Software Engineering
14:332:452
Spring 2013

Biometric Health Monitoring

Group 12 Members
Jose Villamor-Delgado
Cody Goodman
Jie Huang
Kyle Raucci
Florian Pranata

Class Description Table	
Class Name	**Description**
Account	Holds all of user's data and account information such as username/password, and health/fitness data
Account Settings	Provides functions for changing account settings and information, such as user password/email address and allows user to link their Social Networks
Analyzer	Contains functions that perform operations on data and generates pertinent statistics and graphical representations
Database	Contains a list of all accounts as well as functions for creating a backup and restoring user accounts
Data Display	Allows the user to view all their previously entered data and generated graphs as well as share them via email or social networking
Home Screen	Provides user to links to the other interfaces as well as allowing them to input new health/fitness data.
Login Screen	Verifies user login information

Class Attribute Tables

Class: Account	
Attribute / Data Type	**Description**
Username	String data type used for identifying user
Password	String data type used to verify user identity
Account Type	Integer data type used to specify whether account is of type: Client, Professional or Admin
Health Data	Various data types used to represent the user's health records
Fitness Data	Various data types used to represent the user's fitness records
User Permissions	List of strings used to determine which professional accounts my access the data of the current account.
Email Address	String data type to store user's email adress
Social Network (SN) Account	List of various social network accounts linked with the user's account

Class: Account Settings	
Attribute / Data Type	**Description**
changePassword()	Changes password to a new one specified by the user: requires validation of old password
changeEmail()	Changes account email address to new a new one specified by the user
linkSN()	Links account with social network accounts provided by the user
requestProfessional()	Take a professional account's username inputted by the user and adds it to the user's userPermissions list.

Class: Analyzer	
Attribute / Data Type	**Description**
analyzeData()	Performs operations on raw user input to generate refined information for the user
createGraphs()	Generates graphs depicted the user's raw input and information gained from analyzeData()
refreshGraphs()	Updates existing graphs with new input data, or specified parameters

Class: Database	
Attribute / Data Type	**Description**
Account List	List of all created instances of class:Account
createBackup()	Exports everything stored on the database to a secondary external copy
createAccout()	Initializes and stores a new account and adds it to the account list
storeData()	Writes all entered user input data to the database
deleteAccount()	Deletes an account from the database's memory and removes it from the list
storeGraphs()	Stores graphs generated by Analyzer, in the database

Class: DataDisplay	
Attribute / Data Type	**Description**
displayGraphs()	Displays a user's stored/generated graphs to the screen in an organized fashion
displayData()	Displays user's stored raw health and fitness data in an organized display
shareViaSN()	Posts data/graphs to a social network linked to the user's account
shareViaEmail()	Sends a link of the user's graphs/data to a user supplied email address

Class: Home Screen	
Attribute / Data Type	**Description**
accountSettings()	Sends user to the account settings interface
inputData()	Allow to input new raw health/fitness data and sends it to the database
viewGraphs()	Sends user to Data Display interface and calls the displayGraphs() function
viewData()	Sends user to Data Display interface and calls the displayData() function
uploadDeviceData()	Reads input data from health monitoring device and sends to database

Class: Login Screen	
Attribute / Data Type	**Description**
registerUser()	Sends new user information to the database for creation of new user account
verifyLogin()	Checks that the username and password combination entered by the user is valid
resetPassword()	Sends a link to the email address assigned to the username provided by the user that allows the user to change their password in the case that they forgot it.
userNameReminder()	Checks if there is a username associated with a user supplied email address and if there is sends an email reminder to that email address

Domain Concepts \ Class	Account	Account Setting	Analyzer	Database	Home Screen	Login Screen
Create Account	X			X	X	X
Manage Account	X	X		X	X	X
Insert Data	X	X		X	X	
Manage Data	X	X		X	X	
Create Graph			X	X		
Display Graph			X		X	
Display Data			X		X	

3. System Architecture and System Design

a. Architectural Styles

Our system uses the Client/Server style of architecture. This model provides our system with centralized data storage as well as allowing multiple user accounts access and provide data to the database. The client side of the system runs on the client's own machine such as a computer or mobile device. This side handles all the local procedures such as logging in and accessing the database. The server side of the system runs on a different, much more powerful machine. This side handles the data storage in one place for data analysis and processing. The client/server system also reduces the workload of client machines by having the server do all the data processing. This system also makes it easy to share user data with other users or groups of users. Updating the system algorithm, along with other software updates, is also made easier with the Client/Server style. However, a centralized system will generate high amounts of traffic for the server side of the system; thus we will need machines that can handle the workload being sent towards our servers.

b. Identifying Subsystems

The system is split into a client side subsystem and a server side subsystem. The client side consists of user interface and data input, while the server side system contains the information and database of each client's data. The client side uses a web browser to access the client side system (html, php), which allows the user to create account and to add personal information and data of their progress. The server side system takes the data that comes from the client side system and inputs into a database in MySql and stores the data.

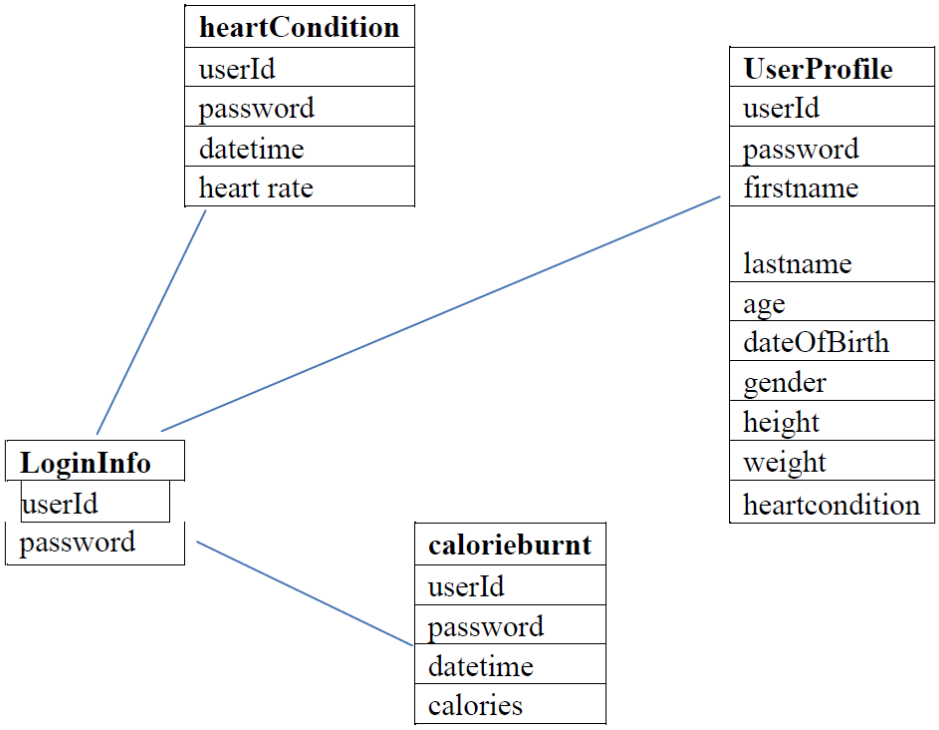
The server side allows the user to add, manipulate or adjust their information.

c. Mapping Subsystems to Hardware

There are two needed subsystems, a client side computer/phone and a server side computer. The server will store the database and information of clients while the client side computer will allow the user to access the programs and their information. The client side could run on multiple computers/phones and the server side can only be access by authorized personals.

d. Persistent Data Storage

The client's data must be stored in a database on a server in MySql. The user's information and health related information will be stored in an array in the database. The following shows the information the array will store of different clients.



e. Network Protocol

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

f. Global Control Flow

Biometric Health Monitoring is an event-driven system. What this means is that, after user creates his/her account, user will be able to do every function

implemented in the software in random order. After setting up his/her exercise routine, user can do any implemented function at any given time, such as edit their own exercise routine, ask for feedback from a professional, or share their exercise results to connected social network.

g. Hardware Requirements

The software will require the use of a monitoring device such as Metria Wearable Sensor. The mobile application will require a mobile device such as a phone or tablet that runs on the Android platform. The mobile application and web service will require an internet connection to access the social media aspect of the software.

4. Algorithms and Data Structures

A. Algorithms

We currently do not have or in the future create an algorithm for the program for it does not need or have an algorithm.

B. Data Structures

a. We currently have a database that has tables. The choice of having tables is due to the limited availability of choices. This choice was used because most data input from user is easier to be stored in a table. It is also easier for the program to take the data and output it to the user. Tables are also flexible since we could add additional rows or columns when needed.

5. User Interface Design and Implementation

5.1 Currently our design of the website as reported in report 1 is being coded in html, php and css. We have added additional information to the webpage so that if the user has any questions about the website could ask and send comments to the administrator. Up till now we have been following what we have intended to create on the website and will continue with the intended plan. We also added additional questions of information about the user so that the program could work more efficient and calculate a better return output to user. The additional questions include their country and ethnicity. We also added an additional option that allows the user to return to home page quickly and view their information. In addition we created a database to also record the client's ip address and time of date the user has logged in.

We currently have implemented the following items on the webpage:

1. Home
2. Login and registration
3. Input of user's data
4. Create a graph
5. Output of suggestions of dietary actions, eg. Exercise and diets
6. Logout and manage user information

6. Unit Test Cases

The purpose of the unit test cases is to test individual parts of the system's code. By breaking the overall code into smaller parts to test it is easier to more accurately and thoroughly test each part of the code, and if a problem is observed the smaller the tested parts are the easier it will be to find the source of the problem in the code. The unit test cases generally correspond to specific classes in the code or a specific class function (for more complicated functions such as sharing to social networks).

Test ID: TC1_Register User

Assumption: User is at Login Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
Valid Registration Info	Account Created	Pass if a new account with the user's info is created and saved to the database	
Incomplete User Entries	Error Message	Pass if error message is received and user is prompted to re-enter data	
Account information already in use	Error Message	Pass if error message is received and user is alerted that account already exists	Can be triggered by either having a duplicate username, or provided email address is already linked to another username

Test ID: TC1_Login

Assumption: User is at Login Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
Valid Username/Password Combination	Loggin Successful	Pass if user is directed to Home Screen	
Invalid Username/Password Combination	Login Failure	Pass if user remains at login screen and receives invalid username/password message	Repeated failures should result in account being temporarily locked

Test ID: TC3_Input Data and Analysis

Assumption: User is at Home Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
New Health/Fitness data	Data Updated	Pass if data is displayed at the Data Display Screen and is correctly analyzed, refined and organized	Testing will have to be repeated with various types and quantities of data

Test ID: TC4_Social Networking (SN)

Assumption: User is at Display Data Screen

Input Requirement	Expected Output	Pass/Fail	Comments
SN Account and Data	Data Shared	Pass if user's selected data/graph is displayed on the user's linked social network account	Test should be repeated for common social networks such as, Facebook, Twitter, Instagram, Tumblr, etc.

Test ID: TC5_Change Settings

Assumption: User is at Account Settings Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
Current Password & New Password	Password Changed	Pass if user account's password is changed from old password to new password.	Should only work if the old password provided is valid
New Email Address	Email Address Changed	Pass if user's linked email address is changed from the old one to the new one	Should only work if user provided a valid password
Social Network Username/Password	SN Account Linked	Pass if SN account information is saved to user's account	
Professional's Username	Professional Permissions Granted	Pass if user name provided is granted access to user's data if and only if the username provided is a "Professional" account	

Test ID: TC6_View Data and Graphs

Assumption: User is at Data Display Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
User request to view data	Data Displayed	Pass if all previously entered data is properly displayed at the Data Display Screen	
User request to view graphs	Graphs Displayed	Pass if graphs are properly displayed and correctly represent all previously entered data	

Test ID: TC7_Database Backup and Restore

Assumption: The database contains a populated list of user accounts.

Input Requirement	Expected Output	Pass/Fail	Comments
Database Backup Command	Database Backup Created	Pass if an External copy of the information contained in the database is created	
Database Backup	Database Restored	Pass if the internal database is restored to its previous state.	The database must be emptied (delete account list) then provided a backup copy to be restored from

Test ID: TC8_MonitoringDevice

Assumption: User is at Settings Screen.

Input Requirement	Expected Output	Pass/Fail	Comments
Monitoring device is turned on	Connection Successful	Pass if a connection between smartphone and device is established	Only works if the monitoring device is supported

Test Coverage

The test cases listed above should adequately cover all typical uses of the system.

Integration Testing

Our strategy for integration testing is to repeat the unit tests on the system as a whole. Implementation testing will consist of running through all the basic typical operations of the system in the following order.

1. Register User
2. Login
3. Input and Analyze Data
4. View Data
5. View Graphs
6. Link New Social Network
7. Change Password
8. Change Email
9. Request Professional Review
10. Share Data to Social Network

7. Project Management and Plan of Work

Contribution Breakdown

Section	Cody	Florian	Jie	Kyle	Jose
Interaction Diagrams	20	20	20	20	20
Class Diagrams and Interface Specification	100	-	-	-	-
System Architecture and System Design		12	88		
Algorithms and Data Structures	20	20	20	20	20
User Interface Design and Implementation			100		
Design of Tests	90	10	-	-	-
Project Management and Plan of Work			100		

A. Issues
   a. There was a problem with the php code that did not allow the webpage to access the MySql database - currently we are trying to figure out the problem with the code and to correct the implementation. (still in progress)
   b. The look of the webpage did not look as intended in the first report – we modified the css to make the images and code look as intended (fixed)
   c. The user was not allowed to correctly register – fixed php code that was receiving the input from user (fixed)

B. Project Coordination and Progress Report
   The following use cases have been implemented but have not yet been tested.
   a. Login
   b. createProfile
   c. createMontlyGraph
   d. insertData
   e. deleteData
   f. registerUser
   g. deleteUser
   h. displayTip

   We are currently working on the use cases that are included in creating a profile, user's password, and any cases that are related to creating a graph and to view it. We will also be working on the use cases related to the user's data in the future implementations.

C. Plan of Work
   February 05 - Research on monitoring devices
   February 06 - Website to be started
   February 12 - Statement of Work & Requirement
   February 16 - Start of App development
   February 18 - Functional Requirements Spec & UI

February 22 - Full Report # 1
March 1 - Interaction Diagram
March 8 - Class Diagram and System Architecture
March 15 - Full Report # 2
March 16 - Website launch
March 23 - App for android/apple device to be finished
March 24 - Start of testing and debugging
April 2 - Project Demo # 1
April 6 - Implement new changes to program (if necessary)
April 13 - Test and debug new implemented changes of program
April 27 - Full Report
May 1 - Project Demo # 2
May 3 - Finished Product

*Currently we are behind schedule about a couple of days but we will make up the few days and follow up to the intended schedule.

D. Breakdown of Responsibilities
    a. Login Jie
    b. createProfile Cody
    c. updateProfile Cody
    d. viewProfile Cody
    e. createMontlyGraph Jose
    f. viewMontlyGraph Jose
    g. insertData Kyle
    h. deleteData Kyle
    i. viewData Kyle
    j. registerUser Florain
    k. deleteUser Florain
    l. addUser Florain
    m. displayTip Jie

Each part created by the respective persons will be checked and tested by each member of the group to ensure the correctness and proper development of the modules and classes. As for coordinating the integration, it will be done by our team captain who will initiate the final process of integration when the classes and modules are completed and tested.

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) "Google Code University: Android." *Code.google.com*. Google, n.d. Web. 23 February 2013. <http://code.google.com/edu/android/index.html>.
3) "Heart Diseases & Disorders." *Hrsonline.org*. Heart Rhythm Society, n.d. Web. 23 Feb. 2013. <http://www.hrsonline.org/Patient-Resources/Heart-Diseases-Disorders>.

4) "Heart Rate." *Wikipedia*. Wikimedia Foundation, 10 Aug. 2012. Web. 23 Feb. 2013. <http://en.wikipedia.org/wiki/Heart_rate>.
5) McGee, Marianne Kolbasuk. "11 Telemedicine Tools Transforming Healthcare." *Informationweek*. InformationWeek, 23 Mar. 2012. Web. 23 Feb. 2013. <http://www.informationweek.com/healthcare/mobile-wireless/11-telemedicine-tools-transforminghealt/232602982>.
6) "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>.
7) Sun Microsystems, Inc. *Java Look and Feel Design Guidelines*. Mountain View, CA, 1999
8) "Pedometer" http://en.wikipedia.org/wiki/Pedometer
9) Health Care Monitoring of Mobile Patients: http://www.ercim.eu/publication/Ercim_News/enw60/amato.html
10) Palatini P, Casiglia E, Julius S, Pessina AC. "*High heart rate: a risk factor for cardiovascular death in elderly men*." Arch Intern Med 1999;159:585–592.
11) Heart Disease Fact Sheet: http://www.cdc.gov/dhdsp/data_statistics/fact_sheets/fs_heart_disease.htm