

Software Requirements Specification for Mechatronics Engineering: subtitle describing software

Team #1, Back End Developers

Jessica Bae

Oliver Foote

Jonathan Hai

Anish Rangarajan

Nish Shah

Labeeb Zaker

October 4, 2022

Contents

1	Reference Material	iv
1.1	Table of Units	iv
1.2	Table of Symbols	v
1.3	Abbreviations and Acronyms	1
1.4	Mathematical Notation	1
2	Introduction	1
2.1	Purpose of Document	2
2.2	Scope of Requirements	3
2.3	Characteristics of Intended Reader	3
2.4	Organization of Document	3
3	General System Description	3
3.1	System Context	4
3.2	User Characteristics	5
3.3	System Constraints	5
4	Specific System Description	5
4.1	Problem Description	6
4.1.1	Terminology and Definitions	6
4.1.2	Physical System Description	6
4.1.3	Goal Statements	6
4.2	Solution Characteristics Specification	7
4.2.1	Assumptions	8
4.2.2	Theoretical Models	8
4.2.3	General Definitions	10
4.2.4	Data Definitions	11
4.2.5	Data Types	12
4.2.6	Instance Models	13
4.2.7	Input Data Constraints	14
4.2.8	Properties of a Correct Solution	14
5	Requirements	15
5.1	Functional Requirements	15
5.2	Nonfunctional Requirements	15
6	Likely Changes	16
7	Unlikely Changes	16
8	Traceability Matrices and Graphs	17

9 Development Plan	20
10 Values of Auxiliary Constants	20
11 Users and Stakeholders	20
12 Project Attributes	20
13 Normal Operation Overview	22
14 Undersired Event Handling	22
15 FSM and Dataflow Diagram	23

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
kg	mass	kilogram
s	time	second
°	angle	degree
rad	angle	radian
V	potential	volts
A	current	ampere
Ω	resistance	ohm ($\Omega = \text{V A}^{-1}$)
F	capacitance	farad
H	inductance	henry
N	force	newton ($\text{N} = \text{kg m s}^{-2}$)
Pa	pressure	pascals ($\text{Pa} = \text{Nm}^{-2}$)
Hz	frequency	hertz
J	energy	joule
W	power	watt ($\text{W} = \text{J s}^{-1}$)

Below are some derived units that do not use a specific SI symbol.

Derived unit	SI
area	m^2
volume	m^3
velocity	m s^{-1}
acceleration	m s^{-2}

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with kinematics and existing documentation of existing activity trackers. The symbols are listed in alphabetical order.

symbol	unit	description
f_c	Hz	clock frequency of microcontroller
$m_{tracker}$	g	mass of activity tracker (device)
a_r	m s^{-2}	accelerometer resolution
$A_{tracker}$	m^2	estimated area of activity tracker
$V_{battery}$	V	supplied voltage of battery

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
ACC	Acceleration
BAR	Barometer
CV	Controlled Variables
DD	Data Definition
EMA	Ecological Momentary Assessment
FSM	Finite State Machine
GD	General Definition
GS	Goal Statement
GYR	Gyrometer
IM	Instance Model
LC	Likely Change
LSS	Lumbar Spinal Stenosis
MV	Monitored Variables
OOP	Object Oriented Programming
PID	Proportional-integral-derivative
PS	Physical System Description
R	Requirement
SI	Système International d'Unités
SReS	School of Rehabilitation Sciences
SRS	Software Requirements Specification

1.4 Mathematical Notation

N/A

2 Introduction

Researchers at the School of Rehabilitation Sciences (SReS) at McMaster University are investigating treatment strategies for victims of spinal disorders and back pain. Currently, there is great interest in performing Ecological Momentary Assessment (EMA) of those who suffer from back pain and disorders in order to make treatment options more effective. EMA aims to study the thoughts, experiences, and behaviours of a patient's daily life by repeatedly collecting data in an

individual's normal environment, at or close to the time they carry out that behaviour.

Specifically, the type of EMA that the SReS is interested in is focused on analyzing the daily activities and symptoms of mostly-older adults with mobility and spinal issues. For example, they may be interested in performing EMA on a patient with spinal compression issues which prevent them from being able to walk long distances without experiencing pain or distress. They wish to track that patient's walking activity as they go about their daily life. If this patient stops walking or moving, they wish to prompt them with questions such as, "Did you stop moving? If so, was it due to pain? Describe your pain, describe which portion of the body it is in." The answers to these questions will be combined with activity monitoring data to form a better picture about the experience this patient has with their spinal and mobility issues.

In order to accomplish this, the researchers have attempted to use various smart-watch-esque activity tracking devices to track the activity of their patients, and have attempted to use apps and different pieces of software for them to prompt their patients with questions. However, they have been frustrated with very limited success, and are looking for a system which integrated precise and relevant activity tracking with user-friendly self-reporting symptom functionality.

The researchers require a method to perform EMA analysis in a manner which even older patients can use, integrated into one package which gathers relevant data about a patient's activities and allows them to easily report what is currently going on and how they feel. They also are looking for a way to access EMA data in various ways. This includes graphical representations of the data which are meaningful to researchers, along with the raw data itself. This data could be activity data, symptoms reporting data, both types of data collated together, and so on.

This introductory section of the document is intended to provide a reader with an overview of the purpose, overall scope, and organization of the document. As well as a description of the characteristics of the intended reader of this document.

2.1 Purpose of Document

This document is intended to describe the requirements of Team #1's system in a structured and collected manner. In other words, this is a document focused on what the system needs to do and the metrics and methods used to measure its performance. This groundwork is laid in a way to allow every individual who reads this document to understand the details essential to the requirements of this system.

Additional purposes of this document are to plan for the design stage of the project, foster effective communication between members of the team, and to establish reference material to which team members will be able to refer back to at later stages in the project should they need to acquire a higher-level view of the system.

2.2 Scope of Requirements

The scope requirements for a system which satisfies the goals of EMA must be defined properly. In the case of this project, the requirements must include a method of activity monitoring and a way to prompt users with questions relevant to EMA. In addition, this system will be used by older adults in Canada, specifically in a typical Canadian household environment, likely with a fixed daily routine. This system is not intended to be used by bedridden or sedentary users. If a user is unable to maintain a certain level of movement on a regular basis, the entire point of EMA is rendered moot. It is also necessary to exclude children from the sphere of users. This system must be designed for adults, aimed specifically towards older adults.

2.3 Characteristics of Intended Reader

While this document is intended to describe the requirements of the system in as abstract of a method as possible, it is unreasonable to expect a layman without any knowledge within the field of mechatronics to understand the contents of this document. Therefore, there are certain characteristics that a reader must have in order to fully grasp the scope of this SRS.

A reader should have a grasp of software development, electronics, mechanics and dynamics at a level equivalent to a 3rd-Year Mechatronics Undergraduate University student, as well as a basic understanding of human anatomy of which a high school level is adequate.

In addition, a reader should have a high-level understanding of Ecological Momentary Assessment. This does not have to be particularly in depth, but a reader should be able to identify the main objectives, practices, and difficulties of performing EMA. Introductory resources related to this can be found here [specify where].

2.4 Organization of Document

I'm leaving this section until we finalize section 5 and beyond. It won't take long. -JH

[This section provides a roadmap of the SRS document. It will help the reader orient themselves. It will provide direction that will help them select which sections they want to read, and in what order. This section will be similar between projects. —TPLT]

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

This system in theory is very simple. An array of sensors grabs info regarding the position, speed, orientation, etc. and uses that information to understand the current state of the user. This data is then used to generate prompts that the user will answer and all the collected data is compiled, processed and stored within the system. Finally, researchers will analyze the collected data and generate observations.

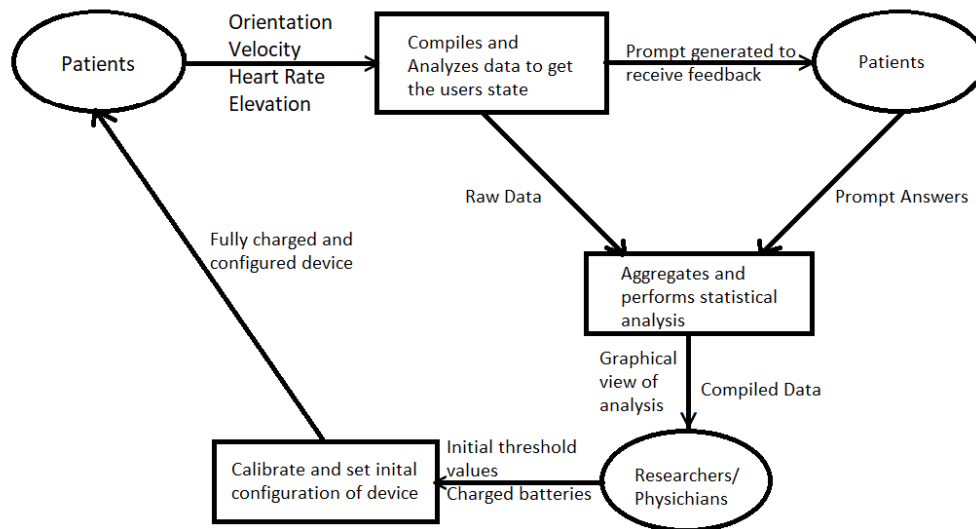


Figure 1: System Context

- **Patient Responsibilities:**
 - Patients are required to set up the device correctly so that their various attributes (Orientation, velocity, etc.) can be collected.
 - Patients are also required to answer the prompts given to them so that adequate data can be collected.
- **Researcher Responsibilities:**
 - Required to confirm that data is being collected.
 - Required to set up the initial thresholds and activities for tracking.
 - Offshoot the data.
 - Charge and calibrate the system.

3.2 User Characteristics

Patients should have a general understanding of using smart devices with touch screen interfaces. Patients will also be required to wear the device on their body so a good understanding of how to setup the device and secure it will be required. Finally the device will prompt users to answer certain prompts, thus knowledge of what to expect and how to respond is required.

Researchers will have the option of tweaking the configuration of the device, allowing them to manipulate certain activity thresholds. Thus a working knowledge of the system is required. Moreover, a basic understanding of graphs and statistical analysis will be required to understand the research data collected by the device.

3.3 System Constraints

The design will consist of both a hardware and a software system. The constraints for both systems are as follows:

- **Hardware:**

- The system must weigh less than 100g to facilitate being lightweight and comfortable.
- The system must have an average battery life of atleast 7 days to facilitate monitoring of the users.
- The dimensions of the system must be quite small with an area lower than 2500mm².

- **Software :**

- The system will have hard time constraints so as to store data and prompt users with low latency .
- To maintain privacy, the system will only store data for two weeks after which the data will be deleted.
- The system must prompt users based on activities and not based on time of day.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models. **[Add any project specific details that are relevant for the section overview. —TPLT]**

4.1 Problem Description

Mechatronics Engineering is intended to solve ... [What problem does your program solve? The description here should be in the problem space, not the solution space. —TPLT]

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

-

4.1.2 Physical System Description

[The purpose of this section is to clearly and unambiguously state the physical system that is to be modelled. Effective problem solving requires a logical and organized approach. The statements on the physical system to be studied should cover enough information to solve the problem. The physical description involves element identification, where elements are defined as independent and separable items of the physical system. Some example elements include acceleration due to gravity, the mass of an object, and the size and shape of an object. Each element should be identified and labelled, with their interesting properties specified clearly. The physical description can also include interactions of the elements, such as the following: i) the interactions between the elements and their physical environment; ii) the interactions between elements; and, iii) the initial or boundary conditions. —TPLT]

The physical system of Mechatronics Engineering, as shown in Figure ?, includes the following elements:

PS1:

PS2: ...

[A figure here makes sense for most SRS documents —TPLT]

4.1.3 Goal Statements

[The goal statements refine the “Problem Description” (Section 4.1). A goal is a functional objective the system under consideration should achieve. Goals provide criteria for sufficient completeness of a requirements specification and for requirements pertinence. Goals will be refined in Section “Instanced Models” (Section 4.2.6). Large and complex goals should be decomposed into smaller sub-goals. The goals are written abstractly, with a minimal amount of technical language. They should be understandable by non-domain experts. —TPLT]

Given the [inputs —TPLT], the goal statements are:

GS1: [One sentence description of the goal. There may be more than one. Each Goal should have a meaningful label. —TPLT]

4.2 Solution Characteristics Specification

[This section specifies the information in the solution domain of the system to be developed. This section is intended to express what is required in such a way that analysts and stakeholders get a clear picture, and the latter will accept it. The purpose of this section is to reduce the problem into one expressed in mathematical terms. Mathematical expertise is used to extract the essentials from the underlying physical description of the problem, and to collect and substantiate all physical data pertinent to the problem. —TPLT]

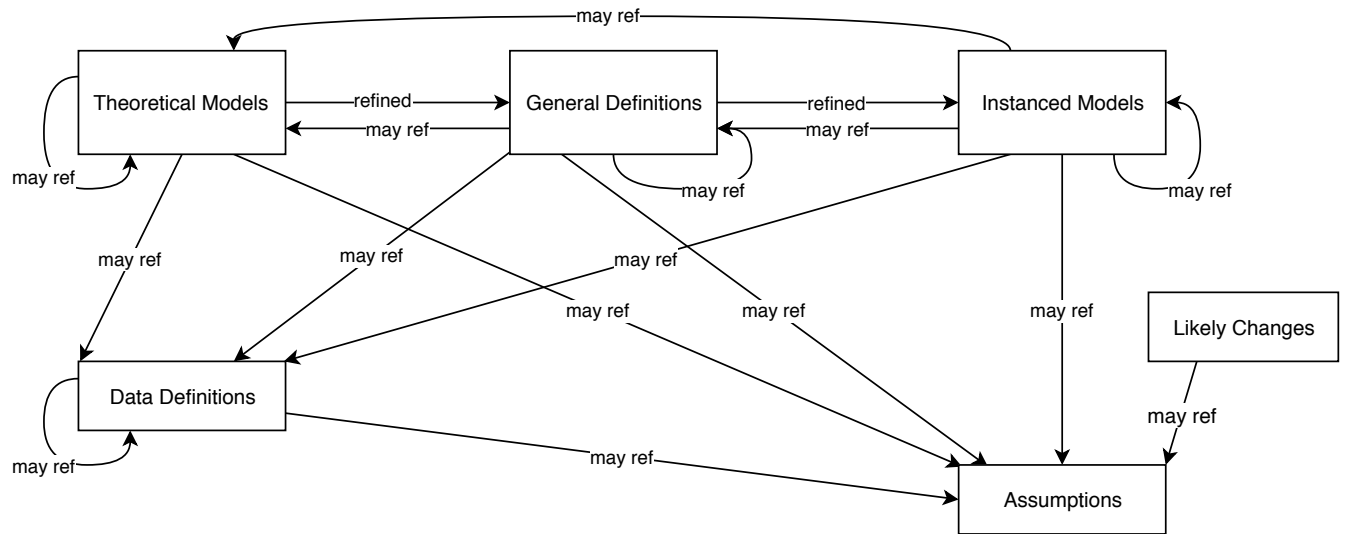
[This section presents the solution characteristics by successively refining models. It starts with the abstract/general Theoretical Models (TMs) and refines them to the concrete/specific Instance Models (IMs). If necessary there are intermediate refinements to General Definitions (GDs). All of these refinements can potentially use Assumptions (A) and Data Definitions (DD). TMs are refined to create new models, that are called GMs or IMs. DDs are not refined; they are just used. GDs and IMs are derived, or refined, from other models. DDs are not derived; they are just given. TMs are also just given, but they are refined, not used. If a potential DD includes a derivation, then that means it is refining other models, which would make it a GD or an IM. —TPLT]

[The above makes a distinction between “refined” and “used.” A model is refined to another model if it is changed by the refinement. When we change a general 3D equation to a 2D equation, we are making a refinement, by applying the assumption that the third dimension does not matter. If we use a definition, like the definition of density, we aren’t refining, or changing that definition, we are just using it. —TPLT]

[The same information can be a TM in one problem and a DD in another. It is about how the information is used. In one problem the definition of acceleration can be a TM, in another it would be a DD. —TPLT]

[There is repetition between the information given in the different chunks (TM, GDs etc) with other information in the document. For instance, the meaning of the symbols, the units etc are repeated. This is so that the chunks can stand on their own when being read by a reviewer/user. It also facilitates reuse of the models in a different context. —TPLT]

[The relationships between the parts of the document are show in the following figure. In this diagram “may ref” has the same role as “uses” above. The figure adds “Likely Changes,” which are able to reference (use) Assumptions. —TPLT]



The instance models that govern Mechatronics Engineering are presented in Subsection 4.2.6. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

[The assumptions are a refinement of the scope. The scope is general, where the assumptions are specific. All assumptions should be listed, even those that domain experts know so well that they are rarely (if ever) written down. —TPLT] [The document should not take for granted that the reader knows which assumptions have been made. In the case of unusual assumptions, it is recommended that the documentation either include, or point to, an explanation and justification for the assumption. —TPLT]

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

- A1: [Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. Each assumption should be atomic - that is, there should not be an explicit (or implicit) “and” in the text of an assumption. —TPLT]

4.2.2 Theoretical Models

[Theoretical models are sets of abstract mathematical equations or axioms for solving the problem described in Section “Physical System Description” (Section 4.1.2). Examples of theoretical models are physical laws, constitutive equations, relevant conversion factors, etc. —TPLT]

This section focuses on the general equations and laws that Mechatronics Engineering is based on. [Modify the examples below for your problem, and add additional models as appropriate. — TPLT]

RefName: T:COE

Label: Conservation of thermal energy

Equation: $-\nabla \cdot \mathbf{q} + g = \rho C \frac{\partial T}{\partial t}$

Description: The above equation gives the conservation of energy for transient heat transfer in a material of specific heat capacity C ($\text{J kg}^{-1} \text{ } ^\circ\text{C}^{-1}$) and density ρ (kg m^{-3}), where \mathbf{q} is the thermal flux vector (W m^{-2}), g is the volumetric heat generation (W m^{-3}), T is the temperature ($^\circ\text{C}$), t is time (s), and ∇ is the gradient operator. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A??). In general, the material properties (ρ and C) depend on temperature.

Notes: None.

Source: http://www.efunda.com/formulae/heat_transfer/conduction/overview_cond.cfm

Ref. By: GD??

Preconditions for T:COE: None

Derivation for T:COE: Not Applicable

[“Ref. By” is used repeatedly with the different types of information. This stands for Referenced By. It means that the models, definitions and assumptions listed reference the current model, definition or assumption. This information is given for traceability. Ref. By provides a pointer in the opposite direction to what we commonly do. You still need to have a reference in the other direction pointing to the current model, definition or assumption. As an example, if T1 is referenced by G2, that means that G2 will explicitly include a reference to T1. —TPLT]

4.2.3 General Definitions

[General Definitions (GDs) are a refinement of one or more TMs, and/or of other GDs. The GDs are less abstract than the TMs. Generally the reduction in abstraction is possible through invoking (using/referencing) Assumptions. For instance, the TM could be Newton’s Law of Cooling stated abstracting. The GD could take the general law and apply it to get a 1D equation. —TPLT]

This section collects the laws and equations that will be used in building the instance models.

[Some projects may not have any content for this section, but the section heading should be kept. —TPLT] [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	GD1
Label	Newton's law of cooling
SI Units	W m^{-2}
Equation	$q(t) = h\Delta T(t)$
Description	<p>Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings.</p> <p>$q(t)$ is the thermal flux (W m^{-2}).</p> <p>h is the heat transfer coefficient, assumed independent of T (A??) ($\text{W m}^{-2} \text{ } ^\circ\text{C}^{-1}$).</p> <p>$\Delta T(t) = T(t) - T_{\text{env}}(t)$ is the time-dependent thermal gradient between the environment and the object ($^\circ\text{C}$).</p>
Source	Citation here
Ref. By	DD1, DD??

Detailed derivation of simplified rate of change of temperature

[This may be necessary when the necessary information does not fit in the description field. —TPLT] [Derivations are important for justifying a given GD. You want it to be clear where the equation came from. —TPLT]

4.2.4 Data Definitions

[The Data Definitions are definitions of symbols and equations that are given for the problem. They are not derived; they are simply used by other models. For instance, if a problem depends on density, there may be a data definition for the equation defining density. The DDs are given information that you can use in your other modules. —TPLT]

[All Data Definitions should be used (referenced) by at least one other model. —TPLT]

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	DD1
Label	Heat flux out of coil
Symbol	q_C
SI Units	W m^{-2}
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil ($^{\circ}\text{C}$). T_W is the temperature of the water ($^{\circ}\text{C}$). The heat flux out of the coil, q_C (W m^{-2}), is found by assuming that Newton's Law of Cooling applies (A??). This law (GD1) is used on the surface of the coil, which has area A_C (m^2) and heat transfer coefficient h_C ($\text{W m}^{-2} ^{\circ}\text{C}^{-1}$). This equation assumes that the temperature of the coil is constant over time (A??) and that it does not vary along the length of the coil (A??).
Sources	Citation here
Ref. By	IM1

4.2.5 Data Types

[This section is optional. In many scientific computing programs it isn't necessary, since the inputs and output are straightforward types, like reals, integers, and sequences of reals and integers. However, for some problems it is very helpful to capture the type information. —TPLT]

[The data types are not derived; they are simply stated and used by other models. —TPLT]

[All data types must be used by at least one of the models. —TPLT]

[For the mathematical notation for expressing types, the recommendation is to use the notation of Hoffman and Strooper (1995). —TPLT]

This section collects and defines all the data types needed to document the models. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Type Name	Name for Type
Type Def	mathematical definition of the type
Description	description here
Sources	Citation here, if the type is borrowed from another source

4.2.6 Instance Models

[The motivation for this section is to reduce the problem defined in “Physical System Description” (Section 4.1.2) to one expressed in mathematical terms. The IMs are built by refining the TMs and/or GDs. This section should remain abstract. The SRS should specify the requirements without considering the implementation. —TPLT]

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goals [reference your goals —TPLT] are solved by [reference your instance models —TPLT]. [other details, with cross-references where appropriate. —TPLT] [Modify the examples below for your problem, and add additional models as appropriate. —TPLT]

Number	IM1
Label	Energy balance on water to find T_W
Input	$m_W, C_W, h_C, A_C, h_P, A_P, t_{\text{final}}, T_C, T_{\text{init}}, T_P(t)$ from IM?? The input is constrained so that $T_{\text{init}} \leq T_C$ (A??)
Output	$T_W(t), 0 \leq t \leq t_{\text{final}}$, such that $\frac{dT_W}{dt} = \frac{1}{\tau_W}[(T_C - T_W(t)) + \eta(T_P(t) - T_W(t))]$, $T_W(0) = T_P(0) = T_{\text{init}}$ (A??) and $T_P(t)$ from IM??
Description	T_W is the water temperature ($^{\circ}\text{C}$). T_P is the PCM temperature ($^{\circ}\text{C}$). T_C is the coil temperature ($^{\circ}\text{C}$). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^{\circ}\text{C}$, where 0°C and 100°C are the melting and boiling points of water, respectively (A??, A??).
Sources	Citation here
Ref. By	IM??

Derivation of ...

[The derivation shows how the IM is derived from the TMs/GDs. In cases where the derivation cannot be described under the Description field, it will be necessary to include this subsection. —TPLT]

4.2.7 Input Data Constraints

Table 1 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\min} \leq L \leq L_{\max}$	1.5 m	10%

(*) [you might need to add some notes or clarifications —TPLT]

Table 2: Specification Parameter Values

Var	Value
L_{\min}	0.1 m

4.2.8 Properties of a Correct Solution

A correct solution must exhibit [fill in the details —TPLT]. [These properties are in addition to the stated requirements. There is no need to repeat the requirements here. These additional properties may not exist for every problem. Examples include conservation laws (like conservation

Table 3: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A??)

of energy or mass) and known constraints on outputs, which are usually summarized in tabular form. A sample table is shown in Table 3 —TPLT]

[This section is not for test cases or techniques for verification and validation. Those topics will be addressed in the Verification and Validation plan. —TPLT]

5 Requirements

[The requirements refine the goal statement. They will make heavy use of references to the instance models. —TPLT]

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

- R1: [Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]
- R2: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]
- R3: [Calculation related requirements. —TPLT]
- R4: [Verification related requirements. —TPLT]
- R5: [Output related requirements. —TPLT]

[Every IM should map to at least one requirement, but not every requirement has to map to a corresponding IM. —TPLT]

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT] [The goal is for the nonfunctional requirements to be unambiguous, abstract and verifiable. This isn't easy to show succinctly, so a good strategy may be to give a "high

level" view of the requirement, but allow for the details to be covered in the Verification and Validation document. —TPLT] [An absolute requirement on a quality of the system is rarely needed. For instance, an accuracy of 0.0101 % is likely fine, even if the requirement is for 0.01 % accuracy. Therefore, the emphasis will often be more on describing how well the quality is achieved, through experimentation, and possibly theory, rather than meeting some bar that was defined a priori. —TPLT] [You do not need an entry for correctness in your NFRs. The purpose of the SRS is to record the requirements that need to be satisfied for correctness. Any statement of correctness would just be redundant. Rather than discuss correctness, you can characterize how far away from the correct (true) solution you are allowed to be. This is discussed under accuracy. —TPLT]

- NFR1: **Accuracy** [Characterize the accuracy by giving the context/use for the software. Maybe something like, "The accuracy of the computed solutions should meet the level needed for <engineering or scientific application>. The level of accuracy achieved by Mechatronics Engineering shall be described following the procedure given in Section X of the Verification and Validation Plan." A link to the VnV plan would be a nice extra. —TPLT]
- NFR2: **Usability** [Characterize the usability by giving the context/use for the software. You should likely reference the user characteristics section. The level of usability achieved by the software shall be described following the procedure given in Section X of the Verification and Validation Plan. A link to the VnV plan would be a nice extra. —TPLT]
- NFR3: **Maintainability** [The effort required to make any of the likely changes listed for Mechatronics Engineering should be less than FRACTION of the original development time. FRACTION is then a symbolic constant that can be defined at the end of the report. —TPLT]
- NFR4: **Portability** [This NFR is easier to write than the others. The systems that Mechatronics Engineering should run on should be listed here. When possible the specific versions of the potential operating environments should be given. To make the NFR verifiable a statement could be made that the tests from a given section of the VnV plan can be successfully run on all of the possible operating environments. —TPLT]
- Other NFRs that might be discussed include verifiability, understandability and reusability.

6 Likely Changes

- LC1: [Give the likely changes, with a reference to the related assumption (aref), as appropriate. —TPLT]

7 Unlikely Changes

- LC2: [Give the unlikely changes. The design can assume that the changes listed will not occur. —TPLT]

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an “X” may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —TPLT]

[The traceability matrix is not generally symmetric. If GD1 uses A1, that means that GD1's derivation or presentation requires invocation of A1. A1 does not use GD1. A1 is “used by” GD1. —TPLT]

[The traceability matrix is challenging to maintain manually. Please do your best. In the future tools (like Drasil) will make this much easier. —TPLT]

	T??	T??	T??	GD1	GD??	DD1	DD??	DD??	DD??	IM1	IM??	IM??	IM??
T??													
T??			X										
T??													
GD1													
GD??	X												
DD1				X									
DD??				X									
DD??													
DD??								X					
IM1					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure ?? shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure ?? shows the dependencies of instance models, requirements, and data constraints on each other.

	IM1	IM??	IM??	IM??	4.2.7	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD1		X																	
GD??			X	X	X	X													
DD1							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM1											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

9 Development Plan

[This section is optional. It is used to explain the plan for developing the software. In particular, this section gives a list of the order in which the requirements will be implemented. In the context of a course this is where you can indicate which requirements will be implemented as part of the course, and which will be “faked” as future work. This section can be organized as a prioritized list of requirements, or it could should the requirements that will be implemented for “phase 1”, “phase 2”, etc. —TPLT]

10 Values of Auxiliary Constants

[Show the values of the symbolic parameters introduced in the report. —TPLT]

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]

[The value of FRACTION, for the Maintainability NFR would be given here. —TPLT]

11 Users and Stakeholders

- Stakeholders
 - Dr. Luciana Macedo of School of Rehabilitation Science at McMaster University (Project Supervisor).
 - Dr. Spencer Smith (Project Supervisor).
- Users: The design has 2 different users:
 - **Researchers:** Primary end user who will use the data collected by the device for analysis and research.
 - **Patients :** Users part of the monitoring study who will be wearing the device, and whose activities will be collected for research and analysis.

12 Project Attributes

This is a comprehensive list of the various Open Issues, Costs and Risks involved with the project. This will be updated to reflect changes in the Design. Moreover this section also includes some stretch goals that may be accomplished depending on the pace and scale of the project.

- Open Issues:
 - A decision must be made whether the design will be worn on the wrist like a smart watch or on the waist like a belt.

- Sensor choices must be decided for optimal state detection of the user.
- Data transfer protocol must be decided, (Bluetooth, Ethernet, Wifi etc.)
- Cost analysis must be performed to see whether designing a custom PCB would be economical.

- Costs:

SNo.	Item.	Cost (\$CAD)
1	Arduino Nano	\$10
2	MPU6050 (Accelerometer/ Gyroscope)	\$12/unit
3	BMP-388 (Barometer)	\$2.5/unit
4	XD-58C (Heart Rate Sensor)	\$3
5	STM32f405rgt6 (Primary Control Chip)	\$8/unit
6	Electrical Components (Diodes, Resistors, Capacitors ,etc.)	\$20
7	Lipo Battery	\$30
8	OLED mini display	\$15
9	Structural Materials (MDF, PLA plastic, wood, etc.)	\$10
	Average Total Cost	\$110.5

- Risks:

- Battery life will degrade over time.
- Device is not sanitized between different monitoring periods.
- Device is improperly calibrated causing improper data collection.
- User error when answering prompts.
- Internal memory storage may become full if redundant data isnt regularly removed.

- Stretch goals: Detailed explanations of the following are provided in the Specification/Problem Statement Document.

- Detection of Anomalous behaviour may be added to track irregular movements.
- Geotagging location of the activities being tracked to achieve observations about high traffic activity areas.
- To optimize charging the device, a movement based charging system may be implemented.

13 Normal Operation Overview

Initially, the researcher will connect to the device using the provided software and either register a new participant or select a pre existing participant profile. Once a participant is selected, the researcher can set the thresholds, activities to be tracked and the monitoring period.

Once the device is handed off to the participant, an initial set up process is done, where the device is worn by the user and a test prompt is sent. Once this initial set up process is done, the user will simply carry on with their normal day to day and the device will track all registered activities using the onboard sensors. As the data is collected, it is logged into the internal memory and is stored based on the date and time of entry.

Finally after the monitoring period is finished, the device will go into an idle state until it is returned to the researcher. The researcher will then log in using their master credentials at which point the device will off load all data into the host software which will compile and present the data in a graphical manner.

14 Undersired Event Handling

- **Incorrect start up prompt response:** When the device is turned on by the user for the first time. A simple start up prompt is given just to check that system booted correctly ("Is this device on?"). If the user incorrectly enters their response, the system will simply perform a quick restart and ask the same question again.
- **Stagnant sensor data:** Should the device store the same sequence of data for an extended period of time, the system will perform a soft reboot of the sensor array to try to fix the issue. If the problem persists, An alert will be sent out to the researchers informing them of a system failure.
- **Improper prompt request:** Should the device generate a prompt whe no activity is being conducted. The user can simply report the prompt and the sytem will record an error, perform a soft reboot and continue normal operation.
- **Early battery failure:** The device's battery should normally last one week. If the charge goes below a predefined threshold before the monitoring period ends, activity tracking will cease and the user will be prompted to charge the device.

15 FSM and Dataflow Diagram

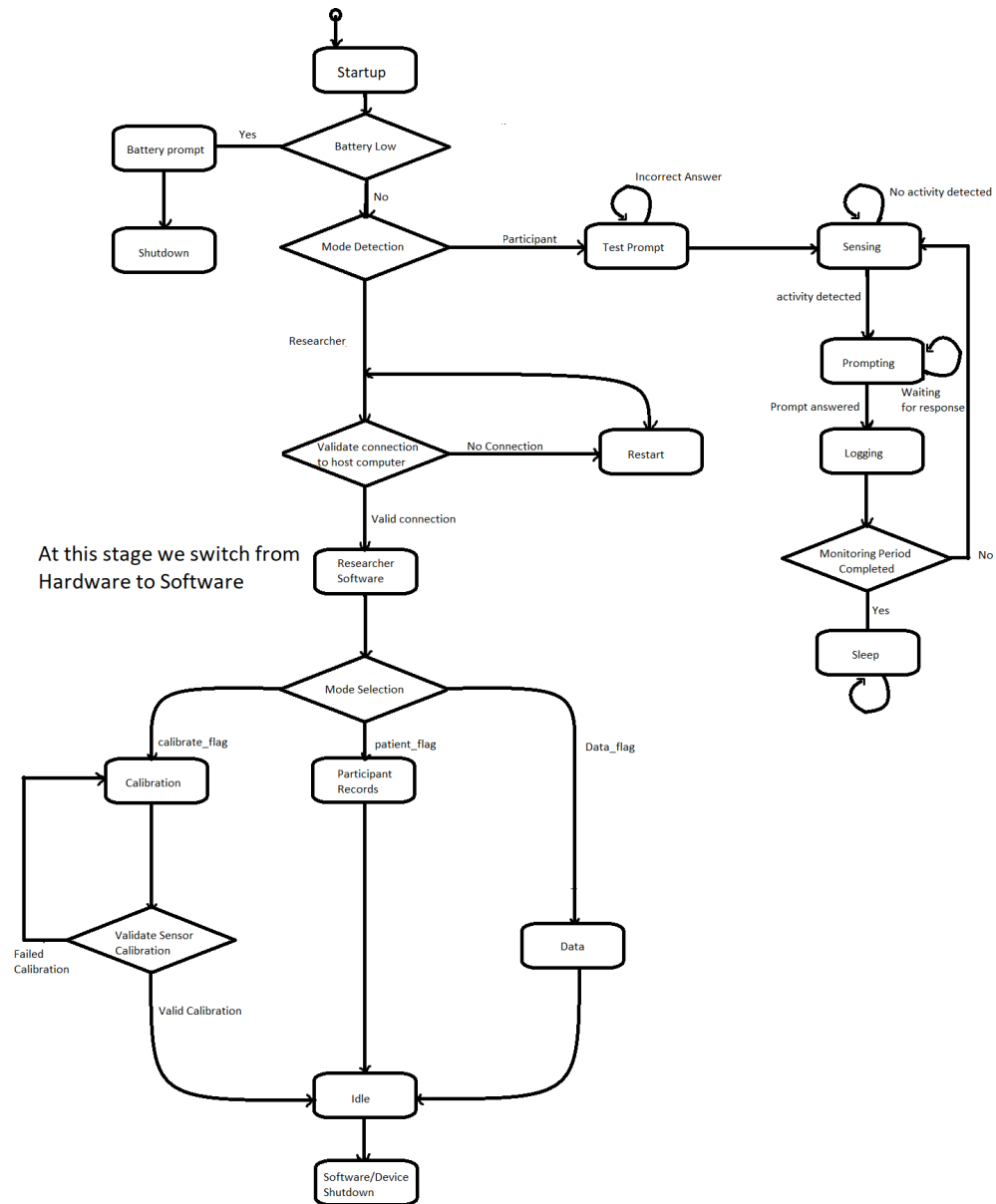


Figure 2: FSM

The following explains the states presented in the above Finite State Machine:

- **States:**

1. **Startup:** Initial state of the system when it turns on. Simply transitions after checking battery life.
2. **Battery Prompt:** Sends a prompt to the user informing them that the device needs to be charged.
3. **Shutdown:** This essentially powers of the device if the battery gets too low.
4. **Test Prompt:** Testing the prompting system by giving a simple question to the user, Is the device on?
5. **Sensing:** Reads data from all the sensor and checks whether an activity is detected.
6. **Prompting:** Sends a prompt to the user when a registered activity is found.
7. **Logging:** Logs prompt answers and sensor data in the internal memory.
8. **Sleep:** Temporarily puts the device to sleep until it is returned to the researchers.
9. **Restart:** Restarts the device without going through the startup sequence.
10. **Researcher Software:** Initial state of the software on the host computer. Consists of a simple welcome screen.
11. **Calibration:** Calibrates sensors, allows setting new thresholds and choosing what activities to track.
12. **Participant Records:** Allows creating and modifying participant information.
13. **Data:** Presents all data stored on the device in a graphical manner.
14. **Idle:** Idle state of the software. Awaits any user input.
15. **Software/Device shutdown:** When the software shuts down, the device also turns off.

References

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>.

[The following is not part of the template, just some things to consider when filling in the template.
—TPLT]

[Grammar, flow and L^AT_EX advice:

- For Mac users *.DS_Store should be in .gitignore
- L^AT_EX and formatting rules
 - Variables are italic, everything else not, includes subscripts ([link to document](#))
 - * [Conventions](#)
 - * Watch out for implied multiplication
 - Use BibTeX
 - Use cross-referencing
- Grammar and writing rules
 - Acronyms expanded on first usage (not just in table of acronyms)
 - “In order to” should be “to”

—TPLT]

[Advice on using the template:

- Difference between physical and software constraints
- Properties of a correct solution means *additional* properties, not a restating of the requirements (may be “not applicable” for your problem). If you have a table of output constraints, then these are properties of a correct solution.
- Assumptions have to be invoked somewhere
- “Referenced by” implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract

—TPLT]

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?