Reflection Report on 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

Contents

List of Tables

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

1.1.1 SRS

In response to peer feedback:

1. Edited document inconsistencies related to Bluetooth connective capabilities.

2. Added safety requirement to address harm to users; 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.

3. Added System Constraint, SC6: The App must be developed for iPhone and Android, to address operating system constraints.

4. Removed weatherproof and waterproof requirements; kept them as Assumptions (A3: The lock will be weatherproof) for the purposes of the project.

5. Removed requirement related to language, NFR1: Can be used by people of any language. Instead, added language constraint, A6: The user speaks English.

In response to TA feedback:

1. Edited the Table of Units to reflect only the units relevant to our project.

2. Updated the requirements and their clarity, as well as overall documentation clarity

1.1.2 Hazard Analysis

In response to peer feedback:

1. Added numerical bike dimensions in critical assumption, CA5: Assume operator's bicycle has standard frame and dimensions, and functions properly. A standard frame is 15" to 21" long and 13" to 20" high, with a 28.6mm tube diameter. These specifications allow the group to constrain its design for the specific average bike sizes mentioned.

2. Added numerical weather-related values to Critical Assumption, CA2: Assume weather is typical of Canada, thus the device shall not be operated in temperatures outside the range of -30 degrees Celsius to +40 degrees Celsius. It shall also not be operated under rainfall greater than 5 millimetres per hour or wind speeds greater than 10 kilometres per hour. These specifications allow the group to constraint its design for average Canadian weather conditions.

3. Deemed rainfall and splashing-related Safety Requirements out of scope due to realistic technological capacity and duration of the project.

In response to TA feedback:

1. Added trace of Hazard Analysis requirements to requirements in the SRS. Note that in the VnV Plan, safety requirements from Hazard Analysis are covered as requirements from SRS (because safety requirements from Hazard Analysis are now in SRS).

2. Added a list of tables and figures.

3. Added an overview of the project.

1.2 Design and Design Documentation

1.2.1 Design Documentation

In response to TA feedback:

1. Removed modules: Input Parameters, Output Parameters, Engage Status Signal, Load Power, Bluetooth Connection, and substituted these with the added modules: Arduino Bluetooth Communication, Mobile App Bluetooth Communication, User Input to Phone, Solenoid Actuation, User Disengage, and Hardware Disengage.

2. Reworked module hierarchy and DAG to suit these modules (see MG).

3. Revised MIS to suit these modules, and added much greater detail to the MIS of each module, including more detailed constants and state variables, and Access Routine Semantics.

4. Updated the table captions and added a table of tables to the System Design document.

5. Added computer drawn diagrams for Figure 1,2, and 10 (System Design).

6. Added CAD drawing to complement the original sketch done for the mechanical prototype (System Design).

7. Updated the Introduction to remind the reader of the project (System Design).

In response to peer feedback:

1.

1.2.2 Design

In response to supervisor feedback from initial concept/POC:

1. Rather than having mechanical arm that swings, which requires lots of power, and a bulky motor, changed design to be a pin-locking mechanism using a solenoid, which takes much less volume, and consumes much less power.

2. Changed design so that the lock is engaged on off-state, and only when you disengage the lock does the locking mechanism require current/power. This is also energy-saving.

3. Changed design to take advantage of the user's power to engage the lock, rather than using electrical power.

In response to instructor and TA feedback from Rev 0 Demo:

1. Refocused and slightly altered priorities to get the prototype properly functioning.

2. Added diode across solenoid to protect against inductance surges.

In response to supervisor feedback from Rev 0 Demo:

1. Got technical advice for how to improve the circuit to get it functioning properly: added resistor between Arduino and transistor gate, ensured resistor values were appropriate.

In response to User Testing:

1. Change the lock disengaged window (how long the lock is disengaged after pressing "Unlock") to 10 seconds from 5 seconds, as suggested by users' answer to the question "How long does it take you to open the lock? Is a 5-second window too short? Too long?"

2. Display coordinates of geotagged/pinned location on app. This way, users can then copy these coordinates and paste them into a map app of their choosing to get directions. This change was prompted by users' answers to the question "Is the app intuitive to use? Would you change any features or any part of the app design?"

3. Redesign the App UI to make it more user-friendly and intuitive, as prompted by users' answers to the question, "Was any aspect of using the product confusing?"

1.3 VnV Plan and Report

1.3.1 VnV Plan

In response to TA feedback:

1. Added references for specifics of some test cases (i.e., where we got the value for amount of force for stress test).
2. Removed water resistance test.
3. Added test to simulate strength/stress test using CAD rather than do it in reality.

In response to peer feedback:

1. Verification and Validation team table revised.
2. Added description of how each requirement is fulfilled by each teat to Implementation Verification Plan; Added statement to introduction of Functional and Nonfunctional Requirement Test sections that each requirement maps directly to a test.
3. SRS Verification Checklist hyperlinked.
4. Added descriptive test IDs.

1.3.2 VnVReport

In response to TA feedback:

1. Added extensive unit testing.
2. Added Automated Testing section.

In response to peer feedback:

1. Added references in User Testing section for where changes stemmed from.
2. Added detail to expected results–namely, added the rating which is deemed satisfactory in the Visual Appeal test.
3. *Added Automated Testing section (also received this feedback from TA)*
4. Added reference to VnV Plan for full test descriptions.

2 Design Iteration (LO11)

When initially brainstorming our design, there were many factors to consider. Firstly, we needed to decide whether we wanted the lock to be mounted onto the bike frame or not. After weighing the pros and cons, we decided to have the lock mounted on the bike frame. In doing this, we also had to consider where exactly the product would be mounted on the bike frame. Initially, our design consisted of three separate locks: one on each wheel, and another to lock to an external frame. We very quickly realized that this large of a scope would not be feasible for this project, and reduced the scope to one locking hub mounted

on the frame of the bicycle near the seat, with a chain long enough to be able to wrap around the frame and through one of the wheels.

Secondly, we had to think about the design of the actual locking mechanism. At first, as mentioned previously in this reflection, we intended to have an arm that would move using a motor back and forth/inside the lock housing to engage/disengage the lock. Our supervisor pointed out that this would make for a very bulky design, and would also use a lot of power. Due to this, we changed our design to simply have a locking pin that moves in and out using an electromagnet. Further, we designed the locking mechanism to take advantage of the user's power to engage the lock (the user inserts the large pin into the lock housing themselves to engage the lock), rather than using electrical power to both engage and disengage.

Next, we also thought about the arm that would extend past the lock (the piece that wraps around an external frame)–we had both rigid and flexible design options, and ended up going with a chain–both flexible and sturdy–due to the versatility it would offer.

Once we had this design, we started working on the prototype. During this process, we realized that the electromagnet did not have a strong enough electromagnetic force to push and pull the locking pin. To solve this problem, we replaced the electromagnet with a solenoid that actuates laterally, which suited our use case perfectly. To make the solenoid work, we had to add many different components to the electrical circuit that powers it, such as current-limiting resistors, diodes to protect against magnetic inductance and transistors with lower threshold voltages.

We also began working on the housing of the lock, which holds the electrical circuit and locking mechanism. We used computer-aided design (CAD) to 3D-print the housing. Then we fastened the chain to the pin and the housing, and added a mounting clip to the top of the housing such that it can be attached to a bike frame.

The app was built simultaneously to the electrical and mechanical design components. Its development process began with the goal of using the development platforms of XCode and Swift to build an exclusively iOS app, because it was estimated that the time to build an additional Android app was too great. Ultimately it was decided upon to use Flutter, a cross-platform application, to create an app that was compatible with both Apple and Android phones. After spending countless hours learning how to use Flutter, the rough layout of the app architecture was designed; its user interface and features, based on the established requirements and scope of the project. After this plan was completed, the core layout and features were implemented, namely the Bluetooth connection, geotagging and battery status. These priority functionalities were successfully enacted ahead of schedule, leaving time to implement additional features such that the app could go above and beyond our initial expectations. Password protection, a more user-friendly interface, an interactive geotagging map and more visually appealing colours and fonts were added to improve the user experience, despite not being required. Overall, the development process involved prioritizing the established requirements, and then assessing additional

features that could be completed within the remaining design timeline.

The next stage in our design process was to perform user testing. We qualitatively surveyed ten users in several topics; namely the layout of the app and its features. We received valuable feedback that the user interface could be more easy to use with more self-explanatory buttons and a grouping and separation of related and unrelated features. The app usability was then improved and two additional features were added. Firstly, the ability to move the geotagging map such that the greater area around the geotagged location could be viewed at a user's demand was created using a Google Maps API. Secondly, test users told us that a way to improve the functionality of the geotagging feature would be the capability to copy the geotagged location so that it could be inputted into Google Maps for directions. We responded by implementing a feature to copy the precise coordinates of the geotagged location, so they could be pasted into any GPS mapping app. Lastly, we performed several quantitative user tests to measure the effectiveness of the locking mechanism. We ultimately determined from this testing that the time allowed for the user to remove the pin after disengagement (unlocking) and replace it inside the housing should be increased to allow users more time to execute this action. We responded by increasing the energized state holding time of the solenoid from 5 seconds to 10 seconds, giving the users more time to remove and replace the pin. Therefore, carrying out user testing gave us valuable feedback that we wouldn't have otherwise thought of, and we used to greatly improve our design.

3 Design Decisions (LO12)

As was mentioned in the Design Iterations section, there were several assumptions, constraints and limitations that influenced our decisions during the design process.

Our decision to replace the electromagnet with the solenoid was a choice driven out of necessity due to a design flaw. We had originally built our mechanical and electrical circuit designs around using the magnetic force of an electromagnet to pull the pin of the locking mechanism by attaching another magnet to it. We had tested the electromagnet and magnetic pin together to confirm that they worked, however, when we added them to the first prototype of the housing, we realized that the magnetic faces of the pin and electromagnet were too far apart - the electromagnet did not produce a high enough magnetic force to pull the pin into place to actuate the locking mechanism. Therefore, we immediately began troubleshooting the electromagnet to achieve a larger magnitude of magnetic force; we looked into a stronger power supply and even purchased a larger magnet. Neither of these solutions worked, so we decided to change our design by using a solenoid instead of an electromagnet. This design change perfectly fit our use case because it meant that a successful execution relied only on the ability of the plunger bar of the solenoid to laterally actuate out of the gap of the pin to 'disengage' the locking mechanism, instead of relying on a magnetic force that had to be high enough to pull the pin into place. It

was essentially transforming a force threshold constraint problem into a binary, non-performance-based problem, since the solenoid would succeed as long as it received power, that is, it wasn't a force-limited solution; it didn't depend on how much power it received or outputted.

We chose to use a chain as the part of the design that attaches to an external bike rack due to its versatility in being flexible and strong. We decided this because not only does a chain allow the user to lock the bike around almost any external frame shape, but it also fits through a bike wheel, so that the wheels do not get stolen.

Our design decision to use a pin for the locking mechanism in order to use less energy and keep the design compact was also a great iteration. It allowed us to eliminate a motor from the design and swap it for an electromagnet, which could be much more easily powered and controlled by an Arduino with minimal software. Mounting the lock on the bike frame also constrained the size of the housing, meaning that these design decisions complemented each other to keep it relatively small.

Regarding the app several important design decisions were made, including the choice of the cross-platform development framework. Both Flutter and React Native were considered, the two leading cross-development apps, but ultimately Flutter was chosen because it is better equipped for Arduino Bluetooth interaction and GPS and it is newer, so has better compatibilites. Another important design decision was which Bluetooth package to implement on Flutter. We chose Flutter Blue because it has the most reliable Arduino connection capability, or its ability to hold a secure device connection. Furthermore, it has the most robust communication protocols for reading and writing to Arduino, which was the highest priority of functionality based on our requirements. The drawbacks of Flutter Blue are that it has a limited physical range for its Bluetooth connection capabilities but we determined that this is not a deciding factor because in our use case, the user interacts with the SmartLock and bike in close proximity.

4 Economic Considerations (LO23)

According to Fortune Business Insights, in 2021, the global bicycle market was valued at 78.33 billion USD, with a compound annual growth rate (CAGR) of 6.5%. Rising traffic congestion and eco-consciousness propel this growth, with more and more people opting to cycle. Within the McMaster community specifically, the University reports that there have been more than 86,000 SoBi (the Hamilton Bike Share company) trips to and from campus since 2015 and that the McMaster Student Center Hub remains the busiest and most popular hub in the city. The booming cycling industry, in turn, means a booming bike-lock industry–after all, every cyclist needs to lock their bike at their desired location. All of these users likely experience the pain that comes from fiddling and struggling with a typical keyed or combination bike lock, meaning that there is most definitely a market for our product.

According to a Transportation Demand Management report completed by McMaster University, it was estimated that in 2025, about 3% of students will use bicycles as their main form of transportation. With a current student population of around 36,500 students, this means that within the McMaster student community alone, our product would have over 1,000 potential users.

To market this product, we would use the video we created for the Capstone EXPO as promotional material, and publish this video, or a condensed version of it, to social media, including TikTok, Instagram, and FaceBook. We would target our advertisements at students in the Hamilton area to start, growing our target audience as the product becomes more successful, and after some further design iterations.

At the current moment, we do not believe our product is ready for the commercial market. Some key features that we could not implement due to resource and time constraints would have to be implemented for this to happen. Firstly, and most importantly, the lock housing must be machined such that the chain is welded to the lock housing (currently, the chain is attached to the lock housing by hooking it over the open loops integrated into the housing, which is obviously not secure at all). Next, to reduce manufacturing costs, and the size of the lock, we would likely design our own PCB and microcontroller.

Due to the changes that are necessary before we can bring the product to market, we cannot give an accurate estimate of how we would price the SmartLock. However, we can provide these estimates based on the current, existing product. The budget for this project is as follows:

Table 1: Project Items and Costs

Item	**Cost (CAD)**
Arduino Nano 33 BLE	$34.99
3D Printing Lock Housing & Mechanism	$16.45
Circuit components (resistors, transistor, diode)	$5.00
Apple Developer Program	$133.22 per year
Solenoid	$20.00

From the budget, we can see that our total cost to produce one SmartLock is currently $106.44 CAD, with an additional $133.22 per year for the Apple Developer Program, which allows us to publish the SmartLock app to the Apple App Store.

According to industry standard, the typical markup for a product is about 50%. As such, with the current budget, we would sell the SmartLock for around $160 CAD.

Again, it is difficult to estimate how many units we would need to sell to generate profit because the existing product is still a ways away from being able to be commercially sold. For example, manufacturing and tooling costs would change, as the full housing and mechanism would need to be machined rather than 3D printed, and the size of the product itself would also differ once we design a PCB and shrink the size of the circuit. Other factors we would

need to consider include inventory and fulfillment–if we chose to use Amazon for fulfillment, they can store inventory and fulfill items for you, and have a calculator to identify exactly how much inventory & fulfillment would cost. We would also have to consider the cost of labour for all employees making the SmartLock a reality, and marketing costs.

Nonetheless, we can make an estimate for the cost of machining the mount, housing, pin and chain, which would cost approximately $60 CAD. On a large scale we would be manufacturing our product, attempting to ramp up sales to a minimum of 1,000 units sold, levelling costs to around $60,000. The cost of manufacturing the electrical components, (resistors, transistor, wires, magnet, diode, battery), which are relatively inexpensive, must also be added and is estimated at around $2,000 CAD for 1,000 units. This number is excluding the cost savings of bulk manufacture as this value is largely variable and difficult to measure. As was mentioned above, to bring our product to market we would also design our own embedded electronics. It would cost roughly $10,000 CAD to design our own PCB, plus an additional $5,000 CAD to implement its embedded software. Therefore, total costs, including our Apple Store license, are roughly $77,133.22 to manufacture 1,000 units. This number is likely as high as the cost would ever be due to economies of scale, thus, at this maximum cost, we would price our commercial product at $77.13 per unit to break even if we were to sell 1,000 units.

5 Reflection on Project Management (LO24)

5.1 How Does Your Project Management Compare to Your Development Plan

1. Team Meeting Plan

   We consistently held meetings twice a week in first semester (as outlined in the Team Meeting Plan section of the Development Plan), and then once a week throughout second semester on Wednesdays at 4 PM. Meeting minutes were taken every week, and posted to the Teams group, however usually an agenda was not posted ahead of time. Instead, we usually briefly mentioned what the goal of the meeting was in our group text chat before the meeting. We also found meeting with our supervisor every other week far too often, especially in the first half of the course when our primary focus was documentation and ended up meeting with him less frequently, which was fine.

2. Team Communication Plan

   We followed this section pretty well–quick communication through the group text chat, used GitHub for documentation, hosted any virtual meetings through Teams, and posted our meeting minutes on Teams. However, rather than using GitHub for task tracking, we found ourselves writing each member's task/actions/deliverables in the meeting minutes directly.

This didn't necessarily "go wrong", since actions were still documented, and each member still knew what they were responsible for, but it just didn't exactly match the Development Plan.

3. Team Member Roles

   We found that it didn't really make sense to only have one or two people on documentation–in the first half of the course, the majority of our work was documentation, and we needed the whole team's help to do that–this did not match the original Development Plan. We also found that some roles, namely the Software (Wireless Communication) role to be unnecessary, and these responsibilities ended up in the realm of other roles.

4. Workflow Plan

   As the Development Plan states, the main branch of our repository always contained a working copy only, and the repository was committed to frequently, with descriptive commit messages. However, GitHub issues were not used as often as the Development Plan may have suggested–we likely did not take full advantage of the issue tracking features GitHub has to offer. This being said, we did still track team member's responsibilities and bugs in the meeting minutes, and every member always knew what their action items were–we just didn't use GitHub issues for this project management.

5. Technology

   As the Development Plan indicates we did develop the mobile app using Flutter and Dart. We also did in fact use the Arduino Nano 33 BLE as our microcontroller, and developed the firmware using the Arduino IDE and C/C++. For prototyping, we did 3D print the lock housing and mechanism, as the Development Plan indicates, but did not get to laser cut and machine the housing or mechanism–we ran out of time and resources.

5.2 What Went Well?

1. Team Meeting Plan

   By having consistent meetings, we ensured that we were making progress on some front every week, and kept in touch about what every group member was responsible for. Meeting minutes were consistently taken and posted on the Teams group for future reference, which was very helpful throughout the project.

2. Team Communication Plan

   We used the group text chat effectively to give reminders of meetings, and other not nuanced messages. We made sure to promote any real, in-depth conversations about the project to team meetings. We also effectively used GitHub to host documentation and source code.

3. Team Member Roles

   Most team members were flexible, and able and willing to move onto any task that required extra assistance, whether they were technically assigned to it in the Development Plan, or not.

5.3 What Went Wrong?

1. Team Meeting Plan

   Very often the full group was not in attendance at group meetings, which made communication confusing because even though absent members could read the meeting minutes, sometimes there would be certain details that they missed (whether on the meeting minutes or not), and the rest of the team would be unaware that the absent group members did not have awareness of said detail. In other words, meeting attendance was a hindrance to our communication.

   Also, near the beginning of the year, there were some meetings where one or two members would join the meeting over Teams, while the rest of the team was in person, together. We very quickly realized that this didn't work well at all–those who were online could not hear everything that was said in person, and those that were in person could not hear what the people online were saying, people were talking over each other, etc. We then encouraged all team members to come in person to meetings, or, when we weren't all in Hamilton, host meetings with everyone virtually.

2. Team Communication

   In between meetings, and especially before key deliverables (like the Rev 0 Demo), we found that we were lacking communication in updating the team if there was a vital flaw or issue. This was especially the case when, as mentioned earlier, not all team members were present at meetings. If there were better status updates to the whole group throughout the project, then perhaps problems might have been solved more quickly and efficiently.

3. Team Member Roles

   We found that we often did not have enough time to thoroughly check/faux mark our documentation.

4. Workflow Plan

   While informal testing was conducted throughout the development of the project, official unit testing was done completed until the end. Because it was left until later, it became a large and arduous task, which could have been avoided if a proper unit testing methodology was established and documented early on, however, we did not anticipate this need.

5.4 What Would You Do Differently Next Time?

1. Team Meeting Plan

   We would make sure it is a known expectation for all members to be present at all group meetings. Obviously, there should be room for flexibility though–if a team member cannot make the meeting, they should give enough notice so that the meeting can be rescheduled for some other time in the week. This would make communication to the whole team much easier, and the whole team would be present for every member's status update each week.

2. Team Communication

   In order to resolve the problems identified in the section above, we would set and enforce an expectation early on in the project to create GitHub issues for bugs, making sure to tag all necessary members so that they are notified. Further, all team members being present at all group meetings would help with communication.

3. Team Member Roles

   We would manage our time better so that the faux marker actually had enough time to go "faux mark" the documentation. Perhaps we would set a deadline a few days or a week before the actual deadline.

4. Workflow Plan

   In order to resolve the problems identified in the section above, we would establish a testing and unit testing methodology as early as possible, so that modules and functions could be formally tested right all throughout development, rather than right at the end.