# Software Requirements

Version 51, last updated by [rylandm](https://uocis.assembla.com/profile/rylandm" \t "_blank) at 2018-02-02

# Software Requirements Specification (SRS)

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 3-20 | Jiaxi Yan | Adding/Editing Use Cases |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[Software Requirements 1](#_Toc99546556)

[Software Requirements Specification (SRS) 1](#_Toc99546557)

[1.  Introduction 3](#_Toc99546558)

[1.1    Intended Audience and Purpose 3](#_Toc99546559)

[1.2    How to use the document 3](#_Toc99546560)

[2.  Concept of Operations 4](#_Toc99546561)

[2.1    System Context 4](#_Toc99546562)

[2.2 System capabilities 4](#_Toc99546563)

[3.  Use Cases 5](#_Toc99546564)

[Case 1: User Wants to Quit the web page 5](#_Toc99546565)

[Case 2: User Wants to Open web page 6](#_Toc99546566)

[Case 3: User Wants to Register a New Account 6](#_Toc99546567)

[Case 4: User Wants to Log in to the web page 6](#_Toc99546568)

[Case 5: User Wants to View User Information 7](#_Toc99546569)

[Case 6: User Wants to View Auxiliary Instructions 9](#_Toc99546573)

[4.    Behavioral Requirements 11](#_Toc99546578)

[4.1 System Inputs and Outputs 11](#_Toc99546579)

[4.2 Detailed Output Behavior 11](#_Toc99546580)

[4.3   Quality Requirements 12](#_Toc99546581)

[5.    Expected Subsets 12](#_Toc99546582)

[6.    Fundamental Assumptions 13](#_Toc99546583)

[7.    Expected Changes 13](#_Toc99546584)

## 1.  Introduction

### 1.1    Intended Audience and Purpose

This document is intended to provided information development process, ensuring that all system requirements are met. The following entities may find the document useful:

Primary Customer - This page will detail all of the web page requirements as understood by the production team. The customer should be able to determine that their requirements will be correctly reflected in the final product through the information found on this page.

User - A prospective user will be able to use this document to identify the main functionailty included in the web page. Furthermore, the web page will have a set of system requirements before the web page can be run. Details regarding these requirements can be found here.

Development Team - Details of specific requirements that the final software build must include will be located here. Developers can use this document to ensure the software addresses each of these requirements.

QA Team - By developing testing procedures founded in the system requirements, the QA Team can create a comprehensive testing regimen that will guarantee requirements are met.

### 1.2    How to use the document

Table of Contents:

1. Introduction

2. Concept of Operations - broad description of the purpose of the web pages

  2.1 System Context - details any specific system requirements the web pages will require to run

  2.2 System Capabilities - description in prose of all capabilities available to the user in the interaction

  2.3 Use cases - A detailed look at each functional requirement, describing the web pages context both before and after an action is taken

3. Behavioral Requirements - How the web page will interact with a user

  3.1 Input and output requirements - A description of allowed inputs and generated outputs

    3.1.1 Input - Describes any restrictions that will be placed on allowed input

    3.1.2 Output - Describes the range of outputs that can be generated

  3.2 Detailed Output Behavior - Output descriptions in prose

4. Quality Requirements - Requirements not pertaining to the function of the web page will be listed here

5. Expected Subsets - Expected levels of functionality at checkpoints during development

6. Fundamental Assumptions - Some specifics about input, output, or behavior upon which other requirements are founded will be listed here

7. Expected Changes - Future features and directions the project is expected to take

8. Appendence - Details aiding the understanding of this document

  8.1 Definitions and acronyms - Any technical terms or abbreviations will be spelled out here for ease of use of the document

    8.1 Definitions - Definitions of technical or unusual terminology

    8.1.2 Acronyms and Abreviations - Any abreviated terms will be expanded here

  8.2 References - any external references necessary or helpful to understanding this document will be listed here

## 2.  Concept of Operations

  The goal is to create a user friendly assistive interaction of the sensor. It will allow its users to manage prosthetics and help users use them better. The web page uses document box, the web page page, and the jump between web page pages implemented by buttons to interact with the user. For more details on the usage and capabilities of the web page read the section, [System Capabilities](https://uocis.assembla.com/spaces/cis422w18-team2/wiki/Software_Requirements#System%20Capabilities).

### 2.1    System Context

**System Requirements(Not Functional):**

Requires a system with Web pages display because all of the operations are performed through these pages.

**Supported Browsers:**

* Chrome
* Firefox
* Opera
* Brave

**Supported Language:**

* HTML
* CSS
* JavaScript
* Ajax Bootstrap

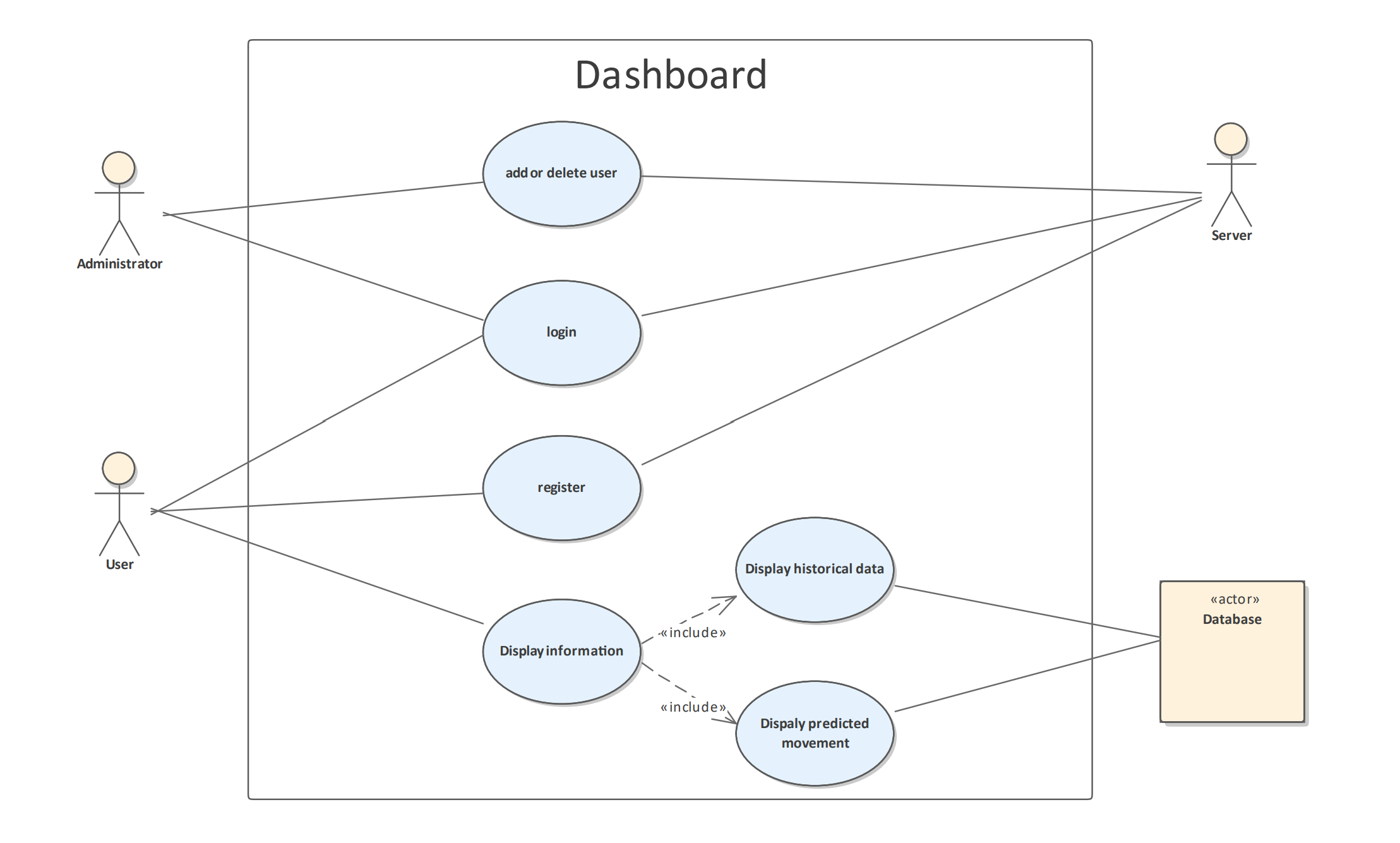
### 2.2 System capabilities

This auxiliary interaction is a standalone program. When the user uses it, they need to log in to enjoy the services provided by the auxiliary interaction. If the user does not have an account, they need to register an account by jumping to the registration page and when they complete registration then they can use this auxiliary interaction.

After the user logs in, they enter the page which the user can see all the charts which reflect the predictions, data Storage in Database or other stuff.

Users can view and modify personal information by going to the personal information interface through the personal information button in the menu. Users also can view the help Information window which displays guidelines for using the interactive system.

## 3.  Use Cases



### Case 1: User Wants to Quit or Re-log on the Web Page

**Players:**End User

**Goals:**The end user wants to close the pages and any opened equipment.

**Preconditions:**The web page is open and running.

**Case:**

1.1 From the File menu, the end user selects the "Close all and Quit" option.

1.2 The server disconnects from all currently added equipment

1.3 The pages terminates itself.

**Alternate Flows:**

1.2.1 Collected data have not uploaded or discarded

         The user is warned about quitting before uploading or discarding the data

      1.2.1.1 The user decides to upload data.

                  The web page closes after all data is uploaded.

  1.2.1.2 The user decides to discard data.

                  Discard data. The web page terminates.

**Exception Flows:**

1.2.2 The user forces the termination (by shutting down their machine, using Task Manager to Force Quit, etc)

         All data that is not uploaded or discarded is discarded.

**Postconditons:**If a user navigates to Task Manager (or equivalent process manager), there is no trace of the pages still run on Browser. No data should be displayed the next time the user opens web page.

### Case 2: User Wants to Open web page

**Players:**End User

**Goals:**The end user would like to run the web page so that they can register a new account or log in the web page.

**Preconditions:**The web page is opened on the user's browser.

**Case:**

2.1 The end user opens the page

**Postconditions:**The page is open, waiting for its next instruction from the end user.

### Case 3: User Wants to Register a New Account

**Players:** End User

**Goal:** The user would like to be applied for a new account by entering personal information.

**Preconditions:** The web page is opened.

**Case**:

3.1 From the login interface, the end user selects the "Register" option.

3.2 The window turns to the register page which included some personal information (e.g. email, password, etc.) input boxes.

3.3 Users can enter personal information.

3.4 User submit personal information.

3.5 The page reflect the result wrong or success.

**Alternate Flows**:

3.3 User chooses to return to the login page.

3.4.1 If the user is successfully registered, the login interface is returned.

3.4.2 If the user fails to register, let the user improve the information.

**Exception Flows:**

3.3.1 The user has not entered all the personal information or some of the information is not in the correct format or is not valid.

         The user is warned about submit before filling in all personal information correctly.

3.3.2 The account name already exists

         The user is prompted to change account name.

**Postconditions:**User gets a new account and can use it to log in to the web station.

### Case 4: User Wants to Log in to the Web Station

**Players:** End User

**Goal:**The end user would like to log in to the web station so that they can add equipment and collect data.

**Preconditions:** The log-in web page is opened.

**Case:**

4.1 From the login interface, the end user enters account number and password.

4.2 User selects the "Login" option.

**Alternate Flows:**

**Exception Flows:**

4.2.1 If the account or password is incorrect or invalid. The end user will be warned by the page.

         The end user is prompted to change account or password.

**Postconditions:** The web page is running, waiting for its next instruction from the end user.

### Case 5: User Wants to View User Information

**Players:**End User

**Goals:**The end user would like to view and change the user information.

**Preconditions:**The page is opened.

**Case**:

5.1 The end user enters the "User Information" interface

5.2 A new window opens with all user information

5.3 The end user can view all information.

**Alternate Flows:**

5.3.1 The end user can select and change one or more user information, such as passwords, etc.

         The end user confirms their changes. The web page saves all changed information

**Exception Flows:**

5.3.2 The information after user changes is invalid or incorrectly formatted. The end user will be warned by the page.

         The end user is prompted to change the information.

**Postconditions:**User information should be permanently changed and the next time the end user views user information or logs in, the changed information should be displayed.

### Case 6: User Wants to View Usage Guide

**Players:** End User

**Goals:** **The user is able to access information about the function of each function key**.

**Preconditions:**The web page is open and running.

**Case**:

9.1 The user hovers his mouse over a function key.

9.2 web page shows function information of the function key

9.3 User views the information and gain help.

**Alternate Flows:**

**Exception Flows:**

**Postconditions: User should be able to obtain detailed function instructions when user’s mouse hovers over a function key**

## 4.    Behavioral Requirements

### 4.1 System Inputs and Outputs

#### 4.1.1 Inputs

Inputs to the web page come from the user. Users can select options such as personal information, device information, help, and so on from the main menu. On the login screen, you can enter the text box, and on the personal information screen, you can choose to modify the information.

Input

Inputs When Editing Personal information:

\*account number: An account number that is unique to each user, consisting of letters and numbers

\*password: A string consisting of at least n digits number and characters to whether the guest is the user

\*user name: A contact can have a first name, or last name, or both. The first and last names will be separate fields. The system will accept any characters in the first and last name fields.

\*user information: The information provided by the user, which is mainly the user's contact information and physical condition

#### 4.1.2 Outputs

The GUI display all information, such as user information, device information, help information and each windows contains any created contacts and interactable buttons for the user.

Outputs to The User:

\*data record: The program starts to collect data, the "Stop Collecting" and "Re-Collecting" buttons appear new, if you re-collect or stop collecting data, the system will generate a window asking whether to discard the data.

\*personal information of user: user detailed information, including the account number, email, real name, body information and so on. Users can click the "Modify Information" button to modify the information and display it.

\*help information: the window contains the instruction manual to guide the user to use the system

### 4.2 Detailed Output Behavior

The output is a GUI that provides the user with the ability to give inputs to the interaction.

After the user logs in,we can see any charts or data show in the page if we have collected the data on the mobile terminal.

Through the "User Information" and "Help Information" buttons on the main menu, users can view the user information page and the help information page. The user information page displays the user account, real name, physical condition and other related information, and the bottom of the page is composed of a "Modify Information" button, where users can modify personal information. The Help Information screen displays guidelines for using the interactive system.

### 4.3   Quality Requirements

The web page must be competitive with similar web pages in regards to performance, reliability, consistency, and scalability.

Performance: Responsiveness to user input

      \* Standard actions that manipulate interaction should not exceed 500ms execution time.

Reliability: Confidence that actions taken will not result in errors, and that changes made to program are persistent

      \* User input should not produce faults or errors that impact or hinder use of the web page

      \* Any modifications to program should produce a lasting change that persists through any following series of actions taken by the user.

Consistency: Persistent data in program contents

      \* program contents should be modifiable after being added in such a way that target fields can be changed without affecting data in other fields.

Scalability: Ease of extending web page capabilities

      \* web page should be modularized such that adding/extending features and functions only require changes to a single component and the interface with that component, if applicable.

## 5.    Expected Subsets

L0:

-Basic GUI with ability to check the device and view device details

-Ability to register and sign in

-Ability to add new device and delete useless device

-Ability to modify user information

-Ability to control data

L1:

-Complete GUI for making the interface user-friendly

-Complete the Administrator module

## 6.    Fundamental Assumptions

The web page can run on any system that is capable of running Qt.

The web page will not terminate when all windows are closed.

Software updates will be downloaded by the end user as opposed to pushed out by the developers.