A Wellness Android Application with Social Networking Capabilities

Prachi Pendse  
Texas A&M University  
College Station, TX 77843  
prachirp@hotmail.com

Jalisa Greene  
College of Charleston  
Charleston, SC 29424  
jmgreene@g.cofc.edu

ABSTRACT
Wellness is an important issue and there is a large market of health-conscious individuals who try to eat more nutritious foods and exercise regularly. With cell phones increasingly integrated into our lives, it is not a new idea to use phones as a tool in encouraging people to be healthy.

This project focuses on developing a cell phone application to not only help people keep track of their health-related habits, but to use social networking to give users further motivation to stay fit and to empower them to spread their own wellness tips to a broad audience of similar users. The application also accesses cell phone sensors like GPS to actively interpret how locations the user frequents have an effect on health. This work is applicable to many pervasive information systems for health care, and this research will lead to advances in analyzing sparse, highly uncertain temporal and spatial data and its application to patterns of human behavior.

ACM Classification Keywords
J.3 [Life and Medical Sciences]: Health

General Terms
Documentation, Design

Author Keywords
Social networking, Mobile application

1. INTRODUCTION
We aim to investigate methods for achieving healthy lifestyle modifications. With the obesity growth rate being one of the most serious and prevalent public health problems of the 21st Century in the United States, there is a need to look into more significant intervention [1]. The objective of this project is to use social networks, behavior sensing and assessment, and automated yet personalized recommendations to create an Android application that influences its users to live a healthier lifestyle.

2. BACKGROUND
At first research studies focused on reaching people who lived in remote locations through mobile devices in order to better administer their health needs. “Mobile Phones: Changing Health Care One SMS at a Time” gives a survey of ways that cell phone messaging capabilities have been used by health professionals and their patients. These include treatment diaries, appointment reminders, self-assessment tests, social support networks, surveys, and as information services. While applications like treatment diaries and appointment reminders are useful only to those needing daily care and professional medical attention; social support networks can be helpful to a general audience looking for improved health and wellness. Social networking “can be particularly effective in programs to encourage smoking cessation, weight control or exercise, where there is substantial evidence that peer support among patients can improve adherence to therapy” [2]. Since it is settled that social networking is applicable to the type of holistic health and wellness application built for this project, communication with other users is a cornerstone of the application that is developed. The majority of research studies referenced in this paper are focused on the sexual health of people from developing countries like South Africa and Mexico, for this reason the cell phones used in the studies are generally simple and low-cost and the cell phone technology used only involves text messaging.

Other research that uses text messaging to monitor health, but is used in developed countries, involves a smoking cessation information service. “Do u smoke after txt?” is a cell phone application trial that took advantage of the ubiquity of cell phones as well as their constant connectivity to deploy intervention text messages to people who wanted to quit smoking all over New Zealand. In their trial, users text back and forth with a server that sends messages on a schedule and which replies with previously written responses to certain texts. The message bodies “covered information relevant to quitting”, “motivational support and distraction”, among others [3]. Those receiving the intervention service had better quit rates than a control group of participants who received no intervention. On noting the success of a service giving timely reminders to users, this project aims to integrate a notification service into its application.

Cell phone technology has improved since “Do u smoke after txt?” was published in 2005. Cell phones now have faster processors, more memory, and longer-lasting batteries. They also have many more sensors and accessible operating systems. Earlier generation phones housed an antenna and a microphone as their only sensors. In striking contrast, the iPhone 4S has a gyroscope, a compass, a proximity sensor, an ambient light sensor, an accelerometer, GPS and GLONASS, a multi-touch screen, and an HD camera, along with an antenna and microphone. It also has...
Bluetooth and Wi-Fi capabilities in addition to a cellular signal [4]. In terms of software engineering, “most of the smartphones on the market are open and programmable by third party developers and offer SDKs, APIs, and software tools. It is easy to cross-compile code and leverage existing software ....” [5].

The variety of sensors available on phones as well as the ease of access to device hardware through software interfaces in the phone operating system means it is possible to sense data regularly, learn and draw conclusions from that data, and to share those results with the user and the network. As an example of turning raw data into learning, “the variance of the accelerometer magnitude over a small time window could be useful for separating standing and walking classes” [5]. One aim of this project is to access GPS and network location data and project it onto a map to inform the user of nearby healthy dining options. The ambition of health sensing applications is to eventually become advanced enough that the person's cell phone may become “the hub of a body area network (BAN), a set of wearable devices or sensors on the body that monitor health-related parameters such as glucose or oxygenated blood” [6].

Health management research has tried to use advanced technology in smart phones to provide better health feedback to users. An instance of this is “monitoring user location and activity on the mobile phone, recognizing past behavior and augmenting the logging of blood glucose levels with context data” in a diabetes application case study [7]. This paper has users track what they eat and their physical exercise level on their phones, uses GSM signal strength to sample the user's location, and uses blood glucose levels inputted by the user from a personal glucose meter to provide a customized recommended insulin dosage for people suffering from diabetes. The information in this application is designed for access by a single individual; this is an example of a personal sensing application that uses both participatory sensing in having users input food intake and sugar levels and opportunistic sensing in automatically inferring the user's location using phone sensing [5].

Another instance of using sensors in smart phones is the “Dietary Intake Monitoring Application (DIMA) for dialysis patients”, which researched using bar code scanners in cell phones to let users monitor what foods they eat [8]. These researchers also investigated using networking by with members of a clique monitoring each other's activity level through the day in an application called Chick Clique and having them provide encouragement and motivation when needed. Theirs was a small network application that passively provided opportunistic sensing and operated on a group sensing scale, where users “want to share sensing information freely” [5]. The networking aspect of this project hopes to incorporate participatory sensing by having users log their health information on their phone as well as opportunistic sensing, where the phone will log the user's location automatically. The application being developed operates mostly on a personal sensing scale, with users self-monitoring health information, and has the option of being a group sensing application if users share information on achieving personal fitness goals with others in the health application's network.

Nowadays, “phones come with computing and communication resources that offer a low barrier of entry for third-party programmers.” [5] As a result, it is much easier to start developing phone applications and companies have tailored cell phone health applications to various purposes. Some, like Calorie Counter and Water Your Body, focus on eating right while others, like Workout Coach, RunKeeper, Daily Ab Workout FREE, Push Ups, and Noom CardioTrainer, target specific exercises [9, 10]. All of these applications are meant for a general audience as cell phone applications have become commercial ventures from the research programs that were devoted to improving the lives of patients with particular diseases. This project, too, focuses on improving public health and wellness and does not concentrate on any specific ailment.

When deciding on which phone system platform to develop on (i.e. iPhone, Windows, Android, Blackberry, etc.), special consideration was taken into what system made it easier for us to design. We decided to use Google’s Android OS (Operating System), which is a key platform for developing mobile enterprise applications [11]. For us developers, that meant using the Android SDK (Software Development Kit), writing in Java, and using an Eclipse IDE (Integrated Development Environment). Of course, we could have used our own favorite editors and tools, but Google provides a plug in to Eclipse that makes development much easier. Included with the Android SDK are a tremendous set of Java libraries, Apache classes for server connection, and World Wide Web Consortium classes for Document Model Object processing [11]. The Android SDK also comes with an Android Emulator, which is a virtual device that shows an image of an Android phone with a full OS running on which you can test any applications you make. The emulator uses your computer’s Internet connection, Bluetooth, etc. to work just as a phone would and even comes with a few applications and widgets you can explore. All of these features being available through Eclipse, a widely used IDE, made development tremendously easier and perhaps is why many refer to Android as “Linux, made easy”[12].

The main goal of this project is to give users a health application that shows they are not alone in pursuing healthy goals. In view of this, we have made an Android application aimed at allowing users to build a healthy social network to send and receive health tips and encouragement. This application accesses a web server and gives the user an account through which to message other users with the application. It is hoped that messaging others and getting replies about well-being will motivate users to take up and continue healthy habits.

To allow users to monitor their health habits, this application is being interfaced with two companies that make health electronics, Withings and Fitbit. Withings produces body scales that measure weight, fat mass, and lean mass. These scales are wifi enabled and transfer their data to company servers so that users with Withings accounts associated with scales can track their measurements online [13]. Fitbit makes pedometers that store the number of footsteps taken each day along with activity level and other counts. These pedometers can upload that data to Fitbit servers and users can access them through Fitbit accounts [14]. Both Withings and Fitbit have come out with APIs through which third parties can access user data. This application tries to use these APIs and pull user data from both companies so that the user can monitor measurements over time and see conclusive results from healthy lifestyle changes.

Our project also hopes to encourage normal people to eat healthier and to exercise more. We are working on developing an Android service to regularly remind people of their health goals and hope this service is incorporated into the application in the future.
3. METHODS
The application consists of two components, a client-side Android application and a web server (Figure 1). The Android side consists of XML files to display screens the user interacts with and Java files to send user input to the server and to perform basic program logic. The application starts with a login screen which takes the user to the main menu after a successful login, or gives the user the option to create a new account. The main menu starts a wellness service complete with buttons to perform various operations. The application will continue to run in the background of the Android device until the user closes the application.

![Figure 1 Client-Server Application](image)

The server side of this application runs on a LAMP stack, which stands for Linux, Apache, MySQL, and PHP [15]. It has an Ubuntu server operating system, an Apache HTTP server, a MySQL database to house user data, and PHP files to transfer data from HTTP format into the database [16]. Java files execute HTTP POST requests to PHP scripts whenever the user performs a major operation in the application. These PHP scripts perform MySQL database queries and print results [17]. The Java file then executes HTTP GET requests to receive the results and translate them onto the phone screen.

4. CLIENT-SIDE IMPLEMENTATION
We will begin by explaining the client-side GUI, which includes all information users can access.

4.1 Login
The user sees a login screen on startup of the application. If network connectivity is disabled the application prompts the user to turn on the phone's network or Internet connection, since this connection is necessary for the application to connect to the web server. The startup screen also prompts the user to start the phone's GPS in case it is turned off as well, and that screen looks similar to the acknowledgment of there being no Internet. On acknowledging these prompts, the application redirects the user to the phone's GPS settings or location settings screens so that the user can easily turn on networking or GPS as need be.

The login screen itself has options to either login using a pre-existing user name and password combination or create a new account. If the user tries to enter an incorrect user name or password, the application displays a note saying what went wrong during login. No network connectivity when the user tries to login, even after checking and prompting the user to connect the phone to the Internet, also results in an error message. The application displays a main menu screen when the user logs in correctly.

4.2 New Account
The new account screen has the user enter a first name and last name, and pick a user name and password. It checks that the user name and password are 20 characters or less, since that is the maximum string length that can be stored in the database. The application tells the user to pick another user name or password if either is too long. It also checks with the server that the user name in question is not already taken and prompts the user to try another username if it is taken. Once the program verifies that all necessary constraints are valid, the server enters a new user into the database and the application takes the user to the main menu screen. There is also a back button that goes back to the login screen.

4.3 Main Menu
This screen (Figure 2, left) has buttons for all the application's functionality. When returning users log in, the main menu displays a simple welcome greeting at the top, but when new users create an account and login for the first time, the main menu has a special instruction screen (Figure 2, right) that directs the user to initialize important account information.

![Figure 2 Usual and Initial Main Menu Screens](image)

Research is in progress to have a wellness notification service start to run in the background when users log into the main menu as well. This service will check the database for new messages to the user, send notifications to the user at preset times as reminders to eat healthy and exercise, and ask if the user requires assistance from the application social network. At this current time, the application is running an alarm service in the background that will execute commands to check the database for incoming notifications from the web server itself, but not incoming messages from other users.

Most buttons on this screen lead the user to a different display to give more input so as to perform more complex database operations, but some, like the Logout button, do a simple database operation and do not involve a new screen.

4.4 View or Change User Information
This button leads to a screen (Figure 3, left) from which the user can control information from Withings and Fitbit accounts through one application. On new accounts the input boxes on this
screen are blank and the user has the option to enable or disable either Withings or Fitbit options. On accounts with previously stored information, the user's current settings are displayed on the screen so that the user does not unnecessarily change account information. When the user enables the Withings or Fitbit check boxes, the application asks for the user name and password combination for the user's account with that company so that the user can automatically login to the account and access that interface through the wellness application. On clicking the “Done” button, the application updates all the information in the database using input from the screen and displays that input on the screen the next time the user wants to view or change the information.

**Figure 3 Withings and FitBit Interface Options**

### 4.5 Connect to Withings and FitBit

The Withings screen (Figure 3, right) and the program logic behind it, are samples taken from the Withings API. Due to lack of resources, this application was developed without access to any Withings hardware. Thus the compatibility of this application and this screen with Withings equipment has yet to be tested. Currently, the screen prompts a “No users on account.” warning when it starts. Even though this screen and the java files associated with it are taken from the company software website, connecting to Withings servers is complicated. Third-parties wishing to do so need to register their app with Withings, save public and secret application keys, implement OAuth authentication, and perform HTTP posts to Withings [16]. This application may be using the Withings API improperly and may not be connecting to Withings servers. Investment in a Withings body scale and further investigation into Withings accounts should solve any issues associated with this screen.

The Fitbit API is similar to the Withings API in that it requires registration with Fitbit as well as OAuth authentication. Although the process of connecting to Fitbit servers is like that of connecting to Withings servers, the Fitbit API is currently in beta and the company software website does not offer any example mobile application of its API. This, along with lack of access to Fitbit pedometers, hampered the development of an interface or screen between the application and the user's Fitbit account. Hence, clicking the Connect to Fitbit button does not elicit any response from the application at the moment.

### 4.6 Message Another User

This button leads to a simple screen (Figure 4, left) that asks the user to type a message and provide the user name of the user to send a message to. Sending the message means running a PHP script that puts the message and the sender's user name into the database record for the receiving user. The next time this user logs in, he or she will receive a prompt on the main menu with the contents of message and who it was from with the options of simply acknowledging the message or of replying back with a message to the sender (Figure 4, right).

Further research could lead to auto-fill for the inputting of a user’s name, and making sure that these two users are connected to each other in some way before sending the message.

**Figure 4 Send and Receive Message Screens**

### 4.7 Message All Users

The screen that this button shows simply asks for a mass message to send. Mass messages are sent to all users that are currently logged in. A user wishing to stop receiving mass messages can formally log out of the application by clicking the Logout button on the main menu.

### 4.8 Logout

This button logs the user out and displays the cell phone's home screen. This means that the user in question stops receiving messages aimed at all users. Messages meant only for him or her are still received and the user can see them on the next login. Users can navigate out of the application's screens without logging out of the application, and doing so will allow the application to run in the background and the user will continue to receive both mass messages, directed messages, and notifications.

### 5. SERVER SIDE IMPLEMENTATION

As stated earlier, we set up a local Apache server on campus. Having the IP address of that server allowed us to connect to it anytime we were on the same network.

### 5.1 Database Setup

We set up the database for the application, which contains four tables: users, location, kCal, and weight. The users table contains fields for a user ID (which automatically generates), first name, last name, unique user name for login, password, Withings Body Scale website user name and password, and the user's phone.
number. The user ID from the this table was linked to each of the other tables playing the role of the primary key. The users table also contains columns that store unread messages for a user, or a notification from the server to broadcast to the user.

![Database Schema](image)

**Figure 5 Database Schema**

The location table holds the locations for where the users have been, and saves the longitude, latitude, and date/time. We actually used the Android device’s GPS capability to track where the user has been, and to suggest healthy places to eat in those areas. The phone’s GPS records the longitude and latitude of the user’s current position and inserts this information as text into a pre-inscribed 3rd party website link. The application then opens a web browser and goes to this link, which shows healthy dining options in that area. We included this functionality in our original design of the application, but did not give the user access to it in our ending design for further testing purposes. The big challenge in this feature is that the GPS functionality is not as accurate as it could be.

The weight table stores the weights for the users filtered by user ID. This table will update from the Withings Body Scale servers. The kCal table will eventually store how many calories a user has eaten after certain meals of the day. The server currently sends out reminder messages for the user to enter into this table around breakfast, lunch, and dinner time. The main idea about this functionality was to record this information and send out specific messages to the users based on how many calories they have eaten during the day. The table can hold the information, but the application currently does not use it.

### 5.2 Cronjob

Cron jobs were set up on the server using Cronjob, a time-based job scheduler. These Cron jobs run PHP scripts at specific times of the day to switch the server message (located in the users table) that will be sent out to all of the users. We also set up an alarm service on the application itself which wakes the phone up (if asleep) and sends a broadcast to a simple broadcast receiver. The receiver calls an Asynchronous task that runs in the background of the Android device. This task connects to a PHP page on our server that returns the server message that should be sent to the user’s device as notification. Because this task is in the background the notification can be fetched and displayed even if the application is not visible on the device’s screen. The notification appears on the status bar originally and can be pulled down to see the entire text. When the notification is pulled down you can see the text and who the message is from. When clicked on, the notification will take you to the message screen within the Wellness Application no matter what screen you were previously using.

![Server Notification](image)

**Figure 6 Server Notification Sent at Specified Times**

### 6. CONCLUSION

With further development this application can give users real benefits by enabling them to conglomerate many health and wellness activities into one interface. The networking aspect of this application will improve user confidence in pursuit of a better lifestyle and will help users with motivation issues in continuing to pursue improved well-being. The monitoring aspect should provide users satisfaction by giving users objective evidence of the effect of their health pursuits. Its’ messaging and notification features set it apart from most healthy-lifestyle applications because users can message everyone registered with the application without having to be friends with them or knowing their user names. Also, the notifications from the server will be rich encouragement for the users because they do not have to have the application on their main screen to receive them and they are very convenient to read. Additions like a message notification service have great potential to keep users on track day after day. We would highly recommend such an application for those looking to improve their wellness.

### 7. ACKNOWLEDGMENTS

We would like to thank Dr. Homayoun Valafar for giving us structure in our work and checking up on all of us as participants of the program. He also provided the web server we used for the application and allocated space in his lab for the REU students assigned to the project. We would also like to thank Mr. Arjang Fahim for helping us set up the server and for offering programming and software development advice when needed. We would also like to thank Dr. Michael Huhns and Dr. Turner-McGrievy for providing the inspiration and direction for this project.

This work was done at the University of South Carolina as part of the Research Experiences for Undergraduates Site: Multi-
Disciplinary Research in Secure Computing project and supported by the National Science Foundation. Acknowledgements also go out to Lewis Cawthorne, the REU program coordinator, and to Dr. Caroline Eastman and Dr. John Bowles for leading program activities.

8. REFERENCES


