Health Monitoring Network Project - Design and First Steps of Implementation

Regina Dorokhova
St.-Petersburg Electrotechnical University
St.-Petersburg, Russia
reginadorohova@gmail.com

Suraj Agarwal
PES Institute of Technology,
Bangalore, India
surajagarwal89@gmail.com

Abstract

There are lots of health monitoring systems and social networking platforms, but there is no joint solution for sharing fitness/diet data. We propose a new open source **Healthnet** project for social networking around health, and personal improvement. This paper describes the first experience and general architectural decisions.

Index Terms: social networking, health monitoring, mobile development, symbian, Maemo/MeeGo, healthnet.

I. INTRODUCTION

Nowadays we can notice a great number of fitness clubs with their proposals, almost everywhere. Newspapers, Internet and TV are full of advertisements promising their help in loosing weight and developing figure. The problem of extra weight is one of the most discussed problem, and lots of people get troubled. It is not by chance. More persons have to live sedentary life and as a result they have great problems with their health. The most noticeable problem is obesity giving serious disturbance of health and psychological discomfort in the society. It is the main reason for buying products for slim figure. One of the ways in attaining the necessary physical health is calculating consumed and burned calories. More and more program products are appearing now. They offer different kinds of weight control.

We consider such products as FitDay, myfitnesspal, Calorie count and compare them. The result of comparison is in Table I.

Examining the present program products we should note their similarity. Table I shows the most developed ones we have found. As you can see their functionality is identical. We could not find any other schemes with the open original code. Integration with the social networks in found projects is not enough realized. Only one of the considered schemes is integrated with Facebook. The next important problem for all the same projects is lack of clear interface and lots of advertisements. The principal platforms for mobile clients are iPhone, Android and Blackberry.

II. HIGH LEVEL DESIGN

The healthnet is based on client server architecture and contains next components:

- *Data management infrastructure* is targeted to gathering, storing and providing access for all data in system. Below, the first relational data model is described;

TABLE I
COMPARISON SOME DIET SYSTEMS

Estimation criterion/ Program product	FitDay	Myfitnesspal	Calorie Count
Website	www.fitday.com	www.myfitnesspal.com	http://caloriecount.about.com/
Price	Free/$5.49/ Month(for Premium version)/$29.95 (for PC version)	Free	Free
Mobile client platforms	iPhone	Iphone, Android, BlackBerry, Windows Phone	Iphone, Android, BlackBerry
Integration with social networks	-	-	Facebook
Open Source	No	No	No
Functionality			
Track food	Yes	Yes	Yes
Track activity	Yes	Yes	Yes
BMI calculation	Yes	Yes	Yes
Product energy calculation	No (in free version)	Yes	Yes
Calculation burned calorie through activity	No (in free version)	Yes	Yes
Reports	Yes	Yes	Yes
Set target	Yes	Yes	Yes
Talk with other users	Forum	Forum and personal messages	Forum and personal messages

- *web front-end* will be used as standard lightweight user interface;
- *Mobile assistances* will help end users track their activities and product consuming. Currently, diet and activity monitoring are implemented. In future we will implement support of training programs, location based service integration.

Only open source technologies are being used for the project. This rule will guarantee involving more users and programmers in product testing and development.

III. DATABASE IMPLEMENTATION

A. Table Structure Description

We followed a tabular approach to store all the data relevant to the user to monitor its progress over a period of time and to facilitate effective diagnosis. The data was organized into several tables to provide easy querying and managing. The various data that we maintained about the user are age, gender, height, weight, fitness plan, diet plan, waist size, state of health, etc.

The above mentioned data was divided into 7 tables in order to serve the sole purpose of keeping the related data together and separated from the unrelated data.

The diagram (Fig. 1) depicts the attributes of each table, data types of the attributes and interrelationships between the tables.

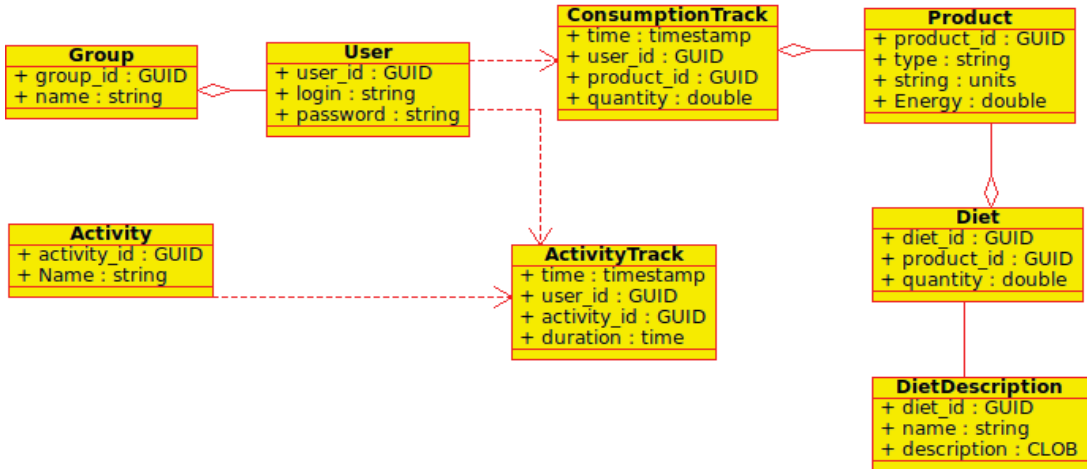

Fig.1. Database structure

B. Substantiation of Chosen Database Engine

We chose to use PostgreSQL as the backend database for storing the data pertaining to the user. Below are the reasons why we felt Postgres scores over other databases.

- It is free and runs on many types of UNIX .
- It is fast, stable and very well complies with SQl standards.
- It's a nice system and easy to learn.
- It has profound transaction support.
- It has robust authentication mechanism.

C. Different ORM Model Analysis

Object-relational mapping is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

The most commonly available ORM solutions for postgresql are listed below

Database Template Library – The goal of this library is to make ODBC records look just like an STL container. The abstraction can run on multiple platforms and C++ compilers. The library's compliance with the STL iterator and container standards means you can plug our abstractions into a wide variety of STL algorithms for data storage, searching and manipulation. In addition, the C++ reflection mechanism used by this library to bind to database tables allows us to add generic indexing and lookup properties to the containers with no special code required from the end-user. Because the code takes full advantage of the template mechanism, the library adds minimal overhead compared with using raw ODBC calls to access a database.

SOCI - SOCI is a database access library for C++ that makes the illusion of embedding SQL queries in the regular C++ code, staying entirely within the Standard C++. The idea is to provide C++ programmers a way to access SQL databases in the most natural and intuitive way.

For the purpose of our project, we implemented our own code to achieve object relational mapping. We wrote C++ classes and functions that facilitated mapping the

database objects to appropriate class members. Additionally, we provided class functions like insert, select, delete and update to simulate the equivalent database functionality.

Thus, for each of the 7 tables in the database, we had 1 class corresponding to the table. The data members of the class corresponded to the table attributes. Further, each class had a set of functions to facilitate database operations.

The C code makes use of UnixODBC interface and libpq library to access the postgresql database. These modules provide the interface to connect to the postgresql database and access/query the database. UnixODBC provides open database connectivity to gain a handle to the database. Libpq is used to manage querying the database

Thus, the functionality to manage the people's health record is implemented using the set of libraries and interfaces which are accessed by our code.

IV. Mobile client for Symbian/MeeGo

This application provides the user to control his/her physical activity, diet and weight. Using the application, a person is able to calculate ideal weight, the value of protein, fat, carbohydrates and calorific value as well. The user can also determine whether it is rational nutrition or not and define if he or she has normal weight or not by means of BMI (The body mass index). The application informs a person about necessary kcal and consumed ones. The user has an opportunity to edit databases of products and physical activities. The main application window is shown in Fig. 2.

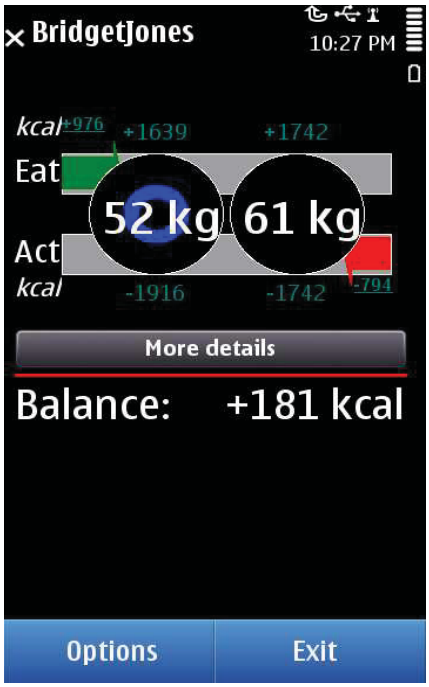

Fig.2. Main application view on Nokia N8

You are able to find this application via the Ovi Store on-device client or at these URL [1, 2].

A. Use Cases

Bellow we consider the application functionality in more details.

We have listed use cases which define application behavior:

"Edit the personal data" — the user is able to set or edit the follow information: weight, height, age at the last birthday, gender (male or female), living style.

"Track food and activity" — application displays how many calories the user should eat and burn in a day to achieve desired weight or maintain existing weight. Also the user can see eaten calories, burned calories and balance between them. It is extended by precedent «Track nutrition value».

"Track nutrition value" - it's possible to view how much fats, proteins and carbohydrates are consumed, how to achieve desired weight or to maintain existing weight and to define whether it is rational nutrition or not.

"Calculate some product" - the user can calculate energy and nutrition value of some products, and if it is necessary to add a new product and information about the content of fat, protein, carbohydrate and caloric content in database.

"Calculate some action" - the user can calculate amount of the calorie burned as a result of performing some actions. It is possible to add a new action in database.

"Do record in journal" - It allows to record information about activities and eaten products.

"Set target" - The user can descry his or her ideal weight, view his or her last target weight and set new target weight.

"Get BMI" -the user can get the body mass index (BMI) and define his or her weight category.

"Get help" - The application is fully manual.

B. Architecture overview

The mobile client architecture consists of the following items:

1) database
2) The module of interaction with database - DbSession
3) Calculating component - Calculator
4) GUI

The general architecture and data flows without its main components can be seen in Fig. 3.

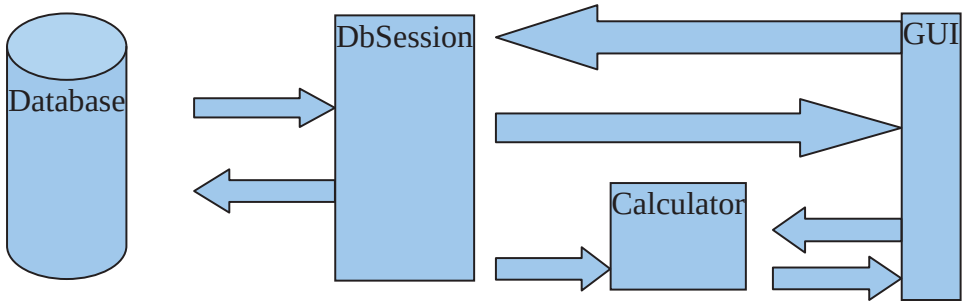

Fig.3. Main components of the architecture and data flows

Database is a collection of several xml files: basePersonal.xml (serves for personal data storage), baseProduct.xml (product database storage), baseAct.xml (kinds of activities database storage), Journal.xml (user's consumed products and done exercises storage).

The component DbSession is responsible for interacting with the database - getting data from database and writing data to the database. DbSession is mediator between database and other components. Component calculator is responsible for the primary calculations associated with the diet and fitness area. It gets data from database through DbSession and from GUI, then performs the required calculations and returns the results in GUI.

C. Development Tools

Nokia Qt SDK was used for development. Remote Compiler was used to assemble software package. It allowed building package for following platforms: Symbian^3, Symbian^1, S60 5th Edition, S60 3rd Edition Feature Pack 1, S60 3rd Edition Feature Pack 2, Symbian^3 v0.9, MeeGo 1.2 Harmattan, Maemo 5.

Testing had 4 stages:

1) On desktop.
2) The usage of emulator, supplying with Nokia Qt SDK.
3) Remote Device Access represented by Forum Nokia.
4) Real device – Nokia N8.

To debug an application on real device we used TRK debugging application.

V. CONCLUSION

This work is only in preliminary stage of development. We discovered market needs, did comparison of set similar products, specifies thir weak and strong characteristics. As the first step (which is done) the product architecture has been developed. The short-term future plans are:

- building web front end;
- extending data domain by adding health developing courses;
- adding geo awareness.

In the log term period we will add more social networking features, internationalizations, and desktop/mobile seamlesness.

REFERENCES

[1] Bridget Jones application in OVI. Nokia mobile browser: http://store.ovi.mobi/content/183022
[2] Bridget Jones application in OVI. Fixed web browser: http://store.ovi.com/content/183022
[3] https://wiki.ubuntu.com/DocumentationTeam
[4] http://www.postgresql.org/docs/