

# Deliverable 1: Documentation Outline

Group #2

Fall 2025

# Contents

<b>1</b>	<b>Group Members</b>	<b>3</b>
<b>2</b>	<b>Part 1</b>	<b>3</b>
2.1	Introduction . . . . .	3
2.1.1	Purpose . . . . .	3
2.1.2	Goals . . . . .	3
2.1.3	Scope . . . . .	3
2.2	Requirements . . . . .	4
2.2.1	Modes . . . . .	4
	a) AOO Mode . . . . .	4
	b) VOO Mode . . . . .	4
	c) AAI Mode . . . . .	4
	d) VVI Mode . . . . .	4
2.3	Design . . . . .	5
2.3.1	Pacemaker . . . . .	5
2.3.2	DCM . . . . .	5
	a) Programmable Parameters . . . . .	5
<b>3</b>	<b>Part 2</b>	<b>6</b>
3.1	Requirements / Potential Changes . . . . .	6
3.2	Design Decision / Potential Changes . . . . .	7
3.3	Module Description . . . . .	7
	HelpWindow . . . . .	7
	ParamEnum . . . . .	7
	ParameterManager . . . . .	8
	WelcomeWindow . . . . .	8
3.4	Testing . . . . .	8
3.5	GenAI Usage . . . . .	10
<b>4</b>	<b>General Notes</b>	<b>10</b>
<b>5</b>	<b>Figures and Tables</b>	<b>11</b>

# 1 Group Members

Jeffrey Yueh	yuehj	400495097
Johnson Ji	jih21	400499564
Kelby To	tok13	400507403
Hongliang Qi	qih25	400493278

## 2 Part 1

### 2.1 Introduction

#### 2.1.1 Purpose

The purpose of a pacemaker is to regulate and restore a normal heart rhythm in patients with cardiac disorders such as arrhythmias, bradycardia, and heart failure. The pacemaker accomplishes this by delivering small electrical pulses to the atria and ventricles to make the heart beat at the correct speed and pattern.

The system consists of two main components: the **Pacemaker** and the **Device Controller-Monitor (DCM)**. The Pacemaker handles sensing and pacing functions, while the DCM allows users to configure, monitor, and manage the settings of the Pacemaker.

#### 2.1.2 Goals

The main goal of Deliverable 1 is to design and implement the foundational components of the Pacemaker and DCM. Specifically:

- **Pacemaker:** Create stateflow models for AOO, VOO, AAI, and VVI modes with parameters specified in the Deliverable 1 document.
- **DCM:** Develop an interface that enables user registration and login, displays pacing modes, and allows input and storage of modifiable parameters.
- **Documentation:** Provide a detailed document outlining the design process, decisions, implementation, and testing procedures for the Pacemaker and DCM.

#### 2.1.3 Scope

Deliverable 1 focuses on developing the initial components of the Pacemaker and DCM. This includes:

- Creating stateflow models for the AOO, VOO, AAI, and VVI pacing modes.
- Building the DCM interface with registration, login, and parameter customization functionality.

- Documenting the design and implementation process.

This deliverable does not include advanced features such as wireless communication, complex arrhythmia detection algorithms, or additional pacing modes beyond those specified. Hardware testing is limited to simulation and software verification environments.

## 2.2 Requirements

### 2.2.1 Modes

#### *a) AOO Mode*

- **Pacing:** Atrial only
- **Sensing:** Atrial
- **Response to Sensing:** Inhibited
- **Behaviour:** Paces the atrium only if no intrinsic atrial activity is sensed within the programmed interval.

#### *b) VOO Mode*

- **Pacing:** Ventricular only
- **Sensing:** None
- **Response to Sensing:** None (asynchronous)
- **Behaviour:** Delivers ventricular pacing pulses at a fixed rate, regardless of intrinsic activity.

#### *c) AAI Mode*

- **Pacing:** Atrial
- **Sensing:** Atrial
- **Response to Sensing:** Inhibited
- **Behaviour:** Paces the atrium only if no intrinsic atrial activity is sensed within the programmed interval

#### *d) VVI Mode*

- **Pacing:** Ventricular
- **Sensing:** Ventricular
- **Response to Sensing:** Inhibited
- **Behaviour:** Paces the ventricle only if no intrinsic ventricular activity is sensed within the programmed interval

## 2.3 Design

### 2.3.1 Pacemaker

### 2.3.2 DCM

#### *a) Programmable Parameters*

In this section, you should expand on design decisions based on the requirements. Be specific about your system design and how components interact.

- System architecture (major subsystems, hardware abstraction, pin mapping)
- Programmable parameters (rate limits, amplitudes, pulse widths, refractory periods, etc.)
- Hardware inputs and outputs (signals sensed, signals controlled)
- State machine design for each pacing mode (include diagrams or tables if applicable)
- Simulink diagram
- Screenshots of your DCM, explaining its software structure

Explicitly explain how each design decision maps directly to the stated requirements.

## 3 Part 2

### 3.1 Requirements / Potential Changes

Module	Requirements	Potential Changes / Evolution
HelpWindow	Display help documentation for parameters and pacing modes, with navigation and formatted text.	Adding new help topics (e.g., D2 modes), support for multimedia content, or online help updates.
ParamEnum	Store pacemaker modes and parameters, provide getter and setter interfaces with validation and stepping rules.	Adding new pacing modes (D2, AOOR, etc.), new parameters for advanced therapy modes.
ParameterManager	Manage parameter operations: save, load, reset, and apply; GUI to edit parameters based on mode.	Improve validation rules, support bulk import/export, additional GUI elements for future parameters.
WelcomeWindow	Provide login and registration functionality; launch dashboard upon successful login.	Integration with secure authentication APIs, multi-user support, and potential cloud-based storage.

### 3.2 Design Decision / Potential Changes

Module	Design Decisions	Potential Changes / Revisions
HelpWindow	Tkinter-based GUI; JSON files for content; text widget formatting.	Consider switching to web-based help, dynamic content loading, or modular UI frameworks.
ParamEnum	Python class with validated getters/setters; uses constants for mode definitions.	Might adopt database-driven parameters, use enums for clarity, or external config files for scalability.
ParameterManager	GUI-bound parameter management; state tracking; method resolution via string names.	Refactor to MVC pattern, improve input validation, or integrate real-time device updates.
WelcomeWindow	Simple Tkinter GUI for login/registration; dashboard launch.	Upgrade authentication security, support OAuth, and redesign GUI for modern UX.

### 3.3 Module Description

#### HelpWindow

- **Purpose:** Display help documentation for pacemaker modes and parameters.
- **Key functions/methods:**
  - Public: `update_content`, `load_help_content`
  - Internal: `_display_param_document`, `_display_mode_document`, `_display_text_content`
- **Global/state variables:** `topics`, `current_topic`, `content_area`.
- **Interactions:** Reads JSON help files; updates GUI content dynamically.

#### ParamEnum

- **Purpose:** Store parameter values and enforce validation rules.
- **Key functions/methods:**
  - Public: getters (`get_*`) and setters (`set_*`), `get_default_values`
  - Internal: `_is_number`, `_round_to_step`

- **Global/state variables:** parameter values (e.g., `Lower_Rate_Limit`, `ARP`); `MODES` dictionary.
- **Interactions:** Used by `ParameterManager` and GUI modules.

#### ParameterManager

- **Purpose:** Enable saving, loading, resetting, applying, and editing parameters through a GUI.
- **Key functions/methods:**
  - Public: `save_params`, `load_params`, `reset_params`, `apply`, `save_and_round`
  - Internal: `_resolve_method`, `_getter_candidates_for_key`, `_setter_candidates_for_key`, `_mark_unsaved`, `_on_close`
- **Global/state variables:** `param_entries`, `mode_var`, `_saved_ok`, `param_manager`.
- **Interactions:** Reads/writes JSON parameters; updates `ParamEnum` values; GUI reflects changes.

#### WelcomeWindow

- **Purpose:** Manage user authentication and launch the dashboard.
- **Key functions/methods:** `register`, `login`
- **Global/state variables:** `name_entry`, `pass_entry`, `root`.
- **Interactions:** Calls `register_user` and `login_user`; launches `DashboardWindow`.

### 3.4 Testing

#### HelpWindow Module

Test Case	Purpose	Input	Expected Output	Result
Display help content	Ensure help topics load correctly	Open HelpWindow	Help text displayed for each topic, formatted correctly	Pass
Switch topic	Verify topic switching	Click different topic button	Content area updates with selected topic	Pass
Missing JSON file	Check error handling	Remove <code>Param_Help.json</code>	Display error message in content area	Pass

#### ParamEnum Module

Test Case	Purpose	Input	Expected Output	Result
Set Lower Rate Limit	Validate stepping rules	<code>set.Lower.Rate.Limit(47)</code>	Value rounded to nearest step (45 or 50 depending on rule)	Pass
Set Upper Rate Limit	Check upper bound	<code>set.Upper.Rate.Limit(180)</code>	Error raised: out of range	Pass
Get default values	Ensure getters return defaults	<code>get.default.values()</code>	Dictionary with correct default values	Pass

#### ParameterManager Module

Test Case	Purpose	Input	Expected Output	Result
Save parameters	Verify JSON saving	Edit parameters, click Save	parameters.json created/updated	Pass
Load parameters	Verify JSON loading	Click Load	Entries updated with stored values	Pass
Apply parameters	Check apply function	Modify parameters, click Apply	Mode applied, entries validated	Pass
Reset parameters	Check reset functionality	Click Reset	All parameters revert to defaults	Pass
Validation	Input invalid values	Enter 200 for Upper Rate Limit	Error message displayed	Pass

#### WelcomeWindow Module

Test Case	Purpose	Input	Expected Output	Result
User registration	Validate registration	Enter name/password, click Register	Success message or error if user exists	Pass
User login success	Test login workflow	Enter valid credentials, click Login	Launch DashboardWindow	Pass
User login failure	Test invalid login	Enter wrong credentials	Display error message	Pass

#### WelcomeWindow Module

Test Case	Purpose	Input	Expected Output	Result
User registration	Validate registration	Enter name/password, click Register	Success message or error if user exists	Pass
User login success	Test login workflow	Enter valid credentials, click Login	Launch DashboardWindow	Pass
User login failure	Test invalid login	Enter wrong credentials	Display error message	Pass

### 3.5 GenAI Usage

Provide a summary of any usage of GenAI tools in developing the model, DCM, or documentation. If no GenAI tools were used, state that explicitly.

## 4 General Notes

- This outline is based on the Deliverable 1 handout; ensure all required sections are included.
- Include screenshots of Simulink diagrams and the DCM interface where applicable.
- Ensure requirements are traceable to design and test cases.
- Keep content concise and clear.
- You may add other sections or modify this structure as needed, but these are the main expected components.

## 5 Figures and Tables

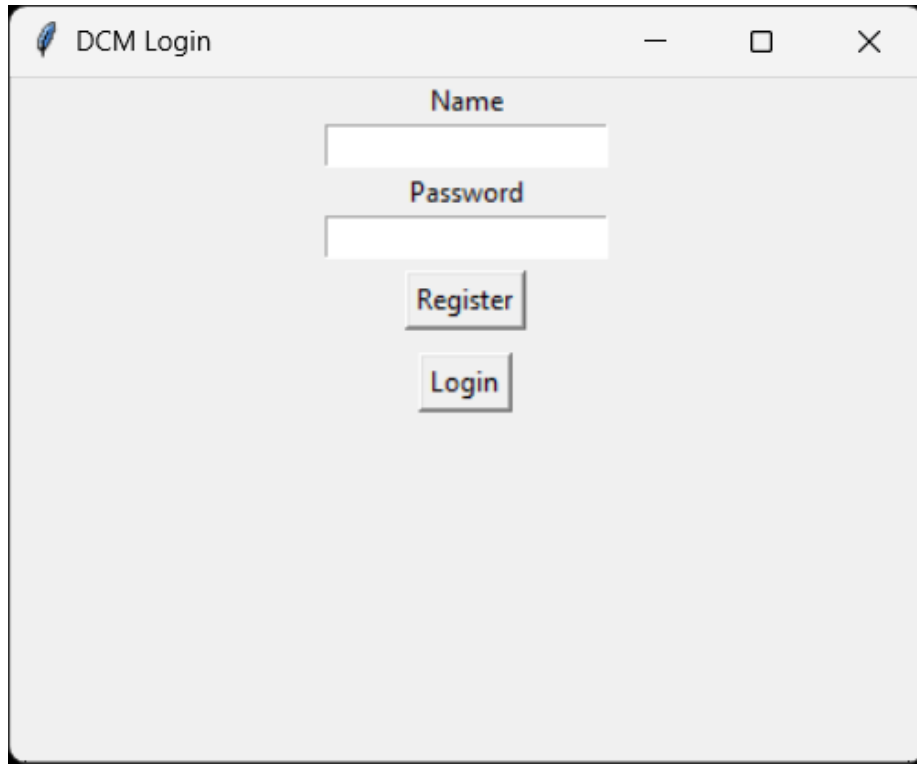


Figure 1: Login Screen

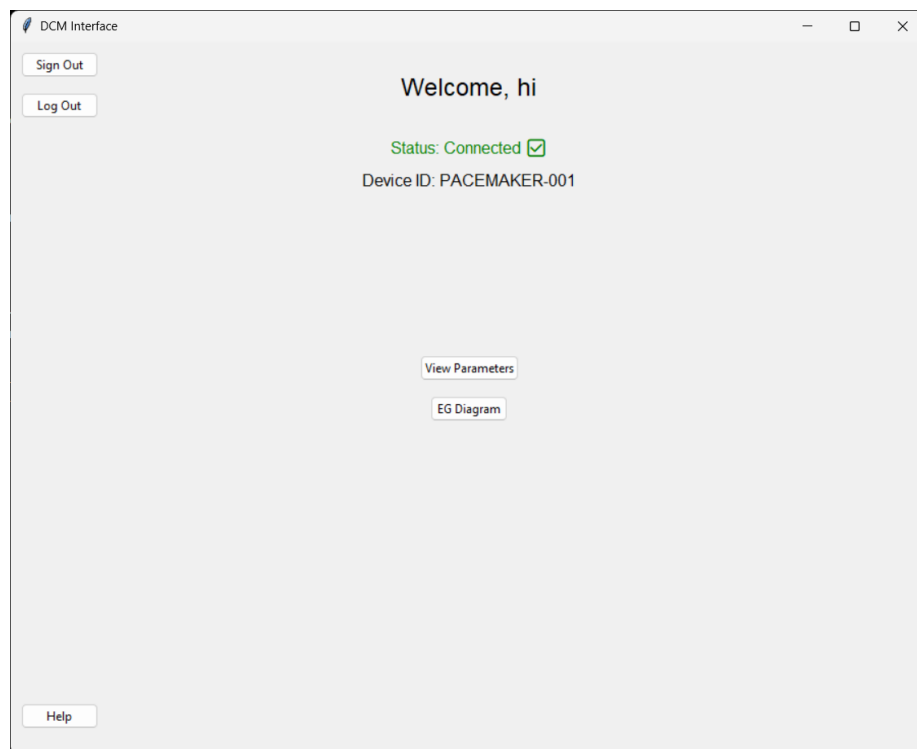


Figure 2: Home Screen