# Embedded control systems EV Protective Case Project Plan Rough Draft

# Group 8 Holly Iles, Dominic Wenger, Alex Riester

**Project Plan Format**

* [Background](#_heading=h.30j0zll)
* [Project organization](#bookmark=id.1fob9te)
* [Work Breakdown](#bookmark=id.tyjcwt)
* [Hardware and software requirements](#bookmark=id.1t3h5sf)
* [Monitoring and reporting mechanisms](#bookmark=id.lnxbz9)
* [Risk Analysis](#bookmark=id.2s8eyo1)

* [Project schedule](#bookmark=id.3rdcrjn)
* [Process Flow Diagrams](#bookmark=id.35nkun2)
* [Appendix](#bookmark=id.44sinio)

# Background

Our software will operate a device that will keep the capacity of Electric Vehicle batteries protected and allow for EV batteries to be transported more efficiently. There will be a climate control system that will keep the battery stored in its optimal temperature range of 40-80 degrees Fahrenheit, either through means of electrical heating to prevent capacity drainage through the cold, or an air filtration system that moves air in and out of the case to avoid air stagnation. There will also be a critical failure system in place that will activate if the EV battery experiences an impact, or a fire breaks out inside the device that will release a fire suppression system that will neutralize the toxic fumes and flames that can be generated from these critical events. While we will not have access to the hardware equipment necessary for this project, we were planning to test out software logic protocols with data values stored in database files that we would use to simulate situations that the software would encounter like temperature variations, moisture entering the box or critical safety failures like an impact or fire.

# Project organization

* Alex Riester
  + Database manager (SQLite)
  + Python Coder assistant
  + Head of Research
* Dominic Wenger
  + Head of Testing and Debugging
  + Database assistant (SQLite)
  + Python Coder Assistant
* Holly Iles
  + UI/UX Manager
  + Chief Python Coder
  + System Architect

# Work breakdown

1. Create a graph of temperature inside the case over time: **Displays graph but only every hour over a 50-hour period**
   1. Record temperature value
   2. Record datetime for Temp value
   3. Display graph showing the Temp values over a 24-hour period in 30 min intervals
   4. Activate heating or cooling system as needed
2. Create display showing if water has entered the case and if it needs maintenance or not: **This was complete, and the window provides instructions on how to deal with the water**
   1. System will continue normally if container is dry
   2. System will close container and send/display a notification if container is wet
3. Create a display showing if an impact above a certain threshold has occurred: **This was complete, and the system will show a graph if a critical impact is not detected.**
   1. System will close container and send/display a notification of impact
   2. System will continue normally if no impact is detected
4. Create a critical failure system that in the event of: **This was not complete**
   1. Temperature value above a certain threshold
   2. An impact value above a certain threshold
5. Create a setting to clear data to free up room in memory: **This was complete**
6. Test the Software: **This was complete**
   1. Individual implementation
   2. Final overall implementation

# 

# Hardware Software Requirements:

1. Software Requirements**:** Python application developed using VS Code. Matplotlib is required to download to run these applications graphing feature see source: <https://matplotlib.org/stable/users/installing/index.html>.
2. Operating System: While Python can run on most operating systems it is important to ensure that the systems used to run this product are Python compatible.
3. Driver Compatibility: Because our product will be receiving data from specialized hardware sensors it is imperative that the hardware drivers can communicate with our software properly.
4. Packages that need to be downloaded to run: sqlite3 to access database, matplotlib see source above, tkinter to create graphic interface with program.

Hardware Requirements:

1. Primary Storage: Any device running his software must be able to contain data in the form of a DB file and have enough space to hold 24 hours’ worth of data.
2. Peripherals: This software product requires that the sensory equipment used for the container can send information in a format that is compatible with the software.
3. Power Supply: This software product requires a power supply that can run for at least 24 hours or longer and is portable enough to be moved with the container

# Risk Analysis

Risk Identification:

1. The power supply may fail or run out of power. This would be bad because the software not being able to perform any of the requirements including detecting and stopping critical failure events like a fire occurring inside the case.
2. There is the potential for sensory equipment to fail. This could lead to the sensors reporting bad data that could compromise the safety of the EV battery or could lead to the inability of the software to perform the identified requirements.
3. The heating or cooling system could fail. This would lead to the damage of the EV battery that is being damaged or stored by damaging the capacity that the battery can produce.
4. The heating or cooling system could inadvertently injure someone. While the risk is quite low, if someone were to open the container while the system was operating there is still a chance that the systems responsible for maintaining the appropriate temperature range could injure that person in some capacity.
5. There is the potential for the EV battery to be tampered with or stolen. Our software does not prevent the container being controlled from being open due to not wanting to limit access to the battery inside the container by workers interacting with the batteries. We feel that the risk of theft is not high enough to constitute adding this feature into the product.

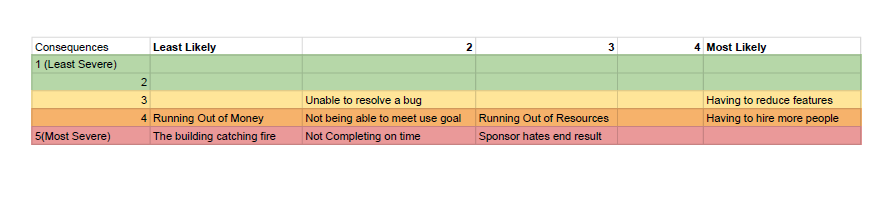
**Risk Analysis:**

In order of descending likelihood of occurring are:

1. The most likely risk is that the power supply would fail, and that the software would not be able to operate. This is the more likely to happen than any of the other risks described above and has the potential to cause the most damage to people or property due to not being able to prevent a critical event from occurring.
2. The second most likely risk that will be encountered is that the sensory equipment would fail. The equipment used to measure impact, temperature, and especially moisture content all must be properly calibrated and tested to ensure that they are working properly. Once again, this risk also carries the danger of not being able to prevent a critical failure event or damage to the battery.
3. The third most likely risk to occur would be the hardware for the heating and cooling system failing. This hardware is critical to ensuring that the battery is not damaged and that if the battery is damaged that no one is injured as a result. This hardware is less likely to fail then the sensory equipment failing and can be fixed if the sensory hardware was still operational and therefore presents less of a risk.
4. The next risk most likely to occur would be the theft of the battery the software was protecting from damage. This is not highly likely to occur because the batteries themselves are very heavy and hard to transport in the first place but because we do not have a security system in the software there is still the possibility of theft or tampering to occur to the battery.
5. The least likely identifiable risk to occur would be that someone is injured because of the heating or cooling system in the container. This is not very plausible because none of the equipment used could harm someone unless they were actively trying to work on the container as it was running. However, because there is not system to prevent the case from being opened by anyone it is possible just not highly likely.

**Risk Planning:**

1. To account for a failure in the power supply the only real preventative measure that can be implemented is having a backup power supply. Our team should consider having a protocol in place in case of power supply failure and having to restart.
2. Later, in the system evolution our team should consider adding reminder notifications to calibrate and test sensory hardware every 3 months or so to ensure accuracy.
3. Again later, in the system evolution our team should consider adding reminder notifications to calibrate and test heating and cooling hardware every 3 months or so to ensure accuracy.
4. While the risk of theft is not zero at this point, I do not see adding any security measures. Maybe later on in the system evolution we might consider adding validation measures like a password but only if it proved necessary.
5. Again, while the risk of someone being injured by the heating or cooling system is not zero at this point, I do not see adding any security measures to ensure that the case have the heating and cooling system is not running while the case is open. Maybe later on in the system evolution we might consider adding these measures but only if it proved necessary.



# 

# Project Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Team Member | Date Started | Date Completed |
| 1.1- | Evie | 4/15/2022 | 4/22/2022 |
| 1.12 | Evie | 4/17/2022 | 4/18/2022 |
| 1.3 | Evie | 4/18/2022 | 4/20/2022 |
| 2.1 | Alex | 5/12/2022 | 5/13/2022 |
| 2.12 | Evie | 4/16/2022 | 4/17/2022 |
| 2.13 | Dominic | 4/14/2022 | 4/14/2022 |
| 2.2 | Evie | 4/21/2022 | 4/22/2022 |
| 2.21 | Alex | 4/22/2022 | 4/22/2022 |
| 2.3 | Dominic | 4/25/2022 | 4/26/2022 |
| 3.1 | Dominic | 4/26/2022 | 4/26/2022 |
| 3.12 | Evie | 4/20/2022 | 4/20/2022 |
| 3.2 | Alex | 5/1/2022 | 5/2/2022 |
| 3.3 | Evie | 5/1/2022 | 5/1/2022 |
| 3.4 | Alex | 5/15/2022 | 5/15/2022 |
| 4.1 | Dominic | 4/15/2022 | 4/16/2022 |
| 4.12 | Evie | 4/22/2022 | 4/23/2022 |
| 4.2 | Alex | 5/3/2022 | 5/9/2022 |
| 4.3 | Evie | 4/22/2022 | 4/22/2022 |
| 5.1 | Alex | 5/5/2022 | Not complete no critical protocol |
| 5.2 | Alex | 5/5/2022 | Not Complete no critical protocol |
| 6.1 | Dominic | 5/15/2022 | 5/15/2022 |
| 6.12 | Alex | 4/12/2022 | 4/13/2022 |
| 6.2 | Alex | 5/14/2022 | 5/15/2022 |

# Monitoring and Reporting Mechanisms

# The only monitoring mechanisms in place for this program are mostly local reports displaying if any abnormality in the routine protocol has been disrupted. The monitoring is conducted by the sensory hardware that keeps track of abnormal temperature values, water levels, and impact incidents.

# Process Flow Diagrams

Diagram

Description automatically generatedGraphical user interface, application, Word

Description automatically generated

# 

# Appendix

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Work Milestones** | **Time Estimate** |
| 1.1 | Create a GUI that will direct users to the various features of the container | **1 week** |
| 1.12 | Create Buttons to guide user to specific features | 1 day |
| 1.3 | Test to ensure successful implementation | 2 days |
| 2.1 | Create plot graph using Matplotlib | 2 days |
| 2.12 | Create conditional statement to ensure temperature value is normal | 1 day |
| 2.13 | Create SQLite command to record Temperature values and datetime values | 1 week |
| 2.2 | Create Button to return to main menu | 1 day |
| 2.21 | Create a command to activate the heating or cooling system based on the temperature value at that time | 2 days |
| 2.3 | Test to ensure successful implementation | 2 days |
| 3.1 | Create conditional statement to ensure moisture has not been detected in the container | 4 days |
| 3.12 | Create a command to send a notification and display a message if moisture is detected in the container | 1 day |
| 3.2 | Create SQLite command to record Boolean values of if moisture is in the container or not and to record the datetime values | 1 week |
| 3.3 | Create Button to return to main menu | 1 day |
| 3.4 | Test to ensure successful implementation | 3 days |
| 4.1 | Create conditional statement to ensure an impact has not been detected in the container | 2 days |
| 4.12 | Create a command to send a notification and display a message if an impact is detected in the container | 2 days |
| 4.2 | Create SQLite command to record Boolean values of if an impact has been detected in the container or not and to record the datetime values | 1 week |
| 4.3 | Create Button to return to main menu | 1 day |
| 5.1 | Create a conditional statement that will determine if a critical event in the form of a Fire or Critical impact has occurred | 2 days |
| 5.2 | Create a command to activate the critical failure system and close the case | 2 days |
| 6.1 | Create a DB file with tables that contain test values for all features in the program are working correctly | 2 days |
| 6.12 | Create another DB file with Tables that will collect test values and will serve as the storage file for the software product | 2 days |
| 6.2 | Combine all software features into one project | 2 weeks |