FEASIBILITY STUDY

The Group:

Derek Campaniello, derek_campaniello@student.uml.edu
Johnathan Wydola, Jonathan_Wydola@student.uml.edu
Madhumathi Prakash, Madhumathi_Prakash@student.uml.edu
Alexander DiPaola, Alexander_DiPaola@student.uml.edu

The Client:

The project will encompass different customers including the general, aging, physical rehabilitation populations, and/or physicians.

The Task to be Undertaken:

The project is to develop a web system that is able to store, collect and index data from a data aggregator which will be receiving different inputs from one or more sensors.

This web system consists of two parts. The first is a database to accomplish the storage, collection and indexing of data. The second will be a backend web user interface where the user can login and retrieve, filter and sort the stored data.

Benefits:

The benefits of this project would be to assist the aging population, post stroke patients, and/or weight loss patients in retrieving the data and providing a web user interface in order to analyze and display the data. It will also provide health care providers/physicians access to patient records. While accomplishing the tasks above, the user will be able to access data securely from the web interface.

A Preliminary Requirements Analysis:

The web-system requires the following functional requirements:

1. Web Interface
    a. Client Backend
        i. Allow the user to login to their personal account.
        ii. Securely retrieve information from the database in order to display saved data to the end user in which they can analyze through SSL. This data will be filtered and index appropriately.
        iii. Authenticate with certain devices and certain sensors.
        iv. Allow user to update personal information such as age, height, weight, etc…
2. Database to store data from the data aggregator
    a.) This database will store and index information from the data aggregator.
    b.) The database will also retrieve the necessary to pass to the web interface.
    c.) If necessary the server will preprocess certain data into the database.

The web-system may have the following functional requirements:

Undecided

1. The system may require particular filters and/or searches that are currently not known at this time.
2. The system may require certain resources to preprocess information to integrate into the database schema.
3. The system may require certain data to be displayed in a particular way.

The extent of non-functional requirements are not fully known at this time. Certain non-functional requirements may include: requests handled per second, storage size, memory, CPU power, uptime, reliability and response times.

Technical Requirements – Feasibility:

1. *Server* – The web-system will be running on a HTTP server. Most likely this server will be running on the Amazon Web Service (AWS) platform and the system will most likely be Linux based.
2. *Database* – The system will be on a separate server from the HTTP server (if available, also on the AWS platform) and will be a MySQL server.
3. *Web UI* – The web UI will be programmed using HTML, CSS, and PHP. It will retrieve information from the database and display it to the end user.
4. *Security* – The server will use SSL to encrypt traffic over HTTPS. The traffic will also be encrypted between the database and HTTP server. Devices and Sensors will also be authenticated through the database.

Scope:

For this project, we will be focusing on displaying information received from the data aggregator with one or more sensors. The information will be displayed within a UI to the user. The user will also be able to view personal information through a login and profiling system. The UI and database will run on a single server or multiple servers running on the AWS platform.

Suggested Deliverables:

Management Deliverables:

1. *Requirements Analysis* – This document will overview the functional and non-functional requirements. This document ensures that the development team is following the requirements set by the client. This also allows the client to add, change or remove any requirements set forward to ensure a successful final product.
2. *Design Document* – This document will overview the design of the project. This will allow the development team to outline how particular features are to be implemented.
3. *Source Code* – A document along with the final version of the code. This version of the code will encompass the requirements and designs as stated in the previous documents. This code will also test the system using various methods of unit testing before being deployed into production.

Technical Deliverables:

1. A **database** that contains the proper database schema. This database will most likely be implemented on the Amazon Web Services (AWS) platform.
2. A **login system** that allows the user to register, login and manage their personal account information including name, age, height, weight, sex, etc…
3. A **user interface** that allows the user to view their personal information. This could be applied to several different populations including the elderly (fall detection and other human motions), physical rehabilitation patients (human motions), weight loss patients (daily diet), physicians (access patient records with patient authentication) and/or general population (human motions). The user will also be able to filter/search for specific data.

Walk-Through:

This walkthrough will act as a common ground between the client and the development team to ensure that the requirements and design are up to the standards of the client. This acts as a rough outline of the functionality for the system and may experience some changes before the final product is released to the client.

User/Patient walk-through:

First, the user will register an account that includes their personal information. After this account is created, they can then register devices and sensors to their account and select the specific application for the device and sensor (i.e. the type of client that is using the system.). Once data is retrieved from the data aggregator, it will be stored within a database. This database will then be accessed by the web user interface to display the information to the user. The user then can filter or search for specific data points including dates, times, etc…

Administrator/Physician walk-through:

First, the user will allow the physician/administrator access to their records through authentication. The administrator/physician will then be able to view, modify and analyze patient/user records. Finally, they would be able to provide feedback to the user based on the results.

Software Development Process:

This project will use an **incremental development model** along with **some reuse** to accomplish this web system. This will effectively allow the development team to split up functionalities and release new versions with these functionalities. The benefits of using this this model include:

1. **Separation of tasks –** This will allow different members of the development team to work on different functionalities with minimal conflicts.
2. **Reassessment of design/requirement –** With the traditional incremental development model, the design and requirements are interleaved. This allows for more flexibility in accomplishing the requirements set forward by the client.
3. **Integration of existing functionalities –** With the reuse model, the development team will be able to spend more time developing new and higher priority features as laid out by the client.

Outline Plans:

1. Milestone 1 (3/27/2015) – Requirements Analysis Document
2. Milestone 2 (4/10/2015) – Design Document.
3. Milestone 3 (4/17/2015) – Database. A database will be created with a tentative schema and sample data.
4. Milestone 4 (4/24/2015) – User Login/Admin Login. – The user will be able register, login, and authenticate. Unit testing will be done along with design following the incremental design model.

5. Milestone 5 (4/30/2015) – Query database and display information with proper authentication. Along with final write-up report and presentation.

Visibility Plan:

External – The development team will conduct biweekly meetings with the client. If any other issues or requirements arise, the development team will communicate with the client and schedule additional meetings if necessary.

Internal – The development team will conduct virtual/in-person meetings weekly. The team will also meet with the other subgroups to discuss the integration process periodically. Any other communications will be conducted through email. The source code will be hosted on Github. Finally, during meetings or communications progress will be monitored and reported in order to verify milestones are accomplished in a timely manner.

Business Considerations:

The development team owns the product under copyright laws, but as this project is being released as open-source, any client can use/modify the product under the terms of the GPL V3 license.

Risk Analysis:

1. **Changing Requirements**
   Risk: The development team may have different ideas about the system during the course of the project. Depending on the situation presented, the application may require little or many changes to the architecture.
   Solutions: To prevent major changes to the architecture, the development team will need to meet with the client to develop a clear visibility plan.
2. **Incomplete Requirements**
   Risk: It is possible that requirements may be misunderstood or ignored.

Solutions: The development team will present the completed requirements as well as the requirements to be completed in order to assure that no requirements are missing. If a requirement is not presented at the time of the meeting, then it is the client's responsibility to address and inquire the status of the inquired requirement.

3. **System Integration**

   Risk: Depending on the access to the servers the development team receives, the development team may be working for a majority of the project in a local environment for testing and integration reasons.

   Solutions: The development team will need to discuss the requirements of the system configuration as early in the development process as possible.

4. **Human resources**

   Risk: The development team consists of three active members and a fourth that has yet to respond to our development team messages. With the deadline approaching, it is possible that time may impose some restraints on the requirements provided by the client

   Solutions: Notify the client that human resources is an implication to the success of the project; therefore the client should be reasonable and fair when issuing requirements for the final product

5. **Technical Requirements**

   Risk: The current server platform and hardware is not known at the time. Therefore, there are risks involved that may affect the final product architecture.

   Solution: Further communication between the development team and the client as soon as possible to ensure a stable system.

6. **Non-functional Requirements**

   Risk: Certain non-functional requirements may be added or changed during development such as response times, stability, security, processing power, database lookup times, etc…

   Solution: Meetings will be setup to discuss additional non-functional requirements.

Conclusion:

Based on this document, the development team has agreed on the following terms:

1. This project is feasible and the development team is ready to accomplish the task at hand.
2. It is believed by the development team that the only cost involved would be personal labor.
3. At this point the deadline is the week before finals begin therefore, somewhere around May 1st 2015.