Software Requirements Specification for CXR: subtitle describing software

Team 27, Neuralyzers
Ayman Akhras
Nathan Luong
Patrick Zhou
Kelly Deng
Reza Jodeiri

October 7, 2024

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

[This template is intended for use by CAS 741. For CAS 741 the template should be used exactly as given, except the Reflection Appendix can be deleted. For the capstone course it is a source of ideas, but shouldn't be followed exactly. The exception is the reflection appendix. All capstone SRS documents should have a refelection appendix. —TPLT]

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
°C	temperature	centigrade
J	energy	joule
W	power	watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

[Only include the units that your SRS actually uses. —TPLT]

[Derived units, like newtons, pascal, etc, should show their derivation (the units they are derived from) if their constituent units are in the table of units (that is, if the units they are derived from are used in the document). For instance, the derivation of pascals as $\mathrm{Pa} = \mathrm{N\,m^{-2}}$ is shown if newtons and m are both in the table. The derivations of newtons would not be shown if kg and s are not both in the table. —TPLT]

[The symbol for units named after people use capital letters, but the name of the unit itself uses lower case. For instance, pascals use the symbol Pa, watts use the symbol W, teslas use the symbol T, newtons use the symbol N, etc. The one exception to this is degree Celsius. Details on writing metric units can be found on the NIST web-page. —TPLT]

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 the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
A_C	$\mathrm{m^2}$	coil surface area
A_{in}	$\mathrm{m^2}$	surface area over which heat is transferred in

[Use your problems actual symbols. The si package is a good idea to use for units. —TPLT]

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
CXR	[put an expanded version of your program name here (as appropriate) —TPLT]
TM	Theoretical Model

[Add any other abbreviations or acronyms that you add —TPLT]

1.4 Mathematical Notation

[This section is optional, but should be included for projects that make use of notation to convey mathematical information. For instance, if typographic conventions (like bold face font) are used to distinguish matrices, this should be stated here. If symbols are used to show mathematical operations, these should be summarized here. In some cases the easiest way to summarize the notation is to point to a text or other source that explains the notation. —TPLT]

[This section was added to the template because some students use very domain specific notation. This notation will not be readily understandable to people outside of your domain. It should be explained. —TPLT]

[This SRS template is based on Smith and Lai (2005); Smith et al. (2007); Smith and Koothoor (2016). It will get you started. You should not modify the section headings, without first discussing the change with the course instructor. Modification means you are not following the template, which loses some of the advantage of a template, especially standardization. Although the bits shown below do not include type information, you may need to add this information for your problem. If you are unsure, please can ask the instructor. —TPLT]

[Feel free to change the appearance of the report by modifying the LaTeX commands. —TPLT]

[This template document assumes that a single program is being documented. If you are documenting a family of models, you should start with a commonality analysis. A separate template is provided for this. For program families you should look at Smith (2006); Smith et al. (2017). Single family member programs are often programs based on a single physical model. General purpose tools are usually documented as a family. Families of physical models also come up. —TPLT]

[The SRS is not generally written, or read, sequentially. The SRS is a reference document. It is generally read in an ad hoc order, as the need arises. For writing an SRS, and for reading one for the first time, the suggested order of sections is:

- Goal Statement
- Instance Models
- Requirements
- Introduction
- Specific System Description

—TPLT]

[Guiding principles for the SRS document:

- Do not repeat the same information at the same abstraction level. If information is repeated, the repetition should be at a different abstraction level. For instance, there will be overlap between the scope section and the assumptions, but the scope section will not go into as much detail as the assumptions section.

—TPLT]

[The template description comments should be disabled before submitting this document for grading. —TPLT]

[You can borrow any wording from the text given in the template. It is part of the template, and not considered an instance of academic integrity. Of course, you need to cite the source of the template. —TPLT]

[When the documentation is done, it should be possible to trace back to the source of every piece of information. Some information will come from external sources, like terminology. Other information will be derived, like General Definitions. —TPLT]

[An SRS document should have the following qualities: unambiguous, consistent, complete, validatable, abstract and traceable. —TPLT]

[The overall goal of the SRS is that someone that meets the Characteristics of the Intended Reader (Section 2.3) can learn, understand and verify the captured domain knowledge. They should not have to trust the authors of the SRS on any statements. They should be able to independently verify/derive every statement made. —TPLT]

2 Introduction

[The introduction section is written to introduce the problem. It starts general and focuses on the problem domain. The general advice is to start with a paragraph or two that describes the problem, followed by a "roadmap" paragraph. A roadmap orients the reader by telling them what sub-sections to expect in the Introduction section. —TPLT]

2.1 Purpose of Document

[This section summarizes the purpose of the SRS document. It does not focus on the problem itself. The problem is described in the "Problem Description" section (Section 4.1). The purpose is for the document in the context of the project itself, not in the context of this course. Although the "purpose" of the document is to get a grade, you should not mention this. Instead, "fake it" as if this is a real project. The purpose section will be similar between projects. The purpose of the document is the purpose of the SRS, including communication, planning for the design stage, etc. —TPLT]

2.2 Scope of Requirements

[Modelling the real world requires simplification. The full complexity of the actual physics, chemistry, biology is too much for existing models, and for existing computational solution techniques. Rather than say what is in the scope, it is usually easier to say what is not. You can think of it as the scope is initially everything, and then it is constrained to create the actual scope. For instance, the problem can be restricted to 2 dimensions, or it can ignore the effect of temperature (or pressure) on the material properties, etc. —TPLT]

[The scope section is related to the assumptions section (Section 4.2.4). However, the scope and the assumptions are not at the same level of abstraction. The scope is at a high level. The focus is on the "big picture" assumptions. The assumptions section lists, and describes, all of the assumptions. —TPLT]

[The scope section is relevant for later determining typical values of inputs. The scope should make it clear what inputs are reasonable to expect. This is a distinction between scope and context (context is a later section). Scope affects the inputs while context affects how the software will be used. —TPLT]

2.3 Characteristics of Intended Reader

[This section summarizes the skills and knowledge of the readers of the SRS. It does NOT have the same purpose as the "User Characteristics" section (Section 3.2). The intended readers are the people that will read, review and maintain the SRS. They are the people that will conceivably design the software that is intended to meet the requirements. The user, on the other hand, is the person that uses the software that is built. They may never read this SRS document. Of course, the same person could be a "user" and an "intended reader." —TPLT]

[The intended reader characteristics should be written as unambiguously and as specifically as possible. Rather than say, the user should have an understanding of physics, say what kind of physics and at what level. For instance, is high school physics adequate, or should the reader have had a graduate course on advanced quantum mechanics? —TPLT]

2.4 Organization of Document

[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 project. —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. [This text can likely be borrowed verbatim. —TPLT]

[The purpose of this section is to provide general information about the system so the specific requirements in the next section will be easier to understand. The general system description section is designed to be changeable independent of changes to the functional requirements documented in the specific system description. The general system description provides a context for a family of related models. The general description can stay the same, while specific details are changed between family members. —TPLT]

3.1 System Context

[Your system context will include a figure that shows the abstract view of the software. Often in a scientific context, the program can be viewed abstractly following the design pattern of Inputs $\rightarrow$ Calculations $\rightarrow$ Outputs. The system context will therefore often follow this pattern. The user provides inputs, the system does the calculations, and then provides the outputs to the user. The figure should not show all of the inputs, just an abstract view of the main categories of inputs (like material properties, geometry, etc.). Likewise, the outputs should be presented from an abstract point of view. In some cases the diagram will show other external entities, besides the user. For instance, when the software product is a library, the user will be another software program, not an actual end user. If there are

system constraints that the software must work with external libraries, these libraries can also be shown on the System Context diagram. They should only be named with a specific library name if this is required by the system constraint. —TPLT]

Figure 1: System Context

[For each of the entities in the system context diagram its responsibilities should be listed. Whenever possible the system should check for data quality, but for some cases the user will need to assume that responsibility. The list of responsibilities should be about the inputs and outputs only, and they should be abstract. Details should not be presented here. However, the information should not be so abstract as to just say "inputs" and "outputs". A summarizing phrase can be used to characterize the inputs. For instance, saying "material properties" provides some information, but it stays away from the detail of listing every required properties. —TPLT]

- User Responsibilities:
  -
- CXR Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  -

[Identify in what context the software will typically be used. Is it for exploration? education? engineering work? scientific work?. Identify whether it will be used for mission-critical or safety-critical applications. —TPLT] [This additional context information is needed to determine how much effort should be devoted to the rationale section. If the application is safety-critical, the bar is higher. This is currently less structured, but analogous to, the idea to the Automotive Safety Integrity Levels (ASILs) that McSCert uses in their automotive hazard analyses. —TPLT]

[The —SS]

3.2 User Characteristics

[This section summarizes the knowledge/skills expected of the user. Measuring usability, which is often a required non-function requirement, requires knowledge of a typical user.

As mentioned above, the user is a different role from the "intended reader," as given in Section 2.3. As in Section 2.3, the user characteristics should be specific an unambiguous. For instance, "The end user of CXR should have an understanding of undergraduate Level 1 Calculus and Physics." —TPLT]

3.3 System Constraints

[System constraints differ from other type of requirements because they limit the developers' options in the system design and they identify how the eventual system must fit into the world. This is the only place in the SRS where design decisions can be specified. That is, the quality requirement for abstraction is relaxed here. However, system constraints should only be included if they are truly required. —TPLT]

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

CXR 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 section is expressed in words, not with equations. It provide the meaning of the different words and phrases used in the domain of the problem. The terminology is used to introduce concepts from the world outside of the mathematical model The terminology provides a real world connection to give the mathematical model meaning. —TPLT]

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 elements of the physical system do not have to correspond to an actual physical entity. They can be conceptual. This is particularly important when the documentation is for a numerical method. —TPLT]

The physical system of CXR, 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 4.2.9). 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

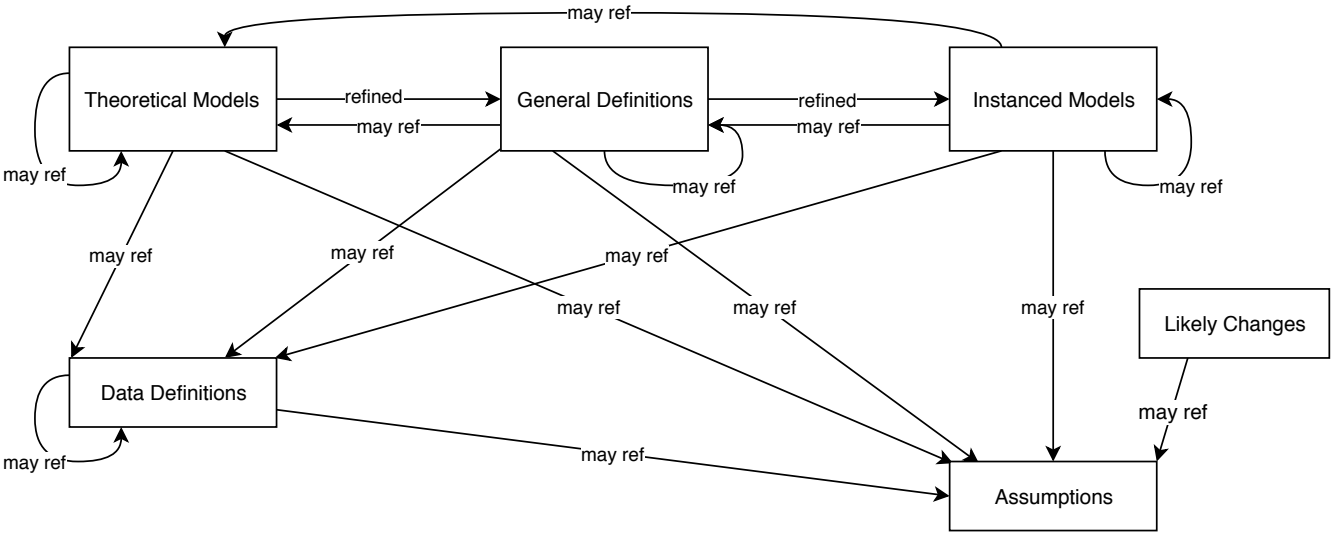
are not derived; they are just given. TMs are also just given, but they are refined, not used. If a potential DD includes a derivation, then that means it is refining other models, which would make it a GD or an IM. —TPLT]

[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 CXR are presented in Subsection 4.2.9. 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 Types

[This section is optional. Defining types can make the document easier to understand. —TPLT]

4.2.2 Scope Decisions

[This section is optional. —TPLT]

4.2.3 Modelling Decisions

[This section is optional. —TPLT]

4.2.4 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] [If it helps with the organization and understandability, the assumptions can be presented as sub sections. The following sub-sections are options: background theory assumptions, helper theory assumptions, generic theory assumptions, problem specific assumptions, and rationale assumptions —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 [TM], 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 TM, 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]

4.2.5 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]

[Optionally the theory section could be divided into subsections to provide more structure and improve understandability and reusability. Potential subsections include the following: Context theories, background theories, helper theories, generic theories, problem specific theories, final theories and rationale theories. —TPLT]

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

RefName: **TM: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 ($\mathrm{J\,kg^{-1}\,{}^{\circ}C^{-1}}$) and density ρ ($\mathrm{kg\,m^{-3}}$), where $\mathbf{q}$ is the thermal flux vector ($\mathrm{W\,m^{-2}}$), g is the volumetric heat generation ($\mathrm{W\,m^{-3}}$), T is the temperature (°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 TM:COE: None

Derivation for TM: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 TM1 is referenced by GD2, that means that GD2 will explicitly include a reference to TM1. —TPLT]

4.2.6 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	$\mathrm{W\,m^{-2}}$
Equation	$q(t) = h\Delta T(t)$
Description	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. $q(t)$ is the thermal flux ($\mathrm{W\,m^{-2}}$). h is the heat transfer coefficient, assumed independent of T (A**??**) ($\mathrm{W\,m^{-2}\,{}^{\circ}C^{-1}}$). $\Delta T(t) = T(t) - T_{\mathrm{env}}(t)$ is the time-dependent thermal gradient between the environment and the object (°C).
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]

4.2.7 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	$\mathrm{W\,m^{-2}}$
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil (°C). T_W is the temperature of the water (°C). The heat flux out of the coil, q_C ($\mathrm{W\,m^{-2}}$), 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 ($\mathrm{m^2}$) and heat transfer coefficient h_C ($\mathrm{W\,m^{-2}\,°C^{-1}}$). 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

4.2.8 Data Types

[This section is optional. In many scientific computing programs it isn't necessary, since the inputs and outpus 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

4.2.9 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 4.2.7 to replace the abstract symbols in the models identified in Sections 4.2.5 and 4.2.6.

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_{final}, T_C, T_{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 (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°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, $0 < T_W < 100^o$C, where 0^oC and 100^oC 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]

4.2.10 Input Data Constraints

Table 2 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 2 are listed in Table 4.

Table 2: 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 4: Specification Parameter Values

Var	Value
$L_{\min}$	0.1 m

4.2.11 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 of energy or mass) and known constraints on outputs, which are usually summarized in tabular form. A sample table is shown in Table 6 —TPLT]

Table 6: Output Variables

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

[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 Functional Requirements

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.

FR1	The system shall accept chest X-ray images as input from authorized users, including healthcare professionals and patients.
Rationale	To perform analysis, the system requires chest X-ray images to be uploaded by users in supported formats (DICOM, JPEG, PNG).
Verification	Test the upload functionality with different image formats and ensure the system can process the images without errors.
Priority	High
Traceability	R1

Table 8: Functional Requirement FR1

FR2	The system shall enable users to input additional patient symptoms, such as cough, chest pain, or fever.
Rationale	Including patient symptoms helps provide more context for the AI model to enhance disease detection accuracy. Our goal is to make the AI more useful by giving it more information to work with.
Verification	Verify that users can enter patient symptoms and that they are correctly linked to the associated X-ray images.
Priority	Medium
Traceability	R2

Table 9: Functional Requirement FR2

FR3	The system shall analyze chest X-ray images to detect the presence or absence of specific diseases with an accuracy of 85% or higher.
Rationale	One of the core functionalities of our system is its ability to detect diseases like pneumonia from X-rays. This ensures the AI model is effective in clinical settings.
Verification	Evaluate the accuracy of the disease detection model against a test dataset and confirm it meets or exceeds the 85% accuracy threshold.
Priority	High
Traceability	R3

Table 10: Functional Requirement FR3

FR4	The system shall indicate whether a patient's condition has improved, worsened, or remained stable between scans.
Rationale	This feature helps clinicians monitor disease progression or regression, which is vital for evaluating the effectiveness of treatments over time.
Verification	Verify that the system provides a status of the patient's condition after comparing multiple scans.
Priority	High
Traceability	R4

Table 11: Functional Requirement FR4

FR5	The system shall generate visual aids by highlighting affected areas on the chest X-ray images.
Rationale	By highlighting affected regions, our system makes it easier for clinicians to identify the areas of concern in a chest X-ray. This improves their ability to act quickly.
Verification	Ensure that the system overlays the appropriate markers on the X-ray images to highlight abnormal areas.
Priority	Medium
Traceability	R5

Table 12: Functional Requirement FR5

FR6	The system shall produce a structured, human-readable report summarizing key findings, disease detection results, and progression status.
Rationale	A comprehensive report makes it easier for clinicians to interpret the analysis results, especially in fast-paced environments where time is critical.
Verification	Verify that the system generates a report containing all necessary details in an organized format.
Priority	High
Traceability	R6

Table 13: Functional Requirement FR6

FR7	The system shall store patient data, including images and reports, in a secure database for future reference.
Rationale	Keeping patient records secure and accessible for future consultations or research is important for patient care continuity.
Verification	Verify that the system correctly stores and retrieves patient data from the database.
Priority	Medium
Traceability	R7

Table 14: Functional Requirement FR7

FR8	The system shall provide alerts to clinicians if significant changes in a patient's condition are detected between scans.
Rationale	Automated alerts can save time and ensure that urgent cases are handled quickly, improving patient outcomes.
Verification	Test that alerts are generated when the system detects a significant change in disease severity or condition.
Priority	Low
Traceability	R8

Table 15: Functional Requirement FR8

FR9	The system shall allow healthcare professionals to adjust treatment plans based on the X-ray analysis results.
Rationale	Allowing clinicians to modify treatment plans based on X-ray results helps to tailor patient care more effectively. This ensures that the system supports decision-making in real-time clinical settings.
Verification	Verify that healthcare professionals can modify treatment plans directly in the system, and that these changes are linked to the corresponding X-ray analysis results.
Priority	High
Traceability	R9

Table 16: Functional Requirement FR9

FR10	The system shall allow integration with electronic health record (EHR) systems to import and export patient data.
Rationale	Integrating with existing EHR systems improves workflow efficiency and reduces the need for manual data entry, making the system more useful in a clinical setting.
Verification	Test the system's ability to connect with standard EHR systems and exchange data using appropriate protocols.
Priority	Low
Traceability	R10

Table 17: Functional Requirement FR10

FR11	The system shall display confidence levels for disease detection and progression analysis results through an intuitive user interface that requires minimal training to operate.
Rationale	Providing confidence levels helps clinicians assess the reliability of the results.
Verification	Check that the system displays a confidence score alongside each analysis result.
Priority	High
Traceability	R11

Table 18: Functional Requirement FR11

FR12	The system shall allow patients to upload their own chest X-ray images for self-diagnosis purposes, subject to appropriate disclaimers.
Rationale	Allowing patients to upload their own images increases engagement in their healthcare journey, though it must be made clear that professional interpretation is still necessary.
Verification	Ensure that patients can upload images and that the system displays appropriate disclaimers before analysis.
Priority	Low
Traceability	R12

Table 19: Functional Requirement FR12

FR13	The system shall support multiple user roles with appropriate access levels (e.g., physician, radiologist, patient).
Rationale	Different users need access to different features and data, so assigning roles ensures that each user can only perform actions they are authorized to.
Verification	Verify that users are assigned roles and that their access is restricted to the appropriate features based on their role.
Priority	High
Traceability	R13

Table 20: Functional Requirement FR13

FR14	The system shall create a new copy of a patient's X-ray before running the AI model for analysis.
Rationale	Creating a new copy ensures that the original X-ray remains unchanged for future reference, while the AI model works on the new version for tagging and analysis.
Verification	Verify that the system creates a new X-ray copy, preserving the original file while the AI model runs on the duplicate.
Priority	High
Traceability	R14

Table 21: Functional Requirement FR14

FR15	The system shall support additional medical imaging modalities, such as CT scans and MRIs, for comprehensive analysis.
Rationale	Supporting multiple imaging types expands the system's utility in various clinical scenarios.
Verification	Test the system's ability to accept and analyze different image types (e.g., CT, MRI) without errors and provide accurate results.
Priority	Low
Traceability	R15

Table 22: Functional Requirement FR15

FR16	The system shall support regular updates to the AI model to incorporate new data and improve accuracy over time.
Rationale	Continuous learning enhances the model's performance and keeps it up-to-date with the latest medical knowledge.
Verification	Ensure that the system can accept and integrate updated models without disrupting service.
Priority	Medium
Traceability	R16

Table 23: Functional Requirement FR16

6 Non-Functional Requirements

6.1 Look and Feel Requirements

LF1	The system shall have a user interface consistent with standard medical imaging software, using familiar layouts and terminology.
Rationale	Radiologists and healthcare professionals are accustomed to specific interface designs; maintaining consistency reduces the learning curve.
Fit Criterion	At least 90% of radiologists surveyed find the interface intuitive and comparable to existing medical imaging software.
Traceability	FR1, FR5

Table 24: Non-functional Requirement LF1

LF2	The system shall use color schemes and fonts that minimize eye strain during prolonged use.
Rationale	Medical professionals often work long hours; a comfortable visual interface helps prevent fatigue.
Fit Criterion	The interface adheres to ergonomic standards (e.g., ISO 9241-3) for visual displays.
Traceability	FR1

Table 25: Non-functional Requirement LF2

6.2 Usability and Humanity Requirements

UH1	At least 90% of healthcare professionals shall be able to perform common tasks (e.g., uploading images, viewing analysis results) without assistance after a 30-minute training session.
Rationale	Busy professionals need to use the system efficiently without extensive training.
Fit Criterion	Usability testing shows that users can complete key tasks unassisted after initial training.
Traceability	FR1, FR2, FR5

Table 26: Non-functional Requirement UH1

UH2	The system shall provide context-sensitive help and tooltips for all interface elements to assist users in understanding functionalities.
Rationale	Immediate assistance reduces errors and enhances the user experience.
Fit Criterion	All interactive elements display helpful tooltips when hovered over, and a help section is accessible from all screens.
Traceability	FR2, FR6

Table 27: Non-functional Requirement UH2

6.3 Performance Requirements

PR1	The system shall process and analyze a standard chest X-ray image within 30 seconds.
Rationale	Quick analysis is critical for timely diagnosis and treatment decisions in clinical settings.
Fit Criterion	Performance tests demonstrate that 95% of standard images are processed within the 30-second threshold.
Traceability	FR3, FR4, FR5

Table 28: Non-functional Requirement PR1

PR2	The system shall have an uptime of at least 99.9%, ensuring high availability for users.
Rationale	Continuous access is essential in healthcare environments to prevent delays in patient care.
Fit Criterion	System logs indicate an uptime of 99.9% over a 6-month monitoring period.
Traceability	FR5, FR7

Table 29: Non-functional Requirement PR2

PR3	The system shall support concurrent processing of up to 20 images without significant performance degradation.
Rationale	Radiologists may need to process multiple images simultaneously, especially in busy clinics or hospitals.
Fit Criterion	System performance tests confirm that processing time per image does not exceed 45 seconds when 20 images are processed concurrently.
Traceability	FR5, FR7

Table 30: Non-functional Requirement PR3

6.4 Operational and Environmental Requirements

OE1	The system shall integrate with existing hospital Picture Archiving and Communication Systems (PACS) using standard protocols like DICOM.
Rationale	Seamless integration with PACS improves workflow efficiency and reduces manual data handling.
Fit Criterion	The system successfully exchanges data with at least three major PACS solutions in a test environment.
Traceability	FR7, FR10

Table 31: Non-functional Requirement OE1

OE2	The system shall operate effectively in network environments with latency up to 200ms and packet loss up to 1%.
Rationale	Hospital networks may have variable conditions; the system must remain functional under less-than-ideal circumstances.
Fit Criterion	System testing under simulated network conditions confirms acceptable performance and responsiveness.
Traceability	FR4, FR6

Table 32: Non-functional Requirement OE2

6.5 Security and Privacy Requirements

SR1	The system shall encrypt all patient data, including images and reports, both in transit and at rest, using AES-256 encryption.
Rationale	Protecting sensitive patient information is critical to comply with legal requirements and maintain trust.
Fit Criterion	Security audits confirm that all data storage and transmissions use AES-256 encryption.
Traceability	FR7, FR12

Table 33: Non-functional Requirement SR1

SR2	The system shall implement role-based access control, ensuring users can only access functionalities and data appropriate to their roles (e.g., radiologist, radiographer, administrator).
Rationale	Restricting access based on roles prevents unauthorized data access and potential data breaches.
Fit Criterion	Access control tests verify that users cannot access data or functions outside their permissions.
Traceability	FR13, FR26

Table 34: Non-functional Requirement SR2

6.6 Maintainability and Support Requirements

MS1	The system shall be developed using modular architecture with well-documented code to facilitate future maintenance and upgrades.
Rationale	A modular design simplifies debugging, testing, and integration of new features, reducing maintenance costs.
Fit Criterion	Code reviews confirm adherence to coding standards and modular design principles; documentation covers all modules.
Traceability	FR2, FR10

Table 35: Non-functional Requirement MS1

MS2	The system shall include automated testing suites covering at least 80% of the codebase to ensure reliability during updates.
Rationale	High test coverage helps detect issues early and maintains system stability over time.
Fit Criterion	Test coverage reports indicate that automated tests cover 80% or more of the code.
Traceability	FR7, FR14

Table 36: Non-functional Requirement MS2

6.7 Cultural Requirements

CR1	The system shall support both English and French languages for the user interface and reports.
Rationale	Supporting Canada's official languages ensures accessibility for all users and meets legal requirements in certain provinces.
Fit Criterion	Users can select English or French as the system language, with all interface elements and reports displayed accordingly.
Traceability	FR6, FR12

Table 37: Non-functional Requirement CR1

6.8 Legal Requirements

LR1	The system shall comply with all applicable healthcare data protection laws, including HIPAA in the United States and PIPEDA in Canada.
Rationale	Legal compliance is mandatory to protect patient rights and avoid legal repercussions.
Fit Criterion	Compliance audits verify adherence to relevant data protection laws and regulations.
Traceability	FR12, FR13

Table 38: Non-functional Requirement LR1

LR2	The system shall adhere to the ISO 13485 standard for medical device software development.
Rationale	Following recognized standards ensures quality and safety in medical software.
Fit Criterion	Certification or compliance reports demonstrate adherence to ISO 13485.
Traceability	FR7, FR10

Table 39: Non-functional Requirement LR2

6.9 Health and Safety Requirements

HS1	The system shall ensure that all AI-generated diagnoses are reviewed and confirmed by a qualified radiologist before being used in patient care decisions.
Rationale	To prevent misdiagnoses and ensure patient safety by involving human oversight.
Fit Criterion	System workflow requires radiologist validation before finalizing reports; logs confirm this process.
Traceability	FR3, FR6

Table 40: Non-functional Requirement HS1

HS2	The system shall provide clear disclaimers indicating that AI analysis is a diagnostic aid and not a replacement for professional medical judgment.
Rationale	Users must understand the limitations of AI to prevent overreliance and potential errors.
Fit Criterion	Disclaimers are prominently displayed on analysis results and require user acknowledgment upon first use.
Traceability	FR12, FR6

Table 41: Non-functional Requirement HS2

6.10 Rationale

[Provide a rationale for the decisions made in the documentation. Rationale should be provided for scope decisions, modelling decisions, assumptions and typical values. —TPLT]

7 Phase in Plan

7.1 High Priority Functional Requirements

Req ID	Completion Date and Rationale
FR1	**Completion Date:** November 13, 2023 **Rationale:** FR1 is essential because accepting chest X-ray images is the foundation of our system. We need this functionality first to enable all other features to build upon it.
FR3	**Completion Date:** November 13, 2023 **Rationale:** FR3 is critical as it provides the core capability of analyzing X-rays for disease detection. Implementing this early allows us to validate the system's primary purpose and proceed with dependent features.
FR4	**Completion Date:** November 13, 2023 **Rationale:** FR4 is important for monitoring patient progress. By completing it alongside FR3, we ensure clinicians can assess changes over time from the initial release.
FR6	**Completion Date:** November 13, 2023 **Rationale:** FR6 allows us to present analysis results in a usable format for clinicians. Having this ready by November 13 is vital for user acceptance and practical use of the system.
FR11	**Completion Date:** November 13, 2023 **Rationale:** FR11 improves user trust by displaying confidence levels. An intuitive UI is essential for user adoption;therefore, we aim to have it completed by November 13.
FR13	**Completion Date:** November 13, 2023 **Rationale:** FR13 is necessary for security and compliance, ensuring proper access control from the start. Implementing it early helps us protect sensitive data effectively.
FR14	**Completion Date:** November 13, 2023 **Rationale:** FR14 safeguards the integrity of original X-ray images. Completing this by November 13 is important for legal considerations.

7.2 Medium Priority Functional Requirements

Req ID	Completion Date and Rationale
FR2	**Completion Date:** February 5, 2024 **Rationale:** FR2 adds the ability to input additional patient symptoms, enhancing analysis accuracy. We scheduled it after the core functionalities to focus on essential features first.
FR5	**Completion Date:** February 5, 2024 **Rationale:** FR5 provides visual aids, improving usability. Implementing it after the main analysis features allows us to refine the user experience based on initial feedback.
FR7	**Completion Date:** February 5, 2024 **Rationale:** FR7 involves secure data storage for future reference. We plan to develop this after the initial deployment to ensure data handling aligns with user needs and compliance requirements.
FR16	**Completion Date:** February 5, 2024 **Rationale:** FR16 enables regular updates to the AI model, allowing continuous improvement. Scheduling it for February gives us time to stabilize the initial model and plan for updates without disrupting service.

Table 42: High and Medium Priority Functional Requirements

7.3 Low Priority Functional Requirements

Req ID	Completion Date and Rationale
FR8	**Completion Date:** March 18, 2024 **Rationale:** FR8 offers alerts for significant condition changes. It's useful but not essential for initial deployment, so we scheduled it after higher-priority tasks.
FR10	**Completion Date:** March 18, 2024 **Rationale:** FR10 involves integrating with EHR systems, which adds complexity. We plan to address this later to focus on core functionalities first.
FR12	**Completion Date:** March 18, 2024 **Rationale:** FR12 allows patient self-upload of images. Given legal considerations and our initial focus on clinicians, we scheduled it for later development.
FR15	**Completion Date:** March 18, 2024 **Rationale:** FR15 expands support to other imaging modalities. We aim to perfect chest X-ray analysis before adding CT and MRI support, so we scheduled this for the last phase.

Table 43: Low Priority Functional Requirements

7.4 High Priority Non-Functional Requirements

Req ID	Completion Date and Rationale
PR1	**Completion Date:** November 13, 2023 **Rationale:** PR1 ensures the system processes images quickly, which is crucial for usability in clinical settings.
HS1	**Completion Date:** November 13, 2023 **Rationale:** HS1 mandates that AI diagnoses are reviewed by a radiologist, ensuring patient safety. This oversight is necessary from day one.
LF1	**Completion Date:** November 13, 2023 **Rationale:** LF1 requires an interface consistent with standard medical software, aiding user adoption. We want users to feel comfortable using the system immediately.
PR2	**Completion Date:** November 13, 2023 **Rationale:** PR2 aims for high system uptime. Reliability is critical, so we must ensure the system is stable from the initial launch.
SR2	**Completion Date:** November 13, 2023 **Rationale:** SR2 implements role-based access control, necessary for security and compliance. We need to control data access from the beginning.
MS1	**Completion Date:** November 13, 2023 **Rationale:** MS1 involves modular architecture and documentation, facilitating future maintenance.
LR1	**Completion Date:** November 13, 2023 **Rationale:** LR1 ensures compliance with data protection laws, which is legally required. We must adhere to regulations from the outset to avoid legal issues.
LR2	**Completion Date:** November 13, 2023 **Rationale:** LR2 requires adherence to ISO 13485, ensuring quality and safety. Meeting industry standards builds trust and facilitates future certifications.
HS2	**Completion Date:** November 13, 2023 **Rationale:** HS2 mandates clear disclaimers about AI limitations. Providing this information from the start is important for user awareness and legal protection.

7.5 Medium and Low Priority Non-Functional Requirements

Req ID	Completion Date and Rationale
LF2	**Completion Date:** February 5, 2024 **Rationale:** LF2 improves the user interface to minimize eye strain. While important for user comfort, it can be refined after the main features are operational.
UH2	**Completion Date:** February 5, 2024 **Rationale:** UH2 adds context-sensitive help and tooltips. Enhancing usability is beneficial, but we plan to implement it after gathering initial user feedback.
PR3	**Completion Date:** February 5, 2024 **Rationale:** PR3 enhances performance by supporting concurrent image processing. Scheduling this for later allows us to optimize the system after core functionalities are stable.
MS2	**Completion Date:** February 5, 2024 **Rationale:** MS2 involves automated testing for reliability. Implementing it after initial development helps us maintain code quality as the project grows.
OE1	**Completion Date:** March 18, 2024 **Rationale:** OE1 integrates with hospital PACS systems. Given the complexity, we plan to tackle this once the system's core features are solidified.
OE2	**Completion Date:** March 18, 2024 **Rationale:** OE2 ensures operation under variable network conditions. This enhancement is scheduled for later, focusing first on standard network environments.
CR1	**Completion Date:** March 18, 2024 **Rationale:** CR1 adds multilingual support. We aim to include additional languages after ensuring the system functions well in the primary language.

Table 44: Medium and Low Priority Non-Functional Requirements

8 Likelyhood of Changes

9 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 45 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 46 shows the dependencies of instance models, requirements, and data constraints on each other. Table 47 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]

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

Table 45: 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

	IM1	IM??	IM??	IM??	4.2.10	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R??			X	X			
R??		X					
R??		X					

Table 46: 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??
TM??	X																		
TM??																			
TM??																			
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 47: Traceability Matrix Showing the Connections Between Assumptions and Other Items

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.

10 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]

11 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]

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.

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2):404–418, April 2016. ISSN 1738-5733. doi: http://dx.doi.org/10.1016/j.net.2015.11.008. URL http://www.sciencedirect.com/science/article/pii/S1738573315002582.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

[The following is not part of the template, just some things to consider when filing in the template. —TPLT]
[Grammar, flow and LaTeXadvice:

- For Mac users `*.DS_Store` should be in `.gitignore`
- LaTeX 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

[Not required for CAS 741 —SS]

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

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

5. 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.

6. 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?