

Development Plan for SmartLock 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

April 5, 2023



Table 1: Revision History

Date	Developer(s)	Change
25-09-22	Abi	Drafted Team Meeting, Communication Plan, & Team Member Roles
26-09-22	Elsa	Drafted Workflow Process
26-09-22	Stephen	Drafted Intro, Demo Plan, & Scheduling
10-11-22	Steffi	Updated Development Plan, & Technology
19-11-22	Steffi	Updates for grammar, formatting and terminology
23-11-22	Steffi	Updates for consistency across documentation
04-04-23	Steffi	Final Updates

Contents

1	Introduction	1
2	Team Meeting Plan	1
3	Team Communication Plan	1
4	Team Member Roles	1
5	Workflow Plan	2
5.1	Code Development	2
5.2	Rules for Merging	2
5.3	Commits	2
5.4	Workflow Process	2
5.5	Issue Tracking	3
5.6	Milestones	4
6	Proof of Concept Demonstration Plan	4
7	Technology	5
8	Coding Standard	5
9	Project Scheduling	5

List of Tables

1	Revision History	i
2	Team Member Roles	2

1 Introduction

The SmartLock will be the first model of smart bike locks produced by Locked & Loaded. The lock will be mounted to your favourite bike with many features that can be controlled from your smartphone. Additionally, other outputs may be monitored such as the lock state, battery level and bike position.

2 Team Meeting Plan

Our team will have weekly meetings on Mondays at 10:30 AM and Thursdays at 2:30 PM, in Thode library. If additional meeting times are still necessary, our group will arrange for a meeting on Friday or over the weekend, or if it is a lighter week Monday meetings may be cancelled at the discretion of the team.

Meetings with our supervisor, Dr. Sirouspour, will occur biweekly (once every two weeks), either in-person or online via Teams depending on Dr. Sirouspour's schedule.

An agenda will be created and committed to our Teams channel prior to meeting. This responsibility will be assigned to a different team member each week, on a rotation, however, all team members may suggest agenda topics. The agenda will include suggested meeting topics, who will present/lead each topic, and an estimated time of discussion. Meeting minutes will also be recorded in the same document as the agenda, by the same team member, and be uploaded to our Teams channel after the meeting. This team member will also be responsible for chairing the meeting.

3 Team Communication Plan

Our primary form of fast, short communication will be through our text group chat. We will use GitHub to post issues and host documentation for deliverables. In addition to Github, we will also use our meeting minutes for project management and task tracking. Lastly, we will use Teams to host virtual meetings and meeting minutes.

4 Team Member Roles

Each member has been designated as a lead for the various features/topics of our project, however, each member should have some knowledge of every aspect. Team members should be prepared to be flexible and give support to whichever aspect/feature has a current need.

Table 2: Team Member Roles

Role	Primary Lead	Support
Embedded Systems Design	Elsa	Abi
Documentation/Latex/"Faux Marker"/Git	Steffi	Elsa
Lock Frame Design (CAD)	Stephen	Steffi
Software (App Development)	Anthony	Abdul
Software (Wireless Communication)	Abi	Anthony
Lock Mechanism Design (CAD)	Abdul	Stephen

5 Workflow Plan

5.1 Code Development

Team members are expected to use the team repository on Github for code development. The master branch will be used as the current working copy of the code. To develop code, they will fork the main branches to create sub-branches that can then be merged back into the main branch according to the rules outlined below.

5.2 Rules for Merging

Merging will take place under the following branch protection rules:

1. The main branch can only be merged into and not committed to directly.
2. Merging pull requests are disallowed when tests fail.
3. Pull requests are required before merging such that one other group member must approve the code.
4. Status checks and actions must pass before merging.
5. Branches must be up to date before merging to reduce merge conflicts.

5.3 Commits

Commits should take place as often as possible, preferably for every 50 lines of code, or per documentation task. Commit messages must be specific, concise and descriptive.

5.4 Workflow Process

Our workflow process will be as follows:

1. Create a detailed and structured plan for software development
2. Pull any new changes from the master branch

3. Create a new branch to develop on
4. Implement any independent modules
5. Perform unit testing on the independent modules
6. Push the changes made to the new branch
7. Implement any dependent modules
8. Perform unit testing on the dependent modules
9. Merge all changes to the main branch after a pull request is approved by another group member

5.5 Issue Tracking

GitHub *Issues* will be used to track bugs and for accountability purposes. An issue will be created when a developer is unsure about their path forward or if a bug is detected in the code. Issues will be assigned to specific team members explicitly. When an issue is detected and the team member is not able to resolve it themselves, the team lead for that issue will review and advise. Issues can also be raised if a team member foresees a conflict or problem with merging as well as for general tasks.

They will be categorized according to the following labels:

- Bug: Code that is not working properly
- Documentation: Improvements or additions to the documentation
- Help Wanted: Extra attention is needed
- Wontfix: This will not be worked on
- CAD: A task related to CAD in this location
- Wireless Communications: A task related to wireless communications (i.e., Bluetooth, GPS) in this location
- App: A task related to an application in this location
- Embedded Systems Design: A task related to a microcontroller in this location
- Stretch Goal: A task related to a stretch goal in this location
- Template: A task related to a template

5.6 Milestones

Milestones will be created for each deliverable to keep team members accountable. They will be added continuously as necessary. These milestones will be recorded and tracked in the meeting minutes.

6 Proof of Concept Demonstration Plan

Our project requires a large amount of research and development; therefore we must be able to prove that our concept ideas will function as desired and our goals are accomplishable. We will prove our concept with a demonstration in November 2022. For our proof-of-concept demo, we will produce a prototype of our locking mechanism as well as a crude model of the circuitry that will be required to unlock our mechanism with technology. We will demonstrate that we are able to complete our objectives with technologies that currently exist and that we can understand, and that our physical design will be able to function as desired. Our demo will be a primary prototype/model and not a final refined version.

The largest risk we foresee in our project development is our lack of knowledge of wireless communications. To complete our project, we first must understand which technologies exactly we must use and implement, then learn how to use those technologies, and finally implement those technologies specifically for our project. This will have a steep learning curve, and we must manage our time effectively in order to complete our project on time.

Our demo will need to include various parts to demonstrate each goal for our product, that with some refinement and revisions will be able to fulfill each of our requirements. In our demo, we will need to show the software, hardware and mechanical functionality of our product with models and prototypes. We will show our code/software through a basic UI that can be utilized on both iOS and Android devices. To demonstrate our hardware, we will show our selected component and prove that we can accomplish wireless communication as this is a primary requirement for our product to function. Additionally, we will create a sample circuit which shows the basic functionality of our product and helps to calculate the power requirement that will be necessary. Finally, to prove that the mechanical design will be functional we will demonstrate the locking mechanism by creating a CAD drawing and prototype it by 3D printing. While this is not the only physical component of our product it will be the most technical, therefore, achieving this milestone is great proof that we will be able to prototype our whole product. We will also need to test that all the components and devices will be able to communicate and work as an ensemble. Therefore, the demonstration will touch on how each component will work together to function using the required inputs and giving the desired outputs.

7 Technology

The mobile application will be developed in the Visual Studio Environment using the Flutter framework to allow for cross-platform development. Within Flutter, the development language used will be Dart and a DartLint will be used to maintain the code throughout its evolution. Finally, for the mobile application development/testing, XCode will be used to emulate iOS devices and Android Studio will be used to emulate Android devices.

The wireless connection between the mobile application and the lock will be accomplished using the Arduino – Nano 33 BLE, via Bluetooth connection. The coding will be completed in an Arduino IDE using C/C++ languages.

For simple prototyping and early models, the team will 3D print components as it is cheaper and the team has extensive knowledge of the software. Once the model begins to be finalized, parts will be laser-cut and machined for higher-quality testing. During the prototyping and testing phases, the team will also run an FEA analysis on solid works for free and vigorous testing. The model will also be tested in a controlled wet environment for waterproof/weather-resistant testing. Multiple sensors, small electromagnets, and batteries will also be tested during this phase. Other miscellaneous parts and tooling will be completed in the Hatch Centre as team members have access to the tools and resources there. The team has decided to complete all documentation using Latex and will have them posted on the team Git.

8 Coding Standard

The coding style will follow camel case for local variables (lowercaseFirstUpperSecond). Global variables will follow a similar same style, however, the first word will also start with an uppercase letter. Constants and macros will be in full capitals. The code will be modular with no function existing that is longer than 20 lines of code.

Each function will be responsible to complete a single task to enhance testing and reliability. The functions will also be well commented above the respective function declaration. Additionally, functions will start with an action verb to describe exactly what the purpose of that function is. For linting, DartLint will be used for enforcing Dart styles and conventions.

9 Project Scheduling

Our project, work, deliverables, deadlines, and dates will be managed and scheduled accordingly so that our team will be able to complete all tasks and requirements before their due dates. Our team will have twice-weekly meetings in which we will discuss the upcoming dates and deadlines and our goals and strategies to manage them. To manage all our work, we will use various software including GitHub milestones and our meeting minutes. We will post our required tasks and subprojects that need to be completed into GitHub milestones. The GitHub

milestones will be associated with a person, a required date of completion, and a set priority. Using this we will be able to see all open issues for our project and be able to easily delegate them to get our work done.

Our major milestones are outlined in the course deliverable schedule, with the Proof of Concept Demo, Revision 0 Demo, Final Demo, & Final Documentation being the major milestones.

When a larger issue arises, or we are going to tackle a larger problem we will need to be able to split these issues into smaller more manageable ones. To decompose these large tasks, we will modularize our code/functions, utilize small commits, and be intentional with our creation of subbranches. Breaking things up in terms of the skills, technologies, and knowledge required to complete each task. We will all need/be able to work on each issue so it is important to decide who will do what for our team. Each member has different skills and interests, meaning tasks will be split in terms of people's preferences. When tasks can not be split easily, they can be assigned by the team lead. Time will need to be managed by all team members knowing that everyone has a lot of other requirements and priorities.