**Great Catch**

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

(HELP)

**System Requirements 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](#_Toc479512663)

[1.1. Changes compared to SRS-C2-G2 4](#_Toc479512664)

[1.2. Purpose 4](#_Toc479512665)

[1.3. Scope 4](#_Toc479512666)

[1.4. Definitions, Acronyms, and Abbreviations 4](#_Toc479512667)

[1.5. References 5](#_Toc479512668)

[1.6. Overview 5](#_Toc479512669)

[2. Overall Description 6](#_Toc479512670)

[2.1. Product Perspective 6](#_Toc479512671)

[2.1.1. System Interface 6](#_Toc479512672)

[2.1.2. User Interface 7](#_Toc479512673)

[2.1.3. Hardware Interfaces 11](#_Toc479512674)

[2.1.4. Software Interfaces 11](#_Toc479512675)

[2.1.4.1. Fitbit API Component 11](#_Toc479512676)

[2.1.4.2. Data Request Component 11](#_Toc479512677)

[2.1.4.3. Processed Data to Webserver 11](#_Toc479512678)

[2.1.4.4. User Database Connection 11](#_Toc479512679)

[2.1.5. Communication Interfaces 11](#_Toc479512680)

[2.1.6. Memory Constraints 11](#_Toc479512681)

[2.1.7. Operations 11](#_Toc479512682)

[2.1.8. Site Adaption Requirements 12](#_Toc479512683)

[2.2. Product Functions 12](#_Toc479512684)

[2.3. User Characteristics 12](#_Toc479512685)

[2.3.1. Automatic Timed Sync 12](#_Toc479512686)

[2.3.2. User Input 14](#_Toc479512687)

[2.3.3. Data Sync Issue Alert 15](#_Toc479512688)

[2.3.4. Low Activity Alert 16](#_Toc479512689)

[2.3.5. Abnormal Pattern Alert 17](#_Toc479512690)

[2.3.6. User Creation 18](#_Toc479512691)

[2.3.7. User Deletion 19](#_Toc479512692)

[2.3.8. Report Generation 20](#_Toc479512693)

[2.4. Constraints 21](#_Toc479512694)

[2.5. Assumptions and Dependencies 21](#_Toc479512695)

[2.6. Apportioning of Requirements 22](#_Toc479512696)

[3. Specific Requirements 22](#_Toc479512697)

[3.1. External Interfaces 22](#_Toc479512698)

[3.1.1. System Interfaces 22](#_Toc479512699)

[3.1.2. User Interfaces 22](#_Toc479512700)

[3.2. Functions 23](#_Toc479512701)

[3.2.1. GC1 - User Creation 23](#_Toc479512702)

[3.2.2. GC2 - User Deletion 23](#_Toc479512703)

[3.2.3. GC3 - Caretaker Contact Information Validation 23](#_Toc479512704)

[3.2.4. GC4 - User and Caretaker Information Update 23](#_Toc479512705)

[3.2.5. GC5 - Pulling Data from Fitbit Server 24](#_Toc479512706)

[3.2.6. GC6 - Data Validating 24](#_Toc479512707)

[3.2.7. GC7 - Data Formatting 24](#_Toc479512708)

[3.2.8. GC8 - Data Processing 24](#_Toc479512709)

[3.2.9. GC9 - Determine Abnormal Parameter 24](#_Toc479512710)

[3.2.10. GC10 - Low-Severity Alert 24](#_Toc479512711)

[3.2.11. GC11 - Mid-Severity Alert 25](#_Toc479512712)

[3.2.12. GC12 - High-Severity Alert (E-mail) 25](#_Toc479512713)

[3.2.13. GC13 - High-Severity Alert (Call) 25](#_Toc479512714)

[3.2.14. GC14 - Report Generating (Summarized) 25](#_Toc479512715)

[3.2.15. GC15 - Report Generating (Raw) 25](#_Toc479512716)

[3.3. Performance Requirements 25](#_Toc479512717)

[3.4. Logical Database Requirements 26](#_Toc479512718)

[3.4.1. User and Caretaker Credential 26](#_Toc479512719)

[3.4.2. Abnormal Pattern Data 26](#_Toc479512720)

[3.4.3. Log Information 26](#_Toc479512721)

[3.5. Design Constraints 26](#_Toc479512722)

[3.5.1. Standards Compliance 26](#_Toc479512723)

[3.6. Software System Attributes 26](#_Toc479512724)

[3.6.1. Reliability 26](#_Toc479512725)

[3.6.2. Availability 26](#_Toc479512726)

[3.6.3. Security 26](#_Toc479512727)

[3.6.4. Maintainability 26](#_Toc479512728)

[3.6.5. Portability 26](#_Toc479512729)

[Table of Contributions 28](#_Toc479512730)

### Introduction

#### Changes compared to SRS-C2-G2

|  |  |
| --- | --- |
| **Which Section** | **What Changes** |
| All sections | 1. Proofread by Wenyu to make grammar and term phrasing uniform |

Table 1. Changes compared to SRS-C2-G2 using previous feedbacks

#### Purpose

The purpose of this document is to present a detailed description of the Great Catch. This document will list and explain purpose, overall description, and requirement specification of the Great Catch. It is intended for the stakeholders, the technical advisors, and the developers of the Great Catch.

#### Scope

This software system will be named Great Catch. This system is designed to help monitor elderly people’s health by continuously and automatically accessing and analyzing both data from wearable devices and manually inputted user data. If there are any abnormal patterns detected within the data set, the system will immediately alert a designated contact person about the detected pattern and advise any further actions to be taken. However, it is important to note the Great Catch will not provide any diagnostic features.

Great Catch will be designed to reduce the duration of hospital stays, or even reduce preventable deaths by constantly monitoring seniors’ health conditions and automatically detecting any issues at an early stage, which are often ignored.

#### Definitions, Acronyms, and Abbreviations

* + 1. SRS - Software Requirements Specification
    2. N/A - Not Applicable
    3. HTTP - Hypertext Transfer Protocol
    4. HTTPS - Hypertext Transfer Protocol Secure
    5. RAM - Random-Access Memory
    6. GB - Gigabyte
    7. GUI - Graphical User Interface
    8. JSON - JavaScript Object Notation
    9. DFD - Data Flow Diagram
    10. DOB - Date of Birth
    11. UNIX - Computer Operating System
    12. IPC - Inter Process Communication
    13. PostgreSQL - Object Relational Database
    14. MySQL - Database Management System

#### References

[1] Software Engineering Standards Committee of the IEEE Computer Society, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, June 25, 1998.

[2] "Web API documentation — Fitbit web API Docs,". [Online]. Available: https://dev.fitbit.com/docs/. Accessed: Oct. 31, 2016.

#### Overview

There are two additional sections in this SRS report. Section 2 pertains to Great Catch’s overall description. It describes the general factors that affect the system and its requirements. This section also provides a background for those requirements, but the specific requirements will be described in the following section. Section 3 will contain detailed and technical software requirements which will be sufficient to design and test the system.

## 

### Overall Description

#### Product Perspective

##### System Interface

Great Catch has 4 external factors: Wearable Device User, Caretakers, Health Professionals, and Wearable Device (Figure 1). The Wearable Device Users are the ones who use the wearable device such as Fitbit. The users can provide additional information, such as current medicine intakes and any existing medical conditions into the Great Catch. They can also view data reports, data statuses, and all level of severity alerts from the Great Catch. Caretakers are the family members or friends who want to monitor and be notified of the main user’s activities, and they can also view the same data reports and data statuses of the user. However, the Caretakers will only receive mid- and high-level of severity alerts. Low severity alerts are not sent to Caretakers. Health Professionals are trained professionals in a hospital setting. They will have the options to view periods of raw data or summarized data reports, and they will receive high severity alerts from the Great Catch. Lastly, the Wearable Device is the data gathering mechanism, such as a Fitbit. The Wearable Device will send its raw data to the Great Catch for analyzing.

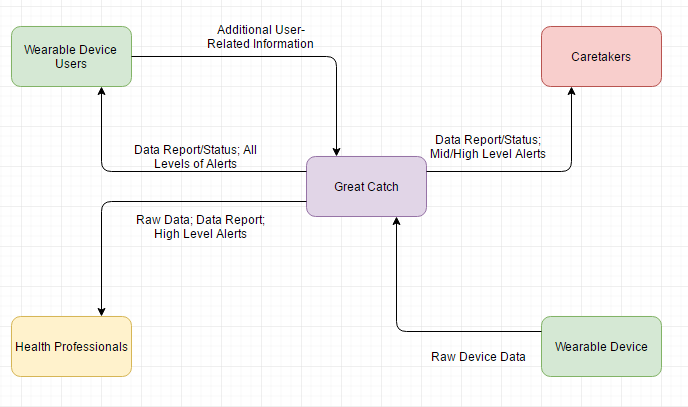


Figure 1. Context Diagram

##### User Interface

The user interface will be a responsive website. Users will be able to access the website from any type of device and observe a functional interface.

The website should be functional in Firefox, Chrome, and Edge on personal computers.

Users will be allowed to create accounts with their basic information such as email address, name, age, gender, etc. (Figure 3) and securely sign in with their chosen credentials (Figure 2). Once logged in, first-time users shall be prompted to fill in more information about current medicines and emergency contacts (Figure 4). There will also be options for users to view and generate reports containing their choice of information and date range (Figure 5). These reports can then be shown to physician for further analysis. The report format is not shown. For more advance features, users or caretakers can verify alert histories by viewing the alert log (Figure 6). Lastly, if users or caretaker face any technical difficulty, help can be reached within the system (Figure 7).

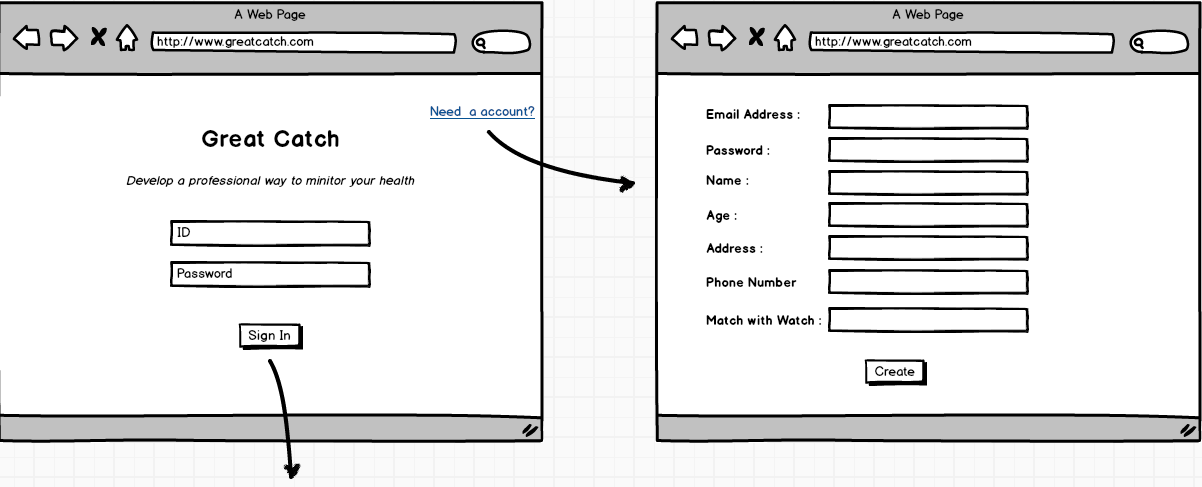


Figure 2. User sign in page Figure 3. Create an account

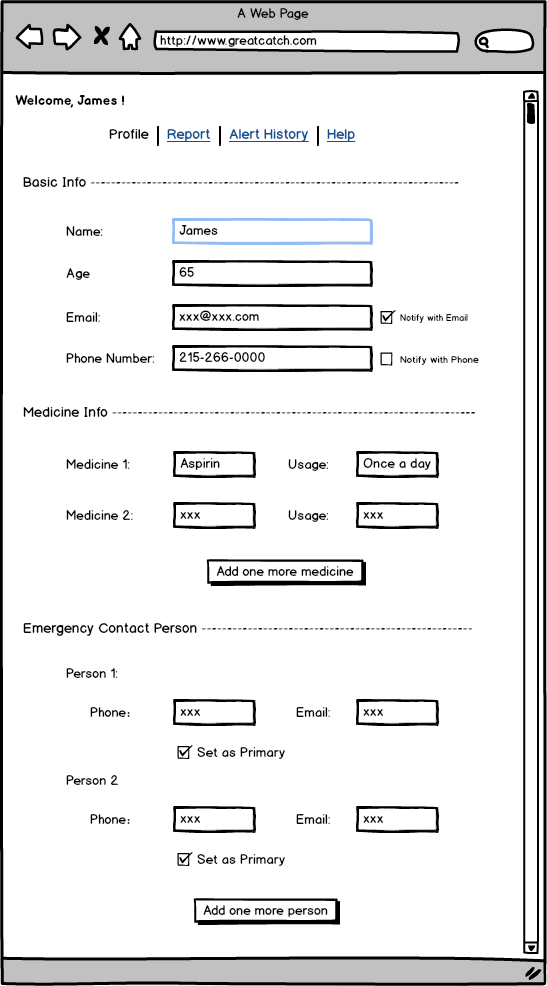


Figure 4. User information and emergency contact information

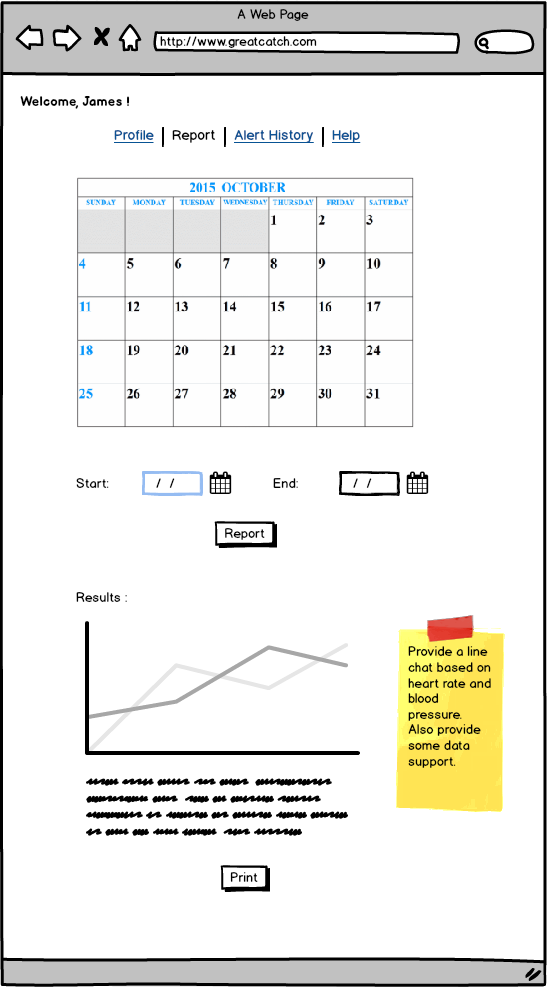


Figure 5. View/Print report

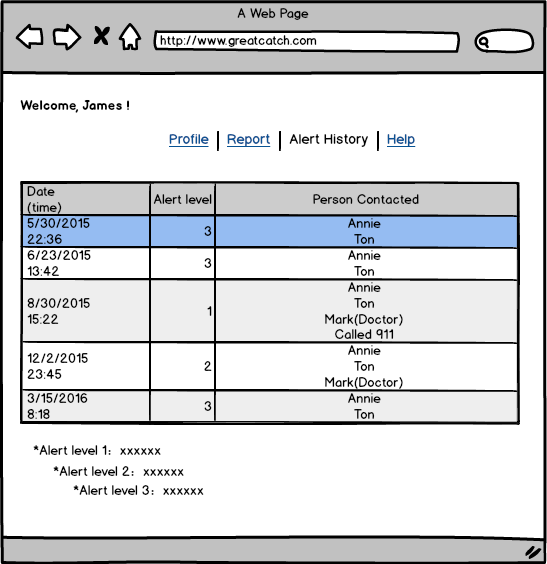


Figure 6. Alert history



Figure 7. Help Page

##### Hardware Interfaces

The system will not interface directly with any hardware components, outside of normal web browser interaction using the mouse and keyboard.

##### Software Interfaces

###### Fitbit API Component

The system will interact with the exposed Fitbit API endpoints. The user will provide authentication for the system to access their health data stored within the Fitbit servers, which will be read using HTTP. The API responses will be formatted in JSON and processed by the system [2].

###### Data Request Component

The system will provide an abstracted component to perform generic API calls and data formatting work. This component will manage the generic HTTP requests as well as passing retrieved data to the component associated with an individual wearable API.

###### Processed Data to Webserver

The Data Request Component (2.1.4.2) will be invoked by the web server upon user request. This call will return the processed data from the component back to the webserver process. This interaction will invoke an external script. The format of this data will be JSON.

###### User Database Connection

The web server will store and read data from a database which contains user information. Depending on the database implementation selected, this may be either HTTP or UNIX sockets. The format of this data will be a database query string containing the relevant request or values to modify.

##### Communication Interfaces

* + - 1. The system shall use HTTP to transfer data from the web server to the users, as well as obtain information from the various wearable APIs.
      2. The System shall use either HTTP or UNIX sockets to communicate with the database, depending on the implementation chosen.

##### Memory Constraints

* + - 1. The system shall use no more than 2GB of RAM and 100GB of external storage.

##### Operations

* + - 1. The system shall be operable with all mobile devices using the Chrome/Safari browsers.

##### Site Adaption Requirements

N/A

#### Product Functions

The following are a list of key functions of Great Catch

* Configure user settings and grant approvals to access wearable devices’ cloud server
* Pull user data from wearable devices’ cloud server using devices’ given API
* Pull user input data from system’s database
* Format all data input into a predefined data format
* Apply statistical analysis to check for abnormal pattern
* Alert designated contact person if abnormal patterns are detected, which are categorized in the level (low, mid, high severity) of alert
* Provide secondary statistical analysis using correlation for suggested reason for abnormal pattern
* Generate data visualization reports on the abnormal patterns
* View data reports and alert histories of any period

##### User Characteristics

##### Automatic Timed Sync

|  |  |
| --- | --- |
| **Scenario Name:** Automatic data syncing between Great Catch and Fitbit Server **ID number:** 1  **Short Description:** A senior decides he or she will not be wearing the device after 8:00 PM, Great Catch will start pulling data from Fitbit server after 8:00 PM  **Trigger:** User local time pass specific time point  **Type**: External | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * API token | * User |
| * Time | * Great Catch |
| **Major Outputs:**  **Description** | **Source** |
| * JSON data | * Fitbit server |
| **Major steps performed:** | **Information for Steps** |
| * Place Fitbit device next Fitbit server syncing area (either phone or computer) | * N/A |
| * Syncing between device and server begins | * User credential |
| * Local time passes specific point | * Geolocation and time zone |
| * Great Catch sends request to Fitbit server requesting data | * API token |
| * Request successful and data sync | * Server responses |

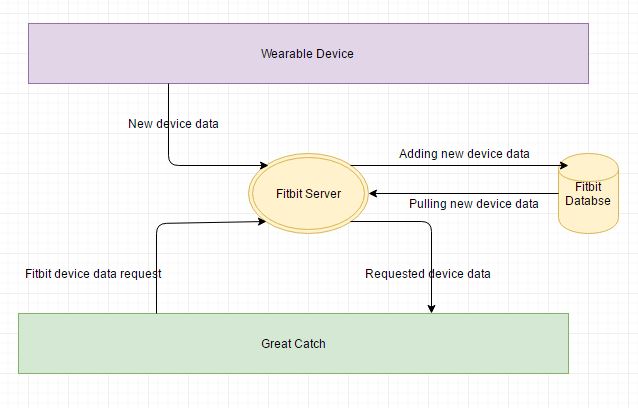


Figure 8. DFD - pulling data from server

|  |  |  |
| --- | --- | --- |
| **Data** | **Detail about data** | **Example data** |
| New Device Data | Any differences between new data and existing data in the database | * Steps * Distance * Calories * Active Minutes * Floors * Sleep * Heart Rate * Location |
| Request | HTTP request and response | * 0, 1, 2, 3, 4 * 200, 400 * GET, POST |

Table 2. Detail on data in ID Number 1

##### User Input

|  |  |
| --- | --- |
| **Scenario Name:** User information input **ID number:** 2  **Short Description:** User wants to manually input addition information such as medicines or existing medical conditions  **Trigger:** User submits new information  **Type**: Temporal | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * User input information | * User |
| **Major Outputs:**  **Description** | **Source** |
| * Data added confirmation | * Great Catch server response |
| **Major steps performed:** | **Information for Steps** |
| * User types in new information | * N/A |
| * User submits information | * User input information |
| * Information format validation | * User input information |
| * Attempt to save into Great Catch database | * User input information |
| * Great Catch server response |  |

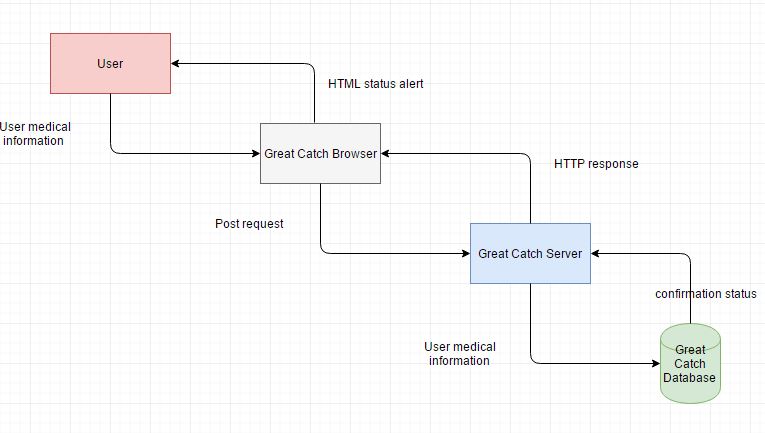


Figure 9. DFD - adding user medical information

|  |  |  |
| --- | --- | --- |
| **Data** | **Detail about data** | **Example data** |
| User medical information | Anything user wants to provide manually through Great Catch system | * medicine currently taking * existing medical condition * pain level |
| Request | HTTP request and response | * 0, 1, 2, 3, 4 * 200, 400 * GET, POST |
| HTML alert | Web page UI status display | * Succeed. Failed |

Table 3. Detail on data in ID Number 2

##### Data Sync Issue Alert

|  |  |
| --- | --- |
| **Scenario Name:** Data Sync Issue Alert **ID number:** 3  **Short Description:** Email alert is sent after failure to sync data from wearable API  **Trigger:** Failure response from wearable API endpoint  **Type**: External | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * HTTP failure response code | * Wearable API component |
| **Major Outputs:**  **Description** | **Source** |
| * Formatted email describing failure to retrieve wearable data * SMTP response | * User and caretaker * The system alert log database |
| **Major steps performed:** | **Information for Steps** |
| * Search email address of user and caretaker | * Email address |
| * Compose email | * General information on sync issue and steps to attempt to resolve |
| * Send email | * N/A |
| * Record SMTP response | * Email confirmed sent |

##### Low Activity Alert

|  |  |
| --- | --- |
| **Scenario Name:** Low Activity Alert **ID number:** 4  **Short Description:** Email alert is sent after low activity detected  **Trigger:** Detection of low user activity  **Type**: Temporal | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * Low wearable activity | * System data processing segment |
| **Major Outputs:**  **Description** | **Source** |
| * Formatted email describing low level of user activity * SMTP response | * User and caretaker * The system alert log database |
| **Major steps performed:** | **Information for Steps** |
| * Search email address of user and caretaker | * Email address |
| * Compose email | * General information on low activity and suggestion to increase |
| * Send email | * N/A |
| * Record SMTP response | * Email confirmed sent |

##### Abnormal Pattern Alert

|  |  |
| --- | --- |
| **Scenario Name:** Abnormal Pattern Alert **ID number:** 5  **Short Description:** Email alert is sent after an abnormal data pattern detected  **Trigger:** Detection of an abnormal data pattern  **Type**: Temporal | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * Abnormal data pattern | * System data processing segment |
| **Major Outputs:**  **Description** | **Source** |
| * Formatted email containing what section of data has determined abnormal * SMTP response | * User and caretaker * The system alert log database |
| **Major steps performed:** | **Information for Steps** |
| * Search email address of user and caretaker | * Email address |
| * Compose email | * General information on which section of data is determined abnormal |
| * Send email | * N/A |
| * Record SMTP response | * Email confirmed sent |

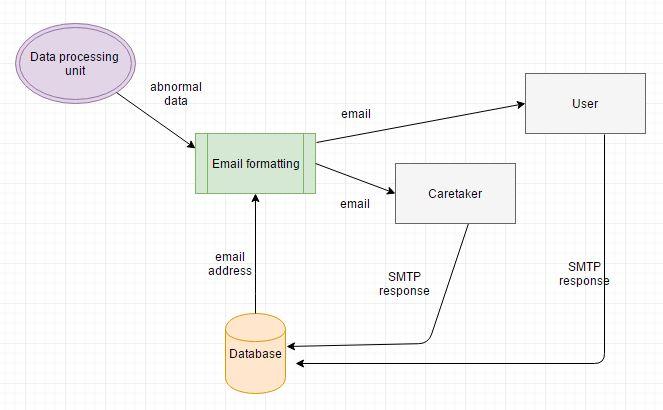


Figure 10. DFD - Email Alert

|  |  |  |
| --- | --- | --- |
| **Data** | **Detail about data** | **Example data** |
| Abnormal data | Data pattern that machine learning algorithm determined to be out of range compared to previous data | * Normal, Abnormal |
| Request | SMTP and HTTP request and response | * 0, 1, 2, 3, 4 * 200, 400 * GET, POST |
| Email | Email content containing alert information | * Low level: forgot to take medicine * Mid-level: no new data detected today * High level: cannot contact caretaker or user for a long period of time |

Table 4. Detail on data in ID Number 3

##### User Creation

|  |  |
| --- | --- |
| **Scenario Name:** User creation **ID number:** 6  **Short Description:** User account credential is added to the system’s database  **Trigger:** User click on create account and request is sent  **Type**: External | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * Username and password | * User input |
| * Gender, age, real name, DOB, email address, physical address | * User input |
| **Major Outputs:**  **Description** | **Source** |
| * Copy data to system’s database | * System’s database |
| **Major steps performed:** | **Information for Steps** |
| * Validate user input fields and format correctness | * User input |
| * Send information to system’s database |  |
| * Commit database change |  |

##### User Deletion

|  |  |
| --- | --- |
| **Scenario Name:** User deletion **ID number:** 7  **Short Description:** User account credentials are being deleted from the system’s database  **Trigger:** User click on delete account and request is sent  **Type**: External | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * Username and password | * User input |
| * Final confirmation from user | * User input |
| **Major Outputs:**  **Description** | **Source** |
| * Deletion of current user credential record from database | * System’s database |
| **Major steps performed:** | **Information for Steps** |
| * Validate user input fields sufficiency and format correctness | * User input |
| * Send information to system’s database |  |
| * Commit database change |  |

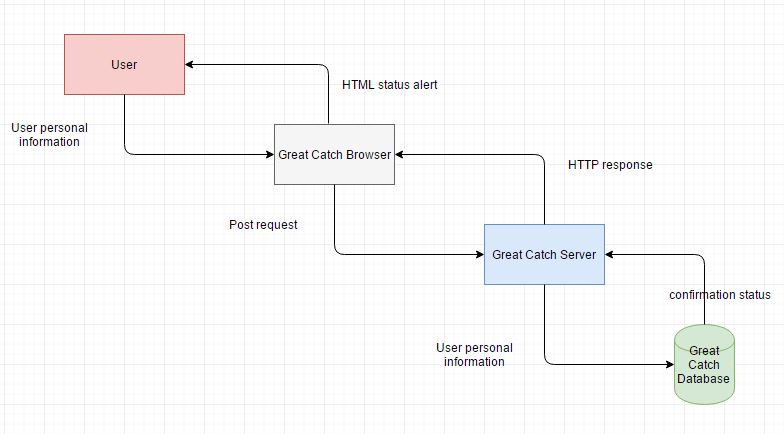


Figure 11. DFD - user creation and deletion

|  |  |  |
| --- | --- | --- |
| **Data** | **Detail about data** | **Example data** |
| User personal information | Personal information related to user | * Name * address * DOB * Gender * etc. |
| Request | SMTP and HTTP request and response | * 0, 1, 2, 3, 4 * 200, 400 * GET, POST |
| HTML alert | Web page UI status display | * Succeed. Failed |

Table 5. Detail on data in ID Number 4 and 5

##### Report Generation

|  |  |
| --- | --- |
| **Scenario Name:** Report Generation **ID number:** 8  **Short Description:** User requests to view certain data records during certain date range  **Trigger:** User clicks on view report and request is sent  **Type**: External | |
| **Major Inputs:** |  |
| **Description** | **Source** |
| * Data type | * User input |
| * Date range | * User input |
| **Major Outputs:**  **Description** | **Source** |
| * Summary of data for selected data type and date range | * Monitor |
| **Major steps performed:** | **Information for Steps** |
| * Validate that user input data type and date range matches existing data in database | * User input |
| * Send information to system’s database |  |
| * Query data and send to Crystal Report * Display report on monitor | * Actual data from database |

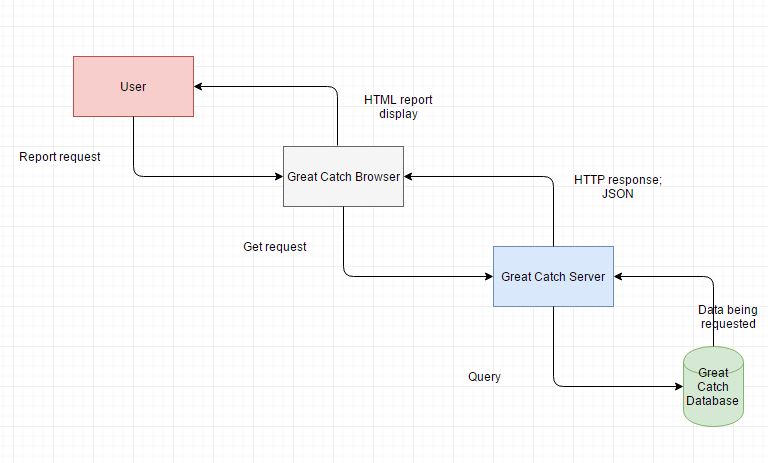


Figure 12. DFD - report generation

|  |  |  |
| --- | --- | --- |
| **Data** | **Detail about data** | **Example data** |
| User report request | Data category and range user want to view | * Calories from 1/1/2016 to 1/1/2017 * Steps from 1/1/2016 to 1/8/2016 |
| Request | SMTP and HTTP request and response | * 0, 1, 2, 3, 4 * 200, 400 * GET, POST |
| HTML display | Web page report displaying | * Bar/pie/line chart of selected data |

Table 7. Detail on data in ID Number 6

#### Constraints

Some of the conditions that limit the options of development for Great Catch include: GUI language will be limited to English, limited to HTTP/HTTPS.

#### Assumptions and Dependencies

The following assumptions and dependencies are needed to generate Great Catch requirements. First, it is assumed that the users have internet access and frequently use the wearable device. Second, it is assumed that the users have fundamental knowledge of accessing websites and utilizing computer components such as keyboard and mouse.

#### Apportioning of Requirements

All mobile application platform requirements are delayed until Cycle 3, time-permitting.

### Specific Requirements

#### External Interfaces

##### System Interfaces

The system shall be a website and will not require any special hardware.

The website shall be designed with a PostgreSQL, MySQL, or Neo4J database.

The data from users will be entered system using JSON format.

##### User Interfaces

System entry -- Figure 2: User can log in to their account with ID and password or create an account (Figure 3). When creating an account, users need to fill in all fields.

Once logged in, users will be redirected to their profile page (Figure 4). First-time users will need to complete their profile first.

*Note: Person who is set as the primary contact person will get alert by phone and email in any level of situation.*

Report page (Figure 5) -- Users can select a certain period and click on the report button. The system will generate a report based on personal data collected by the wearable. The report will include line charts (heart rate, blood pressure...) and some certain outrange data.

Alert History (Figure 6) -- This page will show the alert history, and how we define each level of alert.

Help page (Figure 7) -- Developer contact information. Any questions or suggestions can be submitted.

#### Functions

|  |  |
| --- | --- |
| **Function Category** | **Which Function Belong to This Category** |
| User and caretaker credential control | GC1, GC2, GC3, GC4 |
| Wearable device data gathering | GC5 |
| Data formatting and processing | GC6, GC7, GC8, GC9 |
| Alert categorizing and sending | GC10, GC11, GC12, GC13 |
| Report generating | GC14, GC15 |

Table 8. Summary of all functions for better understanding

##### GC1 - User Creation

Input: User’s basic information inputted by user

Action: The system will validate information sufficiency and format before storing this information into the database

Output: Success or failure message prompt will be shown

Notes: N/A

Priority: Low

##### GC2 - User Deletion

Input: User’s username and password, along with request to delete account

Action: The system will verify user’s credentials and delete associated account information out of the database

Output: Success or failure message prompt will be shown

Notes: N/A

Priority: Low

##### GC3 - Caretaker Contact Information Validation

Input: Caretaker’s contact information

Action: Sends email or text alert to provided address and asks for verification

Output: Verification success or failure message prompt will be shown

Notes: N/A

Priority: Low

##### GC4 - User and Caretaker Information Update

Input: User’s username and password and new information

Action: The system will verify user’s credentials and validate new input information format before committing the changes

Output: Success or failure message prompt will be shown

Notes: N/A

Priority: Low

##### GC5 - Pulling Data from Fitbit Server

Input: Fitbit user account token

Action: Data GET request to Fitbit server using the provided token

Output: Raw wearable device data

Notes: Token referring to OAuth generated access token through user permission to access Fitbit API

Priority: High

##### GC6 - Data Validating

Input: Status indicating GET request is successful and completed

Action: Data will be send to Python script in JSON format, convert to list format, and validate that there are enough data points to be processed

Output: If validation failed, an alert will be sent. Otherwise, the system will proceed with data formatting

Notes: Failure in data validation means either data was not synced correctly, or the user failed to wear the device

Priority: High

##### GC7 - Data Formatting

Input: Status indicating Data Validation was successfully completed

Action: Data will then be normalized and the system will parse out any empty fields

Output: Formatted data

Notes: Empty fields are being excluded because SVM does not take empty fields; if another algorithm is applied then such action will be modified

Priority: High

##### GC8 - Data Processing

Input: Status indicating Data Formatting was successfully completed

Action: Data will be trained and predicted using SVM algorithm

Output: 0 or 1 indicating normal or abnormal

Notes: N/A

Priority: High

##### GC9 - Determine Abnormal Parameter

Input: Abnormal status received from Data Processing

Action: Calculate each category of data’s linear regression and mean average standardization, and determine which category of data is statistically different

Output: If validation failed, an alert will be sent. Otherwise, the system will proceed with data formatting

Notes: Steps and Floors are two example categories of data; Definition of statistically different is change of slopes in linear regression

Priority: High

##### GC10 - Low-Severity Alert

Input: Data Validating failure

Action: Compose an email alert to the user indicating that data is not being synced; provide possible solution in detailed steps

Output: Formatted email

Notes: This level of alert will only be send to user

Priority: Low

##### GC11 - Mid-Severity Alert

Input: Abnormal status and results from GC9 highlight which parameter is of concern

Action: Compose an email alert to the user and their caretaker(s) indicating that this category of data showed an abnormal pattern, and advise actions to be taken

Output: Formatted email

Notes: This level of alert will send to user and caretaker

Priority: Low

##### GC12 - High-Severity Alert (E-mail)

Input: All categories of data are abnormal

Action: Compose an email alert to the user, caretaker(s), and health professionals indicating data is significantly abnormal

Output: Formatted email

Notes: This level of alert will send to user, caretaker, and health professionals

Priority: Low

##### GC13 - High-Severity Alert (Call)

Input: Long period of time with no data registered from user

Action: 911 will be called

Output: Phone call

Notes: N/A

Priority: Low

##### GC14 - Report Generating (Summarized)

Input: Data category and date range

Action: Query the database for desired data

Output: Summarized charts displaying on application

Notes: N/A

Priority: Low

##### GC15 - Report Generating (Raw)

Input: Data category and date range

Action: Query the database for desired data

Output: Table of raw data are shown

Notes: N/A

Priority: Low

#### Performance Requirements

Static numerical requirements include: supporting 3 simultaneous users accessing the user data; handling numeric, Boolean, and string data type; and reporting up to 3-year worth of data.

Dynamic numerical requirements include machine learning algorithm processing time of less than 1 minute and sending out alerts within 5 seconds of detecting an abnormal pattern.

#### Logical Database Requirements

The system will store 3 categories of information in the system: user and caretaker credentials, abnormal pattern data, and log information.

##### User and Caretaker Credential

The user and caretaker credential information shall be stored in text-based format. Information will include username, password, real name, age, email address, phone number, gender, and physical address. Each field shall be stored in a separate column in the table.

##### Abnormal Pattern Data

Any data set detected as abnormal shall be stored in numeric-based format. This information will be stored as raw numeric data in the table.

##### Log Information

Log information such as date and time of all the alerts sent, data sync, and data processed shall be stored in both text and date-time format. Each field shall be in separate columns in the table.

#### Design Constraints

##### Standards Compliance

N/A

#### Software System Attributes

##### Reliability

The system shall run in a modern web browser and will not require any special hardware.

##### Availability

The system shall have 99% availability.

##### Security

The system shall encrypt all connections that are used to access sensitive data.

##### Maintainability

All code shall be fully documented.

##### Portability

The software shall be designed to run on the following platforms:

• The system shall be portable to all current versions of Windows

• The system shall be portable to all current versions of the Marcos

### 

### 

### Table of Contributions

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

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Section** | **Writing** | **Editing** |
| **1** | **Introduction** | **WX, DG, TB, AM, YS** | **WX, DG, TB, AM, YS** |
| **2** | **Overall Description** | **WX, DG, TB, AM, YS** | **WX, DG, TB, AM, YS** |
| **3** | **Specific Requirements** | **WX, DG, TB, AM, YS** | **WX, DG, TB, AM, YS** |

A

Good.

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 based on 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.