Great Catch

**H**azards **E**liminating **L**eague **P**rofessionals

(HELP)

System Design Specification

[Cycle 3]

Date:

[4/9/2017]

**Team Members (Initials):**

Name: Tiane Bai (TB)

Name: Daniel Grayson (DG)

Name: Alex Manning (AM)

Name: Yupeng Sun (YS)

Name: Wenyu Xin (WX)

**Table of Contents**

[1. Introduction 4](#_Toc479533758)

[1.1 Changes made comparing to SDS Cycle 2 4](#_Toc479533759)

[1.2 Purpose 4](#_Toc479533760)

[1.3 Scope 4](#_Toc479533761)

[1.4 Definitions, Acronyms, and Abbreviations 4](#_Toc479533762)

[1.5 References 4](#_Toc479533763)

[1.6 Overview 4](#_Toc479533764)

[1.7 Requirements Traceability Matrix 5](#_Toc479533765)

[2. Architectural Description 6](#_Toc479533766)

[2.1. Top down layering of functions and processes 6](#_Toc479533767)

[2.2. Structure and relationships 8](#_Toc479533768)

[2.2.1. Level 0 - Item 1 - Wearable Device and Great Catch Communication 8](#_Toc479533769)

[3. Interface Description 9](#_Toc479533770)

[3.1. User Interface 9](#_Toc479533771)

[3.1.1. User Sign In Page 10](#_Toc479533772)

[3.1.2. Create Account Page 10](#_Toc479533773)

[3.1.3. Profile Page 10](#_Toc479533774)

[3.1.4. Report Page 11](#_Toc479533775)

[3.1.5. Alert History Page 12](#_Toc479533776)

[3.1.6. Help Page 13](#_Toc479533777)

[3.2. Data Interface 13](#_Toc479533778)

[3.2.1. Entities 13](#_Toc479533779)

[3.2.2. User Information 14](#_Toc479533780)

[3.2.3. Caretaker/Health Professional Information 14](#_Toc479533781)

[3.2.4. Wearable Activity Data 14](#_Toc479533782)

[3.3. Programming Interface 15](#_Toc479533783)

[4. Detailed Design 15](#_Toc479533784)

[4.1. Component Template Overview 15](#_Toc479533785)

[4.2. Function 15](#_Toc479533786)

[5. Reuse and relationships to other products 18](#_Toc479533787)

[6. Design decisions and tradeoffs 19](#_Toc479533788)

[6.1. Web or Mobile platform 19](#_Toc479533789)

[6.2. Machine Learning or Simple Statistical Analysis 19](#_Toc479533790)

[6.3. Frequency of notification 19](#_Toc479533791)

[7. Pseudo code for components 19](#_Toc479533792)

[7.1. Retrieve wearable activity data from Fitbit website 19](#_Toc479533793)

[8. Appendices 20](#_Toc479533794)

# Introduction

## Changes made comparing to SDS Cycle 2

Table 1. Changes made comparing to SDS Cycle 2

|  |  |
| --- | --- |
| **Which section** | **What is changed** |
| All Section | 1. Proofread by Wenyu to ensure constant grammar and word phrasing |
| Architectural Description | 1. Re-organized functions and processes to show level layering and relationships |

## Purpose

The purpose of this document is to provide system design specifications for Great Catch. This document will contain: an architectural description; interface description; detailed design; reuse and relationships to other product; design decisions and tradeoff; pseudo code for components; and any applicable appendix items. It is intended for the stakeholders, technical advisors, and developers of the Great Catch.

## Scope

The Great Catch will monitor elderly people’s health by periodically and automatically pulling data from wearable devices’ remote server via a provided API for the device data. The data will be styled to a predefined format before sending to a computer cluster to conduct statistical analysis and machine learning. The analysis will determine whether or not the data is significantly different from older data patterns. If abnormal patterns are detected, email and/or text alerts will be sent out to the senior user and their caretaker(s), notifying them that an abnormal pattern has been detected. The Great Catch will also provide reporting options for both users and caretakers to view alert histories, as well as any historical abnormal patterns. The Great Catch will run on a web platform with graphical user interface.

## Definitions, Acronyms, and Abbreviations

* + 1. N/A - Not Applicable
    2. User Information - FirstName, MiddleName, LastName, Age, Email, Phone, Mobile, UserName, Password, CreatedAt, UpdatedAt, DeviceModel, FitbitID

## References

* + 1. Software Engineering Standards Committee of the IEEE Computer Society, “IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions”, September 23, 1998.
    2. CI 492 Senior Design Group 2, “SRS-C3-G2”, Jan. 30th, 2016.

## Overview

There are five major sections in this SDS report. Section 2 is on the Great Catch’s architectural description. This section describes the way Great Catch has been structured, and the purpose and functionality of each entity. Section 3 details the Great Catch’s interface design. This section describes the detailed design of the user interface. Section 4 is on the Great Catch’s detailed design which describes a collection of design entities and their attributes. Section 5 is on the Great Catch’s reuse and relationship to other products, which is not included in this report. Section 6 is on Great Catch’s design decisions and tradeoffs. This section covers the reason for some of the design choices made, including their tradeoffs. Section 7 contains pseudocode for the components, which are code templates to be used for the Great Catch. Section 8 is appendices which is not applicable in current version of SDS.

## Requirements Traceability Matrix

This section maps the relationship between requirement statements and detailed design entities. As such, it shows how the requirements are covered by the design, and demonstrates why these design entities exist.

The values in the cells of the table show which requirements provide the purpose for each entity. The cell values are:

* **Blank** – the design entity does not implement any of that requirement
* **P** for Primary - the design entity implements all or most of the requirement
* **S** for Secondary – the design entity implements a smaller but essential part of the requirement

Table 2. Requirements Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirement** | **1.1.1** | **1.1.2** | **1.2.1** | **1.2.2** | **2.1.1** | **2.2.1** | **2.2.2** | **3.1.1** | **3.2.1** | **3.2.2** | **3.3.1** | **3.3.2** | **3.4.1** | **5.1.1** | **5.1.2** | **5.2.1** | **5.3.1** | **5.3.2** | **5.3.3** | **6** |
| **User Creation** |  |  |  |  |  |  |  |  |  |  | p | p |  |  |  |  |  |  |  |  |
| **User Deletion** |  |  |  |  |  |  |  |  |  |  | p | p |  |  |  |  |  |  |  |  |
| **Caretaker Contact Information Validation** |  |  |  |  |  |  |  |  |  |  | s | s |  |  |  |  |  |  |  |  |
| **User and Caretaker Information Update** |  |  |  |  |  |  |  |  |  |  | s | s |  |  |  |  |  |  |  |  |
| **Pulling Data from Fitbit Server** | s | s | p | p |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Data Validation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | p |  |  |  |  |  |
| **Data Formatting** |  |  |  |  |  |  |  |  |  |  |  |  |  | p | p | p |  |  |  |  |
| **Data Processing** |  |  |  |  |  |  |  |  |  |  |  |  |  | p | p | p |  |  |  |  |
| **Determine Abnormal Parameters** |  |  |  |  |  |  |  |  |  |  |  |  |  | s | s | s | p | p | p |  |
| **Low-Severity Alert** |  |  |  |  |  | s | s |  | s | s |  |  |  |  |  |  |  |  |  | p |
| **Mid-Severity Alert** |  |  |  |  |  | s | s |  | s | s |  |  |  |  |  |  |  |  |  | p |
| **High-Severity Alert (Email)** |  |  |  |  |  | s | s |  | s | s |  |  |  |  |  |  |  |  |  | p |
| **High-Severity Alert (Call)** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | p |
| **Report Generating (Summarized)** |  |  |  |  | p |  |  | p |  |  |  |  |  |  |  |  |  |  |  |  |
| **Report Generating (Raw)** |  |  |  |  | p |  |  | p |  |  |  |  |  |  |  |  |  |  |  |  |

# Architectural Description

## Top down layering of functions and processes

Table 3. Top down layering of functions and processes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Functions/Processes** | | | | | |
| **Level 0** | 1. Wearable Device and Great Catch Communication | 2. Caretakers and Great Catch Communication | 3. Wearable Device Users and Great Catch Communication | 4. Health Professional and Great Catch Communication | 5. Great Catch Data Process | 6. Great Catch Alert |
| **Level 1** | 1.1 Fitbit Authentication  1.2 Data Request & Receive | 2.1 Data Report  2.2 Alerts | 3.1 Data Report  3.2 Alerts  3.3 Account Setup  3.4 Account Access | 4.1 Data Report  4.2 Alerts | 5.1 Gather Data  5.2 Format Data  5.3 Process Data  5.4 Categorize Result | 6.1 Compose Alert  6.2 Send Alert  6.3 Log Alert History |
| **Level 2** | 1.1.1 Redirect to Fitbit website  1.1.2 Redirect back to Great Catch  1.2.1 Data Request  1.2.2 Data Receive | 2.1.1 Request Data Report  2.2.1 Receive Mid/High-Severity Alerts  2.2.2 Respond to Mid/High-Severity Alerts | 3.1.1 Request Data Report  3.2.1 Receive Low/Mid/High-Severity Alerts  3.2.2 Respond to Low/Mid/High-Severity Alerts  3.3.1 Validate Input Information  3.3.2 Commit Database Insert  3.4.1 Validate User Credential | N/A | 5.1.1 API Request to Fitbit Server  5.1.2 Validate Data Integrity  5.2.1 Parse JSON Data  5.3.1 Apply Linear Regression  5.3.2 Apply Mean Average Deviation  5.3.3 Apply Correlation | 6.3.1 Commit Database Insert |

I see that you are decomposing the processes from Level 0 to Level 2. Why didn’t you present it DFDs? The data flows are missing if you only present it as a table.

## Structure and relationships

## Level 0 - Item 1 - Wearable Device and Great Catch Communication

Wearable device will need to provide raw data of users to Great Catch, and this is done using Fitbit’s authentication process. This process will require users to be redirected to Fitbit’s main website where the users will provide their Fitbit account credential to login (Table 3 - 1.1.1) . Once the users are in the Fitbit system, they will be prompted to authorize Great Catch to access the Fitbit data. If users deny, there will be no communication between wearable device and Great Catch. If users agree, then users will be redirected back to Great Catch website (Table 3 - 1.1.2) where the users can proceed to other functionality of Great Catch. Moreover, when Great Catch needs to acquire raw data from Fitbit, a HTTP request will be sent to Fitbit and JSON data will be received at Great Catch (Table 3 - 1.2). Once the data is received from Fitbit by Great Catch, data reporting (Table 3 - 2.1, 3,1, 4.1) and data processing (Table 3 - 5) can be started at will.

* + 1. **Level 0 - Item 2 - Caretakers and Great Catch Communication**

Caretakers will have 2 main processes, which is data report and alert (Table 3 - 2.1, 2.2). Data report requires a GET request to Great Catch which will translate selected data of interest criteria into query language and reply back desired data. Alerts on the other hand, has 2 sub-processes (Table 3 - 2.2.1, 2.2.2), which is used to send, receive, and reply to alerts. These sub-processes are related to Great Catch Alert (Table 3 - 6).

* + 1. **Level 0 - Item 3 - Wearable Device Users and Great Catch Communication**

Wearable Device Users have 4 main processes, data report, alert, account setup, and account access (Table 3 - 3.1, 3.2, 3.3, 3.4). Data report is like the one in Caretakers and Great Catch Communication Data Report (Table 3 - 2.1) where user’s data selection criterias will be translated to query language, and data of interests will be displayed. Alert is like the one in Caretakers and Great Catch Communication Alert (Table 3 - 2.2) where alerts will be send, receive, and reply. Account Setup is where user input necessary information into Great Catch and Great Catch will validate the given information and create the account for the user (Table 3 - 3.3.1, 3.3.2). Account Access is where user will use provided unique credential to login/logout of Great Catch. The credential provided will be validated in Great Catch (Table 3 - 3.4.1). All processes in Wearable Device and Great Catch Communication (Table 3 - 1) are depended on current main process to start.

* + 1. **Level 0 - Item 4 - Health Professionals and Great Catch Communication**

Health Professionals will have 2 main processes which are the same as those of Caretakers and Great Catch Communication (Table 3 - 2).

* + 1. **Level 0 - Item 5 - Great Catch Data Process**

Great Catch Data Process will have 4 main processes, which are Gather Data, Format Data, Process Data, and Categorize Result (Table 3 - 5.1, 5.2, 5.3, 5.4). All processes here depended on an existing user account in Great Catch (Table 3 - 3.3, 3.4) and Fitbit account authentication (Table 3 - 1.1). Gather Data will send API request to Fitbit server (Table 3 - 5.1.1) and once data will data integrity will be validated to insure security is not breached or data is complete without any technical difficulties (Table 3 - 5.2.1). Then Process Data will begin to spot any potential abnormal pattern. First linear regression is applied to determine slope changes (Table 3 - 5.3.1). If the slope is flat or negative in value, then mean average deviation is applied to see the difference between new data and existing data to make further judgement (Table 3 - 5.3.2). Lastly, if mean average deviation is lower than before, then correlations will be applied to check if any changes in medicine might cause such decrease (Table 3 - 5.3.3). At the end, the process will determine the severity of this result (Table 3 - 5.4.1) and trigger Great Catch Alert (Table 3 - 6) if needed.

* + 1. **Level 0 - Item 6 - Great Catch Alert**

Great Catch Alert has 3 main process which handles alert composing, sending, and logging. Composing and Sending are built-in SMTP/HTML functions and logging is saving alert sent date and information into Great Catch database (Table 3 - 6.3.1).

# Interface Description

## User Interface

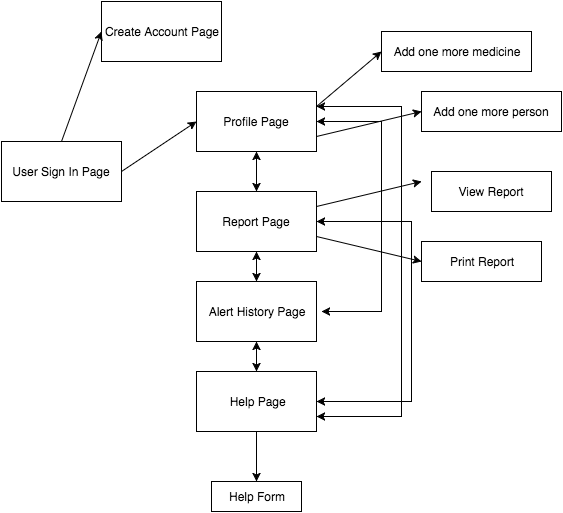


Figure 1. High-level overview of the system

Above is the high-level overview for the system. The system contains six pages: User Sign-in, Create Account, Profile, Report, Alert History, and Help. Upon initial site access, the user will see the User Sign-in page. From there, they may access any other page. On the Profile Page, users can add their medications or add emergency contacts. The Report Page allows users to view or print reports. The user can view and verify the alert history on the Alert History page. For the Help Page, a user can email the webmaster by filling out the help form.

### User Sign In Page

**Type:** Screen on system page

**Description:** This page includes secure sign-in with user information

and the “Need an account” link (Figure 2).

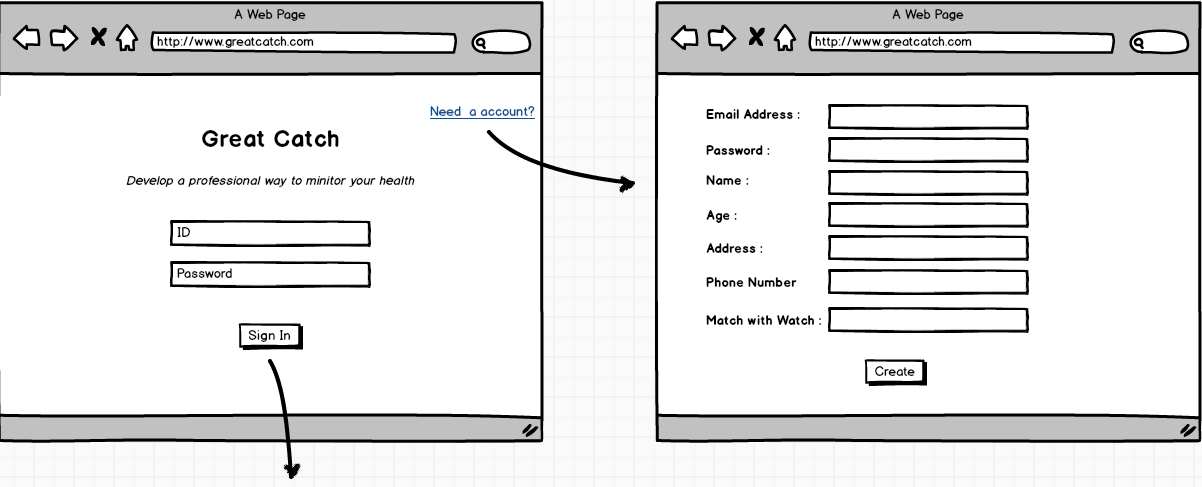


Figure 2. User sign in page

### Create Account Page

**Type:** Screen on system page

**Description:** This page allows users to create accounts with their basic information such as

email address, name, age, gender (Figure 3).

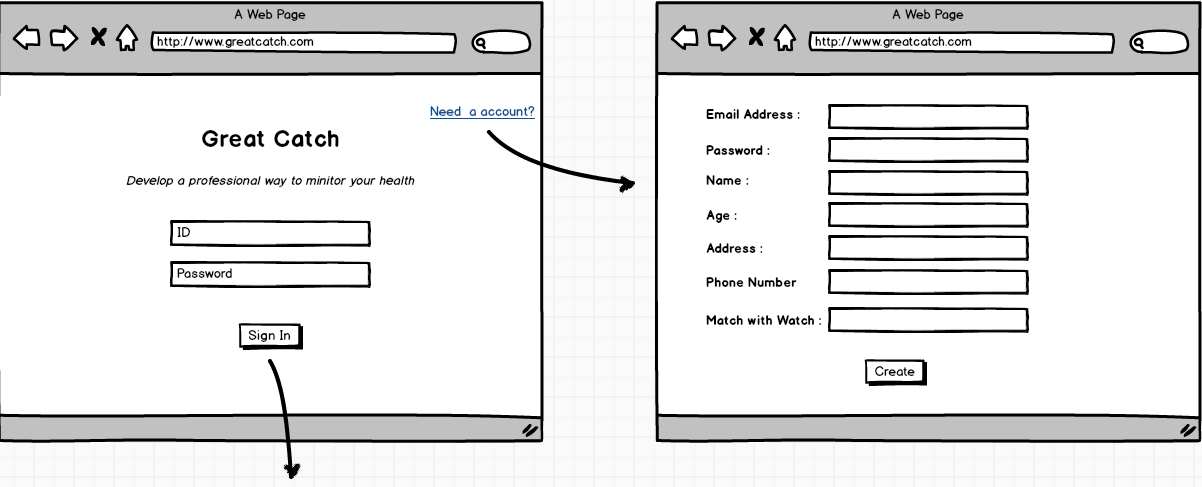


Figure 3. Create an account

### Profile Page

**Type:** Screen on system page

**Description:**  This page provides information with the users logged in. Once logged in, first time users shall be prompted to filled in more information about current medicine and emergency contacts (Figure 4).

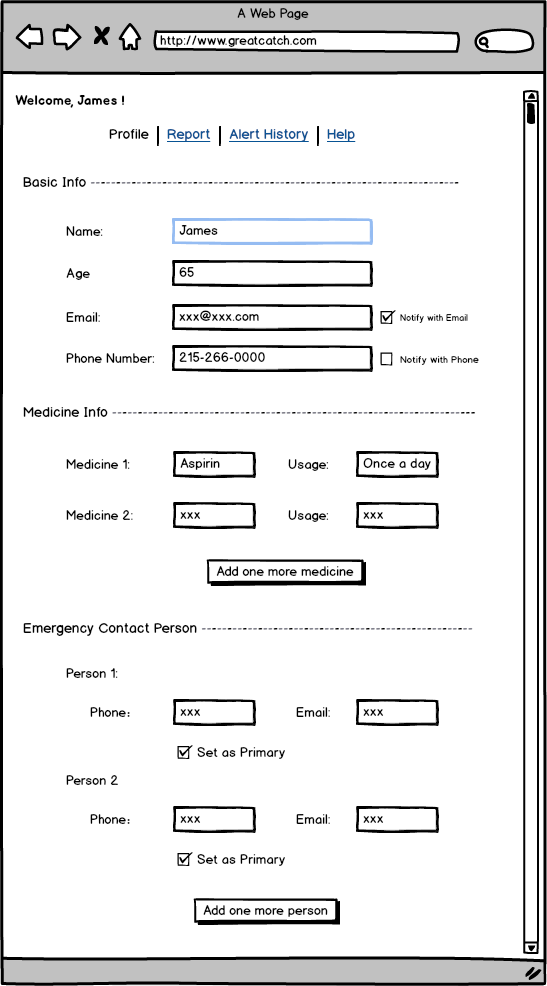


Figure 4. User information and emergency contact information

### Report Page

**Type:** Screen on system page

**Description:** This page has options for users to view and generate reports with their choice of

information and date range (Figure 5). These reports can then be shown to physician for further

analysis. Report format shown is not finalized.

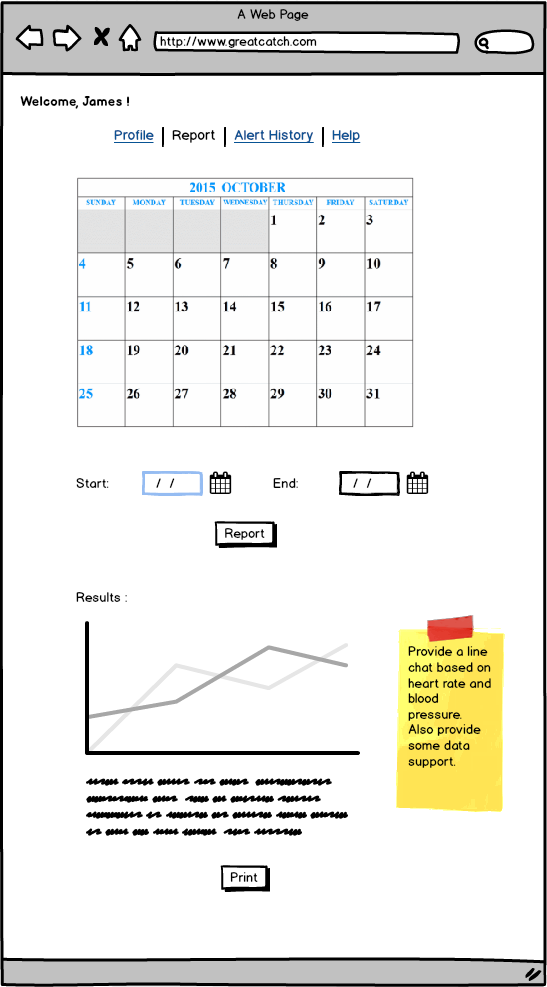


Figure 5. View/Print report

### Alert History Page

**Type:** Screen on system page

**Description:** This page provides more advanced features. Users or caretakers can verify

alert histories by viewing the alert log (Figure 6).

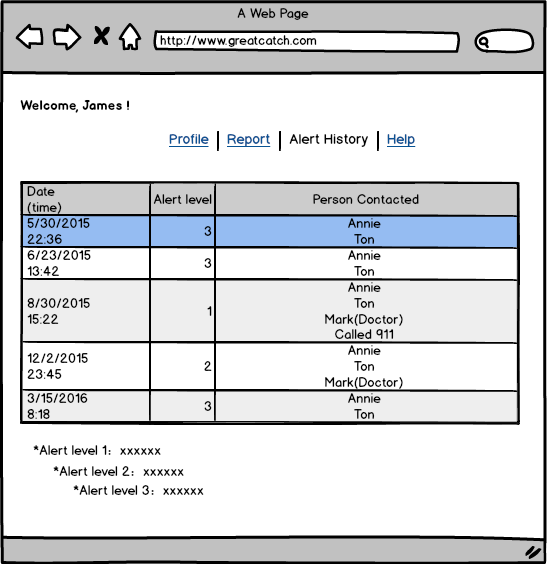


Figure 6. Alert history

### Help Page

**Type:** Screen on system page

**Description:** The help page can be reached within the system if user or caretaker face any

technical difficulty (Figure 7).



Figure 7. Help Page

## Data Interface

### Entities

* + - 1. Alerts
      2. Medicine
      3. Caretaker/Support (Health Professional
      4. Caretaker/Support (Health Professional) Request (reports/raw data/status reading)
      5. User

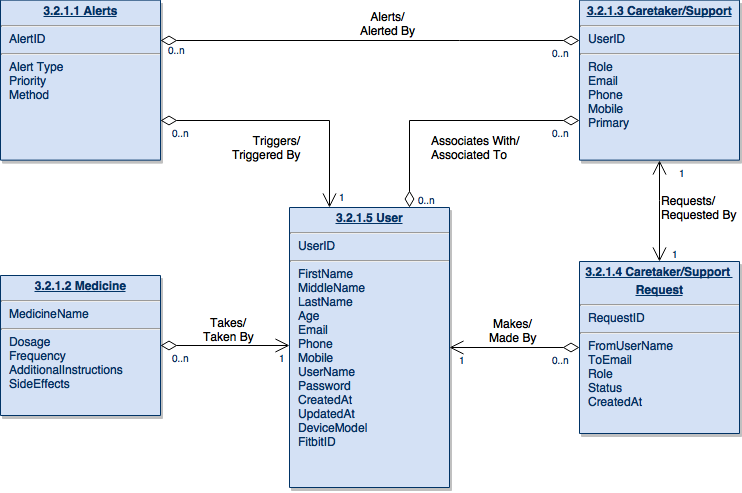


Figure 8. ERD

### User Information

**Type:** Data Object

**Description:** When a new user is created a unique user ID will be generated. The user will add contact, demographic, and medicinal information.

### Caretaker/Health Professional Information

**Type:** Data Object

**Description:** Users can share reports of their data with peers, caretakers, and doctors. Contact information along with roles and privileges to facilitate communicating activity alerts.

### Wearable Activity Data

**Type:** Data Object

**Description:** Activity data from the user’s wearable is uploaded to the wearable company’s servers. In order to display reports and detect abnormal data for alerts, data from the wearable is pulled from the companies servers. This data can include steps, heart rate, distance, floors climbed, and calories burned.

## Programming Interface

Python 2.7; downloaded from<https://www.python.org/download/releases/2.7/>

D3.js 3.5.15; downloaded from<https://d3js.org/>

Eclipse Luna; downloaded from<https://www.eclipse.org/downloads/>

Git 2.7.1; downloaded from<https://git-scm.com/>

BitBucket; accessed from<https://bitbucket.org/>

SourceTree 1.8.1; downloaded from<https://www.sourcetreeapp.com/download/>

Visual Studio 2012; premium tool provided by Drexel University

Sublime build 3103; downloaded at<https://www.sublimetext.com/>

Neo4J 3.1; download at <https://neo4j.com/>

# Detailed Design

## Component Template Overview

* + 1. Database - Not yet decided on what type of database. It will contain tables to represent various information.
    2. Computer Cluster - Drexel Proteus. It will be used as data processing unit.
    3. Reporting Application - Not yet decided on what package. It will use information in database to produce useful reports for users.
    4. Data analysis package - SimpleStatistic. It will be used to analyze data.

## Function

* + 1. **Alert**

|  |  |
| --- | --- |
| **Identification** | Alert (3.2.1.1) |
| **Type** | Process |
| **Purpose** | Let user and caretakers know about the abnormal data pattern that is detected by Great Catch system. |
| **Function** | Takes the summarized information about the abnormal data pattern, composes email and text message using stored contact information in database, and sends them to user and caretakers via proper protocols. |
| **Subordinates** | If the alert is not being sent or responded for over 12 hours of time, 911 will be alerted. |
| **Dependencies** | User and caretaker’s contact information must be stored in the database in proper format. |
| **Interfaces** | No interface needed to set up this process. The user and caretakers will directly see the alert message. |
| **Resources** | No additional resource is required. |
| **Processing** | If no contact information is stored in the system or the stored information is not accurate to compose any message, then 911 will be called.  If the alert is not replied within 12 hours of time, 911 will be called.  All alert attempts, include both failed and successful, will be logged in database. |
| **Data** | See ERD in Section 3 |

* + 1. **Medicine**

|  |  |
| --- | --- |
| **Identification** | Medicine (3.2.1.2) |
| **Type** | data object |
| **Purpose** | Useful information to help determine abnormal data |
| **Function** | Stored medicine information in the database. When new data are being processed, medicine information are being taken into consideration. |
| **Subordinates** | N/A |
| **Dependencies** | User must manually input and update information. |
| **Interfaces** | Web form interface that allows users or caretakers to input medicine information including drug name, dosage, frequency, and other additional notes. |
| **Resources** | No additional resource is required. |
| **Processing** | If the information is not provided upon setting up the application, message will be prompted to suggest input of current medicine history.  If user has no medicine or choose to not input any information, no message will be prompted. |
| **Data** | See ERD in Section 3 |

* + 1. **Caretaker/Health Professional**

|  |  |
| --- | --- |
| **Identification** | Caretaker/Health Professional (3.2.1.3) |
| **Type** | data object |
| **Purpose** | caretaker information |
| **Function** | Stored caretaker information in the database. When alert is being composed, this information is being pulled. |
| **Subordinates** | N/A |
| **Dependencies** | User must manually input and update information. |
| **Interfaces** | Web form interface that allows users or caretakers to input contact information role, email, phone, mobile, primary. |
| **Resources** | No additional resource is required. |
| **Processing** | If the information is not provided upon set up, the app will not work.  If the information provided is not in valid format, the app will prompt error message.  Once the information is entering, confirmation message will be sent. If no reply from end user, message will be resent upon confirmed. |
| **Data** | See ERD in Section 3 |

* + 1. **Caretaker/Health Professional Request**

|  |  |
| --- | --- |
| **Identification** | Caretaker/Health Professional Request (3.2.1.3) |
| **Type** | process |
| **Purpose** | User information being pulled from database and producing reporting. |
| **Function** | The database will get a request from the client in form of query. The database will pull the data and generate report that can be sent back. |
| **Subordinates** | N/A |
| **Dependencies** | N/A |
| **Interfaces** | Web form interface for user to select date range, and data type. |
| **Resources** | No additional resource is required. |
| **Processing** | User submit request and database process the query. Information are being pulled and used to generate report. |
| **Data** | See ERD in Section 3 |

* + 1. **User**

|  |  |
| --- | --- |
| **Identification** | User (3.2.1.4) |
| **Type** | data object |
| **Purpose** | user information |
| **Function** | User must manually input and update information. |
| **Subordinates** | N/A |
| **Dependencies** | User must manually input and update information. |
| **Interfaces** | Web form interface that allows users or caretakers to input contact information role, email, phone, mobile, primary, and etc. |
| **Resources** | No additional resource is required. |
| **Processing** | If the information is not provided upon set up, the app will not work.  If the information provided is not in valid format, the app will prompt error message.  Once the information is entering, confirmation message will be sent. If no reply from end user, message will be resent upon confirmed. |
| **Data** | See ERD in Section 3 |

# Reuse and relationships to other products

N/A

# Design decisions and tradeoffs

## Web or Mobile platform

The first issue was deciding implementation for the web or a mobile platform. Mobile platforms have more impact for users as many always carry and check their mobile devices. However, mobile devices have limitations in processing power and storage. The web platform is easier to implement, and can easily access remote server power to process and store information. Thus, the team decided to implement the system on web platform first, and if time permits simplified version of system will be implemented on mobile platform.

## Machine Learning or Simple Statistical Analysis

The second issue was deciding whether to use machine learning or statistical analysis to determine any abnormal data pattern trends. Machine learning can provide more insights to hidden information without explicitly programming, but such algorithm requires heavy computing power and long processing time. On the other hand, statistical analysis is rather simple but lack in depth of insights. The team decided to implement machine learning using Drexel’s Proteus cluster computer first, and if such approach hinders progress of project, simple statistical analysis will be used instead.

## Frequency of notification

The third issue was deciding the frequency of notifications. If the system sends too many alerts to users or caretakers, the users and caretakers may disregard the alerts or become frustrated. For now, the team decided to sync data once per day, at the end of day when the user goes to bed, and process the data during this time. If abnormal patterns are detected, alerts will be sent. If no significant anomalies are detected, then no alert will be sent.

# Pseudo code for components

## Retrieve wearable activity data from Fitbit website

There are two acceptable formats for retrieving activity time series data:  
  
GET https://api.fitbit.com/1/user/[user-id]/[resource-path]/date/[date]/[period].json  
GET https://api.fitbit.com/1/user/[user-id]/[resource-path]/date/[base-date]/[end-date].json

|  |  |
| --- | --- |
| user-id | The encoded ID of the user. Use “-” (dash) for current logged-in user. |
| resource-path | The resource path; see options in the "Resource Path Options" section below. |
| base-date | The range start date, in the format yyyy-MM-dd or today. |
| end-date | The end date of the range. |
| date | The end date of the period specified in the format yyyy-MM-dd or today. |
| period | The range for which data will be returned. Options are 1d, 7d, 30d, 1w, 1m, 3m, 6m, 1y. |

# Appendices

N/A

Table of Contributions

The table below identifies contributors to various sections of this document.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Section** | **Writing** | **Editing** |
| 1 | Introduction | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |
| 2 | Architectural Description | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |
| 3 | Interface Description | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |
| 4 | Detailed Design | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |
| 5 | Design Decisions and Trade Off | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |
| 6 | Pseudo Code for Components | WX, YS, TB, DG, AM | WX, YS, TB, DG, AM |

I certify that:

* This paper/project/exam is entirely my own work.
* I have not quoted the words of any other person from a printed source or a website without indicating what has been quoted and providing an appropriate citation.
* I have not submitted this paper / project to satisfy the requirements of any other course.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Grading**

The grade is given on the basis of quality, clarity, presentation, completeness, and writing of each section in the report. This is the grade of the group. Individual grades will be assigned at the end of the term when peer reviews are collected.

A-

I see that you address the comment on DFD decomposition. But it seems that you are only presenting the decomposition as a table but not in DFDs. You have some DFDs in the last SDS but the processes are not labeled and the decomposition was not done. If you present the decomposition as DFDs, the processes and the data flows between the processes will be clearly presented.