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.

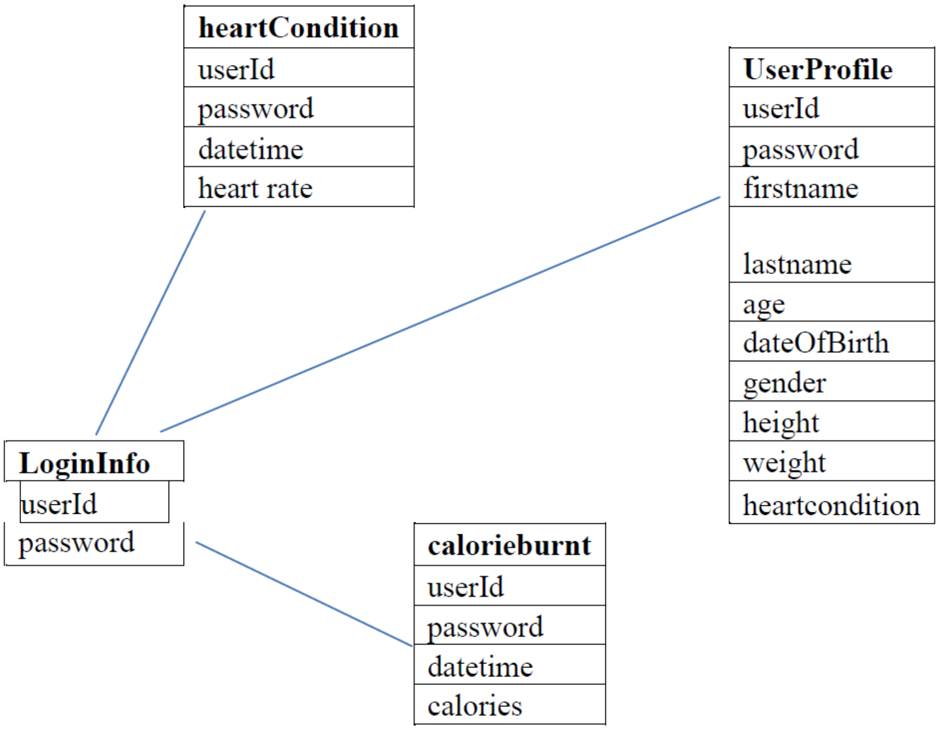
The server side allows the user to add, manipulate or adjust their information.

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.

Project Management

A. Issues and solutions

- there was a problem accessing the sql database, still under review
- problem accessing website and uploading proper files to make website work, fixed the issue by contacting server manager and used new way to ftp files through adobe dreamwork cs6

B. Project Coordination and Progress Report

- currently only the first use case is being implemented but having a few problems accessing and using sql database
- will work on user case 3 after fixing problems in user case 2

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

D. Breakdown of Responsibilities

- Jie will work on user case 2 in php and fix uploading and sql database, will also work on creating the interface of website
- Kyle and Florain will work on implementing the android version of user case 2 and 3
- Cody and Jose will work on user-case 3

References

“Software Engineering Project Report” Marsic. Rutgers University. 16 Feb. 2012. Web. 2. Mar. 2013. <http://www.ece.rutgers.edu/~marsic/Teaching/SE/report2.html>

"Unified Modeling Language" *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, Inc. 2 Mar. 2013. Web. 2 Mar. 2013. <http://en.wikipedia.org/wiki/Unified_Modeling_Language#Interaction_diagrams>