Workout with Friends
Health Monitoring for Fitness Applications

16:332:567 – Software Engineering
Report #3

Mateus Santos Abdul Hassan
Kevin Kobilinski Daihou Wang
Brien Range Sujana Gangadharbatla

https://workoutwithfriends.wikispaces.com/

Fall 2012

Table of Contents

1. Customer Statement of Requirements and Project Solution

1.1 Customer Statement of Requirements

Every January hundreds of thousands of people make it their legitimate goal to improve their overall fitness level. Although the desired targets can vary from shedding a few pounds to completing an Iron Men competition, according to fitness professionals, most people fall short from achieving their goals. Leriken Functional Fitness claims the top 10 reasons why people abandon their fitness routines are:

Lack of Motivation
Setting Unrealistic Goals
Not Having a Clear Plan
Doing Too Much Too Soon
Pain / Injury
No Perceived Results or Rewards
Not Having Any Fun
Lack of Time
Can't Afford the Gym Membership
Self-Conscious to Train in Front of Other People

For many, staying motivated is the key to being able to stick with the commitment to pursue greater fitness level. Then there comes the self-monitoring devices from all venders, to monitor one' s physical characters during the exercises. One can simply read out your realtime heart rate, blood pressure and other charaters from either a bracelet like or a watch like device.

However, what self-monitoring devices can give stays at a bunch of numbers showing how much time I have exercised at a time, or how much length I may have run this time or today according to its monitoring parameters. And we' re still looking for some application and platform to share our working out data, and to set up a more intuitive goal by completing with a friend, some friends, or other people we work out with. Not only this makes our exercises goal more intuitive, but also bring more fun when working out with friends.

I think there may be two possible way to do this:
1. build an application based on a social network, so people can easily share their work out result with his/her friend on the network,
2. build an application on mobile platform, like a phone or tablet, so we can directly form a data sharing or completation during exercising.

Social network application

As social networking has become an indespensable part of our everyday life, we' re getting used to share every piece of our life through the social network. So if we can also share our progress on working out with our friends, form competations with friends or even friend' s friend. It will be a great motivation to have people around you join the progress, and it would certainly brought more fun.

To fulfill the idea, this application should be able to download the work out data from self-monitoring devices. And based on this self-monitoring data, combined with the pre-input personal physical information, the applicaion should give a detailed analysis to your one time or periodical training, which could but not limited to overall calorie consumption, average calorie consumption, peak excerising time, etc. So one can have a brief review of the exercise for one time or a period of time.

Also, this social network based application should based on a popular social network, like Facebook, Twitter or Myspace, and friendly implanted in the social network, so it can directly use all the social informations available. It will be based on your alreay formed social network with your friends or your colleges, and it will always be more motivative to share or complete with people around you.

We also hope that we can use this application as an interface to invite a friend or friends to join a competation. Later maybe post the results of the competation with some photos during the work out online, in which we can make full use of the network.

Mobile platform application

Besides sharing the results of the work out, for more time we may want to form a competation when we' re exercising, which is a real on-going competation. There may be a lot of different competing criterions, like energy consumption, total amount of aerobics working, peak-level exercising time, etc. However no matter what criterion it is, it would definitely bring more motivation and fun to the exercise.

To fulfill this part of the requirement, an application should first of all be able to connected to the self-monitoring device, then be able to download self-monitoring data from the device. Then the application will store and analysis the data for later competation.

More inportant, to forming a competation, the application should be able to share and compare the data between the two users. Then based on certain competing criterion, comes the competing result.

To sum up, in order to exercise more motivated, we need some application or platform to share or compete or working results. And two really good way to implement them is to build application based on a social network where we can analysis and share the exercises resuls, or a mobile platform application we can use to form real time competation during the exercises.

1.2 Backgrounds on Self-monitoring

Current methods (devices, apps) for fitness enhancement

There are several consumer grade, high-end fitness devices with HRM (hear rate monitor) connectivity functionality. Pairing technologies include pairing via Bluetooth and ANT+ protocols. Currently the biggest players in the high-end personal fitness market include Motorola's MotoACTV and Nike's "Nike+ GPS SportsWatch". These devices have a high processing component that calculates a user's delta in GPS position in order to derive work effort data (pace, distance covered) in addition to the ability to connect to a HRM. These powerful devices are targeted for consumers predominantly with running and cycling applications in mind, where it is advantageous to the user to know real-time information such as: HR, calories burned, pace, distance covered, and time elapsed. In addition, with the increases in processing speed of the average consumer grade smart-phone, users now have the option to utilize free and low cost android and iOS applications such as: iMapMyRun and Encomodo Sports Tracker.

Figure 1-1: MotoACTV and Nike+ GPS SportsWatch

Although the current methods described above produce rich content for the user, there are some areas that leaves the user striving for more. For instance, challenging other users is limited to only the set of users who happen to use the same device. There is no cross-platform challenge. Also, device cost is a big drawback and android/iOS integration with wireless HRM is very limited at this point.

There are clear benefits for mobile implementation (direct pairing) with HRM. For one, it frees users from having to connect to a PC to upload workout data since mobile applications utilize the smart-phone cellular or wi-fi networks to initiate the sync process. The main benefit to the user, however, is that it eliminates the need for users to acquire expensive hardware to record their workout data. Typical costs are shown in table 1.

Device	Manufacturer	Price Range
MotoACTV 8GB	Motorola	~$150
Nike+ GPS SportsWatch and Nike+ Sensor	Nike	$169 + $29.99 = ~$200
Wireless HRM	Various (Motorola, Garmin, Polar)	~$50 - $70
iMapMyRun	MapMyFitness - iOS	Free
Encomodo Sports Tracker	Encomodo – Android	Free

Table 1-1: Fitness & Health Monitoring Devices / Apps Landscape

A mobile application for fitness enhancement via HRM pairing requires 1 piece of HW. That is, 75% lower cost based on values listed in table 1.

What is heart rate zones?

Heart Rate is the number of heartbeats per minute (bpm). Heart rate varies depending on your activity, from vigorous exercise to sleep. If you know and monitor your heart rate, you get maximum efficiency from your workouts. Maximum safe heart rates vary greatly in people based on age and gender and there are many formulas and tables available to calculate what those rates might be. Generally, zones 1-5 represent 50-59%, 60-70%, 71-79%, 80-90% and 91-100% of the target Max heart rate. For example, the data shown below represent the heart rate for a 25 year old male according to the equation: Max HR = 220 – Age → 220 – 25 = 195:

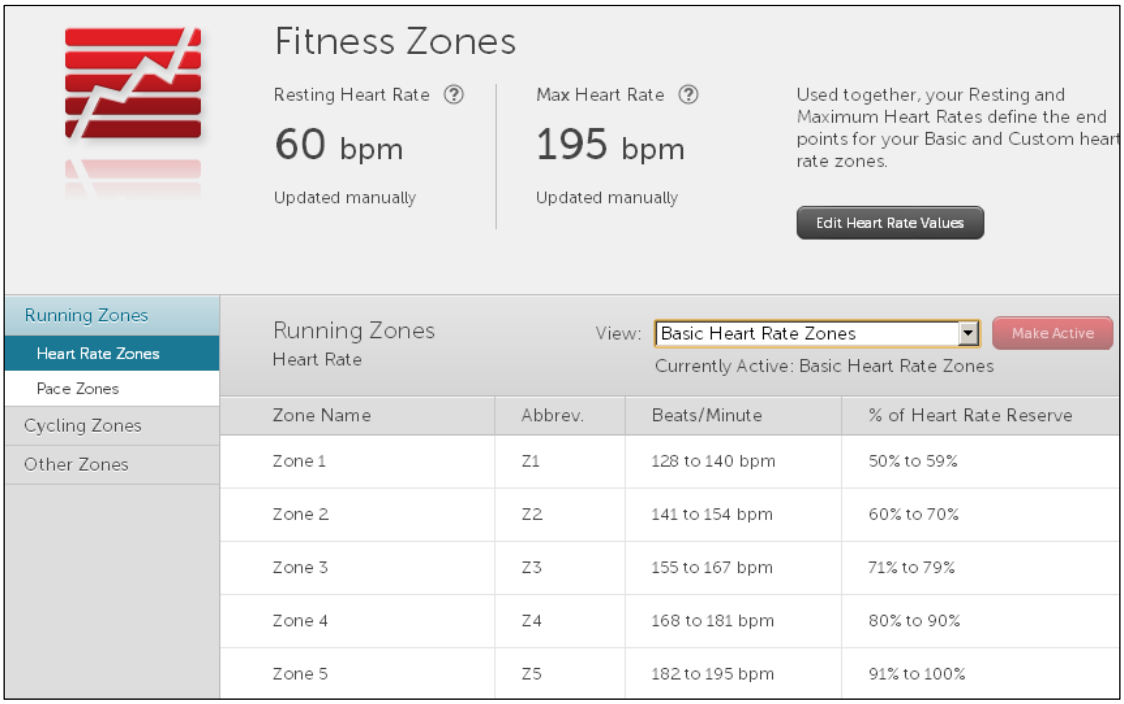


Figure 1-2: Heart Rate Zones for 25 year old Male

The concept behind the HR zones is to classify various stages of physical activity. The key differentiator between the HR zones is where the energy (calories burned) utilized by the body is coming from.

HR Zone	% of Max HR	Description
1	50-59	Warm-up zone, healthy HR (calories burned 85% from fat)
2	60-70	Fitness zone, optimal fat burning (calories burned 85% from fat)
3	71-79	Aerobic zone, endurance training (calories burned 50% from fat)
4	80-90	Anaerobic, performance training (calories burned 15% from fat)
5	91-100	Maximal effort, “red-line”

Table 1-2: HR Zones

Why does heart rate zones matters?

The biggest reason why this matters is because a person can tailor his or her training based on HR zones in order to achieve a particular benefit from working out such as: increased endurance, fitness, maximum sprint pace, performance capacity. It will also allow him or her to ensure that he or she is working out at a proper intensity level. Often when people just go by their ‘feel’, they may not be challenging their body as much as they could be. Knowing your real-time HR can be an indicator to increase intensity. The table shown below illustrates this concept:

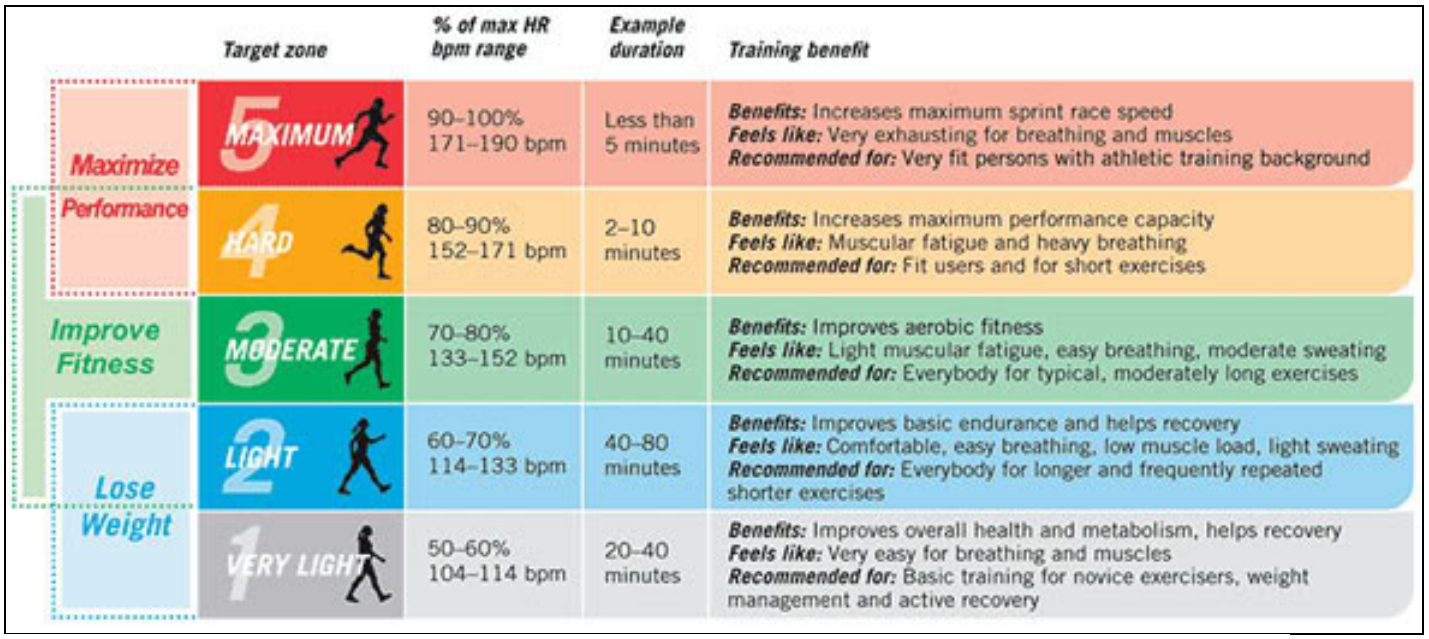


Figure 1-3: Benefits of Heart Rate Zone Training

If your goal is to burn fat as a fuel source during your workout session, you'll want to be working out at a lower overall heart rate because for your body to burn fat (optimally), you must be doing lower intensity exercise (since fat takes longer to break-down so the fuel will be delivered at a slower rate). The desired zones for this type of exercises are Zones 1 and 2. The energy system at play for low intense exercise is much different for high intensity exercise, so depending on your overall workout goals, you can adjust your heart rate to match.

If you're aiming to boost your fitness level on the other hand, then you'll want to be working out at a much higher heart rate, often at a percentage of your maximum heart rate such as 80-90%, which corresponds with Zone 4. It would be very difficult to tell if you're reaching these intensities without a heart rate monitor.

How can a heart rate monitor improve your workout performance?

Heart rate monitoring is a widely used training method by athletes of all levels. The very reason why even use a heart rate monitor is to help improve performance by ensuring that the user is working out at a sufficient intensity level to derive proper benefits. The simple answer is that HRM gives users a tool to record, measure and track performance. For example, a MotoACTV device equipped with a heart rate monitor records 1 sample per second. Below is an example of the data recorded for a ~43 minute bike workout (heart rate and elevation data is displayed):

Figure 1-4: Real data for a 43 minute workout with MotoACTV

Contrasting the workout data collected with a HRM over time should clearly show if more time is spent in Zones 4 and 5 during a workout.

The second reason why using a heart rate monitor is beneficial in increasing running or cycling performance is because it will provide a very good way to increase lactic acid threshold. Since a user is able to engage in more intense workouts over time by monitoring the heart rate data collected and adjusting their workout accordingly, a long term consequence is increased the lactic acid threshold by your body.

1.3 Project Overview

To fulfill the user requirements for the problem statement, Workout with Friends provides multiple platforms (web-based and mobile based) that allow users to challenge friends or themselves based on key biometric data. The service provided to the end user is reliable in determining a workout intensity score in addition to being accessible with ease by a large audience. Workout with Friends strives to address key limiting factors that drive most people to halt their workout activities: lack of motivation. By providing a user-friendly gaming platform, Workout with Friends entices friendly competition in order to motivate users to workout.

The basic requirement for any user to be able to play Workout with Friends is that biometric data must be recorded with a heart rate monitor for the duration of the proposed workout. Any user equipped with a HRM should be able to play Workout with Friends regardless of the device used to collect the required key biometric data (calories burned and heart rate zones). Workout with Friends' mobile platform for Android will address user's concern over cost meanwhile Workout with Friends'

Facebook Web platform will address user's accessibility concerns. For example: a user with a Nike+ GPS SportsWatch paired with a HRM should be able to challenge a user equipped with a MotoACTV paired with a HRM or a user simply equipped with a smart phone paired with a HRM.

As a secondary goal, Workout with Friends strives to make the workout experience a more enjoyable one by the end user through an engaging user interface that marries gaming with social aspects. To this end, communication with a user's friend is central to the design. By delivering Workout with Friends as an embedded Facebook App, users are easily able to login from a central location and challenge friends in the user's Facebook friend list in a straightforward fashion. Additionally, to create a compelling and engaging user experience, Workout with Friends introduces the ability for users share the outcome of the challenge on the user's Facebook wall. This is perhaps the most exciting feature that Workout with Friends introduces, which is ahead of the competition. The ability to post user stories on their friends or opponents Facebook wall adds an additional dimension to the competitiveness nature of the game through bragging rights.

1.4 Glossary of Key Terms

Resting Heart Rate – The heart rate, measured in beats per minute (bpm), measured when the subject is awake, but has not performed physical activity/ exerted themselves recently. The resting heart rate is an indicator of general health.

Max Heart Rate – The highest rate of heart beats per minute that is achieved during exercise with 100% exertion. Percentages of this rate are used to indicate exercise intensity.

Heart Rate Training – Monitoring heart rate data through the use of a measuring device and adjusting one's exercise accordingly to achieve specific fitness goals.

Hear Rate Zones – 5 ranges which are grouped as a percentage of maximum heart rate. Each zone has varying benefits to exercising within that heart rate zone, and a user can choose to exercise within a particular zone, based on their exercise goals.

Lactic Acid Threshold – Under normal exercise, the production and removal of blood lactate is equal. At this threshold of exercise intensity, there is an abrupt increase in the production of blood lactate

Player / User – A participant of the Workout with Friends application.

Challenge – A competitive workout between either two players or a player's own history. A winner is determined through a score comparison based on points assessed based on calories burned and time spent in heart zones (with highest weight placed in zone 4 and 5).

Device - The heart rate monitor used to collect heart rate data during exercise, and upload the results to the Workout with Friends application.

Administrator – The manager who has access to maintenance of the Workout with Friends system

Database – The storage of all relevant Workout with Friends data, including player profiles (name, age, gender, weight), player heart rate data and calories burned.

2. System Requirement Analysis

2.1 Function Requirements Table

Table 2-1: Function Requirement Table

ID	Priority Weight	Requirement
REQ-1a	5	The system shall be able to keep a database for all user data.
REQ-1b	5	The system shall have access to all user data for futher comparison.
REQ-2	5	The system shall calculate a workout intensity score that correlates to a workout data set
REQ-3	5	The system shall be able to execute a comparison between two workout intensity scores to determine a winner
REQ-4	5	The system shall be able to allow users to initiate a challenge against another registered user.
REQ-5	5	The system shall allow user to accept a challenge from another registered user.
REQ-6	5	For the web platform, the user shall be allowed to submit workout data
REQ-7	5	For the mobile platform, the system shall be able to establish wireless connectivity with a heart rate monitor prior to start of a challenge
REQ-8	5	For the mobile platform, the system shall display the user's real-time HR at the top of notification bar while the application runs in the background for the duration of the challenge.
REQ-9	5	For the mobile platform, the system shall run a continuous filter for the duration of the challenge to detect if the user's heart rate surpasses the user's max heart rate by more than 15%. In the event that such criteria is met, the system shall send a txt msg notification alert to the user's designated emergency contact.
REQ-10	5	For the mobile platform, the system shall collect data from the wireless heart rate monitor and store it locally in the smart phone. Only once a workout is completed the system shall synchronize the data with a database automatically
REQ-11	4	The system shall allow users to be able to view his or her’s overall record against previous opponents.
REQ-12	2	The system should allow users to communicate with oponents via in-challenge messaging.
REQ-13	2	The system should allow the user to register an account with their Facebook profile.
REQ-14	1	The system should allow the users to post the outcome of a challenge on his Facebook wall as well as the opponents Facebook wall (if permission is granted)
REQ-15	1	The system should allow users to search for a friend (opponent).
REQ-16	1	The system should allow the users to send an invite an unregistered friend.
REQ-17	1	The system should allow users to be able to set up a challange against a random opponent with similar ranking.
REQ-18	1	The administrator should be able to access user account data.
REQ-19	1	The administrator should be able to retire a user’s account.

2.2 Non-functional Requirements Table

Table 2-2: Non-functional Requirement Table

Non-functional Requirements		
ID	**Priority Weight**	**Requirement**
REQ-20	5	All user data shall be stored in the system's database. No user information should be stored on the user's device. Users shall not be able to directly modify any data. There must be two copies of every record in case of system failure.
REQ-21	3	The system should require minimum maintenance, and of at most once per week.
REQ-22	3	The system should be able to maintain function in the event of any changes to Facebook's API.
REQ-23	2	The system shall be simple and easy to use. Data should be presented to the user in such a way that the user's focus is automatically drawn to it when the users view the page.
REQ-24	2	The system content should be centered in the center of the page. The user should not need to scroll down to view the data or access the majority of the options on the page.
REQ-25	2	The user must be able to set up a challenge in 2 clicks or less.

2.3 On-Screen Appearance Requirements

Because of the intention to integrate the game with Facebook as an app, the application must adhere to all policies as outlined on Facebook's Policy Section – Features and Functionality, found at https://developers.facebook.com/policy?_fb_noscript=1. As it relates to Facebook integration, there shall be an explicit "Log Out" button for both the application and Facebook (6). There shall be no Facebook icons nor shall there be any Facebook derivative icons (8). Any non-Facebook content displayed by the application shall explicitly state that it is not Facebook content. The applications shall not obscure any Facebook elements. Although the implementation may use pop-ups to alert the user to new messages, these messages shall not obscure and Facebook-required functionalities.

Having said that, the following are the major On-Screen Requirements:

1. Landing page: on Facebook Market App. When users search for apps, the results takes to landing page. Landing page gives users the option to look at game description, info, screen shots, supportability, permissions, profile and cover pictures. It gives user the option to play the game via an embedded app or send to mobile.

2.e – Login with Facebook account: takes users to registration page, sets up single-sign-in services from facebook to take users automatically to home screen next time.

2.d – Registration: for new and 1st time users (some info should also be available in About Screen)
- **2.d.1** - part 1 - game description and how-to-play
- **2.d.2** - part 2 - disclaimers & permissions

2.d.3 - part 3 - user info (name, user name, gender, age, resting heart rate, weight, sports preferences, share stores (y/n), share location (y/n), alerts..)

2.a - Home Screen major building blocks:
1) high level status bar: (number of challenges: completed, waiting for opponent, won; weekly points, delta of wk-to-wk) - button takes users to "Performance Tracker screen" (2.j)
2) About
3) Start a new challenge
4) Open challenges
5) Challenge History
6) Help

2.b - Start a new challenge screen

2.b.1 - Search for facebook friends, random local opponent, search by username, select previous workout (self-trainer mode)

2.b.2 - (from select facebook friends) - populates list of FB friends, allow to select 1 or more opponents

2.b.3 - Pre-challenge screen: shows selected friends, asks users to input: challenge nickname, workout duration, options (web or mobile), message opponents

2.h - Game Screen

2.h.1 - Web Game Screen - Directions: please make sure to wear a HRM device to capture calories, HR and HR zone data. Empty spaces for user to input these 3 required fields

2.h.2.1 - Mobile Game Screen - Directions: connect HRM (wireless), warmup options (timer, or start data collection when HR passes warmup threshold (40% of max target HR)), and manual start. Program goes to background once data collection is initiated, displayed in top android notification bar: instant HR, time elapsed, time remaining,

2.h.2.2 - Game Results screen - from 2.h.1 and 2.h.2.1, shows calculated points for calories, HRZ 5, HRZ 4. (opponent results may not be known yet). clicking "next" uploads data to database

2.c - Open Challenge: function residing in Home Screen, lists open challenges waiting for opponents turn

2.f – Help: FAQ's, report user, send feedback to dev team, upgrade SW.

2.g – About: users can find more info (links) on HR, training and tips. Users can also access how-to-play and game description available at registration, Help (takes users to help screen) and Account (takes users to Account Management screen)

2.i – **Account**: users can review and/or edit account attributes such as: nickname, weight, resting HR, alerts, sharing options, age

2.j - **Performance tracker**: shows numerical values for challenges completed, winning %, margin won/lost, calories counter (toggle monthly vs weekly vs all-time view). Also users may select to graph such attributes.

The following sketches(Figure 2-1, Figure 2-2) illustrates these concepts.

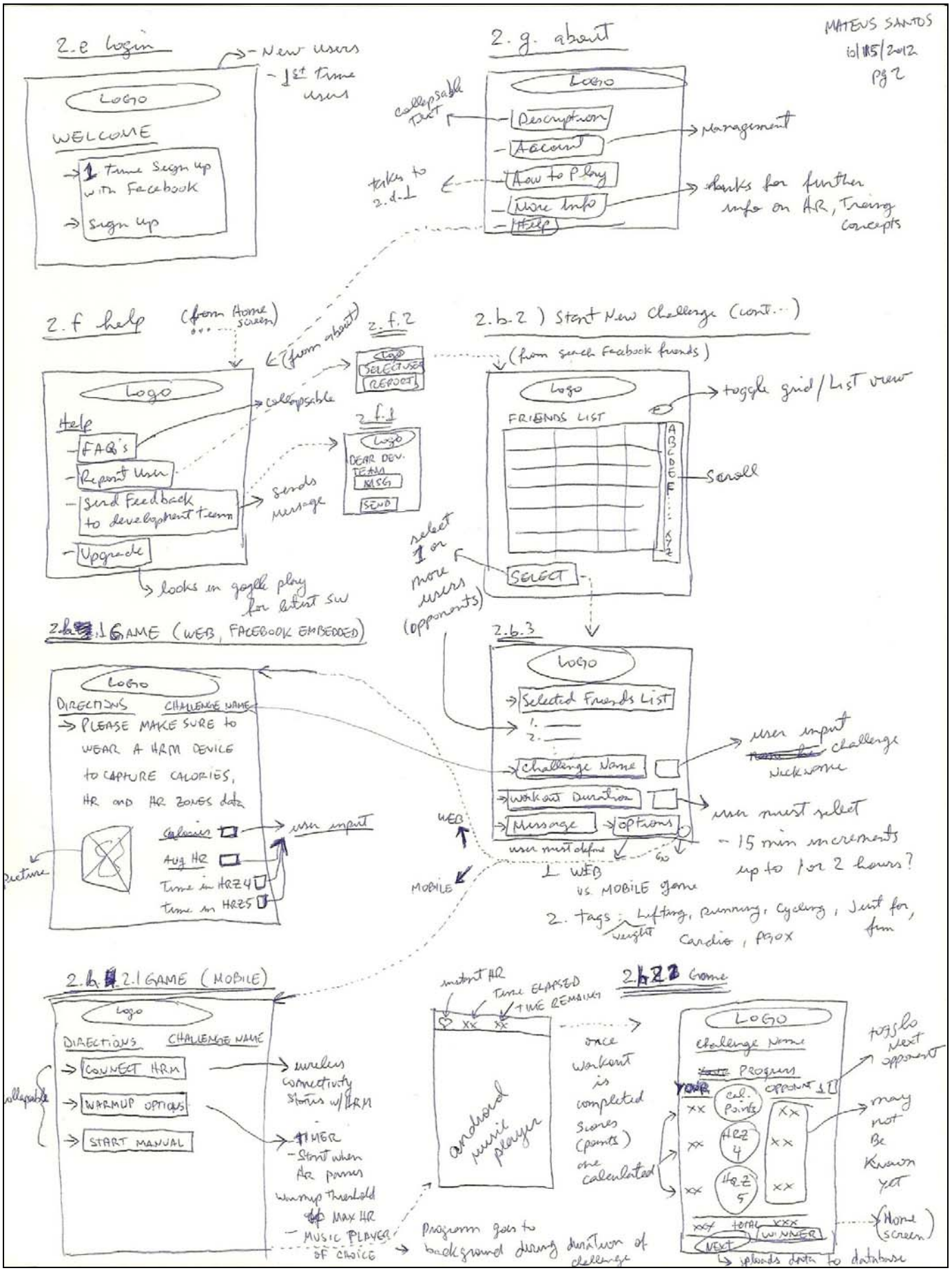


Figure 2-1: Sketch 1 for user interface

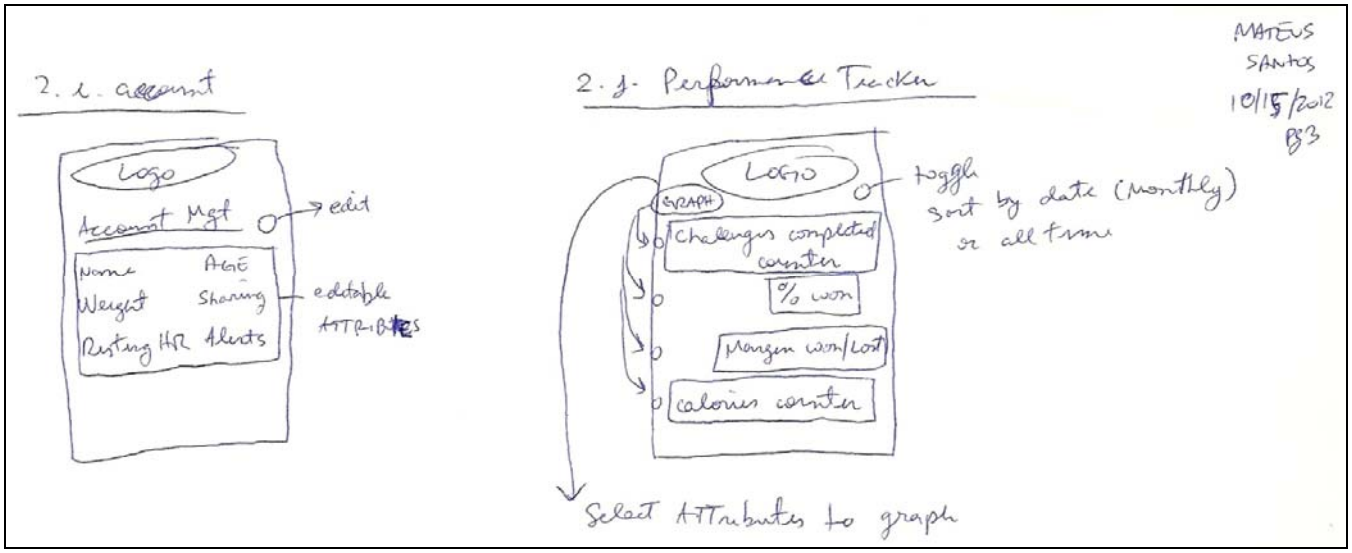


Figure 2-2: Sketch 2 for user interface

3. Functional Requirements Specification

3.1 Stakeholders

There are several primary end-users for fitness based social gaming. First, there are the fitness oriented individuals (and their partners) who exercise daily and who may use heart rate training to achieve their personal goals. This category includes athletes of all levels and people who are just old enough to use their local weight-lifting gym and treadmills, but also kids and adults of many ages. The secondary set of end-users are the people who aim to lose weight through exercising or who will use social gaming to stay motivated to achieve their workout goals. Given the low upfront cost to use and play the game.

Other possible proponents of this game may also include fitness experts, coaches and local gyms that may be use the game as a tool to teach healthy training principles via HR monitoring.

As the game matures, further specialization can be tailored to certain sports type such as: marathon (running), cycling and weight-lifting; in order to address certain groups of fitness

3.2 Actors and Goals

a. Player- a registered user of the software;
b. Opponent- a special case of Play actor, defined relative to the Player who initiated the given use case; this actor can do everything as Player, however, it should be distinguished from the initiating Player to be able to describe the sequence of interactions in use case scenarios;
c. Vistor- an unregistered user;
d. Database- records of all the Player' s performance;
e. Facebook- provider for all social information of Players and Visitors;
f. Administrator- a special user of the software who have top priority access to the software database.

3.3 Use Cases

3.3.1 Casual Description

UC-1: MonitorExercise- Allows the Player to get the self monitoring data analyzed.
Derived from REQ-1a, REQ-1b.

UC-2: ChallengeFriend- Allows the Player to challenge an Opponent to play a competition based on their analyzed monitoring data. Derived from REQ-4.
Initiating Actor: Player
Actor's Goal: To create a new game with a friend
Participating Actors: Challenged Player, Database

UC-3: ChallengeSelf- Allows the Player to challenge his previous workout performance.

UC-4: ViewRecord- Allow the Player to view the ranking of his/ her result against other local users. Derived from REQ-11.

UC-5: SendMessage- Allow the Player to send a message to the Opponent while a challenge session is open. Derived from REQ-12.

UC-6: Register- Allow a Visitor to register an account for the software with his/ her

Facebook profile.
Derived from REQ-13.
Initiating Actor: Player
Actor's Goal: To create an account
Participating Actors: Database, Facebook
Precondition: The system must support account creation.
Postcondition: A new account is in created for the user. This account will store player information and game history.
Flow of Events for Main Success Scenario:
1 → The player navigates to the application Facebook's website and clicks "Get App"
2 ← Facebook displays page asking if the player will allow the app to access information.
3 → The user clicks "Allow".
4 ← Facebook authenticates the user.
5 ← The system signals to the database to create a new account with the above information.
6 ← The database creates the user account and signals to the system that the account was created.
7 ← The system signals to the user that an account has been created.
8 → The user proceeds to registration screen (2.d.1)

UC-7: PostReult- Allow a Player to post the result of a challenge on his Facebook wall as well as the Opponent' s Facebook wall.
Derived from REQ-14.

UC-8: SearchFriend- Allow a Player search for a friend in the registered users.
Derived from REQ-15.

UC-9: **InviteFriend**- Allow a Player to send an invitation to an unregistered friend on Facebook.
Derived from REQ-16.

UC-10: ChallengeRandom- Allow a Player to challenge a random Opponent with a similar ranking from the registered users. Derived from REQ-17.

UC-11: AccessUserAccount- Allow Administrator to access a registered user' s account. Derived from REQ-18.

UC-12: DeleteUserAccount – Allow Administrator to de-register a user and delete the user' s account. Derived from REQ-19.

3.3.2 System Sequence Diagrams

The use case diagram is shown in Figure 3-1. Player, Opponent, Visitor and Administrator <<initiate>> all use cases, except for UC-2 (ChallengeFriend), UC-3 (ChallengeOneself) and UC-10 (ChallengeRandom), which are <<extend>> from UC-1 (MonitorExercise) as sub-use-cases. Database and Facebook store monitoring data and social relation data for Player, Opponent and Visitor, so they are <<participate>> in all use cases. Opponent is generated from a Player when the Player chose to run UC-2 (ChallengeFriend).

3.3.3 Traceability Matrix

Use cases are designed to meet the system requirements, the traceability matrix in Table 3-1 shows the mapping relation between system requirements and use cases of this software.

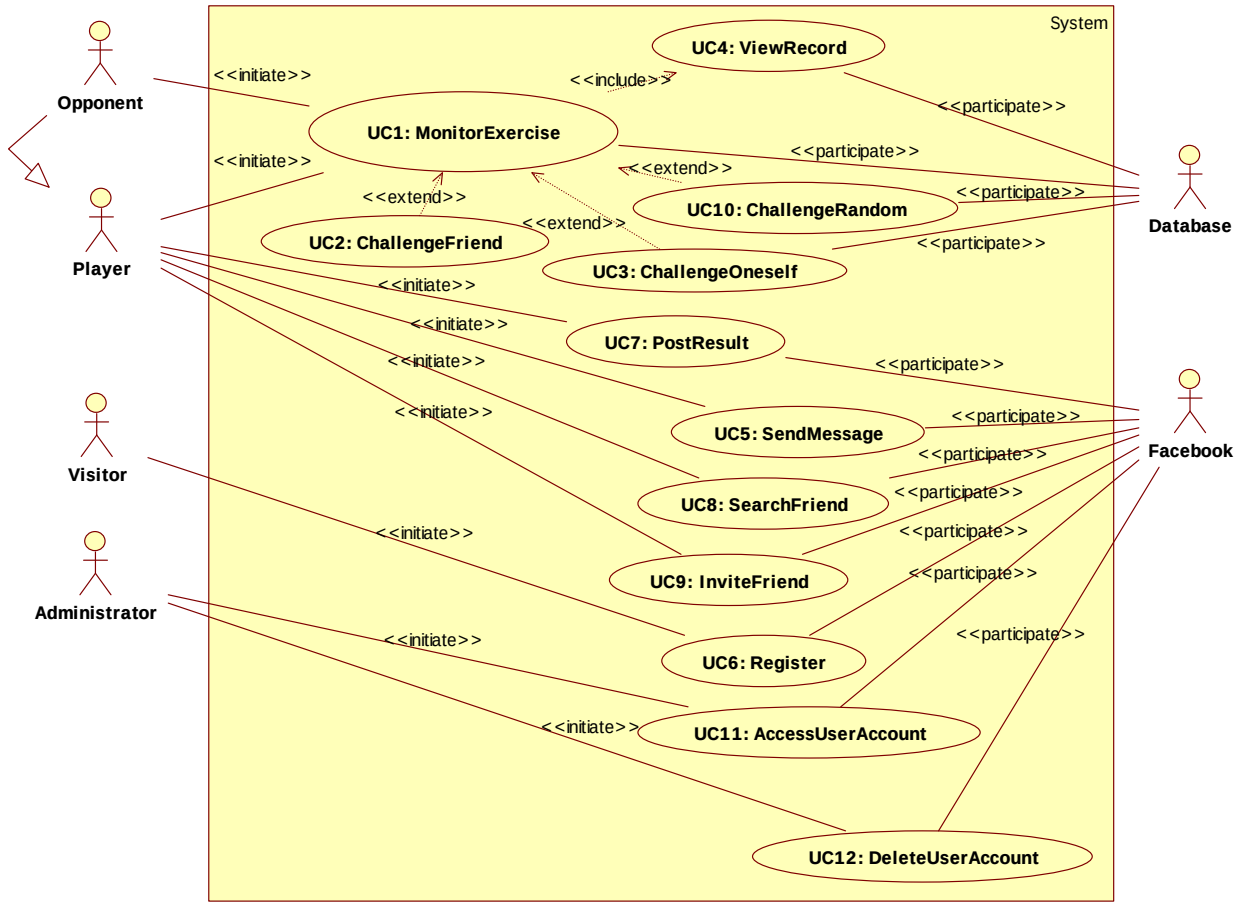


Figure 3-1: Use Case Diagram

Table 3-1: Traceability Matrix

REQ #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
UC1	X																			
UC2				X																
UC3																				
UC4											X									
UC5												X								
UC6																				
UC7														X						
UC8															X					
UC9																X				
UC10																	X			
UC11																		X		
UC12																			X	

3.4 System Sequence Diagram

In this section, the system sequence diagrams of some use cases defined above will be illustrated.

3.4.1 UC-1: MonitorExercise

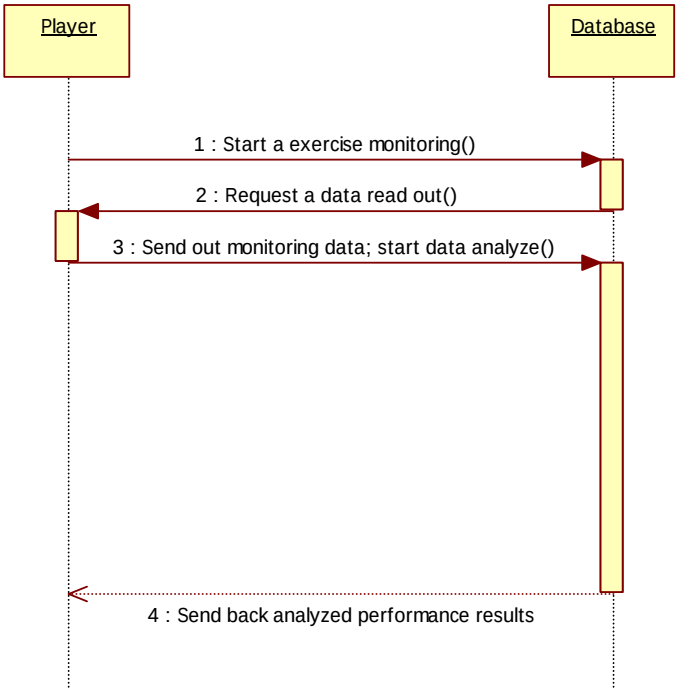


Figure 3-2: Sequence Diagram of UC-1: MonitorExercise

3.4.2 UC-2: ChallengeFriend

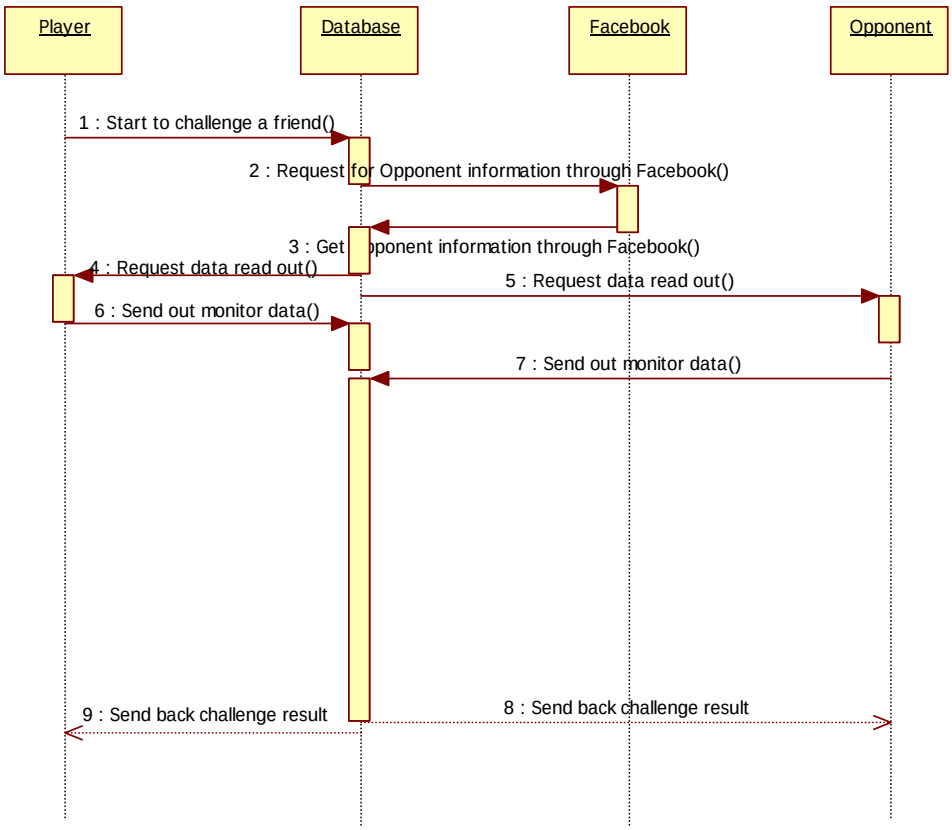


Figure 3-3: Sequence Diagram of UC-2: ChallengeFriend

3.4.3 UC-3: ChallengeOneself

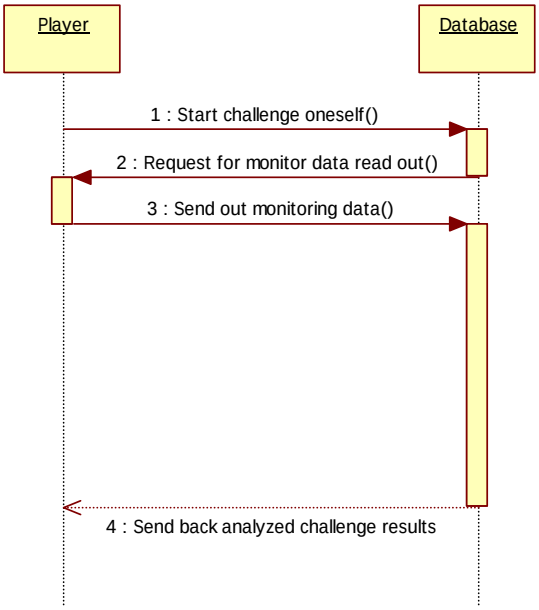


Figure 3-4: Sequence Diagram of UC-3: ChallengeOneself

3.4.4 UC-10: ChallengeRandom

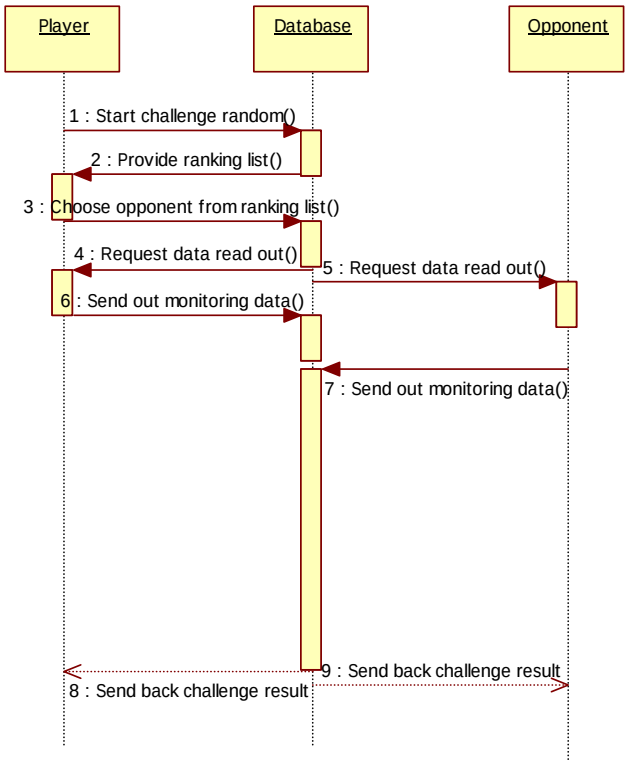


Figure 3-5: Sequence Diagram of UC-10: ChallengeRandom

4. User Interface Specifications

4.1 Preliminary Design

Here are a sample of screen-shots of the current work in progress:

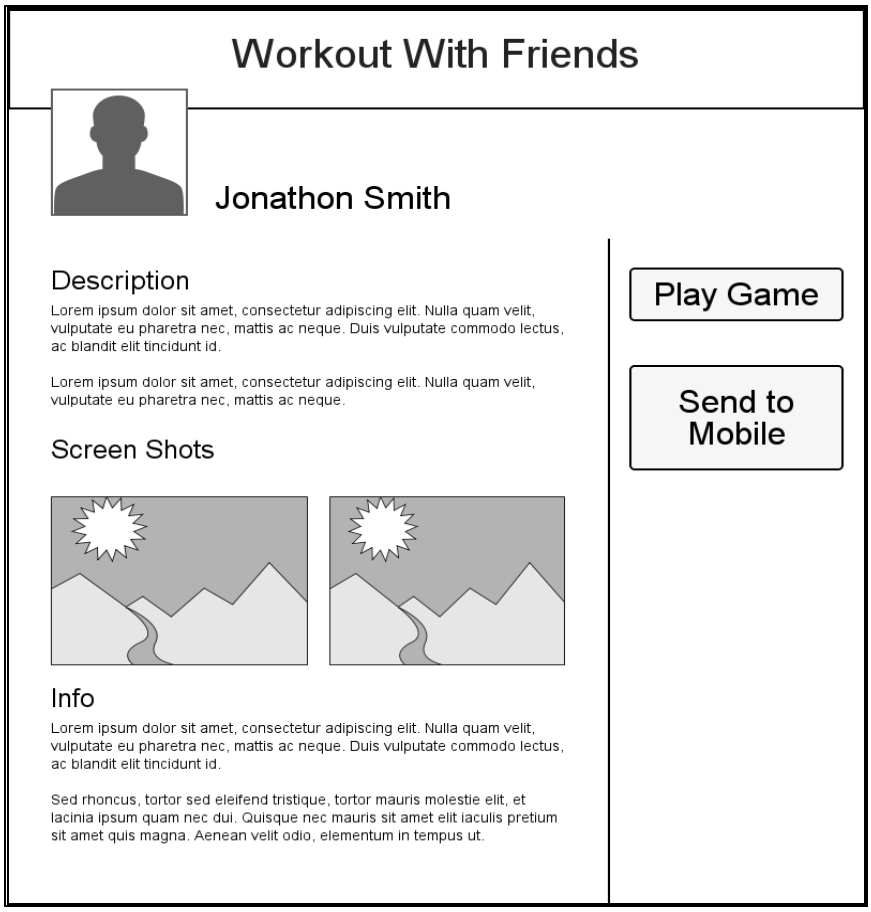


Figure 4-1: Facebook App Landing Page Concept (On-Screen Requirement 1)

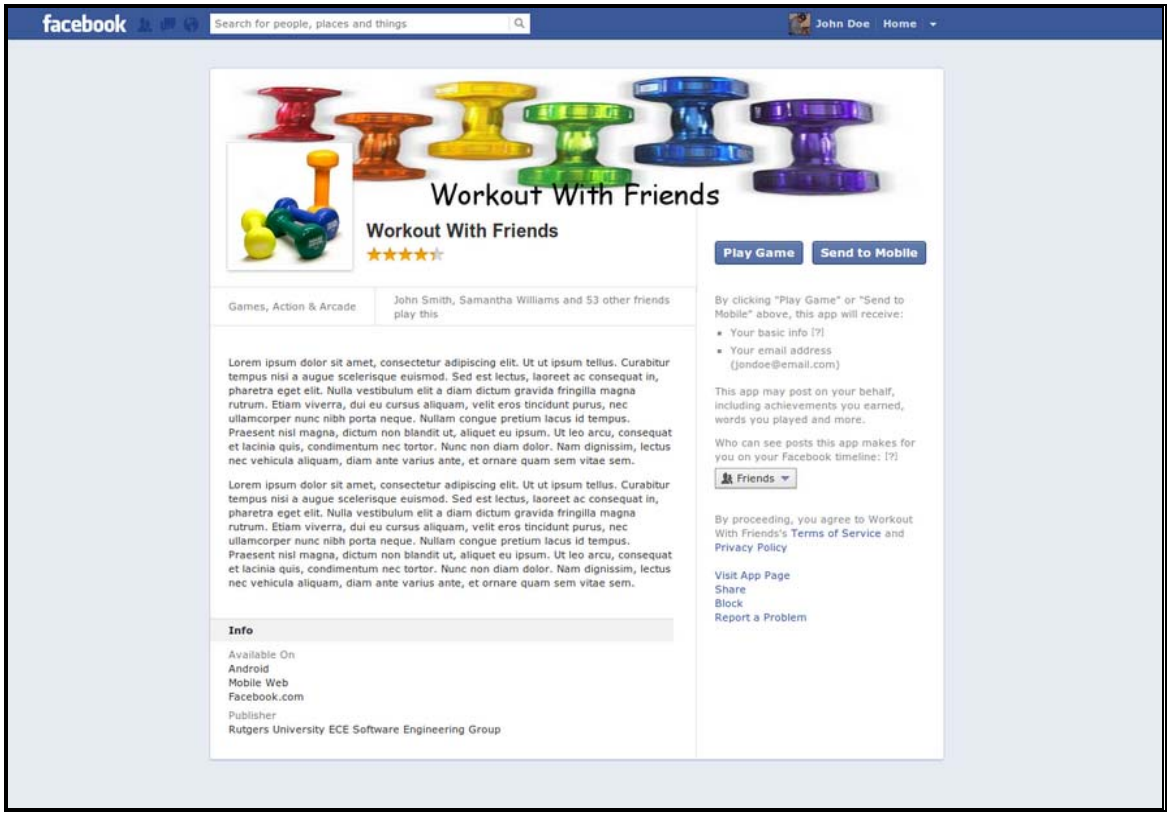


Figure 4-2: Facebook App Landing Page (On-Screen Requirement 1)

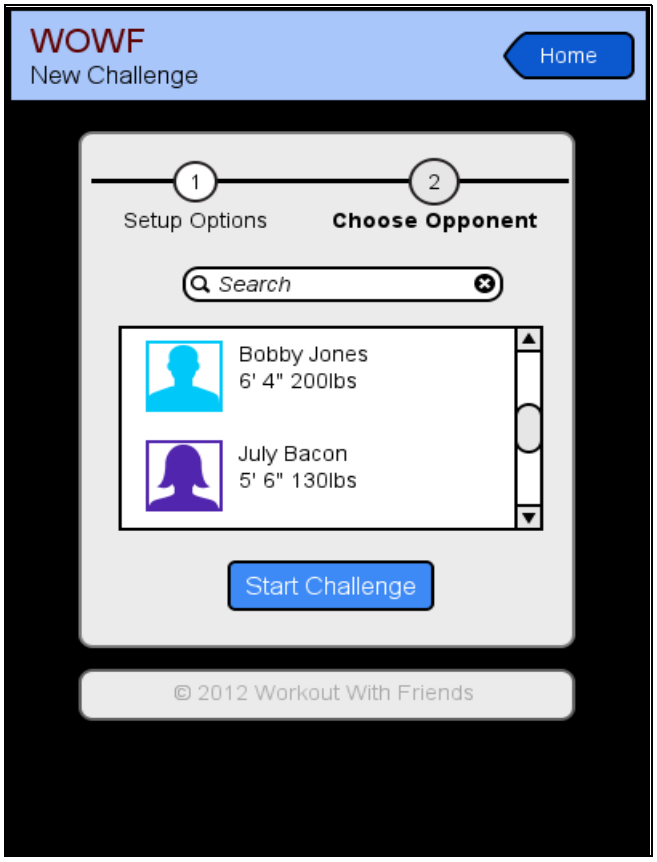


Figure 4-3: New Challenge (On Screen Requirement 1)

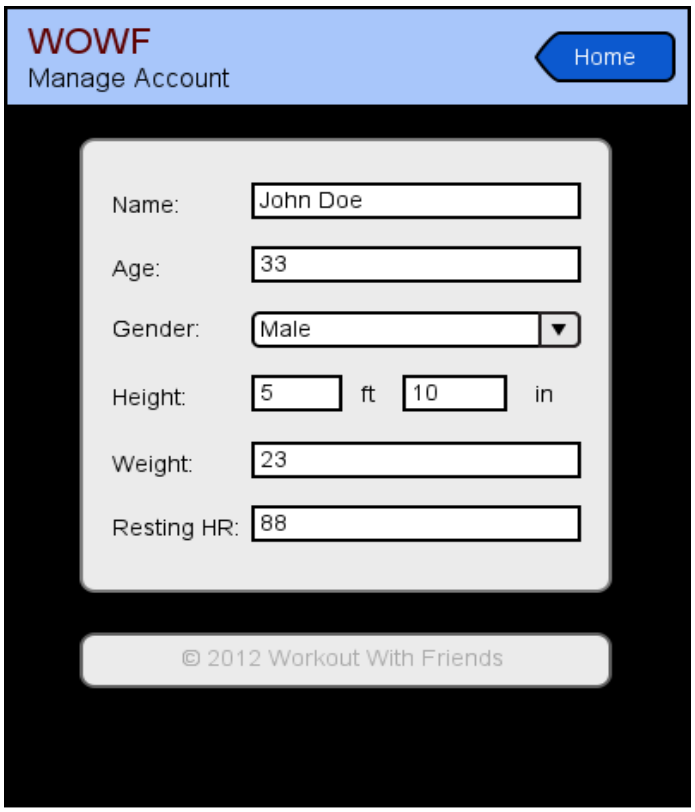


Figure 4-4: Account Manage (On Screen Requirement 2.b.2)

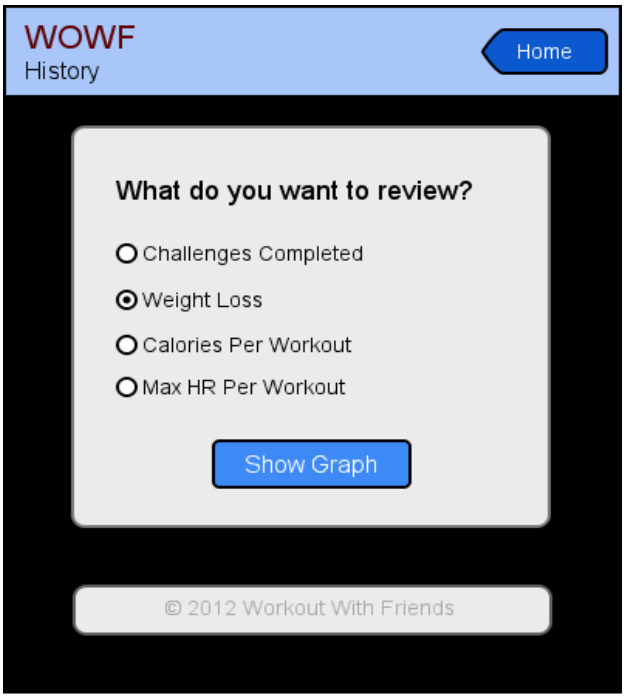


Figure 4-5: Challenge History (On-Screen Requirement 2.j)

4.2 User Effort Estimation

Regarding challenge set up, the game takes an estimated 5 minutes to set up from scratch. In total, there are (9) screens from initial login and registration to challenge set up:
Login with facebook account
2) Account Registration Part 1 (Instructions and how to play)
3) Account Registration Part 2 (Disclaimers and Permissions)
4) Account Registration Part 3 (User information)
5) Home Screen
6) Set up Challenge - Search Facebook Friends List
7) Select Friend(s) and send invitation
8) Challenge Setup Screen (options setup)
9) Manual start / timed start (warm-up)

However, once a user is registered subsequent challenge invitations will take substantially less time to set up. In total there are (5) screens once a user is already registered.

1) Home Screen
2) Set up Challenge - Search Facebook Friends List
3) Select Friend(s) and send invitation
4) Challenge Setup Screen (options setup)
5) Manual start / timed start (warm-up)

This does not yet meet the customer requirement of challenge set up in 2 “clicks” (screens) or less

5. Domain Analysis

5.1 Domain Model

5.1.1 Concept definitions

To analysis the domain model, we first derive domain model concepts and corresponding responsibilities from the formerly defined system use cases. Table 5-1 lists all the domain model concepts and corresponding responsibilities.

Table 5-1: Domain Concept Definition

Responsibility	Type	Concept	Use Case
Load monitoring data from monitor device.	D	MonitoringDataReader	US1: monitor exercise
Store monitoring data into Database.	D	MonitoringDataStorer	
Analysis stored monitoring data, and save into Database.	D	MonitoringDataAnalyser	
Compare analyzed results between Player and its Opponent.	D	FriendChallengeRefree	Us2: challenge friend
Compare analyzed results between Player and his/ her former record.	D	OneselfChallengeRefree	US3: challenge oneself
Show challenge result.	D	RecordViewer	US4: View record
Send a message to its Opponent during a challenge.	D	MessageSender	US5: send message
Allow Visitor to register through a Facebook account.	D	FacebookRegisterInterface	US6: register
Allow Player/ Opponent post the challenge result on Facebook wall.	D	FacebookPostInterface	Us7: Post result
Allow Player to search for a friend through Facebook.	D	FacebookFriendSearchInterface	Us8: Search friend
Allow Player to invite a friend to join the App through Facebook.	D	FacebookFriendInviteInterface	Us9: Invite friend
Provide a ranked list of all the Player' s results from Database, as well as the Player' s ranking in the list.	D	PerformanceRanker	Us10: Challenge random
Allow Player to choose a random Player with certain performance rank to challenge.		RandomChallengeSelector	
Compare analyzed results between Player and another random Player with certain performance rank.	D	RandomChallengeRefree	
Allow Administrator to check user account information.	D	UserAccountViewer	Us11: Access user account

Allow Administrator to delete invalid user account.	D	UserAccountOrganizor	Us12: Delete user account

5.1.2 Association definitions

Some of the concepts defined above as domain concepts have to work in certain pattern to finish some target, Table 5-2 gives the corresponding association definitions based on the defined domain concepts.

Table 5-2: Association Definition

Concept Pair	Association Description	Association Name
MonitoringDataReader←→ MonitoringDataStorer	MonitoringDataStorer store down-sampled data read out from MonitoringDataReader to Database.	Data read & save
FriendChallengeRefree←→ MonitoringDataAnalyzer	FriendChallengeRefree give the comparison result based on the output of MonitoringDataAnalyzer.	Analyze and compare
OneselfChallengeRefree←→ MonitoringDataAnalyzer	OneselfChallengeRefree give the comparison result based on the output of MonitoringDataAnalyzer.	Analyze and compare
RecordViewer←→ MonitoringDataAnalyzer	RecordViewer shows the analyzed results got from MonitoringDataAnalyzer.	Show analyzed data
PerformanceRanker←→ RandomChallengeSelector	RandomChallengeSelector allow Player to choose randomly based on the rank list PerformanceRanker gives out.	Choose from analyzed data
RandomChallengeRefree←→ MonitoringDataAnalyzer	RandomChallengeRefree give the comparison result based on the output of MonitoringDataAnalyzer.	Analyze and compare

5.1.3 Attribute definitions

Among the defined concepts, some concepts share the same attribute, and only different from each other as they have different operands. These concepts are listed in Table 5-3.

Table 5-3: Attribute Definition

Concept	Attributes	Attribute Description
MonitoringDataReader	Data access and storage	Read in and store monitoring data from the device.
MonitoringDataStorer		
MonitoringDataAnalyser	Data analyzer	Analyze data.
PerformanceRanker		
FriendChallengeRefree	Challenge refree	Compare the performance between from different users.
OneselfChallengeRefree		
RandomChallengeRefree		
RecordViewer	User interface	Store user input to the system, or show analyzed result to user.
MessageSender		
RandomChallengeSelector		
FacebookRegisterInterface	Facebook interface	Communicate with Facebook database.
FacebookPostInterface		
FacebookFriendSearchInterface		
FacebookFriendInviteInterface		
UserAccountViewer	User account interface	Allow Administrator have

UserAccountOrganizor		access and control of the user accounts.

5.1.4 Traceability matrix

Table 5-4: Traceability Matrix

UC	PW	Domain Concept															
		MonitoringDataReader	MonitoringDataStorer	MonitoringDataAnalyser	FriendChallengeRefree	OneselfChallengeRefree	RecordViewer	MessageSender	FacebookRegisterInterface	FacebookPostInterface	FacebookFriendSearchInterface	FacebookFriendInviteInterface	PerformanceRanker	RandomChallengeSelector	RandomChallengeRefree	UserAccountViewer	UserAccountOrganizor
UC1		X	**X**	**X**													
UC2					**X**												
UC3						**X**											
UC4							**X**										
UC5								**X**									
UC6									**X**								
UC7										**X**							
UC8											**X**						
UC9												**X**					
UC10													**X**	**X**	**X**		
UC11																**X**	
UC12																	**X**

5.2 System Operation Contracts

MonitorExercise
Preconditions:
- User has an account
- User has completed a workout

Postconditions:
- The associated challenge is updated accordingly

ChallengeFriend
Preconditions:
- User has an account
- Player to be challenged has an account

Postconditions:
- The game is created an made available to both users

ChallengeOneself
Preconditions:
- User has an account

Postconditions:

- The game is created an made available to the user

ViewRecord
Preconditions:
- User has an account

Postconditions:
None

SendMessage
Preconditions:
- User has an account
- Player to receive message has an account
- A game has been created between the two players

Postconditions:
- The message is visible to both users

Register
Preconditions:
- User has a Facebook account

Postconditions:
- User is stored in the database

PostResult
Preconditions:
- A game has been completed between the two users

Postconditions:
None.

SearchFriend
Derived from REQ-15.
Preconditions:
- User has an account

Postconditions:
None.

InviteFriend
Preconditions:
- User has an account
- Invitee has a Facebook account.

Postconditions:
None.

ChallengeRandom
Preconditions:
- User has an account
- Player to be challenged has an account

Postconditions:
- The game is created an made available to both users

AccessUserAccount
Derived from REQ-18.
Preconditions:
- Account Administrator is logged in
- Player to be challenged has an account

Postconditions:

None.

DeleteUserAccount
Preconditions:

- Account Administrator is logged in
- Player to be deleted has an account

Postconditions:

- Deleted user data is no longer in the database

5.3 Mathematical Model

The following describes the mathematical model as implemented by our database structures.

Let uid be the userid, key be the user password, f be the friends list, u be the user, cl be the challenge list, and c be the challenge

$$let\, uid, key, email, age, access, f \in u$$

All uids are unique to one user

$$(\forall\, uid)(\forall\, u_1)(\forall\, u_2)((uid \in u_1) \rightarrow (uid \notin u_2))$$

All emails are unique to one user

$$(\forall\, email)(\forall\, u_1)(\forall\, u_2)((email \in u_1) \rightarrow (email \notin u_2))$$

$$let\, c \in cl$$

$$let\, uid_1, uid_2, typeid, subtype, result1, result2, score1, score2 \in c$$

A user cannot challenge himself

$$(\forall\, uid_1)(\forall\, uid_2)(\forall\, c)((uid_1 \in c) \wedge (uid_1 \in c) \rightarrow (uid_1 \neq uid_2))$$

All users contain one member of each type

$$(\forall\, uid_1)(\forall\, uid_2)(\forall\, u)((uid_1 \in u) \wedge (uid_1 \in u) \rightarrow (uid_1 = uid_2))$$

$$(\forall\, email_1)(\forall\, email_2)(\forall\, u)((email_1 \in u) \wedge (email_1 \in u) \rightarrow (email_1 = email_2))$$

$$(\forall\, age_1)(\forall\, age_2)(\forall\, u)((age_1 \in u) \wedge (age_1 \in u) \rightarrow (age_1 = age_2))$$

$$(\forall\, access_1)(\forall\, access_2)(\forall\, u)((access_1 \in u) \wedge (access_1 \in f) \rightarrow (access_1 = access_2))$$

$$let\, uid \in f$$

A user cannot friend himself

$$(\forall\, u)(\forall\, f)(\forall\, uid)((uid \in f) \wedge (f \in u) \rightarrow (uid \notin u))$$

If a user has a friend, that friend must also have the user as a friend

$$(\forall\, f_1)(\forall\, u_1)(\forall\, uid_1)(\forall\, f_2)(\forall\, u_2)(\forall\, uid_2)((uid_1 \in u_1) \wedge (f_1 \in u_1) \wedge (uid_2 \in f_1) \wedge (uid_2 \in u_2) \rightarrow (uid_1 \in f_2))$$

6. Interaction Diagrams

6.1 Sequence Diagrams

In this section, the system sequence diagrams of some most important use cases defined above will be illustrated.

6.1.1 UC-1: MonitorExercise

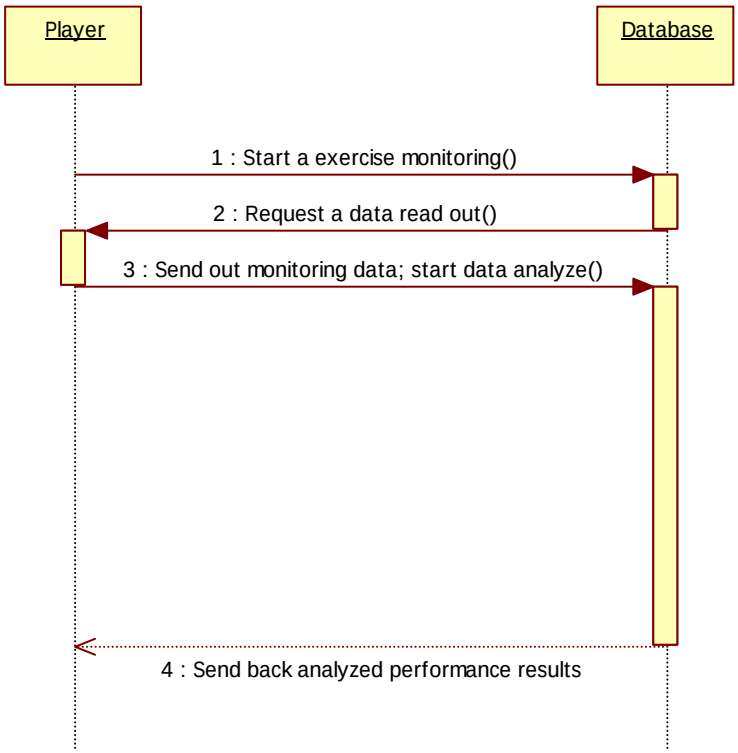


Figure 6-1: Sequence Diagram of UC-1: MonitorExercise

6.1.2 UC-2: ChallengeFriend

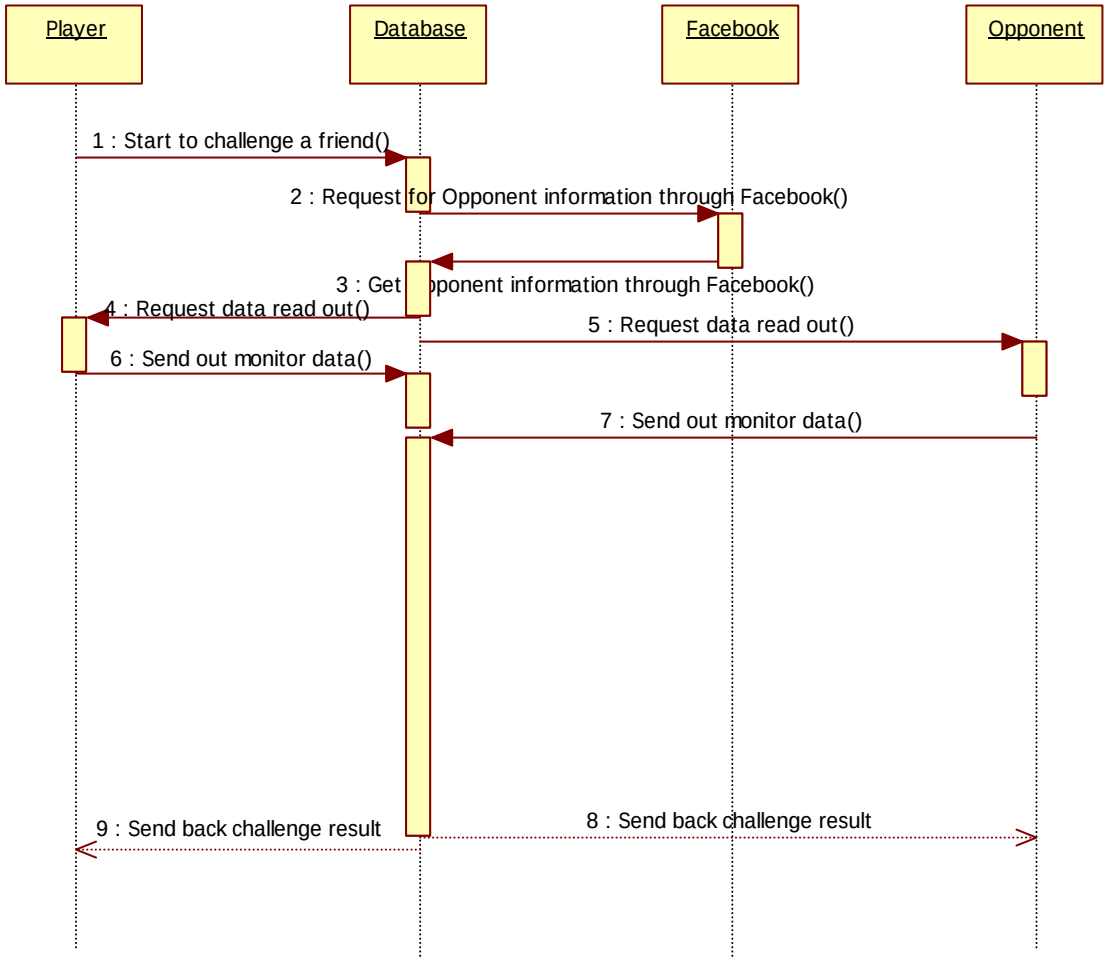


Figure 6-2: Sequence Diagram of UC-2: ChallengeFriend

6.1.3 UC-3: ChallengeOneself

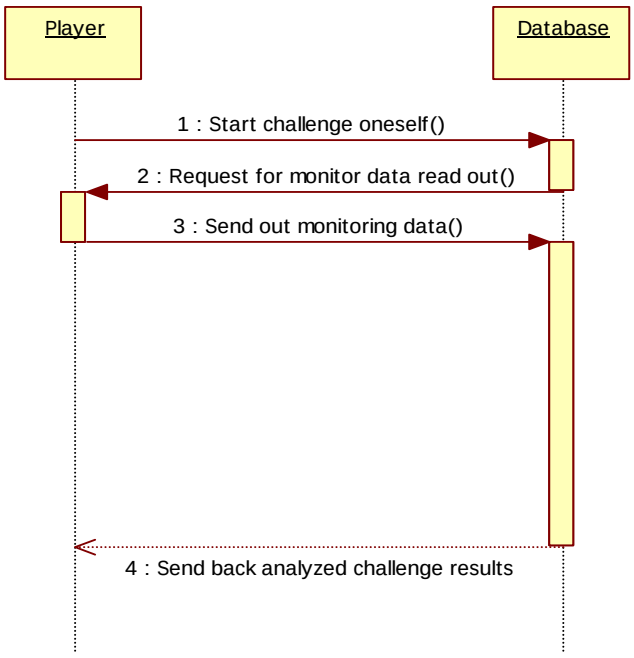


Figure 6-3: Sequence Diagram of UC-3: ChallengeOneself

6.1.4 UC-4: ViewRecord

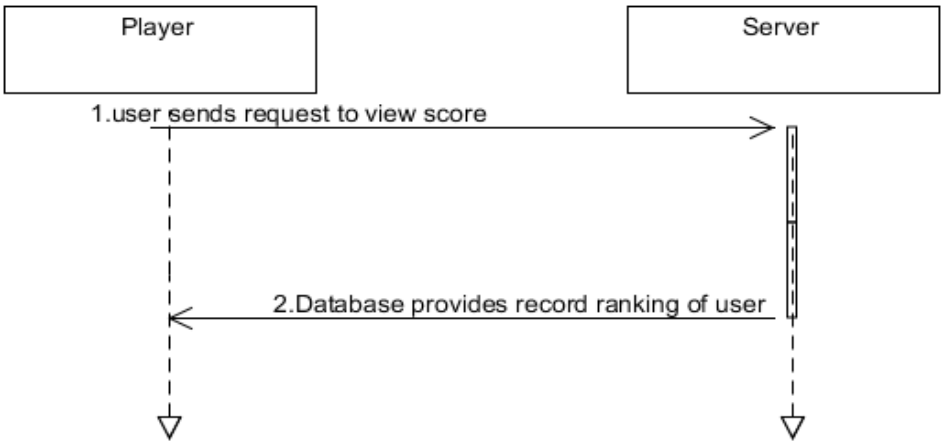


Figure 6-4: Sequence Diagram of UC-4:View Record

6.1.5 UC-5: Send Message

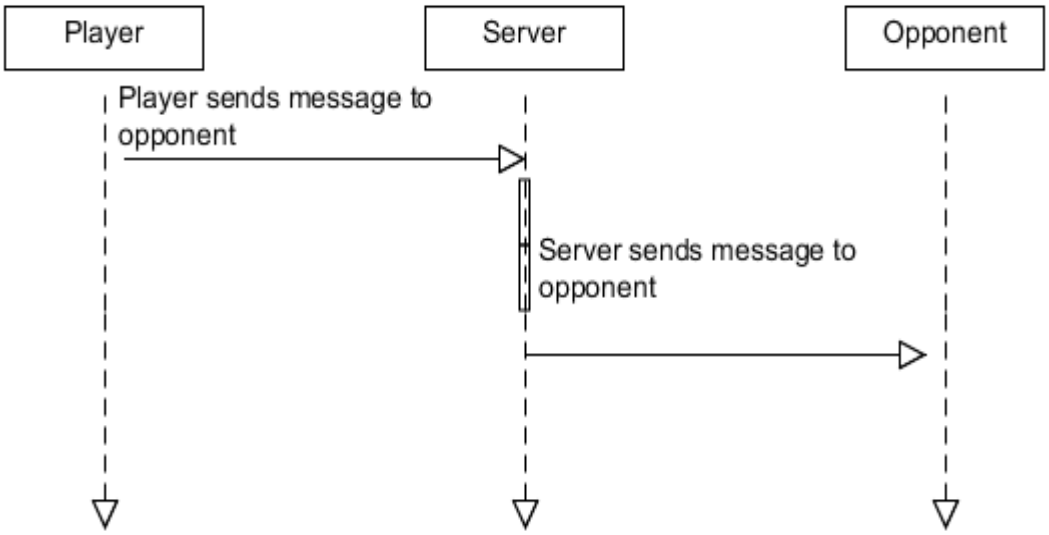


Figure 6-5: Sequence Diagram of UC-5:Send Message

6.1.6 UC -6:Register Profile

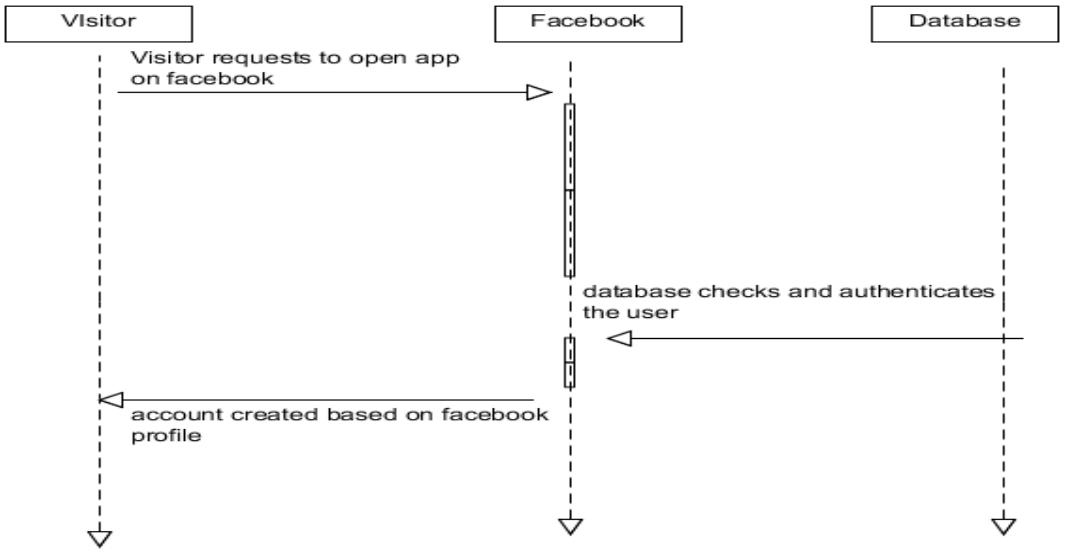


Figure 6-6: Sequence Diagram of UC-6: Register Profile

6.1.7 UC-7:Post Result

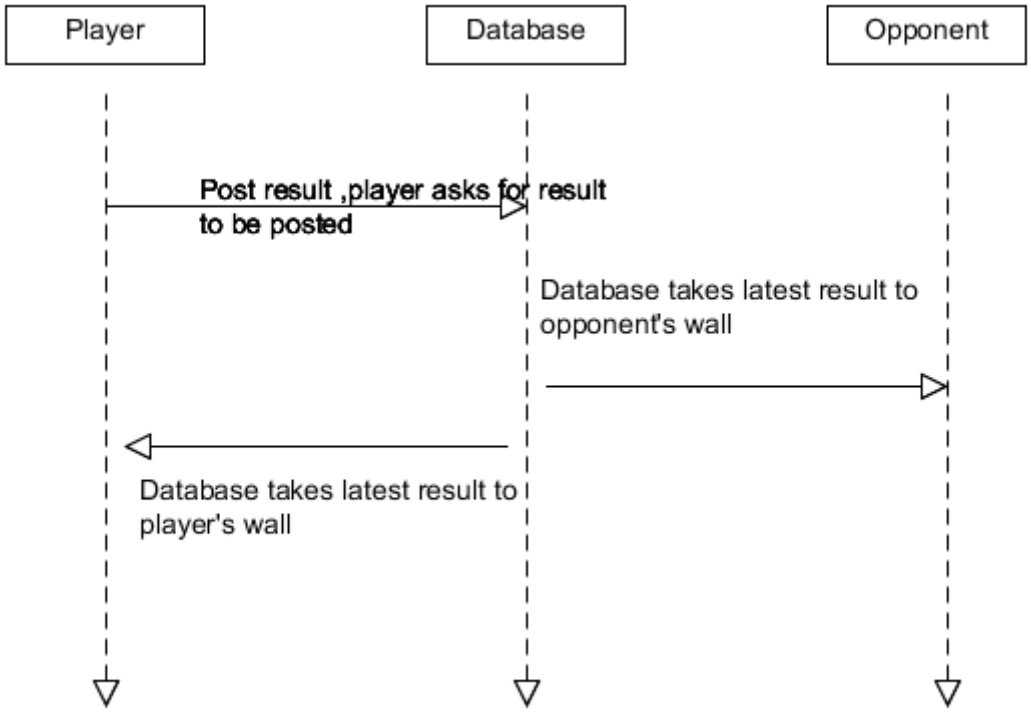


Figure 6-7: Sequence Diagram of UC-7:Post Result

6.1.8 UC-8:Search Friend

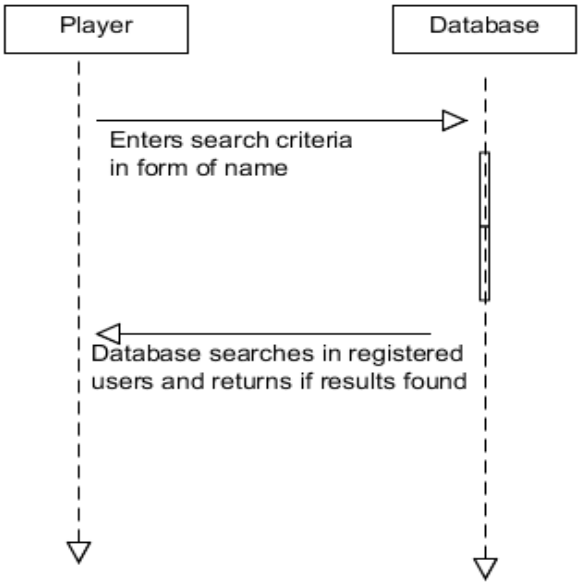


Figure 6-8: Sequence Diagram of UC-8:Search Friend

6.1.9 UC-10: ChallengeRandom

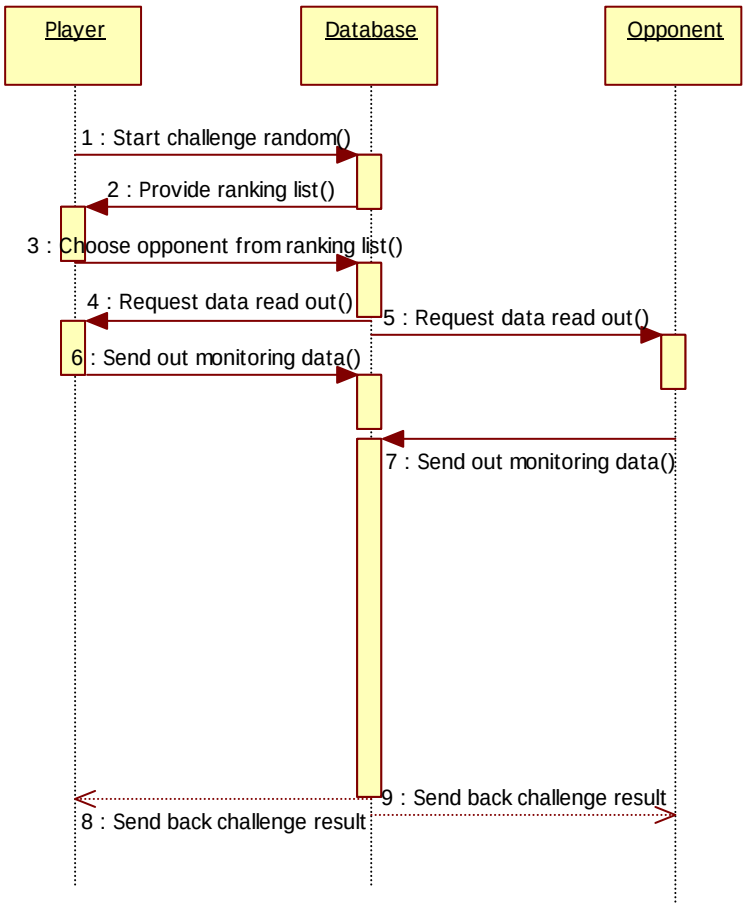


Figure 6-9: Sequence Diagram of UC-10: ChallengeRandom

7. Class Diagrams and Interface Specification

7.1 Class Diagrams

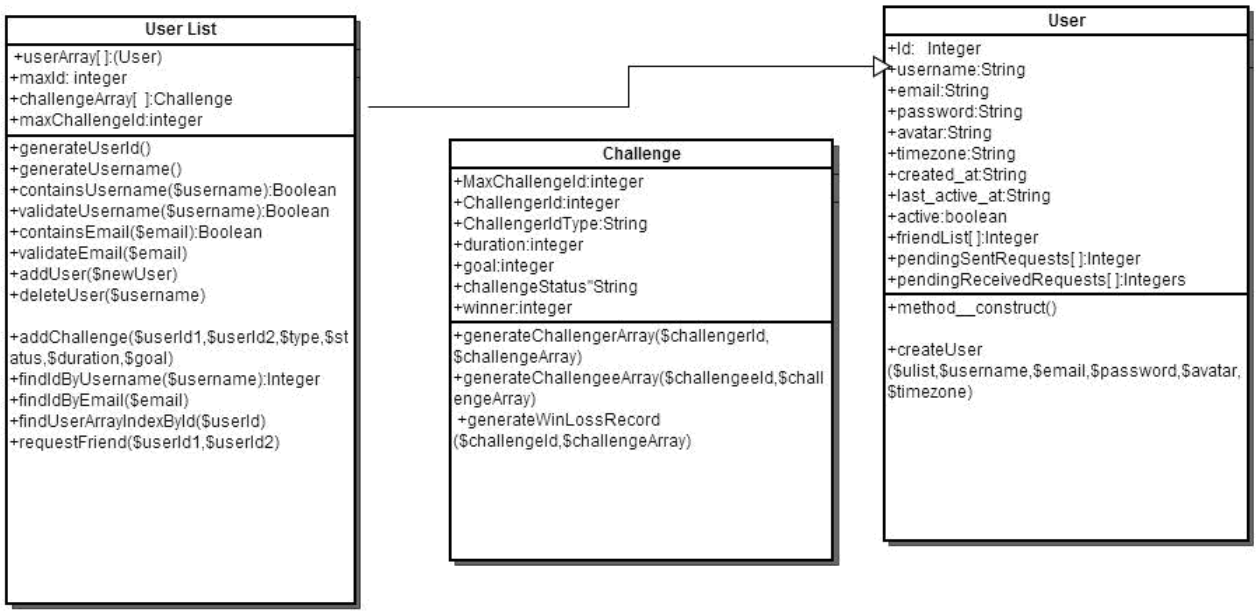


Figure 7-1: Class Diagram

7.1.1 Classes Specifications :

1.Challenge():
The Challenge () class manages the various challenges incorporated by all users .
The variables used in Challenge () and their brief descriptions :

public maxChallengeId:Integer largest challenge id
public challengerId:Integer challenger's id
public type:String The two ways you can play " ("STRENGTH", "ENDURANCE")
public duration: Integer Duration of workout in minutes

*The default value for duration ,if not entered by the user ,is 5 minutes .
public goal:integer Approximate amount of cals.the player wants to burn in calories.
*The default value for goal ,if not entered by the user ,is 5 cals .

public challengeStatus:String ("OPEN","CLOSED")
public winner:Integer User id of winner

Functions in Challenge():

function__construct(maxChallengeId,challengerId,challengeeId,type,status,duration,goal):
The explicitly defined constructor which defines all variables to the user's inputs.

public displayWinner(): A function which prints the id and name of the player who has won .

public generateChallengerArray(challengerId,challengeArray)
return type:array of challenges.
takes the user_list challenge array, and generates all the challenges from an id.

public generateChallengeeArray(challengeeId,challengeArray): for user terminated challenges
return type: array of challenges

function generateWinLossRecord(challengeId,challengeArray)
generates array of integers [W][L][T]

2.User():

The User class gives all details of a user when registering for "workout with friends ".Quite simply , it provides all the details which the user finds when they login and open their dashboard .

The variables used in User() and their brief descriptions :

public id: integer user's unique user id of exactly 10 digits
public username:String user's unique username (should start with alpha)
public email:string The email id of user
public password:string should be 8-20 chars ,should include one upper case,one lower case ,one special character and one digit .
public avatar:undefined (string)
public timezone:string
public created_at:string time when the user registered in "workout " .
public last_active_at:string time when the user last accesses "workout ".
public active:Boolean to check if user is presently active or not .
public friendList[ ]: userid collection of all user ids .

public pendingSentRequests[ ]: userid collection of userids whose requests who have been sent out by the user , but no action has been taken by reciepients .

public pendingReceivedRequests[ ]: userid collection of userids whose requests have not been taken any action by the user .

Functions in User():
method__construct():
Explicitly defined constructor to initialize instance of user, we need to use createUser to add information. This function only initializes arrays-friendslist[],pendingSentRequests[] and pendingRecieved Requests[].

createUser(ulist,username,email,password,avatar,timezone):

createUser() is used to initialize the basic parameters while registering .
$ulist is userlist object.

3.UserList():

The UserList() is a class , quite simply storing the details of “workout with friends” – including the validation functions, and the various operations which can be performed by the user .

The variables used in UserList() and their brief descriptions :

public userArray: (User)	array of user objects
public maxId:integer	largest issued userid???
public challengeArray: (Challenge)	array of challenge objects
public maxChallengeId:integer	largest issued challengeid????

Functions in UserList():
Public generateUserId():
This function generates a unique user id when the user registers into the site for the first time .(it works by incrementing the userid by one for every user)
function generateUsername():
The username given by the user is set in this function .
Public containsUsername(String):
return type:boolean
this function checks if username exists in the database of UserArray[ ] in the UserList() class .
Public validateUsername(String):
return type :Boolean
the function validates and checks if username is of valid type,The username must :contain only alpha numeric characters and start with an alpha character .

public containsEmail(String):
return type:Boolean
the function checks if emailid exists in the database of UserArray[ ] in the UserList() class .

public validateEmail(String):
return type:Boolean
the function validates and checks if email is of valid type.
Public addUser(User type):
adds user object to userArray[].
Public deleteUser(String-userName):
this function only deactivates an account , the user can come back and activate it any time later on.
public addChallenge($userId1,$userId2,$type,$status,$duration,$goal):
adds challenge to challengeArray[ ].
function findIdByUsername($username):
This function enables a user to find a friend by searching for the username .The function returns -1 of username doesn't exist, or is not found .
function findIdByEmail($email):
This function enables a user to find a friend by searching for the emailid .

function findUserArrayIndexById(userId):
This function enables a user to find a friend by searching for the userid .
function requestFriend($userId1,$userId2):
This function ,enables a user to invite a friend for a challenge .When the user sends a request to a friend ,the function adds userid2 to userId1's object's pending friend request array[].

7.2 Tractability Matrix

Table 7-1: Tractability Matrix

Use Case	Challenge()	User()	UserList()
Compare analyzed results between Player and its Opponent.	×		
Compare analyzed results between Player and his/ her former record.			×
Show challenge result.		×	
Send a message to its Opponent during a challenge.	×		
Allow Visitor to register through a Facebook account.		×	
Allow Player/ Opponent post the challenge result on Facebook wall.	×		
Allow Player to search for a friend through Facebook.			×
Allow Player to invite a friend to join the App through Facebook.			×
Provide a ranked list of all the Player' s results from Database, as well as the Player' s ranking in the list.			×
Allow Player to choose a random Player with certain performance rank to challenge.			×
Compare analyzed results between Player and another random Player with certain performance rank.	×		

The most difficult thing to do was to trace the actual classes we had obtained to the domain concepts we had recognized earlier and decided to implement :

Concepts implemented in Challenge ():

1. Compare analyzed results between Player and its Opponent.

2. Compare analyzed results between Player and another random Player with certain performance rank.
3. Allow Player/ Opponent post the challenge result on Facebook wall.

The main reason of a challenge class was to for the interaction between the user and any other friend with whom a challenge has been accepted .
The results of a challenge , is given once a challenge is over and a winner is clearly given .
Then , the analysis of the challenge is present , giveingthe user details of who has won , by how much and also the improvement since the beginning .

The option given to the user to post the challenge result on one's facebook wall could have been given to either the user() class or the challenge() class .However , it iseasy to seethat the result can be posted only once a particular challenge has been completed ,and the results are also present with the challenge() class .Hence , the concept was implemented in the challenge() class .

Concepts implemented in UserList():

Easily the most important class , the user() class helda lot of domain concepts and many were implemented using the UserList() class , as it held all the data of the user ().
1.Compare analyzed results between Player and his/ her former record.
2.Allow Player to search for a friend through Facebook.
3.Allow Player to invite a friend to join the App through Facebook.
4.Provide a ranked list of all the Player' s results from Database, as well as the Player' s ranking in the list.
5.Allow Player to choose a random Player with certain performance rank to challenge.

The UserList () has the data of all previous challenges of the user , and can hence show the improvement the user has been making .

8. System Architecture and System Design

8.1 Architectural Styles

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

By using this client/ server model, the system reduced the procedure done on the client side to a minimum degree, and leave all the process work to the server. In this way, we can provide a light user application, also a better coherence with all the user data.

The high degree of centralization also provide great convince for future system updates. We can easily upgrade the system mathematics model for more accurate data analysis, as well as upgrade the database structure for function scaling.

However, the high centralized degree of the system also provide high standard request to the server, as the server act as both the database and the central data processor in the system. This centralized system structure require high computing capability as well as large storage capability.

8.2 Identifying Subsystems

The system can basically be partitioned into Client subsystem, and Server subsystem. The Client subsystem consist only of a user interface, while the Server subsystem consists of several different functional units. The UML package diagram indicating the workflow between the subsystems can be seen below.

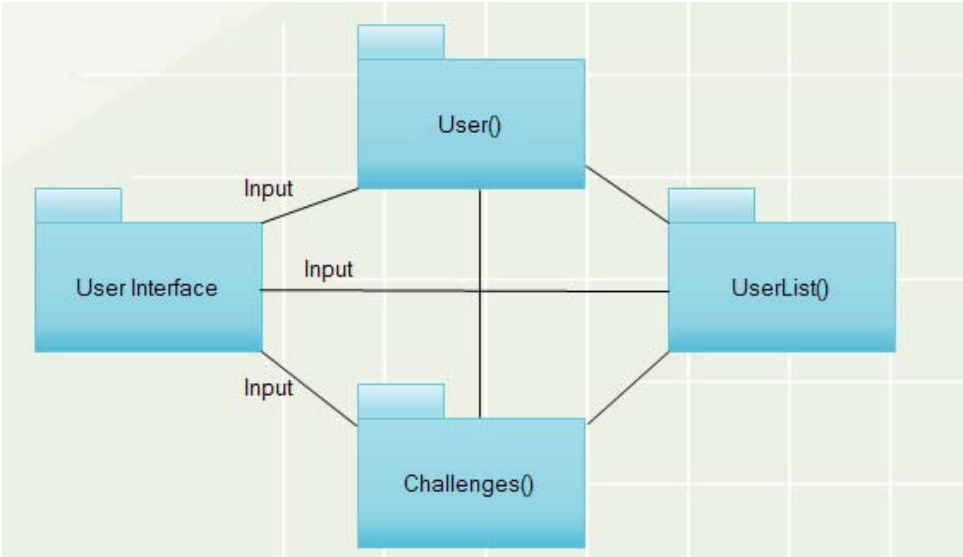


Figure 8-1: UML package diagram

The User Interface is the gateway into the system. It is through this that a user will create his profile, edit said profile, create challenges, view results, and more. The list of users is maintained through the

user list package, which is used to create individual user classes. It is also provides the userId information to the challenge package, for match information between two challengers. The user class can be accessed through the user interface to edit a particular user's personal information as necessary; it can also access the challenge class when creating and maintaining existing challenges. The challenge class is used to create challenges between two users, and must be able to access the initiators information as well as the UserList to inform the challenged user that there is a workout request waiting for them.

8.3 Mapping Subsystems to Hardware

Following the client/ server architecture, the hardware will also be mapped to two sets.

- For the Server subsystem, we use a personal PC running as the server(for the current stage).
- For the Clients subsystem, the application will be running on different user machines.

8.4 Persistent Data Storage

The data stored in the server consist of three main categories: the user information; the user history data; and temporary user data. The temporary user data are the monitoring data that a user uploaded when they start a challenge, and the temporary data would not be stored after the system finish the data analyze.

The user information and the user history otherwise will be stored into the database. The user information kept in the database keeps record of the basic login information for each user, as well as the basic physical characters that will be used for the user' s challenge. The user history keeps record of all the challenge records of each user, and use for history show and further analysis.

8.5 Network Protocol

Our system has a central server, which is hosted through Apache HTTP. Requests to access information in the databases, which are MySQL, are done through php.

8.6 Global Control Flow

Workout with Friends is an event driven system. This means that users can generate actions in different order. Once a User has created his profile, he can, at any time, challenge another user to a match. The challenged user may accept or decline, and either user can complete the workout at their convenience. Once both users have completed their workout challenge, they can view the results of the particular challenge at any time. In addition, that data can also be observed as part of a broader data.

8.7 Hardware Requirement

Workout with Friends requires the use of a MOTO Heart Rate monitor. In addition, the client PC must have an internet connection to be able to access the server.

9. Algorithms and Data Structures

9.1 Algorithms

9.1.1 Basic Scoring Algorithm

Workout with Friends, is a challenge based application, and the challenge result is based solely on the score of each work out. The workout intensity score is computed based on the weighted score of three different inputs: workout time spent in HRZ 5, HRZ 4, and calories burned.

For the simplistic model:
1. Score1 : Time (in seconds) spent in heart rate zone 5 (90-100% of target max HR) x factor of 10 (Weighting = 40%)
- Example: 1:45 min = 105 seconds x 10 = 1050
- Score1 = 1050 x 40% = 420 points

2. Score2 : Number of calories burned – (Weighting = 25%)
- Example: 850 calories
- Score2 = 850 x 25% = 212.5 points

3. Score3 : Time (in seconds) spent in heart rate zone 4 (80-89% of target max HR) – (Weighting = 35%)
- Example: 21:45 min = 1305 seconds
- Score3 = 1305 x 35% = 456.75 points

4. Final Score = Score1 + Score2 + Score3
- Example: Final Score = 420 + 212.5 + 456.75 = 1089.25 points

Based on this model, when calories and HRZ 4 and 5 are known, the calculation is straightforward and the process to determine a winner is simple.

In the embedded Facebook web-app case, where no specific calories burned and HRZ are known, we will use a formula to calculate the calories burned. The starting conditions (known variables) are: gender, weight, age, exercise duration and average heart rate of the workout. The equations derived by LR Keytel, JH Goedecke, TD Noakes, H Hiiloskorpi, R Laukkanen, L van der Merwe, and EV Lambert for their study titled "Prediction of energy expenditure from heart rate monitoring during submaximal exercise" are shown below:

⇒ Male: C = ((-55.0969 + (0.6309 x HR) + (0.1988 x W) + (0.2017 x A))/4.184) x 60 x T
⇒ Female: C = ((-20.4022 + (0.4472 x HR) - (0.1263 x W) + (0.074 x A))/4.184) x 60 x T
Where:
C = Calories burned
HR = Heart rate (in beats/minute)
W = Weight (in kilograms)
A = Age (in years)
T = Exercise duration time (in hours)

9.1.2 Improved Scoring Algorithms

Strength

The strength workouts are based on research of mathematical models that normalize powerlifting abilities based on weight and repetition. The first underlying formula is the Wilks Formula,

developed by Robert Wilks, is used for scoring by the International Powerlifting Federation (IPF). The formula to determine the Wilks Coefficient is as follows (from Wikipedia):

$$Coeff = \frac{500}{a + b \cdot x + c \cdot x^2 + d \cdot x^3 + e \cdot x^4 + f \cdot x^5}$$

x is the body weight of the lifter in kilograms

Coefficients for *men* are:
a=-216.0475144
b=16.2606339
c=-0.002388645
d=-0.00113732
e=7.01863E-06
f=-1.291E-08

Coefficients for *women* are:
a=594.31747775582
b=-27.23842536447
c=0.82112226871
d=-0.00930733913
e=0.00004731582
f=-0.00000009054

The second formula is the Wathan 1 Rep max formula, used to estimate a person's 1RM based on repetitions at a given weight (from Wikipedia).

$$1RM = \frac{100 \cdot w}{48.8 + 53.8 \cdot e^{-0.075 \cdot r}}$$

Where w is weight (dimensions doesn't matter) and r is reps

The normalized lifted weight is determined as follows:

$$x(w, r) = Coeff(0.453592 \cdot w) \cdot \frac{100 \cdot w}{48.8 + 53.8 \cdot e^{-0.075 \cdot r}}$$

(Bench Press): The x derived in the previous equation is used in the equation in the following chart to determine the score for the challenge:

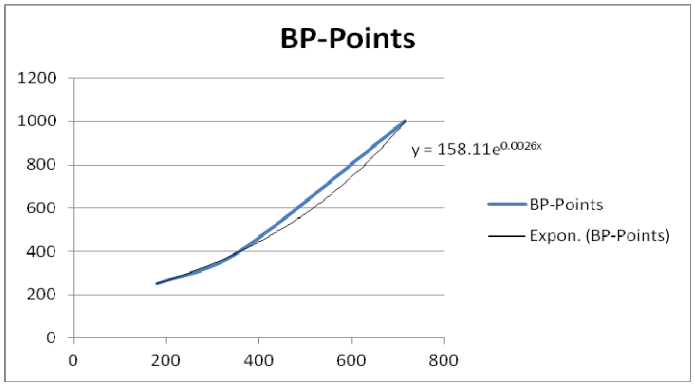


Figure 9-1: Score determined for challenge result.

This equation was designed to score the maximum 1000 points to the person that can perform the maximum bench press rep performed by the record holder, Scot Mendelson, at 715lb. Using an

exponential model allows the challenger to be rewarded even for smaller incremental gains during the plateau period. The following chart shows the squat equivalent:

(Squat):

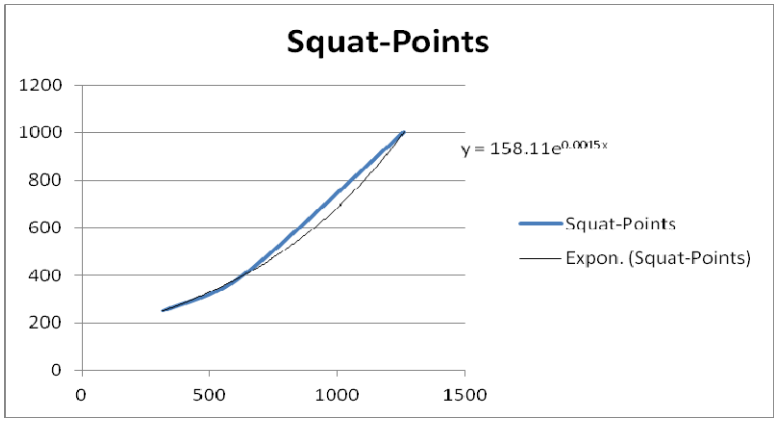


Figure 9-2: Squat equivalent illustration.

(Speed): The following chart shows the exponential algorithm used to determine the points in Speed games. Each line depicts a different distance chosen by the challenger/player. The choices are 100m, 200m, 400m, 800m, and 1600m.

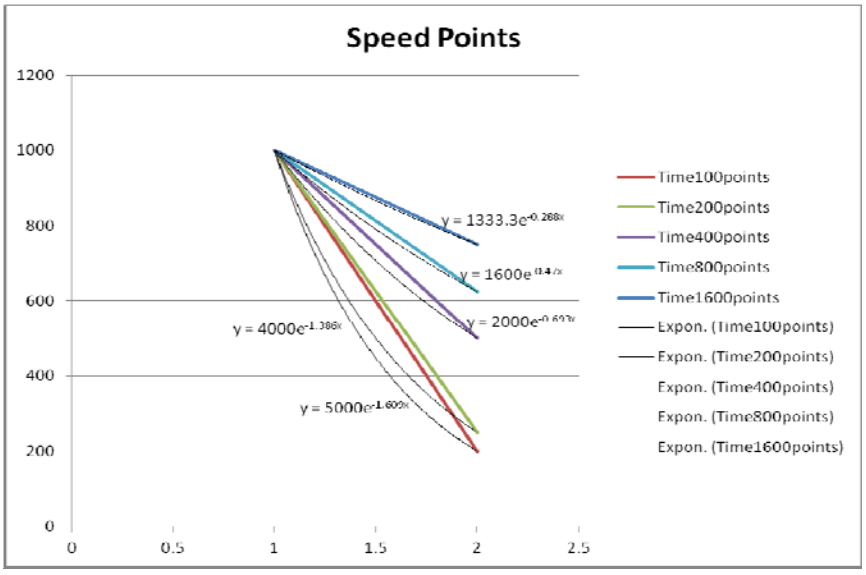


Figure 9-3: Speed points illustration.

These graphs are based on current world best times in these events, and getting these scores would yield the full 1000 points. The x-axis is how many times greater that player's distance is off from the world best in that event, or User_Time/World_Record_Time. This scheme rewards those players who do comparatively worse on a longer distance event than the short distance counterparts, because the difference between average and elite in short distance and long distance is not linear.

(Endurance): The endurance challenge mathematical model is arguably the most complicated, as there is no real metric to which we can give the title "world best". As such, this model calls a 100% heart rate "perfect" and calls the calories burned at 100% heart rate "perfect." Thus, it is more likely that the scores in these challenges will be lower in practice. Below shows the charts depicting the point allocation of both heart rate and calories, and their points are added together to give the overall score.

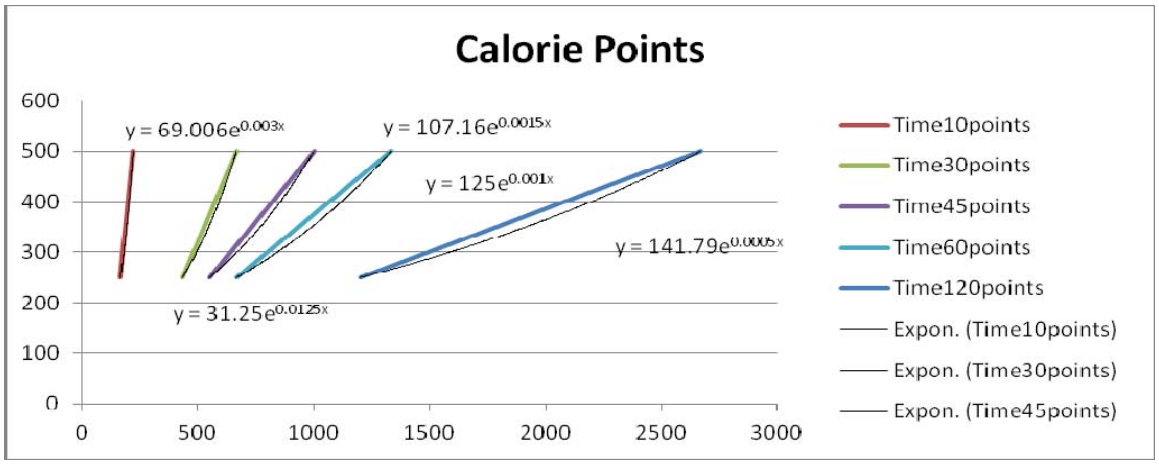


Figure 9-4: Calorie points illustration.

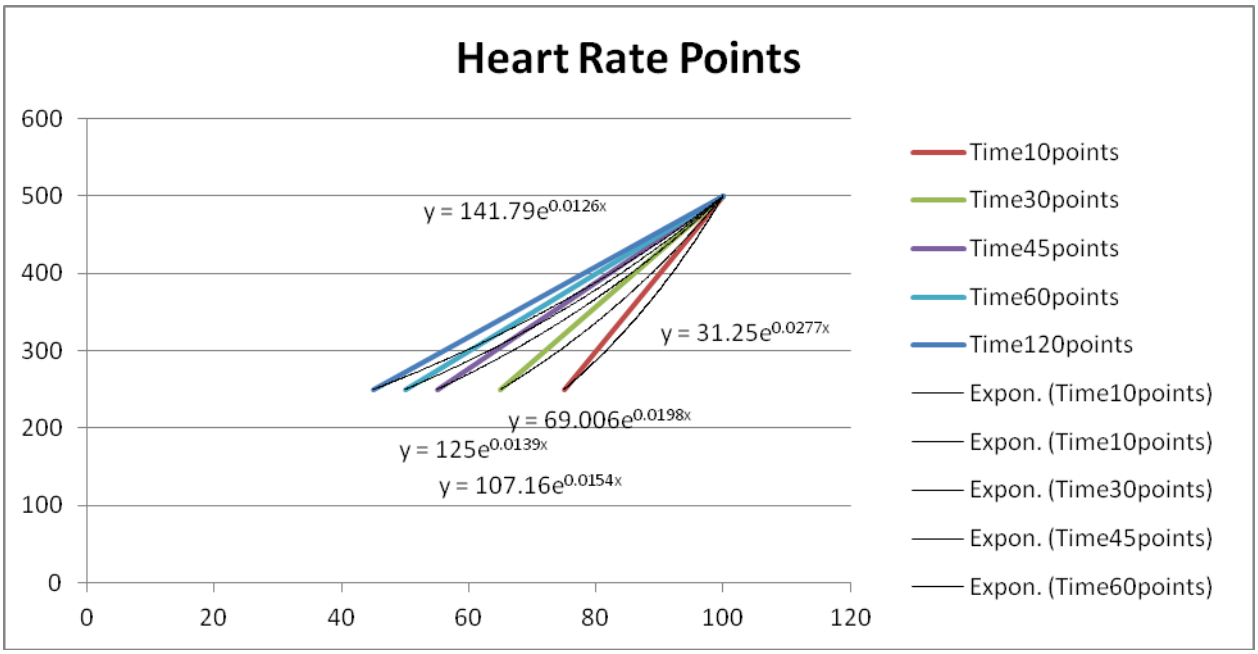


Figure 9-5: Heart rate points illustration.

Again, the challenger has a finite amount of options to choose from: 10minutes,30,45,60, and 120. As with the Speed challenges, it was deemed unfair to reward people more for being able to keep a higher heart rate on shorter events, and the model was adjusted accordingly, by making it exponential.

What makes this model more complicated is the method used to determine average heart rate. In the previous model, points were divvied based on how long one was in a particular zone. This method was open to cheating, where people can essentially play for any length of time and win without having to push himself or herself. The new method takes the maximum moving average for a period the length of the challenge chosen. This method would require the user to push himself for the entire duration of the challenge, as working out longer will more likely result in lower moving averages and not higher ones.

9.2 Data Structures

The server of Work out with friends, has two main databases built, one for user login and administration, and the other for store user performances. Both of them are build using MySQL on the server disk.

9.2.1 User Login Information store

The login information was stored as table entries in a separate table.

Table 9-1: Data structure for user login information.

DATABASE	TABLE	Values	Description
dbweb	user	Vchar(10) uid	User id used when login.
		Vchar(10) key	Password.
		Vchar(20) email	Email.
		Char(1) gender	M/F
		Int(11) age	Age.
		Char(1) access	U- user; A- system administrator.

9.2.2 User performance store

The user performance database was built with the following ideas:
- Construct one data for each use; during registration time.
- In each use there' re tables stores friends info and performance information for challenge.

Table 9-2: Data structure for user performance information.

DATABASE	TABLE	Values	Description
Uid(uer id used to login)	friends	Int(11) id	Index number starts from 1.
		Vchar(10) uid	User id used when login.
	perfor	Int(11) id	Index number starts from 1.
		Double tz4	Time spent in time zone 4.
		Double tz5	Time spent in time zone 5.
		Double eng	Energy consumption of the workout.
		Double scoreSum	Total score calculated from the math model.

10. User Interface Design and Implementation

10.1 User Interface Design

The main concept of user design interface was to provide for an easy , clear picture of wha the user can do on the website application .From the beginning , we have kept it clear that the user should be comfortable with the on screen appearance ,the working of the application.Hence forth , we had identified that the main points in keeping up with the user interface design were(given in the first report) :

1. Landing page: on Facebook Market App.

2.e – Login with Facebook account.

2.d – Registration: for new and 1st time users (some info should also be available in About Screen)
- 2.d.1 - part 1 - game description and how-to-play
- 2.d.2 - part 2 - disclaimers & permissions
- 2.d.3 - part 3 - user info (name, user name, gender, age, resting heart rate, weight, sports preferences, share stores (y/n), share location (y/n), alerts..)

2.a - Home Screen major building blocks:
1) high level status bar: (number of challenges: completed, waiting for opponent, won; weekly points, delta of wk-to-wk) - button takes users to "Performance Tracker screen" (2.j)
2) About
3) Start a new challenge
4) Open challenges
5) Challenge History
6) Help

2.b - Start a new challenge screen
- 2.b.1 - Search for facebook friends, random local opponent, search by username, select previous workout (self-trainer mode)
- 2.b.2 - (from select facebook friends) - populates list of FB friends, allow to select 1 or more opponents
- 2.b.3 - Pre-challenge screen: shows selected friends, asks users to input: challenge nickname, workout duration, options (web or mobile), message opponents

2.c - Open Challenge: function residing in Home Screen, lists open challenges waiting for opponents turn

2.f – Help: FAQ's, report user, send feedback to dev team, upgrade SW.

2.g – About: users can find more info (links) on HR, training and tips. Users can also access how-to-play and game description available at registration, Help (takes users to help screen) and Account (takes users to Account Management screen)

2.i – Account: users can review and/or edit account attributes such as: nickname, weight, resting HR, alerts, sharing options, age

2.j - Performance tracker: shows numerical values for challenges completed, winning %, margin won/lost, calories counter (toggle monthly vs weekly vs all-time view). Also users may select to graph such attributes

The main editions were made when we actually started implementing the project , and found that some of the requirements were making the project cumbersome , or difficult to implement with . We also found new ideas to implement , that would make the user interface much more easy .

The main thing was to start implementing the website ,before jumping to the facebook application .In the website , we recognized that the most important part of easy user interface was to have a "dashboard" or home screen with all the data the user would require as as soon as they logged in .The Dashboard consists of :
1.Home
2.User details
3.Challenges
4.Help
5.Find Friends
6.Contact us

Below is a brief description of each :
1.Home : the home screen reiterates to the dashboard , where the user can find all details pertaining to his/her use of the application .

2.User Details :
The user details give all details of the user, which had been inputted during the time of registration .These details can be edited at any point of time .

3.Challenges :
The challenges block provides for all the challenges the user has been in yet . It will include the scores ,who has won the challenge ,who has initiated the challenge ,and also which challenges are yet to be completed .

4.Help:
The help button on the dashboard will provide one with basic details on how to edit one's information , how to send out a challenge ,and how to invite new friends .

5.Find Friends :
The find friends button can help the user access any friend / or any member through their respective email .
The user can send invites to anyone to register in " workout with friends " ,and further on we will also make a way to invite friends present in one's facebook account .

Main Editions in User Interface Implementation:

TO FIND FRIENDS :
At the beginning the idea was to incorporate the idea of facebook friends , and keep the application within the limitations of facebook .However ,as we started working on the concept ,we decided that it would be better not to limit the application , and we started working on a website ,which we would later integrate onto a application .
Now , a user can not only add friends from one's facebook account , but also add anyone with an email id .

10.2 User Interface Implementation

As a web based application, the user interface is built using PHP/ HTML/JavaScript, and following are some main page from the application.

Figure 10-1: The home page

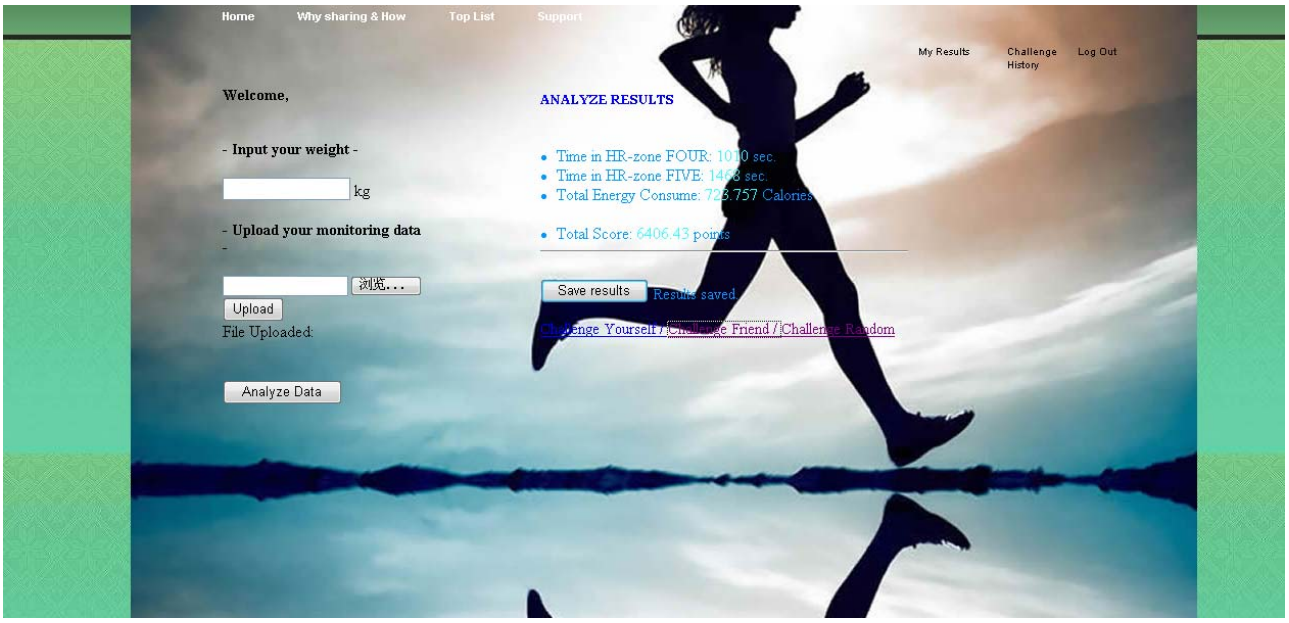


Figure 10-2: The data analyze page

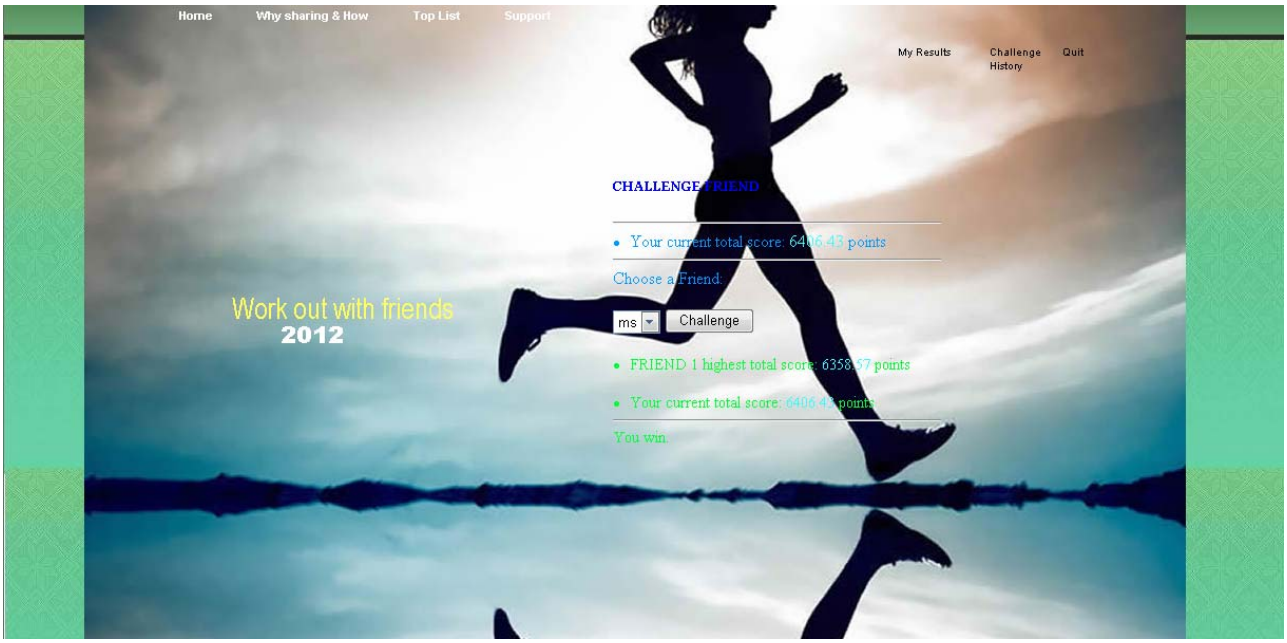


Figure 10-3: The friend challenge page

11. Design of Tests

11.1 System Framework Tests

Goal: To test the basic function of the whole application, the aim is to ensure all the non-critical functions, including user registration, log in, system administration, works smoothly in the system.

Testing approach: Register new users in the system and log in.

Results: All the parts worked well in the system.

11.2 Functional Units Tests

Goal: To test the running condition of all the central parts of the application: the data analysis and challenge parts.

11.2.1. Test ID: TC1_Register

Unit to test: User register.
Assumption: The application is run on the user register screen, and is waiting for the user' s action.

Input Requirement	Expected Output	Pass/ Fail	Comments
Valid information.	Register successfully.	Pass if shows the right display info.	
Register information incomplete.	Display error.	Pass if shows the right display info.	

11.2.2. Test ID: TC2_Login

Unit to test: User Login.
Assumption: The application is run on the login screen, and is waiting for the user' s action.

Input Requirement	Expected Output	Pass/ Fail	Comments
Valid userId, Password	Login successfully.	Pass if shows the right display info.	
Invalid userId, Password	Display error.	Pass if shows the right display info.	

11.2.3. Test ID: TC3_Fileupload

Unit to test: Monitor data upload unit.
Assumption: The application is run on the user main screen, and is waiting for the user' s to upload monitoring data.

Input Requirement	Expected Output	Pass/ Fail	Comments
Upload txt file.	Upload successfully.	Pass if shows the right display info.	
Invalid file type.	Display file upload error.	Pass if shows the right display info.	

11.2.4. Test ID: TC4_Dataanalysis

Unit to test: Data analysis unit.
Assumption: The application is run on the user main screen, and the user has uploaded a valid monitoring data for analyzing.

Input Requirement	Expected Output	Pass/ Fail	Comments
Start data analysis.	Show analyzed workout result.	Pass if shows the right analyzed info.	

11.2.5. Test ID: TC5_Challenge

Unit to test: Challenge unit.
Assumption: The application is run on the user main screen, and the user has already got the analyzed information from the uploaded monitoring data, and choose to start a challenge.

Input Requirement	Expected Output	Pass/ Fail	Comments
Start challenge.	Show challenge result based on the user' s analyzed workout result, and the results he/ she chose to challenge with.	Pass if shows the right challenge result based on the analyzed data.	

12. Project Management

12.1 Overall Project Progresses

Task \ Week		40	41	42	43	44	45	46	47	48	49	50
Website Maintenance												
Web Interface Design												
Mobile Interface Design												
Database Structure Design												
Algorithm Design												
Algorithm Test												
Debugging												
System Test												

12.2 Contribution Breakdown(Report 3)

Contribution Breakdown _ Report 3							
Task	**Mateus**	**Kevin**	**Brien**	**Abdul**	**Daihou**	**Sujana**	**Total**
Summary of Changes (5 points)	16.6%	16.6%	16.6%	16.6%	16.6%	16.6%	100%
Sec.1: Customer Statement of Requirements (6 points)	35%	25%	5%	-	30%	5%	100%
Sec.2: Glossary of Terms (4 points)	35%	35%	-	-	25%	5%	100%
Sec.3: System Requirements (6 points)	25%	25%	25%	-	25%	-	100%
Sec.4: Functional Requirements Specification (30 points)	25%	20%	20%	10%	25%	-	100%
Sec.5: Effort Estimation (4 points)	50%	-	10%	30%	10%	-	100%
Sec.6: Domain Analysis (25 points)	15%	35%	15%	-	35%	-	100%
Sec.7.a: Interaction Diagrams (30 points)	-	-	-	-	50%	50%	100%
Sec.7.b: Design Patterns (10 points)	16.6%	16.6%	16.6%	16.6%	16.6%	16.6%	100%
Sec.8.a: Class Diagram and Interface Specification (10 points)	-	-	-	-	-	100%	100%
Sec.8.b: OCL Contract Specification (10 points)	16.6%	16.6%	16.6%	16.6%	16.6%	16.6%	100%
Sec.9: System Architecture and System Design (15 points)	-	40%	-	-	20%	40%	100%
Sec.10: Algorithms and Data Structures (4 points)	-	20%	35%	35%	10%	-	100%
Sec.11: User Interface Design and Implementation (11 points)	-	-	-	-	45%	55%	100%
Sec.12: Design of Tests (12 points)	-	10%	-	-	90%	-	100%
Sec.13: History of Work, Current Status and Future Work (5 points)	50%	-	-	5%	45%	-	100%
Sec.14: References (-5 points)	50%	50%	-	-	-	-	100%
PROJECT MANAGEMENT (13 points)	22.5%	15%	15%	15%	22.5%	10%	100%
TOTALS (points)	**27.825**	**33.25**	**19.438**	**11.938**	**64.413**	**43**	**200**

13. References

1. **"Adult Obesity Facts."** *Cdc.gov*. Centers for Disease Control and Prevention, 13 Aug. 2012. Web. 08 Oct. 2012. <http://www.cdc.gov/obesity/data/adult.html>.

2. **"Apple Inc."** *Wikipedia*. Wikimedia Foundation, 10 May 2012. Web. 08 Oct. 2012. <http://en.wikipedia.org/wiki/Apple_Inc>.

3. **"Counting Every Moment."** *The Economist*. The Economist Newspaper, 03 Mar. 2012. Web. 08 Oct. 2012. <http://www.economist.com/node/21548493>.

4. **"Facebook Platform Policies."** *Developers.facebook.com*. Facebook, 12 Sept. 2012. Web. 08 Oct. 2012. <https://developers.facebook.com/policy/>.

5. **"Facebook Programming: Facebook API."** *Phpeveryday.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.phpeveryday.com/articles/Facebook-Programming-Facebook-API-P850.html>.

6. **"Facebook Programming: Facebook Platform."** *Phpeveryday.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.phpeveryday.com/articles/Facebook-Programming-Facebook-Platform-P845.html>.

7. **"Facebook Programming: My First Facebook Application."** *Phpeveryday.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.phpeveryday.com/articles/Facebook-Programming-My-First-Facebook-Application-P847.html>.

8. **"Google Code University: Android."** *Code.google.com*. Google, n.d. Web. 08 Oct. 2012. <http://code.google.com/edu/android/index.html>.

9. **"Heart Diseases & Disorders."** *Hrsonline.org*. Heart Rhythm Society, n.d. Web. 08 Oct. 2012. <http://www.hrsonline.org/Patient-Resources/Heart-Diseases-Disorders>.

10. http://www.Amazon.com -> search: **Xbox 360 Kinect combo**

11. **"Heart Rate Based Calorie Burn Calculator."** *Heart Rate Based Calorie Burn Calculator*. Shapesense.com, n.d. Web. 08 Oct. 2012. <http://www.shapesense.com/fitness-exercise/calculators/heart-rate-based-calorie-burn-calculator.aspx>.

12. **"Heart Rate."** *Wikipedia*. Wikimedia Foundation, 10 Aug. 2012. Web. 08 Oct. 2012. <http://en.wikipedia.org/wiki/Heart_rate>.

13. **"How a Heart Rate Monitor Can Improve Your Performance."** *Top10heartmonitors.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.top10heartratemonitors.com/how-a-heart-rate-monitor-can-improve-your-performance/>.

14. **"How to Live by the Numbers: Exercise."** *Wired.com*. Conde Nast Digital, 22 June 2009. Web. 08 Oct. 2012. <http://www.wired.com/medtech/health/magazine/17-07/lbnp_exercise>.

15. **"IPhone."** *Wikipedia*. Wikimedia Foundation, 10 July 2012. Web. 08 Oct. 2012. <http://en.wikipedia.org/wiki/IPhone>.

16. Kravitz, Len, and Lance Dalleck. ***Lactate Threshold Training***. University of New Mexico, n.d. Web. 8 Oct. 2012. <http://www.unm.edu/~lkravitz/Article%20folder/lactatethreshold.html>.

17. McClusky, Mark. **"The Nike Experiment: How the Shoe Giant Unleashed the Power of Personal Metrics."** *Wired.com*. Conde Nast Digital, 22 June 2009. Web. 08 Oct. 2012. <http://www.wired.com/medtech/health/magazine/17-07/lbnp_nike?currentPage=4>.

18. McGee, Marianne Kolbasuk. **"11 Telemedicine Tools Transforming Healthcare."** *Informationweek*. InformationWeek, 23 Mar. 2012. Web. 08 Oct. 2012.

<http://www.informationweek.com/healthcare/mobile-wireless/11-telemedicine-tools-transforming-healt/232602982>.

19. Parmar, Arundhati. **"Tie to Mayo Clinic Prompts Software Company down the Medical Device Path."** *Medcitynews.com*. Medcity News, 2 Apr. 2012. Web. 08 Oct. 2012. <http://medcitynews.com/2012/04/tie-to-mayo-clinic-prompts-software-company-down-the-medical-device-path/>.

20. Sinha, Alex. **"Heart Monitor Training."** *MarathonGuide.com - Heart Monitor Training*. N.p., n.d. Web. 08 Oct. 2012. <http://www.marathonguide.com/training/articles/HeartMonitorTraining.cfm>.

21. Terry, Ken. **"Forget Google Glasses: Meet Wearable Health Monitors."** *Informationweek*. InformationWeek, 12 Apr. 2012. Web. 08 Oct. 2012. <http://www.informationweek.com/healthcare/mobile-wireless/forget-google-glasses-meet-wearable-heal/232900190>.

22. Vecchione, Anthony. **"Health-Monitoring Devices Market Outpaces Telehealth."** *Informationweek*. InformationWeek, 01 June 2012. Web. 08 Oct. 2012. <http://www.informationweek.com/healthcare/mobile-wireless/health-monitoring-devices-market-outpace/240001352>.

23. Waltz, Emily. **"How I Quantified Myself."** - *IEEE Spectrum*. IEEE, Sept. 2012. Web. 08 Oct. 2012. <http://spectrum.ieee.org/biomedical/devices/how-i-quantified-myself>.

24. **"Wearable Wireless Medical Devices to Top 100 Million Units Annually by 2016, ABI Research."** *Wearable Wireless Medical Devices to Top 100 Million Units Annually by 2016, ABI Research*. Business Wire, 17 Aug. 2011. Web. 08 Oct. 2012. <http://www.businesswire.com/news/home/20110817006223/en/Wearable-Wireless-Medical-Devices-Top-100-Million>.

25. **"What Is a Heart Rate Zone? Why Do You Care?"** *Top10heartmonitors.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.top10heartratemonitors.com/heart-rate-zones-why-do-you-care/>.

26. **"Why People Stop Exercising."** *No-iron-fitness.com*. N.p., n.d. Web. 08 Oct. 2012. <http://www.no-iron-fitness.com/why-people-stop-exercising.html>.

27. Wolf, Gary. **"Know Thyself: Tracking Every Facet of Life, from Sleep to Mood to Pain, 24/7/365."** *Wired.com*. Conde Nast Digital, 22 June 2009. Web. 08 Oct. 2012. <http://www.wired.com/medtech/health/magazine/17-07/lbnp_knowthyself>.

28. http://www.Amazon.com -> search: **Motorola MOTOACTV 8GB GPS Sports Watch and MP3 Player - Retail Packaging**

29. http://www.Amazon.com -> search: **Nike+ SportWatch GPS Powered by TomTom (Black/Volt)**

30. http://www.Amazon.com -> search: **Bluetooth Heart Rate Monitor**

31. "**Timex Ironman"**< http://en.wikipedia.org/wiki/Timex_Ironman>

32. "**MotoACTV Web Portal.**"<https://motoactv.com/>

33. "**Training Zones.**"<http://www.thewalkingsite.com/thr.html>

34. **"Software Architecture Styles."**<http://msdn.microsoft.com/en-us/library/ee658117.aspx>

35. **"UMLet."**<http://www.umlet.com/>

36. **"Gliffy - Online Diagram Software and Flowchart Software."<**http://www.gliffy.com**>**

37. **"One Rep Max."**< http://www.shapesense.com/fitness-exercise/calculators/1rm-calculator.aspx >

38. **“Wilks Coefficient (Formula).”**<http://en.wikipedia.org/wiki/Wilks_Coefficient>

39. **“Wilks Coefficient.”<**http://www.marylandpowerlifting.com/wilks.asp**>**

40. **“Bench Press World Record.”<**http://www.bench-pressing.com/bench-press-world-records**>**

41. **“Track and Field World Records.”**
<http://www.trackandfieldnews.com/tfn/records/records.jsp?listId=1>

42. **“Plataeu Effect.”**<http://www.livestrong.com/article/350937-weight-loss-plateau-effect/>

43. **“Game Grinding.”**<http://www.examiner.com/article/what-is-grinding-grinding-video-games>

44. **“Plateau Effect.”**<http://www.livestrong.com/article/538500-what-does-a-plateau-mean-in-the-gym/>

Appendix 1

Database Schema

Table of contents

28 Relational schema

1 accepted_challenge_notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
id	int(11)		No			notifications -> id		

2 bench_press_challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
percentage	decimal(3,2)		No					
id	int(11)		No			challenges -> id		

3 bench_press_workouts

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
repetitions	smallint(6)		No					
id	int(11)		No			workouts -> id		

4 buddies

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
buddy_id	int(11)		No			users -> id		

5 challenge_types

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
name	varchar(40)		No					

6 challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
user_id	int(11)		No			users -> id		
challenge_type_id	int(11)		No			challenge_types -> id		
created_at	datetime		No					

7 denied_challenge_notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
id	int(11)		No			notifications -> id		

8 endurance_challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
duration	smallint(6)		No					
id	int(11)		No			challenges -> id		

9 endurance_workouts

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
heart_rate	smallint(6)		No					
calories_burned	smallint(6)		No					
id	int(11)		No			workouts -> id		

10 groups

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
name	varchar(40)		No					

11 invite_tokens

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
token	char(36)		No					
created_at	datetime		No					
expires_at	datetime		No					
user_id	int(11)		Yes	NULL		users -> id		

12 login_tokens

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
token	char(36)		No					
created_at	datetime		No					
expires_at	datetime		No					
user_agent	char(32)		No					
user_id	int(11)		No			users -> id		

13 new_buddy_notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
id	int(11)		No			notifications -> id		

14 notification_types

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
name	varchar(40)		No					

15 notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
notification_type_id	int(11)		No			notification_types -> id		
created_at	datetime		No					

16 password_tokens

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
token	char(36)		No					
created_at	datetime		No					
expires_at	datetime		No					
user_id	int(11)		No			users -> id		

17 requested_challenge_notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
id	int(11)		No			notifications -> id		

18 speed_challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
distance	smallint(6)		No					
id	int(11)		No			challenges -> id		

19 speed_workouts

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
speed	decimal(4,2)		No					
id	int(11)		No			workouts -> id		

20 squat_challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
percentage	decimal(3,2)		No					
id	int(11)		No			challenges -> id		

21 squat_workouts

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
repetitions	smallint(6)		No					
id	int(11)		No			workouts -> id		

22 uploaded_workout_notifications

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
id	int(11)		No			notifications -> id		

23 users

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
username	varchar(10)		No					
email	varchar(254)		No					
password	char(60)		No					
gender	enum('F', 'M')		No					
dob	date		No					
weight	decimal(7,4)		No					
height	decimal(5,4)		No					
timezone	varchar(50)		Yes	NULL				
avatar	varchar(40)		Yes	NULL				
is_active	tinyint(1)		No					
created_at	datetime		No					
last_active_at	datetime		No					

24 users_challenges

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
is_accepted	tinyint(1)		Yes	NULL				

25 users_groups

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
user_id	int(11)		No			users -> id		
group_id	int(11)		No			groups -> id		

26 users_notifications

Creation: Dec 15, 2012 at 02:23 PM

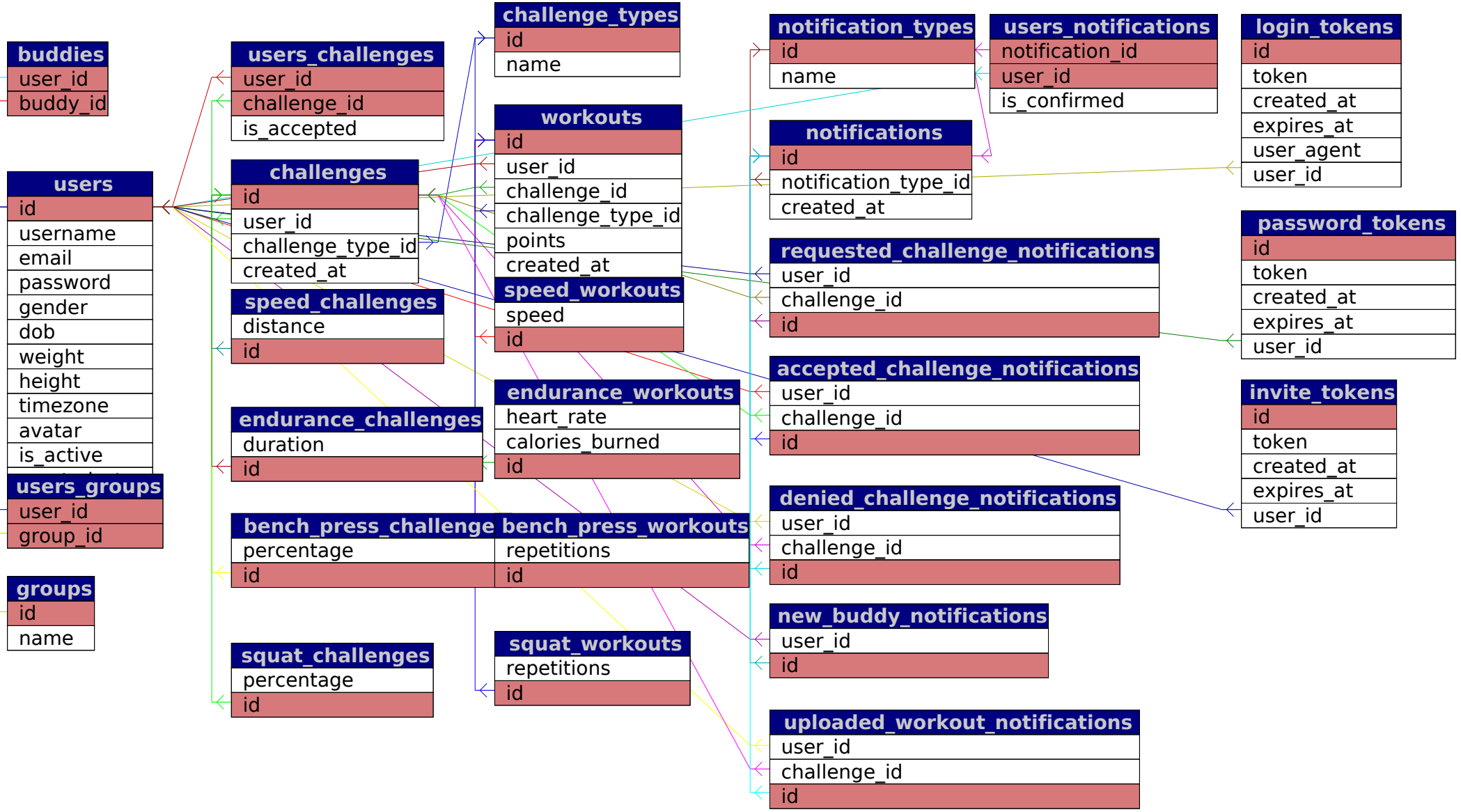
Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
notification_id	int(11)		No			notifications -> id		
user_id	int(11)		No			users -> id		
is_confirmed	tinyint(1)		No					

27 workouts

Creation: Dec 15, 2012 at 02:23 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
user_id	int(11)		No			users -> id		
challenge_id	int(11)		No			challenges -> id		
challenge_type_id	int(11)		No			challenge_types -> id		
points	smallint(6)		No					
created_at	datetime		No					

Schema


buddies
user_id
buddy_id
users
id
username
email
password
gender
dob
weight
height
timezone
avatar
is_active
users_groups
user_id
group_id
groups
id
name
users_challenges
user_id
challenge_id
is_accepted
challenges
id
user_id
challenge_type_id
created_at
speed_challenges
distance
id
endurance_challenges
duration
id
bench_press_challenge
percentage
id
squat_challenges
percentage
id
challenge_types
id
name
workouts
id
user_id
challenge_id
challenge_type_id
points
created_at
speed_workouts
speed
id
endurance_workouts
heart_rate
calories_burned
id
bench_press_workouts
repetitions
id
squat_workouts
repetitions
id
notification_types
id
name
notifications
id
notification_type_id
created_at
users_notifications
notification_id
user_id
is_confirmed
requested_challenge_notifications
user_id
challenge_id
id
accepted_challenge_notifications
user_id
challenge_id
id
denied_challenge_notifications
user_id
challenge_id
id
new_buddy_notifications
user_id
id
uploaded_workout_notifications
user_id
challenge_id
id
login_tokens
id
token
created_at
expires_at
user_agent
user_id
password_tokens
id
token
created_at
expires_at
user_id
invite_tokens
id
token
created_at
expires_at
user_id