**T.C.**

**BAHÇEŞEHİR UNIVERSITY**

****

**FACULTY OF ENGINEERING AND NATURAL SCIENCES**

**CAPSTONE PROJECT PROPOSAL**

**The TITLE of the project goes here**

2104867 Mohmad Yaseen Alwattar Energy System Engineering

1902463 İzem Azra Akyol Energy System Engineering

2105125 Doğa Yeşilyurt Energy System Engineering

2002049 Mete Özeren Software Engineering

2101988 Tuna Ertürk Software Engineering

**The Names of the Students and their Departments go here,**

**In Alphabetical Order by Surname,**

**Capitalize The First Letters Only**

**Advisors:**

**Assist. Prof Nezihe Kucuk Yıldıran Department of Energy System Engineering**

**Assist. Prof Ece Gelal Soyak Department of Computer Engineering**

**ISTANBUL, December , 2024**

**STUDENT DECLARATION**

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that

* they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
* they have not received unpermitted aid for the project design, construction, report or presentation;
* they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

# ABSTRACT

The TITLE of the project goes here

Your Names in alphabetical order by surname

Faculty of Engineering and Natural Sciences

Advisor:

Assist. Prof Nezihe Kucuk Yıldıran Department of Energy System Eng.

Assist. Prof Ece Gelal Soyak Department of Computer Eng**.**

May 2024

Write the abstract here (only necessary for the final report).

**Key Words**:

# TABLE OF CONTENTS

ABSTRACT iii

TABLE OF CONTENTS iv

LIST OF TABLES v

LIST OF FIGURES v

LIST OF ABBREVIATIONS vi

1. OVERVIEW 1

1.1. Identification of the need 1

1.2. Definition of the problem 1

1.3. Conceptual solutions 1

1.4. Physical architecture 3

2. WORK PLAN 4

2.1. Work Breakdown Structure (WBS) 4

2.2. Responsibility Matrix (RM) 4

2.3. Project Network (PN) 5

2.4. Gantt chart 5

2.5. Costs 7

2.6. Risk assessment 8

3. SUB-SYSTEMS 9

3.1. The name of the sub-system 1 9

3.2. The name of the sub-system 2 10

4. INTEGRATION AND EVALUATION 11

4.1. Integration 11

4.2. Evaluation 11

5. SUMMARY AND CONCLUSION 12

ACKNOWLEDGEMENTS 13

REFERENCES 14

APPENDIX A 15

APPENDIX B 16

# LIST OF TABLES

[Table 1. Comparison of the three conceptual solutions. 2](#_Toc64236119)

[Table 2. Responsibility Matrix for the team 4](#_Toc64236120)

[Table 3. Gantt chart for the materialisation phase of the project. 6](file:///C:\Users\andrewjohn.beddall\Desktop\0_Capstone\NewTemplates\CapstoneReportTemplate.docx#_Toc64236121)

[Table 4. Costs 7](#_Toc64236122)

[Table 5. Risk matrix 8](#_Toc64236123)

[Table 6. Risk assessment 8](#_Toc64236124)

# LIST OF FIGURES

Figure 1. Path of electrical energy from the power plant to the home. 2

Figure 2. Interface diagram for the system. 3

Figure 3. Process chart for the system 3

Figure 4. Work breakdown structure for the project. 4

Figure 5. The project network. 5

# LIST OF ABBREVIATIONS

IoT Internet of Things

M2M Machine-to-Machine

IEEE The Institute of Electrical and Electronics Engineers

# 1. OVERVIEW

Begin the first paragraph here.

Begin the second paragraph here.

## 1.1. Identification of the need

Efficiency and safety are the essence of the requirement for a Battery Management System (BMS) in the Direct Current (DC) charging station powered by solar panels. Solar panels ensure clean, renewable energy, but it is a variable source of energy because of variations in weather conditions and times of day. It becomes quite tricky to directly supply a steady power flow to an electric vehicle using such an unpredictable source. This is where a Battery Management System (BMS) can bridge that gap and store energy. When solar generation is high, the Battery Management System (BMS) helps to store excess energy and regulate energy delivery during peak demand periods, always providing electric vehicles with reliable and efficient charging, regardless of variation in solar supply.

A Battery Management System (BMS) is also crucial for the health of the battery system that would be set up with this charging station. In these two states, batteries undergo various forms of chemical and physical stresses; if poorly managed, such batteries could degrade faster or even develop safety risks. The Battery Management System (BMS) observes all the parameters of the batteries, like temperature, state of charge, and state of health, to avoid overheating or overcharge issues, which provide safety and performance throughout their life.

This will help further make the Direct Current (DC) charging station sustainable, too, because of its integrated Battery Management System (BMS) within it. As electric vehicles are increasing in demand, the need for reliable renewable charging stations will further minimize carbon emissions and safely meet our energy demands. A solar-powered charging station with a Battery Management System (BMS) improves the utilization of renewable energy by efficiently storing the generated power from solar panels. This would result in improved energy availability for electric vehicles and a low-carbon energy system.

## 1.2. Definition of the problem

A Battery Management System (BMS) is crucial for a solar-powered Direct Current (DC) charging station because it manages both the incoming solar power and the battery’s overall condition. Solar energy production varies with the weather and time of day, leading to times when there’s either too much or too little energy for electric vehicle (EV) charging. The Battery Management System (BMS) stores extra energy when there’s a lot of solar power and ensures there’s enough energy to charge vehicles when solar power is low.

One of the Battery Management System's (BMS) main purposes is to track the battery’s State of Charge (SoC), and State of Health (SoH), State of Charge (SoC) indicates how much charge is left, while State of Health (SoH) tracks the battery’s long-term health, showing how well it holds charge over time. Accurately measuring SoC, and SoH, can be tricky but is essential to prevent overcharging or undercharging. If this isn’t done correctly, it can harm the battery’s lifespan and performance.

Safety is another critical reason for using a Battery Management System (BMS). Direct Current (DC) fast charging creates heavy loads on batteries, as their capacity can be drained to a state of almost full discharge in a very short period, causing them to heat up. The Battery Management System (BMS) is tuned to continuously observe temperature and current levels to avoid overheating or any other imminent safety risks such as thermal runaway. Thus, by balancing the flow of incoming solar power, energy storage, and charging demands judiciously, a Battery Management System (BMS) maintains the entire station's operational safety and efficiency for reliable power delivery to electric vehicles (EVs).

|  |  |  |
| --- | --- | --- |
|  | Functional Requirement | Description |
| 1 | Battery Monitoring | Measure cell, module, and pack voltages; monitor current and temperature to ensure system safety. |
| 2 | State Estimation | Calculate State of Charge (SoC), State of Health (SoH), for battery management. |
| 3 | Safety Management | Detect and respond to over-voltage, under-voltage, over-current, and temperature anomalies to prevent faults. |
| 4 | Communication and Data Management | Enable communication between modules and the master controller; log and transmit real-time data for analysis. |
| 5 | Balancing and Control | Equalize cell voltages, manage charge/discharge cycles, and adapt to dynamic load and solar conditions. |
| 6 | Scalability and Modularity | Support expansion by adding modules seamlessly while ensuring independent module operation. |
| 7 | Fault Tolerance and Isolation | Isolate faulty modules automatically; ensure uninterrupted operation of the remaining system. |
| 8 | Efficiency and Low Power Consumption | Minimize power loss in the BMS while ensuring efficient operation under varying solar input conditions. |
| 9 | Integration with Solar and DC Systems | Coordinate with solar power input and DC charging station requirements; protect against voltage surges. |
| 10 | Compliance and Standards | Meet safety and operational standards; use standard communication protocols |

### 1.2.1. Functional requirements

A modular BMS aims at safe and efficient operation of the battery packs in applications like solar-powered DC charging stations. The system will continuously monitor vital parameters such as cell voltages, module temperatures, and current flow to maintain battery health and prevent faults related to overcharging or overheating. It applies advanced techniques in estimating the SoC, SoH, and SoP possible, hence accurate control and predictive maintenance. The modular design also allows for cell balancing with a uniform voltage across all cells for optimum performance and long life. Besides its safety and performance features, the modular BMS is very scalable and easy to expand by adding more modules.

Faults will be confined to the level of a module, and the system will keep on working while repairs are carried out. Module controllers communicate through a central unit so that information is fully shared and operation is unified. This architecture also supports integration with solar systems, where features include dynamic adjustment of charging operations according to changing solar input. The system interfaces via any standard communication protocols, such as CAN bus or I2C, guaranteeing compatibility and reliability for a wide range of uses.  
  
**For cell balancing**

Active Balancing in Modular BMS for Solar DC Charging Station In a solar-powered DC charging station, the battery pack consists of cells organized into modules and relies on efficient cell management. Active balancing handles the natural variation in the characteristics of cells due to manufacturing, aging, or other external conditions. By giving back extra energy from the higher-voltage cells to the lower-voltage ones, the system maintains the same voltage or state of charge across the pack. The method involves constant voltage monitoring, which has been developed into inductive balancing for high efficiency and scalability. Energy transfer occurs across cells or modules using DC-DC converter circuits that maintain reliability to avoid such problems as overcharging during periods of peak energy from the sun.

The modular design of the BMS provides enhanced active balancing, with the ability for each module to function independently while communicating to a main controller. Feedback loops prevent overcompensation adjustment, maintaining balance during charging and discharging. Unlike passive systems, which work by dissipating excess energy as heat, active balancing conserves energy by circulating it within the pack, reducing stress on individual cells. This approach is crucial for handling the variability of solar input, ensuring uniformity across the battery pack, extending its lifespan, and supporting efficient EV charging operations.

A diagram of a diagram of a few green containers

Description automatically generated with medium confidence

Source:<https://www.mokoenergy.com/active-balancing-bms/>  
Chowdhury, S., Shaheed, M. N. B., & Sozer, Y. (2021). State-of-charge balancing control for modular battery system with output DC bus regulation. *IEEE Transactions on Transportation Electrification*, *7*(4), 2181-2193.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Description** | **Use Case** | **Advantages for SoC** | **Limitations for SoC** | **Advantages for SoH** | **Limitations for SoH** |
| Coulomb Counting | Integration of charge and discharge currents to calculate SoC and SoH. | Battery charging stations, electric vehicles. | - Real-time SoC tracking.  - Simple and low-cost hardware requirements. | - Error accumulation:accuracy may degrade over time.  - Requires accurate initial SoC measurement. | -Simple SoH calculation.  - Low-cost and quick implementation. | Battery use and environmental factors affect accuracy. |
| Open Circuit Voltage (OCV) | SoC and SoH estimation by measuring the open-circuit voltage of the battery. | Low current battery testing and monitoring. | - Accurate SoC estimation under idle conditions.  - Low-cost. | - Not suitable for real-time measurements.  - Requires the battery to be at rest for accurate results. | -Voltage drop reflects capacity loss.  - Simple to apply. | -Provides limited information on overall battery health.  - Requires a rested battery. |
| Sleek Dual Kalman Filter | State estimation for SoC and SoH by considering noise and uncertainty. | High-accuracy applications (electric vehicles, energy storage). | - High accuracy.  Compensates for noise and uncertainty in measurements. | - High computational complexity.  - Sensitive to model parameters. | - High-accuracy SoH estimation.  - Takes system dynamics into account. | - Model parameters need to be accurately defined.  - High computational requirements |
| Internal Resistane | Estimates SoH by measuring the increase in internal resistance over time. | Battery health monitoring and maintenance. | - Real-time monitoring is possible.  - Easy to integrate into existing systems. | - Requires precise resistance measurements.  - Noise in measurements may affect | - Increase in internal resistance indicates battery aging.  -Simple to implement. | - Requires high-precision measurements.  - Measurement noise and environmental factors may affect accuracy. |
| Capacity Monitoring | Monitors capacity degradation over time to estimate SoH. | Long-term battery health monitoring. | - Simple and direct indication of capacity loss. | - Requires long-term monitoring.  - Not suitable for real-time SoC estimation. | - Directly shows capacity degradation.  - Simple to apply. | -Long-term tracking is necessary.  - Low usage may not show significant capacity loss. |

Methods For measuring SoC and SoH :

### Source of table: Source of table: Pesaran, A. A., Keyser, M., et al. (2012). *State of Charge Estimation for Lithium-Ion Batteries Using Coulomb Counting.* *Journal of Power Sources*.

### 1.2.2. Performance requirements

**A white sheet of paper with black text

Description automatically generated**

Several key performance indicators are used to monitor the performance of a BMS in solar-powered DC charging stations. These include voltage, current, and temperature monitoring that ensure the operating of batteries within the limit of safety. Voltage monitoring prevents damage by not letting each cell overcharge or undercharge, while current monitoring prevents overheating. Temperature control keeps the battery from getting too hot or cold during charging and discharging.

In addition, the BMS monitors the SoC and SoH of the battery. SoC gives the amount of charge remaining in the battery, while SoH reflects the general health of the battery in terms of holding charge over time. These are important in optimizing charging and discharging and ensuring the battery lasts as long as possible.

Other key performance indicators are fault detection and cycle life tracking. The BMS will check any malfunctions of individual cells or modules and then inform the system so it can take remedial measures. Tracking cycle life offers insight into when to expect replacement of the battery in order to ensure reliable operation of the charging station throughout its lifetime. The monitoring of all these parameters is an assurance that the system operates effectively, safely, and is long-lasting.

**Source of table : Harris, A., Cox, R., & Nasipuri, A. Technical Report: Battery Modeling and Performance Metrics.**

**Omar, N., Hegazy, O., Mulder, G., Bossche, P. V., Mierlo, J. V., & Timmermans, J.-M. (2022). *A review on battery management systems and its application in electric vehicles*. IEEE Xplore. https://doi.org/10.1109/9708114**

### 1.2.3. Constraints

**1. Economic Limitations**

The investment cost is a big concern with the installation of a modular BMS on a solar-powered DC charging station. Its components such as sensors, microcontrollers, and active balancing circuitries are really expensive to implement. Most of them use lithium-ion batteries with greater energy density and are consequently much more costly upfront. Ongoing operations expenses maintenance, recalibration, and eventual replacement also have financial tolls. Therefore, the performance optimization should not overlook the affordability of the system to make it viable for wide applicability cost-effectively.

**2. Environmental Constraints**

Although the solar-powered systems are ecologically friendly, the manufacture and disposal of lithium-ion batteries pose an environmental threat. The mining process for raw materials such as lithium, cobalt, and nickel destroys ecosystems and contributes to pollution. If not disposed of properly, toxic chemicals in batteries can be released into the environment. For minimal environmental impact by the system, energy efficiency should be ensured in the BMS design; extending the life of the battery minimizes waste and ensures sustainability.

**3. Social Constraints**

Solar-powered DC charging station adoption also faces challenges in the form of public acceptance and trust. Concerns ranging from overheating to fire risks regarding battery safety need assurance through appropriate safety mechanisms that are reliable. The system must also be in conformity with local regulations and in harmony with the existing infrastructure. Educating the public on the benefits and reliability of renewable energy systems is key towards gaining broader acceptance and support.

**4. Technical Constraints**

The type of battery determines the safety, cost, and efficiency of the whole system. As an instance, lithium-ion batteries, due to their high energy density, see wide use in renewable systems. Nevertheless, they also have hotspots of overheating or degradation if operated under unfavorable conditions. A BMS must, therefore, make sure that safe operation for a long duration is considered with strict monitoring and control:.

Solar is a variable energy source that fluctuates with the weather and time of day. The BMS should have sophisticated algorithms to cope with these fluctuations effectively for non-wastage of energy and non-overloading of batteries. Proper solar integration will make the system store energy when sunlight at peak hours provides maximum output and release continuously during periods of low solar output.

With the growth of the system, scalability is a major demand. The BMS must be in a position to handle additional batteries without losing efficiency. Modular designs, together with good communication between modules, ensure that the system is able to grow without a loss in performance. Similarly important is the accurate state estimation, including precise SoC and SoH measurement. This calls for advanced sensors and algorithms to track the performance of the battery and predict possible failures.

Among these, thermal management is particularly critical because batteries have the tendency to overheat under high rates of charge/discharge. The BMS should be monitoring the temperatures constantly and should trigger control systems for cooling in cases of need to avoid any damage. Besides, fault detection in real time-including imbalances in cell strings, short circuits, and abnormal voltage spikes-is also very important to keep the system safe and prevent further deterioration of the battery packs.

Good communication between the BMS and the solar panels, extended to the electric vehicle, will ensure proper coordination in functionality. Delays or any breaks in transmitting information lead to inefficiencies in operations or even unsafe working conditions. A good network in communication ensures all items operate in harmony to preserve system efficiency and safety across all conditions.

**5. Operational Constraints**

Energy loss is inherent in a battery system, especially when the process of charge redistribution is considered. Even when active balancing is applied, some energy will be dissipated to the environment in one form or another. The system should be so designed that such losses become minimum to ensure that maximum energy from the solar panel is utilized and wastage is reduced to a minimum.

Maintenance procedures, like the replacement of batteries or sensor recalibration, are generally time-consuming and may interrupt system operations. While regular maintenance is important to maintain performance, it has to be carefully scheduled to minimize any downtime. In addition, the operating environment is also a critical factor. The system must be sufficiently robust to operate reliably in extreme heat, freezing cold, or high humidity, which can lead to component degradation and decreased overall performance.

Another critical aspect of operational efficiency is managing the battery life cycle. As batteries age, their performance degrades over time. The BMS must monitor indicators of aging, such as capacity fade and increased internal resistance, and seamlessly integrate replacement batteries without disrupting the system. During periods of high demand, the BMS must also handle dynamic load management effectively, distributing the load evenly across cells and modules to prevent overloading of any specific components.

DC fast charging brings huge stress on the batteries, which increases the possibility of overheating or voltage imbalances. The BMS should actively monitor such conditions and make real-time adjustments to maintain safety. During idle periods, the BMS must conserve power while remaining ready for the next session, ensuring energy efficiency even when the station is not in use. And lastly, integration or replacement of battery modules should be easy. The BMS should automatically balance the new and existing modules for complete reliability and performance of a system without human intervention.

**6. Regulatory Constraints**

The system should be in conformance with international and local standards, such as ISO 26262 for functional safety or IEC 61508 for industrial safety. These various certifications assure reliability but take a lot of time and resources. If connected to the grid, the station has to follow rules about energy exchange. Different government policies, subsidies, or incentives for renewable energy projects can also influence the financial viability of the system.

**7. Logistical Constraints**

Supply chain issues can be pretty tough, especially when high-quality components such as sensors, microcontrollers, or balancing circuits are in demand. In cases of global shortages or logistical disruptions, the procurement of such parts may delay the timeline of the project. A steady and reliable supply chain is crucial to maintaining the project on schedule and avoiding unnecessary delays.

The additional requirement is the installation. Installing a solar-powered DC charging station normally requires an upgrade in the local electrical infrastructure to match and guarantee the safety of the installation. The upgrade may be required on power lines, transformers, or protective equipment, further increasing the cost and prolonging the installation period.

Battery recycling is a crucial environmental and logistical concern. The proper reprocessing of spent batteries requires adherence to stringent environmental laws and access to specialized recycling facilities. This helps ensure that toxic materials are handled in an environmentally safe and sustainable manner. However, these facilities may not always be available, and the cost of recycling can be relatively high, which has to be managed with care regarding the overall lifecycle planning of the system.

**8. User Constraints**

The users demand speedy charging, which actually stresses the battery. In contrast, the BMS must strike a balance between quick recharging and preventing the degradation of the battery. Simplicity and intuitiveness in the interface design would allow users and operators to intuitively understand and interact with the system. A poorly understood interface could lead to abuse or inefficiency.

9. Cybersecurity Constraints

The system needs communication between the BMS, solar panels, and EVs and hence is vulnerable to any cyberattacks. The above data and communications need critical protection. In case connectivity is very important, through the internet or local network, disruption of connectivity may affect the real-time monitoring and controlling aspects.

10. Time and Project Constraints

Development of a BMS includes rigorous testing in terms of fluctuating solar input or very high demand conditions, which of course stretches the project timeline. Integrating many components from different disciplines and ensuring all work in an efficient way adds complexity and time to the project.

Source:

Rama, V., Muniraj, R., Kohila, J., & Jarin, T. (2024). Solar-based smart EV charging station withsmart battery management system. *2024 International Conference on Circuit Power and Computing Technologies*

## 

## 1.3. Conceptual solutions

1.3.1. Literature Review

Energy is an important factor to maintain a sustainable human life.

Using the correct resources holds a significant value when examining energy issues with sensitivity. It is known that the use of fossil fuels and their negative impacts such as global warming, climate change, pollution of air and water etc. These negative impacts can’t be ignored when pursuing a world that is cleaner and more habitable.

To reduce the negative impacts of fossil fuels, renewable energy sources are used to produce energy for households and many other sectors.

Transportation sector is one of the sectors that are mostly reliable to fossil fuels, therefore impacting the environment more. To reverse this and lessen the use of fossil fuels in transportation sector, electric vehicle technologies are being developed. Alongside these developments, energy storage systems became used more and reached a crucial point (Richardson, 2013). With the rapid increase of said storage systems, lithium-ion batteries became the fastest-growing and developing one out of all types of chemical and physical storage solutions (Aaldering et al.,2019; Andrea, 2010).

As the usage of these systems increased, scientists and companies worked to ensure these systems were more efficient, have more longevity and more secure. For this topic, BMS, with its intelligent algorithms and controllers, contribute to the reliability and efficiency of batteries in the storage systems (Febg et al., 2018). The BMS monitors the parameters like voltage, current, temperature, fault prediction etc. They collect, process and ensure that the system operates in the most efficient format (Wang et al., 2017, 2020)

The EV market share almost shat down at 1930 due to oil discoveries and ICE(Internal Combustion Engine) breakthroughs in vibration and noise.(C.C Chan 2002) Nickel metal batteries developed with high energy densities followed by lithium-ion batteries at 1899 after that in 1977 (LIBs) which is triggered battery usage in EV. In 2008, the first recent pure EV was announced to market by Tesla with 245- mile range based on one charge.(U.J. Thomas 2019). Nowadays EVs have big amount of about 500-mile per charge.(J.Chen 2022)

Different types of lithium ion batteries are diversified according to the chemical components and structures used.(Some common types of lithium ion batteries:

**Nickel Cobalt Manganese (NCM)**: A battery type that provides high performance and long lifespan, commonly used in electric vehicles.

**Lithium Iron Phosphate (LiFePO4)**: A safer and longer-lasting option preferred in electric vehicles and energy storage systems.

**Nickel Cobalt Aluminum Oxide (NCA)**: A battery type with high energy density, commonly used in electric vehicles.

**Lithium Manganese Oxide (LMO)**: LMO (Lithium Manganese Oxide) batteries offer high safety, long lifespan, and low cost, though their energy density is not as high as other types of batteries.

**Lithium Titanate Oxide (LTO)**: LTO batteries offer fast charging, long lifespan, and high safety, but their energy density is lower compared to traditional lithium-ion batteries.

These varieties offer different advantages according to the requirements of the application. For example, LiFePO4 batteries are preferred in terms of safety and longevity, while NCM and NCA batteries are used in applications that require more energy density and performance. (Dahlin, G.R and Storm, K.E,2010)

Lithium-ion batteries provide high energy density, making them small, lightweight, and capable of long-lasting use. They have a long lifespan, typically lasting between 500 to 1500 charge cycles, and they can charge quickly without showing memory effect, which prevents capacity loss. Additionally, they are relatively more environmentally friendly compared to other types of batteries. However, lithium-ion batteries are quite expensive due to the use of rare and costly materials such as lithium, cobalt, and nickel. They are also sensitive to temperature changes; exposure to high heat can pose safety risks and shorten the battery’s lifespan. (Linden, D. And Reddy, T.B.,2002)

Overview of BMS Types

Battery Management Systems (BMS) are intended to guarantee safety, performance, and lifetime for batteries by monitoring and controlling their operation. Below are the main types of BMS architectures:

Centralized BMS:

The single controller controls all the battery cells directly through several wires. Centralized BMS is easy and cheap; however, it is unsuitable for large systems because complex wiring and a single point of failure are concentrated in the wiring harness.

Modular BMS:

The battery pack is split into smaller modules; each module would have its controller. These module controllers all converse with a central unit. It features scalability, fault tolerance, and effective management. Therefore, it is especially suitable for medium to large systems.

Primary Controller-Subcontroller BMS:

The main controller, in hierarchical order, oversees the subcontrollers controlling smaller cell groups. This combines the strengths of centralized and modular systems. This permits a wide latitude of flexibility for complex configurations.

Distributed BMS:

Each cell or small group of cells has its independent controller. It reduces wiring complexity and has a high scalability ceiling but is costly and involves complex communication protocols.

In the case of our system, the modular BMS will be the most appropriate.

Reason for the Selection of Modular BMS

A modular design supports medium-to-large battery packs by allowing easy addition of new modules. This is crucial for a solar-powered charging station, where battery capacity might need expansion.If a module fails, the issue can be isolated to that module without affecting the entire system. This improves reliability and reduces downtime.With localized module controllers, troubleshooting and repairs are easier since problems can be pinpointed to specific modules.Each module will process its own data and send it back to the central controller. This way, all parts of the battery system are monitored and controlled with high efficiency.Modular BMS lends itself very well to dynamic load handling, a common feature in solar-powered systems where charging rates are subject to changes in sunlight.

Centralized and distributed topologies have their benefits, but for a solar-powered DC charging station, this modular BMS strikes a balance for scalability, fault tolerance, and ease of maintenance. Such modular battery management architecture ensures reliable and efficient management, thus proving to be the optimal choice for your application.

### 1.3.2. Concepts

Victron Energy's VE.Bus BMS V2, Small BMS, and Lynx Smart BMS models offer different features for various needs. In terms of temperature, the VE.Bus BMS V2 typically monitors battery temperatures more precisely and takes automatic actions in case of overheating, while both the Small BMS and Lynx Smart BMS only provide an alarm feature, notifying the user when the temperature limit is reached but without any automatic intervention. Temperature-wise, all of these systems generally operate within the 60-70°C range, but higher temperatures could pose a risk to system protection.

Regarding pricing, the Small BMS is the most affordable model, ranging between 2,631 TRY, while the Lynx Smart BMS is priced around 48,836 TRY. The VE.Bus BMS V2 falls in the 7,204 TRY range. For voltage input, the VE.Bus BMS V2 is typically compatible with 48V systems, while the Small BMS and Lynx Smart BMS are flexible and compatible with 12V, 24V, and 48V systems.

In terms of current capacity, the VE.Bus BMS V2 can handle higher currents and is generally used for larger systems. The Lynx Smart BMS provides adequate current capacity for medium-sized systems, while the Small BMS is ideal for systems with lower current requirements.

In conclusion, the VE.Bus BMS V2 is the most powerful option for large and industrial applications, while the Lynx Smart BMS offers a good balance for medium-sized systems. The Small BMS provides the most economical solution for smaller systems, and based on the system's requirements, this BMS model is considered the most suitable.

Table 1. Comparison of the three conceptual solutions.

|  |  |  |  |
| --- | --- | --- | --- |
|  | VE.Bus BMS V2 | smallBMS  BMS400100000 | Lynx Smart BMS 500 |
| Cost | 7.204,39 TL | 2.631 Tl | 48.836,43 TL |
| Complexity | Medium | Easy | Hard |
| Input Voltage | 9 – 70 VDC | 8 – 70 VDC | 9 - 60 VDC |
| Max current | 10 mA | 2,2 mA | 1000 A |
| Temprature | -20 - +50 °C  (0 - 120 °F) | 20 - +50 °C  (0 - 120 °F) | 40 °C - +60 °C |
| Size | 23,8 x 94,5 x 105,5 mm | 106 x 42 x 23 mm | 230 x 180 x 100 mm |
| Weight | 120 g | 0,1 kg | 2,7 kg |

|  |  |  |  |
| --- | --- | --- | --- |
| Aspect | Reference 1 | Reference 2 | Reference 3 |
| Study Title | Badoni, M., Pattnaik, M., Varshney, L., & Gupta, A. (2023, November). Development of Solar-powered EV Charging Station With Energy Management System. | Umair, M., Hidayat, N. M., Ahmad, S. A., Ali, N. H. N., Mawardi, M. I. M., & Abdullah, E. (2024). A renewable approach to electric vehicle charging through solar energy storage. | Fatnani, M., Naware, D., & Mitra, A. (2020, September). Design of solar PV based EV charging station with optimized battery energy storage system. |
| Focus | This study emphasizes the integration of an energy management system (EMS) with solar-powered EV charging, focusing on modularity and real-time energy flow monitoring. | The research highlights the optimization of energy storage systems (ESS) and their integration with PV and EV batteries to maintain a stable and reliable charging infrastructure. | The paper explores the design and performance of optimized battery energy storage systems (BESS) for sustainable and cost-effective EV charging infrastructure. |
| Methods Used | - Sliding mode control to ensure dynamic energy balance. - Modular design for scalability and real-time monitoring. - Integration of PV panels with an EMS for efficient energy dispatch. | - MATLAB simulations to analyze ESS behavior under dynamic conditions. - Predictive algorithms to forecast solar energy and match EV charging demand. - Comprehensive modeling of PV-ESS-EV systems. | - Advanced modeling and simulation tools to design optimized BESS. - Analysis of cost-benefit for renewable integration. - Performance testing through energy flow simulation. |
| Energy Management Approach | Dynamic balancing of PV energy generation with EV load demand, facilitated by sliding mode control systems. | A predictive approach to manage ESS operations, ensuring alignment between solar energy generation and EV charging needs. | Energy dispatch is optimized through advanced BESS design, focusing on cost efficiency and sustainable energy utilization. |
| PV Integration | Incorporation of modular PV panels with EMS to handle solar intermittencies effectively. | Solar PV is modeled to account for real-world conditions like irradiance variability, ensuring seamless integration with ESS. | PV systems are connected with BESS and DC charging stations to maximize renewable energy usage. |
| Battery Energy Storage System (BESS) | An integrated BESS with dynamic real-time control to manage fluctuations in energy supply. | ESS designed to mitigate solar intermittencies, with optimized capacity based on projected load profiles. | BESS configured for minimal energy loss and efficient operation, tailored for scalability in large EV fleets. |
| Simulation/Modeling Tools | - Sliding mode controller for energy dispatch simulations. - Real-time monitoring tools to track system performance. | - MATLAB for predictive simulations of energy flows and ESS behavior under various conditions. | - Simulation tools for evaluating BESS performance and cost-benefit analysis for renewable energy systems. |
| Results/Outcomes | - Improved PV utilization by 20%. - Reduced reliance on grid power by 15%. - Enhanced scalability through modularity. | - PV mismatch losses reduced by 18%. - Increased system reliability during low solar generation. - Stable and predictable EV charging performance. | - Carbon footprint lowered by 25%. - Cost efficiency improved by 12%. - Scalable design supports large EV charging infrastructures. |
| Key Challenges Addressed | - Addressed energy wastage due to solar variability. - Designed for scalability to accommodate growing EV demands. | - Tackled solar intermittencies and mismatch between energy supply and EV demand. - Improved load prediction for stable system operation. | - Focused on reducing carbon footprint and improving the cost-effectiveness of renewable energy systems. |
| BMS Specifications | - Dynamic EMS for real-time energy balancing. - Scalable and modular architecture for future expansion. | - Predictive algorithms ensure efficient ESS charging/discharging. - High energy density to manage peak EV demands. | - Low-loss BESS tailored for high-efficiency renewable integration. - Designed to support low-carbon and cost-efficient energy management. |
| Environmental Impact | Significant reduction in fossil fuel dependency through enhanced PV integration. | Improved resource efficiency by minimizing energy wastage and reducing mismatch losses. | Achieved a 25% lower carbon footprint compared to traditional EV charging systems. |

## 1.4. Physical architecture

Diagram of solar power station

Description automatically generated

Figure 2. Process chart for the system

# 2. WORK PLAN

Begin the first paragraph here.

Begin the second paragraph here.

## 2.1. Work Breakdown Structure (WBS)

**Solar Power DC Charging Station**  
1.1. **System Design and Integration**

-1.1.1. Modeling of System

-1.1.2. Simulation   
- 1.1.1. Solar Panel System integration with Battery Charger   
- 1.1.3. BMS Design ( SOC/SOH Algorithms )  
- 1.1.3.1. Monitoring Systems (Voltage, Current, Temperature)

1.2. **Software Development (NOT SURE YET )**   
- 1.2.1. Communication and Data Management  
- 1.2.1.1. Real-time Monitoring Integration  
- 1.2.1.2. Secure Protocols and Testing  
- 1.2.2. Control and User Interface  
- 1.2.2.1. GUI Development  
- 1.2.2.2. Software Updates and Debugging

1.3. **Testing and Validation**   
- 1.3.2. System Integration Testing (ESE)  
- 1.3.2.1. SOC/SOH Accuracy  
- 1.3.2.2. Thermal Management Validation

- 1.3.2.3. Integration Software

-system integration testing

Figure 4. Work breakdown structure for the project.

## 2.2. Responsibility Matrix (RM)

Table 2. Responsibility Matrix for the team

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Azra | Doga | Yaasin | Mete | Tuna |
| Decision of system components | S | S | R | R | S |
| Modeling of the system | S | R | S | S | S |
| Simulation of the system | R | S | S | S | R |
| Integration and design of BMS | S | S | R | S | S |
| Comm. | S | R | S | R | S |
| Testing and validation | R | S | S | R | R |
| Planning | S | R | S | R | S |
| Reporting | R | R | R | R | R |
| Integration | S | S | R | R | S |

R = Respomsible; S = Support

## 2.3. Project Network (PN)

Begin the first paragraph here.

Begin the second paragraph here.

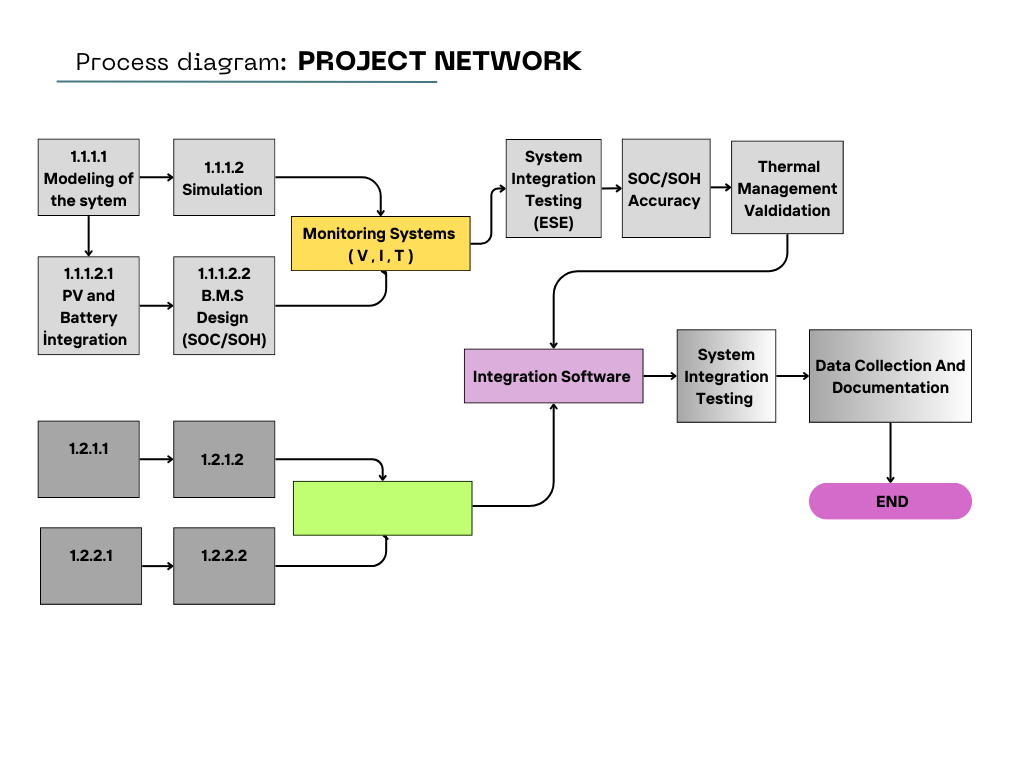


Figure 5. The project network.

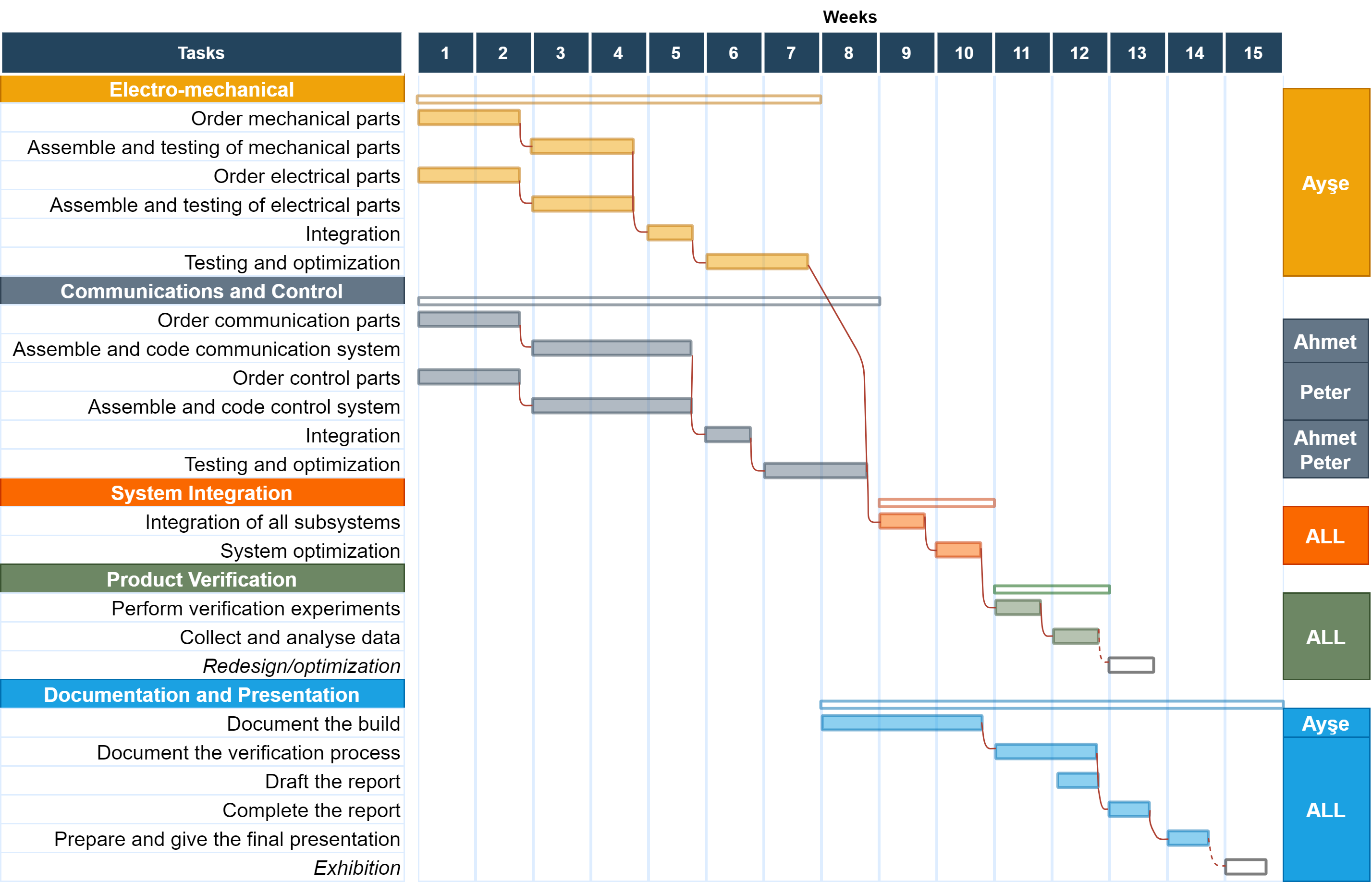
## 2.4. Gantt chart

*Since the Gantt chart covers 15 weeks it can be very wide, therefore in this example it is placed on it’s own page and turned sideways. There are various forms for a Gantt chart, choose the form that is suitable for the size and complexity of your project.*

*Discuss the Gantt chart here (refer to the table).*

Begin the first paragraph here.

Begin the second paragraph here.

**

Created at https://app.diagrams.net/

Table 3. Gantt chart for the materialisation phase of the project.

## 2.5. Costs

Begin the first paragraph here.

Table 4. Costs

|  |  |
| --- | --- |
| **Electro-mechanical** | |
| Arduino Uno | 59.90 ₺ |
| SRC-04 sensor | 19.99 ₺ |
| SG90-RC serve-motor | 21.34 ₺ |
| other materials | 26.90 ₺ |
| **Total** | **128.13** ₺ |
|  | |
| **Communications and Control** | |
| Arduino Nano 328 | 55.39 ₺ |
| Arduino joystick shield | 24.23 ₺ |
| HC06 bluetooth-serial module | 28.99 ₺ |
| **Total** | **108.61** ₺ |
|  | |
| **System Total** | **236.74** ₺ |

## 2.6. Risk assessment

*The tables provided below are examples, you can replace one or both with your own.*

Begin the first paragraph here.

Begin the second paragraph here.

Table 5. Risk matrix

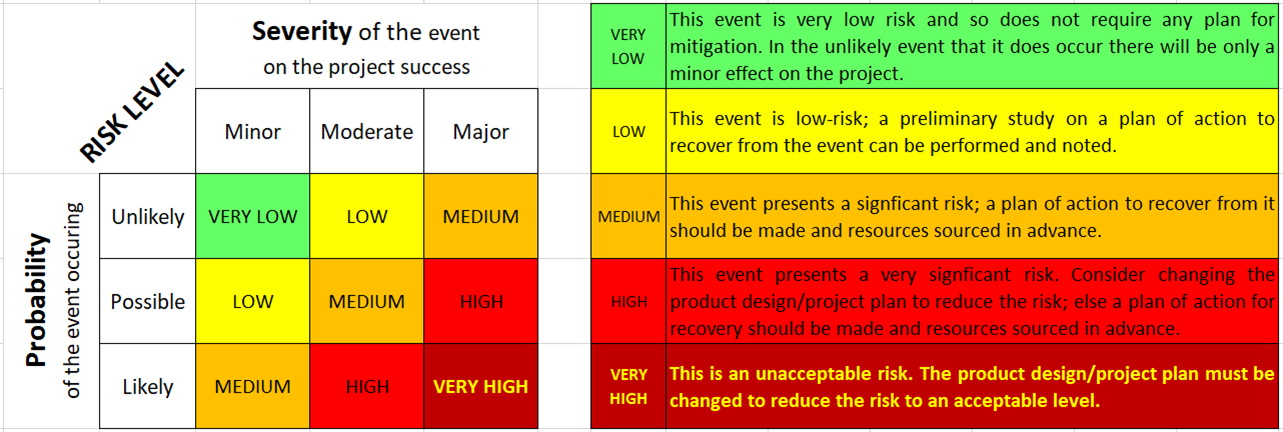
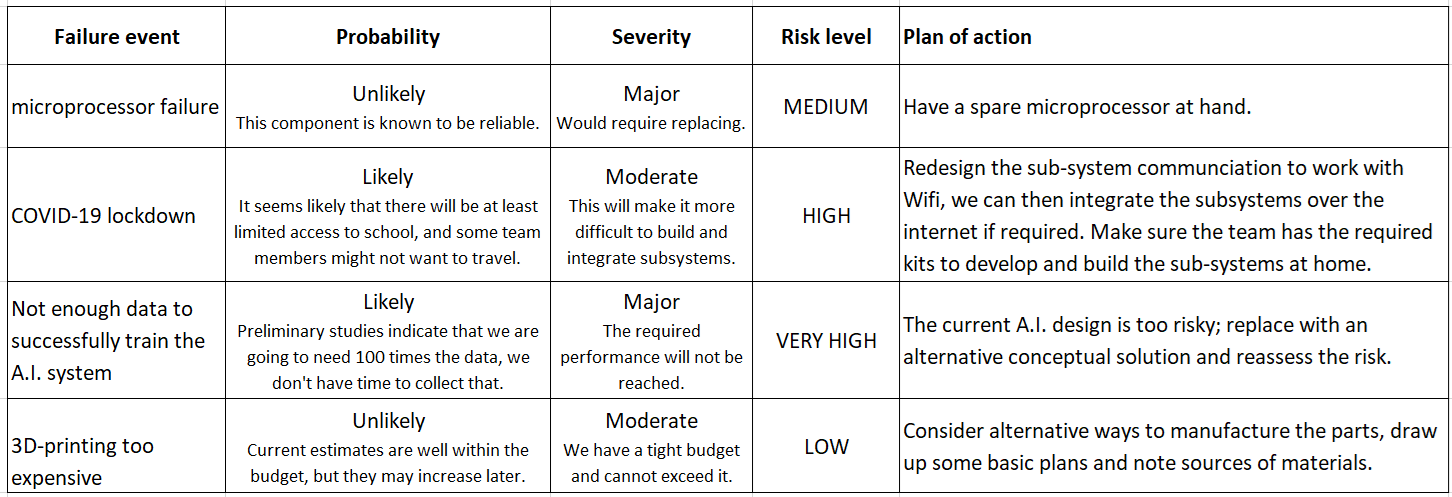
******

Table 6. Risk assessment

******

# 3. SUB-SYSTEMS

***There will be at least two sub-systems corresponding to two faculty departments and/or work groups.***

Begin the first paragraph here.

Begin the second paragraph here.

**3. Subsystems**

**3.1. Energy Systems Engineering**

**3.1.1. Definition of the problem**

**Yukardaki   
3.1.2. Review of technologies and methods**

**State of Charge  
A math equation with black text

Description automatically generated with medium confidence**

SOC(t₀): This represents the initial state of charge of the battery at time t₀​. It serves as the starting point for calculating the SOC at any later time.

η (Coulomb Efficiency): The Coulomb efficiency represents the efficiency of charge transfer during the charging and discharging processes. For lithium-ion batteries, this value is typically set to 1 due to their high efficiency.

Qcell(t): This is the available capacity of the battery at a given time ttt. It reflects the total charge the battery can store at that specific moment, accounting for any degradation.

i(τ): The current flowing into or out of the battery at any given time τ\tauτ. It is measured as a function of time and is used to integrate the charge transferred.

t₀ and t: These are the initial and current time instants, respectively. t0t₀t0​ marks the start of the calculation, and ttt is the specific time at which the SOC is being evaluated  
State of Health (SoH):  
A mathematical equation with a couple of letters

Description automatically generated with medium confidence

* Qcell(t): Available capacity of the battery at time t, estimated dynamically.
* Qcell0: Maximum available capacity of a fresh battery cell.



* θk includes Qcell(t), which represents the available capacity of the battery. –
* qθ is the process noise, accounting for gradual degradation in the battery’s parameters.

A diagram of a circuit

Description automatically generated

* **Vt = OCV + V1 + V2 + R0ik**
* OCV refers to the open-circuit voltage, which depends on the battery’s SOC.
* V1 and V2 are the voltage drops across the RC branches in the battery’s equivalent circuit model.
* R0 is the ohmic resistance, which reflects the internal resistance of the battery.
* ik is the current flowing through the battery at the specific time step k.

**3.1.3. Standards and constraints (Environmental Effects, Social Effects, Economic**

**I found 3 articles**

|  |  |
| --- | --- |
| IEC 61851-1 | This part of IEC 61851-1 applies to EV charging equipment used for supplying power to electric road vehicles. |
| IEC 62196 | Scope This part of IEC 62196 applies to EV plugs, EV socket-outlets, vehicle connectors and vehicle inlets with standardized pin and contact tube arrangements, after This to as accessories. |
| IEC 62196-2 | This section of IEC 62196 pertains to EV plugs, socket outlets, vehicle connectors, and vehicle inlets that feature standardized pin and contacttube configurations. |

**Effects, Ethical Issues, Health and Safety)**

**3.1.4. Conceptualization and Materialization**

**3.1.5. Risk assessment  
yukardaki**

**3.1.6. Evaluation**

## 3.2. Software Engineering

Begin the first paragraph here.

Begin the second paragraph here.

### 3.2.1. Requirements

|  |  |  |
| --- | --- | --- |
| Actor | Attribute/Behavior | Description |
| User | View Real-Time Data | The user can monitor real-time data such as voltage, current, SOC from the BMS. |
|  | View Previously Recorded Data | The user can filter and view past data using spesific dates. |
|  | Export Data | The user can export data and graphs in formats such as CSV, Excel, or image files. |
|  | Custom Settings | The user can set communication parameters such as baud rate, COM port and adjust visualization settings. |
|  | Start/Stop Data Aquisition | The user can control when data acquisition starts or stops. |
|  | View Connection Status | The user can check if the software is successfully connected to the BMS. |
|  | Create/Access Error Logs | The user can view and export system error logs for troubleshooting. |

|  |  |  |
| --- | --- | --- |
| Actor | Attribute/Behavior | Description |
| System | Communicate with BMS | The system establishes communication with the BMS using the appropriate protocols. |
|  | |  | | --- | | Parse BMS Data |  |  | | --- | |  | | |  | | --- | | The system processes raw data from the BMS into the necessary readable formats. | |
|  | Log Data | The system stores real-time data in the database and archives historical data. |
|  | Generate Graphs | The system produces graphs and charts for real-time and historical data. |
|  | Error Handling | The system detects, logs, and notifies users of errors. |
|  | Reconnect to BMS | The system attempts automatic reconnection if communication with the BMS is lost. |
|  | Provide Feedback | The system displays statuses (“Connected”, “Error” etc.) and action confirmations. |

|  |  |  |
| --- | --- | --- |
| Actor | Attribute/Behavior | Description |
| BMS | Send Data | The BMS transmits telemetry data (voltage, current, SOC) to the system. |
|  | Follow Standards Of Protocols | The BMS ensures data format and transmission adheres to protocol standards (UART, CAN, I2C). |

**Nonfunctional Requirements**

**Performance**

-The system shall process and display data in real time, with less than 1 second of latency.

-It should support continuous data logging for a period of at least 24 hours without any degradation in performance.

-Graphs and visualizations shall render within 500 milliseconds of a query or user action.

**Security**

-All input provided by the user, whether configuration settings or date filters, should be validated to avoid any kind of system crashes or misconfigurations.

- The logged data should be securely stored to avoid unauthorized access and loss of data.

- Where the BMS supports it, communication shall be performed using secure communication channels so that data in transit is not tampered with.

**Safety**

- The system shall ensure the integrity of the data acquisition and logging processes for accuracy and reliability.

- Upon failure or instability in the communication with the BMS, the system should immediately notify the user.

- The system shall prevent the overload of the BMS by controlling the frequency of data requests.

**Business Rules**

-The application shall export data to standard formats like CSV and Excel to fulfill user needs.

-Open-source libraries shall be preferred wherever possible to reduce development cost and enhance maintainability.

-The system shall be modular, allowing updates and integration with other tools or systems in the future.

### 3.2.2. Technologies and methods

The application software of BMS integrates several software components on the efficient acquisition, processing, and visualization of data. The software application operates on a Raspberry Pi OS, and for better functionality, a recent 64-bit version should be used. Python 3.x is used as the programming language in this application because it enables us to use libraries like pyserial for UART communication, python-can for CAN protocol support, and smbus2 for I2C interfaces to communicate with the BMS hardware. Real-time and historical data of batteries, including voltage, current, temperature, and SOC, are stored in the SQLite database, which can be managed through Python's sqlite3 or SQLAlchemy library. The incoming data from the BMS hardware is in telemetry messages via UART, CAN, or I2C. These consist of the most important parameters of the battery that a system will process into structural database entries and visual graphs. Outgoing messages include configurations to be sent to the BMS for setting certain changes or requesting data. Also, the system allows us to export data and generate CSV or Excel files for external use. The communication between components is event-driven for responsiveness; the graphical interface, implemented with PyQt or Tkinter, provides an interactive way for users to view data and manage the system efficiently. This setup supports real-time monitoring, secure data storage, and communication for reliable battery management.

**Database Managment System**

The DBMS used for this project is SQLite, which is an open-source, lightweight, and serverless database solution. SQLite is maintained by the SQLite Consortium, and the latest stable version is used in this project, such as SQLite 3.43 or higher. Its self-contained architecture eliminates the need for a separate database server, making it ideal for deployment on resource-constrained devices like the Raspberry Pi. SQLite is a file-based storage mechanism, and it ensures that all the database operations are done within a single file, making setup and maintenance easier.

For connecting to the database with the software, Python's built-in sqlite3 library is used. The application establishes a connection to the database by creating or opening an SQLite database file directly in the application's directory, such as battery\_data.db. The sqlite3.connect() method initiates the connection to this database file. The cursor object is created for executing SQL queries in the case of creating tables, data inserting, record retrieving, and updating management. Integration in this way allows the application logics to interact smoothly with the database layer. The handling mechanism for errors, including connection problems or errors due to queries, ensures that the database operations are carried out in a robust and reliable way. This accesses the SQLite database in real-time, allowing for continuous logging of battery telemetry data and retrieval of historic data for analysis and visualization within the application.

**Tables in the Data Schema**

**BatteryData Table**

**Purpose**: Stores real-time and historical battery telemetry data.

**Columns**:

-id (Primary Key, INTEGER): A unique identifier for each record.

-timestamp (DATETIME): The date and time when the data was logged.

-voltage (REAL): The battery's voltage in volts.

-current (REAL): The current in amperes.

-temperature (REAL): The battery’s temperature in degrees Celsius.

-state\_of\_charge (REAL): The battery's state of charge as a percentage.

**Configuration Table**

**Purpose**: Stores application settings and communication parameters.

**Columns**:

-id (Primary Key, INTEGER): A unique identifier for each configuration.

-parameter\_name (TEXT): The name of the configuration parameter (e.g., baud\_rate, can\_id).

-value (TEXT): The value of the configuration parameter.

-last\_updated (DATETIME): The timestamp when the configuration was last modified.

**ErrorLogs Table**

**Purpose**: Stores error messages and system events for debugging and maintenance.

**Columns**:

-id (Primary Key, INTEGER): A unique identifier for each error log.

-timestamp (DATETIME): The date and time when the error occurred.

-error\_message (TEXT): A description of the error or event.

-severity (TEXT): The severity level of the error (Info, Warning, Critical etc.).

**ExportLogs Table**

**Purpose**: Keeps track of data exports performed by the user.

**Columns**:

-id (Primary Key, INTEGER): A unique identifier for each export event.

-timestamp (DATETIME): The date and time of the export.

-file\_path (TEXT): The file path where the data was exported.

-export\_format (TEXT): The format of the exported data (CSV, Excel).

**Relationships Between Tables**

**BatteryData and Configuration:**

These tables are loosely related, with the Configuration table controlling the parameters (e.g., baud\_rate) that govern how telemetry data is collected in the BatteryData table.

**BatteryData and ErrorLogs:**

There is an indirect relationship where errors in telemetry collection or communication are logged in the ErrorLogs table, referencing the timestamp to match issues with specific telemetry records.

**ExportLogs and BatteryData:**

The ExportLogs table references the BatteryData table implicitly, as data exported is always derived from the telemetry records stored in BatteryData.

(will add example sql code here for the tables)

### 3.2.3. Conceptualization

Actor glossiary fort his project:

**1.System Administrator:** The System Administrator is responsible for the installation, configuration, and overall management of the Battery Management System (BMS) software on the Raspberry Pi. This actor handles the hardware setup, configures communication parameters, and ensures that the system is operating smoothly.

Role:

-Install and configure the software on the Raspberry Pi.

-Configure the communication settings with the BMS hardware (such as UART, CAN, or I2C).

-Monitor system performance and handle software or hardware failures.

-Update system settings and software when needed (e.g., adjusting configuration parameters or running software updates).

Interactions:

-Modify the configuration settings in the Configuration table.

-View and analyze logs stored in the ErrorLogs table.

-Ensure correct hardware operation and connection with the Raspberry Pi.

**2. End User:** This actor uses the BMS application to monitor the battery performance, view live data, and generate reports for analysis or further action.

Role:

-View real-time battery data, including voltage, current, temperature, and state of charge.

-Review historical data through graphs and trends to assess battery health and performance.

-Export telemetry data into CSV or Excel formats for further analysis or reporting.

Interactions:

-Query and retrieve data from the BatteryData table for analysis.

-Generate and export data logs, saving entries in the ExportLogs table.

-Use the graphical interface to interact with the data and receive real-time updates.

**3.BMS Applicaton:** The BMS hardware provides data to the software and receives commands for configuration and control. It communicates with the Raspberry Pi via UART, CAN, or I2C protocols. This is the main goal of this application.

Role:

-Send real-time telemetry data to the Raspberry Pi, including battery voltage, current, temperature, and state of charge (SOC).

-Receive configuration parameters from the software, such as sampling rates, thresholds, or charging settings.

-Respond to system queries or requests for specific data points.

Interactions:

-Provide telemetry data, which is parsed and stored in the BatteryData table for further use.

-Communicate data in real-time and upon request by the user.

-Process configuration commands based on settings in the Configuration table.

### Use-case Glossary

|  |  |  |
| --- | --- | --- |
| **Use-case Name** | **Description** | **Participating Actors** |
| Monitor Battery Data | Allows the End User to view real-time telemetry data from the BMS. | BMS Applicaton |
| Configure System Settings | Enables the System Administrator to manage configurations such as sampling rates and communication settings. | System Administrator |
| Log Errors | Automatically records system errors into the ErrorLogs table. | Battery Management System (BMS) |
| Export Data | Allows End User to export historical data in CSV or Excel format. | End User |
| Login | Ensures secure access by requiring login credentials for different actor roles. | Battery Management System (BMS) Applicaton |

### Use-case Scenarios

|  |  |
| --- | --- |
| **Use-case Name** | Login |
| **Use-case Description** | Ensures secure access by requiring login credentials for different actor roles. |
| **Actors** | End User,BMS Applicaton |
| **Pre-Condition** | System must be connected to a network and also connected to the BMS and plugged into a power source. |
| **Post-Condition** | System is available to use and configure. |
| **Normal Flow** | Step1 : User enters username and password.  Step2 : Authentication service validates user’s username and password.  Step3 : The system will be unlocked. |
| **Business Rules** | After 3 unsuccessfull login attemps the system will terminate. |

### 3.2.4. Physical architecture

In this capstone project, we decided on using a client-server architecture; hence, the Raspberry Pi acts as a server that interfaces directly with the BMS. Software will be a desktop application to be developed in the Capstone 2 class. Layers of the software architecture are as follows:

**Presentation Layer:** Provides the user interface for interaction with the system. Components are as follows:

Desktop GUI: Implemented in Python using the Tkinter library; includes real-time graphs, configuration screens, and export options.

Graphing Tools: Utilizes Matplotlib for real-time visualization of data received, such as voltage, current, state of charge, and more.

Input Validation: Ensures that user inputs-such as sampling rates or file paths-are valid and formatted correctly.

**Application Layer:** This encapsulates the core logic of a software program. Components are as follows:

Data Processing: The data is fed in real-time from the BMS via serial communication, such as UART, SPI, or I2C. It processes and filters the data for meaningful display and storage.

Configuration Management: Applies settings such as sampling rates, communication protocols in real time, and also validates and saves configurations in the database.

Error Handling: Errors are logged into the ErrorLogs table automatically for future troubleshooting.

Authentication: User credentials are checked to validate their access to the interface.

**Data Layer:** concerned with data storage, retrieval, and persistence. The database connects locally between the database file and the desktop application. For database operations, Python will utilize its sqlite3 library. Supports CRUD on tables. SQLite Database Schema (tables numbered below):

1.BatteryData Table: real-time and/or historical telemetry data.

2.Configuration Table: system parameters that include sampling rates.

3.ErrorLogs Table: Maintains a log of errors with timestamps and severity.

**Hardware Interface**: Creates the communication between the Raspberry Pi and the BMS hardware. The Raspberry Pi 4 Model B is equipped with GPIO pins and supports serial communication protocols like UART, SPI, and I2C. The Pi reads telemetry data from the BMS hardware (voltage, current, state of charge). Any anomalies in hardware communication are logged in the ErrorLogs table.

**Communication Flow**

1. Telemetry Data Input: The BMS will send the telemetry data through serial communication to the Raspberry Pi.

2. Parsing and Data Processing: Parse raw data received from BMS and extract meaningful metrics.

The final data logged in the database also shows the real-time view in the GUI.

3. Data Output: The changes made in the GUI are written in the database and, at the same time, applied to the hardware interface. In case of any errors, error messages will be displayed on the GUI and logged in the database for error handling. Historical data export to CSV or Excel is also possible.

**Tools and Libraries:** The project will be developed in python because of the robustness of the libraries it offers. Python language is one of the industry standards when it comes to handling data, making it the perfect choice. For the hardware side, the Raspberry Pi offers versatility and is very economically friendly; hence, the idea choice of hardware for the project, and it also has good community support. It provides the GPIOs and inbuilt serial communication support for UART, SPI, and I2C serial protocols required to communicate with the BMS module.

**Libraries:**

Tkinter: Used for GUI development.

Matplotlib: Used for real-time graphing and visualization.

sqlite3: Database handling.

pyserial: To establish serial communication with the BMS hardwares.

os and csv: Used for file handling to export data.

**Hardware:** Raspberry Pi 4 Model B connected with BMS through UART.

### 3.2.5. Materialization

Begin the first paragraph here.

Begin the second paragraph here.

### 3.2.6. Evaluation

Begin the first paragraph here.

Begin the second paragraph here.

# 4. INTEGRATION AND EVALUATION

Begin the first paragraph here.

Begin the second paragraph here.

## 4.1. Integration

Begin the first paragraph here.

Begin the second paragraph here.

## 4.2. Evaluation

Begin the first paragraph here.

Begin the second paragraph here.

# 5. SUMMARY AND CONCLUSION

In summary, ...Begin the first summary paragraph here.

Begin the second paragraph here.

In conclusion, ...Begin the first conclusion paragraph here.

Begin the second paragraph here.

# ACKNOWLEDGEMENTS

We wish to thank our adviser Title Name Surname for ..... . Also acknowledge any other help/support from friends, technicians and other staff etc...

This work was partly/wholly funded by Bahçeşehir University (remove this if are not requesting funding).

# REFERENCES

1. T. G. Conley, and D. W. Galeson, “Nativity and wealth in mid-nineteenth century cities”, *Journal of Economic History*, vol. 58, no. 2, pp. 468-493, June 1998. http://www.jstor.org/stable/2566742.
2. Wikipedia, “Bee”, [Online]. Available: https://en.wikipedia.org/wiki/Bee. [Accessed: February 21, 2016].

Use the **IEEE style** when listing references. Try to add MINIMUM 10 references.

*A good guide can be found here:* <http://libguides.murdoch.edu.au/IEEE/>,

and many examples here: <https://libguides.murdoch.edu.au/IEEE/all>

# APPENDIX A

*Information that does not fit naturally into the main body of the report can be put into an appendix. Typically this would be long sections of software code, product user manuals, large tables of validation results, etc.*

*An example of providing source code is shown in this appendix. Display the source code in a* monospace (fixed-width) *font and single-spaced.*

*Alternatively give a link to an online code repository.*

Code for packing (and unpacking) an occupancy value [0,1] into an unsigned char.

// maximum error = +-0.25% (standard dev. = 0.5/sqrt(12)

// 0% and 100% occupancies are stored exactly.

unsigned char occByte;

if (occ==0.0) { occByte = 254; } // 0% occupancy stored exactly

else if (occ==1.0) { occByte = 255; } // 100% occupancy stored exactly

else { occByte = int(occ\*200.0); }

// unpack occupancy

float occFlot;

if (occByte==254) { occFlot = 0.0; }

else if (occByte==255) { occFlot = 1.0; }

else { occFlot = (occByte+0.5)/200.0; }

# APPENDIX B

*Another appendix can go here.*