Nanyang Technological University

School of Computer Engineering

Interim Report

Title

Author:
Tuan Phong Nguyen

Supervisor:
Assoc Prof Chiew Tong Lau

March 6, 2016

1 Introduction

Along with the development of more compact and portable platform for measuring health related data, there are many products on the market that are capable of reading and sending data to a computer system. These devices are often built ready for communicating with mobile devices via Bluetooth to exchange data so that those data can be displayed visually. However, a universal platform to collect and display the data to users has not been implemented yet. On the other hand, the measured data is not frequently recorded so that it can be used as reference in diagnosing or study. This project aims to implement such a platform with consideration on capability of adapting multiple types of devices, data sharing as well as the scalability.

2 Architecture design

The application contains 3 main components namely the client, the backend server and the database.

2.1 Client side

The client is responsible for displaying the data acquired from the physical measuring devices visually as graph or reading listing. The client application needs to acquire the the data records from the backend server through network communication and cache it in local database for presenting to the app user. In order to server as a universal interface to various types of medical record, the interface is capable of displaying different types of graph for different record type and designed to be extended when there is need for a new representation of data. The client connects to the back-end to interact with other users of the application to share and access their data in the database. The client also provides the user with an interface to search for another users of the application and interact with them. Users can subscribe to another user to view their data and receive notifications upon certain events are triggered such as when new data is uploaded or when the reading is beyond safe range. As the database is accessed through a RESTful API via HTTP request and response, the client application can be implemented in any platform e.g. Android, iOS, native desktop program or web application. Due to time constraint and man power of this project as well as the extensibility to

other projects, the client is only implemented for android phones.

2.2 Server side

The server is a Web server that exposes a RESTful API to the clients and serves the incoming requests. To perform the business logic, the server interacts with a persistent database to provide the service as application backend. To elevate scalability of the application, the server side is chosen to be hosted on Google App Engine cloud service. By using cloud computing from a credible service provider, the application aims to minimize down-time and bottleneck comparing to self-hosting service on a single computer. In addition, hosting application on a cloud platform also enables the application cost to scale better with the number of users and data traffic since the service provider offers multiple pricing tiers for different amount of traffic. Moreover, Datastore database provided by Google also supports caching and reliably distributes data to different locations with minimal latency. Regarding implementation of ther server program, since client and server communicate through a RESTful API, the server can also be implemented in any programming languages. In this project, the web server is implemented in Java with Google Endpoints API. This helps promote code reuse by sharing Java object between client and server codebase. In the future, the server can be swapped out by any web server if there is need for finer control or performance improvement.

2.3 Database

Google Cloud Datastore is chosen to be the database backend of the system to reliably store and deliver data. Google Cloud Datastore is a schema-less NoSQL Datastore providing robust, reliable storage for the web application. In constrast with SQL-like database, Datastore treats each record in the database as a key-value pair without unnecessary indexes. Comparing to the traditional relation databases, the Datastore uses a distributed architecture to automatically managing scaling to very large data sets. Additionally, Datastore is hosted as part of the Google App Engine platform, hence is fully managed with no planned down time by Google. Due to its different way of representing and managing data, the Datastore can be easily scaled, allowing the application to maintain high performance as the traffic is increased.

3 Client side

3.1 Overall design

The android application is implemented with the following components:

1. SignInActivity that handles signing in of users
2. MainActivity that serve as the hosting activity for different views of the application with a Navigation drawer panel to allow users to switch among the views
3. Google Cloud service that handles the Google Cloud Messaging communication to listen to pushed messages from the servers and redirect them to appropriate components
4. Database services that provides method to access the logic models uses in the class

3.2 Graph display

Displaying of graph is contained in GraphFragment class.

4 Work done so far

4.1 Server side

1. Implemented a RESTful API for different features on the application including:
2. Subscribing and Un-subscribing
3. Posting and Retrieving of Data
4. Register of new users and devices
5. Sending Notification messages to users via Google Cloud Messaging

4.2 Android development practices

In order to approach the latest change in new Android version and Android SDK, the project aims to use the latest tools and practices suggested by Google. The client side application was made with consideration about design and components choices.

4.3 Software engineering practices

Different standards in software engineering was considered and followed throughout the development of the application so that the code can be read and reused by other developers.Serious consideration and effort has been put in the designing the graph components such that the application can be easily extended to other types of data i.e. blood pressure, blood sugar etc. as well as different ways of viewing data in the future.

5 Future work

Although the core codebase has been completed, there are space for improvement for this project.

5.1 Features

1. Notifications pushing depending on the data flow to make the application more user-friendly
2. Implementing display view of different types of data
3. (Optional) Connecting with real measuring devices to complete the full-stack solutionImplementation

5.2 Implementation

Fixing of remaining bugs in the application

5.3 Software engineering practices

Refactor if necessary to have a clean and loosely coupled code base so that it can be reused and integrated with other projects

6 Conclusion

In conclusion, the core code base of the project has been completed but there are still features to be implemented to provide a better interaction with the users.On the other hands, consideration in software engineering practices and latest change of Android library is also required throughout the process of development for the rest of the project. [1]

References

[1] J. Doe, *The Book without Title*. Dummy Publisher, 2100.