

Verification and Validation Report: 4TB6 - Mechatronics Capstone

Team #5, Locked & Loaded

Abi Nevo, nevoa

Elsa Bassi, bassie

Steffi Ralph, ralphs1

Abdul Iqbal, iqbala18

Stephen De Jong, dejons1

Anthony Shenouda, shenoa2

March 6, 2023

1 Revision History

Date	Developer(s)	Notes
04-03-23	Abi, Anthony, Elsa, Steffi	Testing
05-03-23	Abi, Elsa, Steffi	Writeup

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
3.0.1	Area of Testing: User Input Related	1
3.0.2	Area of Testing: Bike Input Related	4
3.0.3	Area of Testing: Output Related	4
4	Nonfunctional Requirements Evaluation	5
4.0.1	Area of Testing: Smart Phone	6
4.0.2	Area of Testing: Physical Lock	6
4.0.3	Area of Testing: Usability	7
4.0.4	Area of Testing: Accuracy	10
5	Comparison to Existing Implementation	11
6	Unit Testing	11
6.1	Physical Locking Mechanism	11
6.2	Microcontroller	11
6.3	Circuit	11
6.4	Mobile App	12
7	User Testing	12
8	Changes Due to Testing	13
9	Automated Testing	14
10	Trace to Requirements	14
11	Trace to Modules	15
12	Code Coverage Metrics	15
13	Appendix	16
13.1	Reflection	16

List of Tables

1	Traceability Table	14
---	------------------------------	----

List of Figures

1	EffectiveLockChart	2
2	EffectiveLockGraph	3
3	EffectiveLock Percentage of Tests Resulting in a result of 4 . .	3
4	QuickLockChart	7
5	QuickLockGraph	8

This document ...

3 Functional Requirements Evaluation

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

EDIT THIS This subset of tests will be used to validate the nonfunctional requirements of our product. Completing these tests will prove various aspects of our product's needs. these aspects include smartphone app features, physical design attributes, accuracy, and the usability of our product. Note that each requirement references, and is directly mapped to, at least one requirement, showing that these test cases robustly cover the defined requirements in the SRS.

3.0.1 Area of Testing: User Input Related

1. DisengageLock:

FR1: LockDisengage input must disengage the lock on the bike.

Expected Results: The lock will successfully disengage upon receiving the DisengageLock Signal.

Actual Result: Pass – The lock successfully disengaged upon receiving the DisengageLock Signal.

2. LockLocation:

FR4: Location (coordinates) of user's phone must be able to be saved in the smartphone application as UserPosition.

Expected Results: Pass/fail if the App places a geotag within 10 metres of the Users current location

Actual Result:

3. EffectiveLock

FR5: Effective Bike Lock: The lock is sturdy and cannot be manually opened by the average human once engaged.

Expected Results: Perform the following user tests and achieve an average score at or above 90%. Optional scores of 1-4 for the following cases: a fail if the lock disengages and breaks, a fail if the lock disengages, a pass if the lock stays engaged but breaks, and a pass if the lock system can stay engaged without breaking.

Actual Results:

Three test subjects' scores from 1-4 out of 50 trials:

	Subject 1	Subject 2	Subject 3
Score			
1	0	0	0
2	0	0	1
3	1	2	3
4	49	48	46

Figure 1: EffectiveLockChart

Therefore, result of pass due to average engagement without breaking of 95% of the time when the test force was applied.

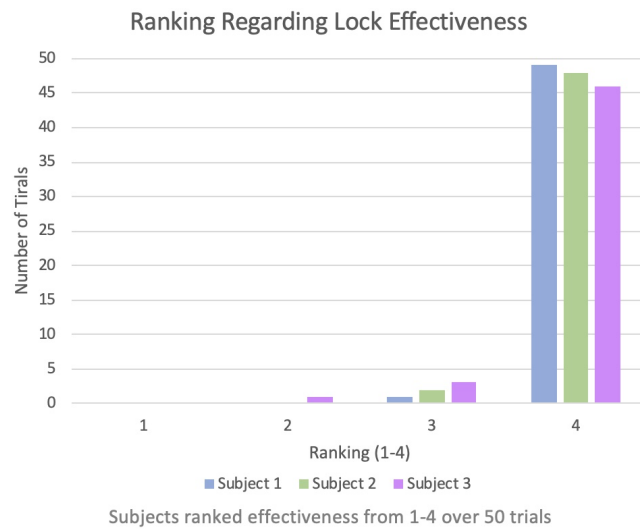


Figure 2: EffectiveLockGraph

Test Results as a Percentage of Total Tests

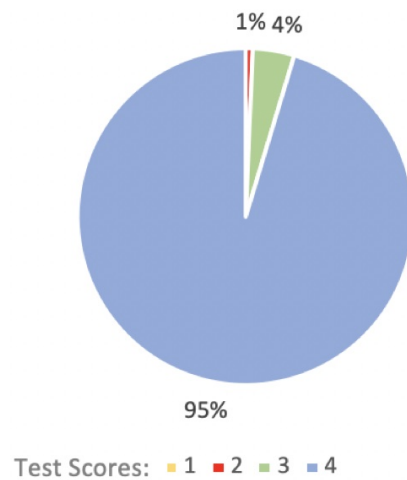


Figure 3: EffectiveLock Percentage of Tests Resulting in a result of 4

4. EffectiveLockSimulation

FR5: Effective Bike Lock: The lock is sturdy and cannot be manually opened by the average human once engaged.

Expected Results: A pass/fail if the simulation meets the 200-400 N threshold.

Actual Results:

5. Security

FR6: Lock must only be engaged/disengaged by the intended user(s).

Expected Results: A Bluetooth developer app is downloaded on an unintended user's phone. The unintended user will be able to connect to the Arduino but not be able to disengage the lock.

Actual Results: Pass – The unintended user was able to connect to the device but not able to unlock the device.

3.0.2 Area of Testing: Bike Input Related

1. LockMount

FR7: The lock can be mounted to the bike's frame.

Expected Results: A pass/fail per bike tested if the lock is/isn't able to be mounted onto the bike as intended. A score from 0-the number of bikes tested will be given.

Actual Results: Three test subjects' results for the number of bikes where successful lock mounting occurred:

1 – 3/3

2 – 3/3

3 – 3/3

Therefore, of nine bikes tested, all passed.

3.0.3 Area of Testing: Output Related

1. BatteryPercentage

FR8: Battery percentage must be shown on the phone app.

Expected Results: The user must be able to view the battery percentage of the lock.

Actual Results:

2. LocationOnApp

FR9: Location (coordinates) of the bike must be shown on the app as BikePosition.

Expected Results: The user must be able to view the saved coordinates

Actual Results:

3. PowerOutput

FR10: Battery must output enough power to engage the lock.

Expected Results: Circuit connected. Battery is successfully able to meet the threshold voltage of the electromagnet to engage the locking mechanism.

Actual Results: Pass – The battery successfully supplied enough voltage to engage the locking mechanism in 5/5 tests.

Note: During one test where the Arduino was powered on for more than an hour, the battery was supplying the correct voltage, however, the Arduino became fried and defective, and the locking mechanism was not engaged successfully. Moreover, it was determined that the failed test was due to the Arduino frying, so it was not counted as a failed PowerOutput test.

4 Nonfunctional Requirements Evaluation

EDIT THIS: This subset of tests will be used to validate the nonfunctional requirements of our product. Completing these tests will prove various aspects of our product's needs. These aspects include smartphone app features, physical design attributes, accuracy, and the usability of our product. Note that each requirement references, and is directly mapped to, at least one requirement, showing that these test cases robustly cover the defined requirements in the SRS.

4.0.1 Area of Testing: Smart Phone

1. LimitedInstructions

NFR1: Can reasonably be used without requiring an instruction manual.

Expected Results: The lock is successfully engaged, and the bike is securely locked by all test users.

Actual Results: GRAPH

2. AppStorage

NFR2: App storage under 50 megabytes. A small mobile app should not take up significant space on the user's phone.

Expected Results: App storage is less than 50 MB.

Actual Results: Pass – App storage does not exceed 50 MB.

4.0.2 Area of Testing: Physical Lock

1. VisualAppeal

NFR3: The design must be visually appealing.

Expected Results: Survey users on their opinions of the visual appeal of the device; see ???. Visual appeal is rated 7 or higher (on a scale of 1-10).

Actual Results: GRAPH

2. NormalBikeFunction

NFR4: The lock must not impede normal bike functions.

Expected Results: The SmartLock does not impede normal bike functionality and operation.

Actual Results: Pass – The SmartLock does not impede normal bike functionality and operation.

3. Safety

NFR5: The design must not inflict harm to the user in any way, such as clamping down on a finger, or moving at a force or speed that could cause injury.

Expected Results: The design must not inflict harm to the user in any way, such as clamping down on a finger, or moving at a force or speed that could cause injury.

Actual Results: Pass – User is not harmed or pinched when SmartLock is in use.

4.0.3 Area of Testing: Usability

1. QuickLock

NFR8: The SmartLock must be quicker to use than a typical keyed or combination bike lock.

Expected Results: The SmartLock must be quicker to use than both a sample keyed and combination lock. One test user will attempt three times each to lock and unlock typical keyed and combination as well as the SmartLock. All tests must be faster for the SmartLock.

Actual Results:

	Keyed lock Time:	Keyed Unlock Time:	Combo Lock Time:	Combo Unlock Time:	SmartLock Lock Time:	SmartLock Unlock Time:
Test 1	28.75	24.52	15.6	18.21	12.31	11.36
Test 2	26.65	20.43	18.45	18.63	11.34	11.41
Test 3	30	21.75	16.3	19.2	11.61	12.26

Figure 4: QuickLockChart

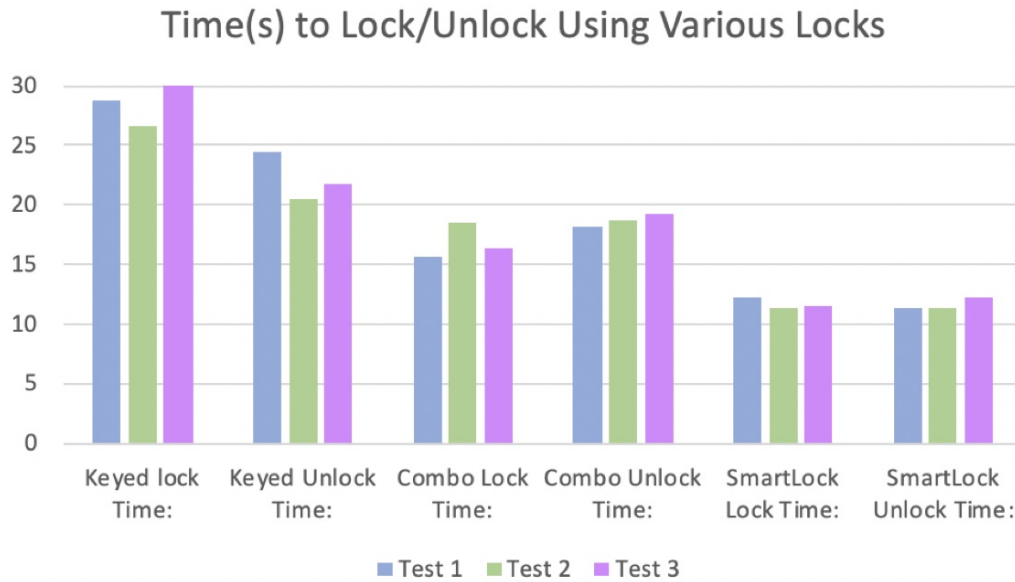


Figure 5: QuickLockGraph

2. UseForce

NFR9: Opening and closing lock must require similar force to a typical keyed/combo.

Expected Results: The amount of force required to open and close the lock frame is comparable to that of a typical keyed or combination lock. Test users score the force required out of 5 where 5 is the maximum amount of force they can physically provide and 1 is the minimum amount of force they can physically provide. The SmartLock should not vary more than 1 score unit from the most different score in each test.

Actual Results: Three tests subjects' results for the amount of force required to open and close the lock frame for the SmartLock, a typical keyed lock and a combination lock respectively out of 5:

1- 2, 1, 2

2- 3, 3, 3

3- 1, 1, 2

Therefore, the SmartLock did not vary more than one score unit from the most different score in each test.

3. BatteryLife

NFR10: Battery must last for greater than 1 month and/or 60 rides before needing to be replaced or charged.

Expected Results: The battery lasts for longer than 1 month and/or 60 rides before needing to be replaced or charged.

Actual Results: Pass – The battery lasted longer than 1 month before needing to be replaced or charged.

4. ComponentAccessibility

NFR11: Batteries and other internal components must be accessible to replace and/or chargeable.

Expected Results: The battery can be replaced, and internal components can be accessed, as intended.

Actual Results: Pass – Three test users were able to access and replace the battery and all other internal components.

5. NoSpecialTools

NFR12: The lock must be easily mounted on the bike frame. It does not require special tools, (i.e., those not found in a typical toolbox, such as power tools), to be installed and does not take more than twenty minutes to install.

Expected Results: The lock is easily mounted on the bike frame. It does not require special tools, (i.e., those not found in a typical toolbox, such as power tools), to be installed and does not take more than twenty minutes to install.

Actual Results: Three test subjects' results for three individual tests for the time to mount the lock on their bike frame without the use of special tools.

1- 6, 5, 4 min

2- 7, 7, 4 min

3 – 5, 3, 3 min

Therefore, all tests were successful and completed within twenty minutes.

6. BikeVersatility

NFR13: The lock can be used for many different models of mountain, city, and road bikes.

Expected Results: The SmartLock can be mounted on three different categories of bikes (road, hybrid and mountain) successfully by each test user.

Actual Results: Three test subjects' results for the number of bikes where successful lock mounting occurred:

1 – 3/3

2 – 3/3

3 – 3/3

Therefore, of nine bikes from three different types of bikes (road, hybrid and mountain) tested, all passed.

7. AppOS

NFR18: The App should run on iOS and Android.

Expected Results: App operates on iOS and Android.

Actual Results: Pass – App operates successfully on iOS and Android as expected.

4.0.4 Area of Testing: Accuracy

1. BatteryAccuracy

NFR7: Battery percentage must be calculated accurately within 10%.

Expected Results: The number of lock engages possible with 1% battery matches our specification for the number of lock engages possible with 100% battery (multiply the measured # of lock engages possible with 1% battery by 100), within 10% accuracy.

Actual Results:

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

6.1 Physical Locking Mechanism

1. When small pin in "locked" position, large pin cannot be pulled out.

Result: PASS

6.2 Microcontroller

1. When Arduino receives signal from Bluetooth development app "nRF Connect", on-board LED turns on.

PASS

2. When Arduino receives signal from Bluetooth development app "nRF Connect", high voltage signal is sent to desired pin.

PASS – validated using multimeter.

6.3 Circuit

1. When high power signal is transmitted to gate of transistor from power source (NOT Arduino), LED connected to drain of transistor turns on.

PASS

2. When high power signal is transmitted to gate of transistor from power source (NOT Arduino), solenoid connected to drain of transistor is enabled.

PASS

3. When high power signal is transmitted to gate of transistor from Arduino, solenoid connected to drain of transistor is enabled.

PASS

6.4 Mobile App

1. Able to connect to Arduino from mobile app.
PASS
2. Arduino receives signals transmitted to it from the mobile app.
PASS – validated using the on-board Arduino LEDs.

7 User Testing

8 users were asked to try the SmartLock, and were asked several questions about their experience. These questions can be found in the [VnV Plan](#), and are also reiterated here, with the common answers we received.

1. How long does it take you to open the lock? Is a 15-second window too short? Too long?
15 seconds was found to be just slightly too long of an unlock window. It was found that 10 seconds was enough time to pull the pin out of the body of the lock. This change will be implemented.
2. Is the app intuitive to use? Would you change any features or any part of the app design?
Many users commented on the confusing nature of listing all active Bluetooth devices in the area, rather than only the SmartLock. In order to limit this confusion, the list of Bluetooth devices will be filtered so that only the SmartLock appears.
3. How did you like the whole product use experience? Is the locking/unlocking process smooth, convenient, and efficient?
Especially due to the current cold weather, users enjoyed not needing to fiddle with a key or combination lock, and just being able to press a button to unlock their bike.
4. Was any aspect of using the product confusing?
Again, users commented on the listing of all Bluetooth devices as being confusing. Users also mentioned that it was not clear that the lock would only be disengaged for a limited period of time after pressing

"Unlock". This concept must be clarified either within the app, or in a separate instruction manual.

5. Would you be comfortable using this product to lock your bike? What would make you feel more secure?

Due to all of the internal components currently being not secured properly, and the solenoid being exposed to the outside, most users said they were not comfortable using the SmartLock. In order to make the lock more secure, the housing could be redesigned to hold all of the components internally.

8 Changes Due to Testing

Changes due to [User Testing](#) are summarized below:

1. Change the lock disengaged window (how long the lock is disengaged after pressing "Unlock") to 10 seconds from 15 seconds.
2. Filter the Bluetooth devices shown so that only SmartLock appears. Alternatively, make the mobile app only able to connect to the SmartLock.
3. Redesign lock housing so that all components fit internally properly and securely.

9 Automated Testing

10 Trace to Requirements

Table 1: Traceability Table

Test Case	Functional Requirement(s)	Non-Functional Requirement(s)
DisengageLock	FR1	
LockLocation	FR4	
EffectiveLock, Effective-LockSimulation	FR5	
Security	FR6	
LockMount	FR7	
BatteryPercentage	FR8	
LocationOnApp	FR9	
PowerOutput	FR10	
LimitedInstructions		NFR1
AppStorage		NFR2
VisualAppeal		NFR3
NormalBikeFunction		NFR4
Safety		NFR5
BatteryAccuracy		NFR7
QuickLock		NFR8
UseForce		NFR9
BatteryLike		NFR10
ComponentAccessibility		NFR11
NoSpecialTools		NFR12
BikeVersatility		NFR13
AppOS		NFR18

Note: FR 2-3 and NFR 6, 14-17 and 19 were deemed out of scope on the SRS, and thus are not present in this Verification and Validation Plan.

11 Trace to Modules

12 Code Coverage Metrics

13 Appendix

13.1 Reflection

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

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)