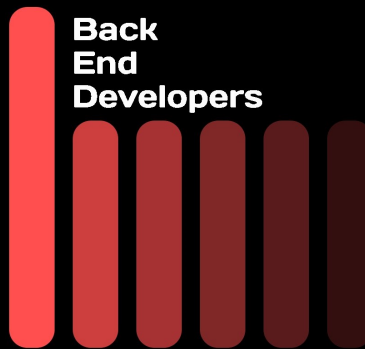


# Module Interface Specification for Mechatronics Engineering



Team #1, Back End Developers

Jessica Bae

Oliver Foote

Jonathan Hai

Anish Rangarajan

Nish Shah

Labeeb Zaker

April 6, 2023

## 1 Revision History

Date	Version	Notes
2023-01-18	1.0	Initial documentation
2023-03-15	2.0	Minor improvements and proof reading for revision 1
2023-04-03	2.1	Incorporated TA feedback
2023-04-03	2.2	Included logo and added style to the document

## **2 Symbols, Abbreviations and Acronyms**

Please refer to the System Requirements Specifications document at [this link](#) for relevant symbols, abbreviations.

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Introduction</b>	<b>1</b>
<b>4</b>	<b>Notation</b>	<b>1</b>
<b>5</b>	<b>Module Decomposition</b>	<b>2</b>
<b>6</b>	<b>MIS of Battery Module</b>	<b>3</b>
6.1	Module . . . . .	3
6.2	Uses . . . . .	3
6.3	Syntax . . . . .	3
6.3.1	Exported Constants . . . . .	3
6.3.2	Exported Access Programs . . . . .	3
6.4	Semantics . . . . .	3
6.4.1	State Variables . . . . .	3
6.4.2	Environment Variables . . . . .	3
6.4.3	Assumptions . . . . .	3
6.4.4	Access Routine Semantics . . . . .	3
6.4.5	Local Functions . . . . .	3
<b>7</b>	<b>MIS of microSD Module</b>	<b>4</b>
7.1	Module . . . . .	4
7.2	Uses . . . . .	4
7.3	Syntax . . . . .	4
7.3.1	Exported Constants . . . . .	4
7.3.2	Exported Access Programs . . . . .	4
7.4	Semantics . . . . .	4
7.4.1	State Variables . . . . .	4
7.4.2	Environment Variables . . . . .	4
7.4.3	Assumptions . . . . .	4
7.4.4	Access Routine Semantics . . . . .	5
7.4.5	Local Functions . . . . .	5
7.4.6	FS Datatype Details . . . . .	5
<b>8</b>	<b>MIS of Local Database Module</b>	<b>6</b>
8.1	Module . . . . .	6
8.2	Uses . . . . .	6
8.3	Syntax . . . . .	6
8.3.1	Exported Constants . . . . .	6
8.3.2	Exported Access Programs . . . . .	6

8.4	Semantics . . . . .	7
8.4.1	State Variables . . . . .	7
8.4.2	Environment Variables . . . . .	7
8.4.3	Assumptions . . . . .	7
8.4.4	Access Routine Semantics . . . . .	7
8.4.5	Local Functions . . . . .	8
8.4.6	path-like Datatype Details . . . . .	8
8.4.7	Connection Datatype Details . . . . .	8
8.4.8	Cursor Datatype Details . . . . .	8
8.4.9	ProgrammingError Datatype Details . . . . .	8
<b>9</b>	<b>MIS of Reading Sensor Module</b>	<b>9</b>
9.1	Module . . . . .	9
9.2	Uses . . . . .	9
9.3	Syntax . . . . .	9
9.3.1	Exported Constants . . . . .	9
9.3.2	Exported Access Programs . . . . .	9
9.4	Semantics . . . . .	9
9.4.1	State Variables . . . . .	9
9.4.2	Environment Variables . . . . .	9
9.4.3	Assumptions . . . . .	10
9.4.4	Access Routine Semantics . . . . .	10
9.4.5	Local Functions . . . . .	10
<b>10</b>	<b>MIS of Sensor Data Processing Module</b>	<b>11</b>
10.1	Module . . . . .	11
10.2	Uses . . . . .	11
10.3	Syntax . . . . .	11
10.3.1	Exported Constants . . . . .	11
10.3.2	Exported Access Programs . . . . .	11
10.4	Semantics . . . . .	11
10.4.1	State Variables . . . . .	11
10.4.2	Environment Variables . . . . .	11
10.4.3	Assumptions . . . . .	11
10.4.4	Access Routine Semantics . . . . .	12
10.4.5	Local Functions . . . . .	12
<b>11</b>	<b>MIS of Display System Module</b>	<b>13</b>
11.1	Module . . . . .	13
11.2	Uses . . . . .	13
11.3	Syntax . . . . .	13
11.3.1	Exported Constants . . . . .	13
11.3.2	Exported Access Programs . . . . .	13
11.4	Semantics . . . . .	13
11.4.1	State Variables . . . . .	13

11.4.2 Environment Variables . . . . .	14
11.4.3 Assumptions . . . . .	14
11.4.4 Access Routine Semantics . . . . .	14
11.4.5 Local Functions . . . . .	14
<b>12 MIS of Prompt Generation Module</b>	<b>15</b>
12.1 Module . . . . .	15
12.2 Uses . . . . .	15
12.3 Syntax . . . . .	15
12.3.1 Exported Constants . . . . .	15
12.3.2 Exported Access Programs . . . . .	15
12.4 Semantics . . . . .	15
12.4.1 State Variables . . . . .	15
12.4.2 Environment Variables . . . . .	15
12.4.3 Assumptions . . . . .	15
12.4.4 Access Routine Semantics . . . . .	15
12.4.5 Local Functions . . . . .	16
12.4.6 Prompt Datatype Details . . . . .	16
<b>13 MIS of Real Time Clock Module</b>	<b>17</b>
13.1 Module . . . . .	17
13.2 Uses . . . . .	17
13.3 Syntax . . . . .	17
13.3.1 Exported Constants . . . . .	17
13.3.2 Exported Access Programs . . . . .	17
13.4 Semantics . . . . .	17
13.4.1 State Variables . . . . .	17
13.4.2 Environment Variables . . . . .	18
13.4.3 Assumptions . . . . .	18
13.4.4 Access Routine Semantics . . . . .	18
13.4.5 Local Functions . . . . .	18
13.4.6 RTC_DATE_TIME Datatype Details . . . . .	18
13.4.7 RTC_DS1307 Datatype Details . . . . .	18
13.4.8 DateTime Datatype Details . . . . .	18
<b>14 MIS of Configuration Module</b>	<b>19</b>
14.1 Module . . . . .	19
14.2 Uses . . . . .	19
14.3 Syntax . . . . .	19
14.3.1 Exported Constants . . . . .	19
14.3.2 Exported Access Programs . . . . .	19
14.4 Semantics . . . . .	19
14.4.1 State Variables . . . . .	19
14.4.2 Environment Variables . . . . .	20
14.4.3 Assumptions . . . . .	20

14.4.4	Access Routine Semantics	20
14.4.5	Local Functions	20
14.4.6	QPushButton Datatype Details	20
<b>15</b>	<b>MIS of Graph Plotter</b>	<b>21</b>
15.1	Module	21
15.2	Uses	21
15.3	Syntax	21
15.3.1	Exported Constants	21
15.3.2	Exported Access Programs	21
15.4	Semantics	21
15.4.1	State Variables	21
15.4.2	Environment Variables	22
15.4.3	Assumptions	22
15.4.4	Access Routine Semantics	22
15.4.5	Local Functions	22
15.4.6	QPushButton Datatype Details	22
15.4.7	path-like Datatype Details	23
15.4.8	Connection Datatype Details	23
15.4.9	Cursor Datatype Details	23
15.4.10	ProgrammingError Datatype Details	23
<b>16</b>	<b>MIS of Watch Case and Straps Module</b>	<b>24</b>
16.1	Module	24
16.2	Uses	24
16.3	Syntax	24
16.3.1	Exported Constants	24
16.3.2	Exported Access Programs	24
16.4	Semantics	24
16.4.1	State Variables	24
16.4.2	Environment Variables	24
16.4.3	Assumptions	24
16.4.4	Access Routine Semantics	24
16.4.5	Local Functions	24

### 3 Introduction

The following document details the Module Interface Specifications for the EMAnator; the system currently being developed by the Back End Developers designed to aid in Ecological Momentary Assessment research. This document describes the various relevant details of interfacing with each module. These details include module descriptions, the uses of each module, the syntax of each module, and the semantics associated with each module.

Complementary documents include the System Requirement Specifications and the Module Guide. The Back End Developers highly recommend a thorough read-through of each document prior to a reading of this document to attain the prerequisite knowledge necessary to fully understand this MIS. The System Requirements Specifications can be found at [this link](#), and the Module Guide can be found at [this link](#).

### 4 Notation

The structure of the MIS for modules comes from [Hoffman and Strooper \(1995\)](#), with the addition that template modules have been adapted from [Ghezzi et al. \(2003\)](#). The mathematical notation comes from Chapter 3 of [Hoffman and Strooper \(1995\)](#). For instance, the symbol  $:=$  is used for a multiple assignment statement and conditional rules follow the form  $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$ .

The following table summarizes the primitive data types used by Mechatronics Engineering.

Data Type	Notation	Description
Character	char	A single symbol or digit
Integer	$\mathbb{Z}$	A number without a fractional component in $(-\infty, \infty)$
Natural number	$\mathbb{N}$	A number without a fractional component in $[1, \infty)$
Real	$\mathbb{R}$	Any number in $(-\infty, \infty)$

The specification of Mechatronics Engineering uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, Mechatronics Engineering uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.



## 5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2	Level 3
Hardware-Hiding Module	Battery Management	Battery
	Data Storage	microSD Database
	Sensor Array	Sensor Reading Sensor Data Processing Sensor Prompt Validity
	Physical Design	Watch Straps Watch Case
Behaviour-Hiding Module	Display System	Display Screen
	Prompt Generation	Prompt Generation
	Real Time Clock	RTC
Software Decision Module	Parameter Selection	Create New User Configuration
	Data Processing	Graph Data Display

Table 1: Module Hierarchy

## 6 MIS of Battery Module

### 6.1 Module

Bat\_Man

### 6.2 Uses

None.

### 6.3 Syntax

#### 6.3.1 Exported Constants

- BAT\_LOW\_THRESHOLD:  $\mathbb{Z}$

#### 6.3.2 Exported Access Programs

Name	In	Out	Exceptions
bed_get.bat_level	-	battery_level: $\mathbb{Z}$	-

### 6.4 Semantics

#### 6.4.1 State Variables

None.

#### 6.4.2 Environment Variables

- battery\_level:  $\mathbb{Z}$

#### 6.4.3 Assumptions

System responds instantaneously to changes in flags (exported constants).

#### 6.4.4 Access Routine Semantics

- bed\_get.bat\_level: This function returns the battery voltage level as a percentage of the battery's full charge.

#### 6.4.5 Local Functions

None.

## 7 MIS of microSD Module

### 7.1 Module

microSD\_Stor

### 7.2 Uses

Sensor Prompt Validity Module (Section ??), Real Time Clock Module (Section 13)

### 7.3 Syntax

#### 7.3.1 Exported Constants

None.

#### 7.3.2 Exported Access Programs

Name	In	Out	Exceptions
listDir	fs: type FS, dirname: char *, levels: $\mathbb{Z}$	-	-
createDir	fs: type FS, path: char *	-	-
removeDir	fs: type FS, path: char *	-	-

### 7.4 Semantics

#### 7.4.1 State Variables

None.

#### 7.4.2 Environment Variables

- fs: type FS

#### 7.4.3 Assumptions

- MicroSD card is formatted correctly.
- MicroSD card is inserted correctly.

#### 7.4.4 Access Routine Semantics

- `listDir`: This function lists all files in the path given.
- `createDir`: This function creates a new folder at the designated path.
- `removeDir`: This function removes a folder at the designated path.

#### 7.4.5 Local Functions

Name	In	Out	Exceptions
<code>readFile</code>	fs: type FS, path: char *	-	-
<code>writeFile</code>	path: char *, message: char *	-	-
<code>appendFile</code>	fs: type FS, path: char *, message: char *	-	-
<code>renameFile</code>	fs: type FS, path1: char *, path2: char *	-	-
<code>deleteFile</code>	fs: type FS, path: char *	-	-

#### 7.4.6 FS Datatype Details

The FS object as defined by the SD.h class.

## 8 MIS of Local Database Module

### 8.1 Module

Database\_Stor

### 8.2 Uses

microSD Module (Section 7)

### 8.3 Syntax

#### 8.3.1 Exported Constants

- MAX\_CHAR\_LIMIT:  $\mathbb{Z}$
- MAX\_FIRST\_NAME\_SIZE:  $\mathbb{Z}$
- MAX\_LAST\_NAME\_SIZE:  $\mathbb{Z}$
- MAX\_GENDER\_SIZE:  $\mathbb{Z}$
- MAX\_PHONE\_SIZE:  $\mathbb{Z}$
- MAX\_EMAIL\_SIZE:  $\mathbb{Z}$
- MAX\_ADDRESS\_SIZE:  $\mathbb{Z}$
- MAX\_DEVICE\_MODEL\_SIZE:  $\mathbb{Z}$

#### 8.3.2 Exported Access Programs

Name	In	Out	Exceptions
sqlite3.connect	database: type path-like	connection: type Connection	ProgrammingError: type Exception
conn.cursor	-	cursor: type Cursor	ProgrammingError: type Exception
cursor.execute	sql: char array	-	ProgrammingError: type Exception
conn.commit	-	-	ProgrammingError: type Exception
conn.close	-	-	ProgrammingError: type Exception

## 8.4 Semantics

### 8.4.1 State Variables

None.

### 8.4.2 Environment Variables

- FirstName: char array
- LastName: char array
- Gender: char array
- PhoneNumber: char array
- EmailID: char array
- Address: char array
- MonitoringPeriod: char array
- TrackerModel: char array
- Age:  $\mathbb{Z}$
- ParticipantID:  $\mathbb{Z}$
- StudyID:  $\mathbb{Z}$
- Weight: float
- Height: float

### 8.4.3 Assumptions

None.

### 8.4.4 Access Routine Semantics

- sqlite3.connect: Performs a handshake between the database and the host software
- conn.cursor: Establishes an object through which database transactions occur
- cursor.execute: Executes the SQL statement to the database on the current transaction
- conn.commit: Commits any pending transaction to the database
- conn.close: Closes the database connection

#### **8.4.5 Local Functions**

None.

#### **8.4.6 path-like Datatype Details**

The path-like-object is an object which contains the string of the path to the .db database file

#### **8.4.7 Connection Datatype Details**

An object representing the sqlite3 object.

#### **8.4.8 Cursor Datatype Details**

An object which contains the functions which manipulate the database

#### **8.4.9 ProgrammingError Datatype Details**

A subclass of DatabaseError datatype.

## 9 MIS of Reading Sensor Module

### 9.1 Module

Sensor\_Reading

### 9.2 Uses

Battery Management (Section 6)

### 9.3 Syntax

#### 9.3.1 Exported Constants

- ACCEL\_SENSITIVITY:  $\mathbb{Z}$
- GYRO\_SENSITIVITY:  $\mathbb{Z}$
- Threshold:  $\mathbb{Z}$
- MPU\_CALIBRATION:  $\mathbb{Z}$

#### 9.3.2 Exported Access Programs

Name	In	Out	Exceptions
bed_mpu_detect	currentTime: $\mathbb{Z}$	-	-
bed_hr_detect	-	-	-

### 9.4 Semantics

#### 9.4.1 State Variables

- currentTime:  $\mathbb{Z}$

#### 9.4.2 Environment Variables

- curr\_ax:  $\mathbb{Z}$
- curr\_ay:  $\mathbb{Z}$
- curr\_a $\mathbb{Z}$ :  $\mathbb{Z}$
- curr\_gx:  $\mathbb{Z}$
- curr\_gy:  $\mathbb{Z}$
- curr\_g $\mathbb{Z}$ :  $\mathbb{Z}$



### 9.4.3 Assumptions

- All activity thresholds are provided from the configuration file.

### 9.4.4 Access Routine Semantics

- `bed_mpu_detect`: returns the current values of accelerometer and gyroscope.
- `bed_hr_detect`: returns the current values of heart rate sensor.

### 9.4.5 Local Functions

Name	In	Out	Exceptions
<code>bed_mpu_setup</code>	-	<code>b32_err_code</code> : $\mathbb{Z}$	<code>b32_err_code</code> : BED_ERR_MPU
<code>bed_hr_setup</code>	-	-	-

## 10 MIS of Sensor Data Processing Module

### 10.1 Module

Sensor\_Data

### 10.2 Uses

Sensor Reading (Section 9)

### 10.3 Syntax

#### 10.3.1 Exported Constants

- ACTIVITY\_STEPS:  $\mathbb{Z}$
- ACTIVITY\_IDLE\_RESET:  $\mathbb{Z}$
- ACTIVITY\_IDLE\_WAIT:  $\mathbb{Z}$

#### 10.3.2 Exported Access Programs

Name	In	Out	Exceptions
bed_mpu_detect	currentTime: $\mathbb{Z}$		-
bed_hr_detect	-	-	-

### 10.4 Semantics

#### 10.4.1 State Variables

- currentTime:  $\mathbb{Z}$

#### 10.4.2 Environment Variables

- step\_count:  $\mathbb{Z}$
- TOTAL\_STEP:  $\mathbb{Z}$
- step\_flag:  $\mathbb{Z}$
- activity\_flag:  $\mathbb{Z}$

#### 10.4.3 Assumptions

- There is space available in microSD card.

#### 10.4.4 Access Routine Semantics

- `bed_mpu_detect`: returns the current values of accelerometer and gyroscope
- `bed_hr_detect`: returns the current values of heart rate sensor

#### 10.4.5 Local Functions

Name	In	Out	Exceptions
<code>bed_mpu_setup</code>	-	<code>b32_err_code</code> : $\mathbb{Z}$	<code>b32_err_code</code> : <code>BED_ERR_MPU</code>
<code>bed_hr_setup</code>	-	-	-

## 11 MIS of Display System Module

### 11.1 Module

Disp\_Sys

### 11.2 Uses

Prompt Generation Module (Section 12), Real Time Clock Module (Section 13), Battery Management (Section 6)

### 11.3 Syntax

#### 11.3.1 Exported Constants

- TEXT\_SIZE:  $\mathbb{Z}$
- BED\_TFT\_CS:  $\mathbb{Z}$
- BED\_TFT\_DC:  $\mathbb{Z}$
- BED\_TFT\_MOSI:  $\mathbb{Z}$
- BED\_TFT\_SCK:  $\mathbb{Z}$

#### 11.3.2 Exported Access Programs

Name	In	Out	Exceptions
bed_display_prompt	direction: $\mathbb{Z}$ , prompt_index: $\mathbb{Z}$ , no_of_options: $\mathbb{Z}$ , flag: $\mathbb{Z}$	-	-
bed_display_date_time		-	-

### 11.4 Semantics

#### 11.4.1 State Variables

- draw\_flag:  $\mathbb{Z}$
- wait\_flag:  $\mathbb{Z}$

### 11.4.2 Environment Variables

- scroll\_index:  $\mathbb{Z}$
- curr\_hr:  $\mathbb{Z}$
- curr\_min:  $\mathbb{Z}$
- curr\_sec:  $\mathbb{Z}$
- prompt\_buff: char array
- inPain: char array

### 11.4.3 Assumptions

None.

### 11.4.4 Access Routine Semantics

- bed\_display\_prompt: This function takes the type of prompt and prompt index and accordingly generates the corresponding questions and options. Upon answering all the questions, it will save the answers in the SD card.
- bed\_display\_date\_time: This function grabs the current time and draws it to the screen.

### 11.4.5 Local Functions

Name	In	Out	Exceptions
bed_init_display	-	Response: string	BED_ERR_DISPLAY_SYSTEM: $\mathbb{Z}$
bed_splash_screen	-	-	-
bed_display_one_line	displayText: char *, coordX: $\mathbb{Z}$ , coordY: $\mathbb{Z}$ , clearFlag: $\mathbb{Z}$	-	-
drawHour	-	-	-
drawMinute	-	-	-
drawSecond	-	-	-

## 12 MIS of Prompt Generation Module

### 12.1 Module

Prompt\_Gen

### 12.2 Uses

Sensor Array Module (Section 10)

### 12.3 Syntax

#### 12.3.1 Exported Constants

- MAX\_PROMPTS:  $\mathbb{Z}$
- WALK\_PROMPT\_SIZE:  $\mathbb{Z}$
- ALARM\_PROMPT\_SIZE:  $\mathbb{Z}$
- prompt\_test: type Prompt (struct)
- prompt\_walking: type Prompt (struct)
- prompt\_alarm: type Prompt (struct)

#### 12.3.2 Exported Access Programs

None.

### 12.4 Semantics

#### 12.4.1 State Variables

None.

#### 12.4.2 Environment Variables

None.

#### 12.4.3 Assumptions

None.

#### 12.4.4 Access Routine Semantics

None.

### 12.4.5 Local Functions

None.

### 12.4.6 Prompt Datatype Details

- Prompt is a struct composed of 5 fields:
- `prompt_id`:  $\mathbb{Z}$
- `no_of_options`:  $\mathbb{Z}$
- `prompt_question`: char array
- `prompt_possible_answers`: char array (2D)
- `prompt_response`: char array

## 13 MIS of Real Time Clock Module

### 13.1 Module

RTC

### 13.2 Uses

None.

### 13.3 Syntax

#### 13.3.1 Exported Constants

- RTC\_ADDRESS:  $\mathbb{Z}$
- ALARM\_NUMBER:  $\mathbb{Z}$
- alarms:  $\mathbb{Z}$  array (2D)
- rtc: type RTC\_DS1307
- now: type DateTime
- daysOfTheWeek: char array (2D)
- monthsOfTheYear: char array (2D)
- rtc\_date\_time: type RTC\_DATE\_TIME

#### 13.3.2 Exported Access Programs

Name	In	Out	Exceptions
bed_display_info	-	-	-
bed_alarm_prompt	prev_input: $\mathbb{Z}$	new_input: $\mathbb{Z}$	-
bed_set_explicit_date_time	date: char *, time: char *	-	-

### 13.4 Semantics

#### 13.4.1 State Variables

- alarm\_flag:  $\mathbb{Z}$
- curr\_alarm:  $\mathbb{Z}$



### 13.4.2 Environment Variables

None.

### 13.4.3 Assumptions

- Initial date and time is correctly set.

### 13.4.4 Access Routine Semantics

- `bed_display_info`: This function grabs the current time from the `DateTime` struct, and stores it in the `RTC_DATE_TIME` struct.
- `bed_alarm_prompt`: This function checks whether or not an alarm should be triggered based on the time, and accordingly updates the state.
- `bed_set_explicit_date_time`: This function sets the date and time of the RTC explicitly.

### 13.4.5 Local Functions

Name	In	Out	Exceptions
<code>bed_init_rtc</code>	-	<code>b32_err_code</code> $\mathbb{Z}$	<code>b32_err_code</code> : <code>BED_ERR_RTC_SYSTEM</code>

### 13.4.6 RTC\_DATE\_TIME Datatype Details

`RTC_DATE_TIME` is a struct composed of 6 fields:

- `rtc_year`:  $\mathbb{Z}$
- `rtc_month`:  $\mathbb{Z}$
- `rtc_day`:  $\mathbb{Z}$
- `rtc_hour`:  $\mathbb{Z}$
- `rtc_min`:  $\mathbb{Z}$
- `rtc_sec`:  $\mathbb{Z}$

### 13.4.7 RTC\_DS1307 Datatype Details

The RTC object as defined by the `RTCLib` system class.

### 13.4.8 DateTime Datatype Details

The `DateTime` object as defined by the `RTCLib` system class.

## 14 MIS of Configuration Module

### 14.1 Module

Config\_Param

### 14.2 Uses

MicroSD Module (Section 7)

### 14.3 Syntax

#### 14.3.1 Exported Constants

- MAX\_CHAR\_LIMIT:  $\mathbb{Z}$
- MAX\_FIRST\_NAME\_SIZE:  $\mathbb{Z}$
- MAX\_LAST\_NAME\_SIZE:  $\mathbb{Z}$
- MAX\_GENDER\_SIZE:  $\mathbb{Z}$
- MAX\_PHONE\_SIZE:  $\mathbb{Z}$
- MAX\_EMAIL\_SIZE:  $\mathbb{Z}$
- MAX\_ADDRESS\_SIZE:  $\mathbb{Z}$
- MAX\_DEVICE\_MODEL\_SIZE:  $\mathbb{Z}$

#### 14.3.2 Exported Access Programs

Name	In	Out	Exceptions
object.text	-	enter_text: char array	-

### 14.4 Semantics

#### 14.4.1 State Variables

- CreateRecord: type QPushButton

#### **14.4.2 Environment Variables**

- FirstName: char array
- LastName: char array
- Gender: char array
- PhoneNumber: char array
- EmailID: char array
- Address: char array
- MonitoringPeriod: char array
- TrackerModel: char array
- Age:  $\mathbb{Z}$
- ParticipantID:  $\mathbb{Z}$
- StudyID:  $\mathbb{Z}$
- Weight: float
- Height: float

#### **14.4.3 Assumptions**

- All configuration parameters within acceptable limits.

#### **14.4.4 Access Routine Semantics**

- object.text: This function grabs what is typed into the UI input box and stores it into a variable

#### **14.4.5 Local Functions**

None.

#### **14.4.6 QPushButton Datatype Details**

The object from the PyQt class which enables UI buttons.

## 15 MIS of Graph Plotter

### 15.1 Module

Graph\_Plot

### 15.2 Uses

Device Manager Module (Section ??)

### 15.3 Syntax

#### 15.3.1 Exported Constants

None.

#### 15.3.2 Exported Access Programs

Name	In	Out	Exceptions
OpenHeart	-	-	-
OpenSteps	-	-	-
OpenActivity	-	-	-
sqlite3.connect	database: type path-like	connection: type Con- nection	ProgrammingError: type Exception
conn.cursor	-	cursor: type Cur- sor	ProgrammingError: type Exception
cursor.execute	sql: char array	-	ProgrammingError: type Exception
conn.commit	-	-	ProgrammingError: type Exception
conn.close	-	-	ProgrammingError: type Exception

### 15.4 Semantics

#### 15.4.1 State Variables

- HeartGraph: type QPushButton
- StepsGraph: type QPushButton
- ActivityGraph: type QPushButton

### 15.4.2 Environment Variables

None.

### 15.4.3 Assumptions

- Data is in proper format and not corrupted.

### 15.4.4 Access Routine Semantics

- OpenHeart: This function fetches heart rate and time data from the database and plots it on the graph
- OpenSteps: This function fetches step and time data from the database and plots it on the graph
- OpenActivity: This function fetches activity and time data from the database and plots it on the graph
- sqlite3.connect: Performs a handshake between the database and the host software
- conn.cursor: Establishes an object through which database transactions occur
- cursor.execute: Executes the SQL statement to the database on the current transaction
- conn.commit: Commits any pending transaction to the database
- conn.close: Closes the database connection

### 15.4.5 Local Functions

Name	In	Out	Exceptions
Plot	PlotTime: type Pandas Data Frame, Yaxis: type Pandas Data Frame, name: char array, color: char array	-	-

### 15.4.6 QPushButton Datatype Details

The object from the PyQt class which enables UI buttons.

#### **15.4.7 path-like Datatype Details**

The path-like-object is an object which contains the string of the path to the .db database file

#### **15.4.8 Connection Datatype Details**

An object representing the sqlite3 object

#### **15.4.9 Cursor Datatype Details**

An object which contains the functions which manipulate the database

#### **15.4.10 ProgrammingError Datatype Details**

A subclass of DatabaseError datatype.

## **16 MIS of Watch Case and Straps Module**

### **16.1 Module**

Watch\_Case\_Strap

### **16.2 Uses**

None.

### **16.3 Syntax**

#### **16.3.1 Exported Constants**

None.

#### **16.3.2 Exported Access Programs**

None.

### **16.4 Semantics**

Allow the device to be strapped onto the user.

#### **16.4.1 State Variables**

None.

#### **16.4.2 Environment Variables**

None.

#### **16.4.3 Assumptions**

Will not impact the functionality of any other components.

#### **16.4.4 Access Routine Semantics**

None.

#### **16.4.5 Local Functions**

None.

## References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.