Software Design Document

for

Diabetic Patient Monitoring

Version 2.0 approved

Prepared by:
  Christopher Fong
  Koenrad Macbride
  Yosep Kim
  Andrew Padilla
  Wun Woo

Medtronic / Alex Zhong

December 8, 2017
# Table of Contents

1. Introduction 3  
   1.1 Purpose 3  
   1.2 Intended Audience and Reading Suggestions 3  
   1.3 System Overview 3  

2. Design Considerations 4  
   2.1 Assumptions and Dependencies 4  
   2.2 Goals and Guidelines 4  
   2.3 Development Methods 4  

3. Architectural Strategies 5  

4. System Architecture 6  
   4.1 - Level 0 DFD 6  
   4.2 - Level 1 DFD 6  
   4.3 - Detailed descriptions 8  

5. Policies and Tactics 10  
   5.1 Choice of which specific products used 10  

Appendix A: Glossary 11
1. Introduction

1.1 Purpose
The purpose of this document is three-fold:
   a) Completely define a full set of requirements for the this project
   b) Completely define the design for this project.
   c) Define and partially implement feasible modules for this project

1.2 Intended Audience and Reading Suggestions
This document is intended for future Medtronic developers and CSULA senior design students.

1.3 System Overview
The purpose of this project is to obtain insight on glucose level based on activity level, and find any correlation between them. Also, come up with a solution to predict glucose levels using FitBit metrics. Predict continuous glucose levels based on activity levels and glucose samples. Dynamically Process data and display information to patients or authorized people.
2. Design Considerations

2.1 Assumptions and Dependencies

<table>
<thead>
<tr>
<th>#</th>
<th>Dependency</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web Browser w/JavaScript</td>
<td>End User</td>
</tr>
<tr>
<td>2</td>
<td>Web-App Front End</td>
<td>CSULA Team</td>
</tr>
<tr>
<td>3</td>
<td>Real Time Data Feed</td>
<td>Medtronic</td>
</tr>
</tbody>
</table>

2.2 Goals and Guidelines

- Finding insight/relationship between physical activity and glucose levels and visualizing results
- Investigating the possibility of predicting continuous glucose level based on physical activity and blood glucose samples

2.3 Development Methods

This project is being conducted using Agile development. The development process is formal with document and code reviews.

- Time series Research
- API Gateway and API endpoint creation

**Data Manipulation/Machine Learning Process**

- Back-end Python Machine Learning Structure and Code
- Data preprocessing and data cleaning
  - Store raw glucose and fitness data
  - store trained data set
- Sample data feature extraction
  - Prepare data for machine learning algorithm
- Feature selection and dimensionality reduction
- Knowledge extraction/ data analytics, deriving insights
- Data testing and training
- Continuous data processing
3. Architectural Strategies

The real time data stream will be handled by Apache Kafka, which will send the data stream to Apache Storm for data analysis using python data analysis libraries. Storm will send the analysis to a front end for viewing by the end-user. Each module is a Microservice and can be run independently from each other Microservice. These microservices can be easily scaled.
4. System Architecture

4.1 - Level 0 DFD

4.2 - Level 1 DFD
4.2.1 Feature Extraction Engine

1.3 Feature Extraction Engine

Data Preprocessing and Cleaning → Feature Extraction → Dimensionality Reduction → Output

DB

4.2.2 - Prediction Engine

1.5 Prediction Engine

Training Data → ML Training → Predictive Model → ML Testing/Prediction → Output

Real Time Data

4.2.3 - Insight Engine

1.4 Insight Engine

BG Features → Correlation and Statistical Analysis → Insights
Physical Activity Features → Visual Representation and Pattern Analysis
Food Intake Features
4.3 - Detailed descriptions
Major functional subunits shown in the DFDs in the previous subsections, are described in detail below.

1.1 Static FitBit Data
FitBit analytics are received from a static directory in production.

1.2 Real Time Data Feed
This module handles the real time data. This includes continuous glucose readings, food intake, and manual glucose readings.

1.2.1 - Database for storing Raw Data
1.2.2 - Real Time Data Feed
1.2.3 - Cassandra Database shall be used to store incoming data
1.2.4 - Apache Kafka shall receive data stream from fitbit
1.2.5 - Apache Kafka shall receive data stream from SG
1.2.6 - Apache Kafka shall create a producer for streaming to Apache Storm
1.2.7 - Apache Storm shall map data in parallel to the stream
1.2.8 - Apache Storm shall send processed data to Feature Extraction Engine (1.3)

1.3 Feature Extraction Engine
The feature extraction engine will extract features, use feature selection and dimensionality reduction. These extracted features will be sent to modules 1.4 and 1.5.

1.3.1 - Data shall be received from Apache Storm.
1.3.2 - Pandas shall handle features of real time data
1.3.3 - Database for storing extracted features

1.4 Insight Engine
The insight engine will make insights.

1.4.1 - Python script shall find features to develop insights from the real time data
1.4.2 - Pandas shall handle good values for correlations of different features.
1.4.3 - Database for storing predictions and insights.

1.5 Prediction Engine
The prediction engine will take training data, and use it to produce a predictive model. It will then use the predictive model on real time data to produce results.

1.5.1 - Data shall be received from Apache Storm.
1.5.2 - RNN shall be implemented using TensorFlow to create predictions
1.5.3 - Database for storing predictions and insights.
1.6 Front End

The front end will display all of our results after our insight and prediction data is put together.

1.6.1 - Html shall show webpage
1.6.2 - Javascript shall show charts
1.6.3 - Data is coming from the Prediction Engine
5. Policies and Tactics

The team shall use SCRUM developing methods

5.1 Choice of which specific products used

<table>
<thead>
<tr>
<th>Technologies Used</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Apache Kafka</td>
</tr>
<tr>
<td>Apache Storm</td>
</tr>
<tr>
<td>Apache Cassandra</td>
</tr>
<tr>
<td>Python</td>
</tr>
<tr>
<td>Flask</td>
</tr>
<tr>
<td>Amazon Web Services</td>
</tr>
<tr>
<td>TensorFlow</td>
</tr>
<tr>
<td>JavaScript</td>
</tr>
<tr>
<td>SciKit-Learn</td>
</tr>
<tr>
<td>Pandas</td>
</tr>
<tr>
<td>HTML5</td>
</tr>
</tbody>
</table>
## Appendix A: Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDD</td>
<td>Software Design Document</td>
</tr>
<tr>
<td>BG</td>
<td>Blood Glucose Level</td>
</tr>
<tr>
<td>SG</td>
<td>Sensor Glucose Data from Real Time Glucose Monitor</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
</tbody>
</table>