Data Communication Group

Daniel Davis Saari
Software Engineering I
University of Massachusetts Lowell
Lowell, Massachusetts
danielsaari@gmail.com

Cameron Morris
Software Engineering I
University of Massachusetts Lowell
Lowell, Massachusetts
zephyrz2712@gmail.com

***Abstract*—This document discusses the research and design of the data communications for the cloud platform to be developed in Software Engineering I**

I. Introduction

The goal of this project is to provide an API to be used with mobile devices and the Microsoft Kinect to transmit data to a cloud based database running MySQL. This API will be implemented both in C++[1] and in Java[2] using the connectors offered by MySQL. The API will transmit data in an encrypted form using HTTPS.

II. Motivation

This API is needed to support the ongoing efforts of the Kinect team and mobile development teams. All data collected by these teams is of little use when left on the medium that generates them as it cannot be analyzed by medical professionals to assist in care. Therefore, this data must be transmitted in a secure and reliable method to a cloud based database where it can be used to provide better healthcare for patients.

III. Related work

The cloud server will be running WAMP which supports a MySQL database. Connectors for MySQL already exist for C++ and Java so we will use these to connect to the database. Android devices have supported JDBC for some time, but it will be a requirement that as part of the Kinect game, a ODBC driver is installed on the computer running it if there is not one already. MySQL supports connections using HTTPS which uses encryption which will ensure that our data is transmitted successfully.

IV. Proposed approach

We will need to develop several functions for other groups to use. The first thing we will need is a function that takes a user's login parameters and ensures that they are able to connect to the database. Then, we will need to implement an insert function for each of the tables in the database. Our functions will ensure that the data it receives meets the type requirements of the database but will allow the database evaluate the validity of the values it is passed to save on local computing power. All data will be sent as fast as it is received unless a connection cannot be made in which case it will be stored in a buffer until it receives another piece of data at which time it will attempt to send the data again. If this buffer becomes too full it will alert the caller so that they can attempt to restore connectivity. This buffer will be configurable in size to allow the user to determine what is appropriate for their type of system. The API will prioritize sending the data as soon as possible due to the fact that it is medical data and a loss of data due to device failure could potentially be dangerous.

V. Evaluation plan

Our API should ultimately perform flawlessly as long as an internet connection is available. Data transmission is a critical part of it's reliant projects and so we cannot afford to let this become a point of failure. If an internet connection is maintained, we should be able to achieve 100% transmission success though we cannot be responsible for the successful transmission of malformed data. A simple set of test transmissions will easily be confirmed by querying the database manually.

VI. Timeline

TABLE I. Development timeline

Date	**Development Timeline**	
	Goal(s)	***Responsibility***
Week 1	***Project Management*** Determine methods of communications. Setup GitHub Repository	***All***
Week 2	***API Design*** Determine the different function calls for the API and their requirements	***All***
Week 3	***User Authentication*** Create methods to authenticate against the cloud database	***Morris***
Week 4	***Buffer design*** Develop the buffer to handle the queuing of actions and implement	***Saari***
Week 5	***Insert function*** Implement method to add data to database through buffer	***Saari***
Week 6	***Read function*** Implement method to read data from the database	***Morris***
Week 7	***Validity testing*** Evaluate the API for data reliability requirements	***All***
Week 8	***User testing*** Have the team evaluate the API for usage	***All***
Week 9	***Feedback*** Based on feed back from user testing adjust API and add functionality	***All***
Week 10	***Completion*** Polish API. Address any additional requirements for release	***All***

References

[1] *C++ MySQL library http://tangentsoft.net/mysql++/*

[2] *Java MySQL library https://dev.mysql.com/usingmysql/java/*