Software Requirements Specification for 4TB6 - Mechatronics Capstone: Smart Bike Lock

Abi Nevo
Elsa Bassi
Steffi Ralph
Abdul Iqbal
Stephen De Jong
Anthony Shenouda

October 5, 2022

Contents

Revision History

Date	Version	Name	Notes
02-10-22	1.0	Elsa	Drafted first draft
03-10-22	1.1	Stephen	Drafted 1 & 5, and formatting updates
04-10-22	1.2	Steffi	Drafted Appendix

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
J	energy	joule
P or W	power	watt ($W = J\,s^{-1}$)
A or I	current	ampere
Ω or R	resistance	ohm
V	voltage	volt
N	force	newton
NM	torque	newton meter

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
B_L	hour	battery lifein hours
B_C	mAH	battery capacity in amp hours
L_C	A/actuation	current drawn per motor actuation
A_{num}	actuation/charge	number of actuations per charge

1.3 Abbreviations and Acronyms

symbol	description
SRS	Software Requirements Specification

2 Introduction

The purpose of the Smart Lock project is to design and build a product that will provide bicycle users with a safer, easier, and more accessible way to secure their bike through their smartphone. Additionally, it will provide users with a GPS feature to locate the lock in case of bike theft or misplacement. It will consist of a physical lock that mounts to a bike and a smartphone application that will function as the user interface through which the lock can be engaged and disengaged wirelessly, as well as located. The project will provide an engineering solution using wireless communication, mechanical design, and smartphone application development. More broadly, it seeks to encourage members of society to pursue biking, in both a transportation and recreational capacity, improving the health of society's citizens and its environment.

2.1 Purpose of Document

The Requirements Documentation seeks to provide a complete overview of the requirements of the Smart Lock system so as to define the project. The assumptions, inputs and outputs will also be defined in order to outline the problem. Several models will be developed and lastly, likely and unlikely changes will be described. This will communicate a unified and documented plan for the project that can be used after the design stage to verify its functionality and provide an important benchmark.

2.2 Scope of Requirements

The Smart Lock project will build off existing designs to tie together principles of wireless communication, GPS, a mechanical locking mechanism, electrical actuation, and smartphone application development. The system will provide all functions necessary for effective and secure locking and location.

The Smart Lock project will be designed to perform the following functions for the user.

1. Wireless communication from the smartphone to the lock and vice versa.

2. Display of lock and battery status information on the app.

3. Display of GPS location on the app.

4. Store and use a replaceable battery.

5. House a mechanical lock frame.

6. Perform electrical engagement/disengagement of a locking mechanism.

7. Be waterproof.

Not included in the project scope will be Bluetooth wireless connective capabilities, a fully autonomous locking mechanism (one that will both open/close and engage/disengage the lock wirelessly) and a rechargeable battery. We will also be ignoring extreme weather and temperature conditions and their impact on the material properties.

2.3 Characteristics of Intended Reader

The reader will have basic knowledge of how bikes are secured in public settings, namely using mechanical locks and keys or combination locks. They will understand why bike locking is necessary and what it means for a bike to be vulnerable to theft, i.e. that all main parts of the bike must be secured including the frame and wheels. They will also have basic knowledge of GPS and wireless communications that can be accessed through smartphone applications. Lastly, the reader will have a high-school level understanding of kinematic and electronics-related physics and math such that they can grasp the mechanical and electrical functionality of the system.

2.4 Organization of Document

The document will follow the base template provided by the Capstone 4TB6 course GitLab. It will be organized with an increasing order of specificity, beginning with the General System Description that will include an overview of the Smart Lock project. The Specific System Description will make up the bulk of the document, describing the project in greater detail with its assumptions, inputs and outputs, goals and models. Finally, since the context of the project is previously stated, the Functional and Non-Functional Requirements will be outlined, as well as any foreseen Likely and Unlikely Changes to the project. Lastly, the Traceability Matrix is included to define the relationships between various relevant formulas, sections and information in the document and how they be used to build the project.

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
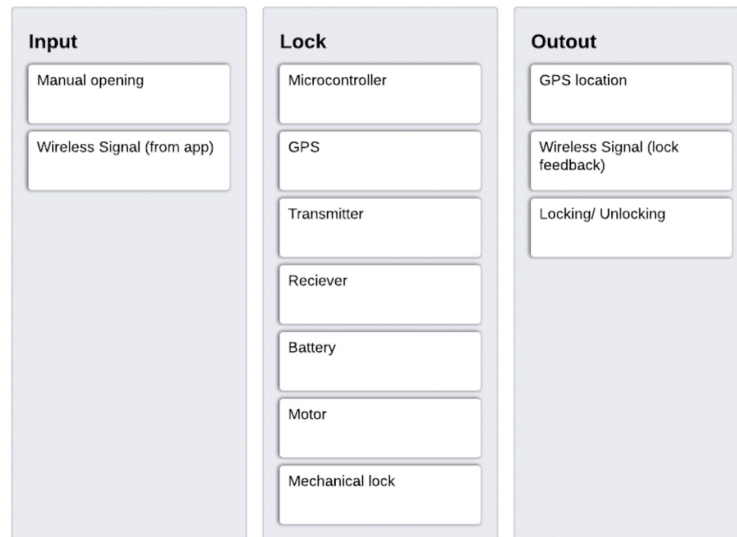
3.1 System Context

Figure 1: The System Context: Shows the inputs, the entities that manipulate them, and the outputs.

- User Responsibilities:

    –

- 4TB6 - Mechatronics Capstone Responsibilities:

    – Detect data type mismatch, such as a string of characters instead of a floating point number

    –

3.2 User Characteristics

3.3　System Constraints

The system has the following constraints.

- GPS to accurately locate the bike within a tolerance of 50 meters.

  - Justification: Complex software like Google Maps ensures an accuracy of around 20 meters. Considering users only require the general location and our software will not be as complex as Google's, the team added this constraint.

- Total mass of lock under 3 kg.

  - Justification: Average weight of a bike lock is just under 3 pounds. To ensure the lock was not significantly heavier the team added this constraint.

- App storage under 50 megabytes.

  - Justification: A small mobile app should not take significant space on the user's phone. Similar GPS applications are around 50 megabytes.

- The budget for testing, designing, and equipment is $750.

  - Justification: Smart Lock has a limited budget to work within to encourage efficient use of materials and ensure low costs

4　Specific System Description

4.1　Problem Description

There are many problems associated with bike locks today. People often forget or lose their keys, lock or combination. Additionally, current locking systems are often not comprehensive – they may not lock all parts of the bike that can be stolen, namely the seat, front and back wheels, and frame.

Furthermore, bike locks can be bulky, heavy and dangerous to carry around. It can also be tedious to find and lock one's bike to an external frame. The combination of these nuisances can lead to individuals leaving their bikes without properly securing them. The city of Toronto reports an average 3625 stolen bikes annually, and the Canadian Cycling Magazine estimates that only 15-20% of stolen bikes are reported, which indicates a rather expansive problem that can be solved [1,2].

4.1.1　Terminology and Definitions

The terms listed below will be used throughout the scope of this document and should be referenced when referring to specific components.

- Smart Lock: The scope of this project and document referring to both the mechanical lock and the mobile application communicating with it.

- Lock: The physical component of the Smart Lock; includes the hardware, battery, motor, and locking mechanism.

- Team: Refers to the members working on Smart Lock

- App: The mobile application used to communicate with the lock

- Open/Close: The process of physically moving the lock frame into a state where it can be latched onto something else.

- Open/Close Status: The current state of the lock frame.

- Engage/Disengage: The automated process to actuate the lock to secure the frame.

- Wireless Signalling: The means of communication between the lock and the mobile application.

- Battery: The power source used to power the motor for engaging and disengaging.

- Battery Status: A measure of the quantity of charge in the battery.

- Git: The platform used to store the documentation and software and track changes made to the files.

- XCode and Flutter: The IDE used to develop the mobile application.

- Cross-Platform Development: The means of developing the mobile application on both iOS and Android devices.

- Transmitter: The device used to send signals to and from the app to give status information and to send the engagement and disengagement signals.

- GPS: The satellite-based navigation system used to locate the bike.

- Receiver: The device used to receive the signals from the app for engagement and disengagement.

- Microcontroller: The programmable device used to control the motor.

4.1.2 Physical System Description
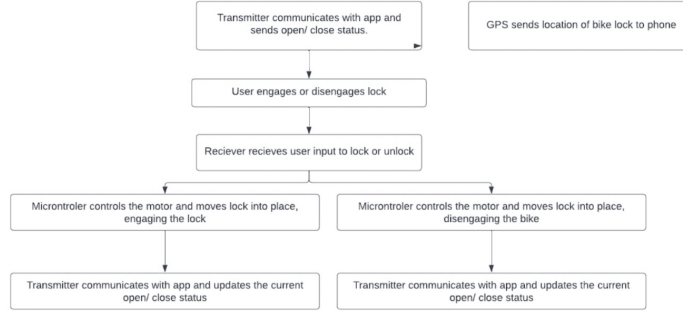
Figure 2: The Physical System

4.1.3 Goal Statements

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

GS1: Effective Bike Lock: The lock is sturdy and cannot be manual opened by the average human once engaged.

GS2: Long Lasting Battery Life: The power source is used efficiently to last a sufficient period of time without requiring replacement.

GS3: Bicycle Versatility: The lock can be used on mountain, city, kids, and road bikes.

GS4: Easily Mountable on Bike Frame: Does not require special tools to be installed.

GS5: Effective GPS Tracking: The lock can accurately transmit location to app regardless of location.

GS6: Weatherproof: The lock is not easily damaged by wind, rain, snow, heat, and cold weather.

4.1.4 Assumptions

- The bike will be left in locations with strong GPS signal. Dependencies: Locations such as garages and underground parking.

- The user's phone will always be able to connect to the lock (wireless connection works as intended). Dependencies: All users have modern phones that can operate the application. The phones will also have functional location settings.

- Users will have no limiting physical disabilities. Dependencies: Average person is strong enough to manually lock and unlock the bike.

- The lock will be weatherproof. Dependencies: Rain, snow or extreme temperatures will not damage the lock.

- Battery life is only reduced when the motor is being used. Dependencies: Battery life is independent of weather and time.

- Efficiency is always constant. Dependencies: The battery's power does not decrease over time.

4.1.5 Theoretical Models

4.1.5.1 RefName:

Torque of a Motor

4.1.5.2 Label:

$$\mathbf{T} = \mathbf{F} \cdot \mathbf{d} = \mathbf{I} \cdot \mathbf{k_T}$$

4.1.5.3 Equation:

4.1.5.4 Description:

The above equation gives the torque T (N m) required to actuate the opening and closing mechanism of the lock through the motor. Torque is equal to the force F (N) to open or close the lock, multiplied by the distance d (m) from the motor's axis of rotation. To extract current, torque is also equal to the load (motor) current I (A) multiplied by the motor's torque Constant k_T (N m A^{-1}).

The net force will be calculated once the design is finalized. The distance and torque constant will be dependent on the motor. These equations will be used to calculate the current required from the battery. They must be computed twice; once for engaging, and again for disengaging.

4.1.5.5 Notes:

None.

4.1.5.6 Source:

https://www.motioncontroltips.com/faq-whats-the-relationship-between-current-and-dc-motor-output-to:

4.1.5.7 Ref. By:

GD??

4.1.5.8 Preconditions for Torque of a Motor:

None

4.1.5.9 Derivation for Torque of a Motor:

Not Applicable

4.1.5.10   RefName:

Ohm's Law

4.1.5.11   Label:

$V = I \cdot R$

4.1.5.12   Equation:

4.1.5.13   Description:

The above equation gives the voltage V (V), which is equal to the load (motor) current I (A) multiplied by the resistance R (Ω) of the motor. This can be used to find the rating of battery needed to power the motor. This calculation must also be done twice as outlined above.

4.1.5.14   Notes:

None.

4.1.5.15   Source:

4.1.5.16   Ref. By:

GD??

4.1.5.17   Preconditions for Ohm's Law:

None

4.1.5.18   Derivation for Ohm's Law:

Not Applicable

4.1.5.19 RefName:

Battery Life

4.1.5.20 Label:

$\mathbf{BL} = \frac{\mathbf{BC}}{\mathbf{I}}$

4.1.5.21 Equation:

4.1.5.22 Description:

The above equation gives the battery life BL (h), which will be calculated by dividing the battery capacity BC (A h) selected and the load current I (A).

4.1.5.23 Notes:

None.

4.1.5.24 Source:

4.1.5.25 Ref. By:

GD??

4.1.5.26 Preconditions for Battery Life:

None

4.1.5.27 Derivation for Battery Life:

Not Applicable

4.1.6 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**??**) ($\mathrm{W\,m^{-2}\,°C^{-1}}$).
	$\Delta T(t) = T(t) - T_{\mathrm{env}}(t)$ 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]

4.1.7   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 ($\mathrm{W\,m^{-2}\,^{\circ}C^{-1}}$). 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

4.1.8   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 **?**. —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

4.1.9   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 4.1.7 to replace the abstract symbols in the models identified in Sections 4.1.5 and 4.1.6.

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, $0 < T_W < 100^oC$, 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 ...

4.1.10 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%

(*)

Table 2: Specification Parameter Values

Var	Value
$L_{\min}$	0.1 m

4.1.11 Properties of a Correct Solution

A correct solution must exhibit

Table 3: Output Variables

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

(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 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

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

5.1.1 User Input Related

FR1: LockButton input must engage the lock on the bike.

FR2: UnlockButton input must disengage the lock on the bike.

FR3: When lock is engaged, change lock status to locked.

FR4: When lock is unengaged, change lock status to unlocked.

FR5: Moving locking bar closed must move OpenClosedstatus to Closed.

FR6: Moving locking bar open must move OpenClosedstatus to open.

FR7: Location (coordinates) of user's phone must be able to be saved in the smartphone application as UserPosition. Doing so enables memory of where the user's phone was at a specific time, such as when locking the bike, which can then later be used to determine where the bike is parked/locked.

FR8: Effective Bike Lock: The lock is sturdy and cannot be manually opened by the average human once engaged.

FR0: Lock must only be engaged/disengaged by the intended user(s) (I.e., not everyone with the app can engage/disengage the lock).

5.1.2   Bike Input Related

FR10: The lock can be mounted to the bike's frame.

5.1.3   Output Related

FR11: Battery percentage must be shown on phone app.

FR12: Location (coordinates) of bike must be shown on app as BikePosition.

FR13: Battery must output enough power to engage/disengage the lock.

5.2   Nonfunctional Requirements

5.2.1   Smart Phone

NFR1: Can be used by people speaking different languages.

NFR2: Can reasonably be used without requiring an instruction manual.

NFR3: App storage under 50 megabytes. A small mobile app should not take significant space on the user's phone.

5.2.2   Physical Design

NFR4: The design must be visual appealing.

NFR5: The design must not impede normal bike functions.

NFR6: The lock must be waterproofed to withstand normal rainfall.

NFR7: The lock must be waterproofed to withstand normal splashing while riding.

5.2.3   Accuracy

NFR8: Accuracy of bike lock status must be above 90%.

NFR9: The accuracy of gps positioning must be accurate within 10M.

NFR10: Battery percentage must be calculated accurately within 10%.

5.2.4 Usability

NFR11: iPhone app locking must be quicker to use than a typical keyed/combo bike lock.

NFR12: Opening and closing lock must require similar force to a typical keyed/combo.

NFR13: Battery must last for greater than 1 month and/or 60 rides before needing to be replaced or charged.

NFR14: Batteries must be accessible to replace or charge.

NFR15: The lock must be easily mounted on the bike frame. It does not require special tools (those not found in a typical toolbox, such as power tools) to be installed and does not take more than 20 minutes.

NFR16: The lock can be used for many different models of mountain, city, and road bikes.

5.2.5 Maintainability

NFR17: Should take less than X_FRACTION of total development time.

NFR18: Making changes to the GUI.

NFR19: Motor/Battery Controller Software.

NFR20: Electrical Circuit.

NFR21: Mechanical physical design.

5.2.6 Portability

NFR22: The app should run on iOS.

NFR23: The app should be easily maintained through iOS updates.

6 Likely Changes

LC1: Depending on the battery life, showing battery percentage on the phone app, as required in FR11, might not be necessary. If the battery life is very long, then perhaps a warning that battery is low is sufficient.

LC2: Having never created an app like this before, the amount of storage needed for the smartphone application is unknown. Requirement NFR3 provides ample space, if any change, this requirement might be revised so that it be required that the app take up even less storage.

LC3: Even though aesthetics is a selling point of any product, it is not a priority for this product. As such, the requirement NFR4 is subject to change.

LC4: The degree of accuracy mentioned in NFR8, NFR9, and NFR10

LC5: Given that we will likely not be maintaining the system, requirements regarding maintabil-ity will be difficult to test and verify. While products should be designed with maintabil-ity in mind, it is not a priority for this project. Thus, requirements NFR17, NFR18, NFR19, NFR20, NFR21 are subject to change.

LC6: Requiring an iOS application as stated in NFR22 and NFR23 was chosen given the team's skillset; however, the product's nature does not require the application to be iOS based. Given different preferences and/or skillset, this application could be substituted or extended to an Android application. Note that NFR23 is dependent on NFR22

7 Unlikely Changes

ULC1: These requirements are core to the functionality of the product. The product would not accomplish it's purpose without meeting these requirements. Thus, these requirements are unlikely to change: FR1, FR2, FR3, FR4, FR5, FR6, FR7, FR8, FR9, FR13

ULC2: These requirements are key selling featueres of the system, and are therefore unlikely to change: FR10, FR11

ULC3: These requirements are necessary to ensure accessibility for all users: NFR1, NFR2

ULC4: Dependent on requirement FR10, that the lock can be mounted on the bike's frame, NFR5 that the design must not impede normal bike function is necessary for the bike to function at all, and, if the bike does not function, the SmartLock product is useless. As such, this requirement is unlikely to change.

ULC5: To maintain usability in imperfect weather conditions, requirements NFR6 and NFR7, and are therefore unlikely to change.

ULC6: The following requirements must be satisfied to maintain an edge over typical, manually engaged/disengaged bike locks, and are therefore unlikely to change: NFR11, NFR12, NFR14, NFR13, NFR15, NFR16

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??	4.1.10	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 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]

[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]

24

Appendix

10.1 Table of Monitored Variables

Variable Name	Description	Type	Units
m_SignalEngaged	Monitors whether or not the locking mechanism is engaged	Digital	Boolean
m_SignalDisengaged	Monitors whether or not the locking mechanism is disengaged	Digital	Boolean
m_SignalOpen	Monitors whether or not the physical mechanism is open	Digital	Boolean
m_SignalClosed	Monitors whether or not the physical mechanism is closed	Digital	Boolean
m_Location	Monitors the location of the bike when it is locked	Analog	Coordinates
m_BatteryPower	Monitors the current battery percentage	Analog	Percentage

10.2 Table of Controlled Variables

Variable Name	Description	Type	Units
c_LockEngaged	Engages the lock	Digital	Boolean
c_LocklDisengaged	Disengages the lock	Digital	Boolean
c_LockOpen	Indicates to the user that the latch is open	Digital	Boolean
c_LockClosed	Indicates to the user that the latch is closed	Digital	Boolean
c_BikePosition	Marks the location of the bike when it is locked	Analog	Coordinates
c_BatteryPercentStatus	Indicates what the percentage of the battery is	Analog	Percentage

10.3 Constants – NA
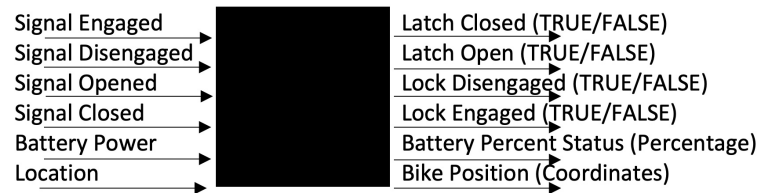
25

10.4   Stimulus and Responses
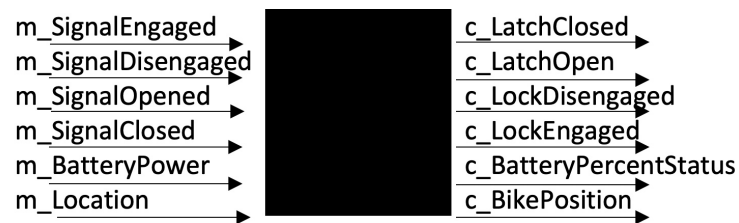

Figure 3:   Stimulus and Responses

10.5   Notation

Figure 4:   Notation

10.6 Responses

	c_LatchClosed	c_LatchOpen	c_LockDisengaged	c_LockEngaged	c_BatteryPercentStatus	c_BikePosition
No stimulus	FALSE	TRUE	TRUE	FALSE	No Percent	No Coordinates
m_SignalClosed	TRUE	FALSE	FALSE	FALSE	No Percent	No Coordinates
m_SignalClosed. m_SignalEngaged	TRUE	FALSE	FALSE	TRUE	No Percent	No Coordinates
m_SignalClosed. m_SignalEngaged. m_SignalDisengaged	TRUE	FALSE	TRUE	FALSE	No Percent	No Coordinates
m_SignalClosed. m_SignalEngaged. m_SignalDisengaged. m_SignalOpen	FALSE	TRUE	TRUE	FALSE	No Percent	No Coordinates
m_BatteryPower	-	-	-	-	Battery Percent Shown	No Coordinates
m_Location	-	-	-	-	No Percent	Coordinates Shown

Figure 5: Responses

10.7 Effect of Each Stimulus

		Current State			
		Latch open – Lock Engaged	Latch open – Lock Disengaged	Latch closed – Lock Engaged	Latch closed – Lock Disengaged
Current Stimulus	m_SignalEngaged	-	Bike ready to ride	-	Bike Secure
	m_SignalDisengaged	Device ready to be closed	-	Device ready to be opened	-
	m_SignalOpened	-	-	-	Bike removed from fixed frame
	m_SignalClosed	-	Bike attached to fixed frame	-	-
	m_Location	BikePosition	BikePosition	BikePosition	BikePosition
	m_BatteryPower	BatteryPercentStatus	BatteryPercentStatus	BatteryPercentStatus	BatteryPercentStatus

Figure 6: Effect of Each Stimulus

27

10.8 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.

   Steffi:
   My lead role on this project is in regard to Documentation and Latex. While I have experience with some project documentation, the extent to which it is necessary for this project is not something I have done before. Looking at this project in such detail will help to more deeply understand every component of the project and help to better see the connection between every aspect. Additionally, technical communication is a skill I have never used before so learning to use Github and Latex will be critical to the delivery of the project and developing skills that will be useful in future technical projects.

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?

   Steffi:
   Documentation - Two approaches that I can use to improve my skillset for documentation are reading old documentation that other groups have produced, or examples of documentation online, and writing documentation and paying careful attention to the focus of the section. The emphasis will be placed on writing documentation because with this skillset I believe the best approach is to try and then learn from what was difficult and feedback on what was produced. However, old documentation will still be referenced as well.
   Latex - Two approaches that can be used to improve my latex skill set are referencing external explanations (videos & Tutorials), and trial and error with external reference materials. I will be focusing on trial and error using some external materials. I believe that seeing what happens for yourself when you make a change is the best way to understand what is going on. The external resources will be used to give an idea of what it is possible to try to accomplish in latex, for example, certain formatting that I may not have considered existed.