SWRENG/MECHTRON 3K04

Assignment 2: DCM Documentation

Submitted: December 3, 2023

Completed by: Group 5

Table of Contents

DCM Platform

Pacemakers play an important role in ensuring the health of many people. However, in order to ensure the functionality of a pacemaker and customize its settings, a user interface is necessary. The Device Controller Monitor (DCM), our graphical user interface (GUI), is used as an intermediate between doctors and their patients' pacemakers. Through this software, doctors should be able to access current pacing records in VVI, VOO, AOO, and AAI modes as well as change settings including pulse rate, pulse width, pulse amplitude, sensitivity of the sensing circuit, pacing mode, refractory period, etc. The system will store up to 10 users and require them to sign in before accessing personal records. A full list of requirements and documentation will be further explained and outlined throughout this document.

For our DCM, our team decided to use Python and the pre-installed Tkinter package. One of the reasons our team decided to use Python was because of our collective familiarity with the language. Tkinter also supports all the necessary functionality for us to implement our GUI. Additionally, it is user-friendly; Tkinter's accessibility meant we were able to focus on creating the best possible implementation of all our required features. Python's stable serial communication library will also be critical in Assignment 2 when the DCM will be communicating with the pacemaker.

The program structure will follow the design flow pictured below (Figure 1). This flow was determined to be optimal as it naturally modularizes our design by separating each function into separate sections. These sections were then used to create separate files that are independent of each other (other than the global variables stated) so that in the event of a program error, issues can be easily and quickly diagnosed. This flow will be further outlined in the requirements and window definitions.

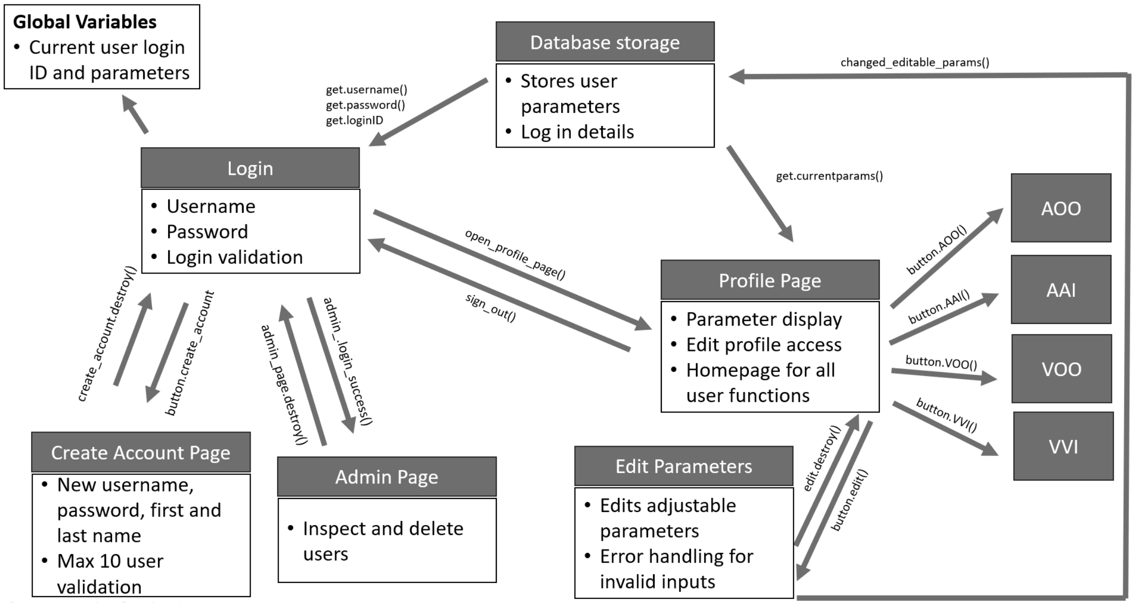


Figure 1 - Program Design Flow

Running the Program

To run the DCM, open **main.py()** and run the file. The login window will appear.

Assignment 2 Changes

In Assignment 1, we created a solid foundation for the functionality of the DCM. Users were able to log in, create accounts, and edit their personalised parameters. In Assignment 2, our goal was to implement serial communication with our Simulink model. Changes made on the DCM were to be sent using serial communication to the Simulink model to be implemented in the pacemaker. Therefore, our system closely models the actual device. We also needed the DCM to display real-time egram data using this serial communication link, allowing the user to view signals from the ventricle, the atrium, or both. Before implementing this, we first made some structural amendments to our code to make it more modularized. To do so, we created two classes: **pages** and **database**. The **pages** class contains a method for every window opened in the DCM. This makes it simple to declare an instance of this class and run the “start” method, which prompts the login window and all subsequent attributes of the DCM. The class **database** handles storing and modifying all editable user parameters. Methods from this class are used in **pages** and are integral to the functionality of our program. Generally, the use of object-oriented programming makes our code more modular and replicable. It also means errors are more isolated and easier to diagnose.

As we improved our code from Assignment 1, there were some changes made to the GUI. With these changes come changes in our documentation. For clarity, all text from Assignment 1 documentation will be greyed out. As some elements of the DCM were deemed no longer necessary as we proceeded with the project, they have been removed. Any element with a strikethrough is no longer part of the DCM, but will still be represented in the documentation to show our thought process and planning throughout Assignment 2.

Aside from these changes, our program is structured similarly to how it was in Assignment 1. These changes are represented in Figure 2, as seen below.

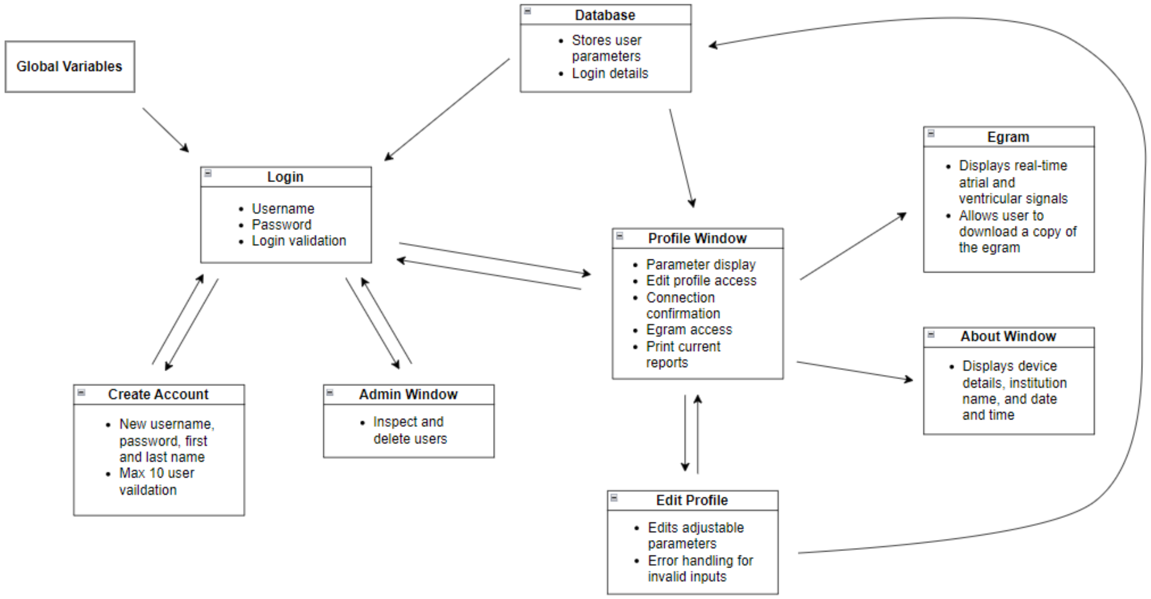


Figure 2 - Program Design Flow (updated for Assignment 2)

Requirements

Below are the DCM system requirements. These are specified by the client (3K04 project outline) and by the team. *Italics* are requirements that were implemented based on the team's discretion.

Element	Element Specifications	Notes
General User Interface (Section 1)	Window management (Section 1.1)	*-Login window will close after profile is opened* *-Login window will reappear once program is exited* *-New windows will have go back function, that brings you to previous window*

	Text and graphics (Section 1.2)	*-Responsive windows–widgets move when window is resized* *-Cohesive colour scheme*
	Accessibility (Section 1.3)	*-Colour scheme should accommodate for visual impairment* *-Buttons should be standardised*
Login screen (Section 2)	Ability to register new user (name and password) (Section 2.1)	-Button to “sign up”, prompts create account screen
	Login as existing user (10 users maximum) (Section 2.2)	-Program checks input vs database, if there is a match the user may proceed *-If wrong login credentials, display error message* *-Password should be hidden (letters appear as asterisks)*
Create account screen (Section 3)	Create new user (10 users maximum) (Section 3.1)	-User is to fill out name and password *-User to create a username* *-User to confirm password, if passwords don’t match display “passwords do not match” pop-up* -Create an entry in an existing database that stores pre-existing users *-redirect to login page if account already exists* -Do not allow account creation if 10 users already exist
	Create account security (Section 3.2)	*-Error message if passwords do not match, if password is too short, or if username is too short* *-’Password’ and ‘confirm password’ should be hidden and appear as asterisks*

Profile screen (Section 4)	Profile general display (Section 4.1)	*-Display first name and last name* *-Pacemaker version/serial number* *-Date of implant* -If pacemaker is connected/not connected -If pacemaker matches logged in user; if no match, display “new pacemaker has been approached” prompt
	Stored information (Section 4.2)	*-Window will have “Edit Profile” button that opens edit profile page* -Will connect to windows showing VVI, VOO, AAI, and AOO displays -Will display Limit, Upper Rate Limit, Atrial Amplitude, Atrial Pulse Width, Ventricular Amplitude, and Ventricular Pulse Width, VRP, and ARP
Edit profile (Section 5)	Changeable information (Section 5.1)	-Will allow changes to all parameters displayed in the profile screen
	Programming safety (Section 5.2)	*-Each change will raise errors if not within the specified bounds or within the specified increments*
~~Pacing mode interfaces (Section 6)~~	~~VOO display~~ (Section 6.1)	-Window will act as a placeholder for VOO plot display
	~~VVI display~~ (Section 6.2)	-Window will act as a placeholder for VVI plot display
	~~AOO display~~ (Section 6.3)	-Window will act as a placeholder for AOO plot display

	~~AAI display~~ (Section 6.4)	-Window will act as a placeholder for AAI plot display
Administrator Window (Section 8)	Edit admin password (Section 7.1)	*-Window will allow the administrator to change their password*
	Show existing users (Section 7.2)	*-A button press will display a list of the existing users and their corresponding ID numbers*
	Delete existing users (Section 7.2)	*-The administrator will enter the ID of the user they would like to delete into the entry field* *-Pressing a button will delete the user from the database*
Egram Display (Section 9)	Display different signals (Section 9.1)	*- The user has the option to display atrial, ventricle, or both signals from egram data using three buttons* *- Pressing a button will graph the egram data retrieved from serial communication*
	Egram Report Available (Section 9.2)	*- Pressing the print report button will allow the user to have a PDF of the current egram signal along with other information*
Temporary Parameters Report (Section 10)	Temporary Parameters Report Available (Section 10.1)	*- Present the reports button on the profile window will download a report on the users computer of all the current programable parameters*
About Window (Section 11)	Display Current Information (Section 11)	*-Window will display Model Number, Software Version, Serial Number, Institution, and Date/Time for the DCM*

Pacemaker Connection (Section 12)	Display Pacemaker Connection (Section 11)	*-Function will test the connection to the pacemaker and inform the user if whether or not the pacemaker is connected*

Table 1 - Requirements

Design Goals

Our team's goal for our DCM design was to keep it consistent, simple, and easy to follow, allowing for an optimal viewing experience no matter the user's visual ability. This was satisfied using a blue background with grey buttons and black text. These are non-complementary colours that have high contrast allowing for maximum distinction. All windows keep this colour scheme as it is consistent and supports the requirements outlined in Sections 1.2 and 1.3. Buttons were also standardized between windows to meet the requirements outlined in Section 1.3. The team felt that consistent button visuals and colour schemes were important since once a patient/doctor becomes familiar with the GUI visuals on one window they can immediately identify similar features on every other window in the system.

Expected Changes to Requirements and Design

The first submission of this documentation marks the halfway point of the project. Moving forward to our second assignment, the DCM will need to interact with the pacemaker. This means that the foundations we have laid through both the DCM and the Simulink model will be implemented and paired with the pacemaker. In Assignment 2, a major change to the DCM will be that the VOO, VVI, AOO, and AAI pages will be required to display the egram data from the pacemaker. All other interactions with the pacemaker, previously implemented as placeholders, will need to be functional as well. This includes the connectivity status, the date of implant, and the version number of the pacemaker. The DCM will also expand to include AOOR, VOOR, AAIR, and VVIR modes. Additionally, we will generate an assurance system for our platform. This will change the profile page as more modes will be added. The database will also need to expand to accommodate these new parameters.

- List all requirements changes that are likely
    - Section 3.1 (Create new user - 10 users maximum)
        - Might have more users in the future
    - Section 2.2 - Login as existing user (10 users maximum)
        - Might have more users in the future
    - (Section 4.1) - Might add new parameters in the future
- List all design decisions that are likely to change
    - Egram: might add three subplot, instead of clicking on buttons to see the three different graphs

    - Scrolling for window, different laptops displays might struggle with version, allows for resizing of window without cutting out info
    -

Following Assignment 2, there are still changes we expect to make to our program. Despite its current state of completion, there are some changes that our design will likely have to endure in the future that are worth considering. To expand the applications of our program we would likely have to increase the capacity of our program to accommodate more users as currently our database only supports a maximum of 10 users. Future models of this program should be expanded to accommodate more users and be able to function and save their user data for continuous use. Future implementations of our program may also improve upon the electrogram display of our system by introducing several subplots together rather than having several menus to click through for ease of viewing. Our program would also continue to be designed to better adapt its usage on a variety of devices, adjusting window size and the ability to scroll through windows that are too large to be properly displayed on the user's device without cutting off info.

As time goes on software will continue to be developed for newer features and to be implemented with new and upcoming hardware. As this software continues to get used, it may begin to accumulate obsolete code, call upon dependencies that no longer work, and develop undetected bugs along the processes while integrating into newer environments. To maintain the long-term functionality and security these points must be reviewed and verified to be kept up-to-date and functional. Alongside software environment changes, our software may be challenged to adapt to the rapidly changing health situations of our clients. As the health situation and needs of our clients change with time, the parameters and features of our system may be required to continuously update to properly service them. Our system may also come across the challenge of competing in a market filled with developing technology. In the future, our system may eventually become obsolete with the rise in newer products that provide simpler or better solutions to our current methods. Our system would have to continue to update its design and technology to compete within the market.

Function Definitions

Login Window

The login page is the first screen that appears after running the program. The user is greeted with labelled entry fields to enter their username and password, as well as buttons to either log in or create a new account. Our team chose to implement a username for the login window instead of using the user’s name. This streamlines the login process and makes our program more secure. This window also creates and stores the program's global variables, which will later be used to display the user’s name on the Profile Screen and their information on the Edit Profile Screen. These global variables include:

- login_ID - stores the ID number of the logged-in user
- login_name - stores the name of the logged-in user
- login_LRL - stores the current lower rate limit of the logged-in user
- login_URL - stores the current upper rate limit of the logged-in user
- login_MSL - stores the current max sensor limit of the logged-in user
- login_AA - stores the current atrial amplitude of the logged-in user
- login_VA - stores the current ventricular amplitude of the logged-in user
- login_APW - stores the current atrial pulse width of the logged-in user
- login_VPW - stores the current ventricular pulse width of the logged-in user
- login_VRP - stores the current VRP of the logged-in user
- login_ARP - stores the current ARP of the logged-in user
- login_PVARP - stores the current PVARP of the logged-in user
- login_H - stores the current hysteresis rate limit of the logged-in user
- login_RS - stores the current rate smoothing of the logged-in user
- login_AT - stores the current activity threshold of the logged-in user
- login_RT - stores the current reaction time of the logged-in user
- login_RF - stores the current response factor of the logged-in user
- login_recT - stores the current recovery time of the logged-in user
- login_M - stores the current mode of the logged-in user

This window satisfies the requirements outlined in Section 2.

Title	Type	Function	Description
Username input	Input	username_entry	User’s username
Password input	Input	password_entry	User’s password
login_button	Button	login	Opens profile window
new_acct_button	Button	create_acct	Opens create account window

Table 2 - Login Window Features

Create Account Window

This screen is prompted by the “Create New Account” button on the login window. This window has labelled entry fields for the user’s first name, last name, username, password, and password confirmation. The user must input a valid text entry for all fields. The username must be at least four characters, the password must be at least eight characters, and the inputs in the “Password” and “Confirm Password” entry fields must match. Once these requirements are satisfied, pressing the “Create Account” button will input the information to the database and reopen the login window, prompting the user to log in to their new account. However, if 10 users exist in the database, account creation will be denied. The administrator must log in to delete an account before another is created. This satisfies Section 3 of our requirements, ensuring that security and user creation features are fully functional.

Title	**Type**	**Function**	**Description**
First name input	Input	firstname_entry	User’s first name
Last name input	Input	lastname_entry	User’s last name
Username input	Input	new_username_entry	User’s requested username
Password input	Input	new_password_input	User’s requested password
Confirm password input		confirm_password_input	Confirming correct and intended password
create_acct_button	Button	submit(username, password, firstName, lastName)	Adds account to the database with a first and last name, username and password input
back_button	Button	create_window.destroy	Destroys create account window and reopens login window

Table 3 - Create Account Feature

Administrator Window

To view and delete users in the database, the administrator must log in using a designated set of credentials. Only this username and password will prompt the admin screen. The administrator can then click a button to show the existing usernames and corresponding IDs (Section 8.2). By inputting an ID into the entry field and clicking “Delete User,” the user connected to that ID will be removed from the database (Section 8.3). This screen also features a “Change Password” button where the administrator can change their password, fulfilling Section 8.1 of our requirements. This is the only way to change the administrator credentials. The “Sign Out” button will destroy the administrator window.

Title	Type	Function	Description
ID input	Input	delete_user_entry	ID of user administrator wants to delete
delete_user_button	Button	delete(idNum)	Deletes a user based on their ID number
show_users_button	Button	query()	Shows all usernames and IDs of users
Admin password	Input	password_entry	Shows current admin password and allows administrator to edit
change_password_button	Button	changePassword(admin_id,password_entry.get())	Saves new password
signout_button	Button	admin_window.destroy()	Destroys admin window

Table 4 - Administrator Features

Profile Window

This window displays the logged-in user's current settings. Users see the current Upper Rate Limit, Lower Rate Limit, Atrial Amplitude, Atrial Pulse Width, Ventricular Amplitude, Ventricular Pulse Width, VRP, ARP, and current pacing mode. These values are called from the global variables defined in the login window. The program matches the user's login credentials to the information in the database and displays the relevant parameters. These values are also refreshed after updates are made in the Edit Profile Window. Users also see which pacemaker version they have, the date of installation, and the current connection status. Users can access the pacing pages for VVI, VOO, AOO, and AAI settings. Finally, two buttons on the lower part of the screen allow the user to either sign out or access the Edit Profile Window to edit the current settings. These implementations satisfy Section 4 of the requirements. In this window, all fonts, button sizes/layout, and colours satisfy those outlined in Section 1.

For Assignment 2, the following modifications were made to the profile window. The parameters displayed now also include the maximum sensor limit, atrial sensitivity, ventricular sensitivity, PVARP, hysteresis rate limit, rate smoothing, activity threshold, reaction time, response factor, and recovery time. The mode buttons were removed from the profile window and were replaced by a single egram display button, which displays the egram data for the currently selected mode. The profile window now also contains a button that allows users to print a temporary report containing the current parameters. It also features an about function that displays information about the current DCM model, and a check connection button that checks whether the DCM is connected to the pacemaker.

Title	Type	Function	Description
~~AAI~~	Button	open_AOO	Opens AAI page
~~AOO~~	Button	open_AAI	Opens AOO page
~~VVI~~	Button	open_VOO	Opens VVI page
~~VOO~~	Button	open_VVI	Opens VOO page
Profile Edit	Button	reopen()	Opens edit page window and refreshes profile page once updated
Sign Out	Button	profile.destroy	Signs user out of account and returns to login page
Egram	Button	egram_win	Opens window that allows users to view egram data
Reports	Button	temp_report	Prints temporary report containing the current user parameters
About	Button	open_about	Opens window that displays date/time as well as version information about the DCM
Check Connection	Button	button_connection_check	Checks the connection to the pacemaker and updates the connection status
Connection* Message	Message	connection.message()	Displays current connection status, pacemaker version, and date of implant
User parameters	Message	parameter.get()	Displays the current user's editable parameters

Table 5 - Profile Features

~~VOO Window~~

This window currently displays a blank screen. It is intended to display all VOO plots as well as store any previous records displayed on the VOO screen. Currently, this screen only displays a back button as other functionalities are not required for this assignment. This screen will have more content added in Assignment 2. It is one of the key design changes that are expected to occur in our system after Assignment 1.

Title	Type	Function	Description
back_button	Button	VOO.destroy	Closes current VOO page

Table 6 - VOO Features

~~AOO Window~~

This window currently displays a blank screen. It is intended to display all AOO plots as well as store any previous records displayed on the AOO screen. Currently, this screen only displays a back button as other functionalities are not required for this specific assignment. This screen will have more content added in Assignment 2. It is one of the key design changes that are expected to occur in our system after Assignment 1.

Title	Type	Function	Description
back_button	Button	AOO.destroy	Closes current AOO page

Table 7 - AOO Features

~~VVI Window~~

This window currently displays a blank screen. It is intended to display all VVI plots as well as store any previous records displayed on the VVI screen. Currently, this screen only displays a back button as other functionalities are not required for this assignment. This screen will have more content added in Assignment 2. It is one of the key design changes that are expected to occur in our system after Assignment 1.

Title	Type	Function	Description
back_button	Button	VVI.destroy	Closes current VVI page

Table 8 - VVI Features

~~AAI Window~~

This window currently displays a blank screen. It is intended to display all AAI plots as well as store any previous records displayed on the AAI screen. Currently, this screen only displays a back button as other functionalities are not required for this assignment. This screen will have more content added in Assignment 2. It is one of the key design changes that are expected to occur in our system after Assignment 1.

Title	Type	Function	Description
back_button	Button	AAI.destroy	Closes current AAI page

Table 9 - AAI Features

About Window

This window is opened from the open profile page and displays the date/time, along with the Model Number, Software Version, Serial Number, and Institution for the DCM.

Edit Profile Window

This window displays the edit screen to the user. Here the user can edit and change any of their pacemaker settings according to the Editable Parameters section (Table 11). Users can increment inputs using the up and down buttons provided beside each input or by typing in the requested input. Users will not be able to input an invalid parameter for any setting but will instead be faced with an error prompt. From this window, users can also change their username, password, first name, and last name. When a user is satisfied with their setting changes, they can click the “Save Changes” button that updates the parameters in the database. For Atrial and Ventricular Pulse Amplitude and Pulse Width, we decided to represent the “off” value as a 0. This makes it easier to store the information in our program.

For Assignment 2, the Edit Profile Window was modified to account for additional parameters required by the new pacing modes we implemented. These new parameters included maximum sensor limit, PVARP, hysteresis rate limit, rate smoothing, activity threshold, reaction time, response factor, and recovery time. Similar to before, for PVARP, hysteresis rate limit, and rate smoothing, “off” is stored as 0 to make storing the value simpler. Some of these parameters are only applicable to the rate adaptive modes. A design decision was made to still display these parameters when the user is not in a rate adaptive mode. However, if they are edited by the doctor, no information will be sent from these parameters to the pacemaker. The intention behind this was to allow the doctor to have the parameters stored and easily accessible if they wanted to transition the user into a rate adaptive pacing mode.

Title	Type	Function	Description
back_btn	Button	editor.destroy	Closes current edit profile page
save_btn	Button	update(welcome_screen.login_id)	Saves valid inputs and prompts errors for invalid inputs
First Name Change	Input	firstName_edit	Changes users front name
Last Name Change	Input	lastName_edit	Changes users last name
Username Change	Input	username_edit	Changes user username

Password Change	Input	password_edit	Changes user password
Lower Rate Limit Change	Input	lowerRateLimit_edit	Changes user lower rate limit
Upper Rate Limit Change	Input	upperRateLimit_edit	Changes user upper rate limit
Atrial Amplitude Change	Input	atrialAmplitude_edit	Changes user atrial amplitude
Atrial Pulse Width Change	Input	atrialPulseWidth_edit	Changes user atrial pulse width
Ventricular Amplitude Change	Input	ventricularAmplitude_edit	Changes user ventricular amplitude
Ventricular Pulse Width Change	Input	ventricularPulseWidth_edit	Changes user pulse width amplitude
VRP Change	Input	VRP_edit	Changes user VRP
ARP Change	Input	ARP_edit	Changes user ARP
Mode Change	Input	mode_edit	Changes user mode
Max Sensor Limit Change	Input	maxSensLimit_edit	Changes user max sensor limit
PVARP Change	Input	PVARP_edit	Changes user PVARP
Hysteresis Change	Input	hysteresis_edit	Changes user hysteresis
Rate Smoothing Change	Input	rateSmoothing_edit	Changes user rate smoothing
Activity Threshold Change	Input	activityThreshold_edit	Changes user activity threshold
Reaction Time Change	Input	reactionTime_edit	Changes user reaction time
Response Factor Change	Input	responseFactor_edit	Changes user response factor
Recovery Time Change	Input	recoveryTime_edit	Changes user recovery time

Table 10 - Database Features

Database

This table shows all the internal functions that are called throughout the GUI system. These are not accessed by the user.

Title	Type	Function	Description
Access database entries	Function	query()	returns all of the entries in the database as a list, with each space in the list containing a list for each entry
Delete user	Function	delete(record_id)	Given a user ID, deletes the user from the database
Edit user parameters	Function	edit(record_id)	Main display for edit profile screen. Calls update(user_ID) when the save button is clicked.
Update parameters	Function	update(record_id)	Takes inputs from parameter_edit and updates the user database
Input new entries	Function	submit(username, password, firstname, lastname)	Takes inputs and creates a new account with default input settings

Table 11 - Database functions

For Assignment 2, more parameters were added to the database, meaning each of the functions was modified to handle the new parameters. These modifications consist mainly of changes to the indexes of parameters stored without changing the input or outputs of the functions.

Egram Display Window

This interface showcases four buttons for the user. By selecting a specific button, the user can effortlessly achieve their desired outcome. Everything in this window is related to the egram data sent via serial communication. The table below will elaborate on the functionality of the egram display window. These implementations satisfy Section 9 of the requirements.

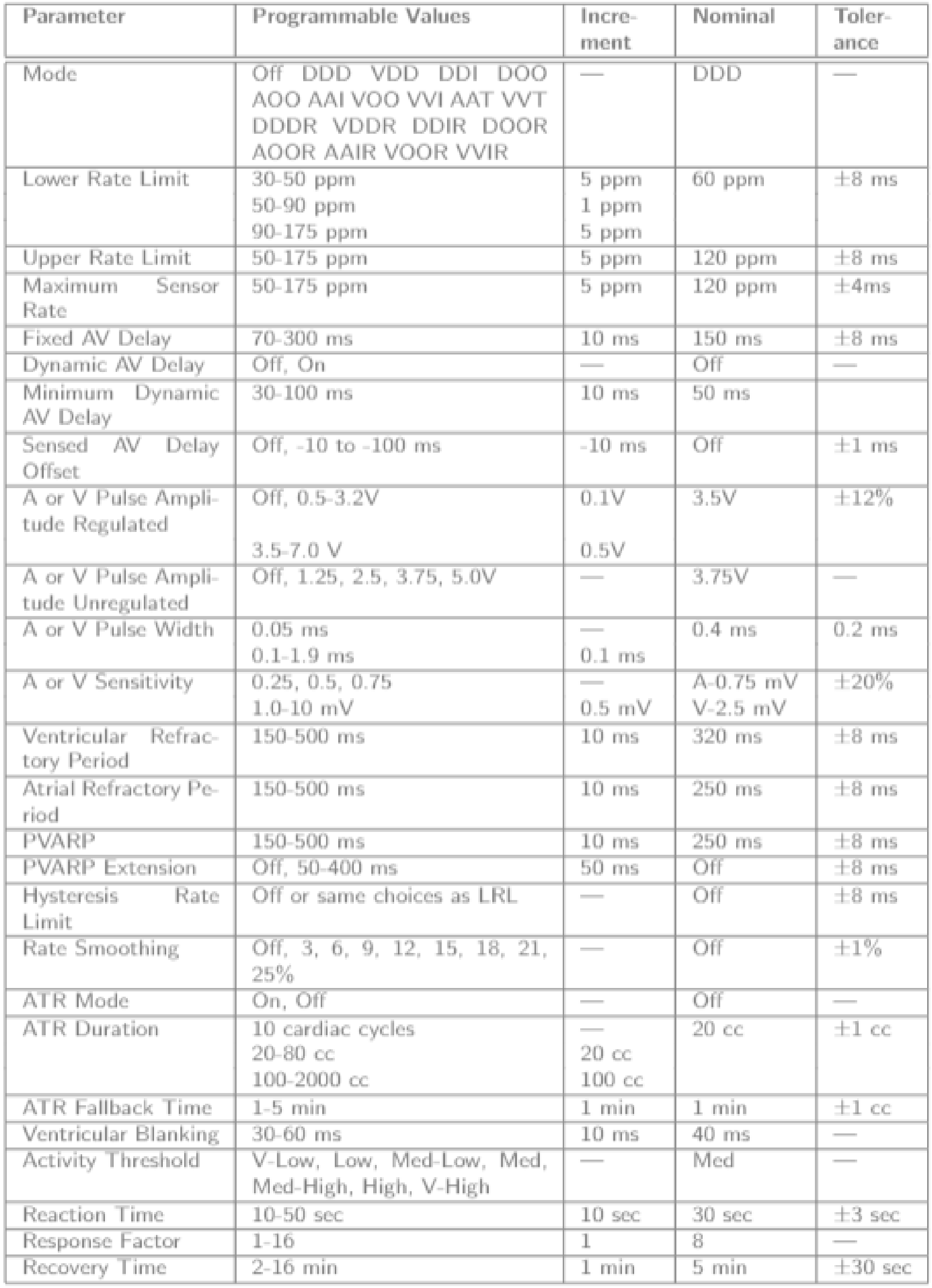
Title	Type	Function	Description
AS_btn	Button	egram_test.graph1(1)	By passing a one through the graph1 function, the atrial egram data from the pacemaker is displayed
VS_btn	Button	egram_test.graph1(2)	By passing a two through the graph1 function, the ventricle egram data from the pacemaker is displayed
BS_btn	Button	egram_test.graph1(3)	By passing a three through the graph1 function, the atrial and ventricle egram data from the pacemaker is displayed
egram_picb	Button	brad_report	Outputs the bradycardia report of the graph that is currently displayed along with the following information: Institution name, date and time of report printing, device model, DCM serial number, application model, and report name

Table 12 - Database functions

Design decisions that were made within the egram window ensures that the functionality of the window is intuitive. The three buttons used to display the different signals allow the user to select which signal they need. The graph displaying the signal is interactive. Buttons allow the user to zoom in, zoom out, reset, shift the graph to the left, shift it to the right, shift it to the top, shift it to the bottom, drag the graph and save it, allowing the user to cater the graph to their needs. A fourth button allows the user to download a PDF of the current graph they selected.

Editable Parameters

All editable parameters are stored in the database file. These values are not accessible unless signed in under a specific user. Parameters are to follow the following standard outlined in the Pacemaker document:



Parameter	Programmable Values	Increment	Nominal	Tolerance
Mode	Off DDD VDD DDI DOO AOO AAI VOO VVI AAT VVT DDDR VDDR DDIR DOOR AOOR AAIR VOOR VVIR	—	DDD	—
Lower Rate Limit	30-50 ppm 50-90 ppm 90-175 ppm	5 ppm 1 ppm 5 ppm	60 ppm	±8 ms
Upper Rate Limit	50-175 ppm	5 ppm	120 ppm	±8 ms
Maximum Sensor Rate	50-175 ppm	5 ppm	120 ppm	±4ms
Fixed AV Delay	70-300 ms	10 ms	150 ms	±8 ms
Dynamic AV Delay	Off, On	—	Off	—
Minimum Dynamic AV Delay	30-100 ms	10 ms	50 ms	
Sensed AV Delay Offset	Off, -10 to -100 ms	-10 ms	Off	±1 ms
A or V Pulse Amplitude Regulated	Off, 0.5-3.2V 3.5-7.0 V	0.1V 0.5V	3.5V	±12%
A or V Pulse Amplitude Unregulated	Off, 1.25, 2.5, 3.75, 5.0V	—	3.75V	—
A or V Pulse Width	0.05 ms 0.1-1.9 ms	— 0.1 ms	0.4 ms	0.2 ms
A or V Sensitivity	0.25, 0.5, 0.75 1.0-10 mV	— 0.5 mV	A-0.75 mV V-2.5 mV	±20%
Ventricular Refractory Period	150-500 ms	10 ms	320 ms	±8 ms
Atrial Refractory Period	150-500 ms	10 ms	250 ms	±8 ms
PVARP	150-500 ms	10 ms	250 ms	±8 ms
PVARP Extension	Off, 50-400 ms	50 ms	Off	±8 ms
Hysteresis Rate Limit	Off or same choices as LRL	—	Off	±8 ms
Rate Smoothing	Off, 3, 6, 9, 12, 15, 18, 21, 25%	—	Off	±1%
ATR Mode	On, Off	—	Off	—
ATR Duration	10 cardiac cycles 20-80 cc 100-2000 cc	— 20 cc 100 cc	20 cc	±1 cc
ATR Fallback Time	1-5 min	1 min	1 min	±1 cc
Ventricular Blanking	30-60 ms	10 ms	40 ms	—
Activity Threshold	V-Low, Low, Med-Low, Med, Med-High, High, V-High	—	Med	—
Reaction Time	10-50 sec	10 sec	30 sec	±3 sec
Response Factor	1-16	1	8	—
Recovery Time	2-16 min	1 min	5 min	±30 sec

Table 13 - Editable Parameters

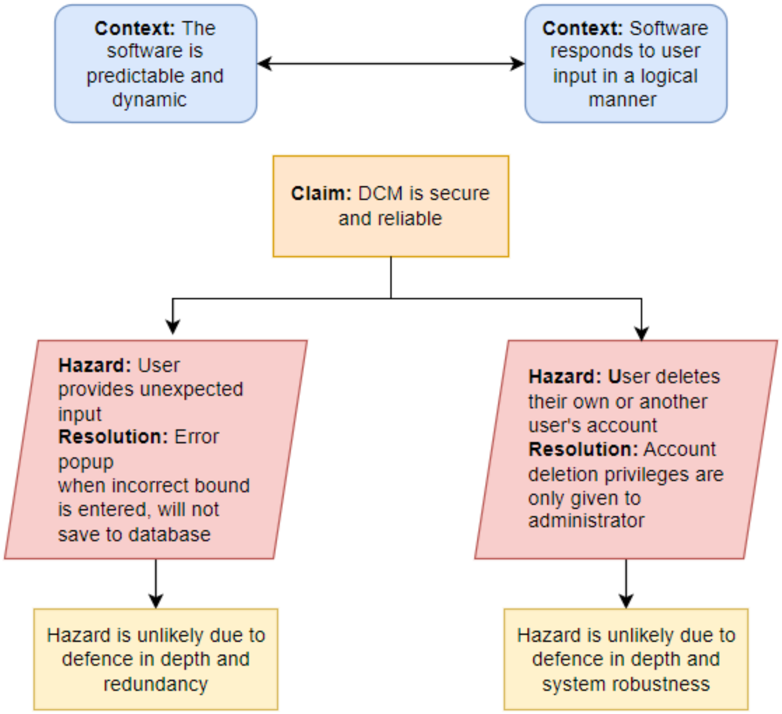
Note: Not all programmable parameters were required for Assignment 1 however, the GUI database has been adjusted to accept all values once they are required. This is outlined more in the sections above.

Serial Communication

A crucial part of this assignment was ensuring that our DCM and Simulink model communicated serially. The two common serial standards are I2C and UART. For this project, UART was chosen as the serial communication method to share information via USB. Sending parameters to the pacemaker requires using the pySerial library. The parameters sent to the pacemaker included the programmable parameters needed for AOO, AAI, VOO, VVI, AOOR, VOOR, AAIR, and VVIR modes. This also includes a transmit bit which when set to one will allow for the information to be transmitted to the pacemaker. All the information sent to the pacemaker is packed for easier transmission using the struct library.

When receiving information from the pacemaker, the information is unpacked first. The information sent to the pacemaker is in the following order: mode, atrial amplitude, ventricle amplitude, ventricle signal, and atrial signal. The signals are then used to display egram data.

Assurance Cases



Context: The software is predictable and dynamic
Context: Software responds to user input in a logical manner
Claim: DCM is secure and reliable
Hazard: User provides unexpected input
Resolution: Error popup when incorrect bound is entered, will not save to database
Hazard: User deletes their own or another user's account
Resolution: Account deletion privileges are only given to administrator
Hazard is unlikely due to defence in depth and redundancy
Hazard is unlikely due to defence in depth and system robustness

Figure 3 - DCM Assurance Cases

Assurance cases ensure the reliability, safety, and security of the DCM software. There are two assurance considerations which include tackling two hazards: unexpected input from the user, and the deletion of user accounts. Regarding the first hazard, an admin account is put in place to only allow authorized personnel to have access to modify all user accounts. With the second hazard, assurance cases verify that programable parameters adhere to defined ranges and constraints If the inputted parameters go against the assurance cases, the input will be rejected and the software will prompt the user for a proper input. Parameters must be consistent with the intended functionality of the pacemaker and should not cause unintended consequences.

Testing

Login Window

The purpose of these test cases is to ensure that the Login Window is fully functional. This includes being able to create a new account and log in when the correct username and password are entered.

Test	System Input	Expected Result	Outcome
User Username and Password	Username: Correct Password: Correct	Login Successful	Pass
	Username: Correct Password: False	Invalid login credentials pop up	Pass
User Username and Password	Username: False Password: Correct	Invalid login credentials pop up	Pass
	Username: False Password: False	Invalid login credentials pop up	Pass
Admin Username and Password	Username: Correct Password: False	Invalid login credentials pop up	Pass
	Username: Correct Password: Correct	Opens admin window	Pass
Create Account	Click of create	Opens create	Pass
Button	Account button	account window	Pass
Login Button	Click of login button	Opens profile page	Pass

Table 14 - Login Window Tests

Create Account Window

The purpose of these test cases is to ensure that the Create Account Window is fully functional. This includes being able to create a new account given that the username is not already taken and both the password and confirm password fields have matching entries greater than 8 characters.

Test	System Input	Expected Result	Outcome
Password and confirm password not matching	Passwords not same	Passwords do not match prompt	Pass
Password too short	Less than 8 characters	Invalid password prompt	Pass
Empty username field	Username is NULL	Invalid username prompt	Pass
Empty first name field	First name is NULL	Invalid first name prompt	Pass
Empty last name field	Last name is NULL	Invalid last name prompt	Pass
Existing username	Username exists in database	Username already exists prompt	Pass
10 users in database	Max amount of users stored in database	Too many existing users prompt	Pass

Table 15 - Create Account Window Tests

Admin Window

The purpose of these test cases is to ensure that the admin window is fully functional. This includes being able to delete accounts, change the admin password, and show all usernames.

Test	Case	Expected Result	Outcome
Show usernames	Click “show usernames”	Shows all usernames	Pass
Delete user	Input user to delete	Deletes suggested user	Pass
Change admin password	Input new admin password	New admin password, old one does not work	Pass

Table 16 - Admin Window Tests

Profile Window

The purpose of these test cases is to ensure that the profile window is fully functional. This includes being able to sign out, open the Edit Profile Window, open the egram window, download a temporary parameters report, open the about window, and check the pacemaker connection.

Test	System Input	Expected Result	Outcome
Open Edit Profile	Clicks “edit profile” button	Opens edit profile window	Pass
Signs user out	Clicks “sign out” button	Opens sign out window	Pass
Updated settings after change	Increase all editable parameters by one increment	Refreshes profile window and updates changed values	Pass
~~Opens VOO page~~	Click “VOO” button	Opens VOO window	Pass
~~Opens VVI page~~	Click “VVI” button	Opens VVI window	Pass
~~Opens AOO page~~	Click “AOO” button	Opens AOO window	Pass
~~Opens AAI page~~	Click “AAI” button	Opens AAI window	Pass
Opens egram window	Click “Egram” button	Opens egram window	Pass
Download temporary parameter report	Click “Reports” button	Downloads report to user’s files	Pass
Opens about window	Click “About” button	Opens about window	Pass
Checks pacemaker connection	Click “Check Connection” button	Message box shows if a pacemaker is connected	Pass

Table 17 - Profile Window Tests

Edit Profile Window

The purpose of these test cases is to ensure that the edit profile window is fully functional. This includes being able to change the user's username, password, and all programmable parameters while ensuring that these changes adhere to requirements.

Test	Case	Expected Result	Outcome
Change username	Valid username	Save data	Pass
	NULL entry	Fail due to invalid input	Pass
Change password	Password < 8 characters	Fail due to invalid input	Pass
	Password >= 8 characters	Save data	Pass
Change first name	Valid first name	Save data	Pass
	NULL entry	Fail due to invalid input	Pass
Change last name	Valid last name	Save data	Pass
	NULL entry	Fail due to invalid input	Pass
Change lower rate limit	176	Fail due to upper bound	Pass
	29	Fail due to lower bound	Pass
	34	Fail due to increment	Pass
	30	Save data	Pass
	50.4	Fail due to non-whole number	Pass
	51	Save data	Pass
	90	Save data	Pass
	101	Fail due to	Pass

		increment	
	175	Save data	Pass
	hello	Fail due to invalid data type	
Change upper rate limit	50	Save data	Pass
	175	Save data	Pass
	100	Save data	Pass
	174	Fail due to increment	Pass
	185	Fail due to upper bound	Pass
	25	Fail due to lower bound	Pass
	hello	Fail due to invalid data type	Pass
Change atrial amplitude	0	Save data	Pass
	0.5	Save data	Pass
	0.7	Save data	Pass
	0.65	Fail due to increment	Pass
	0.4	Fail due to lower bound	Pass
	3.4	Fail due to bounds	Pass
	3.2	Save data	Pass
	3.5	Save data	Pass
	5.0	Save data	Pass
	7.0	Save data	Pass
	6.4	Fail due to increment	Pass

	8.0	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
Change ventricular amplitude	0	Save data	Pass
	0.5	Save data	Pass
	0.7	Save data	Pass
	0.65	Fail due to increment	Pass
	0.4	Fail due to lower bound	Pass
	3.4	Fail due to bounds	Pass
	3.2	Save data	Pass
	3.5	Save data	Pass
	5.0	Save data	Pass
	7.0	Save data	Pass
	6.4	Fail due to increment	Pass
	8.0	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
Change atrial pulse width	0.05	Save data	Pass
	0.07	Fail due to lower bound	Pass
	0.1	Save data	Pass
	1.0	Save data	Pass
	1.9	Save data	Pass
	2.0	Fail due to upper	Pass

		bound	
	1.05	Fail due to increment	Pass
	hello	Fail due to invalid data type	Pass
Change ventricular pulse width	0.05	Save data	Pass
	0.07	Fail due to lower bound	Pass
	0.1	Save data	Pass
	1.0	Save data	Pass
	1.9	Save data	Pass
	2.0	Fail due to upper bound	Pass
	1.05	Fail due to increment	Pass
	hello	Fail due to invalid data type	Pass
Change VRP	140	Fail due to lower bound	Pass
	150	Save data	Pass
	300	Save data	Pass
	500	Save data	Pass
	510	Fail due to upper bound	Pass
	205	Fail due to increment	Pass
	hello	Fail due to invalid data type	Pass
Change ARP	140	Fail due to lower bound	Pass

	150	Save data	Pass
	300	Save data	Pass
	500	Save data	Pass
	510	Fail due to upper bound	Pass
	205	Fail due to increment	Pass
	hello	Fail due to invalid data type	Pass
Change mode	AOO	Save data	Pass
	AAI	Save data	Pass
	VOO	Save data	Pass
	VVI	Save data	Pass
	hello	Fail due to invalid input	Pass
	AOOR	Save data	Pass
	AAIR	Save data	Pass
	VOOR	Save data	Pass
	VVIR	Save data	Pass
Change max sensor limit	40	Fail due to lower bound	Pass
	50	Save data	Pass
	100	Save data	Pass
	123	Fail due to increment	
	175	Save data	Pass
	200	Fail due to upper bound	Pass

	hello	Fail due to invalid data type	Pass
	100.5	Fail due to invalid data type	Pass
Change PVARP	100	Fail due to lower bound	Pass
	150	Save data	Pass
	200	Save data	Pass
	255	Fail due to increment	
	500	Save data	Pass
	550	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
	200.5	Fail due to invalid data type	Pass
Change hysteresis rate smoothing	0	Save data	Pass
	29	Fail due to lower bound	Pass
	30	Save data	Pass
	45	Save data	Pass
	47	Fail due to increment	Pass
	50	Save data	Pass
	62	Save data	Pass
	90	Save data	Pass
	100	Save data	Pass
	102	Fail due to increment	Pass

	175	Save data	Pass
	hello	Fail due to invalid data type	Pass
	100.5	Fail due to invalid data type	Pass
Change rate smoothing	0	Save data	Pass
	3	Save data	Pass
	6	Save data	Pass
	9	Save data	Pass
	12	Save data	Pass
	15	Save data	Pass
	18	Save data	Pass
	21	Save data	Pass
	25	Save data	Pass
	hello	Fail due to invalid data type	Pass
	3.5	Fail due to invalid data type	Pass
Change activity threshold	V-Low	Save data	Pass
	Low	Save data	Pass
	Med-Low	Save data	Pass
	Med	Save data	Pass
	Med-High	Save data	Pass
	High	Save data	Pass
	V-High	Save data	Pass
	hello	Fail due to invalid input	Pass

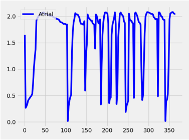
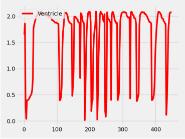
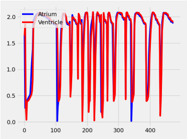
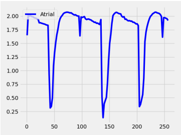
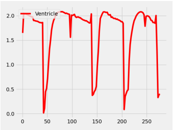
	5	Fail due to invalid data type	Pass
Change reaction time	5	Fail due to lower bound	Pass
	10	Save data	Pass
	30	Save data	Pass
	35	Fail due to increment	Pass
	50	Save data	Pass
	60	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
	30.5	Fail due to invalid data type	Pass
Change response factor	0	Fail due to lower bound	Pass
	1	Save data	Pass
	8	Save data	Pass
	16	Save data	Pass
	17	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
	8.5	Fail due to invalid data type	Pass
Change recovery time	1	Fail due to lower bound	Pass
	2	Save data	Pass
	8	Save data	Pass

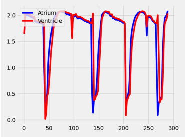
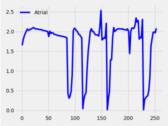
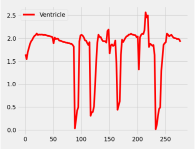
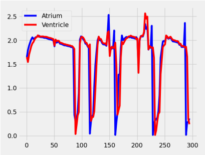
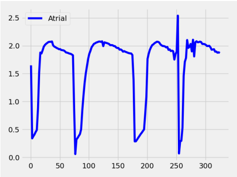
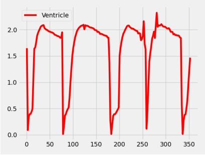
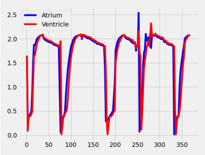
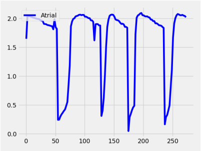
	16	Save data	Pass
	17	Fail due to upper bound	Pass
	hello	Fail due to invalid data type	Pass
	8.5	Fail due to invalid data type	Pass

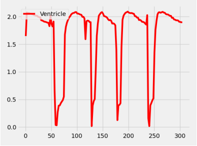
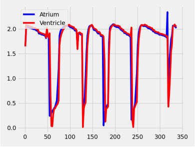
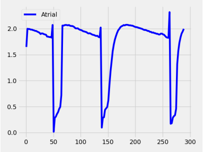
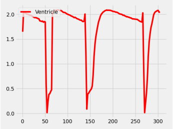
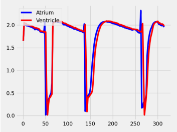
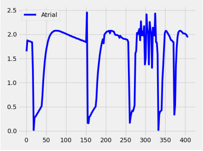
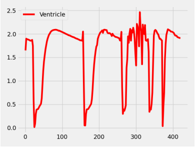
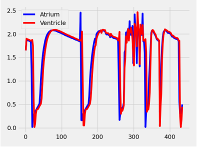
Table 18 - Edit Profile Window Tests

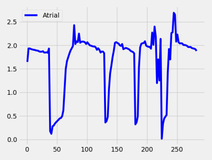
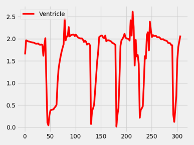
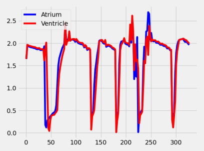
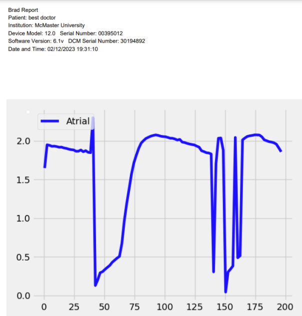
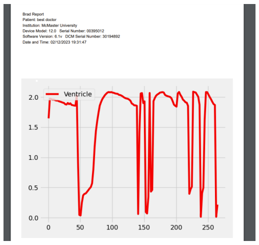
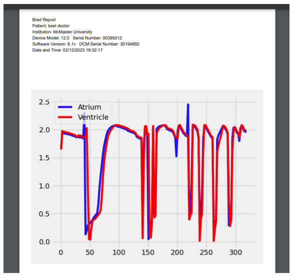
Egram Display Window

The purpose of these test cases is to ensure that the egram display window is fully functional. This includes being able to download the egram report to the user's files and display the atrial and ventricular signals of all the different modes.

Test	System Input	Expected Result	Outcome
AOO Display	Clicks "Atrial Signal" Button	Atrial Signal is Graphed in Live Time	 Pass
	Clicks "Ventricular Signal" Button	Ventricular Signal is Graphed in Live Time	 Pass
	Clicks "Both Signal" Button	Ventricular and Atrial Signals are Graphed in Live Time	 Pass
AAI Display	Clicks "Atrial Signal" Button	Atrial Signal is Graphed in Live Time	 Pass
	Clicks "Ventricular Signal" Button	Ventricular Signal is Graphed in Live Time	 Pass

	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time	 Pass
AOOR Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
AAIR Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time.Graph should show rate adaptive behaviour.	 Pass
	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
VOO Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time	 Pass

	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time	 Pass
VVI Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time	 Pass
	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time	 Pass
VOOR Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass

VVIR Display	Clicks “Atrial Signal” Button	Atrial Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Ventricular Signal” Button	Ventricular Signal is Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
	Clicks “Both Signal” Button	Ventricular and Atrial Signals are Graphed in Live Time. Graph should show rate adaptive behaviour.	 Pass
Print Egram Button	Clicks “Atrial Signal” Button and “Print Egram” Button	Should output a PDF with the current signal	Brad Report Patient: best doctor Institution: McMaster University Device Model: 12.0 Serial Number: 00395012 Software Version: 6.1v DCM Serial Number: 30194892 Date and Time: 02/12/2023 19:31:10 Atrial 2.0 1.5 1.0 0.5 0.0 0 25 50 75 100 125 150 175 200 Pass
	Clicks “Ventricular Signal” Button and “Print Egram” Button	Should output a PDF with the current signal	Brad Report Patient: best doctor Institution: McMaster University Device Model: 12.0 Serial Number: 00395012 Software Version: 6.1v DCM Serial Number: 30194892 Date and Time: 02/12/2023 19:31:47 Ventricle 2.0 1.5 1.0 0.5 0.0 0 50 100 150 200 250 Pass
	Clicks “Both Signal” Button and “Print Egram” Button	Should output a PDF with the current signals	Brad Report Patient: best doctor Institution: McMaster University Device Model: 12.0 Serial Number: 00395012 Software Version: 6.1v DCM Serial Number: 30194892 Date and Time: 02/12/2023 19:32:17 Atrium Ventricle 2.5 2.0 1.5 1.0 0.5 0.0 0 50 100 150 200 250 300 Pass

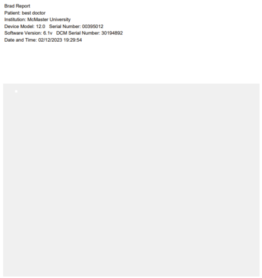
	Does Not Select Any Signal Button and Clicks “Print Egram” Button	A PDF will be created without any signal(s)	 Brad Report Patient: best doctor Institution: McMaster University Device Model: 12.0 Serial Number: 00395012 Software Version: 6.1v DCM Serial Number: 30194892 Date and Time: 02/12/2023 19:29:54 Pass

Table 19 - Egram Display Window Tests