

# REQUIREMENTS SPECIFICATION

## 1 INTRODUCTION

This is a Requirements Specification document for a data-sharing web application platform for cardiomyopathies. This document describes the scope, objectives and goal of the platform. In addition to describing non-functional requirements, this document models the functional requirements with use cases. This document is intended to direct the design and implementation of the web application and to aid discussion.

## 2. USER PERSONA

The system will mainly be used by clinicians, cardiologists and medical/healthcare experimental investigators. It will also be used by academic researchers, medical students and possibly cardiomyopathy patients with an interest in understanding the data.

## 3 BACKGROUND

Cardiomyopathy is a disease of the heart muscle that makes it harder for your heart to pump blood to the rest of your body. The most common forms of cardiomyopathy are Hypertrophic Cardiomyopathy, Dilated Cardiomyopathy and Arrhythmogenic right ventricular Cardiomyopathy. These cardiomyopathies occur due to mutations in genes that encode certain proteins in the cells of the heart.

Many research groups, clinicians, medical centres and labs perform experiments on heart cells from patients and animals to better understand the mechanisms that cause these cardiomyopathies. The aim is disease prevention and the development of therapeutic responses.

The data from these experiments are scattered in published papers in different journals and sitting on the hard drives of disparate and disconnected research groups and medical centres.

In order to foster better collaboration and facilitate cardiovascular research, it would be beneficial to have a central repository for the data that can be queried by interested parties anywhere in the world.

## 4 PROJECT SCOPE

The purpose of the web application is to provide a central repository for experimental data on force/tension development and sarcomere length shortening in the cardiomyopathic cells. The application should also provide details relating to the type of cardiomyopathy and present data visually to users when queried. The system should also provide a comfortable user experience.

## 5 SYSTEM PURPOSE

The primary responsibilities of the web application are:

- to enable experimental data to be submitted to the database.
- to respond to queries relating to the stored data on cardiomyopathies.
- to allow the querying of other existing, related databases for additional useful information relating to cardiomyopathies.
- to graph/plot appropriate experimental data for easier consumption and interpretation

Other desired features of the web application:

- a consistent "look and feel" throughout.
- creation, management and storage of user accounts.
- on-line help in website navigation.

## 6 FUNCTIONAL REQUIREMENTS

### 6.1 High Priority

1. The web application shall allow experimental data to be submitted to the database. The data to be stored should include:
  - a. left/right ventricular end diastolic volume (ledv/redv)
  - b. left/right ventricular end systolic volume (lesv/resv)
  - c. left/right ventricular ejection fraction (lvef/rvef)
  - d. left/right ventricular mass (lvmass/rvmass)
  - e. left/right systolic volume (lsv/rsv)
  - f. fibrosis/scarring (scar)
  - g. gender
  - h. age at MRI
  - i. apical HCM
  - j. sudden cardiac death
  - k. hypertension
  - l. diabetes
  - m. myectomy
2. These data/variables should be stored for each of the following HCM gene mutations:
  - a. MYH7
  - b. MYBPC3
  - c. TNNT2
  - d. ACTC
  - e. TPM1
  - f. TNNCI
  - g. TNNI3
  - h. MYL2
  - i. TTN

3. The web application shall respond to queries relating to the stored data on cardiomyopathies including charting the appropriate curve(s).
4. The web application shall have the capability to query other existing, related databases for additional useful information relating to cardiomyopathies.
5. The web application shall provide a guided search facility based on the data currently in the database, e.g., a drop-down box with a list of mutated genes for a specific cardiomyopathy.
  - a. The system shall provide at least the following guided searches:
    - i. Cardiomyopathy type, e.g., hypertrophic, hypertrophic cardiomyopathy.
    - ii. Mutated gene, e.g., MYBPC3.
6. The web application shall query external databases for additional information to provide to user queries. Additional information should include:
  - a. Gene name.
  - b. Gene mutation description (molecular genetics).
  - c. Clinical synopsis of the cardiomyopathy (clinical features).

The system can use any or all of the following external databases and any other appropriate database:

1. [hpo](#): Human Phenotype Ontology
2. [OMIM](#): Online Mendelian Inheritance in Man.
3. [MGI](#).
4. [NCBI SNP Database](#).

## 6.2 Low Priority

1. The web application shall allow the comparison of experimental data between HCM mutations (e.g., I105V in MYH7 vs I105V in MYBPC3) for different cardiomyopathic genetic mutations on a graph/chart.
2. In addition to Hypertrophic Cardiomyopathy, the web application can store experimental data on other cardiomyopathies, e.g., dilated cardiomyopathy, arrhythmogenic right ventricular cardiomyopathy, etc (explore additional genes in the dataset – columns EG to ET).
3. The web application shall provide a news feed showing the latest cardiomyopathy news or development from one or more sources e.g., [News Medical Life Sciences](#).

## 7 NON-FUNCTIONAL REQUIREMENTS

### 7.1 Usability

- A user should be able to use the application within minutes of navigating to the web site.

### 7.2 Security

- A user account should be created, maintained and secured with a password.
- Stored experimental should be secured with the appropriate security rules that provide access control and data validation.

### 7.3 [Accessibility](#) and Supportability

- The web application should work for all people, whatever their hardware, software, location, or ability.
- The web application should support at least the major browsers: Firefox, Chrome, Edge and Safari.

### 7.4 Online user Documentation and Help

- The web application shall provide a web page that explains how to navigate the site.
- This help page should be accessible from all other pages.

## 8 USE CASE DESCRIPTIONS (EXAMPLE CASES)

### 8.1 Login User

Use Case Name	Login User
Summary	A user must login to the system to access data, add or delete data.
Basic Path	<ol style="list-style-type: none"><li>1. The use case starts when a user indicates that he/she wants to login.</li><li>2. The system requests the username and password.</li><li>3. The user enters his/her username and password.</li><li>4. The system verifies the username and password against all registered users.</li><li>5. The system starts a login session and displays a welcome message.</li></ol>
Alternative Paths	Step 4: If username is invalid, the use case goes back to step 2.  Step 4: If the password is invalid, the system requests that the user re-enter the password. When the user enters another password, the use case continues with step 4 using the original username and new password.
Exception Paths	The system may abandon login after 5 unsuccessful attempts.
Preconditions	The user is registered. The username is the user's email address.
Postconditions	The user can now add data, delete data, query data and visualise data.

## 8.2 Reset Password or Username

<b>Use Case Name</b>	<b>Reset Password</b>
<b>Summary</b>	The user needs to reset their password.
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. The use case starts when the user indicates they want to reset their password.</li><li>2. The system requests the username.</li><li>3. The system emails the user a password reset link.</li><li>4. On clicking the link, the system allows the user to set a new password.</li><li>5. The system starts a new login session.</li></ol>
<b>Preconditions</b>	The user is registered. The username is the user's email address.
<b>Postconditions</b>	The user can now login.

## 8.3 Register User

<b>Use Case Name</b>	<b>Register User</b>
<b>Summary</b>	A user must register a username and password to access the system.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The use case starts when a user indicates that he/she wants to register.</li><li>2. The system requests a username (must be an email address) and password.</li><li>3. The user enters a username and password.</li><li>4. The system checks that the username does not duplicate any existing registered usernames.</li><li>5. The system requests a name (*), address, phone number, institutional affiliation (*) and email address (*). Items marked by (*) are required.</li><li>6. The user enters the information.</li><li>7. The system determines the user's location and stores all user information.</li><li>8. The system sends a verification email with a verification link to the user's email address.</li><li>9. User clicks on link in email address.</li><li>10. User is redirected to a login page.</li><li>11. User logs in.</li><li>12. The system starts a login session and displays a welcome message.</li></ol>
<b>Alternative Paths</b>	Step 4: If the username duplicates an existing username, the system displays a message and the use case goes back to step 2.  Step 5: If the user does not enter a required field, a message is displayed and the use case repeats step 4.
<b>Exception Paths</b>	User already exists, in which case he/she can enter their password if known or request a password reset.
<b>Preconditions</b>	The username is the user's email address.
<b>Postconditions</b>	The user can now access the system to add data, delete data, query data and visualise data.

## 8.4 Add Data

Use Case Name	Add Data
Summary	The user wants to add their experimental data to the repository.
Basic Path	<ol style="list-style-type: none"><li>1. The use case starts when a user indicates that he/she wants to add data to the system.</li><li>2. The system presents a dialog for the user to enter details regarding the data.</li><li>3. The system presents a dialog for the user to upload the data.</li><li>4. The user uploads data.</li><li>5. The system checks that the data file is the required format.</li><li>6. The system checks that there are only two columns of data (see preconditions).</li><li>7. The system stores the data.</li><li>8. The system adds and records a timestamp.</li><li>9. The system plots/graphs/charts the data.</li><li>10. The system sends a confirmation email to the user. Email should contain the recorded timestamp and other pertinent information.</li></ol>
Alternative Paths	<p>Step 4: If the file is the wrong format, the system displays a message and the use case repeats step 2.</p> <p>Step 5: If other than two columns of data exist, a message is displayed and the use case repeats step 2.</p>
Exception Paths	The user can cancel the operation at any time.
Preconditions	The data is stored in a csv file or developer-required format. There are only two columns of data, e.g., force and time. [This is also the developer's choice but there should be a verification of the format and the layout of the data].
Postconditions	The data is now in the repository.

## 8.5 Delete Data

Use Case Name	Delete Data
Summary	The user wants to delete their experimental data from the repository.
Basic Path	<ol style="list-style-type: none"><li>1. The use case starts when a user indicates that he/she wants to delete data from the system.</li><li>2. The system presents a dialog for the user to enter details regarding the data.</li><li>3. The system checks that the data exists.</li><li>4. The system authenticates and verifies the user credentials against the data.</li><li>5. The system retrieves the data.</li><li>6. The system asks for final confirmation from the user.</li><li>7. The system deletes the data.</li><li>8. The system records a timestamp.</li><li>9. The system displays a deletion confirmation message and sends a confirmation email to the user.</li></ol>

<b>Alternative Paths</b>	<p>Step 4: If the system is unable to verify the user's credentials against the stored data, the system displays a message and repeats step 2.</p> <p>Step 3: If the data does not exist, the system displays a message and repeats step 2.</p>
<b>Exception Paths</b>	The user can cancel the operation before step 7.
<b>Preconditions</b>	The data exists in the repository.
<b>Postconditions</b>	The data no longer exists in the repository.

## 8.6 Query Data

<b>Use Case Name</b>	<b>Query Data</b>
<b>Summary</b>	The user wants information about some data.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The use case starts when a user indicates that he/she wants information about some data by entering a search term.</li> <li>2. The system retrieves the data.</li> <li>3. The system presents the data including a data plot.</li> </ol>
<b>Alternative Paths</b>	<p>Step 1: The system presents an advanced dialog for the user to enter more specific details about their query (on user indication), e.g., by clicking a button for 'advanced search'.</p> <p>Step 3: If the data does not exist, the system displays a message and repeats step 1.</p>
<b>Exception Paths</b>	The user can cancel the operation at any time.
<b>Preconditions</b>	The data exists in the repository.
<b>Postconditions</b>	The data is displayed to the user.