount, calculate the average intensity of users' exercises (including different types), etc.

Derived from REQ8-12.

3.3.2 Use case diagram

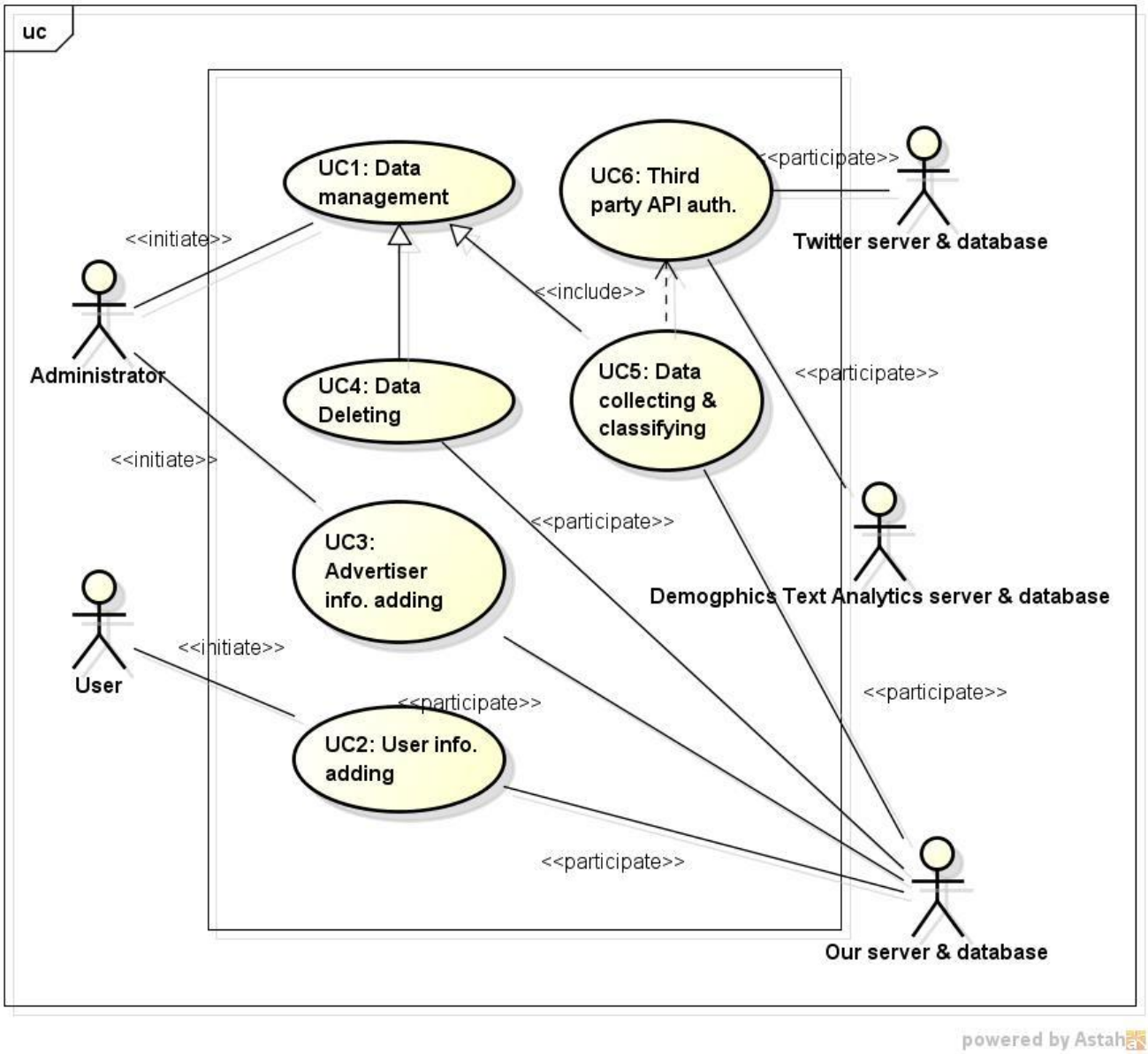


Figure 3-1 Use case diagram of server and database management subsystem

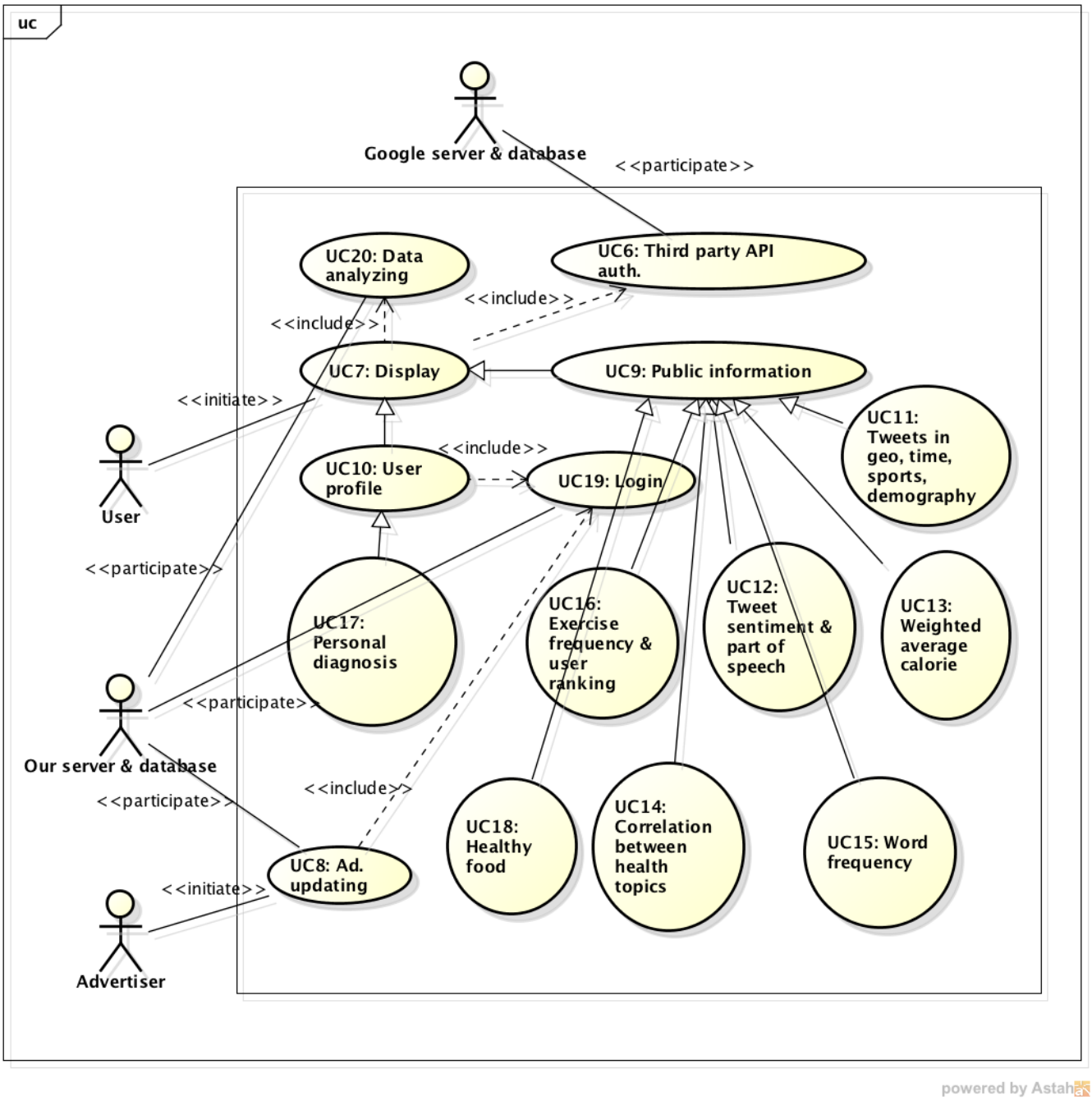


Figure 3-2 Use case diagram of displaying subsystem

3.3.3 Traceability matrix

REQ	PW	UC 2	UC 3	UC 4	UC 5	UC 6	UC 8	UC 11	UC 12	UC 13	UC 14	UC 15	UC 16	UC 17	UC 18	UC 19	UC 20
1a	5				√												
1b					√												
1c					√												
2	3				√												
3	4				√												
4	5					√											
5a	5				√												
5b					√												
6	4				√												
7	3				√												
8	5																√
9	5																√
10	5																√
11a	5																√
11b																	√
12	5																√
13	5									√							
14	5							√									
15	5							√									
16a	5							√									
16b								√									

17	4										√					
18	2							√								
19	5													√		
20	5								√							
21	3	√	√												√	
22	3	√	√													
23	4							√								
24	5												√			
25	2												√			
26	2												√			
27	3												√			
28	3											√				
29	3											√				
30	1											√				
31	1															
32	1						√									
33	3			√												
34	2										√					
35	3									√						√
36	3									√						√
37	3								√				√			√
Max PW		3	3	3	5	5	1	5	5	5	4	2	3	5	5	3	5
Total PW		6	6	3	24	5	1	21	8	5	10	2	7	15	5	3	34

Table 3-1 Traceability matrix

3.3.4 Fully-dressed description

Use Case UC-5:	**Data collecting & classifying**
Related Requirements:	REQ1, REQ2, REQ3, REQ5, REQ6, REQ7
Initiating Actor:	Developer
Actor's Goal:	To find useful information using keyword and store them in our database
Participating Actors:	Twitter's server & database, System's server & database
Pre-conditions:	·Developer has been authorized to crawl information from Twitter ·Database is all set and ready to store new data
Post-conditions:	Useful information from related Tweets is stored in system's database
Flow of Events for Main Success Scenario: → 1. The developer sets up the database and decide the keywords for using Twitter's API ← 2. Twitter gives out the information filtered by the keywords → 3. The developer collects the data and store them in the system's database	

Table 3-2 Fully-dressed description UC-5

Use Case UC-11:	**Exercise heat**
Related Requirements:	REQ14, REQ15, REQ16
Initiating Actor:	User
Actor's Goal:	To get visual charts or diagrams of the information offered by the system
Participating Actors:	System's server and database, Google server & database
Pre-conditions:	Related information is stored in the system's database
Post-conditions:	System displays the chosen diagrams on the screen
Flow of Events for Main Success Scenario: → 1. The user navigate to the exercise intensity interface → 2. The user choose certain distribution and type to see the statistical graphs ← 3. The system displays the diagrams as requested	

Table 3-3 Fully-dressed description UC-11

Use Case UC-12:	**Tweet Sentiment**
Related Requirements:	

Initiating Actor:	Users
Actor's Goal:	To obtain the sentiment analysis of every state in United States.
Participating Actors:	System's server & database
Pre-conditions:	·Related data are stored in the system's database ·Collected data are assumed to be correct
Post-conditions:	Every state is assigned with its calculated sentiment level
Flow of Events for Main Success Scenario: → 1. The user choose the sentiment analysis feature to see results → 2. The system receives the user's request and accesses the database to calculate the corresponding data ← 3. The database gives out the statistical result of the analysis	

Table 3-4 Fully-dressed description UC-12

Use Case UC-13:	**Weighted Average Calorie**
Related Requirements:	
Initiating Actor:	Users
Actor's Goal:	To obtain the calories analysis of Twitter users
Participating Actors:	System's server & database
Pre-conditions:	·Related data are stored in the system's database ·Collected data are assumed to be correct
Post-conditions:	The total consumed calories trend in the time of day
Flow of Events for Main Success Scenario: → 1. The user choose the calories analysis feature to see results → 2. The system receives the user's request and accesses the database to calculate the corresponding data ← 3. The database gives out the statistical result of the analysis	

Table 3-5 Fully-dressed description UC-13

Use Case UC-15:	**Word frequency**
Related Requirements:	REQ34
Initiating Actor:	Developer
Actor's Goal:	To find the exercise-related words with high frequency in tweets
Participating Actors:	Twitter API
Pre-conditions:	·The key words have been set to make the analysis ·System has the access to Twitter API

Post-conditions:	The set of exercise-related words with high frequency are obtained by the developer
Flow of Events for Main Success Scenario: → 1. The developer choose certain key words → 2. The system gets the key words and connect to Twitter API ← 4. The system stores the result into the database	
Use Case UC-15:	

Table 3-6 Fully-dressed description UC-15

Use Case UC-17:	**Personal suggestions**
Related Requirements:	REQ24, REQ25, REQ26, REQ27
Initiating Actor:	User
Actor's Goal:	To acquire suggestions about their daily exercise, facilities, devices, and recommendations of partners with similar interests
Participating Actors:	System's server & database, Twitter's server & database
Pre-conditions:	·User has logged in onto the system ·System has the authorization to get information of user's profile
Post-conditions:	Suggestions are showed on the website in the form of useful links
Flow of Events for Main Success Scenario: → 1. The user navigate to the main interface of the system → 2. The user logs into the system using their username and password ← 3. The system lists out the related links concerning their daily exercise and the profile links of Twitter users with same interest	

Table 3-7 Fully-dressed description UC-17

Use Case UC-20:	**Data analyzing**
Related Requirements:	REQ8, REQ9, REQ10, REQ11a, REQ11b, REQ12
Initiating Actor:	Developer
Actor's Goal:	To obtain the statistical information from the data collected from Twitter
Participating Actors:	System's server & database
Pre-conditions:	·Related data are stored in the system's database ·Collected data are assumed to be correct
Post-conditions:	All statistical data are stored in different tables in the

	database

Flow of Events for Main Success Scenario:

→ 1. The developer chooses certain data from the database

→ 2. The developer decides methods to analyze the selected data

← 3. The database gives out the statistical result of the analysis

Table 3-8 Fully-dressed description UC-20

3.4 System Sequence Diagrams

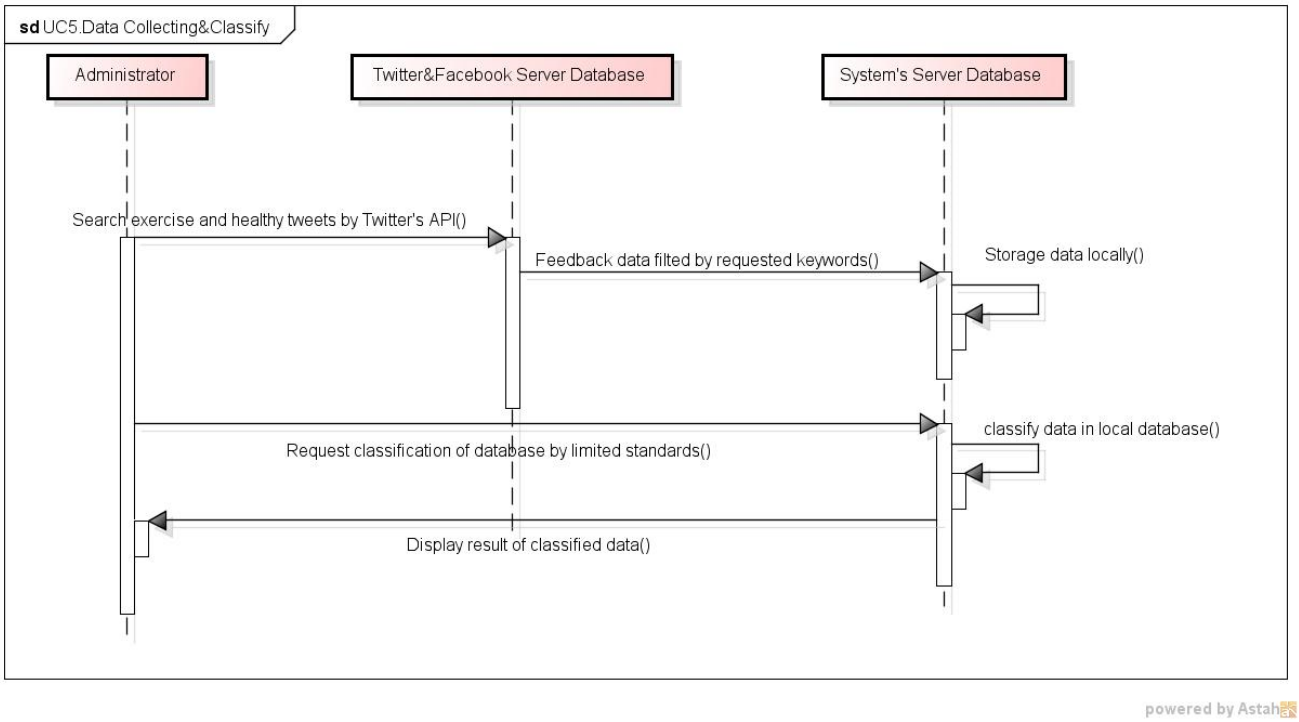


Figure 3-3 Use case diagram for UC-5

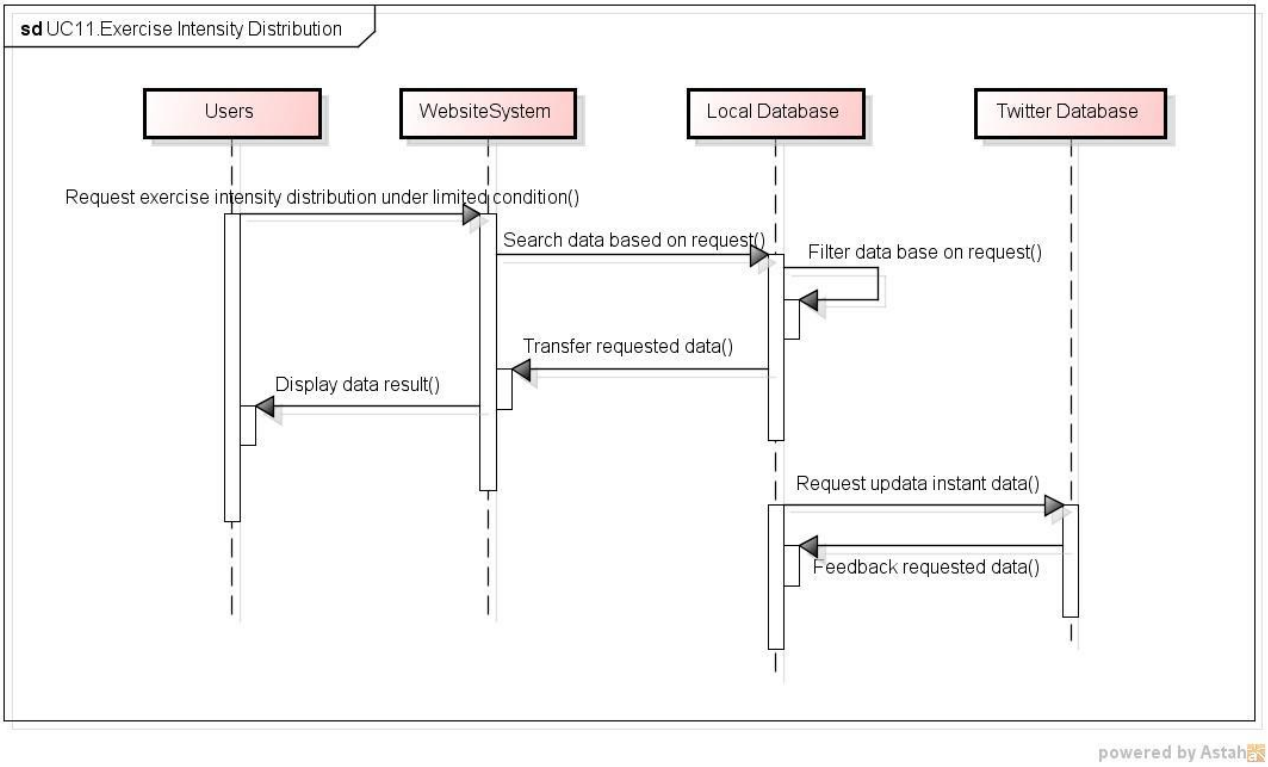


Figure 3-4 Use case diagram for UC-11

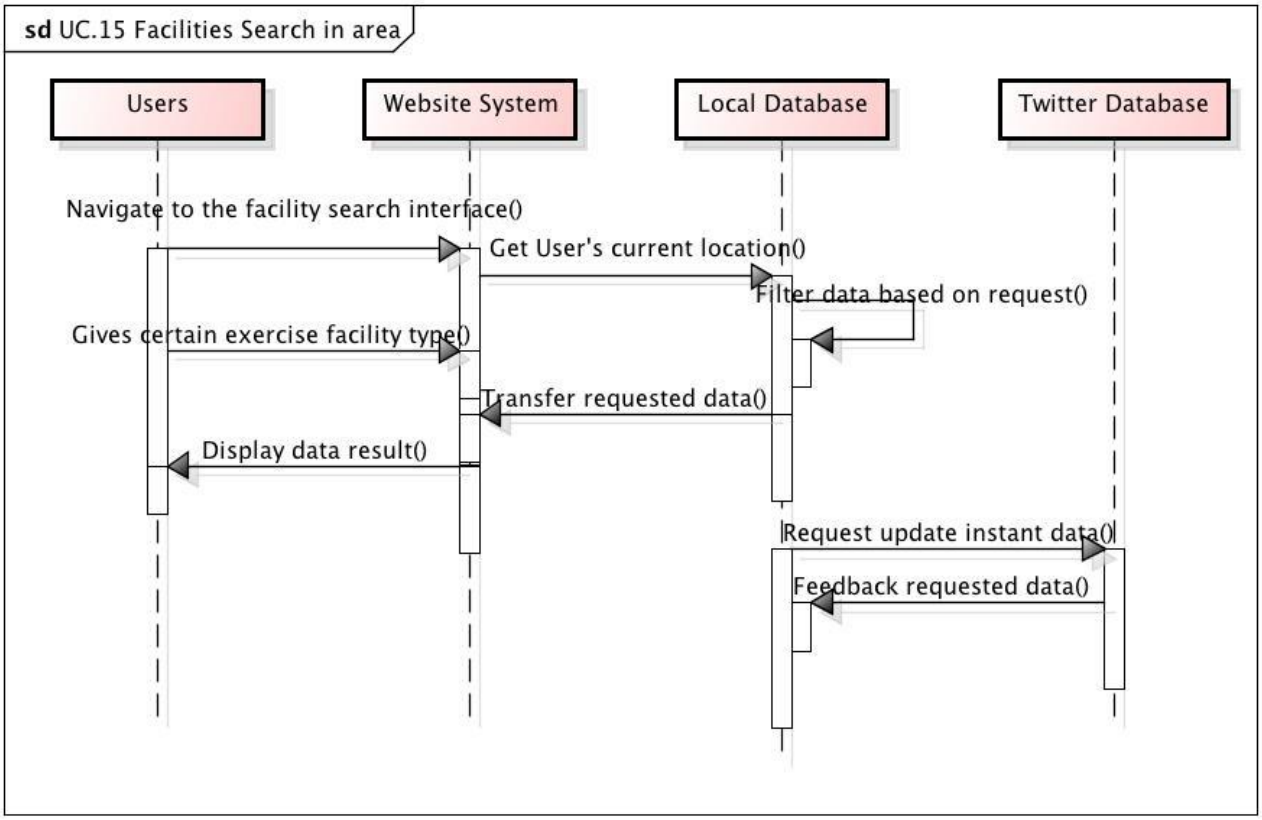


Figure 3-5 Use case diagram for UC-15

3.5 Effort Estimation using Use Case Points

Use Case Points (UCP) method provides the ability to estimate the person-hours a software project requires based on its use cases. UCP method analyzes the use case actors, scenarios, nonfunctional requirements, and environmental factors and joins them in a simple equation: CP = UUCP ∗ TCF ∗ ECF.

- Unadjusted Use Case Points (UUCP) – Measures complexity of functional requirements
- Technical Complexity Factor (TCF) – Measures complexity of nonfunctional requirements.
- Environmental Complexity Factor (ECF) – Assesses development teams experience and heir development environment

3.5.1 Unadjusted use case points

UUCP are computer as a sum of the following two components:

- Unadjusted Actor Weight (UAW) – Combined complexity of all the actors in all the use cases
- Unadjusted Use Case Weight (UUCW) – Total number of activities contained in all the use case scenarios

UUCP = UAW + UUCW

3.5.2 Unadjusted actor weight

The weights for Actor classification are computed via the following table: Actor classification and associated weights, Actor Classification for Health Monitoring Analytics System

Actor	Description of Characteristics	Complexity	Weight
User	Interact with the system, acquire exercise and health information they need, make	Complex	3

	friends and find useful exercise places via the system.		
Administrator	Has the top priority to collect data, access, manage, and maintain the database, provide service to the user	Average	2
Advertiser	Analyze data, put and manage advertisement on the system to attract customers to buy their commodities.	Simple	1
Google server and database	Google server and database is a system interacting through application programming interface.	Simple	1
Twitter server and database	Twitter server and database is another system interacting through application programming interface.	Average	2
Demography text analytics server and database	Demography text analytics server is another system interacting through application programming interface.	Average	2
Our server and database	Database is another system interacting through a protocol.	Complex	3

Table 3-9 Unadjusted Actor Weight

UAW(Health Monitoring Analytics) = 2 ∗ Simple + 3 ∗ Average + 2∗ Complex

=2∗ 1+3∗ 2+2∗ 3 = 14

3.5.3 Unadjusted use case weight

Use case classification for Health Monitoring Analytics

Use Case	Description of Characteristics	Complexity	Weight
User information adding (UC2)	Allow the user to create account in the Health Monitoring System for some private health-related services.	Average	10
Advertiser information adding (UC3)	Allow the administrator to create account in the Health Monitoring System for advertisers.	Average	10
Data deleting (UC4)	Allow the administrator to delete useless, incorrect data or do some necessary adjust in system's database.	Simple	5
Data collecting & classifying (UC5)	Allow the administrator to retrieve Twitter's users' data from Twitter and demography text analytics database, classify them by some specific sort (like location, exercise type, etc.) and store these data in system's database.	Complex	15
Third party API auth (UC6)	Allow the administrator to get verification of accessing and retrieving data from Twitter and demography text analytics.	Simple	5
Advertisement updating (UC8)	Allow the advertiser to change the advertisement post on the website after login.	Simple	5

Exercise heat (UC11)	Allow the public user to have a glance at exercise heat in area, time, type, demography and also their trends. (Shown by heatmap, pie chart, column chart, etc.)	Complex	15
Tweet sentiment & part of speech (UC12)	Allow user know the mood distribution in a certain area.	Average	10
Weighted average calories (UC13)	Allow user knows how many calories are consumed in a certain time through the America.	Average	10
Correlation between health topics (UC14)	Allow the public user to check the overlap between people who concern about wellness and who exercises and also between people who exercises and who talking about diet by pie chart.	Average	10
Word frequency (UC15)	Allow the user to know the key words which represent the topic they are interested in.	Simple	5
Exercise frequency & user ranking (UC16)	Allow the specific user knows how many time he takes exercise in a week and know his ranking and can compared with other user.	Average	10

Personal suggestion (UC17)	Allow the login user acquires suggestions about how their exercise should be (including intensity and regularity), nearest facilities, recommendation of friends (like twitter users that share the same hobby and have high activities).	Complex	15
Healthy food (UC18)	Give user suggestion on what kind of food he should take for doing a specific type of exercise.	Simple	5
Login (UC19)	Allow the user (including the normal user and advertiser) to login and gain specific services (Like normal user would have the system analyzed their health status and provided he/she related services, advertiser could login to change their previous advertisement).	Simple	5
Data analyzing (UC20)	Allow the advertiser to analyzing the data stored in database like count the total and exercise-related numbers of tweets, rank different types tweets by amount, calculate the average intensity of users' exercises (including different types), etc.	Complex	15

Table 3-10 Use case classification

UUCW(Health Monitoring Analytics) = 6 * simple + 6 * Average + 4 * Complex = 6 * 5 + 6 * 10 + 4 * 15

= 150

UUCP(Health Monitoring Analytics) = UAW + UUCW = 14 + 150 = 164

3.5.4 Technical complexity factor

Technical Complexity Factor (TCF) is computed using thirteen standard technical actors to estimate the impact of productivity of the nonfunctional requirements for the application. We then need to assess the perceived complexity of each technical factor in the context of the project. A perceived complexity value is between 0 and 5, with 0 meaning trivial effort, 3 meaning average effort and 5 meaning major effort. Each actor's weight is then multiplied by perceived complexity factor to produce calculated factor. Two constants are used with TCF. The constants utilized are C1 = 0.6 and C2 = 0.01.

Technical complexity factors for Health Monitoring Analytics: PC = Perceived Complexity, CF = Calculated Factor

Technical Factor	Description of Characteristics	Weight	PC	CF
T1	Distributed, web-based system	2	3	6
T2	User expects good performance, but will tolerate network latency	1	3	3
T3	End-user expects efficiency, which is achieved through caching	1	4	4
T4	Internal processing required multiple interactions with other subsystems	1	4	4
T5	No requirement for reusability	1	0	0
T6	No user installation required	0.5	2	1

T7	Ease of use was very important	0.5	5	2.5
T8	No requirement for portable	2	0	0
T9	Relatively simple to add new features	1	2	2
T10	Concurrent use is required	1	4	4
T11	No requirement for security	1	0	0
T12	No direct access for third parties	1	1	0
T13	No training required	1	0	0

Table 3-11 Technical Complexity Factors

TCF = 0.6 + (0.01 * 28.5) = 0.885

This results in a decrease of the UCP by 11.5%

3.5.5 Environment complexity factor

Technical Complexity Factor (TCF) is computed using thirteen standard technical actors to estimate the impact of productivity of the nonfunctional requirements for the application. We then need to assess the perceived complexity of each technical factor in the context of the project. A perceived complexity value is between 0 and 5, with 0 meaning trivial effort, 3 meaning average effort and 5 meaning major effort. Each actor's weight is then multiplied by perceived complexity factor to produce calculated factor. Two constants are used with TCF. The constants utilized are C1 = 0.6 and C2 = 0.01.

Technical complexity factors for Health Monitoring Analytics: PC = Perceived Complexity, CF = Calculated Factor

Technical Factor	Description of Characteristics	Weight	PC	CF
T1	Distributed, web-based system	2	3	6
T2	User expects good performance, but will tolerate network latency	1	3	3
T3	End-user expects efficiency, which is achieved through caching	1	4	4
T4	Internal processing required multiple interactions with other subsystems	1	4	4
T5	No requirement for reusability	1	0	0
T6	No user installation required	0.5	2	1
T7	Ease of use was very important	0.5	5	2.5
T8	No requirement for portable	2	0	0
T9	Relatively simple to add new features	1	2	2
T10	Concurrent use is required	1	4	4
T11	No requirement for security	1	0	0

T12	No direct access for third parties	1	1	0
T13	No training required	1	0	0

Table 3-12 Technical Complexity Factors

TCF = 0.6 + (0.01 * 28.5) = 0.885

This results in a decrease of the UCP by 11.5%

3.5.6 Environment complexity factor

The Environment Complexity Factor (ECF) is computed utilizing eight standard environmental factors to measure the experience level of the people on the project and he stability of the project. We then need to assess the perceived impact based on perception that factor has on projects success. 1 means strong negative impact, 3 is average and 5 means strong positive impact.

TCF is computed utilizing thirteen standard technical factors to estimate the impact of productivity of the nonfunctional requirements for the application. We then need to assess the perceived complexity of each technical factor in the context of the project. A perceived complexity value is between 0 and 5 with 0 meaning that it is irrelevant, 3 means average effort and 5 means major effort. Each factors weight is then multiplied by perceived complexity factor to produce calculated factor. Two constants are used with CF. the constants utilized are C1 = 1.4 and C2 = -0.03. Environmental Complexity Factors for Health Monitoring Analytics:

Environment Factor	Description of Characteristics	Weight	PC	CF
E1	Beginner familiarity with UML-based development	1.5	1	1.5

E2	Half of team has familiarity	0.5	3	1.5
E3	Some knowledge of object-oriented approach	1	3	3
E4	Average lead analyst	0.5	2	1
E5	Highly motivated overall	1	4	4
E6	Requirements were stable	2	5	101.5
E7	Student staff (part-time)	-1	4	-4
E8	Used new programming languages but resources were readily available	-1	5	-5

Table 3-13 Environmental Complexity

ECF = 1.4 – (0.03 * 12) = 1.04

This results in an increase of UDP by 4%.

3.5.7 Calculating the use case points

As mentioned earlier, UCP = UUCP × TCF × ECF.

From above calculations, UCP variables have the following values:

UUCP = 164

TCF = 0. 885

ECF = 1.04

UCP = 164 × 0.885 × 1.04 =150.95 or 151 use case points.

3.5.8 Deriving project duration from use-case points

UCP is a measure of software size. We need to know the teams rate of progress through the use cases. We need to utilize the UCP and Productivity Factor (PF) to determine duration. The equation for computing Duration is: Duration = U C P × P F

Productivity Factor is the ratio of development person-hours needed per use case point. Assuming a PF = 28, the duration of our system is computed as follows:

D u r a t i o n = U C P X P F = 151*28 = 4228

4228 person-hours for the development of the system.

We have 6 developers in our team, and each developer on average spent 20 hours per week on project tasks. This means that team makes 6 * 20 = 120 hours per week. Dividing 4228 person-hours by 120 hours per week, we obtain the total of approximately 35 weeks to complete this project.

4 User Interface Specification

4.1 Preliminary Design

This section represents our preliminary design and analysis. Here is the proposed main user interface webpage.

Figure 4-1 webpage

- The visitor (user who has not registered) can browse both “Public Display”, ”Our Features ”and “Private Profile” sections. In “Public Display” section, the visitor can see the geographical distribution, time variation, demography analysis, exercising type, healthy food and correlation. In “Our Feature” section, the visitor can see the heat map, leaderboard, demographic analysis, sentiment analysis, colories consumption and marker map.

- The visitor is able to see different heat maps varied by area, time, type and demography. He/she can select the display way such as in area, time, type or demography. The map also enables dragging and zooming in/out.

- The visitor can see the leaderboard by selecting different buttons, which includes ranking in users varied by area, type and demography.

- The visitor is able to see different state maps varied by time, type and demography.

- The visitor is able to see marker maps, which can show the real-time update whenever someone post a new tweet about health or exercise.

- If the visitor wants to know the correlation/overlap between the groups that exercises and the group that discusses health and wellness, he/she can easily fins the result by browsing the part of “correlations”.

- The visitor may get some suggestion about frequency, time and amount of exercise varied by demography and exercise type.

- The member (user who has registered) can use these features stated above that all visitors can use.

- The member can log in to use some additional features about personal profile, such as personal overall ranking, exercise record and suggestion.

- After logging in, the member is able to know his/her overall ranking in all of our website members. The ranking is varied by exercise type.

- The member can also check personal record of exercising in time varied by type.

- The website offers particular suggestion of exercising, facility, device and friend, helping the member enjoy the process of exercising in a better way.

Above are the main features for our website. More features would be added in as stated in the system requirements.

4.2 User Effort Estimation

Our website is very easy to use. We try to design it with the minimum user effort to accomplish their goal.

For the visitor who just wants to check the health activities awareness in certain city and obtain some statistical data:

- NAVIGATION (several keystrokes and one click)

Navigate to our software webpage (several keystrokes; inputting http address)

Main interface page is brought to the visitor
Close our webpage when finished (one click)

- DATA ENTRY (several keystrokes and clicks)

Select one button from “heat map”, “leaderboard”, “demographic analysis”, “sentiment analysis”, “colories consumption” and “marker map” to see different features (one click)
Select one kind of heat map distribution (one click)
See the correlation/overlap between the group that exercises and the group that discusses health and wellness (one click)
Select one button from “area”, “type” or “demography” to see exercise suggestions (one click)
Select location and exercise type to search facilities nearby. (three clicks)
Select location, time and exercise type to find partners. (four clicks)
Select one kind of heat map distribution (one click)

For the visitor who wants to register to a member:

- REGISTRATION NAVIGATION (two clicks)

Click on log in button (one click)
A new page pops up asking for user name and password, and an option of registration. (0 effort)
Click on the register link (one click)
A registration page pops up asking for information
Done with registration.

- INFORMATION FILLING (several keystrokes)

Account Registration Part 1 (Instructions and how to use the application)
Account Registration Part 2 (Disclaimers and Permissions)
Account Registration Part 3 (User information)
Done with registration and a personal page is set at the same time.

For the member who wants to log in:

- LOG IN NAVIGATION (one click)

Click on log in button (one click)
A new page pops up asking for user name and password. (two keystrokes)
Done with registration.

5 Domain Analysis

5.1 Domain Model

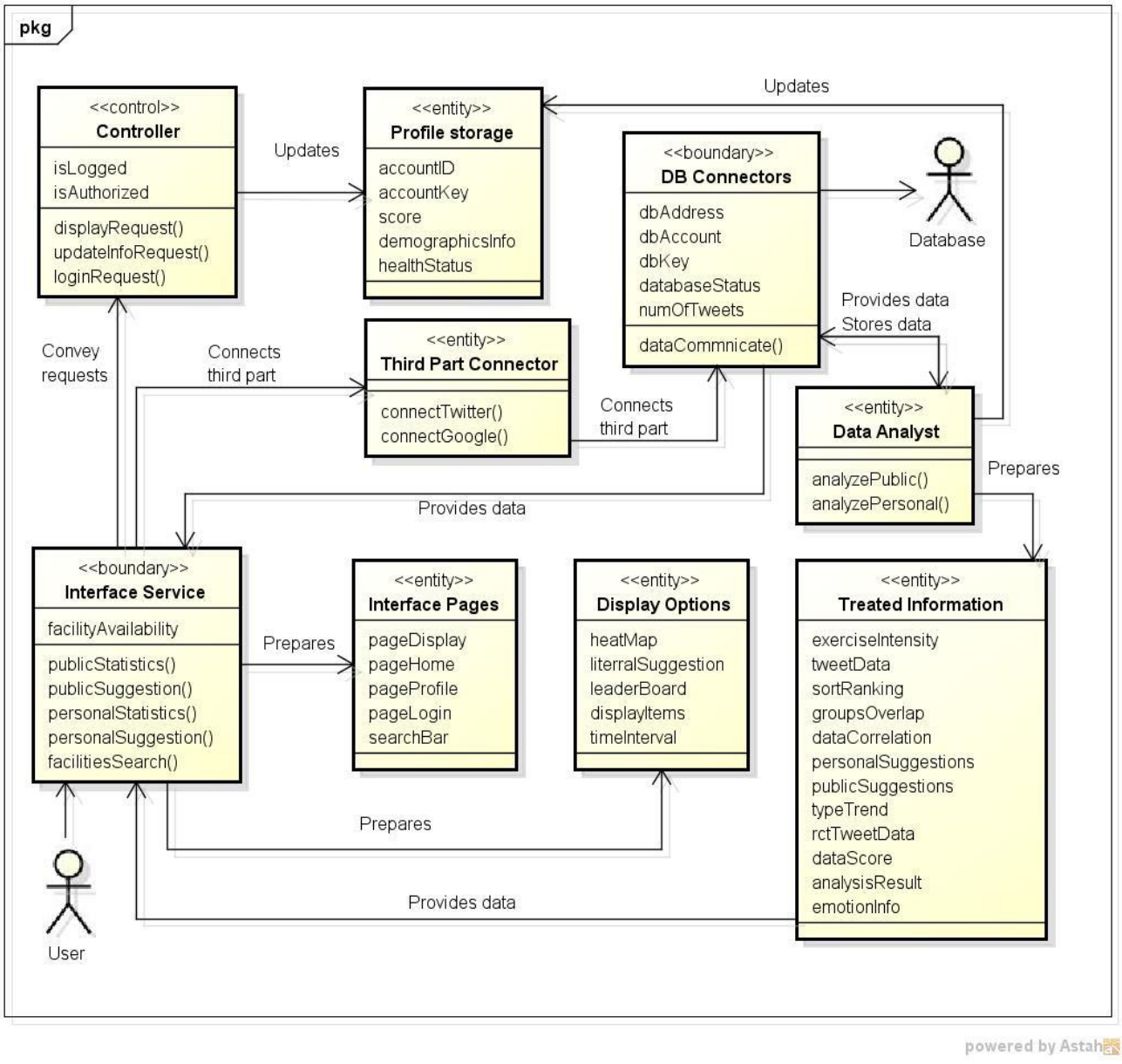


Figure 5-1 Domain model diagram

5.1.1 Concept definitions

We derive the domain model concepts from detailed user cases. Table 5-1 lists the responsibilities and the assigned concepts.

First, the responsibilities 1, 2, 5, 6, 7, 8, 9, 10, 11, 14 are basically identified from the scenarios of UC11, 12, 13, 14, 15. All of them are related to the information display and the interaction between users and the system. Second, the responsibilities 3, 4 are derived from UC2, 3, 8, 16, 17, 18, 19. They deal with the login issues and personal service provided by the system. At last, the responsibilities 12, 13 generate from UC5, 20. Both of them are components that assist the system to analyze and classify tweets data.

Responsibility	Type	Concept
R1: Prompt the user to make movement for available services.	D	Controller
R2: Handle requests from users.	D	Controller
R3: Deal with login issue (check users key, approve or deny).	D	Controller
R4: Container for the collection of valid keys and account profile associated with users.	K	Profile Storage
R5: Show pages for user to create account, login and logout.	D	Interface Service
R6: Show search engine and diversity display buttons for user to choose.	D	Interface Service
R7: Display related information in literal, numerical, graphical and map forms.	D	Interface Service
R8: Static websites and mobile phone interface that shows the user the current context, what services could be used, and outcomes of the previous request.	K	Interface Pages
R9: Specific parameters and options for information display, including search options, types of graph and display items.	K	Display Options
R10: Related information after all analysis, inferring and statistics for display.	K	Treated Information
R11: Manage interactions with the database.	D	DB Connector
R12: Classify, do statistics, analyze and infer related information for suggestion, statistics display and recommendation.	D	Data Analyst
R13: Retrieve related data from Twitter Database.	D	Third Part Connector

R14: Retrieve related services from Google Database.	D	Third Part Connector

Table 5-1 Concept definitions

5.1.2 Association definitions

Some concepts defined above need to work together in order to achieve specific functions. The concepts work together are called concept pair. The definition and description of concept pair of the system are listed in Table 5-2.

Concept pair	Association description	Association name
Controller ↔ Profile Storage	Controller updates User Profile when the user change his/her information.	Updates
Controller ↔ Interface Service	Controller passes requests to Interface Service and receives pages prepared for displaying.	Convey requests
Interface Service ↔ Interface Pages	Interface Service prepares the Interface Pages.	Prepares
Interface Service ↔ Display Options	Interface Service prepares the Display Options.	Prepares
Interface Service ↔ DB Connector	Database Connection passes the retrieved data to Interface Service to render them for display and show.	Provides data
Interface Service ↔ Treated Information	Interface Service extracts information from Treated Information for display.	Provides data
Interface Service ↔ Third Part Connector	Third Part Connector enable Interface Service to connect the Third Part to ask for service (Like the google map and graphs).	Connects Third Part
DB Connector ↔ Third Part Connector	Third Part Connector enable DB Connector to connect the Third Part to retrieve related data.	Connects Third Part
DB Connector ↔ Data Analyst	Database Connection passes the retrieved data to Data Analyst for analysis, inferring and statistics. Data Analyst stores useful data back into Database after analysis.	Provides data, Stores data
Data Analyst ↔	Data Analyst prepares the Treated	Prepares

Treated Information	Information.	
Data Analyst ↔ Profile Storage	Data Analyst changes the information (like score and health status) in Profile Storage.	Updates

Table 5-2 Association definitions

5.1.3 Attribute definitions

Attributes of domain concepts are derived in Table 5-3.

Concept	Attribute	Attribute Description
Controller	displayRequest	Send user's requests to retrieve display service
	updateInfoRequest	Send user's requests to retrieve profile change service
	loginRequest	Use the data the user input to request a login operation
	isLogged	Identity parameter to determine whether the user is login
	isAuthorized	Identity parameter to determine whether the user is authorized
Profile Storage	accountID	Identity number used to determine the user
	accountKey	Specific key to determine the user's credentials
	score	User's current score in the score system
	demographicsInfo	User's personal information like age, gender, location, etc.
	healthStatus	User's exercise-related status given by the analyst
Interface Service	publicStatistics	Show the information about public health-related statistic
	publicSuggestion	Show the exercise suggestions for the whole public
	personalStatistics	Show the information about personal health-related statistic
	personalSuggestion	Show the health-related suggestions for user
	facilitiesSearch	Search for facilities information with given conditions
	facilityAvailability	Identity parameter to determine whether facility is available

Interface Pages	pageDisplay	Pages for related information display.
	pageHome	Home page of the website or application.
	pageProfile	User's profile information page.
	pageLogin	Section on the website or in the application for user to login
	searchBar	In site search bar for user to search for related display service
Display Options	heatMap	Heat map for showing intensity or amount of relation data
	graph	Graph type information display
	literalSuggestion	Literal suggestion given out by the analyst showed in word
	leaderBoard	Leaderboard used to show the ranking of different data
	displayItems	Specific items for user to choose for display, such items are set as exercise amount, intensity, cites, etc.
	timeInterval	Time interval for user to choose for information display
Treated Information	exerciseIntensity	People's exercise regularity and intensity calculated by analyst
	tweetData	Data of amount and location of health-related tweets
	sortRanking	Amount, intensity or other attributes of tweets ranking among different set like cities, states, users, exercise types, etc.
	groupsOverlap	Data of the overlap between people who concern about wellness and who exercises and also between people who exercises and who talking about diet
	dataCorrelation	Correlation between different type of health-related tweets
	personalSuggestions	Personal suggestion include exercise intensity and regularity suggestion, personal ranking, etc.
	publicSuggestions	Public suggestion based on the average intensity and regularity of exercise among the whole country
	typeTrend	Trend of amount, intensity, regularity in different exercise type
	rctTweetData	Contents, location, user information of related tweets collected recently
	dataScore	User's data in score system, including exercise score, ranking in the system, award based on

		the score, etc.
	analysisResult	Analysis based on User's historical exercise data, like average exercise intensity and regularity, etc.
	emotionInfo	Data about distribution, amount and type of emotion extracted from tweets.
DB Connector	dbAddress	Address of relation database
	dbAccount	Account to manage the database
	dbKey	Key to manage the database
	databaseStatus	Identity parameter to determine whether the database is open
	numOfTweets	total number of parsed tweets
	dataCommunicate	Retrieve and store data from third part server, due with data communication inside system.
Data Analyst	analyzePublic	Analysis for public information, including statistics data, suggestion inferring, trends calculating, correlation analysis, etc.
	analyzePersonal	Analysis for personal information, including historical record, personal health status deduction, specific suggestions, score calculating, etc.
Third Part Connector	connectTwitter	Connect Twitter by API Auth. to retrieve tweets
	connectGoogle	Connect Google by API Auth. to retrieve map, chart and place finding services

Table 5-3 Attribute definitions

5.1.4 Traceability matrix

Table 5-4 shows how the system use cases map to the domain concepts. It is generated according to the responsibilities of concepts defined above.

Use Case	PW	Domain Concepts: Controller	Profile Storage	Interface Services	Interface Pages	Display Options	Treated Information	DB Connector	Data Analyst	Third Part Connector
UC2	6	√	√	√	√			√		
UC3	6	√	√	√	√			√		
UC4	3							√		
UC5	24							√	√	
UC6	5							√		√
UC8	1	√	√	√	√			√		
UC11	19			√	√	√	√	√		√
UC12	5			√	√		√	√		
UC13	5			√	√	√	√	√		√
UC14	4			√	√	√	√	√		√
UC15	2			√	√	√	√	√		√
UC16	7		√	√	√			√		
UC17	12		√	√	√		√	√		
UC18	5		√	√	√		√	√		
UC19	3	√	√	√	√			√		
UC20	25						√	√	√	
Max PW		6	12	19	19	19	19	25	25	19
Total PW		16	40	75	75	30	52	132	49	35

Table 5-4 Traceability matrix

5.2 System Operation Contracts

Operation	Data collecting & classifying
Preconditions	·Developer get authorized access to Twitter API through OAuth ·databaseStatus = “open” ·numOfTweets = 0, for the initialization of the database
Postconditions	·databaseStatus = “closed” ·All JASON data are parsed into the database of the system ·numOfTweets = total number of parsed tweets

Table 5-5 Data collecting & classifying

Operation	Exercise heat

Preconditions	·Related information all gets analyzed and stored in the database ·Developer get authorized access to Google API
Postconditions	System displays the Google heat map to show the intensity of people who exercise regularly

Table 5-6 Exercise heat

Operation	Facilities search
Preconditions	·System has the authorization to get information of user's location ·Developer get authorized access to Google API ·facilityAvailabiliy = true, for all facilities nearby to be displayed
Postconditions	The location and some other information of the nearby exercise facilities are showed on the Google map

Table 5-7 Facilities search

Operation	Personal suggestions
Preconditions	·isLogged = true, and ·isAuthorized = true, for current users who has logged in to our system and approved the authorization
Postconditions	·isLogged and isAuthorized remain unchanged ·Suggestions are showed on the website in the form of useful links

Table 5-8 Personal suggestions

Operation	Data analyzing
Preconditions	·Related data are stored in the system's database ·System's database is able to be changed
Postconditions	·New tables have been created into the database ·Statistical data are stored in different tables

Table 5-9 Data analyzing

6. Interaction Diagrams

6.1. Interaction Diagrams of Use Cases

6.1.1. Use Case 5: Data Collecting and Classify

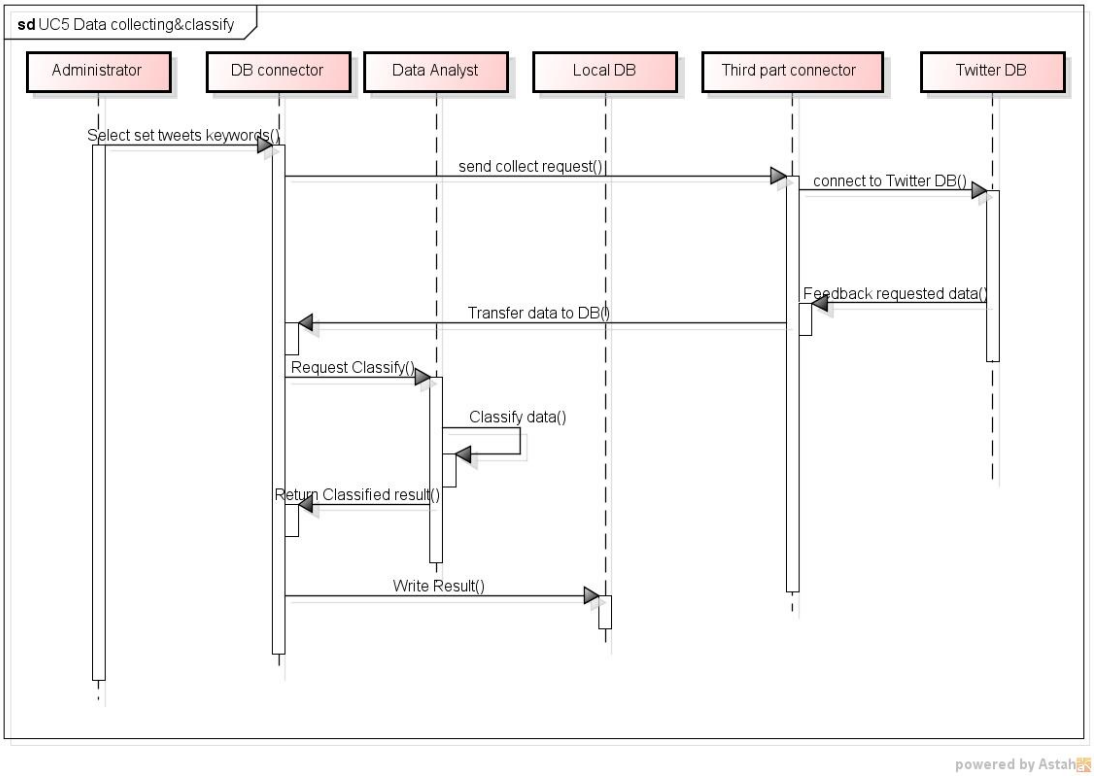


Figure 6-1. Use Case 5 Data: Collecting and Classify

The above interaction diagram is for the Use Case 5 Data Collecting & Classify. Firstly, the administrator connects the DB connector to select set tweets keywords, with using the function of Select set tweets keywords(). Secondly, the system sets up a connection with the DB connector to send collect request to the third party connector. Then the third part connector sends back the data to the DB connector. The DB connector will send data with classify request to the data analyst, and the data analyst will classify the data and send back the result to the DB connector. Finally the DB connector writes the result into the local database.

6.1.2. Use Case 12: Tweet Sentiment

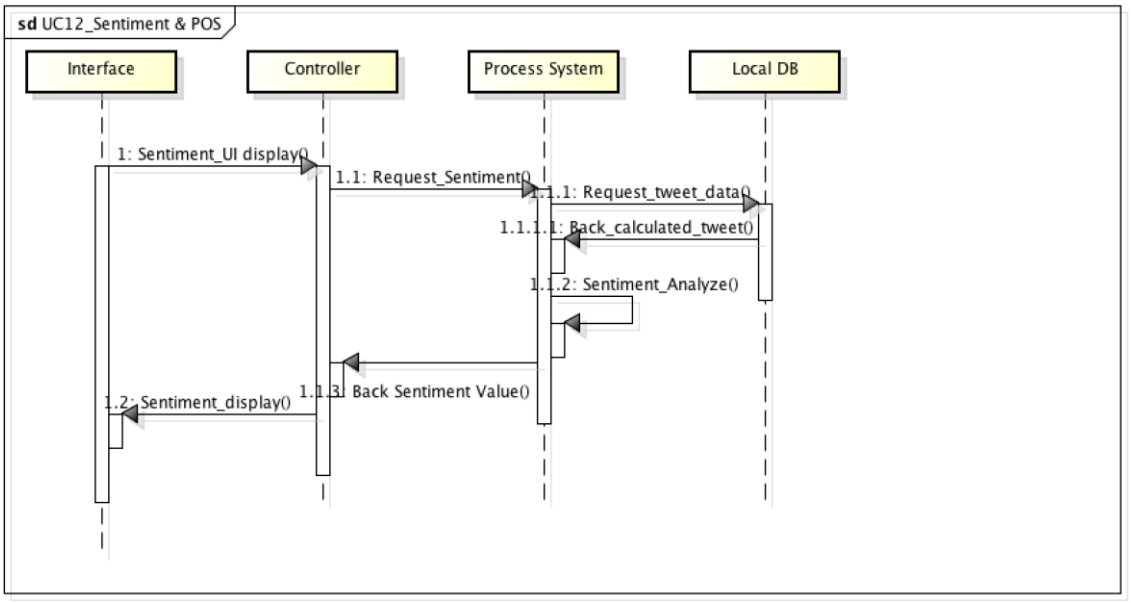


Figure 6-2. Use Case 12 Tweet Sentiment

The UC 12 is used to analyze the sentiment of Twitter users in each State. First the users choose the sentiment analysis feature in our system. Next the database connector will send the request to the database and acquire specific data. Then, the database connector sends the acquired data to the process system, where the data will be analyzed and send back the result to the controller. At the end, the analysis result is shown on the interface.

6.1.3. Use Case 13: Weighted Calories

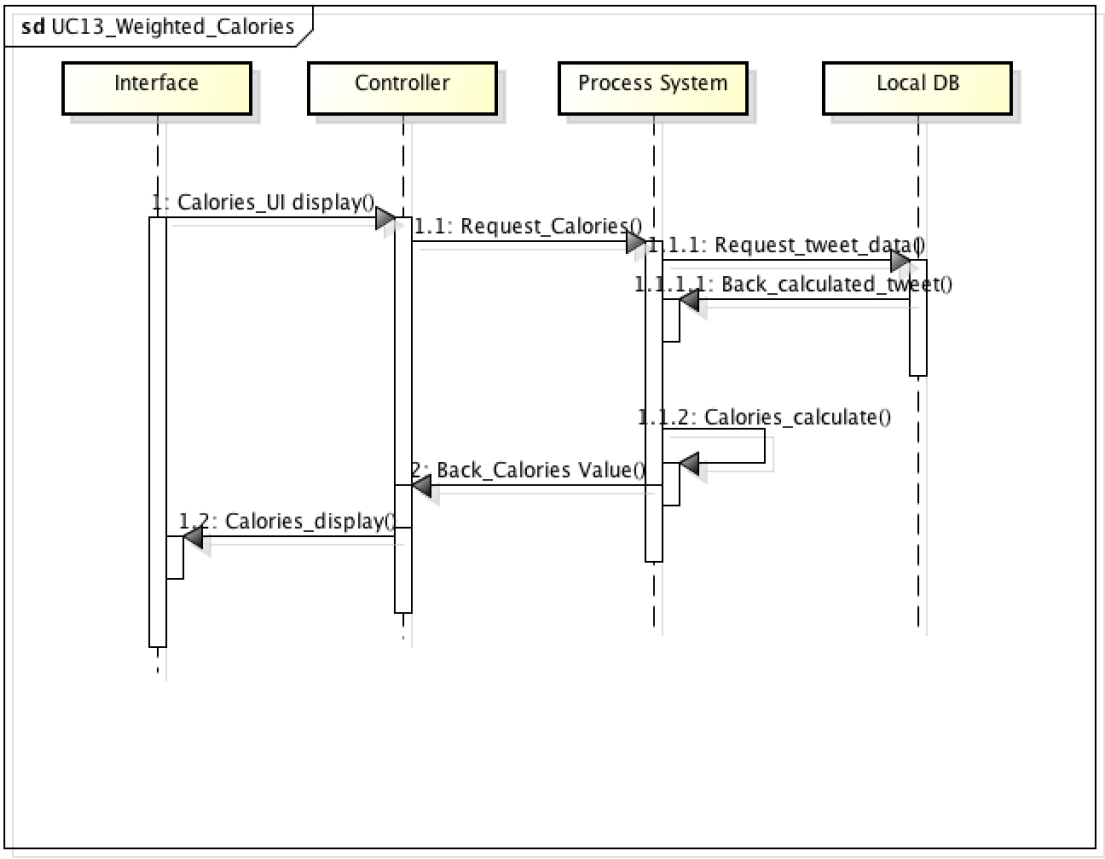


Figure 6-3. Use Case 13 Weighted Calories

The UC 13 is used to analyze the total consumed calories of Twitter users. When the users choose the calories analysis feature in our system. The controller will send the request to the database and acquire specific data. Then, the database connector sends the acquired data to the process system, where the data will be calculated and send back the calculated result to the controller. At last, the analysis result will be shown on the interface for users to see.

6.1.4. Use Case 15: Word Frequency

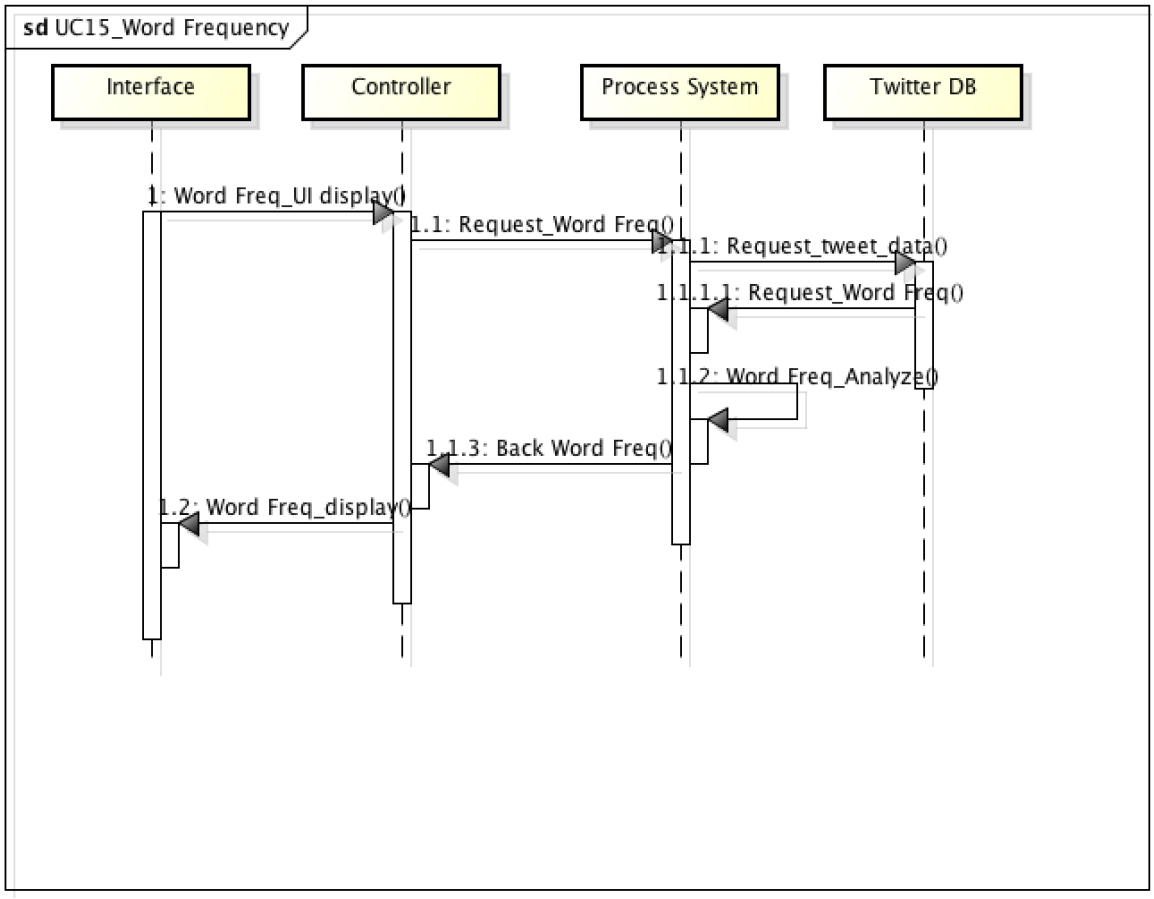


Figure 6-4. Use Case 15 Word Frequency

When users require to display word frequency of one given topic, the controller will request word frequency to the process system which then will request 5000 tweets data from tweet database by using Twitter rest API. After transferring data from Twitter to local system, it processes those tweets through R language which will analyze high frequency of input tweets and output data to interface.

6.1.5. Use Case 17: Personal Suggestions

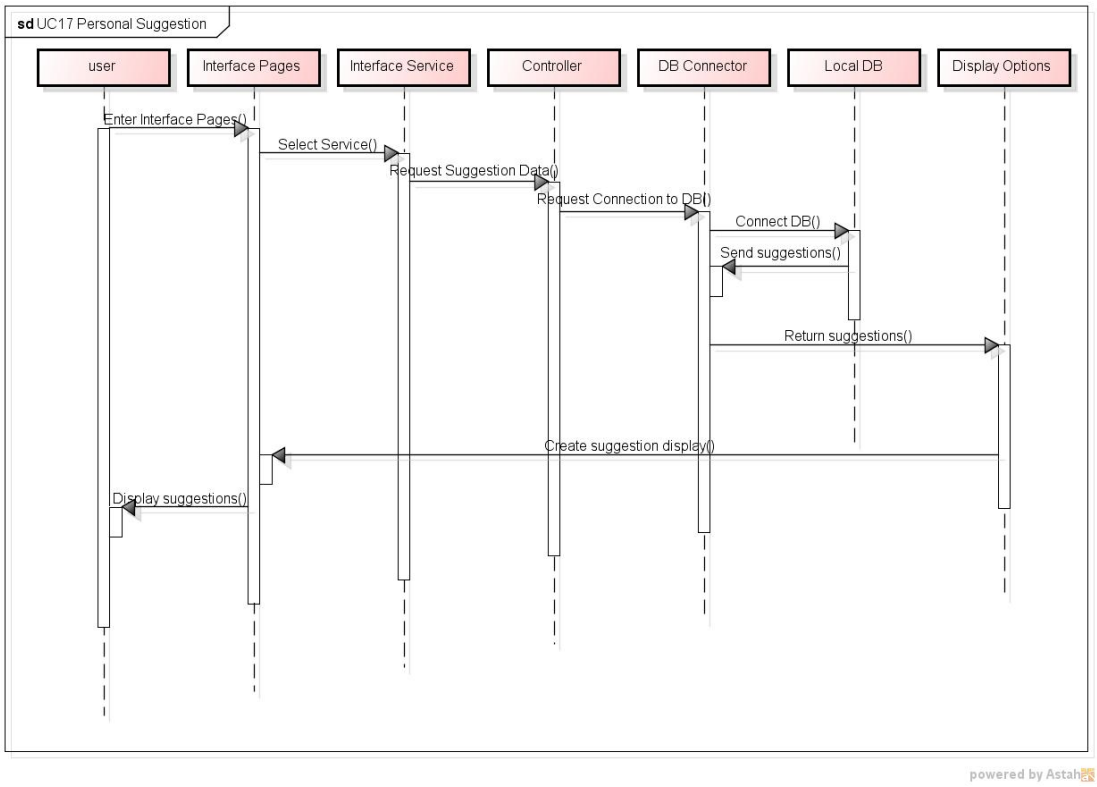


Figure 6-5. Use Case 17: Personal Suggestions

The function of UC17 is basically displaying the personal suggestions for the specific user. First, the login user clicks the 'get personal suggestions' button or enter related page on Interface Pages. After, the interface Pages selects related services existing in Interface Service. Then the Interface Service emits the request to Controller for getting Suggestion Data. Since the controller could not get touch with the database directly, it should send the request to the DB Connector for connection to the local DB. After Local DB returns the related information to the DB Connector. The specific data for user is sent to the Display Options for proper ways to display. Correlate display ways for suggestions are created by Display Options and should be posted to Interface Pages. At last the Interface Pages display the specific suggestions for user on the screen.

6.1.6. Use Case 20: Data Analysis

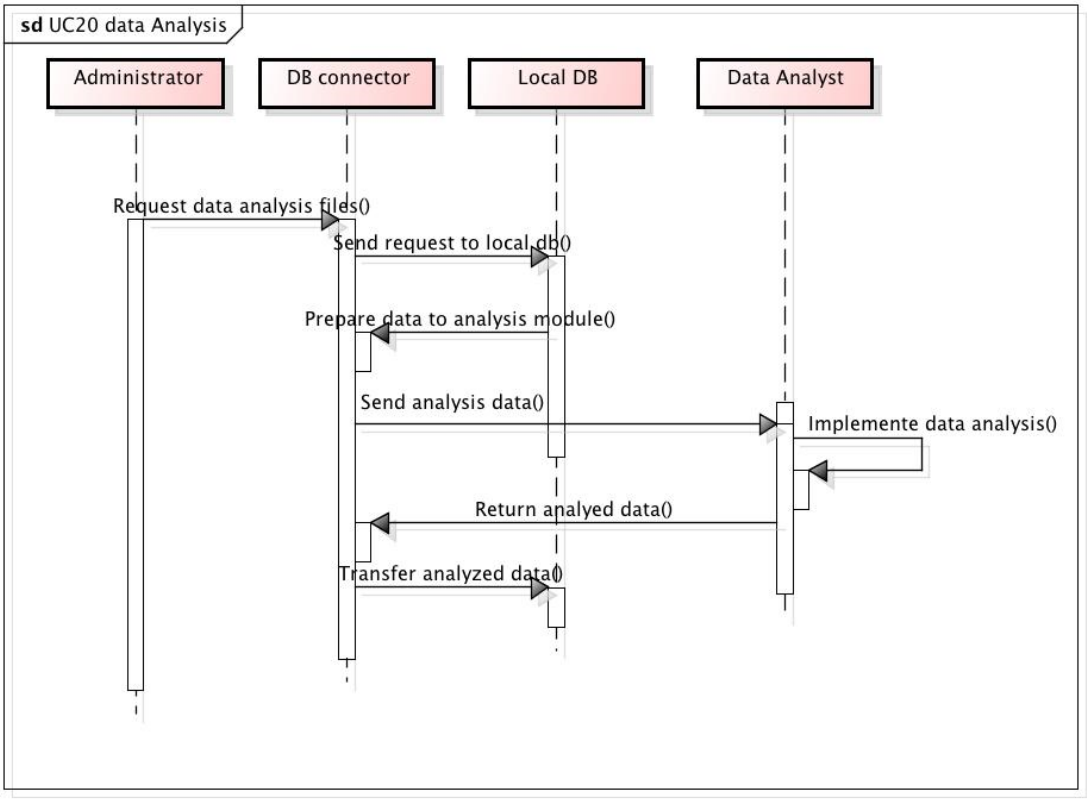


Figure 6-6. Use Case 20 Data analysis

The UC 20 is used to analyze data that collected from Twitter. First the Administrator calls the database connector for specific request. Next the database connector will send the request to the database and acquire specific data. Then, the database connector sends the acquired data to the data analyst, where the data will be analyzed and send back the result to the database connector. At the end, the database connector will write the result to the database.

6.2. Design Patterns

Our system design mainly follows model-view-controller (MVC) design pattern. The communications between our server and website, our server and IOS, our server and Android all follow this pattern. The definition of MVC is as follows from Wiki:

"The central component of MVC, the *model*, captures the behavior of the application in terms of its problem domain, independent of the user interface. The model directly manages the data, logic and rules of the application. A *view* can be any output representation of information, such as a chart or a diagram; multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. The third part, the *controller*, accepts input and converts it to commands for the model or view."

The control flow of MVC can be described as Figure 6-4.

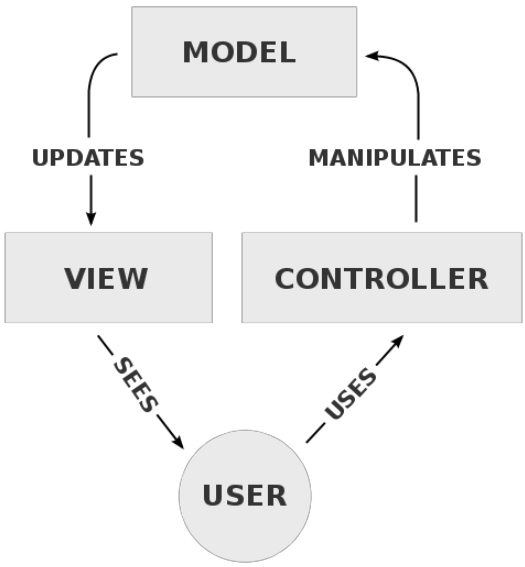


Figure 6-4. MVC design pattern

How we use MVC in our system? The HTML website, IOS and Android user interface are the view part in MVC design pattern. The database management is the model part. We have a communication controller called JSON sender in our system performed as the controller in MVC. The user will interact with the website, IOS, android user interface to trigger the events, and send HTTP POST (IOS) and GET (website, Android) request to the controller. The controller will manipulate database by SQL based on the request with feature id to response the feature data from feature tables. Then the view part will visualize the returned feature data, and reply the user.

7. Class Diagram and Interface Specification

7.1. Class Diagram

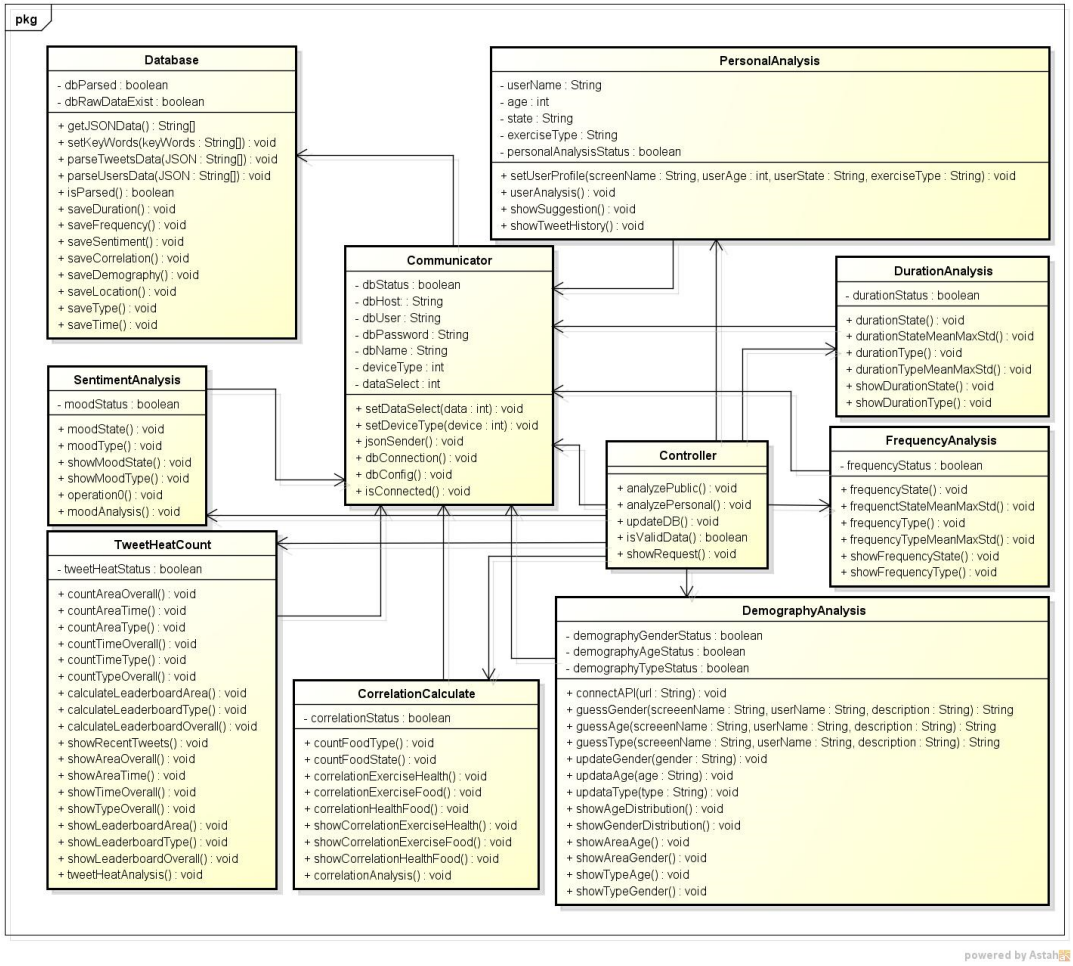


Figure 7-1. Class diagram

Above is the class diagram of the health monitoring system. It is constructed by total ten classes. Firstly, the methods within the database class would be called early because the system needed to retrieved tweets data in the first time. The database class mainly takes the responsibility of retrieving the data from twitter, parse the data in JSON format, manage the raw data and store the data after analysis.

Communicator class focuses on the communication between the showing function called by the user interface and the database. It could identify what device is requesting the data and what data is being requesting by the variables deviceType and

dataSelect.

Controller deals with the issues occur in user's action or the system itself. In general saying, it is used to automatically run the analysis functions derived from all the classes related to analysis. Besides, it would handle the showing request sending by the users.

PersonalAnaylsis class is separated from other public analysis classes here. It is different from other analysis since user is asked to input several value like their screen name in twitter, their actual age and favorite exercise type, etc. The methods within this class would do the analysis about personal information and show the corresponding information such as the suggestion and user's health related tweet history.

Other classes are all about public analysis including the DurationAnalysis, FrequencyAnalysis, etc. They are some specific classes that would do the corresponding analysis about public health data and provide respectively functions to show the useful information we derived from the data after statistics and analysis. Following sections would give more detail about those classes.

7.2. Data Types and Operation Signatures

7.2.1. Database

The "Database" class contains the following five variables that stores the basic information of the database used for the connection.

dbParsed Boolean variable storing the parse status of the tweets data.

dbRawDataExist Boolean variable storing the Exist status of the data.

The functions of this class listed below include several get and parse functions that are used to obtain data from Twitter, and several save functions that are used for storing analytical data into database.

getJSONData() This function is used for obtaining raw tweets data from Twitter Streaming API and returns an array of JSON value, each element represents one piece of tweet data.

setKeyWords(keyWords : String[]) This function is used for setting the keywords that are needed for filtering the tweets. All keywords are related to exercise and

health.

parseTweetsData(json : String[]) This function is used for parsing the collected JSON data containing tweet information into readable tweet data and store them in the tweets table of the database.

parseUsersData(json : String[]) This function is used for parsing the collected JSON data containing user information into readable user data and storing them in the users table of our database.

isParse() This function is used for recording the parse result of each piece of JSON data. If it has been parsed, return true. Else, return false.

saveDuration() This function is used for saving the analytical data related to the exercise duration time into new feature tables of the database.

saveFrequency() This function is used for saving the analytical data related to the keyword frequencies into new feature tables of the database.

saveSentiment() This function is used for saving the analytical data related to the sentiment computing into new feature tables of the database.

saveCorrelation() This function is used for saving the analytical data related to the correlation computing into new feature tables of the database.

saveDemography() This function is used for saving the analytical data related to the demography information into new feature tables of the database.

saveLocation() This function is used for saving the analytical data related to different locations into new feature tables of the database.

saveType() This function is used for saving the analytical data related to the exercise types into new feature tables of the database.

saveTime() This function is used for saving the analytical data related to different time of the day into new feature tables of the database.

7.2.2. TweetHeat

The “TweetHeat” class contains several count functions that are used to count the number of tweets concerning different categories, several calculate functions to sort the count result to make the leaderboard, and also some show functions to display all

the responding charts onto the user interface.

tweetHeatStatus Boolean variable storing the status of the tweetsHeatAnalysis.

tweetHeatAnalysis() This function is used for calling all the analyzing methods in this class.

countAreaOverall() This function is used for counting the number of tweets in different areas.

countAreaTime() This function is used for counting the number of occurrences of different areas in different time periods.

countAreaType() This function is used for counting the number of occurrences of different exercise types in different areas.

countTimeOverall() This function is used for counting the number of occurrences of different time periods.

countTimeType() This function is used for counting the number of occurrences of different time periods corresponding to different exercise types.

countTypeOverall() This function is used for counting the number of occurrences of different exercise types.

calculateLeaderboardArea() This function is used for sorting the count result of different areas and storing them in decreasing order.

calculateLeaderboardType() This function is used for sorting the count result of different exercise types and storing them in decreasing order.

calculateLeaderboardOverall() This function is used for sorting the count result of all tweets concerning exercise and health tweeted by each user and storing them in decreasing order.

showRecentTweets() This function is used for showing on map the most recently posted tweet and its user.

showAreaOverall() This function is used for displaying the analytical chart showing the number of tweets in different states.

showAreaTime() This function is used for displaying the analytical chart showing the number of tweets in different states in different time periods.

showTimeOverall() This function is used for displaying the analytical chart showing the number of tweets in different time periods.

showTypeOverall() This function is used for displaying the analytical chart showing the number of tweets concerning different exercise types.

showLeaderboardArea() This function is used for displaying the leader board ranked by the number of tweets concerning different areas.

showLeaderboardType() This function is used for displaying the leader board ranked by the number of tweets concerning different exercise types.

showLeaderboardOverall() This function is used for displaying the leader board ranked by the number of tweets concerning exercise and health.

7.2.3. FrequencyAnalysis

The "FrequencyAnalysis" class contains several functions that are used to calculate the frequency of exercise corresponding to different states and types and there are also some methods that could calculate the mean, max and standard deviation of such frequencies. Besides, there are some show functions to display all the responding charts onto the user interface.

frequencyStatus Boolean variable storing the status of the frequency analysis.

frequencyState() This function is used for calculating the frequency of exercise in different states according to user's tweets.

frequencyStateMeanMaxStd() This function is used for counting the mean, maximum and standard deviation of those frequency calculated in fregquencyState().

frequencyType() This function is used for calculating the frequency of exercise in different exercise types according to user's tweets.

frequencyTypeMeanMaxStd() This function is used for counting the mean, maximum and standard deviation of those frequency calculated in fregquencyType().

showFrequencyState() This function is used for displaying the analytical chart that would show the frequency values of exercise corresponding to different states.

showFrequencyType() This function is used for displaying the analytical chart that would show the frequency values of exercise corresponding to different exercise

types.

7.2.4. DurationAnalysis

The "DurationAnalysis" class contains several functions that are used to calculate the duration of exercise corresponding to different states and types and there are also some methods that could calculate the mean, max and standard deviation of such durations. Besides, there are some show functions to display all the responding charts onto the user interface.

durationStatus Boolean variable storing the status of the duration analysis.

durationState() This function is used for calculating the Duration of exercise in different states according to user's tweets.

durationStateMeanMaxStd() This function is used for counting the mean, maximum and standard deviation of those duration calculated in durationState().

durationType() This function is used for calculating the duration of exercise in different exercise types according to user's tweets.

durationTypeMeanMaxStd() This function is used for counting the mean, maximum and standard deviation of those duration calculated in durationType().

showDurationState() This function is used for displaying the analytical chart that would show the duration values of exercise corresponding to different states.

showDurationType() This function is used for displaying the analytical chart that would show the duration values of exercise corresponding to different exercise types.

7.2.5. SentimentAnalysis

The "SentimentAnalysis" class contains several functions that are used to calculate the sentiment value of the tweet text corresponding to different states and types. Besides, there are some show functions to display all the related charts on the user interface.

moodStatus Boolean variable storing the status of the mood analysis.

moodAnalysis() This function is used for calling all the analyzing methods in this class.

moodState() This function is used for calculating the mood value of tweets in different states according to user's tweet text.

moodType() This function is used for calculating the mood value of tweets for different exercise types according to user's tweet text.

showMoodState() This function is used for displaying the state map that would show the happiness degree corresponding to different states.

showMoodType() This function is used for displaying the analytical chart that would show the mood values corresponding to different exercise types.

7.2.6. CorrelationCalculate

The "CorrelationCalculate" class contains several functions that are used to count the number of the tweets that metioned food corresponding to different states and exercise types. There are also several functions that would calculate the linear regression value among the tweets count of health, exercise and food. Besides, there are some show functions to display all the related charts on the user interface.

correlationStatus Boolean variable storing the status of the correlation analysis.

correlationAnalysis() This function is used for calling all the analyzing methods in this class.

countFoodType() This function is used for counting the number of tweets related to different kinds of food according to multiple types of exercise.

countFoodState() This function is used for counting the number of tweets related to different kinds of food according to different states.

correlationExerciseHealth() This function is used for calculating the linear regression value between the counts of tweets related to exercise and health.

correlationExerciseFood () This function is used for calculating the linear regression value between the counts of tweets related to exercise and food.

correlationHealthFood () This function is used for calculating the linear regression value between the counts of tweets related to health and food.

showCorrelationExerciseHealth() This function is used for showing two line charts that would indicate the count number of tweets related to exercise and health

separately.

showCorrelationExerciseFood() This function is used for showing two line charts that would indicate the count number of tweets related to exercise and food separately.

showCorrelationFoodHealth() This function is used for showing two line charts that would indicate the count number of tweets related to food and health separately.

7.2.7. PersonalAnalysis

The "PersonalAnalysis" class contains the following four variables that store the user's personal information.

personalAnalysisStatus Boolean variable storing the status of the personal analysis.

username String variable storing the user's name.

age Integer variable storing the user's age.

state String variable storing the user's current location.

exerciseType String variable storing the user's favorite exercise type.

This class include functions that are used to give out the personal suggestions based on the information the users provide.

setUserProfile(screenName : String, userAge : int, userState : String, exerciseType : String) This function is used for setting the basic information that is needed for the analysis.

userAnalysis() This function is used for making analysis on the proper healthy food and exercise duration time based on the information the users provide.

showSuggestion() This function is used for displaying the proper personal suggestions based on the analysis.

showTweetHistory() This function is used for displaying the past tweets the user posted and showing the time when the tweet has been created.

7.2.8. DemographyAnalysis

The "DemographyAnalysis" class has some guess functions that use third party's API to make guesses on the demographical information of the Twitter users based on their description, some update functions that update the guess results into the existing table, and some show functions that display the analytical results.

connectAPI(url : String) This function is used for connecting our database to the third party's API. The parameter "url" is the website of this API.

guessGender(screenName : String, username : String, description : String) This function is used for making a guess on the gender of a certain user based on his/her screen name, user name and the user description.

guessAge(screenName : String, username : String, description : String) This function is used for making a guess on the age of a certain user based on his/her screen name, user name and the user description.

guessType(screenName : String, username : String, description : String) This function is used for making a guess on the user type, organization or person, of a certain user based on his/her screen name, user name and the user description.

updateGender(gender : String) This function is used for updating the guess result of the user gender in the existing users table.

updateAge(gender : String) This function is used for updating the guess result of the user age in the existing users table.

updateType(gender : String) This function is used for updating the guess result of the user type in the existing users table.

showAgeDistribution() This function is used for displaying the age distribution among all the users who have posted exercise-related tweets.

showGenderDistribution() This function is used for displaying the gender distribution among all the users who have posted exercise-related tweets.

showAreaAge() This function is used for displaying the age distribution among users who have posted exercise-related tweets in different states.

showAreaGender() This function is used for displaying the gender distribution among users who have posted exercise-related tweets in different states.

showTypeAge() This function is used for displaying the distribution of different exercise types in different age groups.

showTypeGender() This function is used for displaying the distribution of different exercise types in different gender groups.

7.2.9. Controller

The "Controller" class works as an essential part in the system-to-be. All the executions of its functions are linked to some other classes. The user activates these functions and gets the analytical results corresponding to his/her choice.

analyzePublic() This function is used for making analysis based on public information we obtained from the Twitter.

analyzePersonal() This function is used for making analysis based on personal information we obtained form the user.

updateDB() This function is used for making updates on the existing database if some changes need to be made.

isValidData() This function is used for identifying whether a certain piece of data is valid. If valid, return true, else, return false.

showRequest() This function is used for displaying corresponding analytical charts or tables based on the request from the user.

7.2.10. Communicator

The "Communicator" class is linked to all the other classes that need to get data from the database. It has two variables storing the type of the platform and which feature table we need to use.

dbHost String variable storing the host of the database.

dbUser String variable storing the username of the database.

dbPassword String variable storing the password corresponding to the username.

dbName String variable storing the database name.

dbStatus Boolean variable identifying the status of the database, open or closed.

deviceType Integer variable storing the type of the platform the JSON dada need to be sent to. We choose 0 representing the web platform, and 1 representing the IOS platform.

dataSelect Integer variable storing which feature tables we need to use for analysis. Different numbers represent different feature tables.

This class contains a jsonSender() function which is essential to send the database data to other platforms in JSON formation. Other functions are used for connecting the database.

setDataSelect(data : int) This function is used for setting which feature table we need to use for our analysis.

setDeviceType(device : int) This function is used for setting the type of platform we use to display the analytical results.

jsonSender() This function is used for packaging the data we need in certain tables and sending them to the platform we choose.

dbConnection() This function is used for making connection with the database.

dbConfig() This function is used for initializing the database.

isConnected() This function is used for testing the connection to the database. If connected, return true. Otherwise, return false.

7.3. Traceability Matrix

Domain Concepts	Software Classes									
	Controller	Communicator	Database	PersonalAnalysis	DemographyAnalysis	DurationAnalysis	FrequencyAnalysis	TweetHeatCount	CorrelationCalculate	SentimentAnalysis
Controller	X									
Profile Storage				X	X					
Interface Services				X	X	X	X	X	X	X
Interface Pages				X	X	X	X	X	X	X
Display Options				X	X	X	X	X	X	X
Treated Information			X							
DB Connector		X	X							
Data Analyst		X	X	X	X	X	X	X	X	X
Third Part Connector			X	X	X	X	X	X	X	X

Table 7-1. Traceability matrix

Apparently, there are many classes evolved from the same Domain Concept except the Controller, which is basically achieved by the single class that also named Controller.

Concept Profile Storage include user's personal data (like demography information). These information could derived from Classes PersonalAnalysis and DemographyAnalysis since both of the classes contain related methods to infer and output such message.

Interface Services, Interface Pages and Display Options construct the user

interface part of the system, hence the last seven classes derived from them all.

Because the treated information is basically stored in the database of our system, there are not many classes involved with this concept. Only the class Database is needed to get such information.

Both of the two classes --- Communicator and Database --- take the responsibility of Concept DB Connector. Due to some specific reasons, we separate them by different functions. Database takes charges of the data management within the database. Communicator transfers the data between other classes and the database.

Data Analyst is separated into seven classes that related to specific types' analysis and two classes that enable them to do communication with database.

At last, because it is hard to isolate the third part connector in codes, it is involved in many of the classes that would show the information in the user interface and the class that retrieves data from twitter.

7.4. Object Constraint Language (OCL) Contracts

Important contracts for classes and their operations:

For Database class:

context: Database::parseTweetsData(JSON : String[]) : void

pre: dbRawDataExist = true

post: dbParsed = true

PersonalAnalysis::personalAnalysisStatus = false

DurationAnalysis::durationStatus = false

FrequencyAnalysis::frequencyStatus = false

DemographyAnalysis::demographyGenderStatus = false

DemographyAnalysis::demographyTypeStatus = false

DemographyAnalysis::demographyAgeStatus = false

CorrelationCalculate::correlationStatus = false

TweetHeatCount::tweetHeatStatus = false

SentimentAnalysis::moodStatus = false

context: Database:: getJSONData() : String[]

pre: none

post: dbRawDataExist = true

For Communicator class:

context: Communicator:: jsonSender() : void

pre: dbStatus = true

post: none

context: Communicator:: setDataSelect(data : int) : void

pre: dbStatus = true

post: dataSelect = data

context: Communicator:: setDeviceType(device : int) : void

pre: dbStatus = true

post: deviceSelect = device

context: Communicator:: dbConnection() : void

pre: dbStatus = false

post: dbStatus = true

For PersonalAnalysis class:

context: PersonalAnalysis::setUserProfile(screenName : String, userAge : int, userState : String, exerciseType : String) : void

pre: Communicator::dbStatus = true

post: username = screenname

age = userAge

state = userState

exerciseType = exerciseType

context: PersonalAnalysis::userAnalysis() : void

pre: Communicator::dbStatus = true

post: personalAnalysisStatus = true

For DurationAnalysis class:

context: DurationAnalysis::durationState() : void

pre: Communicator::dbStatus = true

post: durationStatus = true

For FrequencyAnalysis class:

context: FrequencyAnalysis::frequencyState() : void

pre: Communicator::dbStatus = true

post: frequencyStatus = true

For DemographyAnalysis class:

context: DemographyAnalysis::updateGender(gender : String) : void

pre: Communicator::dbStatus = true

post: demographyGenderStatus = true

context: DemographyAnalysis::updateType(type : String) : void

pre: Communicator::dbStatus = true

post: demographyTypeStatus = true

context: DemographyAnalysis::updateAge(age : String) : void

pre: Communicator::dbStatus = true

post: demographyAgeStatus = true

For CorrelationCalculate class:

context: CorrelationCalculate::correlationAnalysis() : void

pre: Communicator::dbStatus = true

post: correlationStatus = true

For TweetHeatCount class:

context: TweetHeatCount::tweetHeatAnalysis() : void

pre: Communicator::dbStatus = true

post: tweetHeatStatus = true

For SentimentAnalysis class:

context: SentimentAnalysis::moodAnalysis() : void

pre: Communicator::dbStatus = true

post: moodStatus = true

8. System Architecture and System Design

8.1. Architectural Style

"Architectural styles are reusable packages of design decisions and constraints that are applied to an architecture to induce chosen desirable qualities". Our system scrawls and stores data from Twitter database, analyzes the data for different business goals (features), finally presents the clear results to users. The users should be able to access our system to find what they want, but the reverse is not allowed. Based on these system characteristics, our system would be better to take the advantage of client-server architectural style instead of peer-to-peer. The client-server architecture is shown as the Figure 8-1.

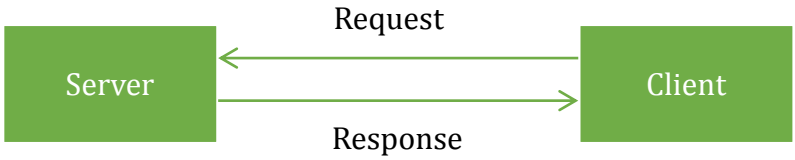


Figure 8-1. Client-Server sketch

Besides, because of the three steps of our system talked above, our system is suit for three-tier client-server architecture – data tier, logic tier and presentation tier. The sketch for three-tier architecture is shown as the Figure 8-2.

Figure 8-2. Three-Tier architecture sketch

8.2. Identifying Subsystems

As shown in the Figure 8-3, there are six main subsystems in our system –

collection, store, analytics, transmission, presentation and plot. There are also dependencies among them shown as the arrows in the Figure 8-3. For example, the package presentation points to the package plot that means the presentation uses some elements in the plot.

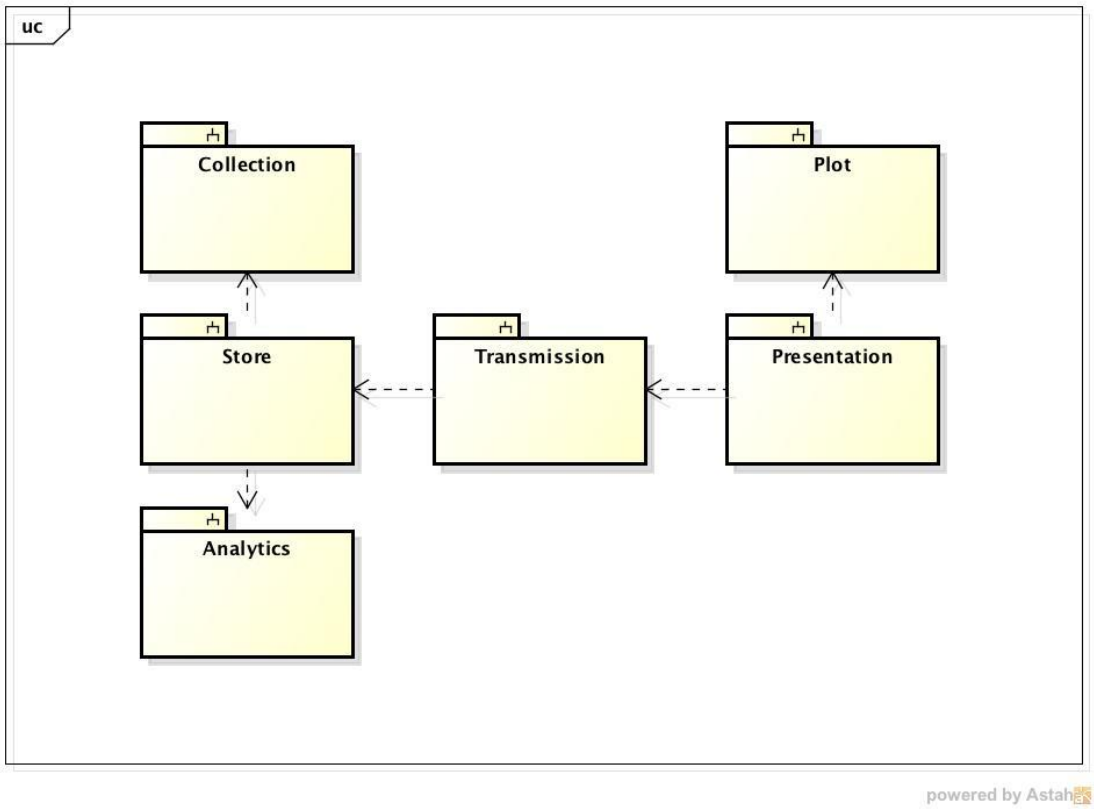


Figure 8-3. Subsystems sketch

The collection subsystem is responsible for scrawling data from Twitter database to our database. The data will be in a JSON format and stored directly in the json_cache table in our database. The store subsystem will parse (extract) the properties such as screen name, date, tweet from JSON data in the json_cache table and insert the properties into other tables. The analytics subsystem will compute results from the data stored in the tables mentioned above in order to fulfill the different business goals, and insert the results in feature tables. The presentation subsystem simply structures the user interface and listens to the events triggered by users. The subsystem will call the functions in the plot subsystem to draw plots like bar chart, pie chart, line chart and maps. Finally, the transmission subsystem is the bridge between the front-end (client) and the rear-end (server). We still use JSON format for transmission. The presentation will not get all the feature data at the initialization stage since data is large and loading is slow. Instead, the presentation will tell the transmission to send which feature table's records by sending an offset. Then the transmission will get the feature table's records from store and send back to the presentation in JSON format.

8.3. Mapping Subsystems to Hardware

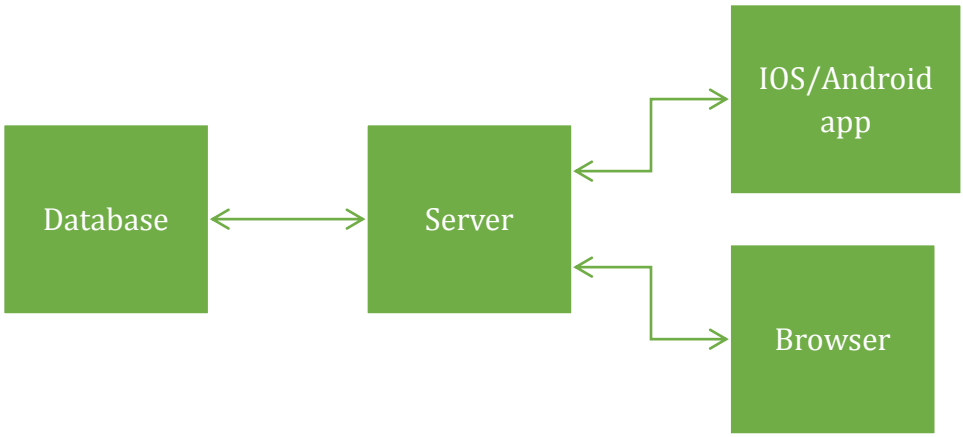


Figure 8-4. Hardware map

In the Figure 8-4, database and server could be distributed into two computers or into one computer. IOS app could be operated on iPhone or iPad, while browser could be operated on any platform as long as it supports the browser, e.g., computer, smart phone.

What is the relationship between the subsystems and the hardware? The collection, store, analytics and transmission subsystems will be deployed on the server. The computer for database only stores the data, and will be accessed by the server. If it is a web application, the presentation and plot subsystems will be also deployed on the computer for server. The user interface structure and the data for plot will be downloaded by the browser. If it is an IOS application, the presentation and plot subsystems will be deployed on the IOS device. It means only the data for plot will be transmit from the server to the IOS device.

8.4. Persistent Data Storage

Our system uses the rational database MySQL to store the data. With the database, the application will maintain the data for the next running. The tables storing Twitter JSON data and the data after being parsed are shown as the Figure 8-5.

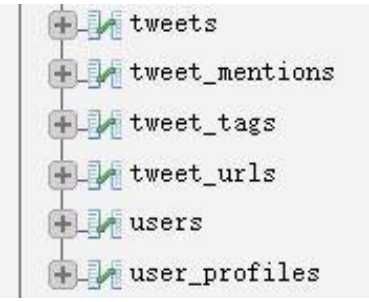


Figure 8-5. Twitter tables

The feature tables are shown as the Figure 8-6.

Figure 8-6. Feature tables

8.5. Network Protocol

In our system, the communication between our server and the Twitter server is based on the GET method in HTTP protocol. Thus, our server should be linked to the Internet. The communication between our server and the browser is also based on the GET method in HTTP protocol because of the jQuery. But the communication between our server and the IOS device is based on the POST method in HTTP protocol. POST is much securer than GET since the data will be appended to the address when using GET. For these two communications, the server, the browser and the IOS device are in the same local area network (LAN) because currently our server is not accessible by the Internet – no domain name. The network is shown in the Figure 8-7.

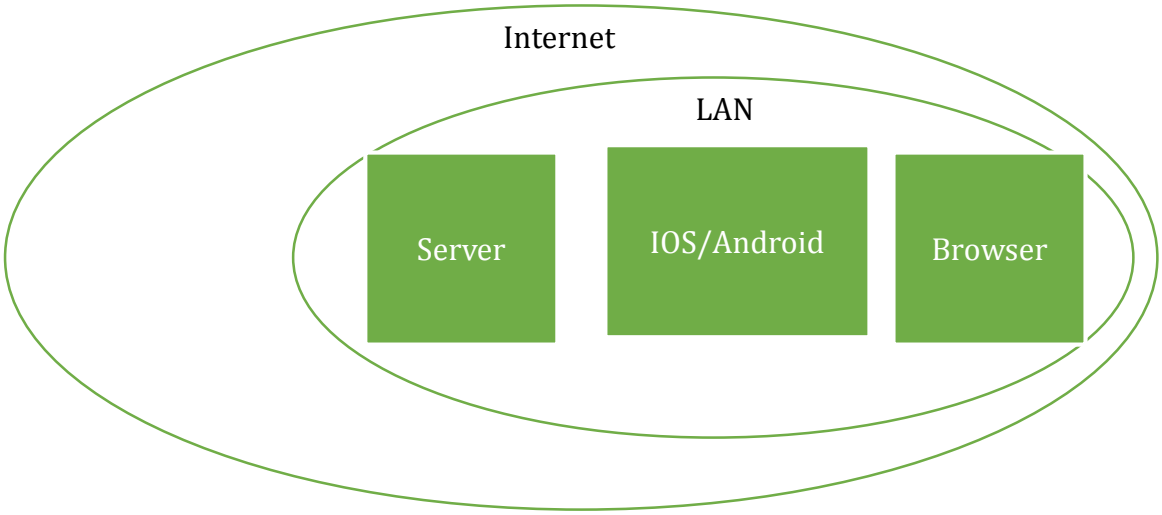


Figure 8-7. Network

8.6. Global Control Flow & Hardware Requirements

Our system is an event-driven system. Users will interact with the user interface elements, and trigger the events and get the responses. Our system uses a timer to reload the data from feature tables to make our system real-time.

Now our database has 700+ thousand tweets, and it needs 5 GB to store the data. At least, it needs a computer to be the database, the server and the browser and an IOS/Android device to run the app.

9. Algorithm and Data Structure

9.1. Algorithms Analysis

9.1.1. Improve data reliability using weight index

When we set key words and collect data, there is unavoidable noise in the result, some tweet text may reflect our searching intention, some may have no relationship with the health topic. For example, if we use run as key words, the result tweet text may contain runny or rune. Thus when we do data analyze, we need to eliminate irrelevant tweet text and keep useful information. The methods we use is putting weight index to each kind of sport. And the methods will implement in emotion analyze and heat map analyze feature.

1. Weight index math model description

The basic weight index equation shows below:

$$\textit{key} \text{ word weight index} = \frac{\textit{each} \text{ field veritable tweet text}}{\text{each field total tweet text}} \qquad 9\text{-}1$$

If weight index is greater than 80%, we define the keyword as reliable, otherwise we define the keyword as unreliable.

However, because the database is too large to test them all, so we use systematic sample method to get limited tweet text for each key words. The above equation will change to

$$\textit{key} \text{ word weight index} = \frac{\textit{each} \text{ field } \textit{sample} \text{ veritable tweet text}}{\text{each field } \textit{sample} \text{ total tweet text}} \qquad 9\text{-}2$$

2. Implementation

a) We implements this method in emotion analyze. First we set a set of key words which can represent a kind of emotion, then we use equation 9-2 to analyze each key word's reliability, if it is unreliable, we eliminate or replace it with another key word.

b) We implements this method in heat map. For each kind of exercise, we use its key word weight index multiple area's total tweet text, and use the result as total number of people in a area who actually doing exercise.

9.1.2. Personal suggestion based on term frequency–inverse document frequency (tf-idf)

Definition :Personal suggestion here means find key words that most related to a certain topic. For example, if a topic is health, then we should find the key words that can represent this topic(such as fitness). By doing so, we need to collect users' Tweet text and extract the words that can represent certain topic. To fulfill this function, we use the tf-idf methods. The steps describe below:

a) We filter the tweet text and extract human readable words by using the R language. R is a free software programming language and software environment for statistical computing and graphics. With the help of twitteR, we can easily obtain the preliminary information.

b) Then we use tf-idf methods to obtain key words that can represent a certain topic. tf–idf, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. The basic function is [13]:

Term Frequency and Inverse Document Frequency can be calculated as

$$tf_{i,j} = \frac{\left|n_{i,j}\right|}{\left|\sum_k n_{k,j}\right|}, \quad idf_i = \log \frac{|D|}{\left|\{j : t_i \in d_j\}\right|} \qquad 9$$

Finally it-idf is calculated as

$$tf_{i,j} \times idf_i \qquad 9\text{-}4$$

For example, if we choose health, diet, sleep and exercise as four topic and want to see their it-idf importance value, it can be shown as below:

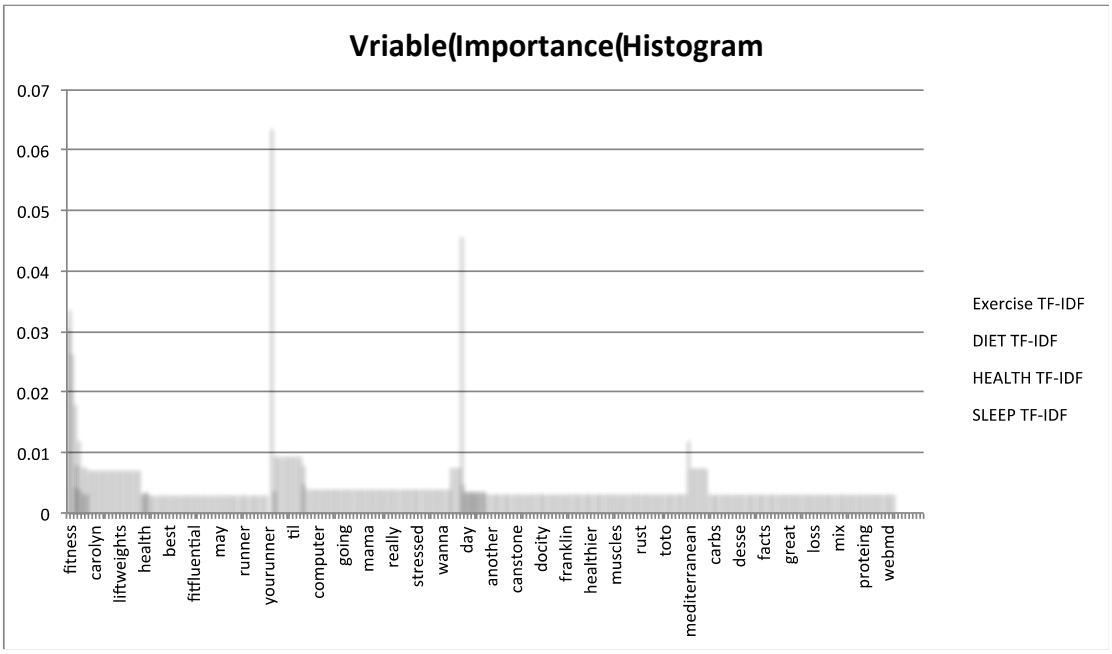


Figure 9-1. Variable Importance Histogram

9.1.3. Sports correlation analyze based on linear regression

To analyze the relationship between two sports, we use linear regression methods to draw a line to show their correlation degree.

1. Linear regression math model description.

To implement linear regression analysis, we use generalized least squares methods (GLS) [14]. The roughly description shows below:

According to wiki [15], The GLS is applied when the variances of the observations are unequal, or when there is a certain degree of correlation between the observations. In our project, we assume the linear is:

$$\varphi(x) = a + bx \tag{9-5}$$

According to the GLS theory, the solution equation will be:

$$\begin{bmatrix} \sum_{i=1}^{m} 1 & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x^2{}_i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \end{bmatrix} \tag{9-6}$$

Within 9-6, m is the total number of point we use. From the solution equation, we can get a and b, then we put them back into 9-5 and get the linear equation.

To improve the accuracy, we introduce the Pearson product-moment correlation coefficient. According to the wiki [15], Pearson product-moment correlation coefficient is a measure of the linear correlation (dependence) between two variables X and Y. The mathematical equation is

$$r_{xy} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{m}(y_i - \bar{y})^2}} \qquad 9\text{-}7$$

The r_{xy} range from -1 to 1, we define $0.8 \quad |r_{xy}| \quad 1$ as highly correlated, $|r_{xy}| \quad 0.3$ as no correlated.

2. Implementation

a) First we divide one day in to 24 hours and choose two sports that we want to compare, then we count how many related tweet text for each sport is showed in each hour. An example is showed as Figure 9-2.

b) Using equation 9-7 to analyze the data correlation degree, if they are highly correlated we use the generalized least squares methods to draw the linear equation in the coordinate and use the slope to judge if they are positive correlated or negative correlated.

	swimming	running
1h	500	1000
2h	250	600
3h	1500	2800
…	…	…

Figure 9-2. Example

9.1.4. Sentiment analysis based on probability density function of a normal distribution

This method is derived from the project website of Visualizing Twitter Sentiment [16].

For sentiment analysis, the ANEW [17] dictionary provides measures of valence, arousal, and dominance for 1,034 English words. Each word is rated on a nine-point scale ranging from 1 to 9.

Ratings for a common word are combined into a mean rating and a standard deviation of the ratings for each dimension.

For example, for the word house, ANEW reports:

house,

$$v = (\mu:\ 7.26,\ \sigma: 1.72),\ a = (\mu:\ 4.56,\ \sigma:\ 2.41),\ d = (\mu:\ 6.08,\ \sigma:\ 2.12),\ f_q = 591 \quad 9\text{-}8$$

This shows that house has a mean valence v of 7.26 and a standard deviation of 1.72, a mean arousal a of 4.56 and a standard deviation of 2.41, a mean dominance d of 6.08 and a standard deviation of 2.12, and a frequency f_q of 591 ratings.

However if multiple words documented in ANEW dictionary for instance:

Congrats to @HCP_Nevada on their health care headliner win!

ANEW's measure of the n = 2 words' means and standard deviations of valence and arousal are:

health,

$$v = (\mu: 6.81,\ \sigma: 1.88),\ a = (\mu: 5.13,\ \sigma: 2.35),\ f_q = 105 \quad 9\text{-}9$$

win,

$$v = (\mu: 8.38, \sigma: 0.92),\ a = (\mu: 7.72, \sigma: 2.16],\ f_q = 55 \quad 9\text{-}10$$

To combine the means for health and win, we assume that the individual ratings reported for each word form a normal distribution. Intuitively, if a word has a higher standard deviation, for example, a higher $\sigma_{v,i}$ for valence, the valence ratings for the word were spread across a wider range of values. If $\sigma_{v,i}$ were lower, ratings for the word clustered closer to the mean. Based on this, we use the probability density function[18] of a normal distribution to estimate the probability of the word's rating falling exactly at the mean.

Notice that if we'd simply used an arithmetic mean to compute the overall mean

valence M_v, we would have reported $M_v = (6.81 + 8.38)/2 = 7.56$. However, the standard deviation of valence for health ($\sigma_{v,1}$ = 1.88) is higher than the standard deviation for win ($\sigma_{v,2}$ = 0.92). Because of this, we weight win's mean valence $\mu_{v,2} = 8.38$ higher than health's mean valence $\mu_{v,1} = 6.81$. How we allocated the weight is explained below:

The normal distribution is parametrized in terms of the mean and the variance, denoted by μ and σ^2 respectively, giving the family of densities

$$f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \qquad 9-11$$

Hence, the probability of the word health's rating falling exactly at the mean could be calculated by the formula above.

When $x = \mu_{v,1} = 6.81$, $\sigma_{v,1} = 1.88$, we derive $f(x) = \frac{1}{1.88*\sqrt{2\pi}} = 0.212$.

Now we allocate the word 'health' with weight $W_1 = 0.212$.

We could use the same steps to derive the weight of the word win $W_2 = 0.434$

Then we can calculate the overall mean:

$$M_v = (\mu_{v,1} * W_1 + \mu_{v,2} * W_2)/(W_1 + W_2) = 7.86 \qquad 9-12$$

Hence, the result is an overall mean M_v = 7.86 that falls closer to win's mean valence. A similar result can be seen for overall mean arousal M_a.

The probabilities are applied as weights when we sum the means. Using this formula, we compute an overall mean valence and arousal of: $M_v = 7.86$, $M_a = 6.48$

For other situations that multiple words documented in ANEW dictionary, we could use the similar way to combine them:

$$M_v = \frac{\mu_1 * W_1 + \mu_2 * W_2 + \cdots + \mu_n * W_n}{W_1 + W_2 + \cdots + W_n} \qquad 9-13$$

In Fig 9-3 it shows the result by using the above mathematical method. Mood value has been calculated for corresponding distinct exercise types.

type	mood_value
running	6.08273
cycling	5.76242
swimming	6.07984
basketball	5.96129
volleyball	6.52989
tennis	6.15888
football	5.56301

Figure 9-3. Exercise types and their mood value

9.1.5. Weighted average exercise calorie consumption

9.1.5.1 Why calorie

1) Consider the intensity of exercise, instead of simply counting the number of tweets.

2) Applied to combine different exercises into the same scale.

9.1.5.2 Step 1: Refine keyword

1) More idiomatic.

2) Obey some rules to eliminate wrong patterns.

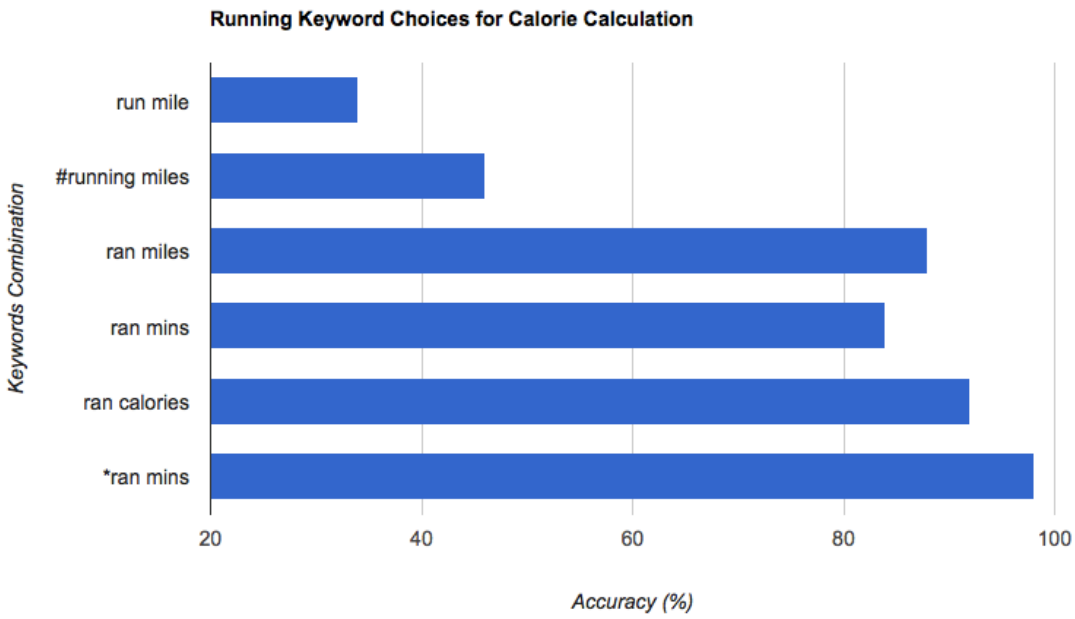
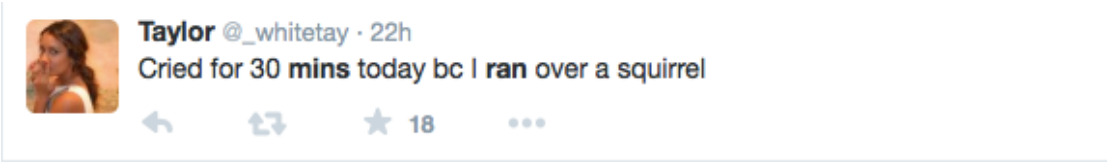


Figure 9-4 Running Keyword Choices for Calorie Calculation

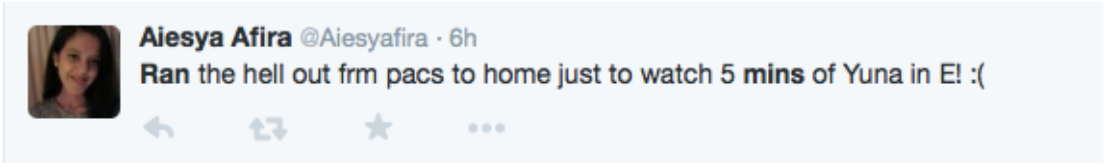
* presents for the keyword restricted by the rules.

Figure 9-4 shows the relevancy of tweets by different keywords. Using past tense contributes the idiomaticity. As a result, the hashtag is not the best choice. *ran miles*, *ran mins*, and *ran calories* are good candidates. But miles cannot be applied to other exercises, and *calories* appear in the tweet at low frequency. The accuracy of *ran mins* improved by obeying the rules below:

1) In-order



2) Short distance



3) No comma, dot, apostrophe between

4) No word group

The solution is the pattern matching. See details for pattern matching in MySQL[20] and PHP[21]. A SQL code can be:

$sql = "SELECT tweet_text, created_at FROM tweets_new WHERE tweet_text LIKE '%".$Exe_type[$j]." % mins %' AND tweet_text NOT LIKE '%ran out%' AND tweet_text NOT LIKE '%".$Exe_type[$j]." %[,.]% mins %'";

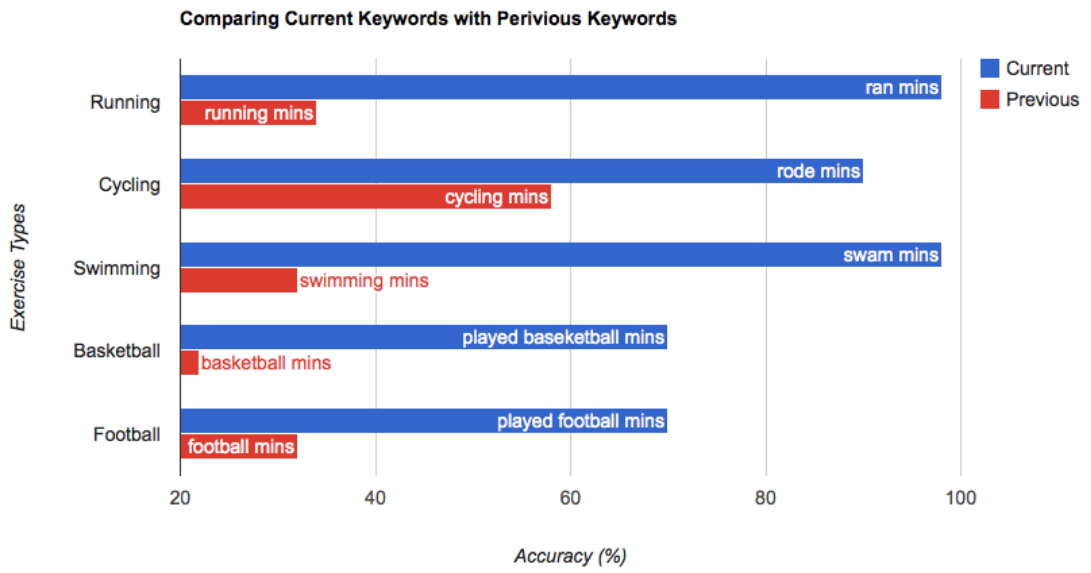


Figure 9-5 Comparing Current Keywords with Previous Keywords

Figure 9-5 shows the relevancy of tweets by previous and current keywords.

9.1.5.3 Step 2: Calculate calorie

1) Extract the time of exercise.

The time extraction accuracy of original keywords is really low.

Tweet: This shower been running for a good 15 minutes I'll get in eventually

Extracted hour: 0.25

The hour value is qualified

Tweet: @jxrxtbuchxnxn I'm running errands like rn so I'll be like ten minutes away

Extracted hour: 0

The hour value is out of scope

Tweet: Cold opening; rising tension; humor; nudity; murder; running & screaming for 90 minutes until it just sort of ends. #SpeedTweetAHorror

Extracted hour: 1.5

The hour value is qualified

Tweet: My dad was running around the house in his boxers after the game today for

nearly 10 minutes straight.. Srsly... http://t.co/UDQyTib47L

Extracted hour: 0.16666666666667

The hour value is qualified

Tweet: superior: 61 running, 3 waiting, 0 held, 2 new jobs w/ .13 mins wait time; 72% http://t.co/3SJDeawrrL

Extracted hour: 0.21666666666667

The hour value is qualified

The time extraction with refined keywords applied rules above and threshold for different exercises has little error.

Tweet: Ran 6.23 miles in 55 mins and felt great. Sturbridge 10K: Sunrise run through banks of fog. It was like eyes

Extracted hour: 0.91666666666667

The hour value is qualified

Tweet: Ran 13.75 miles in 2 hours and 20 mins and felt tired. http://t.co/fEu18g1OW6

Extracted hour: 2.3333333333333

The hour value is qualified

Tweet: Ran 13.08 kilometers in 1 hour and 9 mins and 42 secs and felt alright. Ran to Seepz to attend Swacch abhiyan at ... http://t.co/zAJMq4CmzT

Extracted hour: 1.15

The hour value is qualified

Tweet: Ran 5.03 miles in 49 mins and felt great. Pre-Sunday football run! 5.03 miles. t http://t.co/zQGB8NW44Q

Extracted hour: 0.81666666666667

The hour value is qualified

Tweet: Ran 7.51 kilometers in 45 mins and felt great. Zornjak! http://t.co/3mNYGWOYgm

Extracted hour: 0.75

The hour value is qualified

2) Calculate the weighted average calorie in different exercises.

$$Calorie = \frac{\sum_{i=0}^{n} W_i * C_i * \bar{t}_i}{\sum_{i=0}^{n} W_i} \quad 9\text{-}14$$

n is the number of exercise types.

W_i is the fraction of the exercise among all types.

C_i is the calorie burned rate per hour of different exercises.

$\bar{t}_i$ is the average time of different exercises from tweets.

The threshold and the fraction of exercise are collected from U.S. Bureau of Labor Staistics offical website[22]. And the calorie burned rate is collected from offical website[23].

9.1.5.4 Step 3: Visualize calorie

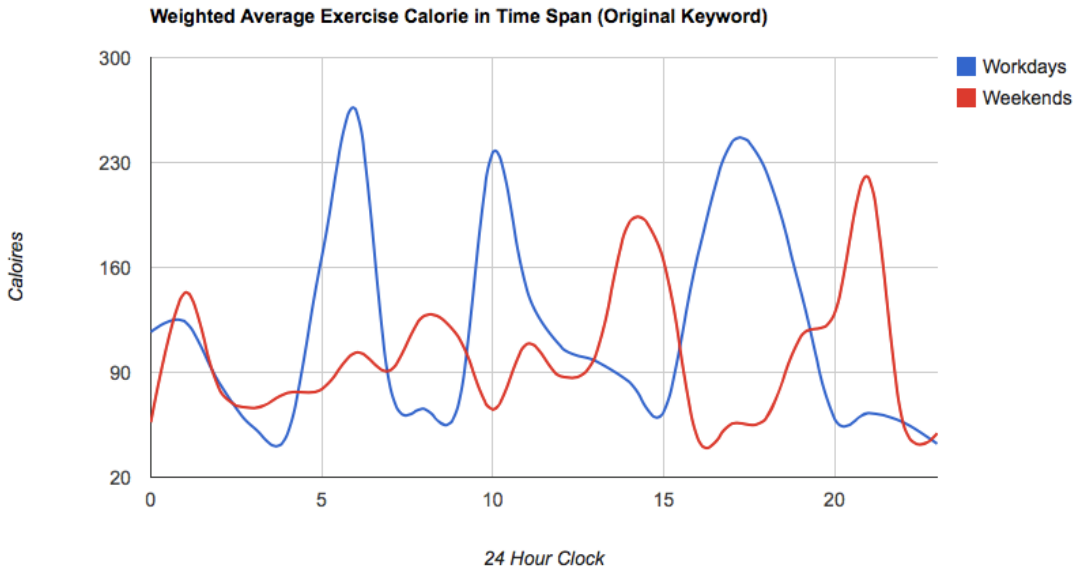


Figure 9-6 Time distribution with previous keywords

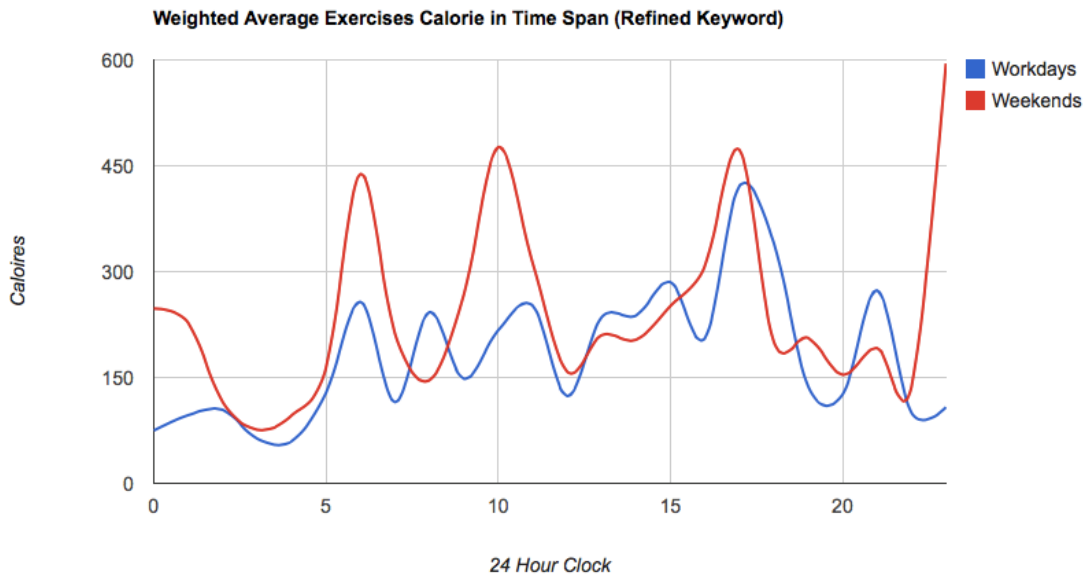


Figure 9-7 Time distribution with current keywords

As we can see in Figure 9-6 and Figure 9-7, the time distribution with current keywords is more reasonable, and much more similar with the official results [22].

9.1.6. Part of speech algorithm

Even though we would take key words within tweets to analyze the feature of ranking high frequency words it still, to some extent, remains at a facial layer since sole searching keywords will produce a large amount of unrelated words such as 'the', 'an', 'she' and etc.. Moreover it also produces practically same results such as 'run' and 'running', which a word with same meaning expresses in different form. Those two problems lay serious negative on the feature of ranking high frequency words. To optimize the feature we have to resolve this problem and thus a new algorithm, part of speech algorithm introduced by researchers in Stanford has been proposed.

In corpus linguistics, part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. [24]. In the following discussion we will use POS instead of part-of-speech.

Through analyzing POS within tweets words within tweets can be divided into different arrays including noun, adjective, verb and etc. according to POS of each word. Based on the discussion in "Opinion mining and sentiment analysis", Bo Pang and Lillian Lee indicate that during the analyze of popular topics and sentiment towards those topics there remains a simple method that nouns occurring in high frequency and adjectives occurring in high frequency can represent popular topics and sentiments towards them respectively.

Thus besides searching keywords within tweets, in a deep analyzing way we can obtain POS of each word and then divide them into arrays of nouns and adjectives, then by using these two arrays it is easy for us to better analyze popular topics and sentiments.

Then we will discuss the algorithm of POS in detail including its modeling, training and parsing [25].

For the modeling section we can see from figure 9-8 the modeling has three layers involving Softmax layer, Hidden layer and Input layer. For the first layer it has three inputs including words, POS tags and arc labels represented by x^w_i and x^t_i x^l_i. Arc labels are relatively small discrete sets which show many semantical similarities like words. The reason we take arc labels as third input is that it operates as an assisting function to decide the POS of each word. For instance, NN (singular noun) should be closer to NNS (plural noun) than DT (determiner), and amod(adjective modifier) should be closer to num (numeric modifier) than subj (nominal subject). Therefore analyzing arc labels can ensure POS in effective way.

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3 \qquad 9-15$$

By inputing those three parameters into hidden layer which defines a cube activation function where W_1 and b_1 are the specific bias factors defined by the Danqi Chen, author of A Fast and Accurate Dependency Parser using Neural Networks. Though calculating the values of h with the equation serval distinct values of h for one input can be obtained because for this step the POS algorithm can have multiple possible results and can't ensure which is the correct one. Thus then to input those values of h into third layer, Softmax layer, it compares those values and then uses the maximum value as the determined one which indicates to a specific POS tags for the word inputed.

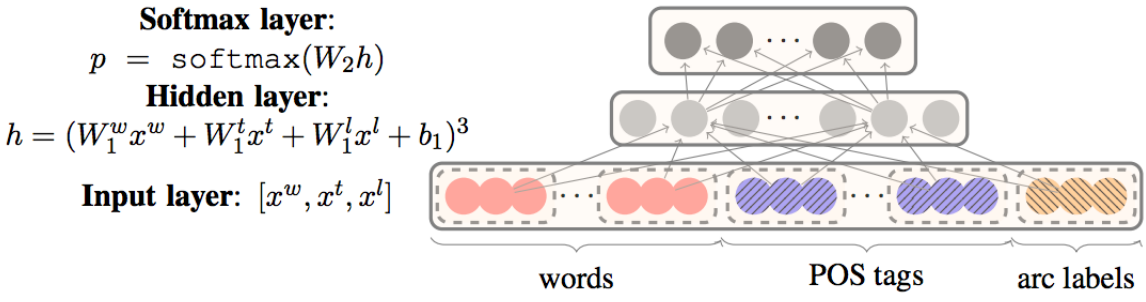


Figure 9-8 Modeling of POS algorithm

The section discussed above is basic modeling of POS algorithm, then it is critical to training the system. The researchers use the pre-trained word embedding from Colorer et al., 2011 for English (#dictionary = 130,000, coverage = 72.7%), and their trained 50-dimensional word2vec embedding Mikolov et al., 2013 on Wikipedia. They will also compare with random initialization of sets of words. The derivatives of training error will be back-propagated to these embeddings during the training process.

After training the POS algorithm it can execute the function of analyzing POS of each input word. Then we will discuss how to utilize this POS algorithm on our project.

First to regard all collect tweets related to exercising and healthy topic as inputs the POS algorithm produces corresponding analyze results of given tweets. For example to input a tweet:

> "100 push-ups a day, 75 golf balls hit a day, run 2 miles a day, workout with trainer 4 times a week. I am especially happy for my plan."

The system will produce the following result showed in Figure 9-9.

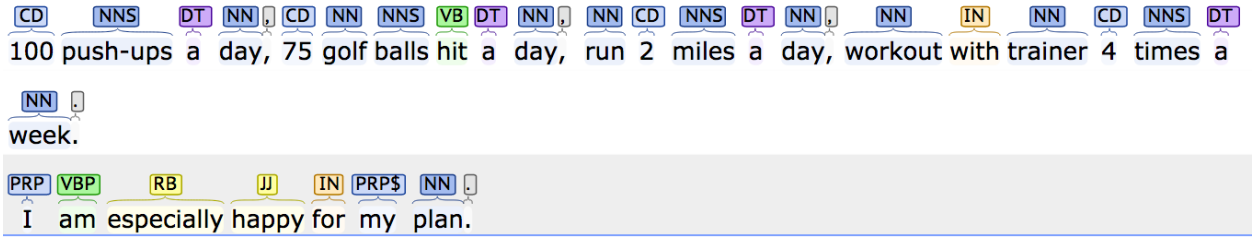


Figure 9-9 Example for POS

The result shows it offers POS of each word of this example tweet. In practical procedure we input all tweets into the system and word can be divided into different arrays of nouns and adjectives. Within individual array we then rearrange and rank them according to their occurring frequencies therefore word with specific POS. The new ranking arrays of nouns and adjective will be set into new tables by which they actually document popular topics and related sentiments on our given topic, health and exercising.

9.2. Data Structures

9.2.1. Set filter keywords

Before embarking on crawling raw data from twitter database keywords libraries has been set by carefully analysis and comparison. First seven popular exercising types involving 'running', 'cycling', 'swimming', 'basketball', 'volleyball', 'tennis' and 'football' have been decided for keywords of exercise.

However only by using the above exercising keywords solely will seriously impact the accuracy of the searching results, which includes a large amount of unrelated information. For example, through searching the tennis, the following

shows the 10 results:

WTA @WTA Nov 8

15 MILLION! @MariaSharapova becomes 1st #tennis player to hit 15 MILLION LIKES on Facebook--> http://wtatenn.is/vOYoQc

SI Tennis @SI_Tennis 15m15 minutes ago

Petra Kvitova wins a thriller to deliver Czech Republic its 3rd Fed Cup title in four years. http://www.si.com/tennis/2014/11/09/petra-kvitova-wins-epic-clinch-fed-cup-title-czechs …

Women Love Sports @Women_Sports 2h2 hours ago

The mark of great sportsmen is not how good they are at their best, but how good they are their worst. -Martina Navratilova (tennis)

Sportupdate_ID @SportUpdate_ID 3h3 hours ago

#SportUpdate - Andy Murray begins ATP World Tour Finals with a straight sets defeat to Kei Nishikori of Japan ... http://dailym.ai/1AMM0OY

SI Tennis @SI_Tennis 3h3 hours ago

Kei Nishikori gets his first ever win over Andy Murray, beats him 64 64. Ugly match all around, but Murray now in a tough hole in Group B.

Ben Rothenberg @BenRothenberg 4h4 hours ago Czech Republic

Kerber won the set this weekend when she trailed a double break 0-3*, but lost all four sets when she had 4-2 leads. Tennis, man. #fedcup

SI Tennis @SI_Tennis 4h4 hours ago

Tough weekend for Kerber. Led by a break in both sets vs. Safarova and lost both, served 3x for 1st set vs. Kvitova, and led 41 in the 3rd.

SI Tennis @SI_Tennis 4h4 hours ago

Petra Kvitova seals it. Back from a break down in 1st & 1-4 in 3rd to beat Kerber 76(5), 46 64. 3rd title in 4 years for the Czechs.

rennae stubbs @rennaestubbs 4h4 hours ago

@TennisChannel thank god for tennis channel plus so I can watch the finals of fed cup! Great match between @AngeliqueKerber v @Petra_Kvitova

Anne Keothavong @annekeothavong 5h5 hours ago

Entertaining tennis and lots of drama in this #FedCup final. Fantastic fans creating a great atmosphere in the arena too

Because the main goal of our project is to research on the exercising time duration, frequency, consuming calories, it is easy to find that none of above tweets is related to our topic. Moreover though calculate the rate of relevance of 100 tweets about 10 tweets are useful. Thus the accuracy of using one exercising keyword is absolute low.

Combination keywords

To improve the searching accuracy combination keywords have been set within the library.

The following are searching keywords.

Exercising Part

'running min', 'running mins', 'running minutes', 'running hour', 'running hours',

'cycling min', 'cycling mins', 'cycling minutes', 'cycling hour', 'cycling hours’,

'swimming min', 'swimming mins', 'swimming minutes', 'swimming hour', 'swimming hours’,

'basketball min', 'basketball mins', 'basketball minutes', 'basketball hour', 'basketball hours’,

'volleyball min', 'volleyball mins', 'volleyball minutes', 'volleyball hour', 'volleyball hours',

'tennis min', 'tennis mins', 'tennis minutes', 'tennis hour', 'tennis hours’,

'football min', 'football mins', 'football minutes', 'football hour', 'football hours',

'exercise min', 'exercise mins', 'exercise minutes', 'exercise hour', 'exercise hours',

'exercises min','exercises mins', 'exercises minutes', 'exercises hour', 'exercises hours',

'exercising min','exercising mins', 'exercising minutes', 'exercising hour', 'exercising hours',

Lifestyle Part

'keepfit', 'fitness', 'keep in shape', 'bodybuilding', 'keep healthy', 'loose weight', 'loosing weight',

'health apple', 'healthy apple', 'fitness apple', 'exercising apple', 'loose weight apple',

'health banana','healthy banana', 'fitness banana', 'exercising banana', 'loose weight banana',

'health lemon','healthy lemon', 'fitness lemon', 'exercising lemon', 'loose weight lemon',

'health orange','healthy orange', 'fitness orange', 'exercising orange', 'loose weight orange',

'health pear','healthy pear', 'fitness pear', 'exercising pear', 'loose weight pear',

'health milk','healthy milk', 'fitness milk', 'exercising milk', 'loose weight milk',

'health meat','healthy meat', 'fitness meat', 'exercising meat', 'loose weight meat',

'health vegetable','healthy vegetable', 'fitness vegetable', 'exercising vegetale', 'loose weight vegetale',

'healthcare', 'health life'

We can see clearly that besides exercising type keywords, time limitations are used to improve accuracy of results. Again we calculate the relevance level of those keywords by testing 50 tweets.

Exercising Type	Searching Keywords	Number of related tweets
running	running min	20/ 50
	running mins	17 / 50
	running minutes	11/50
	running hour	7/50
	running hours	13/50
swimming	swimming min	12/50
	swimming mins	16/50
	swimming minutes	13/50
	swimming hour	18/50
	swimming hours	17/50
cycling	cycling min	20/50
	cycling mins	29/50
	cycling minutes	19/50
	cycling hour	22/50
	cycling hours	25/50
basketball	basketball min	8/50
	basketball mins	11/50
	basketball minutes	15/50
	basketball hour	12/50
	basketball hours	21/50
volleyball	volleyball min	23/50
	volleyball mins	23/50
	volleyball minutes	12/50
	volleyball hour	15/50
	volleyball hours	18/50
football	football min	18/50
	football mins	16/50
	football minutes	17/50

	football hour	20/50
	football hours	16/50
tennis	tennis min	17/50
	tennis mins	21/50
	tennis minutes	14/50
	tennis hour	19/50
	tennis hours	17/50

Table 9-1. Number of related tweets

From the above statistics searching results we find that by using combination keywords the rate of related results is obviously improved to about 48% compared to 10% when using sole keywords.

Besides the exercising combination keywords we also add 9 array food keywords to research on the relationship between food and exercising.

9.2.2 Raw data

After ensuring the above keywords we use streaming API provided by Twitter to obtain tweets. Since the unprocessed raw data includes many unrelated information for the project. Thus it should extract main valuable information including the following ones.

tweet_id	Every tweet has its exclusive tweet id to identify the tweet.
tweet_text	Tweet_text includes concrete tweets information published by users.
created_at	Created_at documents the time when the tweets were published.
geo_lat	Geo_lat documents the latitude position where users published tweets.
geo_long	Geo_long documents the longitude position where users published tweets.
user_id	The exclusive labels that identify tweet users.
screen_name	The name of tweet users that displays on Twitter website.
profile_image_url	The web linkage of image of user profile.
location	The location where Tweet users belong to.

description	A introduction depiction of Tweet users.
followers_count	The follower number of a user.
friends_count	The friends number of a user.
statuses_count	The number of published Tweets.
time_zone	The time zone a Tweet user belongs to.
Gender	Gender of Tweet users
Age	Age of Tweet users

Table 9-2. Extracted useful Twitter information

Location information Analysis

The location information provided by tweet users is relatively unexpected. Here exists two main problems have to been processed.

(a) The first one is part of users only fill out partial location information, however an entire one involves state position and city position such as " San Francisco, CA". When considering different levels of belonging location, results of partial users will be ignored because of missing location information. For example if a user only fills in the location "San Francisco", then it couldn't be identified that belongs to CA.

To deal with the problem we add an extra location matching table to ensure the state locations when only city locations are given.

State	City
CA	Los Angeles
	San Francisco
	San Diego
	San Jose
	Long Beach
	Oakland
N.Y	New York
Illinois	Chicago
	Houston

Tex.	San Antonio
	Dallas
	Austin
Pa.	Philadelphia
Ariz.	Phoenix
Ga.	Atlanta
N.C.	Charlotte
Mass.	Boston
Wash.	Seattle
	DC

Table 9-3. Belonging states of popular cities

By index to the above table when only city locations are given, their corresponding state locations can be obtained. Specifically when count statistics in states, if we could not index state locations provided by users, then will index to the table by using provided city locations, once matching state locations are found, we can get their state locations. However to reduce the system response time only top 20 popular cities are given in the table.

(b) Another problem about location information is identical location information representing by different methods. For example when considering the state "CA", some users filled CA, some C.A, while some California. Thus when we use CA as state keyword to filter tweets another part using different state expressions will be missed.

To resolve the problem we should construct a state information table which involving all possible representation of a state.

state		
Alabama	AL	A.L
Alaska	AK	A.K
Arizona	AZ	A.Z
California	CA	C.A
Colorado	CO	C.O
Delaware	DE	D.E
Florida	FL	F.L

Georgia	GA	G.A
Hawaii	HI	H.I
Illinois	IL	I.L
Indiana	IN	I.N
Minnesota	MN	M.N
New Jersey	NJ	N.J
New York	NY	N.Y
North Carolina	NC	N.C
Texas	TX	T.X
Washington	WA	W.A
Wisconsin	WI	W.I

Table 9-4. Different expressions of cities

To test the numbers of related tweets by using different location expressions, we take five popular state involving "CA, FL, NJ, NY, GA" to test. The following table shows the analysis results.

California	2642
CA	5618
C.A	9
Florida	2086
FL	2028
F.L	8
New Jersey	728
NJ	1036
N.J	21
New York	3942
NY	4296
N.Y	63
Georgia	10
GA	1326

G.A	8

Table 9-5. Number of related tweets in 5 popular cities by different expressions

It is obvious to find from the table that when only Abbr. of states are used to identify locations, about a half information will be missed. Thus by filter tweets by using multiple location expressions will almost double the numbers of tweets.

Gender and Age Information Analysis

Gender and Age information are important ones that could not originally be extracted by using Twitter API. To obtain those information that will be critical index to analyze exercising when considering distinct gender and age people, third part API is used. However its accuracy becomes primary problems to use this API. To test the accuracy we send analysis information involving screen names, names and user descriptions of 30 famous people whose gender and age are known to the API.

```
"user_1" => array("screen_name" => "TomCruise", "name" => "Tom Cruise",
    "description" => "Actor. Producer. Running in movies since 1981.",
    "actural_gender" => "male", "actural_age" => "52"),

"user_2" => array("screen_name" => "ladygaga", "name" => "Lady Gaga",
    "description" => "The lady herself is not just a chameleon in person, but a
                      chameleon in music.",
    "actural_gender" => "female", "actural_age" => "28"),

"user_3" => array("screen_name" => "DwightHoward", "name" => "Dwight
            Howard",
    "description" => "No matter how far you fall you are never out of the fight.",
    "actural_gender" => "male", "actural_age" => "28"),

"user_4" => array("screen_name" => "kobebryant", "name" => "Kobe Bryant",
    "description" => "Dream Epic",
    "actural_gender" => "male", "actural_age" => "36"),

"user_5" => array("screen_name" => "JalenRose", "name" => "Jalen Rose",
    "description" => "Drum Major for Justice, Peace & Righteousness(MLK).
                      JRLA Founder. ABC/ESPN/Grantland Analyst. Phillipians
                      4:13.",
    "actural_gender" => "male", "actural_age" => "41"),

"user_6" => array("screen_name" => "SarahKSilverman", "name" => "Sarah
            Silverman",
```

```
    "description" => "We're all just molecules, Cutie.",
    "actural_gender" => "female", "actural_age" => "43"),

"user_7" => array("screen_name" => "PeteCarroll", "name" => "Pete Carroll",
    "description" => "Seattle Seahawks head coach. Always Compete. Win
                      Forever.",
    "actural_gender" => "male", "actural_age" => "63"),

"user_8" => array("screen_name" => "KevinSpacey", "name" => "Kevin
            Spacey",
    "description" => "Former shoe salesman now making a go at film and
                      theater. Wish me luck...",
    "actural_gender" => "male", "actural_age" => "55"),

"user_9" => array("screen_name" => "AliciaKeys", "name" => "Alicia Keys",
    "description" => "Passionate about my work, in love with my family and
                      dedicated to spreading light. It's contagious!;-)",
    "actural_gender" => "female", "actural_age" => "33"),

"user_10" => array("screen_name" => "Pink", "name" => "P!nk",
    "description" => "it's all happening",
    "actural_gender" => "female", "actural_age" => "35"),

"user_11" => array("screen_name" => "brookeburke", "name" => "Brooke
             Burke-Charvet",
    "description" => "Mommy first, wife, host, actress, fitness guru, CEO of
                      @ModernMom, Author of The Naked Mom,
                      co-creator/designer @CAELUM Lifestyle, Foodie,
                      @operationhmchef",
    "actural_gender" => "female", "actural_age" => "43"),

"user_12" => array("screen_name" => "mindykaling", "name" => "Mindy
             Kaling",
    "description" => "You can sit with us. Instagram: mindykaling",
    "actural_gender" => "female", "actural_age" => "35"),

"user_13" => array("screen_name" => "", "name" => "Nathan Fillion",
    "description" => "It costs nothing to say something kind. Even less to shut up
                      altogether.",
    "actural_gender" => "male", "actural_age" => "43"),

"user_14" => array("screen_name" => "GordonRamsay", "name" => "Gordon
             Ramsay",
```

```
    "description" => "Somewhere always near food.",
    "actural_gender" => "male", "actural_age" => "48"),

"user_15" => array("screen_name" => "Ali_Sweeney", "name" => "Alison
                Sweeney",
    "description" => "",
    "actural_gender" => "female", "actural_age" => "38"),

"user_16" => array("screen_name" => "ElizabethBanks", "name" => "Elizabeth
                Banks",
    "description" => "Amateur Goofball; proud native, Pittsfield, MA",
    "actural_gender" => "female", "actural_age" => "40"),

"user_17" => array("screen_name" => "ninadobrev", "name" => "Nina Dobrev",
    "description" => "Where ever you go... there you are. Going day by day... so
                let's see where it takes me! Namaste.",
    "actural_gender" => "female", "actural_age" => "25"),

"user_18" => array("screen_name" => "AudrinaPatridge", "name" => "Audrina
                Patridge",
    "description" => "~~Host of 1stLook!!! Airing after SNL on NBC~~
                Instagram-AudrinaPatridge",
    "actural_gender" => "female", "actural_age" => "29"),

"user_19" => array("screen_name" => "nerdist", "name" => "Chris Hardwick",
    "description" => "Stand-upper, Zombie Therapist, Talking Snake and
                POINTS giver",
    "actural_gender" => "male", "actural_age" => "42"),

"user_20" => array("screen_name" => "elizadushku", "name" => "Eliza
                Dushku",
    "description" => "Official Eliza Dushku. Be forewarned: I'm accused of
                speaking my own language here... Enjoy",
    "actural_gender" => "female", "actural_age" => "33"),

"user_21" => array("screen_name" => "ColinHanks", "name" => "Colin Hanks",
    "description" => "music geek/fan of sports/ husband/father/brother/son/
                person of interest to few/possibly that guy from that one
                thing you think is way underrated",
    "actural_gender" => "male", "actural_age" => "36"),

"user_22" => array("screen_name" => "paulfeig", "name" => "Paul Feig",
    "description" => "Paul is a guy who wears suits and tries not to screw things
```

```
                    up. He also created Freaks & Geeks, directed Bridesmaids
                    and The Heat and is currently making Spy.",
    "actural_gender" => "male", "actural_age" => "52"),

"user_23" => array("screen_name" => "ShannonElizab", "name" => "Shannon
                Elizabeth",
    "description" => "Co-founder of @ShansenJewelry, actress, director, writer,
                    producer, entrepreneur, vegan, animal activist &
                    philanthropist",
    "actural_gender" => "female", "actural_age" => "41"),

"user_24" => array("screen_name" => "katyperry", "name" => "KATY
                PERRY",
    "description" => "CURRENTLY✨BEAMING✨ON THE PRISMATIC
                    WORLD TOUR 2014!",
    "actural_gender" => "female", "actural_age" => "30"),

"user_25" => array("screen_name" => "selenagomez", "name" => "Selena
                Gomez",
    "description" => "Get ‘The Heart Wants What It Wants’ and pre-order my
                    new collection ‘For You’ - http://smarturl.it/sga1
                    Philippians 4:13",
    "actural_gender" => "female", "actural_age" => "22"),

"user_26" => array("screen_name" => "BradPaisley", "name" => "Brad Paisley",
    "description" => "In 1972, a crack commando unit was sent to prison by a
                    military court for a crime they didn't commit. I was also
                    born.",
    "actural_gender" => "male", "actural_age" => "42"),

"user_27" => array("screen_name" => "OzzyOsbourne", "name" => "Ozzy
                Osbourne",
    "description" => "The Prince of Darkness",
    "actural_gender" => "male", "actural_age" => "65"),

"user_28" => array("screen_name" => "elissakh", "name" => "Elissa",
    "description" => "Lebanese & International singer, 3 times World Music
                    Award! I m in halethob with my new album #halethob",
    "actural_gender" => "female", "actural_age" => "42"),

"user_29" => array("screen_name" => "ashleytisdale", "name" =>
                "AshleyTisdaleFrench",
    "description" => "My official twitter page!! Hoping my tweets inspire
```

```
                    you :)",
        "actural_gender" => "female", "actural_age" => "29"),

    "user_30" => array("screen_name" => "ashleytisdale", "name" => "Cher",
        "description" => "Stand & B Counted or Sit & B Nothing. Don't Litter,Chew
                    Gum,Walk Past Homeless PPL w/out Smile.DOESNT
                    MATTER in 5 yrs IT DOESNT MATTER THERE'S
                    ONLY LOVE&FEAR",
        "actural_gender" => "female", "actural_age" => “68")
```

The feedback results of the third part API are that 23 out of 30 gender prediction is correct, whose accuracy is 76.67% and 18 out of 30 age prediction is correct whose accuracy is 60%.

The test results show the accuracy of analysis gender and age is relatively high used to analyze common users’ genders and ages.

9.3 Data Organization

In computing, a hash table (hash map) is a data structure used to implement an associative array, a structure that can map keys to values. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the correct value can be found.[19]

Ideally, the hash function will assign each key to a unique bucket, but this situation is rarely achievable in practice (usually some keys will hash to the same bucket). Instead, most hash table designs assume that hash collisions—different keys that are assigned by the hash function to the same bucket—will occur and must be accommodated in some way.

In a well-dimensioned hash table, the average cost (number of instructions) for each lookup is independent of the number of elements stored in the table. Many hash table designs also allow arbitrary insertions and deletions of key-value pairs, at constant average cost per operation.

In many situations, hash tables turn out to be more efficient than search trees or any other table lookup structure. For this reason, they are widely used in many kinds of computer software, particularly for associative arrays, database indexing, caches, and sets.

For the above reasons we choose to take hash table to organize extracted Twitter data. For example the following table shows how we construct Tweets table.

tweet_id	tweet_text	created_at	geo_lat	geo_long	user_id	last_updated

Table 9-6. Hash table organization of tweets information

The tweets table shows that we organize tweets information into different hash and it is easy to reference each information by searching hash.

10. User Interface Design and Implementation

In this document, we will tell you, the user, to the mobile apps and the website, and explain how to use it. Our goal is to help users monitor the overall health related information of the American people from different aspects and get health related suggestions. The data we use here is a week data, from Oct. 13 to 19, crawled from the Twitter database.

10.1. IOS Application

When you first navigate to the page, you will encounter the following screen:

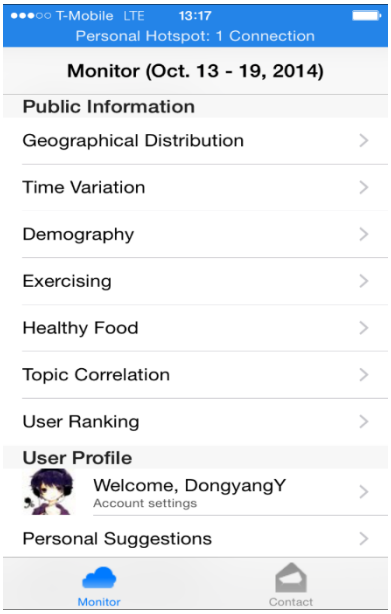


Figure 10-1. Monitor

In Figure 10-1, this interface contains two list views, “Public Information” and “User Profile”. Each item in the “Public Information” list shows the specific analysis of the public who are tweeter users. Each item in “User Profile” list shows analysis concerning user’s own tweet data. If you click the Geographical Distribution, this app will navigate you to the next screen:

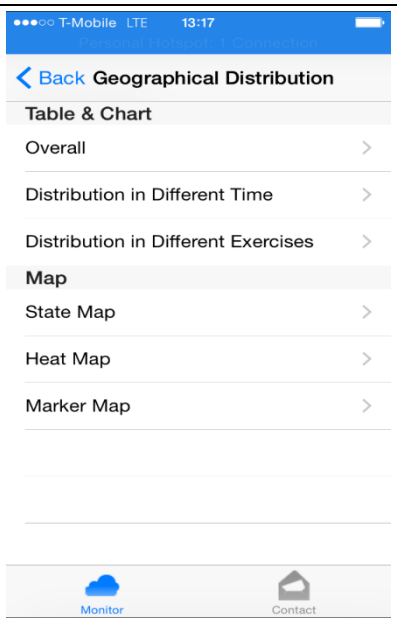


Figure 10-2. Geographical Distribution

In Figure 10-2, this interface contains a sub menu in which each item contains analytical information concerning the exercise intensity, distribution of the public. In the geographical distribution, you can see the amount of tweets in different locations such as a state map as below:

Figure 10-3. State Map

In Figure 10-3, this interface shows a US map in which each state contains information of the amount of tweets concerning exercise and health. The more the number of tweets being tweeted, the darker the color of the certain state will be. Thus California has the most amount of tweets. You may also see a heat map as below:

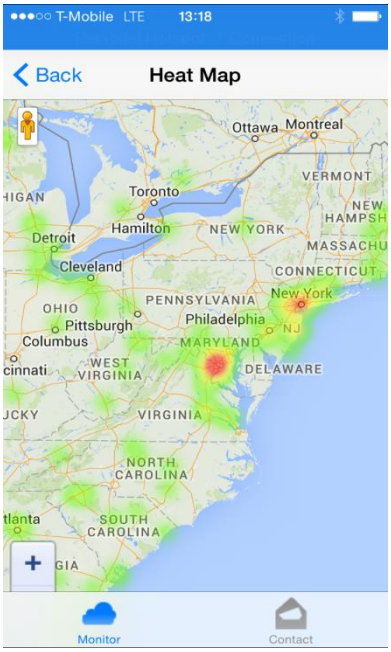


Figure 10-4. Heat Map

In Figure 10-4, when the color is closer to red, the amount of tweets is larger. You may also see a maker map as below:

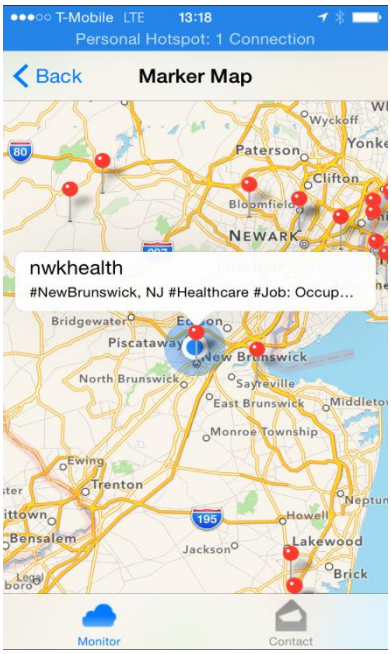


Figure 10-5. Marker Map

In Figure 10-5, this interface shows a Google map where marks will be showed on it if users post their tweet with location info. The interface will pop out certain user's name and his tweets if a single mark is clicked. You can see the recent health tweets around you, such as this figure above. Besides, we are willing to support you to see that information in different time period in a day and in different exercise types.

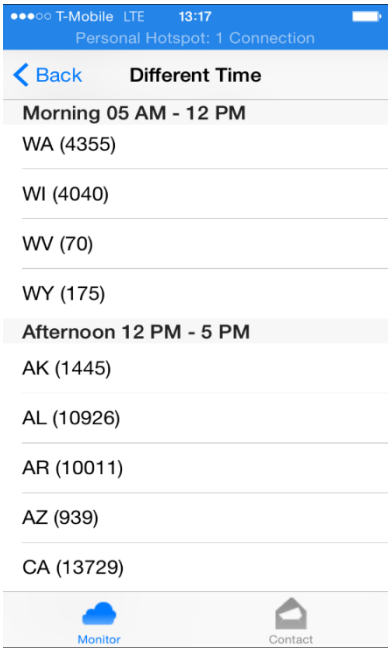


Figure 10-6. State Tweets by Different Time

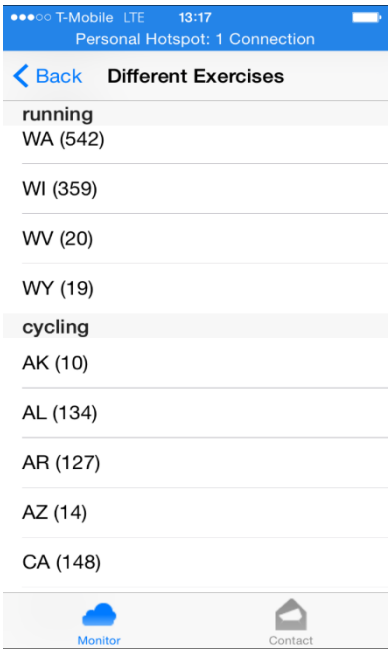


Figure 10-7. State Tweets by Different Exercise

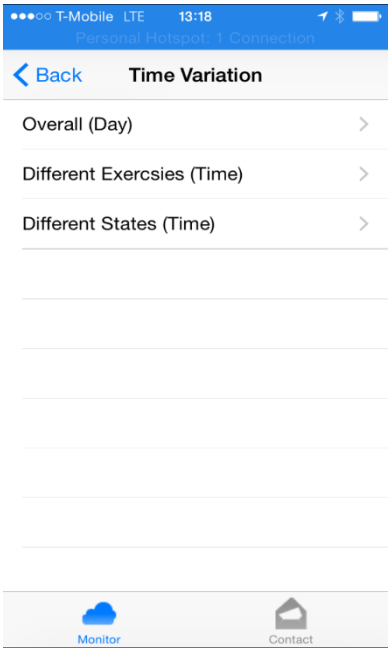


Figure 10-8. Originally: Time Variation

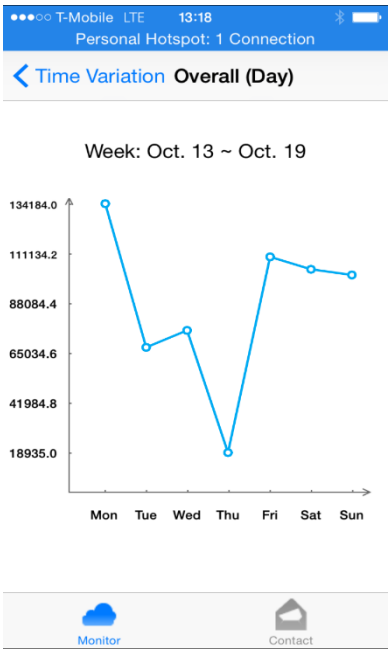


Figure 10-9. Trend of the week

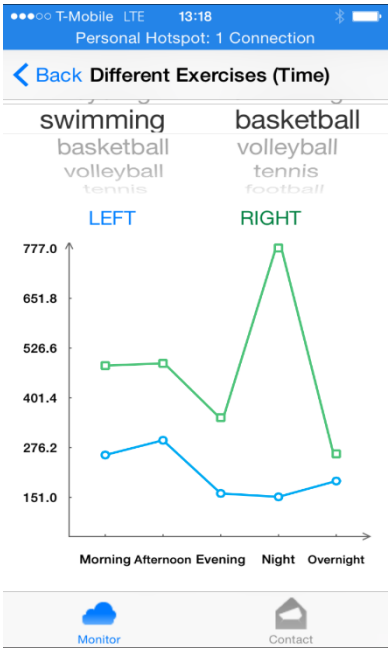


Figure 10-10. Trend by Different Exercise

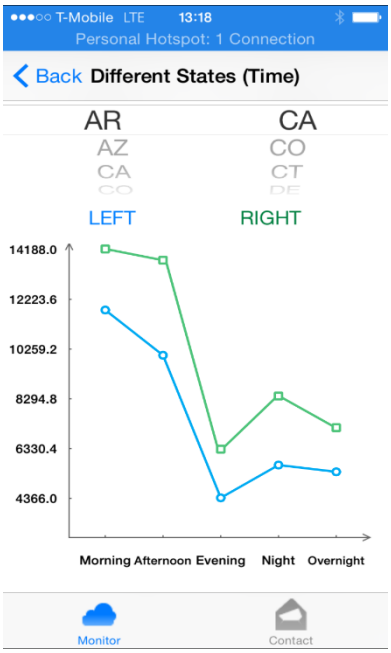


Figure 10-11. Trend by Different States

In the time variation, you may see the trend based on different days in Figure 10-9, and based on different time periods in different exercises in Figure 10-10, you can compare them like this. We also have states classification in Figure 10-11.

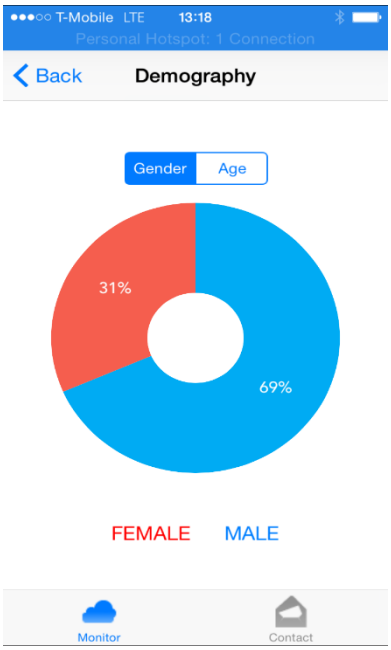


Figure 10-12. Gender Distribution

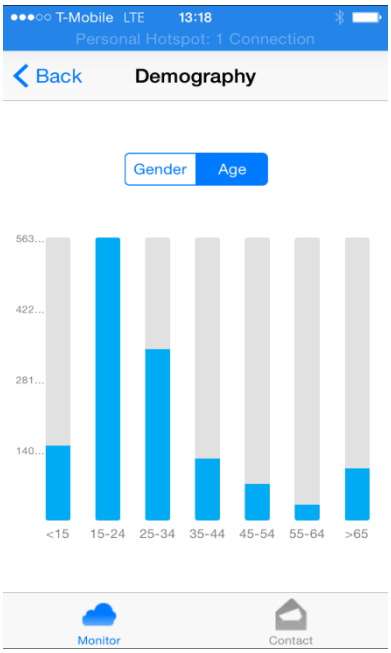


Figure 10-13. Age Distribution

In the demography, we have overall gender distribution in Figure 10-12, and age distribution in Figure 10-13.

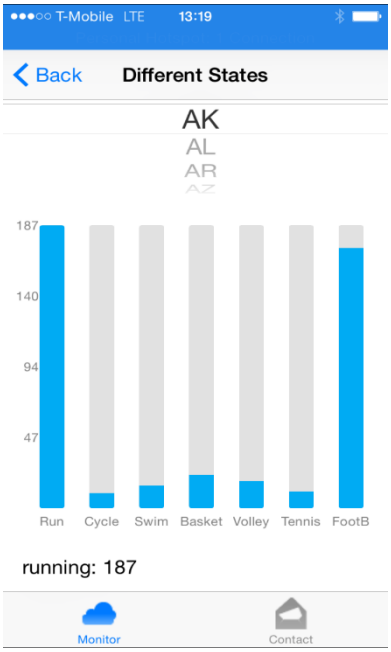


Figure 10-14. Exercise Classification by Different State

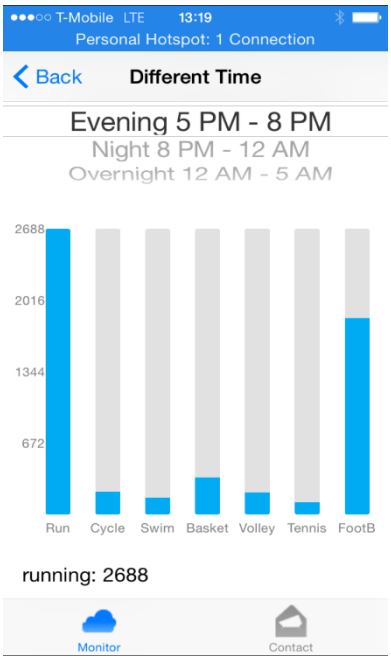


Figure 10-15. Exercise Classification by Different Time

In the exercising classification, you still can see the amount of tweets in different states in Figure 10-14 and time in Figure 10-15, you can touch the bar to see the detail.

Figure 10-16. Exercise time by Different State

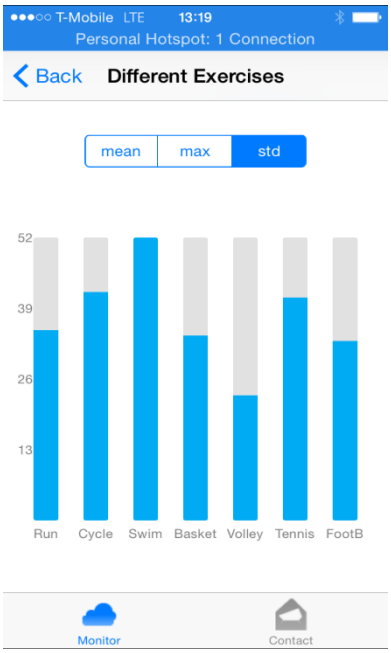


Figure 10-17. Exercise time by Different Types

Also, you can see the average, maximum, and individual difference of exercising time duration in Figure 10-16. This interface displays a bar chart of the “mean”, “max” and “standard deviation” of exercise duration time varied by different exercise types in Figure 10-17.

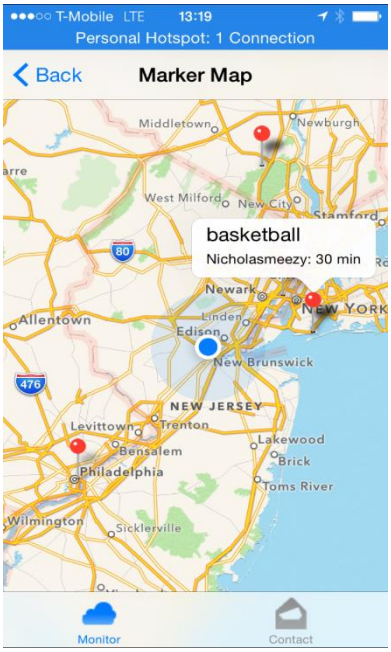


Figure 10-18. Exercise Marker Map

In Figure 10-18, we also have a marker map here. This interface shows markers on the map for users who have tweeted a tweet concerning exercise and mentioned his exercise duration time. Once the marker is clicked, the interface will pop out the duration time and the exercise type on the top of the marker. For example, a person called Nicholasmeezy has just played basketball for 30 minutes.

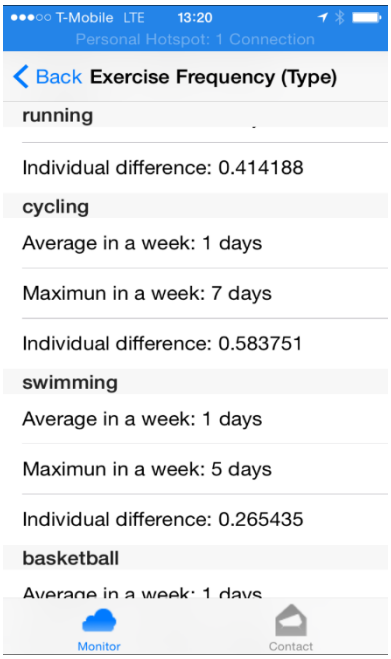


Figure 10-19. Exercise Frequency Estimation

We also have exercising frequency estimation in Figure 10-19.

In the sentiment state map, you can see the mood when people are exercising in Figure 10-20. For example, Alaska's mood value is the average.

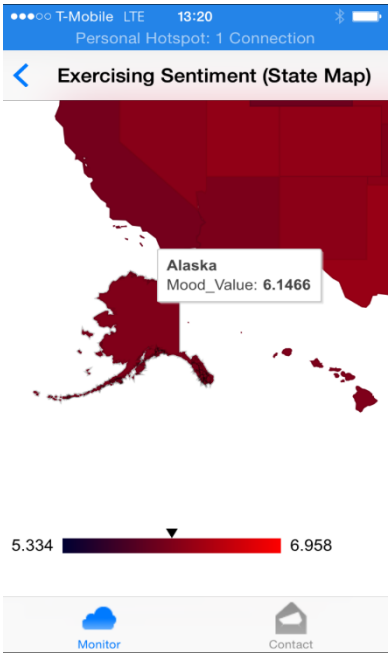


Figure 10-20. Sentiment State Map

In Figure 10-21, this interface includes the analysis of healthy food, sorted by either different states or different exercise. For example, in Arizona people like apple most in Figure 10-22 and in the running type people like apple and pear most in Figure 10-23.

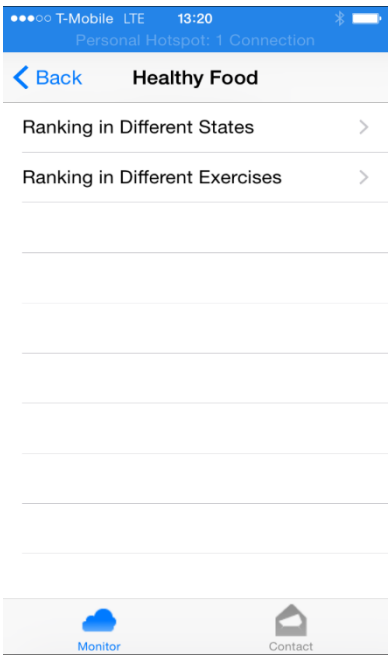


Figure 10-21. Healthy Food

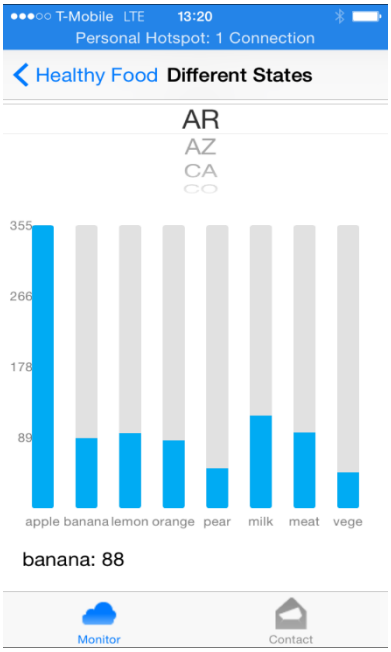


Figure 10-22. Food Analysis by Different State

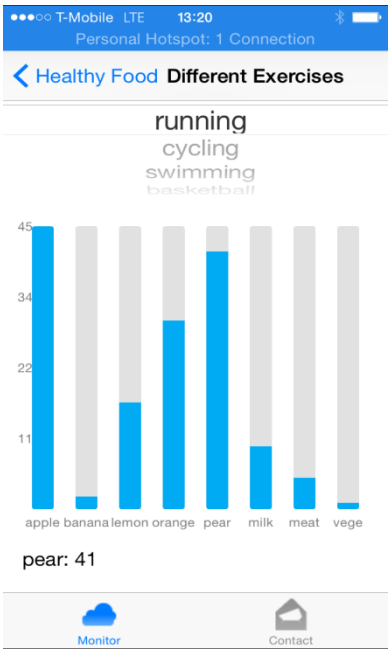


Figure 10-23. Food Analysis by Different Exercise Types

In Figure 10-24, this interface shows topic correlation between exercise and health, exercise and fruit, health and fruit. For instance, form the figure below you may see the amount of tweets about health is positive correlated with the amount of tweets about exercise.

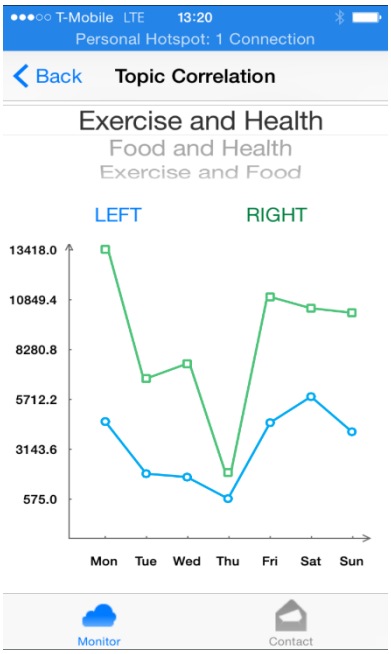


Figure 10-24. Topic Correlation

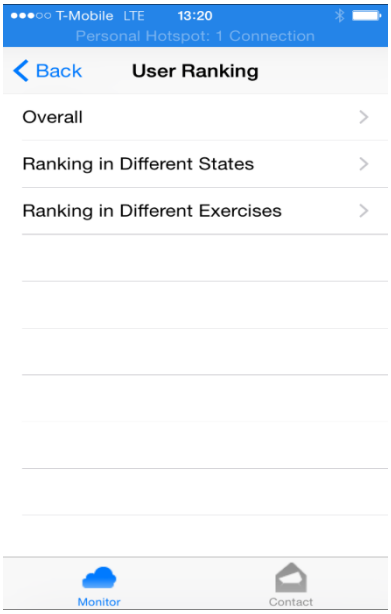


Figure 10-25. User Ranking

In Figure 10-26, this interface displays the ranking list of the users who exercise the most often in different states.

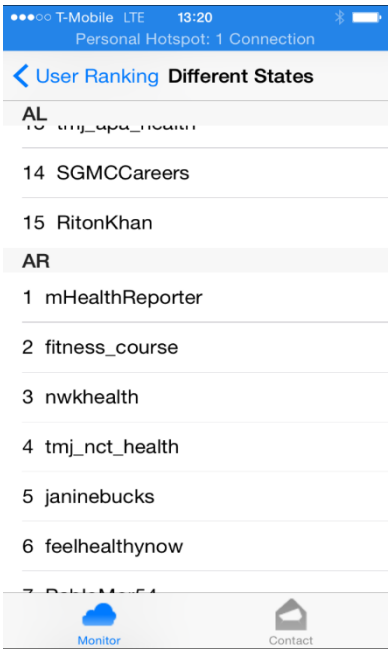


Figure 10-26. User Ranking by Different States

In Figure 10-27, this interface displays the ranking list of the users who exercise the most often in different exercise types.

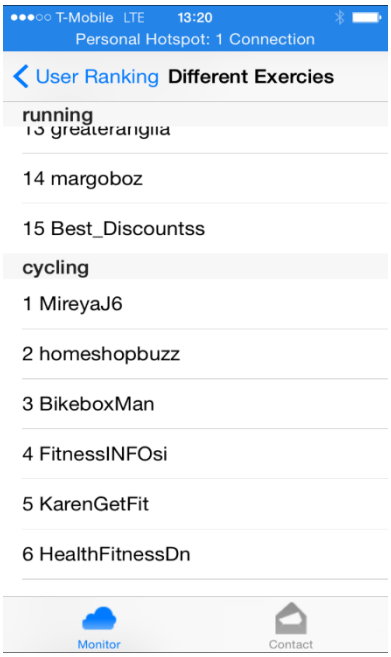


Figure 10-27. User Ranking by Different Exercise

In Figure 10-28, in this interface the users can see an analysis of their personal

exercise history. Personal exercise duration, average level in area and the difference can be shown.

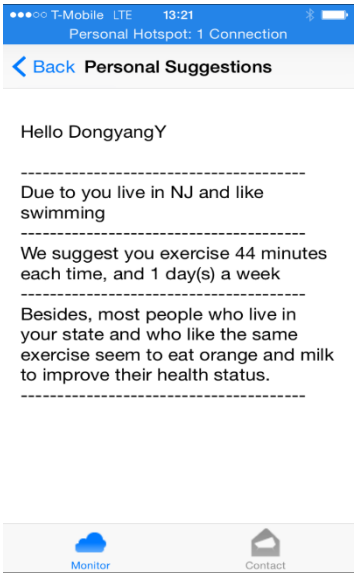


Figure 10-28. Personal Suggestions

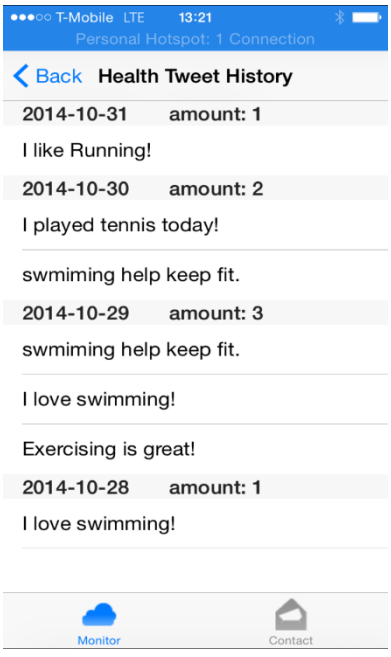


Figure 10-29. Health Tweet History

In Figure 10-29, this interface shows the history of user tweets related about Health.

10.2. Website

Figure 10-30: Homepage of the website

In the Figure 10-30, this interface shows the homepage of our project website. You can see there are four section: Home, Public display, Private profile and About us. What's more, you can see our features on the left as well. You may click any of them if you are interested in.

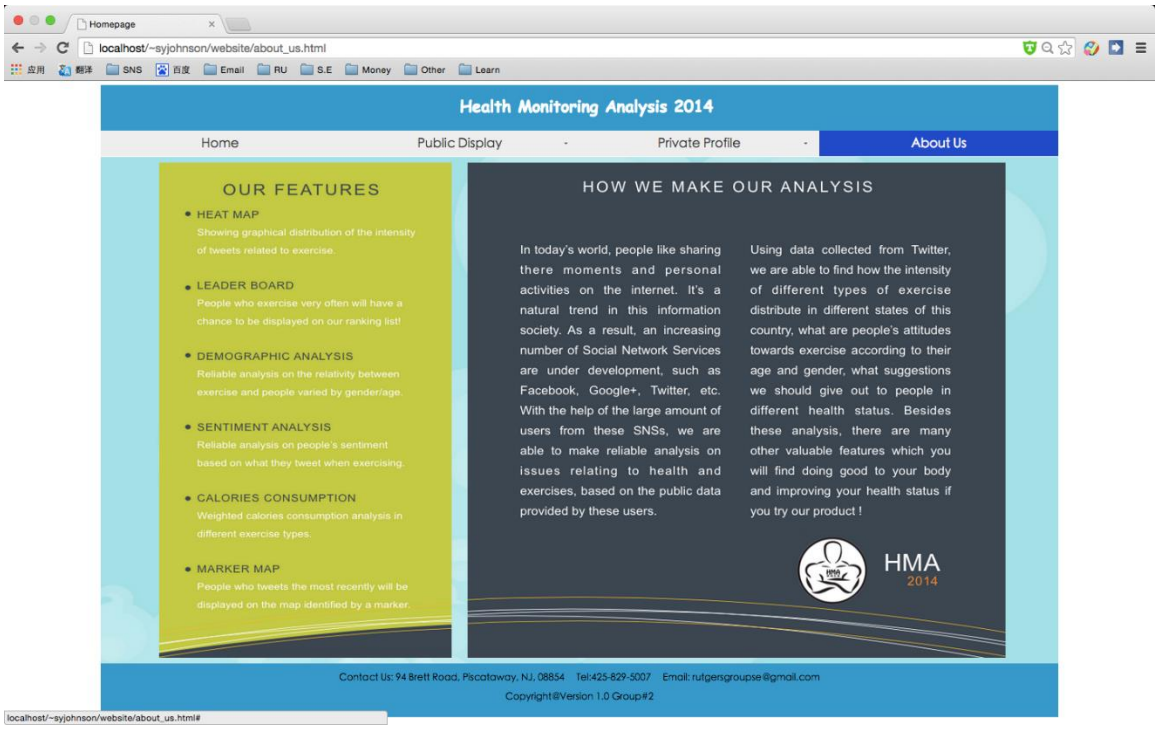


Figure 10-31: About Us

In the Figure 10-31, this interface you may see our project mainly features such as Heat Map, Leaderboard, Demography Analysis, Sentiment Analysis, Calories Consumption, Marker Map and how we make our analysis.

Figure 10-32: Public Display

In the Figure 10-32, when you click the "Public Display" button, you may see the different features such as Geographical Distribution, Time Variation, Demography, Exercising Type, Healthy Food and Correlation. You can click any of them if you are interested in.

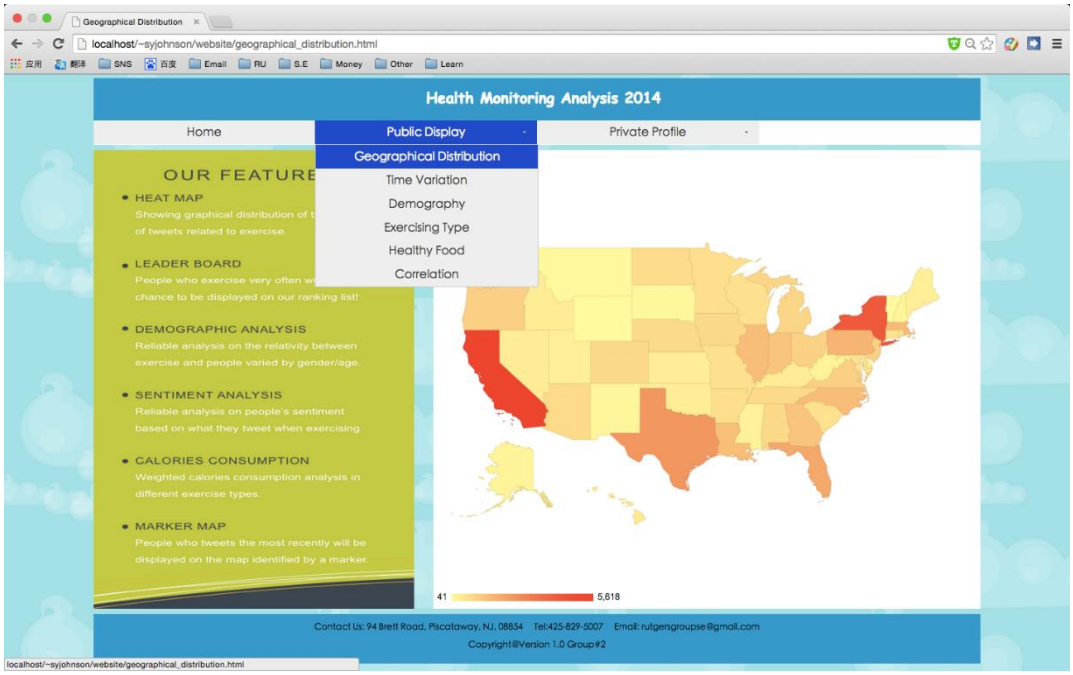


Figure 10-33: Geographical Distribution

In the Figure 10-33, when clicking on this button of geographical distribution, you can see the amount of tweets in different locations such as a state map. This interface shows a US map in which each state contains information of the amount of tweets concerning exercise and health. The more the number of tweets being tweeted, the darker the color of the certain state will be. Thus California has the most amount of tweets.

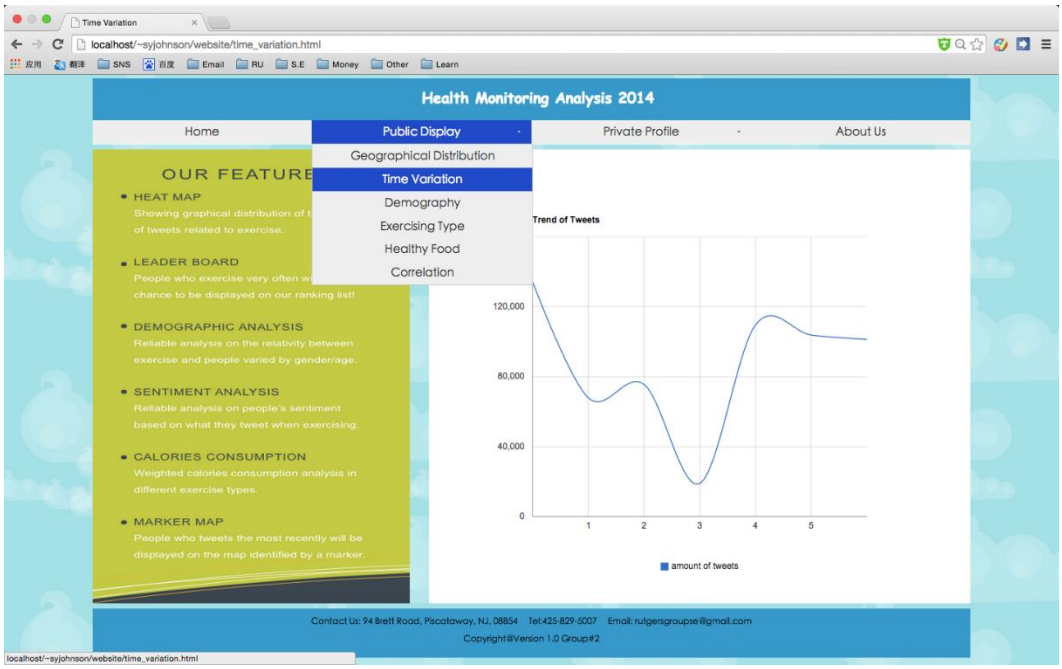


Figure 10-34: Time Variation

In the Figure 10-34, when clicking on this Time Variation button, you may see the line chart of the exercise tendency in different exercise types as time goes by.

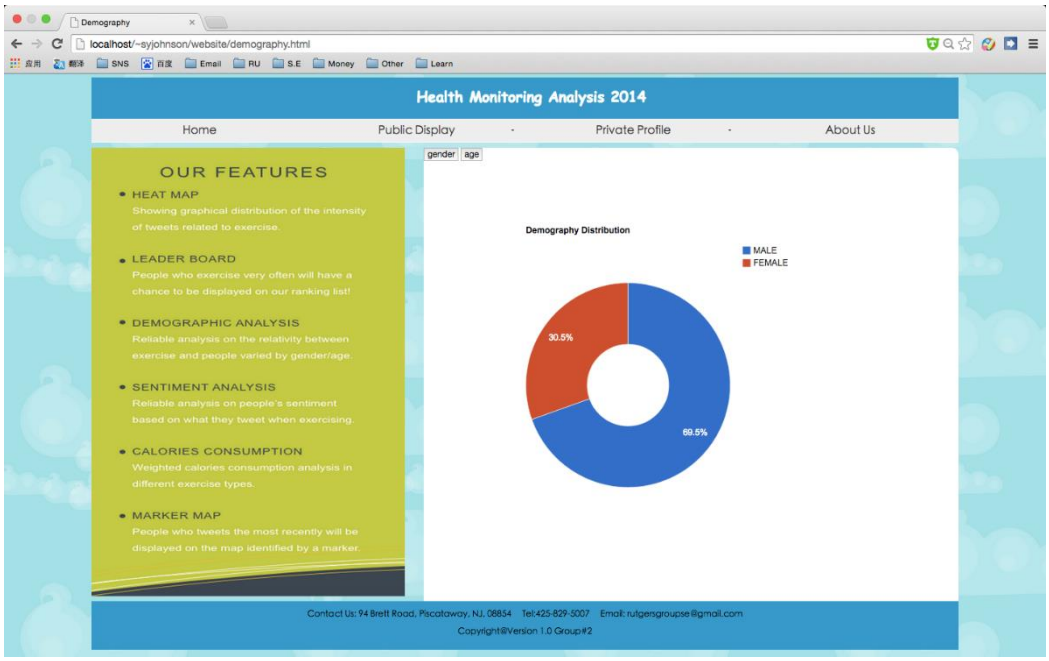


Figure 10-35: Demographical Analysis by gender

In the Figure 10-35: when clicking on the Demographical Analysis button, you may see the demographical distribution by gender.

Figure 10-36: Demographical Analysis by age

In the Figure 10-36: you can also click the age button to see the demographical distribution by age.

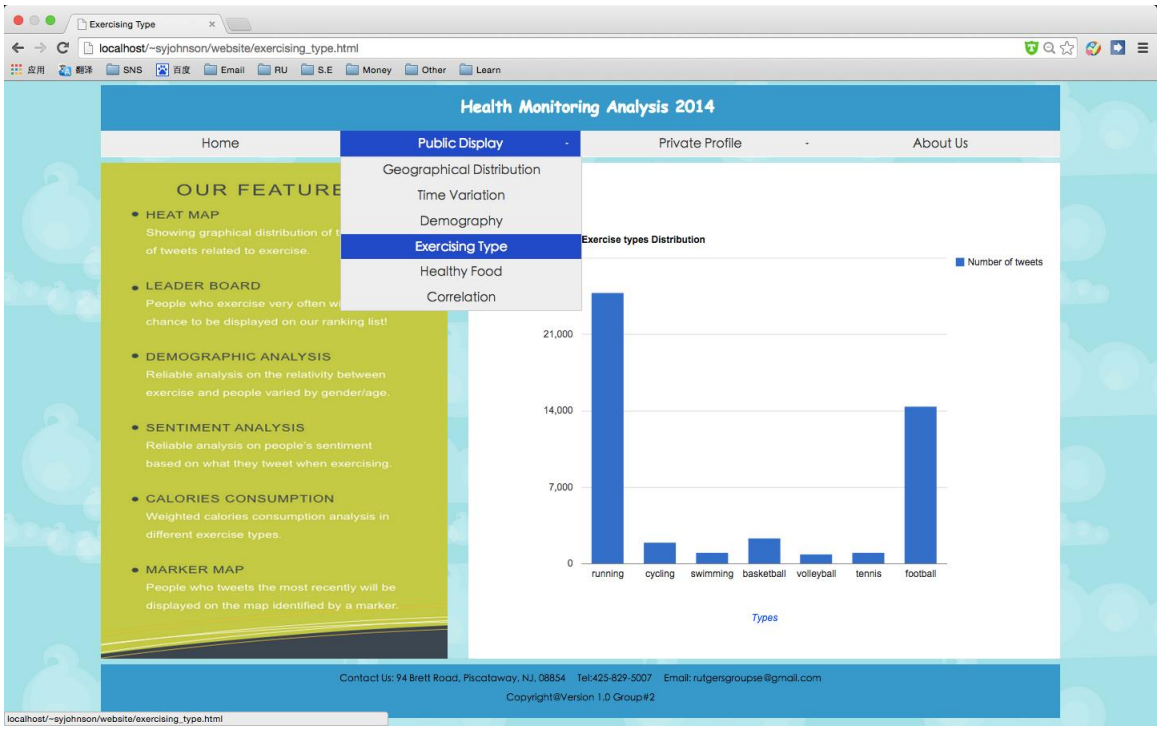


Figure 10-37: Exercising Type

In the Figure 10-37, when clicking on the Exercising Type button, this interface shows the comparison of different exercise type varied by the related amount of tweets

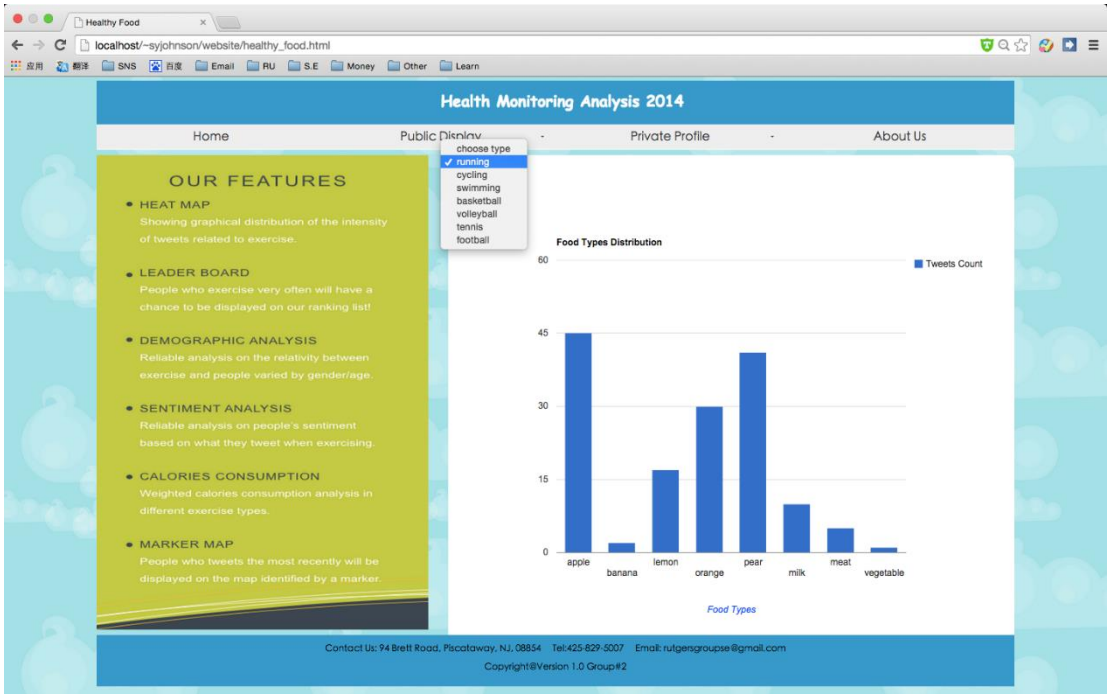


Figure 10-38: Healthy Food

In the Figure 10-38: when clicking on the Healthy Food button, the interface shows the analysis of healthy food, sorted by either different exercise type. For example, when you choose the running type, you may see in the running type people like apple and pear most.

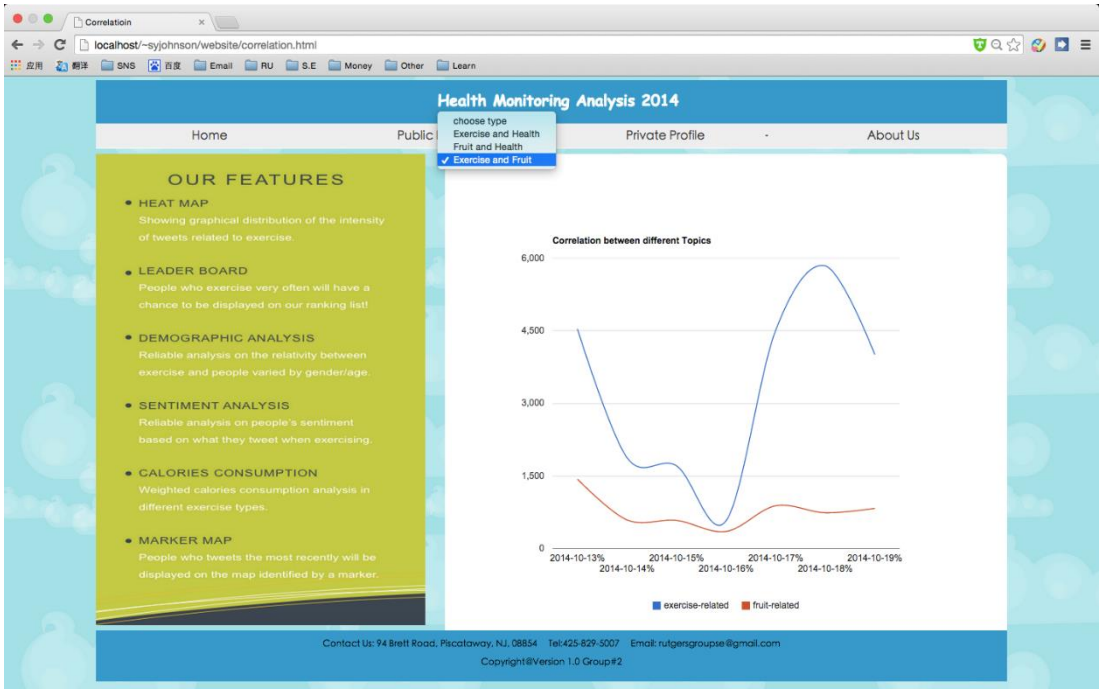


Figure 10-39: Topic Correlation

In the Figure 10-39, this interface shows topic correlation between exercise and health, exercise and fruit, health and fruit. For instance, from this figure you may see the amount of tweets about fruit is positive correlated with the amount of tweets about exercise.

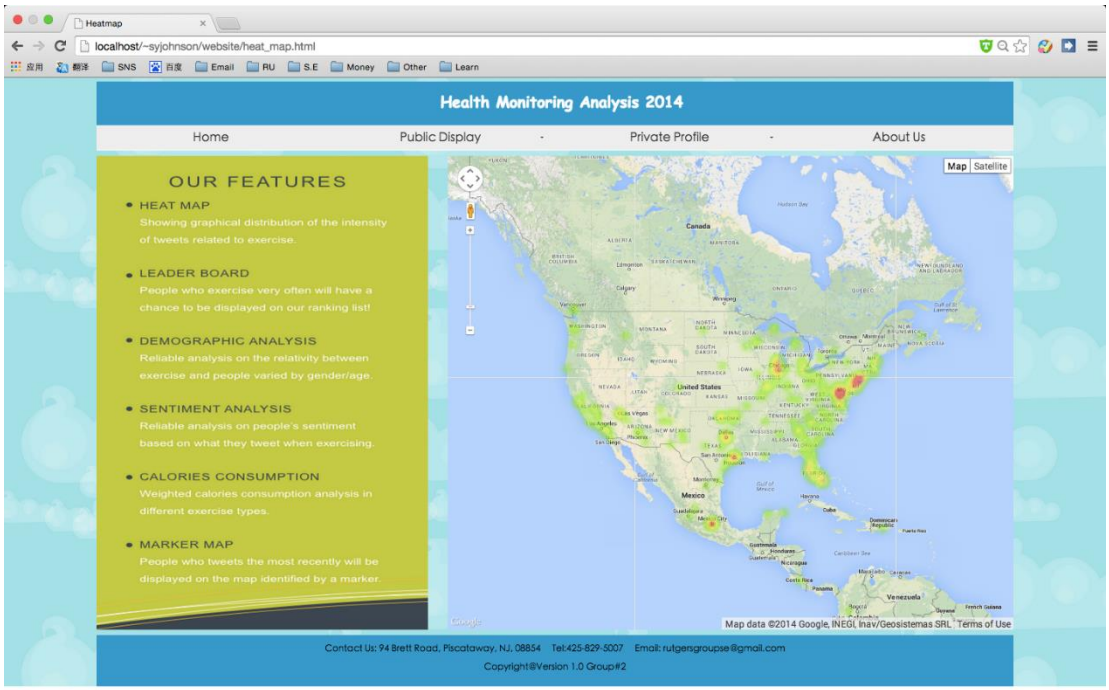


Figure 10-40: Heat Map

In the Figure 10-40, when clicking on the Heat Map button, this interface shows the exercise intensity in different areas all over the world. The red spot shows that there are more exercise fans, while the green spot shows there are less exercise fans.

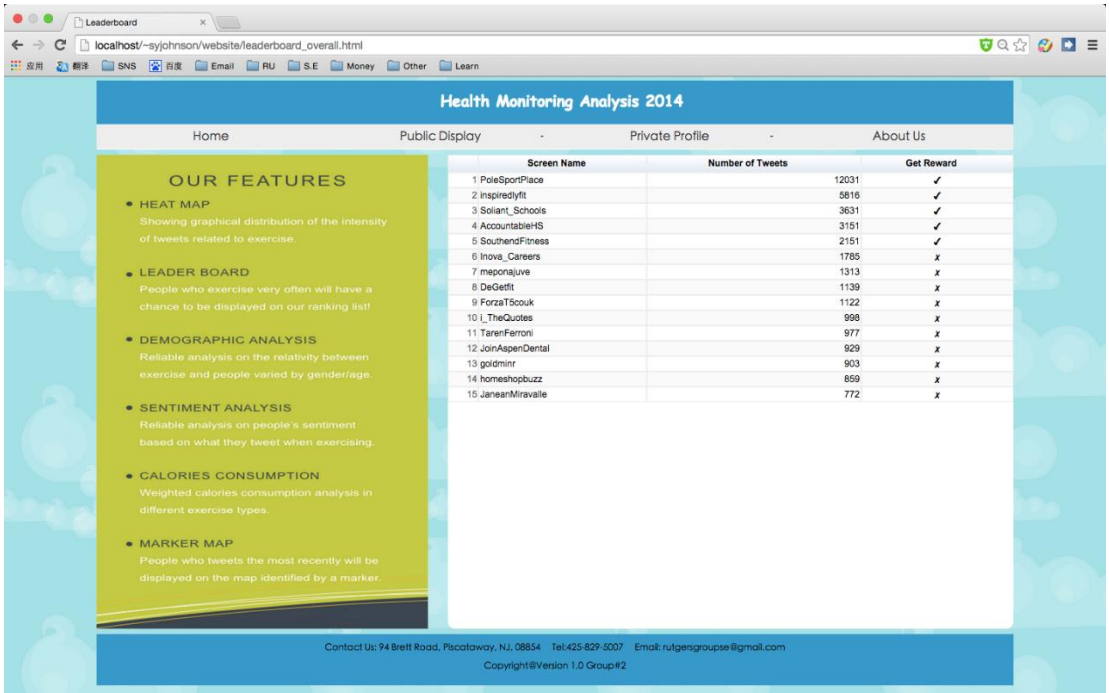


Figure 10-41: Leaderboard

In the Figure 10-41, when clicking on the Leader board button, this interface displays the ranking list of the users who exercise the most.

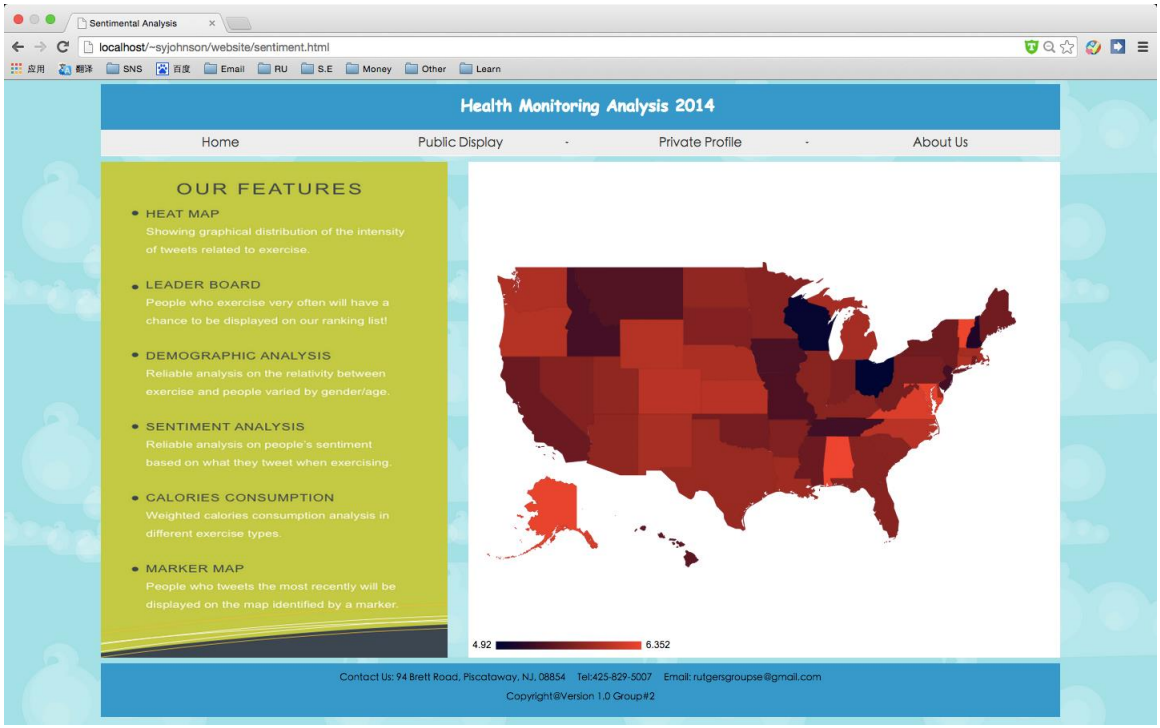


Figure 10-42: Sentiment Analysis

In the Figure 10-42, when clicking on the Sentiment Analysis button, you can see the mood when people are exercising. The sentiment state map shows that the mood varied by colors. The darker the state is , the mood is happier.

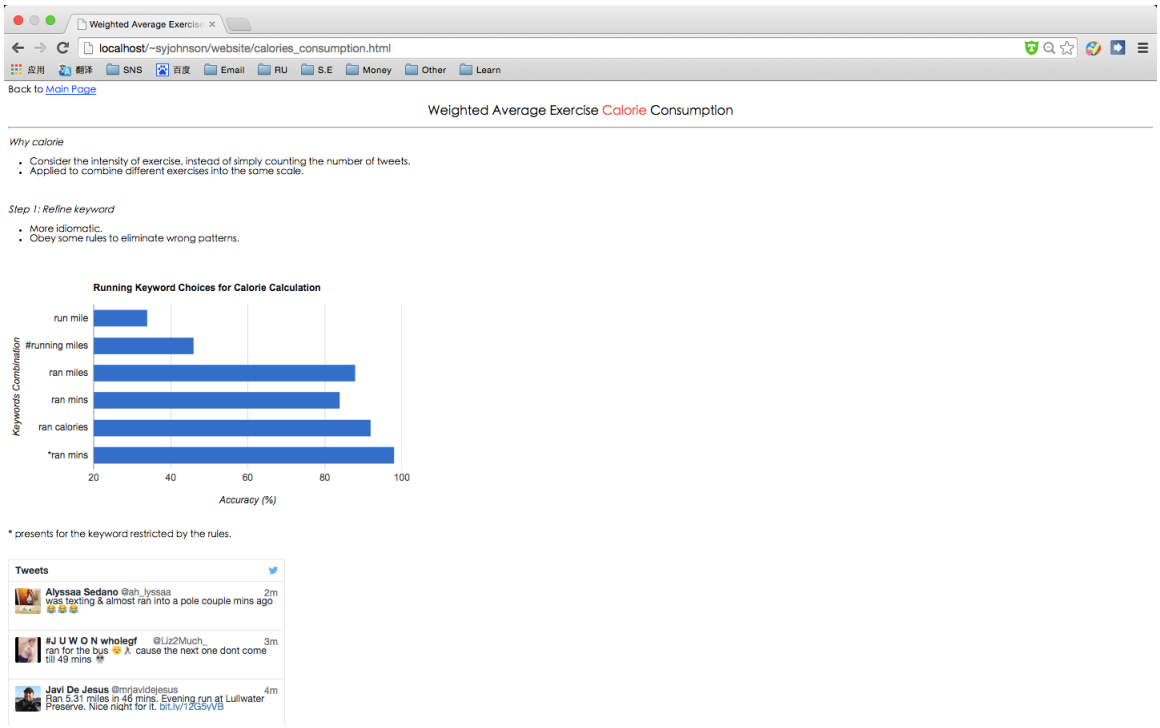


Figure 10-43: Calories Consumption-part1

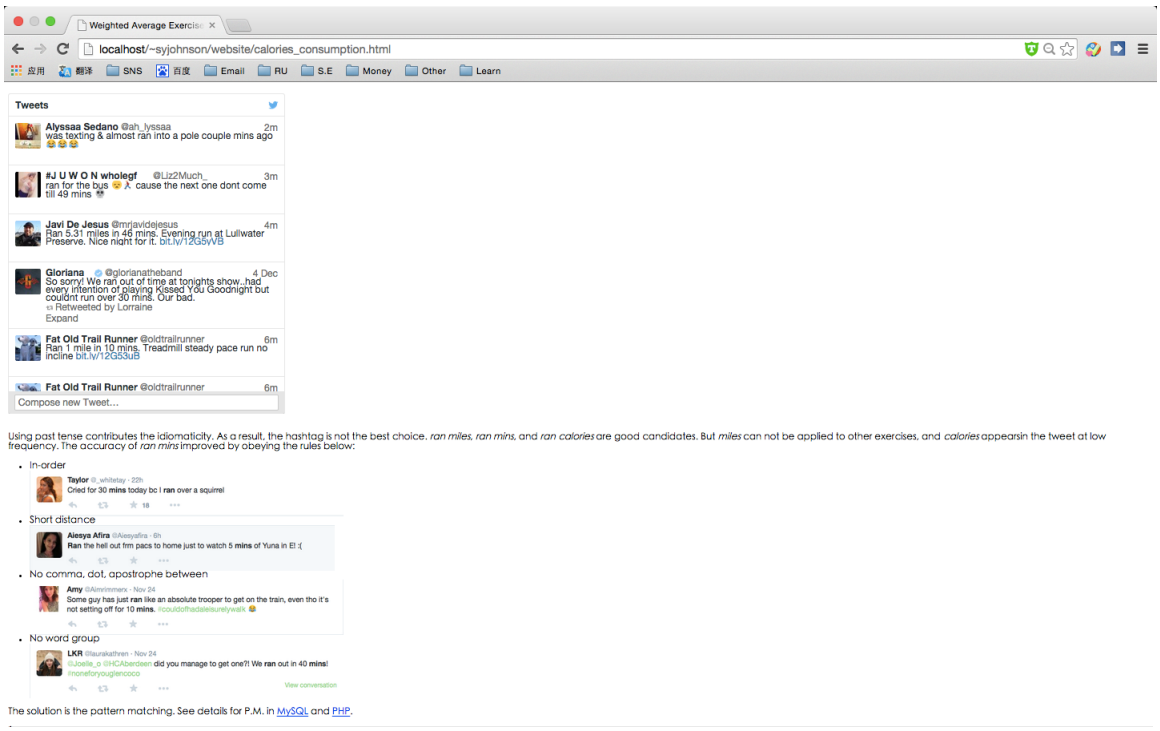


Figure 10-44: Calories Consumption-part2

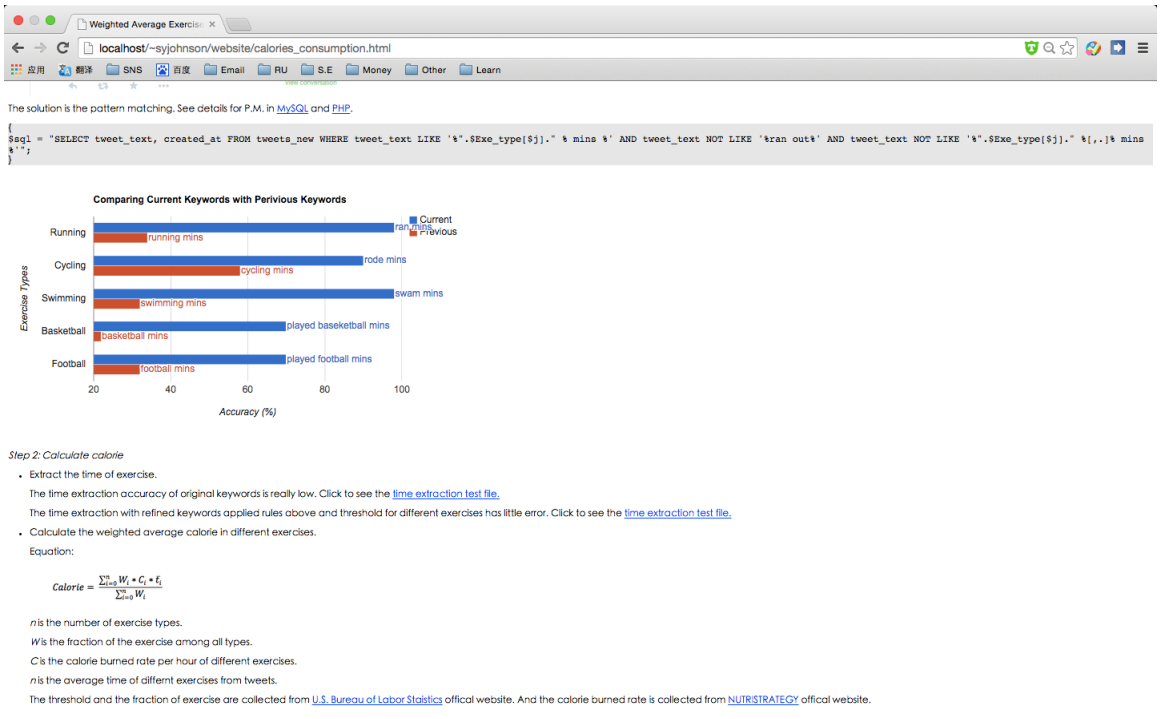


Figure 10-45: Calories Consumption-part3

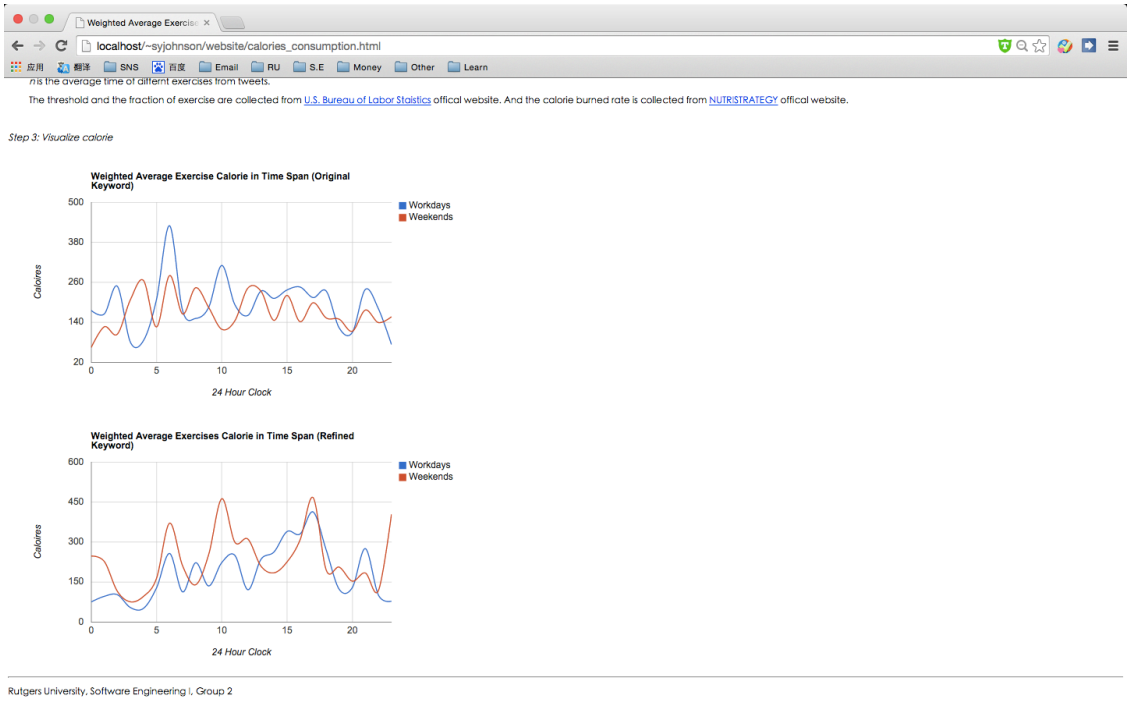


Figure 10-46: Calories Consumption-part4

In the Figure 10-43 to 10-46, when clicking on the calories consumption button, you may see the Weight Average Exercise Calories Consumption.

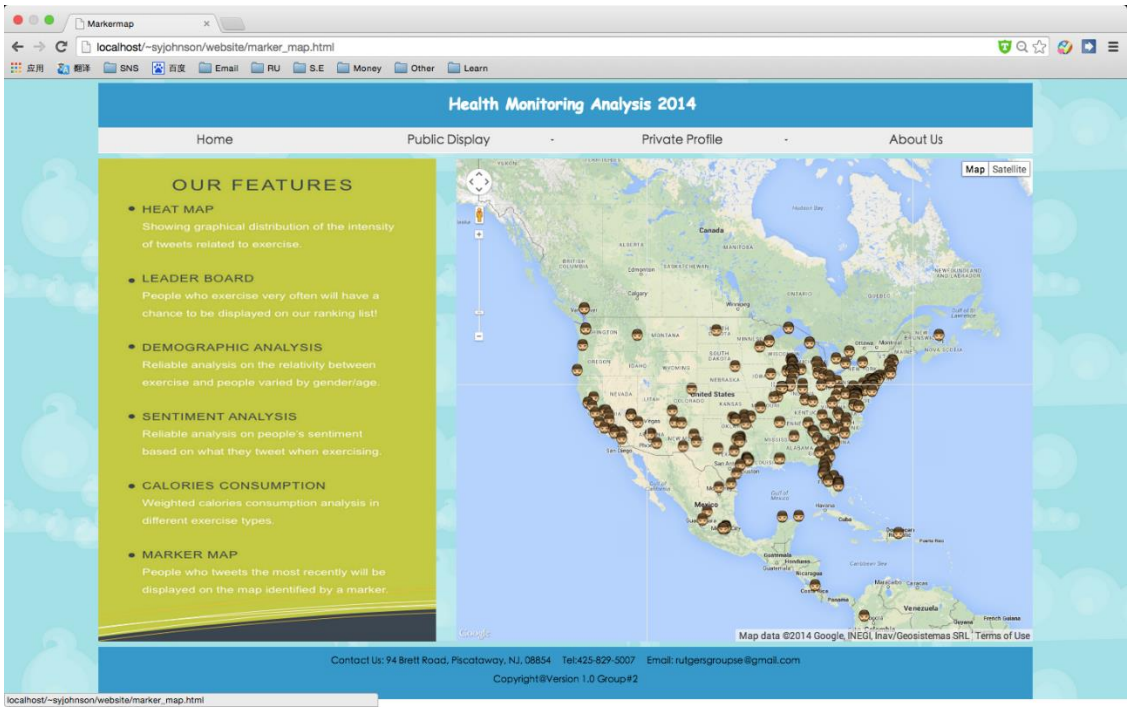
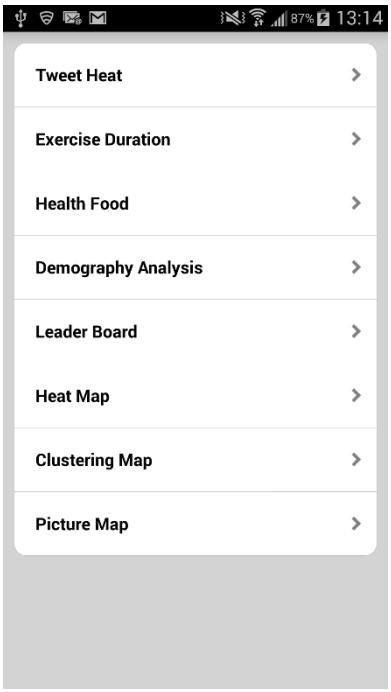


Figure 10-47: Marker Map

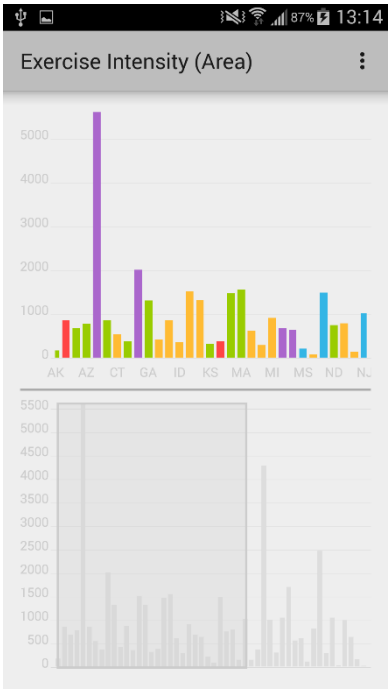
In the Figure 10-47, when clicking on the Marker Map button, you may see markers on the map for users who have tweeted a tweet concerning exercise and mentioned his exercise duration time. Once the marker is clicked, the interface will pop out the duration time and the exercise type on the top of the marker.

10.3. Android Application



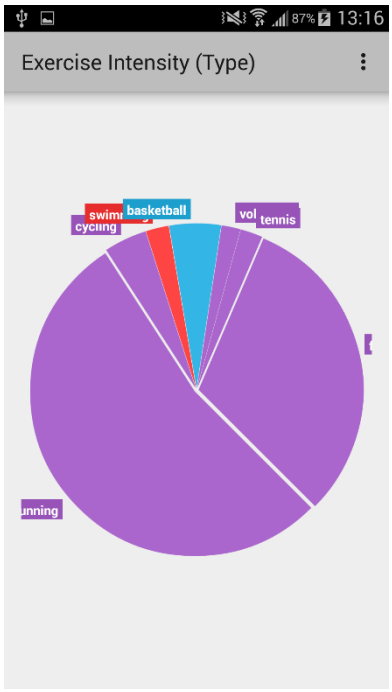
In Figure 10-48, this is the main menu of our Android application, which contains a list view where each cell represents a feature of our project.

Figure 10-48. Main menu



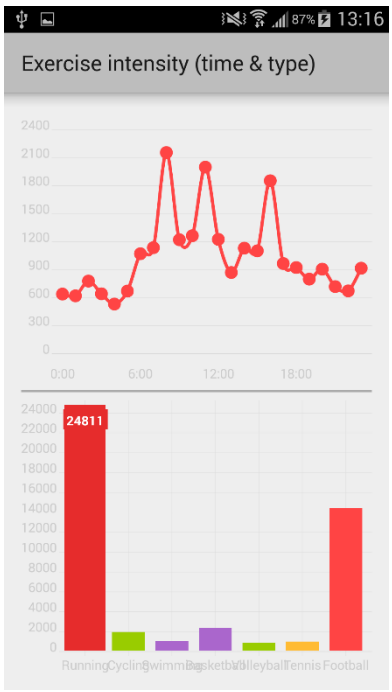
In the Figure 10-49, this interface shows the bar chart of the exercise intensity of all states in the United States. You can zoom in/out to see the detail in each state.

Figure 10-49. Exercise Intensity (Area)



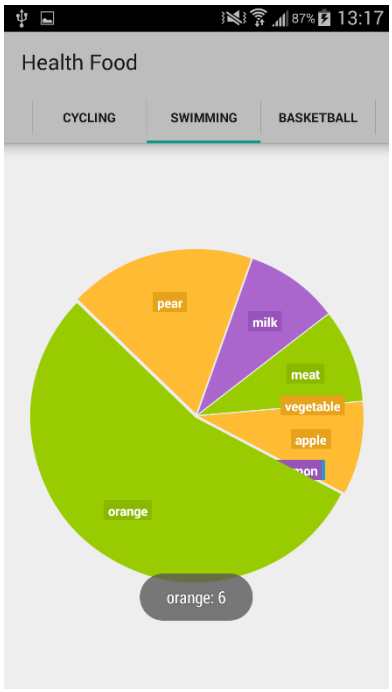
In Figure 10-50, this interface shows the pie chart of the exercise intensity of different exercise types.

Figure 10-50. Exercise Intensity (Type)



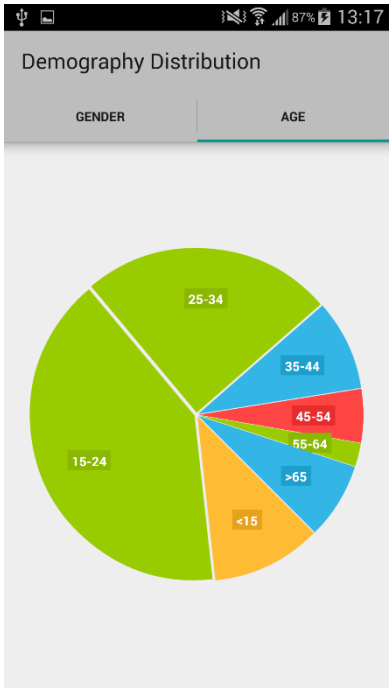
In Figure 10-51, this interface shows a line chart and a bar chart. The bar chart shows the total number of tweets related to each exercise type. By touching on each bar, the line chart will show the intensity tendency of the certain type in the time of day.

Figure 10-51. Exercise Intensity (time & time)



In Figure 10-52, this interface shows the pie charts of food distribution for different exercise types. By swiping the screen, you can change the exercise type.

Figure 10-52. Health Food



In Figure 10-53, this interface shows the pie charts for age distribution and gender distribution. By swiping the screen, you can change the demography type to see details.

Figure 10-53. Demography Distribution

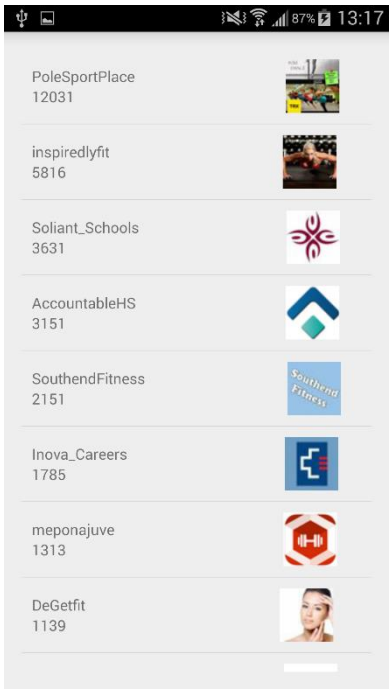


Figure 10-54. Leader Board

In Figure 10-54, this interface shows the leader board ranking of all twitter users, ranked by the number of exercise-related tweets.

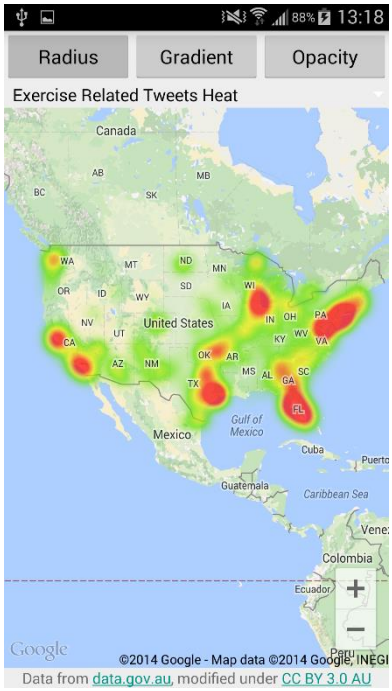
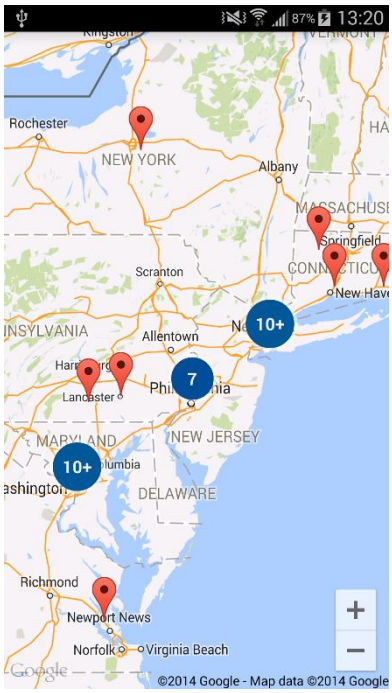


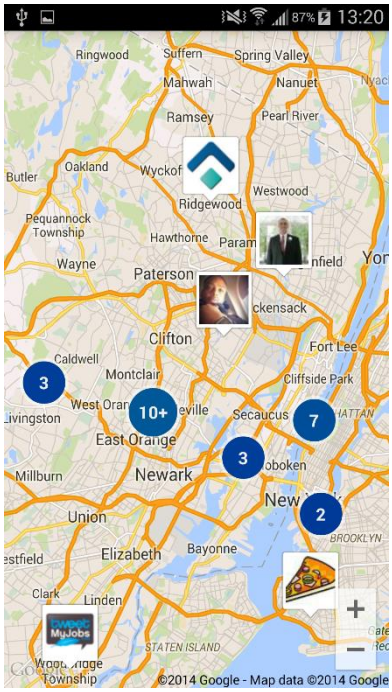
Figure 10-55. Heat Map

In Figure 10-55, this interface shows the heat map for the exercise-related tweets in the United States. You can also change the radius, gradient, and opacity of the radio point.



In Figure 10-56, this interface shows the clustering marker map for the exercise-related tweets. When zooming in, the individual markers show on the map. When zooming out, the markers gather together into clusters with certain numbers on the top.

Figure 10-56. Clustering Marker Map



In Figure 10-57, this interface shows the marker map for the exercise-related tweets with user profile images. By tapping on each picture, the user's screen name will be shown on the top of the picture.

Figure 10-57. Picture Marker Map

11. Design of Tests

11.1. Overall Description

The design of tests aims at testing the basic function of our system. The test will be separated into two parts: the function unit test and the integrating system test. Because some function units use the same coding methodology, therefore we group them as a class and choose one unit to test. The test table is shown in table 6-1.

Function unit group	**Unit within group**	**Test unit**
Twitter data acquisition	Twitter retrieve	Twitter retrieve
Data Base setup	Database	Database
Exercise duration	Duration in different states Duration in different exercise	Duration in different states
Ranking	Leader board in different exercise Leader board in different states	Leader board in different area
Demography	Exercise demography distribution	Exercise demography distribution
Google map display	Heat map State map Marker map	Heat map

Table 11-1. The Test Table

11.2. Functional Unit Tests

The test table show from table 11-2 to table 11-7.

11.2.1. Test unit: twitter retrieve

States	States Description	Input Requirement	Out Put Expected	Comment
Valid	Get Related JOSN File Successfully	Key words related to health and exercise topic	Twitter data related to health and exercise topic	This test is to make sure that whether the certain keywords can get useful tweets, and can return JSON files
Invalid	Get Unrelated JOSN File Successfully	Key words have no relationship with health and exercise topic	Twitter data have no relationship with health and exercise topic	

Table 11-2. Twitter Retrieve Test

11.2.2. Test unit: data base setup

States	States Description	Input Requirement	Out Put Expected	Comment
Valid	Data Base Set Up With Specific Data	JOSN File Extract From Twitter	Tweet Text, User Profile, Geo Information	This test is to make sure that our database will contain the exact data extract from twitter
Invalid	Data Base Set Up With Other Data	Any data	Any Data	

Table 11-3. Database Setup Test

11.2.3. Test unit: exercise duration in different states

States	States Description	Input Requirement	Out Put Expected	Comment
Valid	Table will contain number of max, min, average exercise time in different states	User's exercise time	Max, min, average exercise time in different states	This test is to make sure that

Invalid	Table will contain no information about max, min, average exercise time in different states	Any data but no User's exercise time information	No exercise time information	exercise duration can be calculated correctly

Table 11-4. Exercise Duration Test

11.2.4. Test unit: leader board in different area

States	**States Description**	**Input Requirement**	**Out Put Expected**	**Comment**
Valid	Table will contain top 15 user who send most tweet text in specific area	More than fifteen user send Tweet text.	User profile and their total send tweet text number	This test is to make sure that number of Tweet text can be calculated correctly
Valid	If user number are less than 15 in an area, table will contain all their information but the rest will be set name to none and it's information will all set to zero	Less than fifteen user send Tweet text.	Valid user with their total send tweet text number, fake user is named none and information is zero	
Invalid	Table will contain no user information	Any data but not Tweet text	No tweet text count information	

Table 11-5. Leader Board Test

11.2.5. Test unit: exercise demography distribution

States	States Description	Input Requirement	Out Put Expected	Comment
Valid	Table will contain percentage of male and female who exercise in certain area.	specific amount of male and female users information	The percentage of male and female users.	This test is to make sure that the percentage of male and female user will be calculated correctly.
Valid	Table will contain no information about male	Only female users information	Female users are 100%	
Valid	Table will contain no information about female	Only male users information	Male users are 100%	

Table 11-6. Demography Distribution Test

Note that the gender information is not contain in Twitter database, so we can only guess the gender based on their tweet texts.

To test the gender prediction accuracy, we choose some celebrity who own a Twitter account and we have already known their gender. Then we use the guess method to see the prediction result. If the accuracy is greater than 70%, we treat this method as feasible.

11.2.6. Test unit: heat map display

States	States Description	Input Requirement	Out Put Expected	Comment
Valid	The color of the map will range from green to red according to amount of tweet text in different area	Amount of tweet texts in different area	Area with more tweet texts will become redder.	This test is to make sure that heat map will show correct color which reflect the amount of tweet texts in a certain area.
Invalid	The color of the map will remain green if no tweet texts input	No tweet texts input into any area.	Heat map will be green	

Table 11-7. Map Display Test

11.3. Integrating tests

To test the integration feature of our system, it is essential for us to focus on the jsonsender.php. Because this file takes all responsibilities to enable the communication between the database and the front-end. Once we make sure it works successfully, the integration test is finished.

States	**States Description**	**Input Requirement**	**Out Put Expected**	**Comment**
Success	The jsonsender can communicate with the website and the database	Database table with user's information	Data can be request.	This test is to show who is asking for the data and what data is being asking
Failure	The jsonsender could not communicate with the website and the database	Database table with user's information	Data could not be request	

Table 11-8. Integrating Test

12. History of Work, Current Status, and Future Work

Data Collecting (Server)	**14-09-12**	**14-11-11**
Investigate Twitter API	14-09-12	14-09-15
Investigate the database codes from the former groups	14-09-14	14-09-15
Re-establish the database from the former group 1 and refine the keywords	14-09-16	14-09-16
Discontinuously download the data by Twitter streaming API	14-09-17	14-09-22
Implement Twitter rest API	14-09-25	14-09-26
Investigate Facebook API, Google+ API and text analytics APIs for demography information	14-09-29	14-10-10
Figure out gender, age and type by text analytics API	14-10-11	14-10-16
Download a full week data	14-10-13	14-10-20
Data Analyzing (Database)	**14-09-16**	**14-11-11**
Investigate the features in reports from former groups	14-09-16	14-09-22
Tweet heat > geographical distribution	14-09-23	14-10-09
Tweet heat > variation tendency	14-09-23	14-10-09
Tweet heat > exercising classification	14-09-23	14-10-09
Tweet heat > user ranking	14-09-23	14-10-09
Exercising duration	14-09-23	14-10-16
Personal Diagnosis	14-10-10	14-10-16
Personal diagnosis supplement	14-10-16	14-10-30
Topic correlation	14-10-23	14-10-30
Exercising frequency	14-10-21	14-10-30
Tweet sentiment	14-10-16	14-10-30
Demography	14-10-16	14-10-30
Data Displaying (IOS & Web)	**14-09-16**	**14-11-11**
Set up the communication between front-end and rear-end	14-10-07	14-10-09
Structure the IOS app	14-09-16	14-10-06
Implement the IOS app	14-10-10	14-10-16
Investigate map graphical API	14-09-16	14-10-09
Implement marker, heat and state maps	14-10-10	14-10-16
Implement all features from former groups	14-10-09	14-10-09
Display improves compared to former groups	14-10-16	14-10-16
Refine the IOS app	14-10-16	14-10-30

Table 12-1. Work Stage 1

Data Collecting (Server)	**14-11-11**	**14-12-01**
Adjust the keywords and improve the method for retrieving tweets data	14-11-11	14-11-15
Download another full week data by the new method and keywords	14-11-15	14-11-22
Estimate relevancy of tweets by new keywords	14-11-22	14-12-01
Data Analyzing (Database)	**14-11-11**	**14-12-01**
Word frequency	14-11-11	14-11-18
Healthy food	14-11-11	14-11-18
Weighted average exercise calorie consumption	14-11-11	14-11-25
Part of speech	14-11-11	14-11-28
Improve the accuracy of other features	14-11-18	14-12-01
Data Displaying (Android & Web)	**14-11-11**	**14-12-05**
Structure the website	14-11-11	14-11-15
Implement the website	14-11-15	14-11-20
Refine the website	14-11-21	14-12-05
Implement the displaying on Android	14-11-11	14-12-05

Table 12-2. Work Stage 2

Before demo1, we first set up the database and collect tweets by Twitter API. Important features were the number of tweets in geography, time, exercise type, and demography distribution, topic correlation, exercise frequency and time, tweet sentiment, and personal diagnosis. The platform was IOS app. After demo1, we found the relevancy of tweets in our database was unsatisfied and the measurement was simple. Thus, we refined our keywords to improve the relevancy as mentioned in the algorithm, and presented weighted average exercise calorie consumption, word frequency, part of speech algorithms. Besides, website and Android platform had been developed.

As for the future work, the refined keywords are only implemented for the calorie algorithm, thus other features should also use the new tweets to analyze. Although we get the results of each feature, but we have not a way measure the accuracy of them, thus probably we need to find some official statistics to compare. Finally, the features are too much and messy, it would be better to re-classify the features.

13. Project Management

Conference date and location:

Our team holds conferences twice a week on Tuesday and Thursday at the study room 1 in the Library of Science and Medicine.

13.1 Project basic structure work

Each project basic structure work is distributed to one or more team members. However, we do not suggest each team member take part into the whole basic structure work at the very beginning. The basic structure works are mostly fixed and low-layered, and they need little changed when develop the high-layered works – features development. This arrangement contributes to improve the basic structure setting up efficiency, and diminish the reproduce and conflict when several team members do the same work. Although each team member will only do some parts of the structure work, they need to know the whole picture. Thus each team member would report his work to everyone at the conference. Besides, as pushing the work forward, the team members in different basic structure parts need to communicate with each other for the work integration. So actually, each team member realizes the whole picture of our work in the end.

Assignments	D. Yao	Z. Zheng	W. Zhang	Y. Wu	W. Fang	Y. Sun
Data Collecting	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Data Storing & Rearrangement	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Website Display	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
IOS Display	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Android	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%
Management & Integration	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%

Table 13-1 Basic structure

In the Table 13-1, the target of data collecting is to get the public information, e.g. the users' non-private information and tweets from Twitter, and users' private information from a demography speculation API. The team members in the part also

need to schedule the date to download data, and export and import the database structure and data to other team members. Data storing and rearrangement is the design of the tables in the database. The team members in website and IOS display need to design the UI and display clear charts and maps to the viewer. The work for management is to collect creative ideas and breakdown works to each team member. The work for integration is to combine the work from each team member. The uniform rules of work are needed before the work distribution. Otherwise a lot of renaming works will appear later. The integration work also include writing communication files for JSON sending and receiving between the front-end work and the rear-end work.

References

[1] HCA: http://en.wikipedia.org/wiki/Health_care_analytics
[2] HMA 2014 project description: http://www.tru-it.rutgers.edu/takmac/
[3] S. Fox, M. Duggan. *Health online 2013*. Jan. 15, 2013.
http://www.pewinternet.org/2013/01/15/health-online-2013/
[4] S. Bennett. *Facebook, Twitter, Instagram, Pinterest, Vine, Snapchat – Social Media Stats 2014*. June 9, 2014.
http://www.mediabistro.com/alltwitter/social-media-statistics-2014_b57746
[5] Phirehouse: https://github.com/fennb/phirehose
[6] S. Kumar, F. Morstatter, H. Liu. *Twitter Data Analytics*. Aug. 19, 2013.
[7] JSON: http://en.wikipedia.org/wiki/JSON
[8] Twitter API: https://dev.twitter.com/overview/documentation
[9] Rest API: https://github.com/abraham/twitteroauth
[10] Demographic API: http://textalytics.com/core/userdemographics-info
[11] PNchart: https://github.com/kevinzhow/PNChart
[12] Google API: https://developers.google.com/apis-explorer/#p/
[13] tf-idf: http://en.wikipedia.org/wiki/Tf–idf
[14] *The least square method and accuracy analysis*:
http://wenku.baidu.com/link?url=gwzOkBt_AQSSeWbnytidM9qEQT007jsxqC9Uqpt7B5qSUPBZicnFyQGs2LRFwPrr8zvaR0PmA9nR0uOVKUvpj8s60PDma8mYILzbK617wyq
[15] Pearson product-moment correlation coefficient:
http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient
[16] *Healey & Ramaswamy. Visualizing Twitter Sentiment*:
http://www.csc.ncsu.edu/faculty/healey/tweet_viz/
[17] Margaret M. Bradley and Peter J. Lang. *Affective Norms for English Words (ANEW)*.
[18] Probability density function:
http://en.wikipedia.org/wiki/Probability_density_function
[19] Hash table:
http://en.wikipedia.org/wiki/Hash_table
[20] Pattern Matching in MySQL:
http://dev.mysql.com/doc/refman/5.0/en/pattern-matching.html
[21] Pattern Matching in PHP: http://php.net/manual/en/function.preg-match.php
[22] Exercise Statistics from Government:
http://www.bls.gov/spotlight/2008/sports/
[23] Calorie burned rate per hour:
http://www.nutristrategy.com/caloriesburnedrunning.html
[24] Part-of-speech tagging Wiki http://en.wikipedia.org/wiki/Part-of-speech_tagging

[25] A Fast and Accurate Dependency Parser using Neural Networks, Danqi Chen, Christopher D. Manning.