|  |
| --- |
| UAH Fit Vault Detailed Design |
| CPE 656/658 Software Studio |
| Timothy R. Wilkins  Whit J. Sisulak  Glen L. Riden  James J. Duggan IV |

11/7/2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | Description of Change | Author |
| 0.1 | 11/07/15 | Initial Draft | W. Sisulak |

Contents

[Revision History i](#_Toc434658776)

[1 Introduction 1](#_Toc434658777)

[1.1 Purpose 1](#_Toc434658778)

[1.2 Scope 1](#_Toc434658779)

[1.3 Product Perspective 2](#_Toc434658781)

[1.4 Product Functions 2](#_Toc434658782)

[1.5 User Characteristics 2](#_Toc434658783)

[1.6 Constraints 3](#_Toc434658784)

[1.7 Application Framework 4](#_Toc434658785)

[2 System Architecture Description 4](#_Toc434658786)

[2.1. Architecture Overview 4](#_Toc434658788)

[2.2. Database Schema 5](#_Toc434658789)

[2.3. Class Descriptions 5](#_Toc434658790)

[2.3.1. Views 5](#_Toc434658791)

[2.3.2. Controllers 7](#_Toc434658792)

[Appendix A: Use Case Specifications 11](#_Toc434658793)

[Appendix B: Traceability Matrix 22](#_Toc434658794)

Detailed Design

# Introduction

## Purpose

The purpose of this document is to define the software requirements specification for UAH Fit Vault software projects. In addition, this document describes the scope, design constraints, and interfaces of the system. The intended audience for this document includes system developers, testers, customers, and any other stakeholders.

## Scope

The UAH Fit Vault software package will be a web application that will accept medical data from users and display the data in a meaningful way. There are two major components to this software. The first is the data collection tool that is used by the users to upload their medical data that is recorded by one of the supported wearable medical devices. There are three different medical devices supported for this project that record various types of data. The data provided by these devices consists of different file formats, and the data is different from device to device. The software will have to determine the contents of each file and how to process them. Due to how long data transfers take to download the data from a device, there may be a need in the future to convert the data from a binary format to another format in order to speed up the process of getting data off the device. The software needs to able to take in files provided by the medical devices process the files, and store the data in a database. The software should have the ability to process multiple files at a time as well as individual files and allow for an activity to be assigned to them by date and time.

The other major component of the web application is the data analysis tools used to analyze the data that is captured from the data collection tool mentioned above. The software needs to perform data analysis over different intervals of time such as one week, one month, etc. There will need to be some way to manage user access to the various medical data that has been inserted into the database that this software will access. Below are some proposed data analysis ideas that can be incorporated into the project.

* Simple Moving Average
* Data correlation discovery between the multiple devices.
* Possibly determine when an individual moves from walking to running or simply being able to identify the activities that were being performed while the data was being captured.

The data analysis possibilities will likely not fully be realized until the project team understands the different types of data that are available. Also, there will need to be collaboration with the customer for additions or changes to the data measurements provided by this software. The web application will have to have different levels of user access which will be defined later in this document. Overall Description



## Product Perspective

UAH Fit Vault is a web hosted system that provides the functionality described in the product functions section of this document. It includes subsystems to address all of the defined requirements. In addition, UAH Fit Vault includes interfaces to its web services platform and database services. These interfaces will be implemented according to industry standards.

## Product Functions

* Web based and internet accessible.
* Medical experiment creation.
  + Store experiments for later viewing.
* Account Management including role-base user accounts.
  + Patient
  + Physician
  + Experiment Administrator
  + System Administrator
* Allows for confidential medical data storage.
* Allows for viewing medical data.
* Allows for Physician/Patient confidentiality.
* Provides a means to download experiment results and personal or patient medical data.
* Provides a user friendly interface.

## User Characteristics

* Patient
  + Patient accounts shall not have any personal identifiable information
  + Since there is no PII, password resetting will be performed with either security questions or a temporary password will be mailed to a “call-back email”
  + Patient accounts shall contain information such as userid, username, password, age, gender, weight, height, race, nationality and location (at a high-level to not be identifiable, i.e. state)
  + Patients shall only be allowed to upload data and view their own data from one of the three wearable medical devices listed later in this document.
* Physician
  + Will be able to add patients to the system.
  + Patients shall only be associated with one physician.
  + Physician accounts shall have information like username, password, active status and email.
  + Physicians shall be allowed to view the data for their associated patients.
* Experiment Administrator
  + People looking to gather health/fitness data on **anonymous** individuals that meet certain criteria (specific age, weight, race, etc.).
  + Experiment Administrators shall be able to specify criteria that they want to research and be provided data accordingly.
  + Experiment Administrators shall not be associated with any physician.
  + Experiment Administrators shall contain information such as a username, password, active status, etc.
* System Administrator
  + System administrators shall have the ability to enable, disable, add and remove all users except patients.
  + System administrators will have the ability to reset passwords.
  + System administrators shall be able to update physician, patient and experiment administrator account information.
  + System administrators shall not be allowed to view patient health data.

## Constraints

The software must be accessible on Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari. The follow table illustrates the minimum recommended version for each browser.

|  |  |
| --- | --- |
| Internet Browser | Minimum Recommended Version |
| Google Chrome | v46 |
| Mozilla Firefox | v41 |
| Internet Explorer | v11 |
| Apple Safari | v9.0.1 |

## Application Framework

The application is designed to work with the Microsoft ASP.NET MVC web application framework. This framework uses the Microsoft .NET Framework and Common Language Runtime languages (specifically C#)

Using ASP.NET MVC helps guide the design process. The framework takes ideas from the traditional design patterns of Model-View-Controllers and bootstraps a web application allowing for simple design and allows us to leverage a lot of hard work done by people with far more expertise in web application framework development.

# System Architecture Description



## Architecture Overview



Figure : Architectural Overview

The UAH Fit Vault application is a Web application that is comprised of and encompasses the presentation, business logic, and data access layers. The system interacts with the end user via a web interface and with an external SQL database hosted on a remote server. It is assumed that the user is authenticated to use the application.

* The presentation layer consists of a web interface for data processing and reporting. The data processing portion will have the ability to select files for processing. It will have a button to initiate the processing of files. An addition screen will allow the user to enter and activity to be applied to the data by date/time. Once they have been processed and uploaded into the database the system will return to the user a display of the results once completed. The data metrics and reporting functionality is another portion to be defined later.
* The business logic layer will consist of all the back end code comprised of classes to handle the parsing and processing of files as well as provide the logic to render metrics and reports. It will use the DB handler in the data access layer to interact with the database and the front end GUI to receive data from and display results back to the user.
* The data access layer will consist of all the back end code needed to interact with an external SQL database utilizing the data access library of choice. Examples include ADO.NET (OleDb, ODBC), Entity Framework, or any other chosen data interaction libraries. A precondition that the host computer is connected to the internet exists.

## Database Schema

TBD

## Class Descriptions

### Views

This section contains the views which are passed to other views in the system in order to refer to themselves via ViewModels. ViewModels are simple passive entity classes which should have all their fields set where appropriate. Class diagrams showing how the ViewModels relate to the various controllers can be found in the Controller section.

**Login View**

This view is responsible for providing the user an interface to login to the system.

|  |  |  |
| --- | --- | --- |
| Class Properties | | |
| **Name** | **Type** | **Description** |
| Id | Int | Gets or sets the id of the user |
| Username | String | Gets or sets the username of the user. |
| Password | String(password field) | Gets or sets the password of the user. |

**File Selection View**

This view is responsible for providing the user an interface to select files for processing.

|  |  |  |
| --- | --- | --- |
| Class Properties | | |
| **Name** | **Type** | **Description** |
| Location | List<String> | Gets or sets the file location(s) |

**Activity Selection View**

This view is responsible for providing the user an interface to associate an activity with a file.

|  |  |  |
| --- | --- | --- |
| Class Properties | | |
| **Name** | **Type** | **Description** |
| FileActivity | List<Activity> | Activities for files based on date/time |

**Activity Struct**

Used for activity selection

|  |  |  |
| --- | --- | --- |
| Struct Properties | | |
| **Name** | **Type** | **Description** |
| Activity | Struct  {  Location : String  TimeStamp : DateTime  Activity : Enum  } | Struct to hold file location, date/time stamp, activity |

**Experiment View**

TBD.

|  |  |  |
| --- | --- | --- |
| Class Properties | | |
| **Name** | **Type** | **Description** |
|  |  |  |

**Admin View**

TBD.

|  |  |  |
| --- | --- | --- |
| Class Properties | | |
| **Name** | **Type** | **Description** |
|  |  |  |

### Controllers

This section contains the controllers to the web application. Each public method in a controller is mapped to a single web request by the user and returns a single View. Controller actions are defined with a description of what the action does, the HTTP endpoint (URL) which will be routed to the controller’s action, a listing of all query parameters which can be sent to the method, the expected View which will be returned to the user, as well as any exceptional cases which may occur.

**Login Controller**

The login controller handles the authentication and access level of the users. It is tied to the login view.

**/Login/**

The specific methods in the login controller are detailed below (one per method).

|  |  |  |  |
| --- | --- | --- | --- |
| Authenticate | | | |
| Internal method, authenticates the user and sets the access level. | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
| UserName | In | Int | The username of the user. |
| Password | In | String (password field) | The password of the user. |
| [Return] | Out | Web Response Redirect | Redirects to the appropriate controller based on authentication and use role. |
| **Exceptions** | | | |
| 404 Not Found: No user exists with the given id. | | | |

**File Selection Controller**

The data processing controller handles the file selection of the users. It is tied to the file selection view.

**/Select/**

The specific methods in the process controller are detailed below (one per method).

|  |  |  |  |
| --- | --- | --- | --- |
| SelectFiles | | | |
| Internal methods, initiated by a button press after the file(s) have been selected. The file locations are stored as a list of strings. | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
| [Return] | Out | Web Response Redirect | Redirects to the activity controller. |
| **Exceptions** | | | |
|  | | | |

**Activity Selection Controller**

The data processing controller handles the authentication and access level of the users. It is tied to the activity view and takes in the selected files view model.

**/Activity/**

The specific methods in the process controller are detailed below (one per method).

|  |  |  |  |
| --- | --- | --- | --- |
| SelectActivities | | | |
| Internal method, initiated by a button press each time an activity is added. The activities are stored as a list of activity structs. | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Process | | | |
| Internal method, initiated by a button press after the activitie(s) have been added and the file is ready for processing. Uses the validation method Uses the database model to upload the data. | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
| [Return] | Out | String | Returns the result. |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ValidateFiles | | | |
| Internal methods, initiated by the Process method. Validates the data. | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
| File | In | String | Validates the data. |
| Result | Out | Boolean | Valid or invalid |
| **Exceptions** | | | |
|  | | | |

**Experiment Controller**

TBD.

**/Experiment/**

The specific methods in the experiment controller are detailed below (one per method).

|  |  |  |  |
| --- | --- | --- | --- |
| Create | | | |
| Creates the experiment | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| View | | | |
| Views the experiment | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Delete | | | |
| Deletes the experiment | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

**Admin Controller**

The data processing controller handles the administrative functions of user accounts. It is tied to the admin view.

**/Admin/**

The specific methods in the process controller are detailed below (one per method).

|  |  |  |  |
| --- | --- | --- | --- |
| Create | | | |
| Creates the user account | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Edit | | | |
| Edits the user account | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Delete | | | |
| Deletes the user account | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Enable | | | |
| Enables the user account | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Disable | | | |
| Disables the user account | | | |
| **Query Parameters** | | | |
| **Parameter Name** | **In / Out** | **Type** | **Description** |
|  |  |  |  |
| **Exceptions** | | | |
|  | | | |

Appendices

# Appendix A: Use Case Specifications

This section represents the use case specifications for the functions defined in the requirements specification for the two pieces of software needed to complete this project.

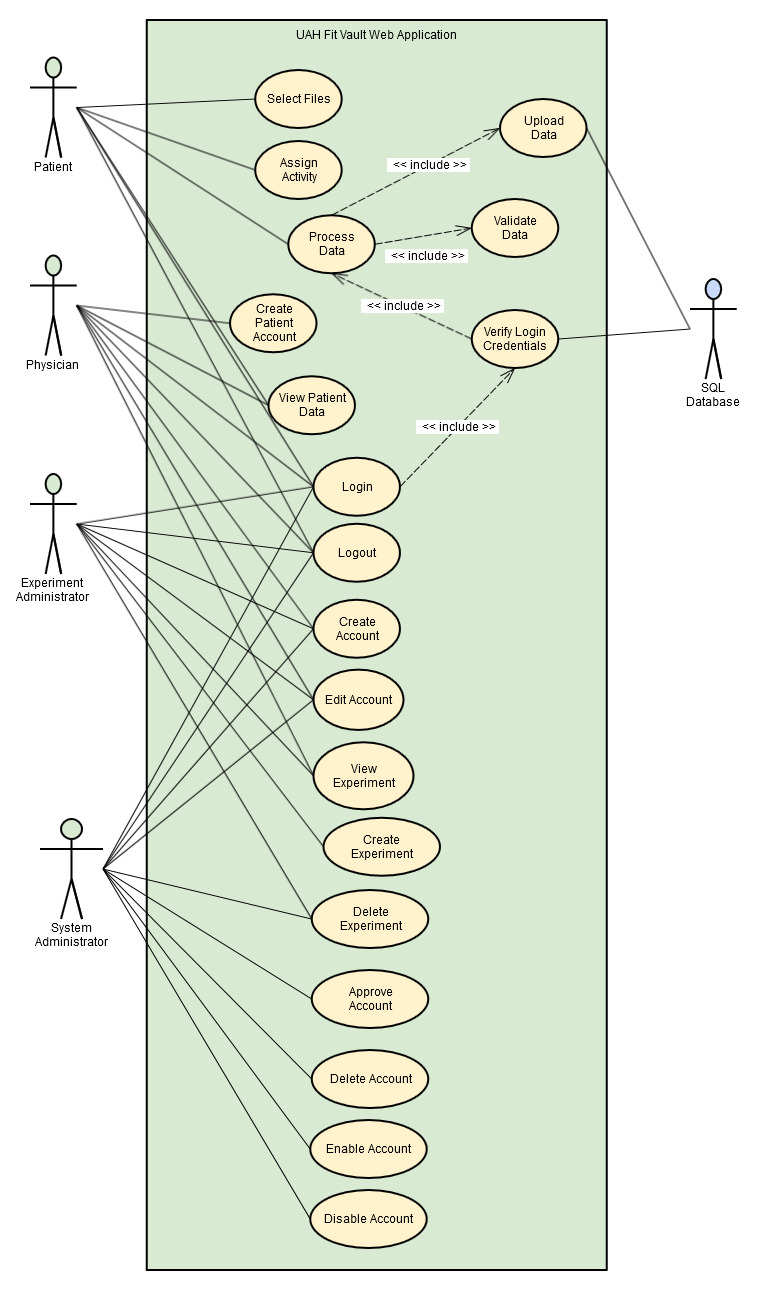


Figure A-1: Use Case Diagram

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_001 | |
| Use Case Name | Select Data Files | |
| Actor(s) | Patient | |
| Purpose | Select the data file or files to be processed and uploaded to the database. | |
| Overview | The patient needs a way to select files to be uploaded | |
| Cross-References | *<requirements to be inserted here after finalized>* | |
| Pre-conditions | The patient has access to the software and is authenticated. The user has the appropriate permissions for importing data. | |
| Post-conditions | **PASS:** The files have been selected for processing.  **FAIL:** No files are available for processing. | |
| Course of Events | | |
| Actor Actions | | **System Response** |
| 1. Patient navigates to the directory of files they wish to process | |  |
| 2. Patient selects all files they wish to upload. | |  |
|  | | 2. The system loads the files. |

|  |  |
| --- | --- |
| Use Case ID | UC\_001a |
| Use Case Name | Select Activity |
| Actor(s) | Patient |
| Purpose | Select the activity or activities for the data by date and time. |
| Overview | The patient needs a way to select an activity or activities to assign to the data. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | The patient has access to the software and is authenticated. The user has the appropriate permissions for importing data. A file or files have been selected. |
| Post-conditions | **PASS:** The files have been assigned and activity or activities.  **FAIL:** No activity or activities were assigned. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. Patient navigates to the directory of files they wish to process |  |
| 2. Patient selects all files they wish to upload. |  |
|  | 2. The system assigns the activity or activities to the data by date and time. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_002 | |
| Use Case Name | Process Data | |
| Actor(s) | Patient | |
| Purpose | Patient initiates file processing | |
| Overview | The patient initiates the action to process the loaded files | |
| Cross-References | *<requirements to be inserted here after finalized>* | |
| Pre-conditions | The files have been selected via UC\_001 or UC\_002. The user has been authenticated and has the appropriate permissions. | |
| Post-conditions | **PASS:** The files are ready to be uploaded in to the database.  **FAIL:** The files will not be uploaded to the database. | |
| Course of Events | | |
| Actor Actions | | **System Response** |
| 1. Patient initiates the file processing action. | |  |
|  | | 2. The system processes the files. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_003 | |
| Use Case Name | Validate Data | |
| Actor(s) | System | |
| Purpose | Validate processed data | |
| Overview | The system validates the data being processed based on the device and medical data type. | |
| Cross-References | *<requirements to be inserted here after finalized>* | |
| Pre-conditions | The files have been loaded via use case #1 or #2 and the processing step has been initiated. The user has been authenticated and has the appropriate permissions. | |
| Post-conditions | **PASS:** The files are determined to be valid or invalid.  **FAIL:** System error message displayed to the user informing that the files could not be processed. | |
| Course of Events | | |
| Actor Actions | | **System Response** |
| 1. The process data use case initiates the validation of the files. | |  |
|  | | 2. The system validates the files. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_004 | |
| Use Case Name | Upload Data | |
| Actor(s) | System | |
| Purpose | Upload data into the database | |
| Overview | The software uploads the validated data into the SQL database. | |
| Cross-References | *<requirements to be inserted here after finalized>* | |
| Pre-conditions | The data has been validated and is ready for the upload and the database is online. The user has been authenticated and has the appropriate permissions. | |
| Post-conditions | **PASS:** The data is uploaded into the database.  **FAIL:**  An error message is displayed informing the user the upload failed. | |
| Course of Events | | |
| Actor Actions | | **System Response** |
| 1. The process data use case initiates the upload of the data | |  |
|  | | 2. The system uploads the data into the SQL database. |

|  |  |
| --- | --- |
| Use Case ID | UC\_005 |
| Use Case Name | Login |
| Actor(s) | Patient, Physician, Experiment Administrator, System Administrator |
| Purpose | Procedure for logging into the system |
| Overview | The user needs to be able to login to the web application software in order to access the system. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | The user must have created an account on the web site.  The user has launched a web browser and navigated to the web application. |
| Post-conditions | **PASS:** The user has successfully logged into the system and can access various tools provided.  **FAIL:** The user remains logged out of the system and cannot use the system. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User enters their user name and password on the login page. |  |
|  | 2. The system validates their user credentials and logs the user into the system. |

|  |  |
| --- | --- |
| Use Case ID | UC\_006 |
| Use Case Name | Logout |
| Actor(s) | Patient, Physician, Experiment Administrator, System Administrator |
| Purpose | Procedure for logging out of the system. |
| Overview | From a security stand point the user shall have the ability to log them out of the web application to prevent unauthorized data entry if a computer is shared with another user. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | The user must have created an account on the web site.  The user has launched a web browser and navigated to the web application.  The user is currently logged into the system. |
| Post-conditions | **PASS:** The user can no longer access the system.  **FAIL:** The user remains logged into the system. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User selects the logout action. |  |
|  | 2. The software deletes the user’s login credentials from the application session. |

|  |  |
| --- | --- |
| Use Case ID | UC\_007 |
| Use Case Name | Verify Login Credentials |
| Actor(s) | Patient, Physician, Experiment Administrator, System Administrator |
| Purpose | Procedure for verifying the login credentials of a user. |
| Overview | The username and password entered by the user must be validated prior to granting the user a successful login into the system. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | The user must have created an account on the web site.  The user has launched a web browser and navigated to the web application. |
| Post-conditions | **PASS:** The user has been logged into the system.  **FAIL:** The user remains logged out of the system and an error message is present to the user informing that there was an invalid username or password entered. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User enters username and password into the login modal dialog. |  |
|  | 2. The software queries the account table in database to authenticate the user. |

|  |  |
| --- | --- |
| Use Case ID | UC\_008 |
| Use Case Name | Create Account |
| Actor(s) | Physician, Experiment Administrator, System Administrator |
| Purpose | Procedure for creating a new account in the system |
| Overview | A new user wishing to use the system must register a new account using the web application account creation tools. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer. |
| Post-conditions | **PASS:** New user account has been created and is pending approval.  **FAIL:** No new account is created and an error message is displayed to the user. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User selects link from login screen to register a new user. |  |
|  | 2. The system loads the account creation page for the user. |
| 3. User enters all information required. |  |
|  | 1. The system creates and account and sends an approval notification to the system administrators to approve the new account. |

|  |  |
| --- | --- |
| Use Case ID | UC\_009 |
| Use Case Name | Edit Account |
| Actor(s) | Physician, Experiment Administrator, System Administrator |
| Purpose | Procedure for editing an existing account in the system |
| Overview | In the event the user needs to edit any of their account information the system will allow the user to do so from the account management view. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer.  User must be logged into the system.  User must be navigate to the account management page. |
| Post-conditions | **PASS:** User account information has been updated.  **FAIL:** User account remains the same. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User enters any changes into the account management view. |  |
| 2. User selects to save changes. |  |
|  | 3. The system updates the user’s account with the updated account information provided by the user. |

|  |  |
| --- | --- |
| Use Case ID | UC\_010 |
| Use Case Name | Delete Account |
| Actor(s) | System Administrator |
| Purpose | Procedure for deleting an existing user account. |
| Overview | When a user account is no longer active or needed a request can be made to the system administrators to delete user accounts. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | System administrator must have an active internet connection.  System administrator must have accessed the web application through a browser on their personal computer.  System administrator must be logged into the system.  Account deletion has been requested. |
| Post-conditions | **PASS:** User account has been removed from the system.  **FAIL:** User account remains the in the system. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. System administrator navigates to their admin tools. |  |
| 2. System administrator clicks the delete button for the account. |  |
|  | 3. The system provides a verification dialog asking the system administrator if they are sure they wish to delete the account. |
| 4. System administrator clicks the “Yes” options. |  |
|  | 5. The system deletes the user. |

|  |  |
| --- | --- |
| Use Case ID | UC\_011 |
| Use Case Name | Approve Account |
| Actor(s) | System Administrator |
| Purpose | Procedure for approving the creation of a new account. |
| Overview | After a user creates a new account a system administrator must approve the new account prior to it becoming active for use. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | System administrator must have an active internet connection.  System administrator must have accessed the web application through a browser on their personal computer.  System administrator must be logged into the system.  A new account has been created pending approval. |
| Post-conditions | **PASS:** User account is now active and the user can now access the various tools in the system.  **FAIL:** User account does not have access to any system tools. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. System administrator navigates to their admin tools. |  |
| 2. System administrator clicks the approve button for any pending accounts. |  |
|  | 3. The system unlocks the account and allows the user to access the system. |

|  |  |
| --- | --- |
| Use Case ID | UC\_012 |
| Use Case Name | Create Experiment |
| Actor(s) | Experiment Administrator |
| Purpose | Procedure for creating a new experiment. |
| Overview | An experiment administrator may build experiments based on the data that is available in the system. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | Experiment administrator must have an active internet connection.  Experiment administrator must have accessed the web application through a browser on their personal computer.  Experiment administrator must be logged into the system. |
| Post-conditions | **PASS:** A new experiment has been created.  **FAIL:**  No experiment has been created. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. Experiment administrator navigates to the experiment module. |  |
|  | 2. The system pulls the various types of data from the database and builds a set of lists that can be used to build an experiment. |
| 3. Experiment administrator uses the tools provided to build an experiment. |  |
|  | 4. The system displays the results of the experiment. |
| 5. Experiment administrator selects to save the experiment. |  |
|  | 6. The system saves the experiment so it can be ran again later. |

|  |  |
| --- | --- |
| Use Case ID | UC\_013 |
| Use Case Name | View Experiment |
| Actor(s) | Physician, Experiment Administrator |
| Purpose | Procedure for viewing an existing experiment’s results. |
| Overview | Once an experiment has been created, it will be viewable again from an existing experiments list. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer.  User must be logged into the system. |
| Post-conditions | **PASS:** Experiment results are displayed to the user.  **FAIL:** Error message is displayed instead of experiment results. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User navigates to experiment module. |  |
| 2. User selects experiment to view from a list of existing experiments. |  |
|  | 3. The system displays the experiment results to the user. |

|  |  |
| --- | --- |
| Use Case ID | UC\_014 |
| Use Case Name | Delete Experiment |
| Actor(s) | Experiment Administrator, System Administrator |
| Purpose | Procedure for deleting an experiment. |
| Overview | In the event that an experiment is no longer needed it can be deleted from the system. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer.  User must be logged into the system. |
| Post-conditions | **PASS:** Experiment has been removed from the system.  **FAIL:** Experiment remains in the system. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. User navigates to experiment admin module. |  |
| 2. User clicks the delete button for an experiment. |  |
|  | 3. The system deletes the experiment. |

|  |  |
| --- | --- |
| Use Case ID | UC\_015 |
| Use Case Name | Enable User |
| Actor(s) | System Administrator |
| Purpose | Procedure for enabling a disabled user. |
| Overview | If a user has been disabled, a system administrator can enable the user from the system administrator tool. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer.  User must be logged into the system. |
| Post-conditions | **PASS:** User account is now enabled.  **FAIL:** User account remains disabled. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. System administrator navigates to admin module. |  |
| 2. System administrator clicks the enable button for the disabled user. |  |
|  | 3. The system enables the user account. |

|  |  |
| --- | --- |
| Use Case ID | UC\_016 |
| Use Case Name | Disable User |
| Actor(s) | System Administrator |
| Purpose | Procedure for disabling an active user. |
| Overview | If a user can request to a system administrator that a user account be disabled. |
| Cross-References | *<requirements to be inserted here after finalized>* |
| Pre-conditions | User must have an active internet connection.  User must have accessed the web application through a browser on their personal computer.  User must be logged into the system. |
| Post-conditions | **PASS:** User account is now disabled.  **FAIL:** User account remains enabled. |
| Course of Events | |
| Actor Actions | **System Response** |
| 1. System administrator navigates to admin module. |  |
| 2. System administrator clicks the disable button for the active user. |  |
|  | 3. The system disables the user account. |

# Appendix B: Traceability Matrix

TBD