



**YILDIZ TECHNICAL UNIVERSITY
FACULTY OF MECHANICAL ENGINEERING**

ELECTRIC SCOOTER DESIGN

15067903 Fateh Isgandarov

17067902 Samir Omarov

1506A019 Ahmet Polat

PREPARED BY MECHATRONICS ENGINEERING DEPARTMENT

GRADUATION PROJECT REPORT

Project Consultant: Dr. Kadir Erkan

İSTANBUL, 2021

ELECTRIC SCOOTER DESIGN

CONTENTS

	Page
1. INTRODUCTION	9
1.1 Aim	10
1.2 Motivation	10
1.3 Scope	11
1.4 Literature Review	11
1.4.1 Electric Scooter	11
1.4.2 Battery Technology for Scooters	12
1.4.3 Usage of Electric Scooters	13
1.4.4 Injuries Associated with Electric Scooters	13
2. THEORETICAL BACKGROUND	14
2.1 Lidar Imaging	14
2.2 REST API	15
2.3 ERP	15
2.4 Finite Element Methods	16
2.5 Manufacturing Methods	16
2.5.1 Sheet Metal Bending	17
2.5.2 Welding	17
3. REQUIREMENT SPECIFICATIONS	19
3.1 Market Requirements	19
3.2 Technical Requirements	19
4. DESIGN	21
4.1 Block Diagram of the System	21
4.2 Subsystems	21
4.3 Mechanical Design	22
4.3.1 Concept Design	22
4.3.2 General Mechanical Design of the Overall System	23
4.3.3 Handlebar	27
4.3.4 Handlebar Stem	28
4.3.5 Deck (Main Body)	29
4.3.6 Fender	31
4.3.7 Wheels	32
4.4 Electronics Design	33
4.4.1 Sensor Selections	33
4.4.2 Motor Selection	35
4.4.3 Battery	41
4.4.4 DC-DC Converter vs. Second Battery	44

4.4.5 Motor Driver	45
4.4.6 Regenerative Braking	46
4.4.7 Electronic Control Unit	46
4.5 Cruise Controller Design	47
4.6 Software Design	49
4.6.1 Scooter-sharing system	49
4.6.2 Embedded Software	56
4.6.3 Safety System	60
5. DESIGN VERIFICATION	64
5.1 Analysis of Mechanical Design	64
5.1.1 Meshing	65
5.1.2 Mesh Quality	66
5.1.3 Structural Analysis	68
5.2 Cruise Control Simulink & VESC	70
5.3 Sharing System Software Tests	72
5.4 Safety Algorithm Simulation	74
6. ASSEMBLY AND PRODUCTION STAGES	76
7. FINAL SYSTEM TESTING	80
8. WORKING SCHEDULE	83
9. BUDGET	84
9.1 List of Materials	84
9.2 Product Tree	85
10. PRODUCT BACKLOG ITEMS	86
11. CONCLUSION	87
12. APPENDICES	91

REVISION HISTORY

Date	Rev. No	Definition	Authors
24/06/2021	1.0	Initial Document	Samir Omarov Fateh Isgandarov Ahmet Polat

LIST OF SYMBOLS

P	Power [W]
F_{total}	Total force required to move the Scooter [N]
$F_{rolling}$	Force due to rolling resistance [N]
$F_{gradient}$	Force due to gradient resistance [N]
$F_{aerodynamic\ drag}$	Force due to aerodynamic resistance [N]
F_{total}	Total force required to move the scooter [N]
v	Velocity [m/s]
I	Current [A]
ω	Angular velocity [1/s]
C_{rr}	Dimensionless rolling resistance coefficient
C_d	Dimensionless drag coefficient
M	total Mass [kg]
g	acceleration due to gravity[m/s ²]
A_f	Frontal area [m ²]
ρ	Air density [kg/m^3]
θ	Slope of a hill
τ	Torque [N.m]
R_f	Resistance factor for frictional losses between bearings.
r_{wheel}	Radius of drive wheel [m]
R_a	Motor Resistance [Ohm]
L_a	Motor Inductance [H]
J	Inertia [kg.m ²]
K_e	Motor Constant

LIST OF ABBREVIATION

M2M	Machine to Machine
ePWT	electric Powered Two Wheeler
BLDC	Brushless DC Electric Motor
CAD	Computer-Aided Design
REST	Representational state transfer
MQTT	Message Queuing Telemetry Transport
UI	User Interface
PID	Proportional-Integrator-Derivative
SDK	Software Development Kit
ECU	Electronic Control Unit
IoT	Internet of Things
ToF	Time of Flight
API	Application Programming Interface
GPS	Global Positioning System
BMS	Battery Management System
CRUD	Create, Read, Update and Delete
MSD	Mechatronic System Design

LIST OF FIGURES

Figure 1.1 Schematic diagram of the first documented electric scooter	12
Figure 2.1 ToF sensor work principle	14
Figure 2.2 REST API architecture	15
Figure 2.3 Finite Element Section Example	16
Figure 2.4 Bending Diagram	17
Figure 2.5 The Metal Transfer Diagram	18
Figure 4.1 Block Diagram of the system	21
Figure 4.2 Subsystem Definition Diagram	22
Figure 4.3 Concept design made in Blender	23
Figure 4.4 Assembly view of the scooter	25
Figure 4.5 Exploded view of the scooter	25
Figure 4.6 Technical drawing of assembled scooter	26
Figure 4.7 Handlebar and Handlebar Holder 3D Model	27
Figure 4.8 Provided Handlebar and Handlebar Holder	27
Figure 4.9 Display Holder 3D Model	28
Figure 4.10 Handlebar Assembly 3D	28
Figure 4.11 Handlebar Stem and Fork 3D Model	29
Figure 4.12 Provided Aluminum Handlebar and Fork	29
Figure 4.13 Deck 3D model	30
Figure 4.14 Mass Properties of Deck (Main Body)	31
Figure 4.15 Fender 3D Model and Mass Properties	31
Figure 4.16 Camera Holder 3D Model and Mass properties	32
Figure 4.17 Scooter 8.5 inch Wheel and Wheel with Hub Motor 3D Model	32
Figure 4.18 Provided Back Wheel	33
Figure 4.19 Provided Disc Brake	33
Figure 4.20 Lidar Sensor	34
Figure 4.21 SIM 808 GPS Sensor	35
Figure 4.22 Disassembled view of BLDC motors	36
Figure 4.23 Diagram of a Scooter.	37
Figure 4.24 Selected Volta H.VT1 Hub Motor	40
Figure 4.25 Battery Pack Configuration	43
Figure 4.26 Main 7.8 Ah Battery Specifications	43
Figure 4.27 Second Battery for ECU & Sensors	44
Figure 4.28 VESC Connection Schematic	45
Figure 4.29 VESC Tool	46
Figure 4.30 ECU connections	47
Figure 4.31 VESC PID Tuning & Experiments	48
Figure 4.32 Sharing System Software Diagram	49

Figure 4.33 Flutter Mobile App UI	50
Figure 4.34 QR Code Scanner	51
Figure 4.35 Admin dashboard web application UI	52
Figure 4.36 TechEmpower Benchmarks	53
Figure 4.37 FastAPI PostgreSQL Async EndPoints	54
Figure 4.38 HTTPS Certificate	55
Figure 4.39 VESC Current Settings	56
Figure 4.40 VESC Speed & Temperature Limitations	57
Figure 4.41 Motor Driving Flowchart	58
Figure 4.42 Safety system block diagram	60
Figure 4.43 Braking Distances for different friction coefficients	61
Figure 4.44 Rear Camera View with Overlay	63
Figure 5.1 Meshing done in Ansys Software	65
Figure 5.2 Element Quality analysis	66
Figure 5.3 Mesh Aspect Ratio analysis	67
Figure 5.4 Mesh Jacobian Ratio analysis	67
Figure 5.5 Mesh Skewness analysis	68
Figure 5.6 Equivalent Elastic Strain analysis	69
Figure 5.7 Equivalent Total Strain analysis	69
Figure 5.8 Total Deformation analysis	70
Figure 5.9 Simulation blocks for speed control	70
Figure 5.10 Comparison of input, output signals, and the error Simulink	71
Figure 5.11 VESC ERPM Graph	72
Figure 5.12 Python Successful UnitTests	73
Figure 5.13 Postman API Tests	73
Figure 5.14 Lidar Distance simulation plot	74
Figure 5.15 Front car's velocity simulation plot	74
Figure 5.16 Results of the Front safety system	75
Figure 6.1 Cutting the sheet in the dimensions determined and bending	76
Figure 6.2 Welding the fork and optimizing the wheel arch	76
Figure 6.3 Bending the neck of the main part	77
Figure 6.4 Welding to the main part	77
Figure 6.5 Welding the Handlebar stem seat to the 'Neck'	77
Figure 6.6 5 th Adding screw slots and other details	78
Figure 6.7 Testing by assembling the produced part	78
Figure 6.8 Painted Deck	78
Figure 6.9 Prototype's Mechanical Assembly	79
Figure 6.10 Final Product	79
Figure 7.1 Pre-Test Vehicle Position Preparation	80
Figure 7.2 Rider testing the Scooter	81
Figure 7.3 VESC Log Analysis	82

LIST OF TABLES

Table 4-1 Determined Reference Dimensions	24
Table 4-2 Rolling Resistances	37
Table 4-3 Drag Coefficients	38
Table 4-4 Different Type of Battery Specifications	41
Table 4-5 Battery consumption for 1 hour	42
Table 4-6 Software Technologies	55
Table 4-7 VESC Encoded Packets	57
Table 4-8 AT Commands for SIM 808 GPS	59
Table 5-1 Bounding Box	64
Table 5-2 Properties	64
Table 5-3 Definition	64
Table 5-4 Ansys Material	65
Table 5-5 Mesh Statistics	65
Table 5-6 Static Structural Data	69

ACKNOWLEDGEMENTS

We would like to express our gratitude to Yıldız Technical University and its administrative staff for the research opportunity they offer us and permission to use their local resources, our advisor, Assoc. Professor Kadir ERKAN and the head of the department, Professor Dr. Haydar LİVATYALI, share their valuable time and technical and theoretical knowledge.

ABSTRACT

This paper presents a new-generation electric scooter design, Mirkeet E-scooter.

Electric scooter services, which aim to reduce the inefficient use of today's transportation vehicles and are compatible with public use, have brought some industrial and individual problems. The designed product aims to solve all these problems and present a more cost-effective design than its competitors. The design process of the project went through more than one step gradually. The concept design was rendered in Blender 3D modeling program. Mechanical design was made using the Solidworks CAD program. The finished design analysis was made with the finite element method with the help of the Ansys workbench. The required electrical components were selected based on market requirements. A cross-platform mobile application was developed in Flutter SDK, while a web admin dashboard was developed in javascript for electric scooter sharing system.

Keywords: electric scooter, shared mobility system, electric scooter safety system

1. INTRODUCTION

Transportation, one of today's most common problems, manifests itself, especially in cities with a high population density, and causes a rapid increase in vehicles in private and public transportation segments. This situation at first causes overloading of the road infrastructure and increases the traffic density. The fact that alternative transportation means are not easily accessible causes people to use private vehicles unconsciously.

Besides, the cost and slow progress of other efforts to reduce traffic density make it difficult to implement these solutions.

Furthermore, in a world where a virus such as SARS-CoV-2 causes a global epidemic, crowded public transportation vehicles are the means that people avoid use. There is a significant risk in a global epidemic time. It is an indisputable fact that people need a fast, safe and cheap alternative.

On the other hand, nowadays, people's interest in micro-mobility is rising. The multi-use e-scooter is another of these micro-mobility means. The absence of specific rental and parking areas makes it more accessible and offers the user a free driving route. One factor that makes it easy to use is that only a smartphone is needed to access the service. This is why the users and providers of these services are growing.

However, the increase in this usage brought some industrial and individual problems that had to be solved with it. One of these industrial problems is cost. The electric scooter manufacturers who provide the products of the service providers only provide their customers with the hardware, so the sharing services do not come with these products, which means that the service providers carry out extra work in the creation of this sharing service, that is, the cost of the work increases.

Instead, the essential problem is the accidents that occur during these vehicles, putting human health at risk. In addition, these vehicles do not contain any extra components that provide security.

The new generation electric scooter solution we designed in this project is successful because its competitors are inadequate with the additional solutions. However, it is entirely suitable to produce a low-cost, effective solution domestically.

1.1 Aim

Reducing the traffic density, avoiding the unnecessary use of carbon dioxide and other emitting vehicles even at short distances, and providing a safer alternative vehicle compared to public transportation vehicles in the event of a pandemic has been determined as the primary purpose of this project.

With electronic and software-based security measures, we aim to minimize the risk of accidents in these vehicles and create a safer product than its competitors, and thus spread the use of environmentally friendly transportation vehicles.

In order to increase safety, which is our primary goal, it is planned to use one lidar with a visibility distance of 12 m at the front and one 8MP camera at the rear side, and the feedback given by these modules will increase the safety of the rider. In addition, the motor braking will limit the movement speed of the electric scooter in dangerous conditions.

On the other hand, by revealing a design suitable for domestic production, it should be a more accessible tool for everyone financially in multiple uses.

If we briefly summarise the purpose of the project:

- Extending the use of renewable energy
- Helping to make transportation more comfortable by reducing traffic density
- Reducing the use of public transport in the event of a pandemic
- Being a more affordable option than its competitors
- Being a safer option that minimizes possible accident risks in the widespread use of E-scooters

1.2 Motivation

Recent statistical studies show that 411 billion vehicles travel per year, but 40% of these journeys are less than 5 kilometers and 24% less than 1.6 kilometers. Let us take into account that 91% of today's cars still use internal combustion engines. It is seen that we are in a wasteful global situation in terms of the pollution it creates on the environment, inefficient energy use, and traffic density. Considering all these problems, our primary motivation is to design a fast and environmentally friendly transportation vehicle that people are familiar with,

make this design safer with equipment that should minimize human error-related accidents, and expand its ease of use.

1.3 Scope

This project should bear the mechanical properties of a vehicle that has already been in people's lives for many years and includes designing an environmentally friendly, fast, and safe mobile vehicle using electrical energy, a renewable energy source. This vehicle, defined as Electric Scooter, has been specially designed for public-use companies that offer rental services. The Electric Scooter design was made in accordance with domestic production and is planned to be more cost-effective than its competitors. Its maximum speed should be a maximum of 25 km/hour. It should be able to charge smart devices with the charging unit on this vehicle. It can be used with low-battery smart devices without affecting the rental device's battery level. Its 12m range lidar sensor and rearview camera should minimize accidents caused by human error and provide safe driving pleasure to its users. We can briefly summarize our scope as follows:

- Must be an easily accessible, fast, and safe mobile vehicle.
- Using a renewable energy source to be environmentally friendly, in this case, Electric Energy.
- Specially designed for public companies that offer rental services
- Making and planning the design in accordance with domestic production, being more cost-effective than its competitors.
- The maximum speed should be 25 km / h

1.4 Literature Review

In this chapter, a general overview of existing literature concerning electric scooters is presented.

1.4.1 Electric Scooter

While doing our literature research, we found that Haerri et al. (2008) was the first academic who researched two wheel-standing scooters [1].

The first academically documented scooter includes

- 3-phase 100W powered BLDC motor,
- Micro-controller,
- Bi-directional stream port along with power buffers made up of supercapacitors to gain energy while riding the scooter.

A schematic diagram obtained from the literature [1] is shown in Figure 1.1

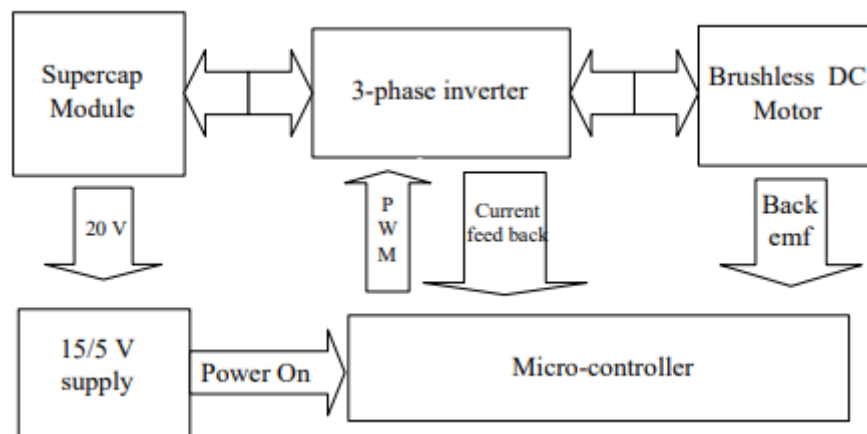


Figure 1.1 Schematic diagram of the first documented electric scooter

This particular scooter has been designed with cost as the primary factor. The BLDC 3-phase motor used was inexpensive. The plan employs a supercapacitor for energy storage. The supercapacitor can be charged either by kicking movement of human power or downhill motion. When the supercapacitor was charged to 5V, the Rider could use the motor power.

The motor calculations required for the BLDC hub motor were done for an electric scooter of a much bigger size. Nevertheless, this research provides key elements of calculating necessary motor power [2]

1.4.2 Battery Technology for Scooters

Electric Scooters use portable rechargeable batteries, mainly Lithium-ion packs. A battery's charging capabilities are reported as present per bulk of electrode material, and the C-rate describes the charging rate. Batteries also have a health (SOH) parameter and a state of charge (SOC) parameter [3]. This parameter is essential for shared electrical scooter companies.

Research has been done by Taborelli (2016) about developing new BMS, which utilizes two proposed algorithms for SOC and SOH estimation, especially for light electric vehicles such as electric scooters. Research's estimation error range is within acceptable %5, but we plan to purchase the battery pack with pre-included BMS to reduce the costs because of budget constraints.

1.4.3 Usage of Electric Scooters

In this paper, a comprehensive introduction of this Chinese marketplace, regulatory distinctions of PTWs in China, and business development is provided. One of the initial analyses of ePTWs and their use is introduced by Weinert et al. (2008) [4]. It was reported in the literature that, driving forces behind the success of ePTWs from the Chinese marketplace are:

- Performance and cost enhancements
- Government coverage support towards ePTWs
- Inferior bus public transit support.

We can assume that crowded cities all suffer from public transport problems. Moreover, with the help of authorities, Electric Scooters can become famous in Turkey as well.

In the US, a Startup Firm called Bird released tens of thousands of scooters on Santa Monica roads, making them accessible to smartphone users with easy battery and motor technology. However, all these scooter-sharing companies purchase E-scooters from manufacturers, add their IoT system, and develop mobile UI applications and a web dashboard to monitor the system. We are aiming to combine all these features directly with our proposed E-scooter. By scanning a Fast (QR) code, the scooter unlocks itself and can be readily available to be used at a relatively low price.

1.4.4 Injuries Associated with Electric Scooters

Electric scooter injuries have been explored in prior studies by Trivedi (2019). "In this study of a case series, 249 patients presented to the emergency department with injuries associated with electric scooter use during a 1-year period, with 10.8% of patients younger than 18 years

and only 4.4% of riders documented to be wearing a helmet. The most common injuries were fractures (31.7%), head injuries (40.2%), and soft-tissue injuries (27.7%)" [4]. Within this study, accidents related to electric scooter usage are common, ranging in seriousness, indicating reduced safety systems developed for electric scooters. A new approach is, therefore, needed for reducing injuries. Our proposed Lidar safety system can reduce this type of injury.

2. THEORETICAL BACKGROUND

2.1 Lidar Imaging

LiDAR is an active method for remote sensing. An active system itself produces energy to calculate objects on the ground - for this case, light. Light is emitted from a rapidly fired laser inside a LiDAR device. This light moves to the ground and reflects from objects such as buildings and vehicles. The reflected light energy then returns to where the LiDAR sensor registers it.

The theory of measurement used for imaging using Lidar is time-of-flight (TOF), where the range is determined by counting time delays in light events emitted from the source. Thus, Lidar is an active, non-contact range-finding technique in which an optical signal is projected onto an area that we call a target. A mirrored signal is detected and analyzed to determine the distance. One can measure the distance from the object based on the forward-backward delay of the light waves, as shown in figure 2.1 [5].

In the case of the LIDAR-Lite v3, one can calculate the distance by using the equation:

$$d = \frac{c \times t}{2}$$

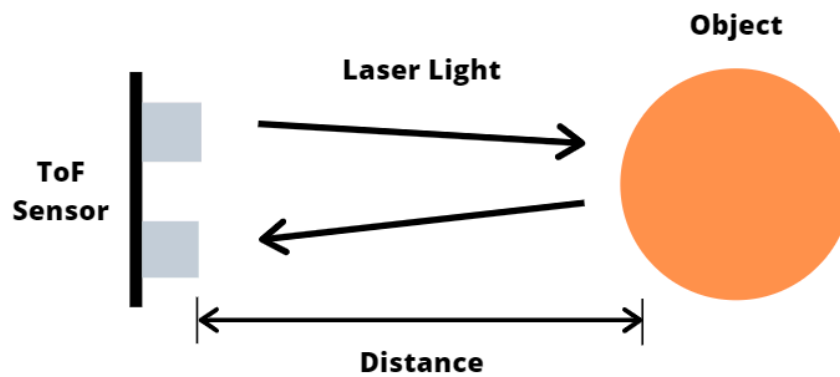


Figure 2.1 ToF sensor work principle

2.2 REST API

With the expected future of the Internet of Things, the need to develop software capable of managing communication devices arises.

A REST API that adheres to REST architectural style constraints and allows for connections with web services, as shown in figure 2.2. REST stands for representational state transfer. "REST requires a set of architectural protocols that, when Implemented as a whole, emphasize scalability of component interactions, the generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency and enforce security." [6].

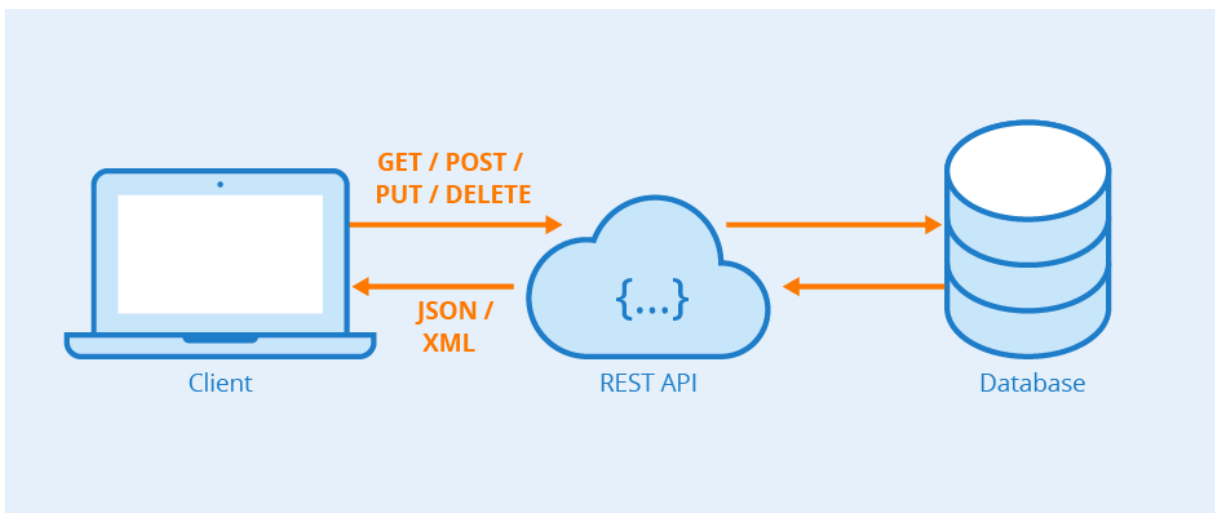


Figure 2.2 REST API architecture

2.3 ERPM

In this Graduation Project, VESC is used as a motor driver. Therefore, instead of traditional RPM – Revelotiuons Per Minutes, VESC uses Electrical Revelotiuons Per Minutes; hence ERPM.

The relation between RPM and ERPM depends on the motor construction type, especially motor poles count. The motor we are using has 30 magnets and 15 magnets/pole pairs. Therefore, RPM can be calculated as $ERPM / \text{Motor Pole Pairs} = ERPM / 15$

2.4 Finite Element Methods

While the mechanical designs of the machines are being carried out, it is expected that the strength analyzes will yield more consistent and sensitive results, especially in cases where the designed parts will be exposed to heavy loads. In these cases, different from fundamental mathematical analysis, more comprehensive and sensitive analyzes can be performed in the computer environment.

The method used in this comprehensive analysis is mostly the finite element method. The basic logic of this method consists of solving the partial differential equations in two or three space variables with numerical methods. To implement this solution, a large system is divided into smaller, simpler parts called finite elements. In the image below, one can see an example finite element section of the 3D drawing of the material that needs to be analyzed [7]. Although the simple finite element used here is a quadratic tetrahedral, it can be different geometric elements in different applications.

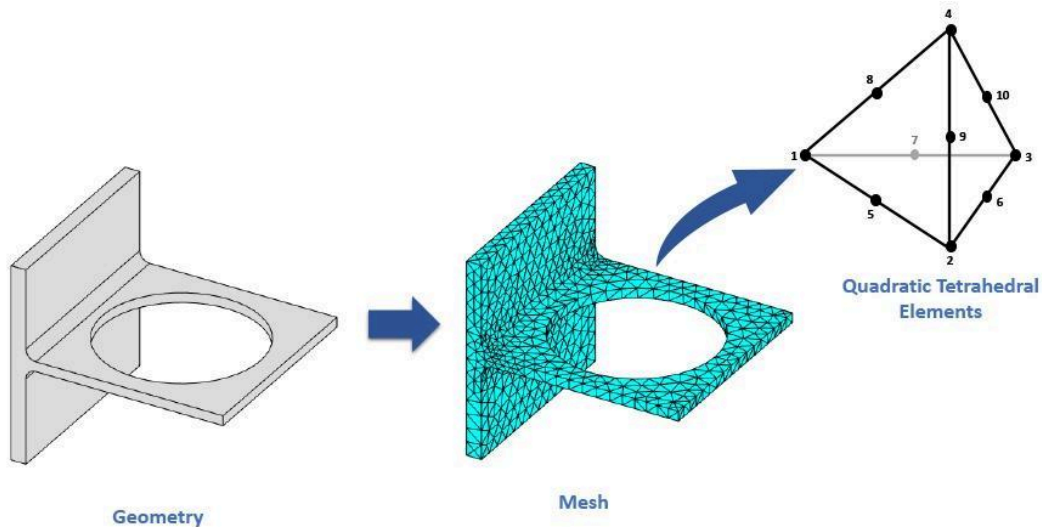


Figure 2.3 Finite Element Section Example

2.5 Manufacturing Methods

While the mechanical parts of the Mirkeet Electric scooter are being manufactured, some materials need to be added, removed, and subjected to various shaping. Several manufacturing methods are used for the realization of these processes. The most common machining

methods are turning, milling, drilling, grinding, while the chipless machining methods are casting, twisting, welding, cold hot drawing, extrusion forging, etc.

2.5.1 Sheet Metal Bending

In the production of the part of the Mirkeet electric scooter named “Deck” (Main Body), the sheet bending process was actively used. Thanks to this method, metal plates can be bent in U, V, or other different shapes at various angles. For the sheet to bend, a force is applied to it that will break the resistance value. Depending on the mold in which the sheet is placed, sheet metal can be bent into various shapes at various angles. After this plastic deformation, tensile and compressive stresses occur in the material [8].

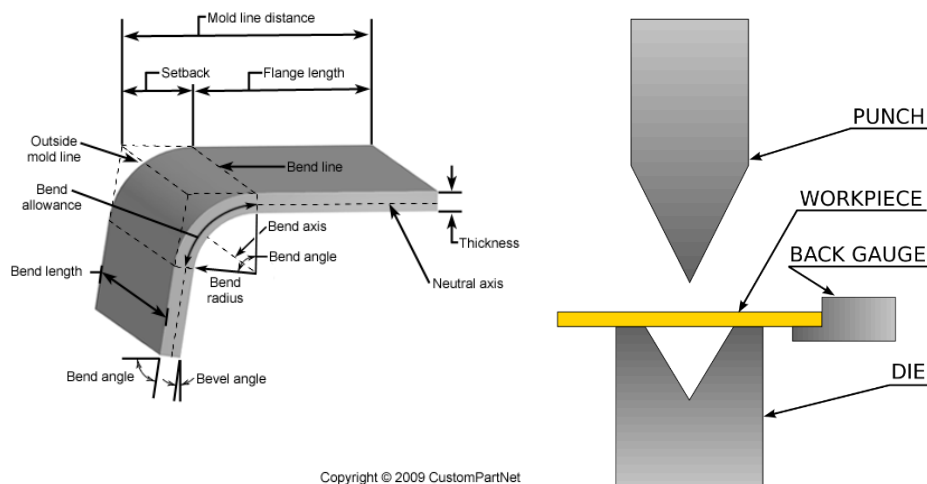


Figure 2.4 Bending Diagram

2.5.2 Welding

One of the most important methods used for metals to form a connection with each other is welding. Heat and, in some cases, pressure are used for the welding process. The materials to be used for the welding process must be weldable. However, different welding methods are used according to the welding conditions, the materials to be welded, and the conditions in which the manufactured product will be used.

The welding method used during the production of the deck is MAG gas welding. Gas welding is an arc welding method. In this welding method place to be welded is protected by

the gas or mixed gases from the welding torch. In this way, the place to be welded is not affected by the weather and other factors, and the long-lasting welding process can be continued without the necessity of interruption. The most important advantages of this welding are

- Suitability for use for thin metal sheets
- High welding travel speed and welding efficiency
- Deeper Welding Penetration

In the image below, the working principle of gas arc welding is explained visually.

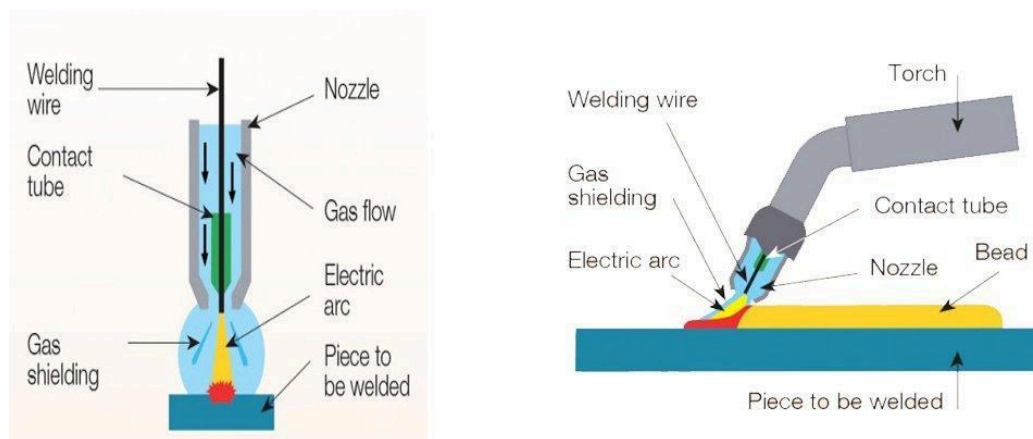


Figure 2.5 The Metal Transfer Diagram

3. REQUIREMENT SPECIFICATIONS

3.1 Market Requirements

- a) Must be suitable for public and individual use.
- b) The scooter must have a high utility load capacity.
- c) Must have a high battery capacity
- d) Must be easy to use
- e) Since there should be no specific rental and delivery area, real-time location control must be provided
- f) Must use environmentally friendly fuel, not causing pollution.
- g) Must provide a safe service that reduces human-based accidents.

3.2 Technical Requirements

Market Requirements	Technical Requirements	Explanation
a,b	The mechanical design must successfully pass total deformation strain and stress analyzes in Ansys Workbench.	The mechanical design has been made by considering the public use, keeping the number of assembled parts at a minimum to ensure high strength values.
b	The electric scooter must have the ability to travel at a speed of 25 km/hour with 90 kilograms of useful load, which means at least 300W motor power is required.	Motor requirements are determined and selected according to these values. The selected motor has 350 W nominal and maximum 550 W maximum power.
b,c	It must travel at least 25 km in full charge at a speed of 25 km/hour with a valuable load of 90 kilograms.	Battery capacity has been calculated according to the 25 km range, and the appropriate battery has been

		selected.
d	The Mirkeet Escooter rental app must be compatible with both Android and IOS operating system	You only need to have a smartphone to use. The application supports all of the most used platforms of our day.
d,e	The E-scooter must have real-time GPS communication, and the required backend system must be developed	Accessible E-scooters should appear in the Map service in the Mirkeet app.
f	The energy source has 0% emissions	The Mirket E-scooter uses eco-friendly electrical energy, also known as a green fuel.
g	12 m front and Backside Camera, Display system should be placed	To protect the user from possible accidents, sensors should be used to minimize errors caused by carelessness.

4. DESIGN

4.1 Block Diagram of the System

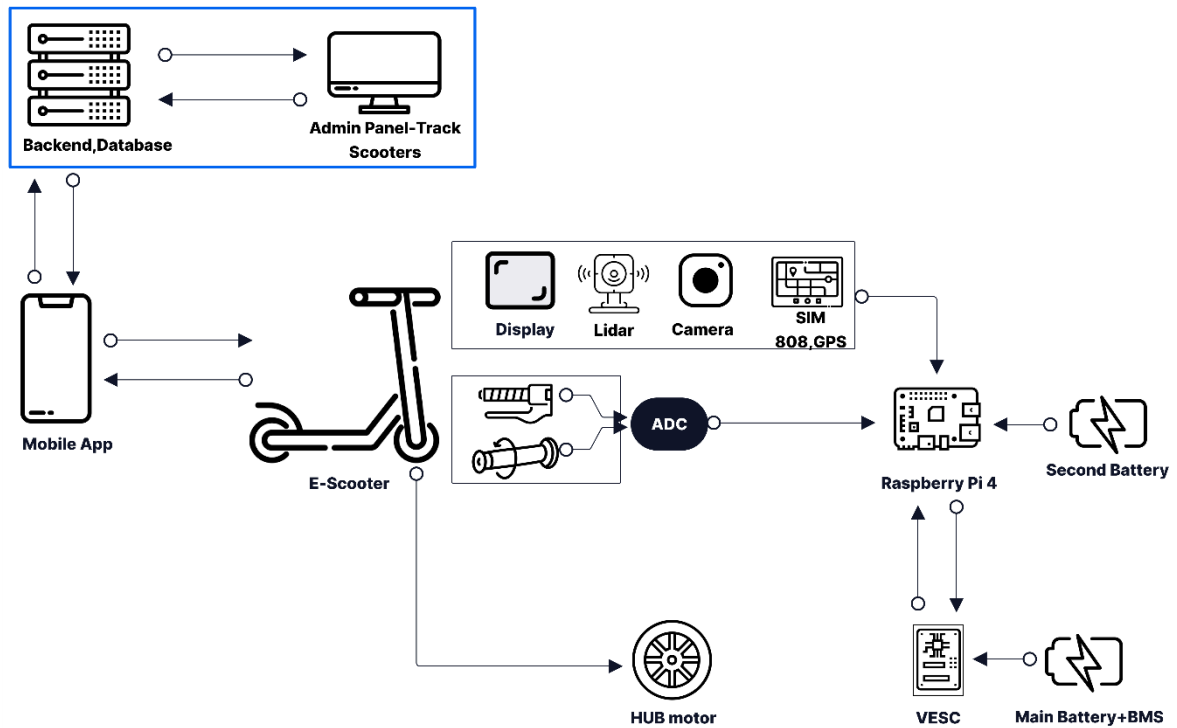


Figure 4.1 Block Diagram of the system

4.2 Subsystems

The Subsystems were identified and designed based on the customer requirements and engineering metrics and were divided into three main categories and their respective subcategories, as shown in Figure below.

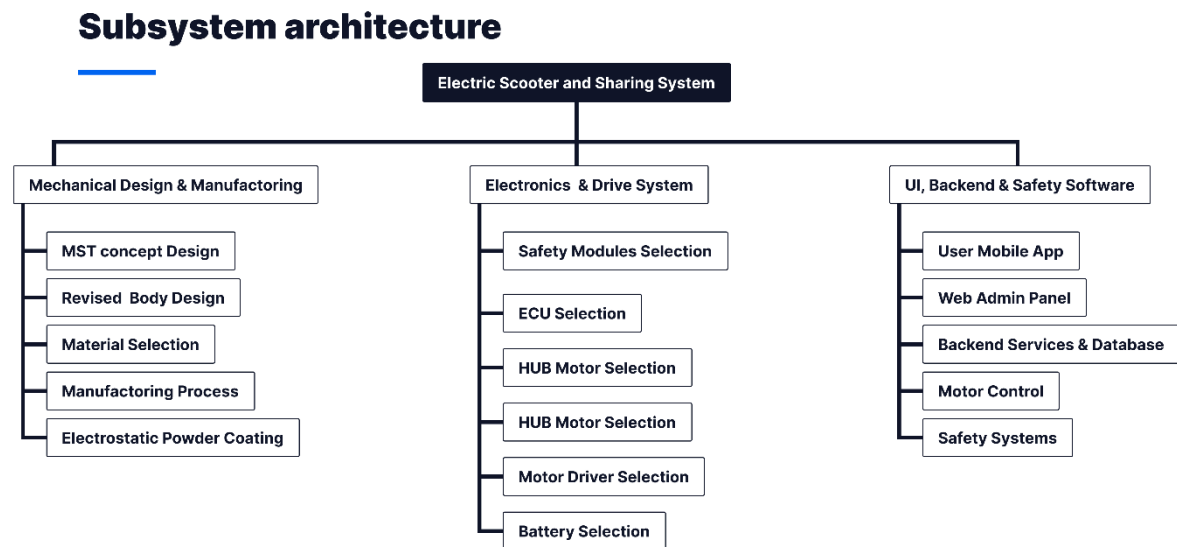


Figure 4.2 Subsystem Definition Diagram

4.3 Mechanical Design

4.3.1 Concept Design

While designing the concept, a subtle design has been presented by blending traditional design elements with the mechanical requirements for public uses.

The most important aspects of a scooter are safety, durability, and ride quality. Therefore, our goals are to design the intelligent e-scooter in general, optimize the features and specifications of this design with analysis, and provide a safe and comfortable driving experience.

While creating the concept design, some critical decisions were using two suspension systems, the motor and brakes' location, and other user-friendly additional features such as phone holder and phone charger are given and for first look Blender, 2.90.0 modeling application used. Finally, the concept design has been visualized with the help of Blender, and the first visual thought is created for the project to be produced, shown in figure 4.3.



Figure 4.3 Concept design made in Blender

4.3.2 General Mechanical Design of the Overall System

To determine the detailed technical dimensions of the parts, some reference dimensions should be taken as a basis first. These technical dimensions are among the most critical factors affecting user comfort, such as the handlebar's width, the height from the central part, etc. In addition, literature analysis shows that many companies have invested in enhancing the reliability and safety of e-scooters by increasing the vehicle's size. For example, the most successful scooter sharing company "Bird " has expanded its scooters by up to 55 percent to make e-scooters last longer [9]. Based on these researches, we determined the handlebar's width, the length, and inclination angle of the handlebar stem, the minimum and maximum width, and the height of the deck. These reference dimensions are given in Table 4-1 below.

Table 4-1 Determined Reference Dimensions

Part Specification	Determined dimensions
Handlebar width	440 mm
Handlebar Radius	22 mm
Handlebar Stem height	650 mm
Handlebar Stem Inclination Angle	15 ⁰
Deck Minimum Depth	50 mm

The detailed dimensions of the other parts are based on these reference dimensions and supported by strength analysis. The suitable dimensions of the prototype and its interior mechanical architecture have been designed with the help of Solidworks CAD. Consistent mechanical analyzes were carried out with the help of Ansys Workbench, using the finite element method. After the technical design and modeling of all these parts were completed, the project's entire CAD assembly was completed, and the technical drawing was drawn. The assembly drawing, technical drawing, and exploded view final part are given in the figures below.

The Assembly of Full CAD can be seen in figure 4.6.



Figure 4.4 Assembly view of the scooter

Exploded view generated by the Solidworks CAD program is shown in figure 4.7.



Figure 4.5 Exploded view of the scooter

suspension system intended to be used in the concept design also should not be present in the prototype.

4.3.3 Handlebar

Due to the shortage of production possibilities caused by the COVID 19 epidemic, it has been decided to supply some simple parts ready-made, not directly produced. In this context, aluminum metal pipe with 440 mm width, 22 mm radius compatible with the designed handlebar size, and the parts required for mounting this pipe to the handlebar stem were supplied from abroad (See Appendix 1 & 2 for technical drawings).

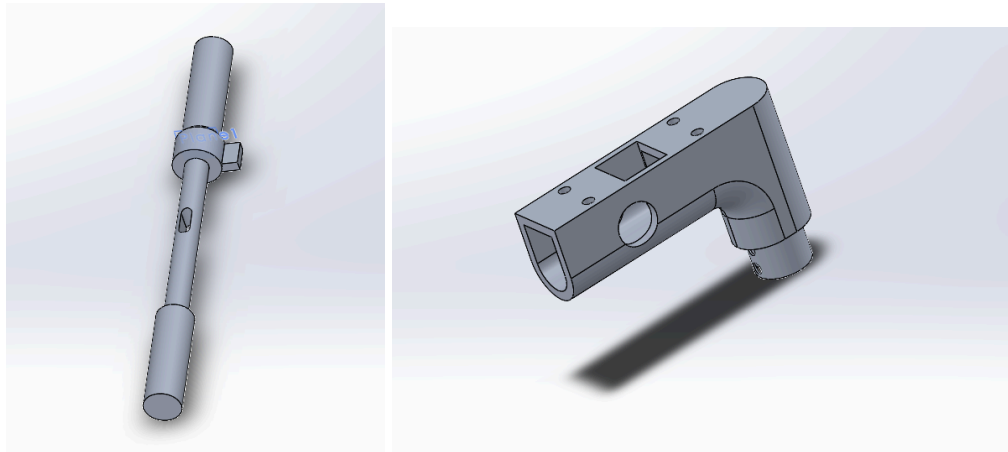


Figure 4.7 Handlebar and Handlebar Holder 3D Model



Figure 4.8 Provided Handlebar and Handlebar Holder

After these parts were procured, a suitable display holder for a 5-inch display was designed, and a 3D drawing was made using the Solidworks CAD program. Finally, the drawing was

produced with the help of a 3D printer by taking the STL file output of the drawing (See Appendix 3).

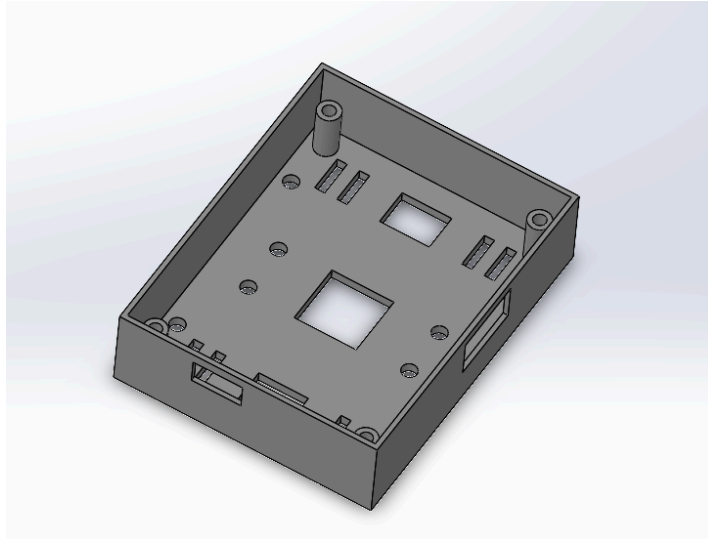


Figure 4.9 Display Holder 3D Model

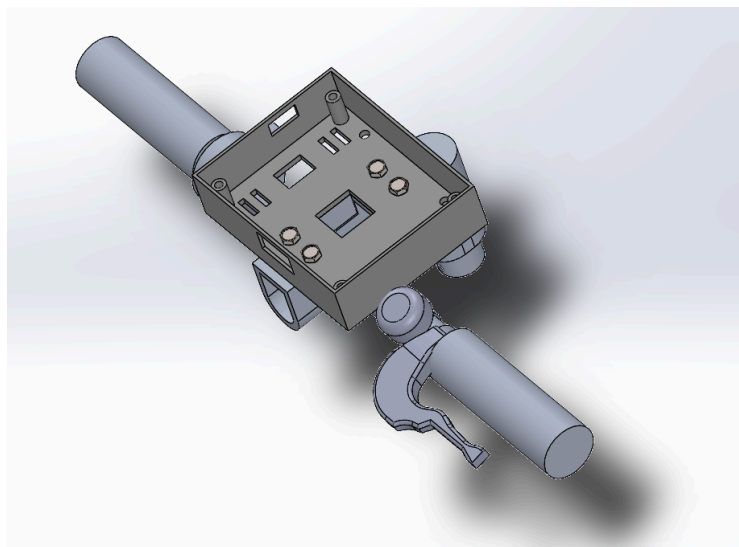


Figure 4.10 Handlebar Assembly 3D

4.3.4 Handlebar Stem

The handlebar stem and fork are the other essential parts supplied from abroad. Like the handlebar we mentioned above, the production of these parts was not made due to the risk of

contagion of the Covid-19 virus caused by the pandemic and was procured from abroad. Despite this, thanks to the realized technical drawings, it can be quickly produced under suitable conditions (See Appendix 4 & 5).

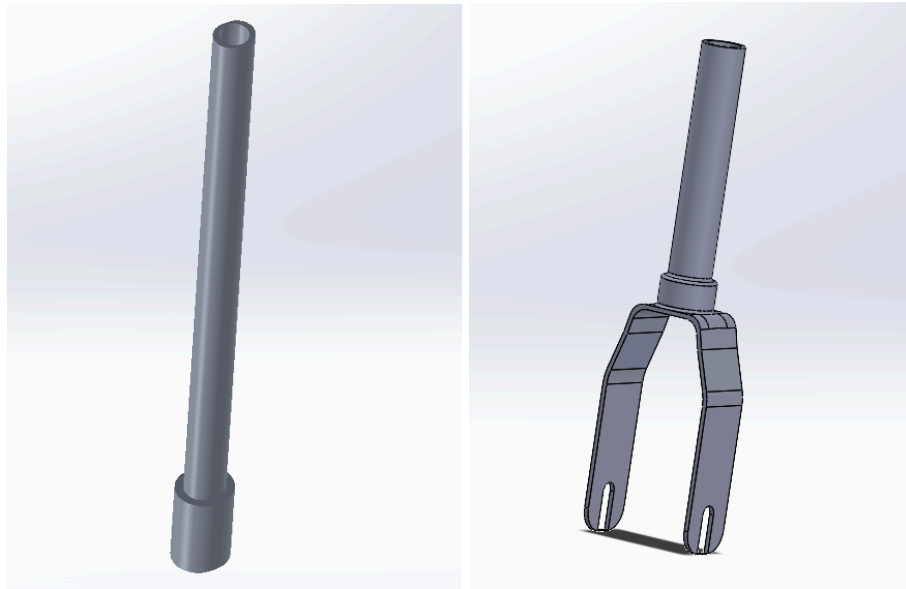


Figure 4.11 Handlebar Stem and Fork 3D Model



Figure 4.12 Provided Aluminum Handlebar and Fork

4.3.5 Deck (Main Body)

The part where the stress will be maximum is the part that we call the deck. While making this part's interior design, notch coefficient was taken into account, and appropriate spaces were left for the electronic components to be placed inside (See Appendix 6 & 7). The Figure below shows the section plane view of the body part. The average thickness of the part is approximately 3mm. This thickness was verified by analysis with the finite element method using the Ansys Workbench application. The relevant analysis results are given in detail in the relevant sections of the report.

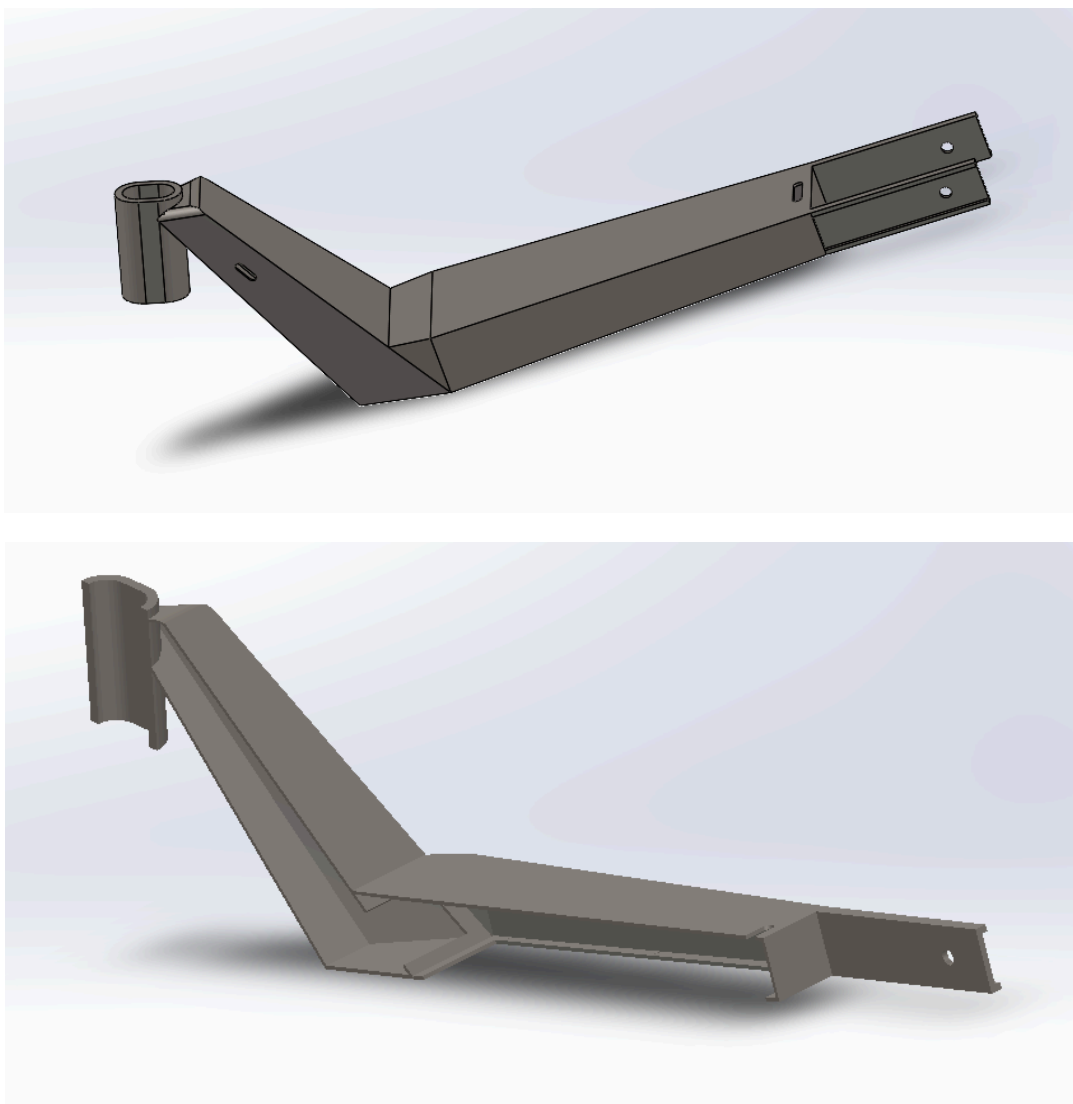


Figure 4.13 Deck 3D model

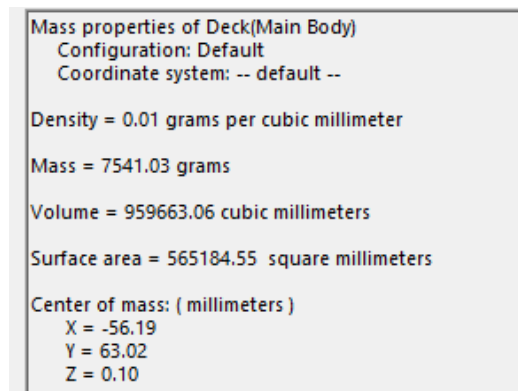


Figure 4.14 Mass Properties of Deck (Main Body)

4.3.6 Fender

The picture below shows the rear fender's CAD design and technical drawing produced by the sheet bending method from carbon steel (See Appendix 8). In the other picture, there is the weight information obtained using Solidworks Mass Properties.

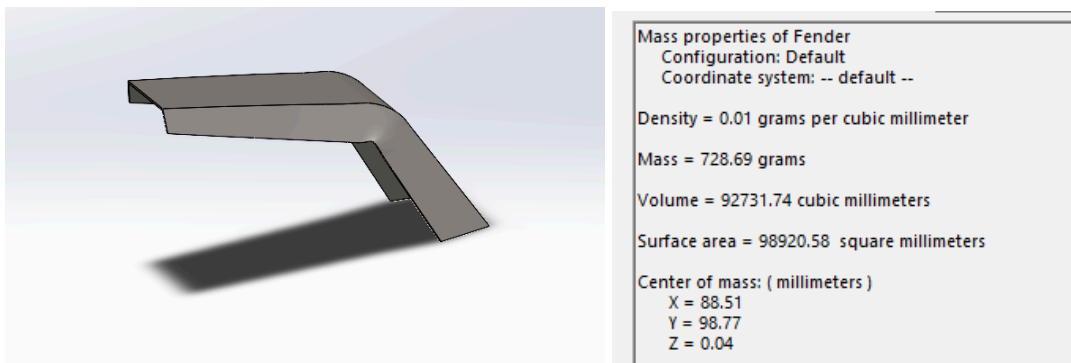


Figure 4.15 Fender 3D Model and Mass Properties

4.3.6.1 Camera Holder

In order to take an image sequence with the rear camera, which is determined as a back security measure, a part that will fix the rear camera is needed. This part is designed according to the dimensions of the camera to be used and produced by 3D printer (See Appendix 9).

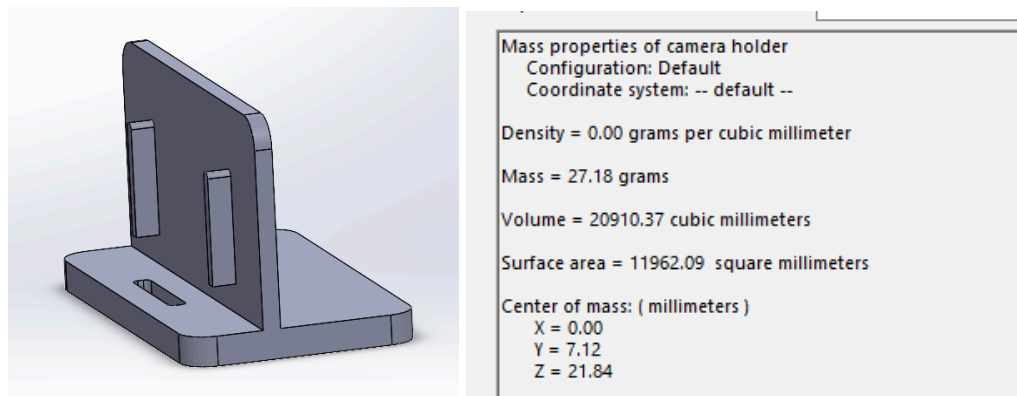


Figure 4.16 Camera Holder 3D Model and Mass properties

4.3.7 Wheels

The 8.5inch front and rear wheels are supplied from abroad. Having pneumatic wheels is preferred mainly because of its positive effect on driving comfort.

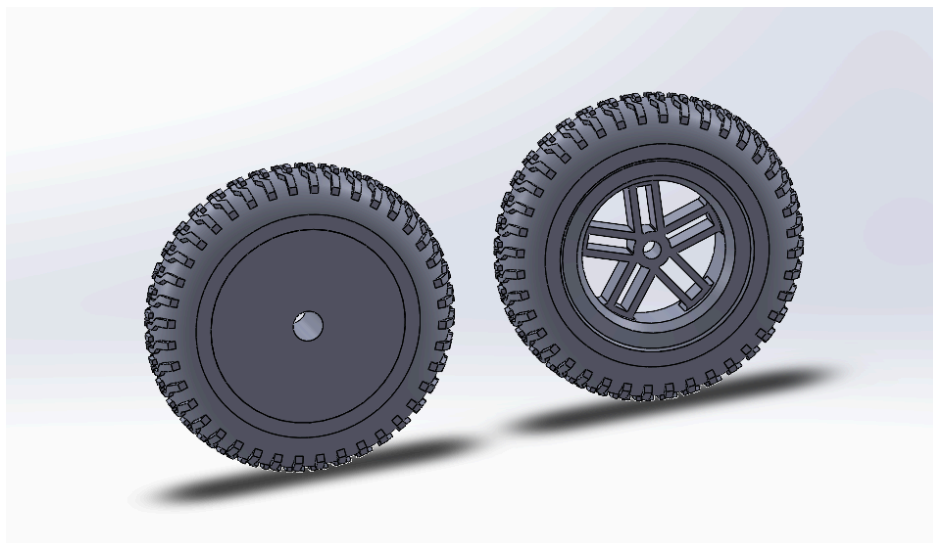


Figure 4.17 Scooter 8.5 inch Wheel and Wheel with Hub Motor 3D Model



Figure 4.18 Provided Back Wheel

4.3.7.1 Mechanical Brake

Since regenerative braking is not very powerful, mechanical braking is needed to eliminate the problems. In this case, considering the dimensions of the wheel, fork, and connection points, it has been deemed appropriate to use disc brakes in order to create a solid mechanical braking effect.

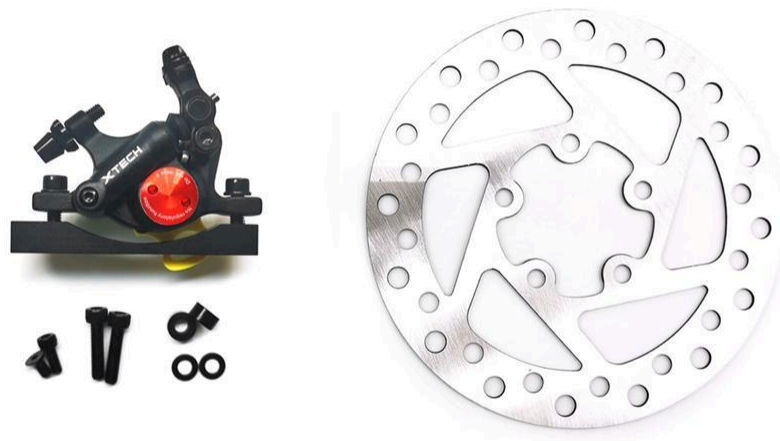


Figure 4.19 Provided Disc Brake

4.4 Electronics Design

4.4.1 Sensor Selections

4.4.1.1 Lidar Selection

The Lidar sensor was used to track the horizontal motion of the vehicles during driving. TF mini Plus Lidar sensor was chosen for this project, shown in the Figure below.

With a refresh rate of 100Hz, the Lidar sensor should be sufficient to track vehicles even at relatively high speeds.



Figure 4.20 Lidar Sensor

Technical Requirements for Lidar Sensor(TF Mini Plus) – Front Side

Operating voltage: 4.5-6V

Refresh frequency: 100Hz

Working distance: 0.3-12m

Total Weight: 110 g

4.4.1.2 GPS & GPRS Module Selection

As Sharing system is one of the primary aspects of this project, a suitable GPS module should be used. For easy integration with software side and mechanical parts, SIM 808 GPS sensor was selected. It can be controlled with AT commands which is a must for integration with Raspberry PI, and its GPS antenna is magnetic, which allows us to mount it to the body part of the scooter. Additionally, SIM 808 has less than 2.5m accuracy errors.



Figure 4.21 SIM 808 GPS Sensor

4.4.2 Motor Selection

As proposed in the “MSD” report, our choice is Hub Motor. The hub-centric motor is probably the most significant space-saver in a scooter. The motor placement into the wheel means that Space is saved for more battery while also allowing easy replacement in case of breakdown. Another advantage is the fact that the motor can be placed on the front wheel. That is not a particularly good placement due to having lower traction than on the back, but it minimizes the wiring required to operate the scooter. Moreover, we are building a Smart

Electric Scooter with many sensors and modules; it is essential to minimize wiring.

A brushless dc (BLDC) Motor is an industry standard in Electric scooters. The rotor of the BLDC motor is a permanent magnet, and the stator has a coil arrangement, as shown in figure 4.7. DC power energizes the coil, and then the coil becomes an electromagnet. The operation of a BLDC is based on force interaction between the permanent magnet and electromagnet. In these motors, torque, and current, voltage and components are linearly related.

BLDC motor is suited for all Electric Scooter needs [10].

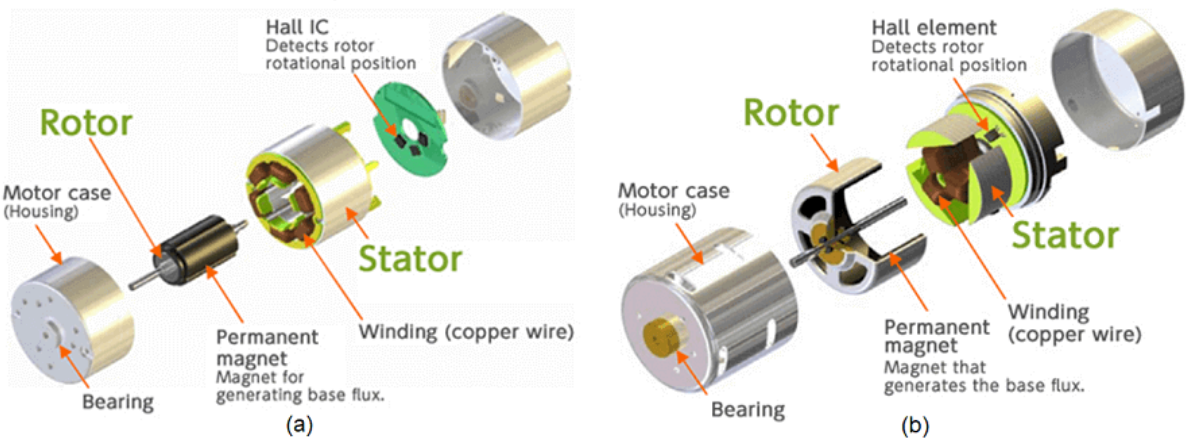


Figure 4.22 Disassembled view of BLDC motors

To determine the motor required to operate E-scooter with given requirements, the E-scooter dynamics such as rolling resistance, gradient resistance, the aerodynamic lift must be included and calculated with the below equations. We assumed a max weight of 120 kg.

The equation of the scooter's force is calculated as in Equation 4.1 [2] :

$$F_{total} = F_{rolling} + F_{gradient} + F_{aerodynamic\ drag} \quad (4.1)$$

4.4.2.1 Rolling Drag

The rolling resistance is contingent upon the co-efficiency of rolling friction, which fluctuates depending upon the material of the surface of the movement's roughness.

Rolling resistance is a sum of the resistive components, as shown in the equation below [11]:

$$F_{rolling} = F_{Tire\ rolling} + F_{Road\ rolling} + F_{Tire\ slip\ angle\ resistance} + F_{Residual\ braking\ resistance}$$

Nevertheless, for the case of simplicity, it can be calculated by following the simplified equation in 4.2:

$$F_{rolling} = C_{rr} \times M \times g \quad (4.2)$$

For application $C_{rr} = 0.004$ was selected as per the below table [2]:

Table 4-2 Rolling Resistances

0.001	Bicycle tire on the wooden track
0.002	Bicycle tire on concrete
0.004	Bicycle tire on asphalt road
0.008	Bicycle tire on roughly paved road

Applying our variables to equation 4.2, we can get Rolling Force :

$$F_{rolling} = C_{rr} \times M \times g = 0.004 \times 120 \times 9.81 = 4.7088N \quad (4.3)$$

4.4.2.2 Gradient Resistance

When the vehicle travels uphill, its weight works in a direction opposite to its motion. The gradient resistance depends on the slope of the road and the weight of the vehicle. The angle is θ which is shown in the below Figure:

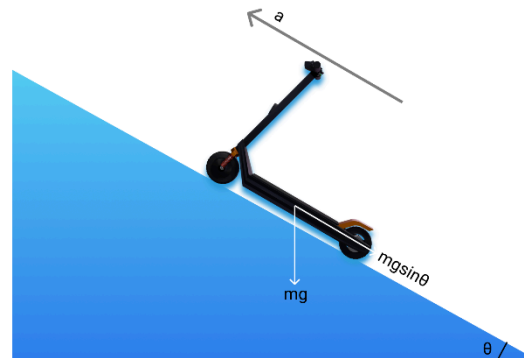


Figure 4.23 Diagram of a Scooter.

The Gradient Resistance can be calculated with equation 4.4. For the application, we consider

$$\theta = 2^\circ$$

$$F_{\text{gradient resistance}} = \pm M \times g \times \sin \theta = 120 \times 9.81 \times \sin 2^\circ = 20.5450 \text{ N} \quad (4.4)$$

4.4.2.3 Aerodynamic Drag

Aerodynamic drag is the force that needs to be overcome when moving through the air at a certain velocity and can be calculated as below equation 4.5 :

$$F_{\text{aerodynamic drag}} = 0.5 \times C_D \times A_f \times \rho \times v^2 = 0.5 \times 1 \times 0.92 \times 1.1644 \times 6. \quad (4.5)$$

For the application we consider, the maximum speed of our scooter, which is 25 km/hour, and that is 6.94444 m/s, and air density is 1.1644 kg/m³ at 30° Temperature and drag coefficient are per table 4.2 (adult human); the frontal area can be calculated using the DuBois Formula (4.6).

$$A_f = 0.5 \times \sqrt{\frac{\text{height(cm)} \times \text{weight(kg)}}{3600}} \quad (4.6)$$

Table 4-3 Drag Coefficients

C_D	Object
.001	Laminar flat plate parallel to the flow ()
.005	Turbulent flat plate parallel to the flow ()
.1	Smooth sphere ()
.47	Smooth sphere ()
.81	Triangular trapeze (45°)
.9-1.7	Trapeze with triangular basis (45°)
.295	"Bullet (not ogive, at subsonic velocity)."
.48	Rough sphere
.0-1.1	Skier
.0-1.3	Wires and cables
.0-1.3	Adult human (upright position)
.1-1.3	Ski jumper
.28	Flat plate perpendicular to flow (3D)
.3-1.5	Empire State Building
.8-2.0	Eiffel Tower
.98-2.05	Long flat plate perpendicular to flow (2D)

4.4.2.4 Total Force

To calculate total force, as shown in equation (4.1), we combine equations (4.12)(4.4)and (4.5).

$$F_{total} = F_{rolling} + F_{gradient} + F_{aerodynamic\ drag} = 4.7088 + 20.5450 + 25.8303 \text{ ; (4.7)}$$

Then, The power required for driving an electric scooter can be calculated with equation 4.8:

$$P = F \times v \times \frac{1000}{3600} = 354.7506 \text{ W} \quad (4.8)$$

4.4.2.5 Torque on the Drive Wheel

The torque that's required in the drive is the one which the drive engine needs to produce in order to obtain the drive features on-road and can be determined by equation 4.8 :

$$\tau = R_f \times F_{total} \times r_{wheel} = 1.11 \times 51.0841 \times 0.10795 = 6.121 \text{ N}\times\text{m} \quad (4.7)$$

R_f is considered 1.11 for the application.

Final Motor Decision

We chose the Volta H.VT1.018 hub motor with all these requirements. It has 350 Watt nominal and 550 Maximum motor power, which is enough for all requirements we need to meet.



Figure 4.24 Selected Volta H.VT1 Hub Motor

Motor Specifications

- Volt(V) = 36 V
- Power(P) = 350 W (Nominal)

- Current(I) = 15.27 A

Motor Speed (RPM) The angular velocity of the wheel (in rpm) can be converted into m/s with the following equations 4.9 and 4.10 :

$$v_{(m.s^{-1})} = r \times \omega_{(rad.s^{-1})} = r \times \frac{2\pi}{60} \cdot N_{(rpm)} \quad (4.8)$$

$$N_{(rpm)} = \frac{25}{3.14 \cdot r} v_{(km.h^{-1})} = 635rpm \quad (4.9)$$

v - Speed in kmph = 25, r - Radius of tier (m) = 8.5 inch wheel = 10.795 cm

The torque of motor at maximum speed

$$T = \frac{P \times 60}{2 \times \pi \times N} = 7.32N \times m$$

4.4.3 Battery

The components that store electrical energy are batteries, which help the vehicle's engine, sensors, ECU, and motor driver run. As shown in Table 4.3 below, there is already an analysis of various types of batteries.

The essential rechargeable materials are nickel-cadmium, nickel-zinc, nickel-metal hydride, and lithium-ion/lithium-polymer, described on the battery analysis table 4.3 as NiCd, NiZn, NiMH, and Li-ion/Li-Po, respectively.

Table 4-4 Different Type of Battery Specifications

Parameter	NiCd	NiZn	NiMH	Li-ion/Li-Po
Specific Energy(Wh/kg)	40-60	100	60-120	100-265
Energy Density(Wh/L)	50-120	280	140-300	250-730
Specific Power(W/kg)	150	≥900	250-1000	250-340
Charge/Discharge Efficiency(%)	70-90	80	66	80-90
Self-Discharge Rate(%)	10	13	30	8-5
Cycle Durability(cycles)	2000	400-1000	500-1000	400-1500
Nominal Cell Voltage(V)	1.2	1.65	1.2	3.6

From table 4.3, Lithium-ion battery has been the industry standard for micro-mobility solutions. However, lithium-ion batteries can be dangerous if implemented without BMS; hence, it is vital to implement BMS.

4.4.3.1 Battery Requirements

$$P = 355W \text{ from e.q (4.8)}$$

Instead of using a full battery charge to zero for healthy battery usage, we use only 85 percent of our battery. So our battery power capacity for 1 hour should be:

$$355 \times 1.15 = 408W \times hr$$

Since the motor we use operates at 36V voltage, if we calculate the current Ah for a 1-hour motor at full speed, it should be:

$$I = \frac{408}{36} = 11.3A$$

Our battery should provide a 36 V voltage 11.3 A current for one hour of use for a motor. These calculations were made assuming that the electric scooter moves at a speed of 25 km / h with an inclination of 2 degrees. Therefore, we are adding total battery consumption to our calculations from table 4.4. If we take that, our current requirement for 1 hour is 13.02 A.

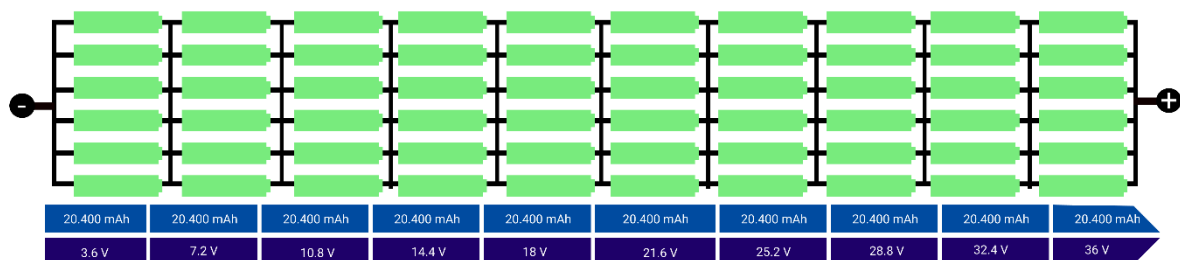
Table 4-5 Battery consumption for 1 hour

Component	Usage in <i>mAh</i>
Lidar Sensor	85
VESC	44
Raspberry PI 4	500
HUB Motor	11300
PI Camera	200
5 Inch Display	400
Sim 808 GPRS module	500
Total	13029

4.4.3.2 Battery Selection

Initially, we had two options for the battery pack; we could make our battery pack by connecting Lithium-Ion Batteries and connecting BMS. For the alternative option, we could buy a battery pack+BMS for nearly 1/3 of the cost.

For the first option, the following combination of 18650 batteries was designed:

**Figure 4.25 Battery Pack Configuration**

Nevertheless, due to Lithium battery regulations and restrictions by the government, it was not optimal to order batteries from abroad. For that reason, we chose the Xiaomi mi 365 36V

7.8 Ah battery for the project. Specifications can be seen below.

Since the prototype should not be used in bulk, a low-capacity battery is provided to reduce the prototype cost.



Nominal voltage: 36V
Capacity: 7.8AH
Nature: 30 knots 10s3P
Output current: 15-20A
Power: 280.8w
Mileage: 30 mile
Charger rated power: 71w
Charger rated input voltage: 100-240V~, 50 / 60Hz
Battery charging limit voltage: 42V
Scope of application: only support M365 M365 Pro scooter (with communication)
Charging time: about 5 hours
Intelligent BMS: temperature abnormality / short circuit / under voltage automatic sleep / over current / double over charge / double over discharge protection

Package Information:

Figure 4.26 Main 7.8 Ah Battery Specifications

4.4.4 DC-DC Converter vs. Second Battery

Initially, at the “MDS” report, we proposed to use a DC-DC converter to convert 36 V from the main battery to 5V for electrical components such as ECU and Sensors. For instance, we tried to use the LM2596VHS DC-DC converter. However, even the specifications were in range with our requirement, the Converter got hot and burned after a couple of minutes.

The solution was to use a high-end converter to support the current and voltage requirements. However, high-end converters are expensive. For this reason, we decided to use a second

battery with 5V output. This decision decreased the expenses and increased the range of the scooter as well.

Raspberry Pi 4 requires a good quality USB-C power supply capable of delivering 5V at a minimum of 2.1A. The second Battery is Lithium-polymer type and has a capacity of 10000mah.



Product Specification

Battery type: Lithium polymer battery
Rated capacity: 3.85V/10000mAh (38.5Wh)
Input type: Micro USB
Output type: USB-A
Input parameters: 5.0V / 2.0A * 9V / 12V – 18W
Output parameters: 5.1V / 2.4A * 9V / 12V 15W MAX
Charging time: 4.2 hours (18W charger, standard USB cable)
Charging time: 6.2 hours (10W charger, standard USB cable)
Product dimensions: 130 x 71 x 14.1mm

Figure 4.27 Second Battery for ECU & Sensors

4.4.5 Motor Driver

As for our Motor Driver Flipsky, FSESC4.20 was chosen, which is based on Open Source Project VESC. The motor driver is the electrical equipment that adjusts the speed of asynchronous motors by changing the frequency. It is equipped with superior protection, control, and communication features besides speed control. The main reasons to choose

VESC are to have the ability to protect critical parts and be programmable to meet every need with its configuration tool shown in Figure 4.17 below.

VESC can be connected to the Motor and Battery like the Figure below.

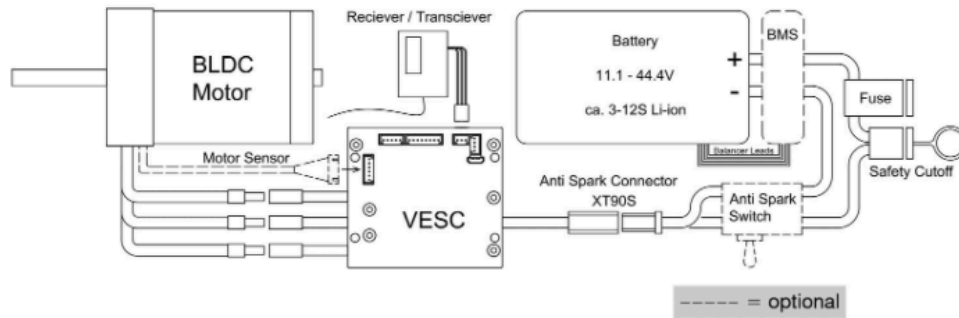


Figure 4.28 VESC Connection Schematic

One of the important reasons for us to choosing VESC is to have Regenerative Braking without additional circuits. Fortunately, our Motor Driver VESC has built-in Regenerative Braking that can charge the battery when braking.

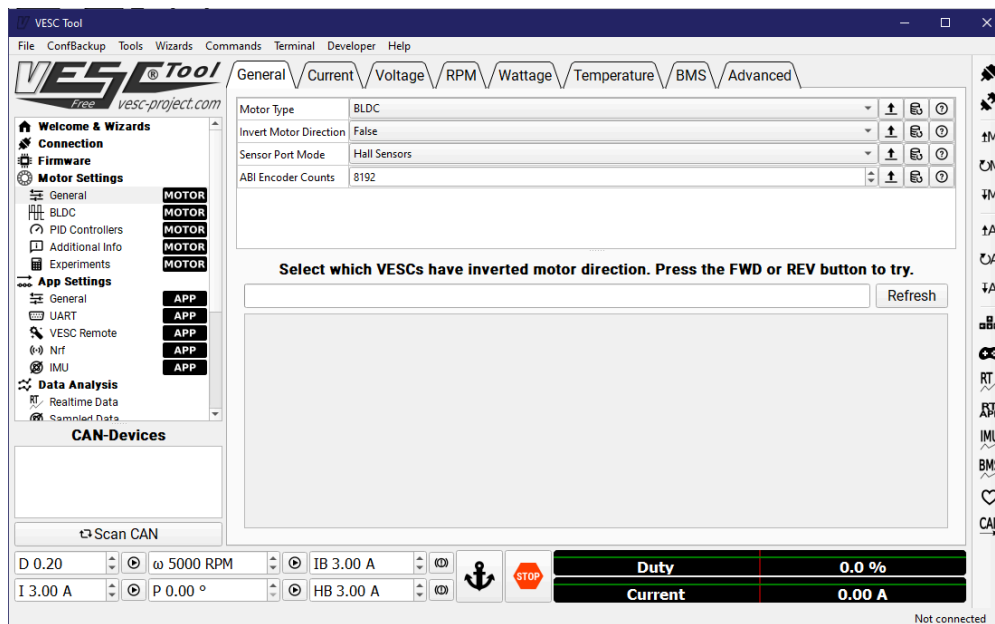


Figure 4.29 VESC Tool

4.4.6 Regenerative Braking

Activating the regenerative brakes on the electric scooter couples the DC motor into a charging circuit, creating a drag on the motor. However, the purported value provides both brake and energy savings, which translates into an extended range.

The VESC prevents outputting electricity in the battery to the engine. Instead, the engine today turns out of momentum as it rolls across the floor. It begins generating its current, called Back EMF (Back electromotive force), sent back in the battery the same as a battery charger. It essentially becomes a generator.

4.4.7 Electronic Control Unit

Electronic Control Unit here controls and modulates the whole architecture. It must read and interpolate values from the Throttle and Brake lever, tune and send necessary values to the motor driver to drive the Electric Scooter. Additionally, ECU must be connected to a server-side program all the time to update geolocation information. All the algorithms need real-time readings that is why they are very computationally expensive. For this reason, Raspberry Pi 4B was chosen as ECU.

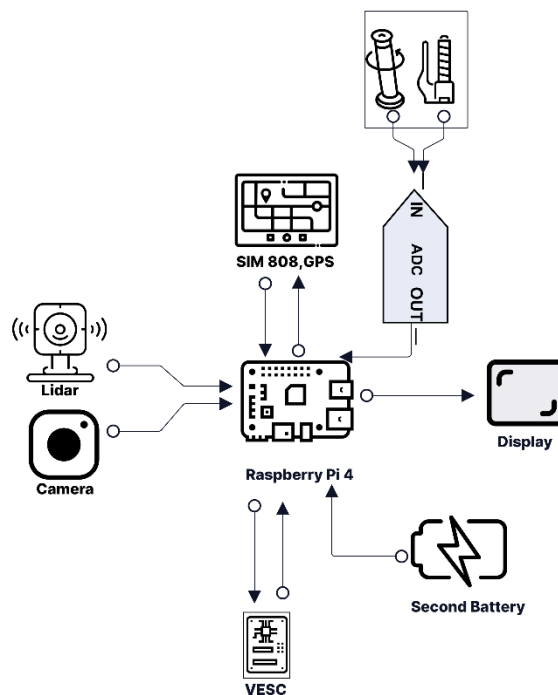


Figure 4.30 ECU connections

4.5 Cruise Controller Design

An electric scooter can be a bit tiresome on long rides, especially to control throttle all the time. To address this problem, a speed cruise control system is developed.

The cruise control system is a loop control system that tries to minimize the error between the entered reference value and the output value. The angular change in the throttle causes the change in speed, and this change in speed is reflected in the engine. The Hall Effect Sensor is used to monitor the actual output speed.

The input signal is the error of the control system. This error is the difference between the output speed and the reference speed. The controller will give the throttle control signal according to this error. The angle of the throttle will be changed by the controller, thereby changing the engine speed. According to the reference speed, we regulate the vehicle's speed thanks to the cruise control, and the speed sensor measures its actual speed. The vehicle speed regulation is made by adjusting the throttle angle while this will change the motor drive force [12].

The equation of the system is (4.1):

$$F_{total} = F_{rolling} + F_{gradient} + F_{aerodynamic\ drag}$$

The model of the forward transfer function is given below, which is :

$$\frac{\Delta V(s)}{\Delta U(s)} = \frac{96845406386}{2.11s^2 + 12.9s + 4067518} \quad (4.11)$$

However, it must be stated that the above transfer function is not accurate because the motor manufacturer does not provide the datasheet. PID is a controller that works by using the error signal repeatedly and establishing a proportional, integral, and derivative relationship. This relationship provides a perfect control method for the motor control we want. The mathematical model of the PID control system can be expressed as given below:

$$u(t) = K_p + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

VESC has a built-in PID control system that accepts PID gain parameters and lets to tune the system with ease. As mentioned above, we used a heuristic method widely known as the

Ziegler-Nichols method to tune our PID controller because we did have access to motor parameters.

VESC toolbox also lets us do experiments and log the data to analyze further and tune PID parameters.

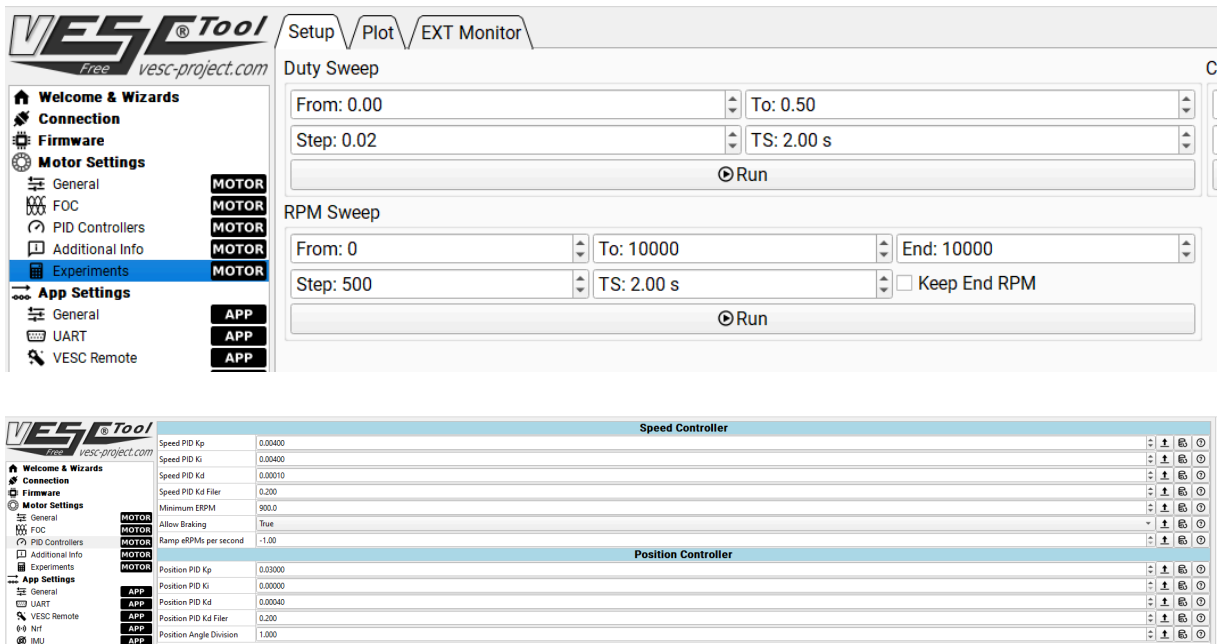


Figure 4.31 VESC PID Tuning & Experiments

4.6 Software Design

The Software Design consist of three integrated parts:

- Scooter-sharing system Frontend & Backend
- Embedded Software
- Safety system algorithms implementation

4.6.1 Scooter-sharing system

As proposed solutions for increasing micro-mobility solutions in the “MSD” report, we built scalable architecture, a dockless sharing system. It provides users to locate and drop off at any available point. Our proposed architecture consists of three parts:

- User Mobile Application
- Web Admin Dashboard
- Backend services

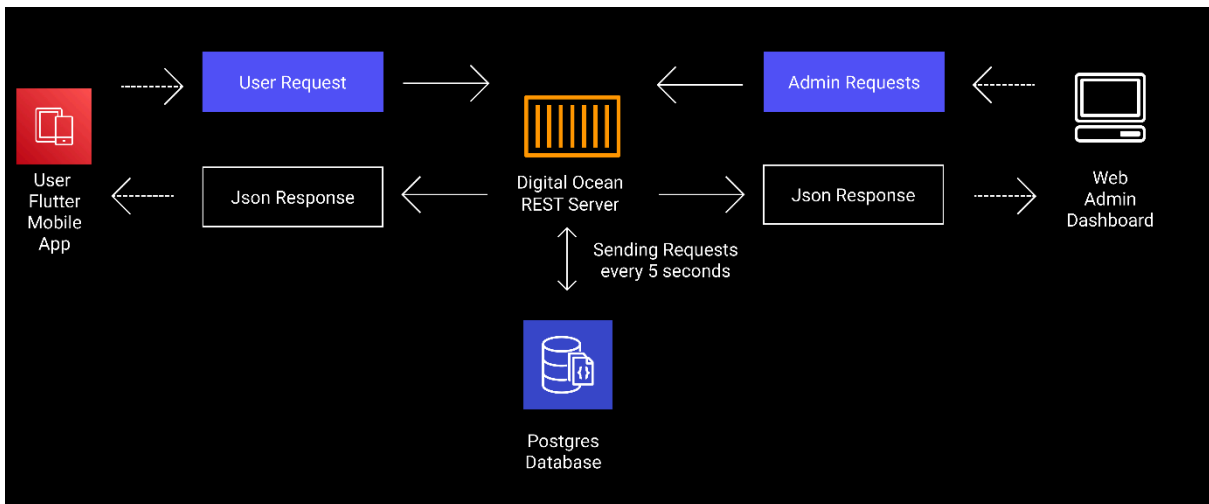


Figure 4.32 Sharing System Software Diagram

The Figure above illustrates sharing system's software modules work principle. The user requests to locate all micro-mobility vehicles and our backend service response with JSON data which is again processed on the Mobile application and shown in Google maps as markers.

4.6.1.1 User Interface

Google Maps integration must be integrated to allow users to find available vehicles that are close to them.

The user interface was designed with new trends of frontend design in mind, which is Neumorphism. Mobile App was developed in Dart programming language with Flutter SDK.

In the Figure below, User Interface is shown, google maps integration is done with relative Flutter library and "DIO" framework.

Hybrid Bike's location (MIRKEET Bike Branch's graduation project) is also integrated into our mobile application.

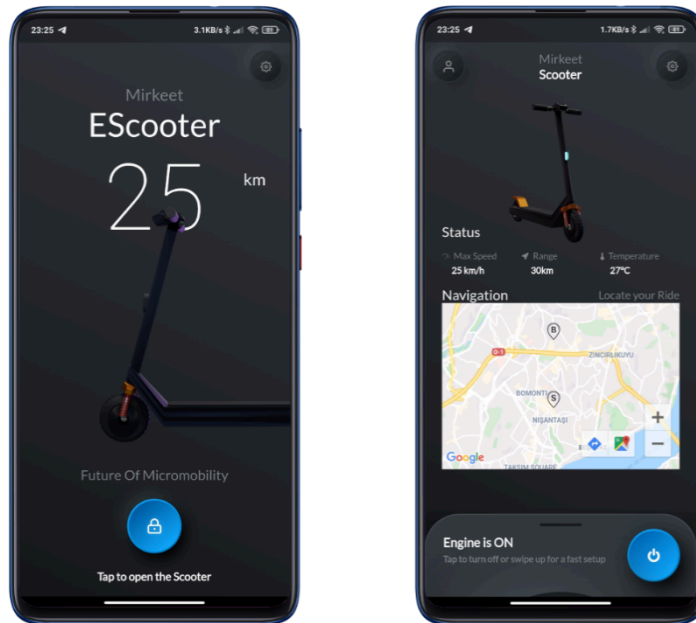


Figure 4.33 Flutter Mobile App UI

Additionally, a QR code scanner module was also developed to start & stop the E-scooters.

When the user scans the QR code of a micro-mobility vehicle with the corresponding ID, the server sends requests to stop regenerative braking and start the driving module.

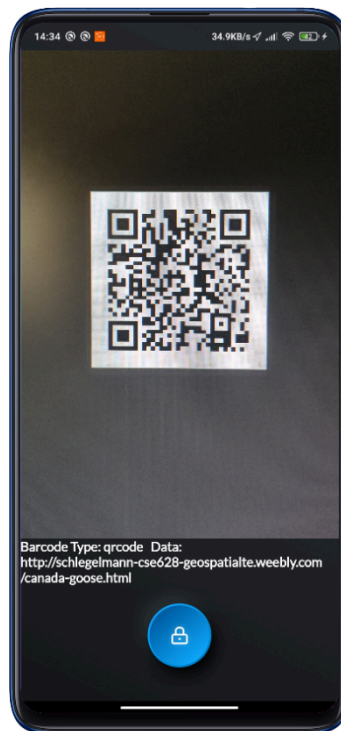


Figure 4.34 QR Code Scanner

```
1. Future<void> _onAddMarkerButtonPressed() async {
2.   _markers.clear();
3.   try {
4.     var response = await Dio().get("https://api.mirkeet.com/vehicles/");
5.     print(response);
6.     for (var markerData in response.data) {
7.       var icon = await getIconById(markerData["id"].toString());
8.
9.       _markers.add(Marker(
10.        // Marker id uniquely identifies each marker.
11.        markerId: MarkerId(markerData["id"].toString()),
12.        position: LatLng(markerData["lat"], markerData["lng"]),
13.        icon: icon,
14.      ));
15.    }
```

The main navigation module is shown in the code snippet above.

On line 4, an asynchronous get request is made to get the location and id of available vehicles; after that, Markers are updated on the Maps module with corresponding Icons.

4.6.1.2 Web dashboard

To interface for monitoring rider's activity and track Scooters in Real-time web admin

dashboard has been developed with the ReactJS programming language. The integration with Backend services is done with the “Axios” library.

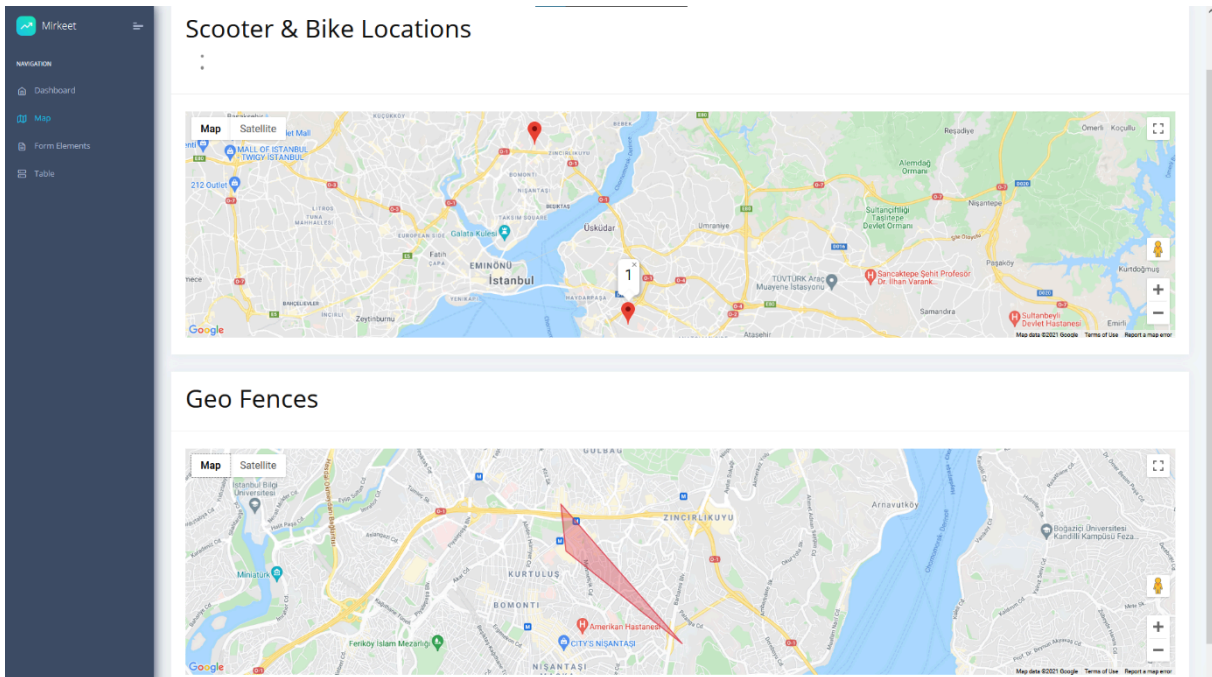


Figure 4.35 Admin dashboard web application UI

Key points:

- Fleet tracking: This allows to locate scooters in the database and show them on the map, updating their information in real-time.
- Geo Fences: Lets admins create digital fences to monitor the scooter's location in case of users riding outside of pre-defined safe zones.

4.6.1.3 Backend Services

The backbone of the sharing system is server-side. The server-side software is responsible for storing data & completing and returning a response to User and Admin requests.

Python programming language was used to develop all the backend services. Python also has a broad range of frameworks to choose from. Considering that the program needs to update geolocation data of multiple vehicles in real-time, the web framework must be

high-performance. Independent TechEmpower benchmarks show FastAPI is the top 1 ranked framework among all Python web frameworks, as shown in the Figure below [13].

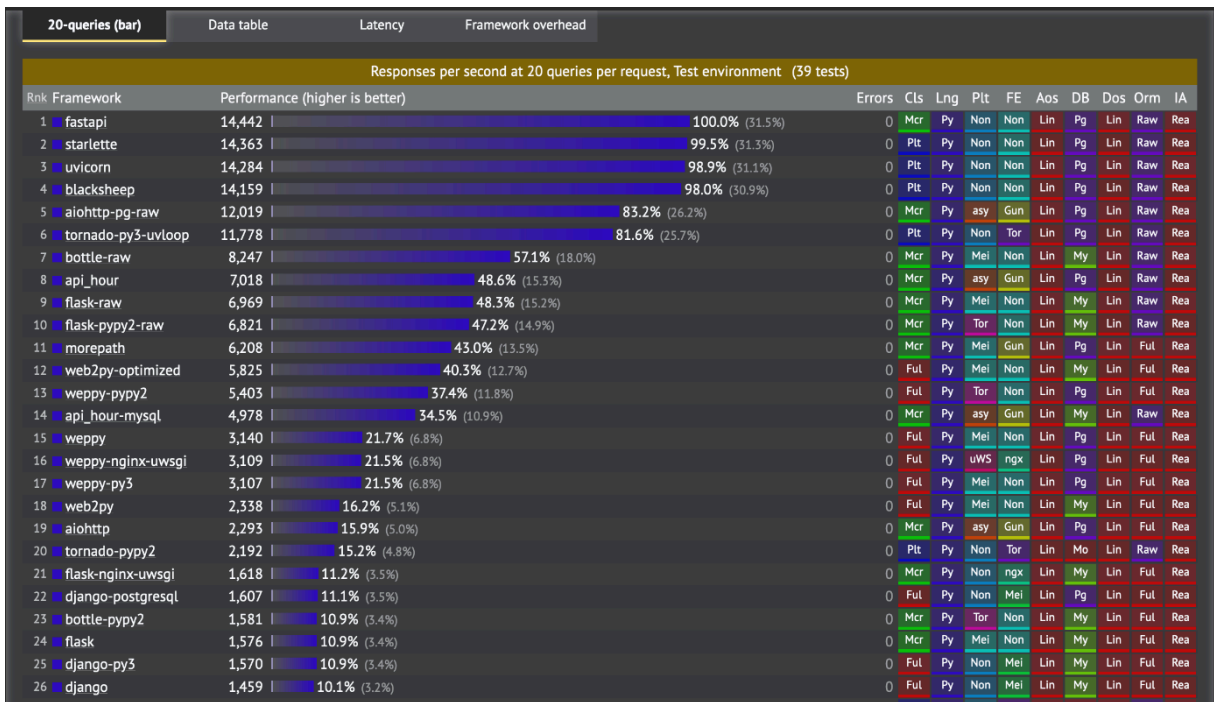


Figure 4.36 TechEmpower Benchmarks

REST API was developed with OOP principles in mind. All of the vehicles are defined and created from the “vehicles” class.

```

1. vehicles = sqlalchemy.Table(
2.     "vehicles",
3.     metadata,
4.     sqlalchemy.Column("id", sqlalchemy.Integer, primary_key=True),
5.     sqlalchemy.Column("lat", sqlalchemy.Float),
6.     sqlalchemy.Column("lng", sqlalchemy.Float),
7.     sqlalchemy.Column("battery", sqlalchemy.Integer),
8.     sqlalchemy.Column("distance", sqlalchemy.Float),
9.     sqlalchemy.Column("rented", sqlalchemy.Boolean, unique=False, default=False),
10. )

```

The code snippet above dynamically creates a table in our PostgreSQL database. The database has the following columns:

- ID – non-unique identifier to determine E-scooter or E-Bike
- coordinate representing vehicle
- Battery – Battery level of a vehicle
- Rented – Rent Status of the corresponding vehicle

The REST API has six methods implemented with endpoints shown in Figure below. Documentation of whole architecture with online run system support was also developed and can be accessed at <https://api.mirkeet.com/docs#/>

default

GET	/vehicles/	Read Vehicles
POST	/vehicles/	Create Vehicle
PUT	/vehicles/{vehicle_id}/start/	Update Vehicle
GET	/vehicles/{vehicle_id}/	Read Vehicles
PUT	/vehicles/{vehicle_id}/	Update Vehicle
DELETE	/vehicles/{vehicle_id}/	Update Vehicle

Figure 4.37 FastAPI PostgreSQL Async EndPoints

- *GET /vehicles/* - command Reads information about all vehicles from the PostgreSQL database and returns it in JSON format.
- *POST /vehicles/* - command Creates new vehicle and stores its initial information in the database
- *PUT /vehicles/id/start/* - command Updates “rented” status of vehicle (User mobile App)
- *GET /vehicles/id* – command Reads information about the vehicle with the corresponding id
- *PUT /vehicles/id* – command Updates information like geolocation from ECU
- *DELETE /vehicles/id* – command Deletes vehicle data from the database (Admin Usage)

Furthermore, to track the distance every user ride, in the *PUT /vehicles/id* command, we calculate distance every time with previous latitude and longitude with current data. The below snippet is the code to update geolocation data and calculate the distance with respective information. As shown in line 6, distance is only calculated when the vehicle is rented to decrease the payload of the Server's CPU.

```

1. @app.put("/vehicles/{vehicle_id}/")
2. async def update_vehicle(vehicle_id: int, payload: VehicleIn):
3.
4.     query = vehicles.select().where(vehicles.c.id == vehicle_id)
5.     old_data = await database.fetch_one(query)
6.     if old_data["lat"] is not None and old_data["rented"]:
7.         coords_1 = (old_data["lat"], old_data["lng"])
8.         coords_2 = (payload.lat, payload.lng)
9.         calculated_distance = distance.distance(coords_1, coords_2).km
10.        query = vehicles.update().where(vehicles.c.id ==
        vehicle_id).values(distance=(old_data["distance"]+ calculated_distance))
11.        await database.execute(query)

```

Additionally, to enhance communication security, the secure version of HTTP - Hypertext transfer protocol secure (HTTPS) protocol was used.

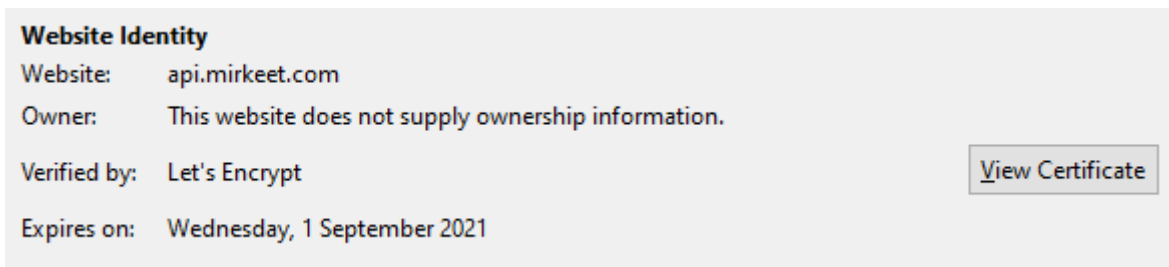


Figure 4.38 HTTPS Certificate

The complex and scalable architecture was designed, developed, and integrated into the User mobile application and Admin Control Panel to support a Micro mobility sharing system.

The following table summarizes the whole architecture:

Table 4-6 Software Technologies

Programming Language for backend	Python
Programming Language for frontend	Vuejs, ReactJS

User Mobile Application	Dart, Flutter SDK
Framework	FastAPI
Database	PostgreSQL
Cloud Servers	DigitalOcean Cloud Services
Deployment	Docker, Gunicorn / Nginx
Connection with IoT	REST API Update method

4.6.2 Embedded Software

Embedded software works were done in VESC and Raspberry PI 4B.

In the previous “MDS” Report, we planned to use VESC in “ADC & UART.” In that mode, the Throttle and Brake lever connect directly to ADC ports of the VESC, and VESC has a built-in mapping system to control the motor with current control similar to torque control. Further, we anticipated using UART mode to read motor values and write Safety Systems modules like Auto Braking System. However, conversely, VESC does not allow any other mode to use “write” privileges other than ADC.

Consequently, it was obligatory to use VESC in “UART” only mode and develop & implement mapping and throttle algorithms by ourselves.

Motor Driver VESC toolbox was used to set protection limitations to prevent overcurrent, wattage, and voltage spikes. The Motor control type was also set FOC to accommodate riders with comfortable rides and prevent high pitch sounds of motors.

Motor	
Motor Current Max	35.00 A
Motor Current Max Brake	-25.00 A
Absolute Maximum Current	35.00 A
Slow ABS Current Limit	True
Max Current Scale	100 % 1.00
Min Current Scale	100 % 1.00
Battery	
Battery Current Max	20.00 A
Battery Current Max Regen	-4.00 A
DRV8301	
DRV8301 OC Mode	Current Limit
DRV8301 OC Adjustment	16

Figure 4.39 VESC Current Settings

In the Figure above, one can see the Motor's and Battery's current protection settings. Motor's maximum current settings were obtained through trial and error, and the battery's maximum current and regenerative current were obtained from the datasheet.

According to the Electric scooter regulation published by Legal Gazette on April 14, 2021, the maximum speed of E-scooters must not exceed 25 km/h; with that in mind, maximum speed was limited in the VESC toolbox. Furthermore, in cases like going downhill, regenerative braking will automatically apply to prevent overspeeding at a speed of 25 km/h.

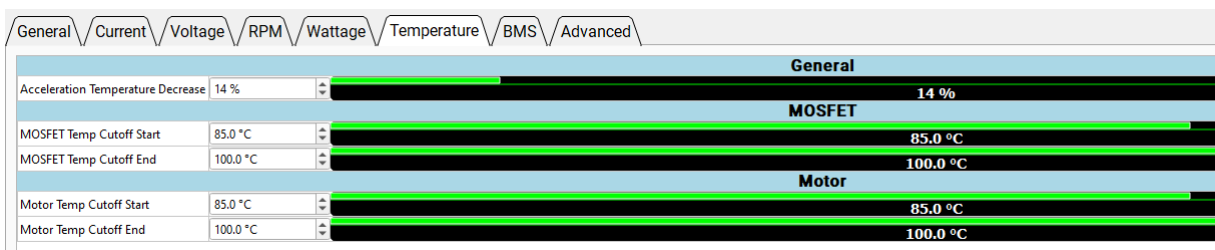
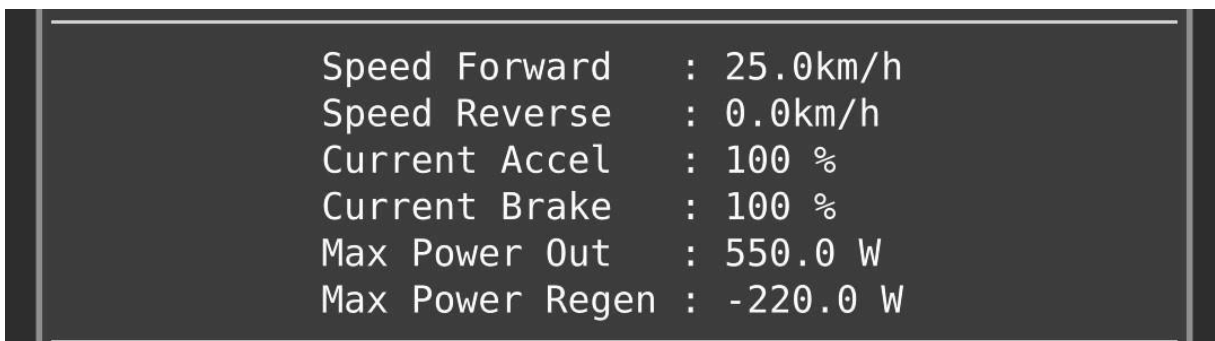


Figure 4.40 VESC Speed & Temperature Limitations

4.6.2.1 Motor Driver UART control

To communicate Raspberry Pi 4B with VESC, we are using UART control mode. VESC only supports its encoded message packets in the following format:

Table 4-7 VESC Encoded Packets

Byte	Function
One Start Byte	2 for short packets, 3 for long packets
One or Two Bytes	Specifies The packet length
Middle Bytes	The payload of the packet
Two Bytes	CRC checksum on the payload
One Stop Byte	Standart Value of 3

Here, we used a third-party open-source PyVESC library to encode and decode messages with Python. To read values from the Throttle and Brake lever, we used Adafruit Python ADS 1115 library. Because the used ADC is 16 bit and Throttle and Brake levers are supplied with 5 V, the values must be divided with $(2^{15}/5)$ to obtain accurate voltage levels.

The motor control software has two levels. The first level is when starting and applying the throttle from the stationary position. In this case, the maximum motor current that can be set is 15 A and is calculated with the NumPy interpolation function:

```

1. throttleVal = float(adc.read_adc(0, gain=GAIN, data_rate=128)/(2**15/5))
2. brakeVal = float(adc.read_adc(1, gain=GAIN, data_rate=128)/(2**15/5))
3.
4. motorCur = float(interp(throttleVal, [1.1, 4.7], [0, 15])) # Current (A)
5. brakeCur = float(interp(brakeVal, [1.1, 4.7], [0, 15]))

```

The second level is when the vehicle has achieved movement; in this case, the maximum current can be set increased to 20 A. There is no perceptible delay within setting motor and motor brake currents. To ensure rider safety, Brake has the highest priority.

Considering the Electric Scooter could be used for long-range trips to preserve rider's comfort Cruise Control function is also implemented. Cruise control only starts if Rider holds the throttle for more than 10 seconds after that last speed is set with *Motor.SetRPM()* function.

The driving decisions flowchart can be seen in Figure 4.41 below.

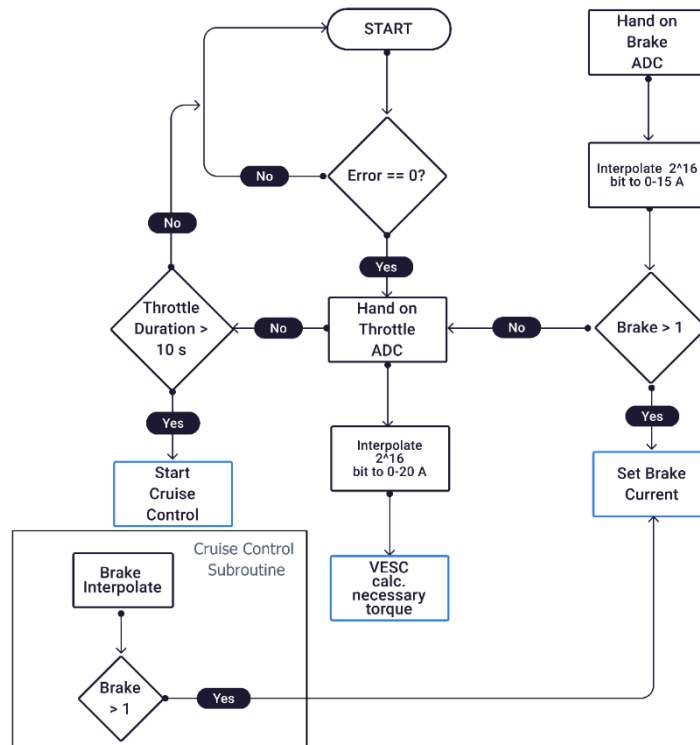


Figure 4.41 Motor Driving Flowchart

Because UART control is not instant, occasionally, the requested packages from the VESC can be sent back empty. To overcome these issues while driving and ensure that the driving program is working, we implemented error-catching exceptions supported in the Python programming language.

4.6.2.2 GPS System

GPS module connected to ECU (Raspberry Pi 4B) is SIM 808, as mentioned in the Electronic Design section. SIM 808 is connected to the UART0 port of RP4, which is the first PL011 port. By default, this UART port is used by the built-in Bluetooth module of Raspberry Pi. Despite the fact that this port can be used for SIM 808 by disabling the Bluetooth module from the kernel in Raspberry PI.

Standard AT commands controls SIM 808. The following AT commands shown in Table below must be used every time the board turns on. AT commands can be set from the minicom terminal or by The Python programming language.

Table 4-8 AT Commands for SIM 808 GPS

AT COMMANDS	DESCRIPTION
AT+CGPSPWR=1	Turn on the GPS module of SIM 808
AT+CGPSRST=0	Hard Reboot the GPS module to clear cache
AT+CGPSSTATUS?	Checks if GPS has fixed coordinates
AT+CGPINF=0	GPS data format: altitude, latitude, longitude

```

1. while(True):
2.     sim.write(b'AT+CGNSINF\r\n')
3.     tmp=''
4.     while(sim.inWaiting):
5.         a = str(sim.read())
6.         tmp = tmp+a
7.         if(a=="b"):
8.             break
9.
10.    tmp = str(result).split(",")
11.    requests.put('https://api.mirkeet.com/vehicles/1', json =
    {'lat':float(tmp[3]),'lng':float(tmp[4]),'battery':80})

```

The code snippet above is responsible for sending AT command to SIM 808, get data from the serial port then parse the GPS data to latitude and longitude only format and then send HTTPS PUT command and update the vehicles location data in the database.

4.6.3 Safety System

Safety system implementation to the electric scooter effectively reduces injuries associated with Rider's carelessness. There two modules: front and rear, that were developed. Those modules are depicted in the following Figure.

Before in the “MDS” report, we proposed implementing lidar to the rear side. However, after the second iteration, we replaced the rear lidar with the camera and a display.

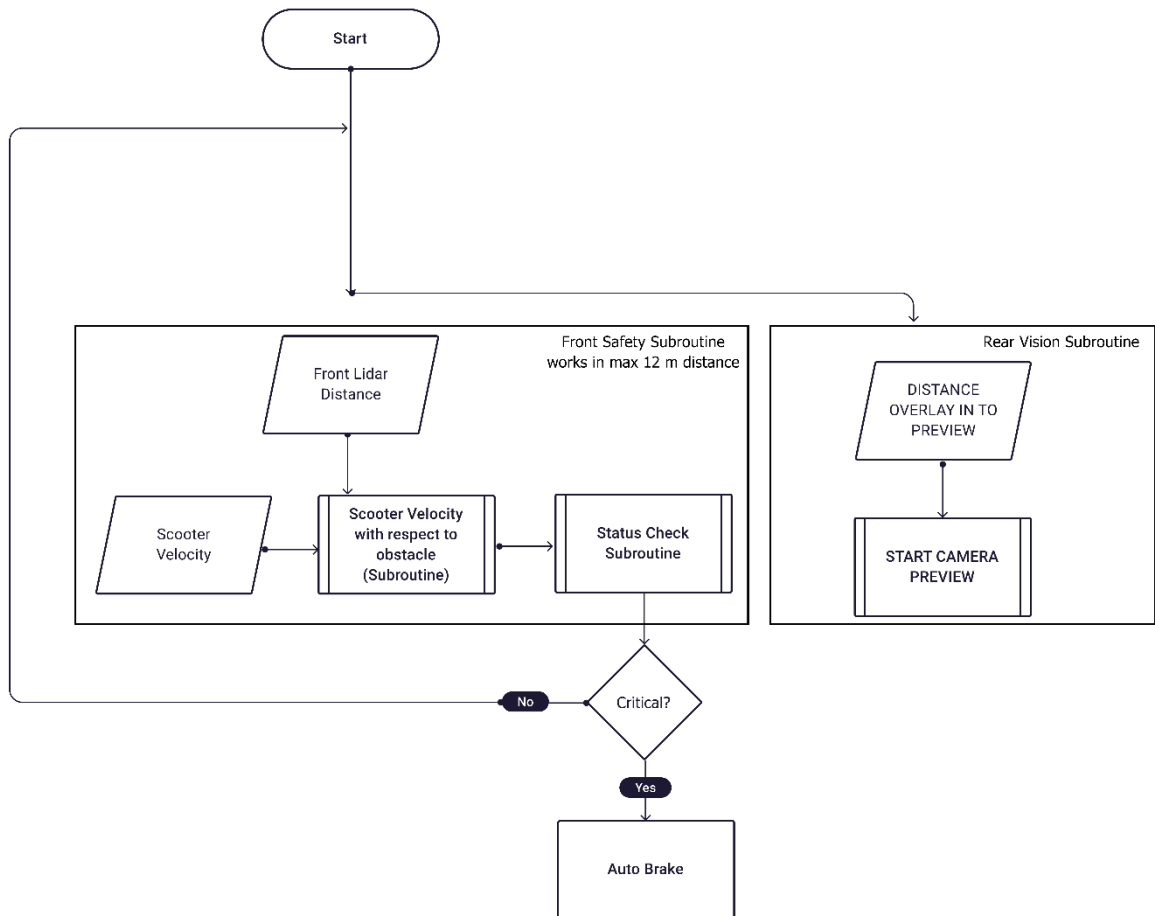


Figure 4.42 Safety system block diagram

As shown in the Figure above, the safety system activates when the frontal incident may occur.

The front safety subroutine activates the automatic braking in case of danger, and the rear safety subroutine previews with the Overlay the backside view.

4.6.3.1 Front Safety

The maximum working distance for the front safety module is 12m because of the Tf Mini Plus Lidar's working range restriction. Fortunately, this is not a restriction for us because if we calculate and plot the braking distance with the following equations in Python, Matplotlib:

$$d = \frac{v^2}{2\mu g} \quad (4.12)$$

Plotting equation 4.12 for three different road conditions (friction coefficients), we achieve the following Figure:

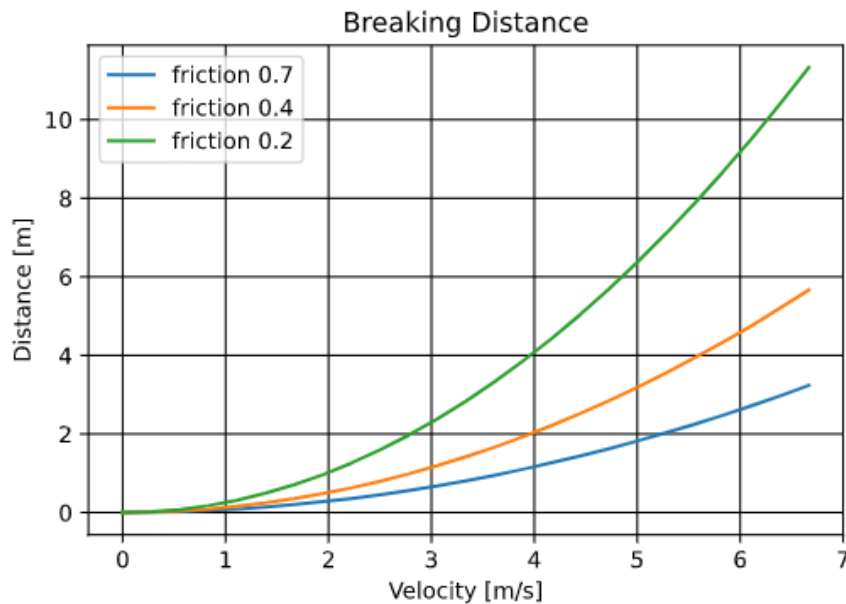


Figure 4.43 Braking Distances for different friction coefficients

As shown in Figure above, the Braking distance is always enough with the 12 m working range of Lidar. However, particularly in snowy weather, road conditions can be critical, which is already recommended by manufacturers not to drive electric scooters in such weather conditions.

On average, 4m is enough to stop the Electric scooters. By collecting lidar's distance data – d, the velocity of the car in front can be calculated by applying derivative:

$$V_{vehicle, scooter} = \frac{d_n - d_{n-1}}{t},$$

't' can be calculated from lidar's frequency.

By summing the scooter's velocity with a relative velocity of vehicle and scooter, the velocity of the vehicle in front can be calculated as below:

$$V_{vehicle} = V_{scooter} + V_{vehicle, scooter}$$

Then, we are calculating the braking distance of the velocity and car. Next, the criticality of the situation can be calculated by comparing their respective braking distances. Finally, we apply the immediate automatic braking system until the criticality status is negative. The formula below illustrates the logical operation:

$$\frac{v_{scooter}^2}{2\mu g} > \frac{v_{vehicle}^2}{2\mu g} + d$$

The Front Safety module is directly integrated with the motor driver module. As a result, lidar sensor data acquisition and braking distance calculation are parallel to the motor driving function. This way, the efficiency, Space, and Time complexities of the system are optimal.

For the process-based parallelism, we are using the multiprocessing library of the Python programming language. The following code snippet illustrates the multi-core process parallelism:

```

1. motorControlProc = multiprocessing.Process(
2. target=motor_control, args=(dist,))
3. lidarProc = multiprocessing.Process(target=lidar, args=(dist,))
4.
5. motorControlProc.start()
6. lidarProc.start()
7.
8. motorControlProc.join()
9. lidarProc.join()

```

Here “dist” variable is processed by lidar, and simultaneously Motor Driver function performs necessary operations with that updated variable.

The direct simulation of the front system is described in the Design Verification chapter.

4.6.3.2 Rear Safety

Because cars approaching from the rear side may have much greater velocity and neither

Scooter's nor human reaction time will be enough if the operating range was the same as front lidar. To address this problem, we used the camera and the display to show the camera preview with the distance overlay to Let the user make the safest decisions while riding.

Standard Overlay similar to the Rear view of the cars can be seen in the Figure below.

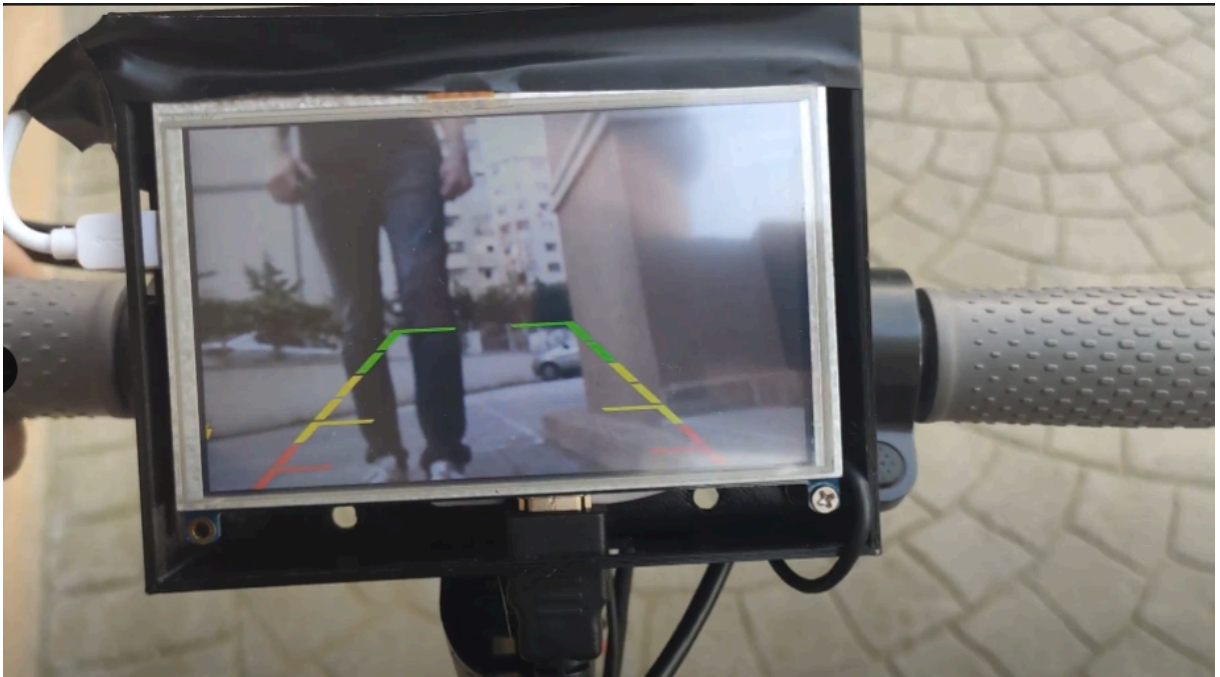


Figure 4.44 Rear Camera View with Overlay

5. DESIGN VERIFICATION

5.1 Analysis of Mechanical Design

Since all Electric Scooters components are not subjected to physical load, the part where the most stress will occur was analyzed.

The finite element method controls the mechanical design, and the Ansys Workbench program was preferred for applying this method.

Properties of these parts are shown in this Table from the Ansys analysis report.

Table 5-1 Bounding Box

Length X	892.51 mm
Length Y	289.68 mm
Length Z	165 mm

Table 5-2 Properties

Volume	9.6005e+005 mm ³
Mass	7.5364 kg
Scale Factor Value	1.

Table 5-3 Definition

Suppressed	No
Stiffness Behavior	Flexible
Coordinate System	Default Coordinate System
Reference Temperature	By Environment

Treatment	None
-----------	------

Table 5-4 Ansys Material

Assignment	Structural Steel
Nonlinear Effects	Yes
Thermal Strain Effects	Yes

5.1.1 Meshing

The general view of the fabricated mesh is as follows. The tetrahedrons method has a minimum mesh size limit of 5mm and a maximum 15mm element size. These measurements are the optimum values based on the conditions to be analyzed and the machine's capacity to be analyzed.

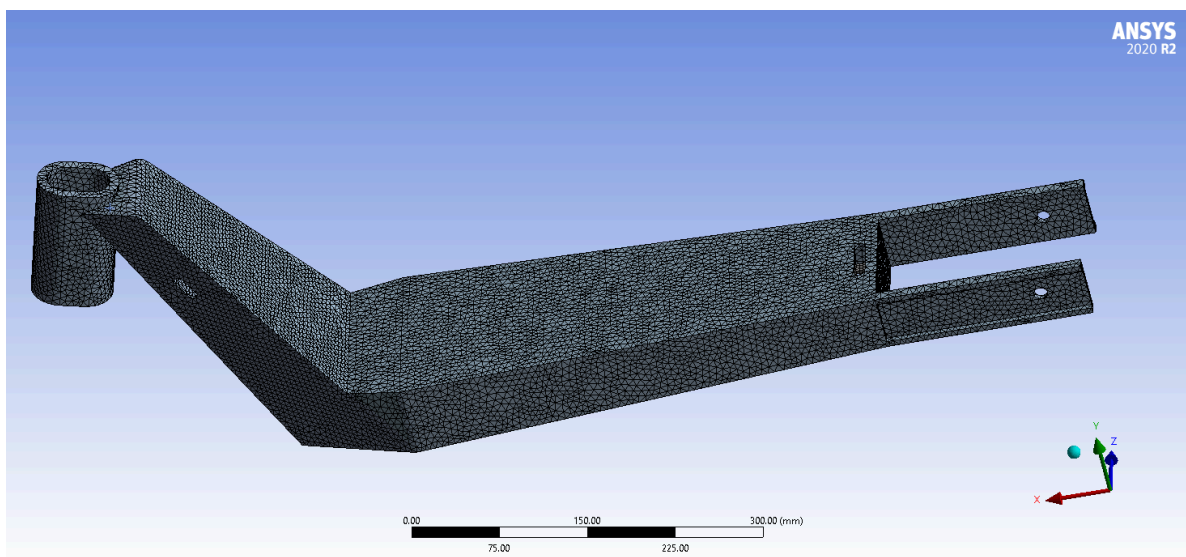


Figure 5.1 Meshing done in Ansys Software

Table 5-5 Mesh Statistics

Bodies	2
Active Bodies	2

Nodes	148416
Elements	68698

5.1.2 Mesh Quality

Mesh quality has been analyzed according to the specified criteria. These criteria are given below with explanations.

The value viewed with element quality is the ratio of a mesh element's volume to the edge length of that mesh element. Thus, it can be said that the closer its value is to 1, the better quality it is.

Here one can see the Element Quality of the mesh we have created for our design in figure 5.2 below.

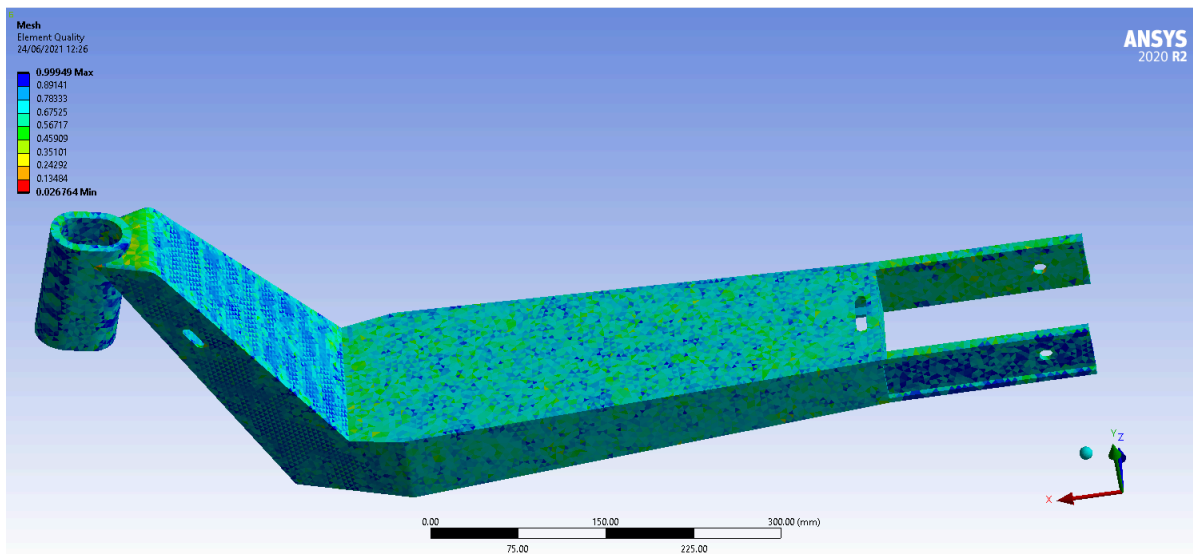


Figure 5.2 Element Quality analysis

Our mesh's evaluation is above average according to element quality if we examine the visual.

The aspect ratio is defined as the ratio of the shortest length of the element to the most extended length of the element, shown in Figure 5.3.

We understand that our mesh quality from this Figure is excellent, according to this criterion.

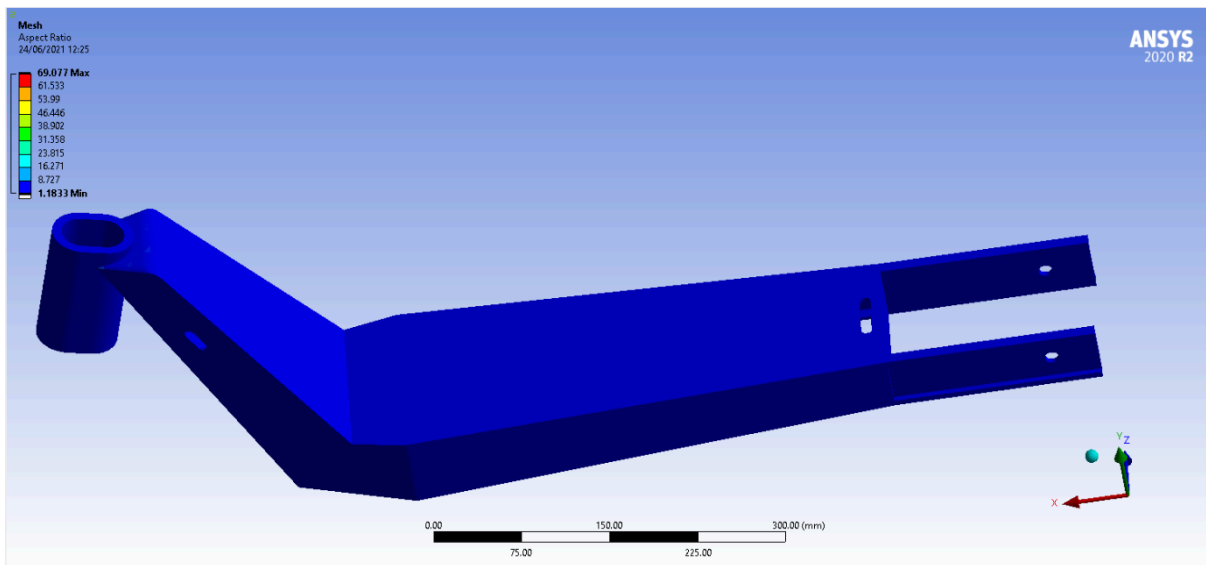


Figure 5.3 Mesh Aspect Ratio analysis

Jacobian is a measure of the deviation of a given element from an ideally shaped element. The jacobian value ranges from -1.0 to 1.0, where 1.0 represents a perfectly shaped element—our jacobian analysis results are shown in figure 5.4. Our mesh quality is excellent, according to this criterion.

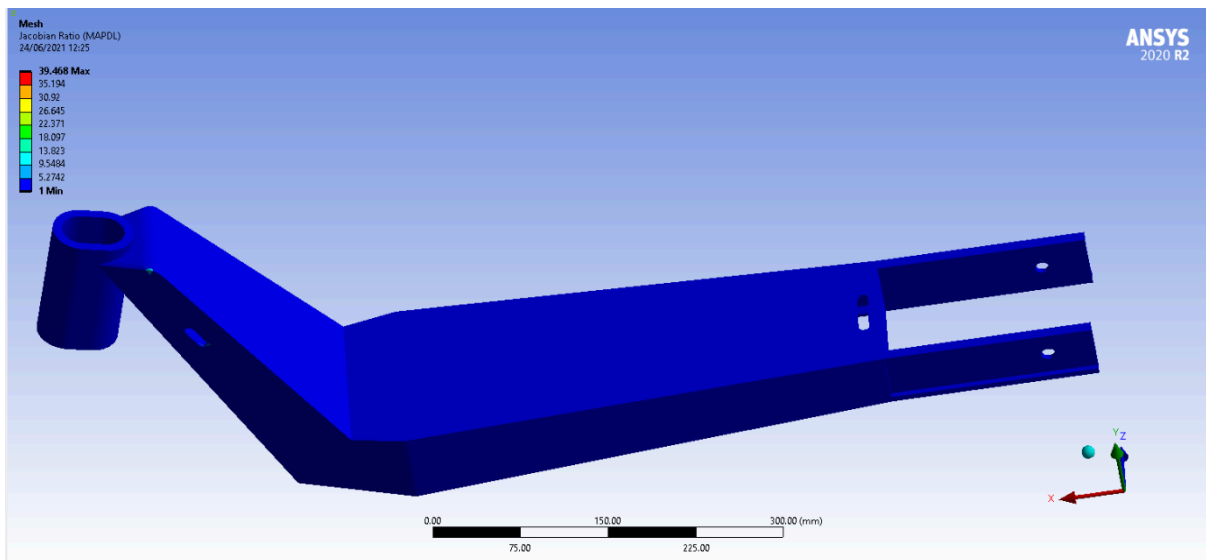


Figure 5.4 Mesh Jacobian Ratio analysis

Skewness determines how close to ideal a face or cell is. Our results which are in the acceptable range, are shown in figure 5.5.

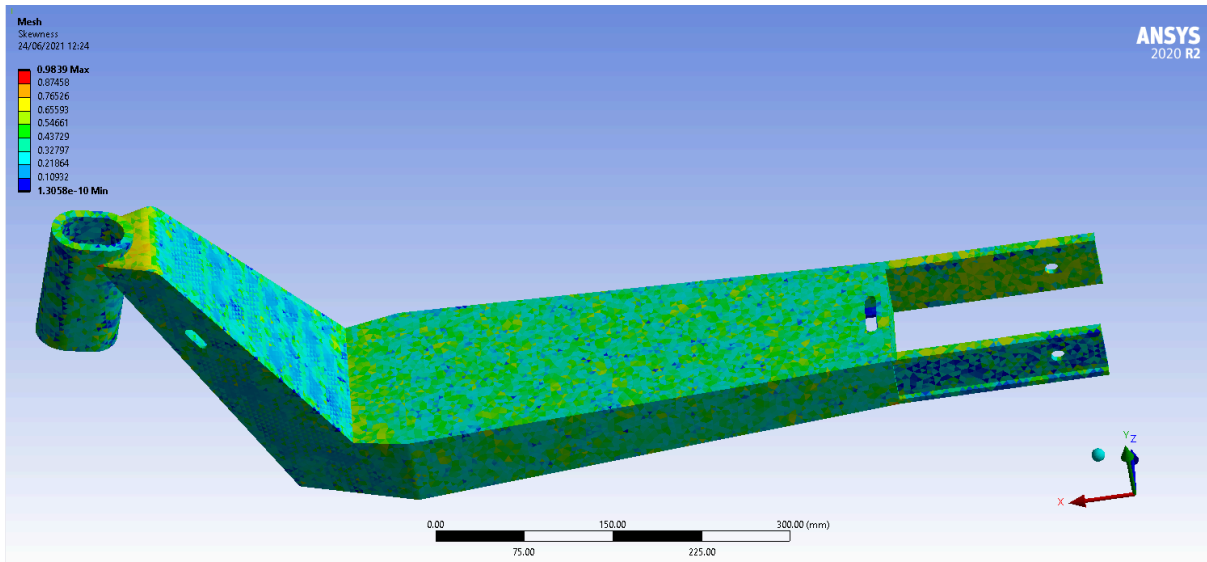


Figure 5.5 Mesh Skewness analysis

5.1.3 Structural Analysis

The equivalent elastic strain is defined as the limit for the strain values up to which the object will rebound and return to the original shape upon removing the load. Results can be seen in figure 5.6. and the Table below. When the maximum value is taken into consideration, it can be said that our model gives good results in this regard.

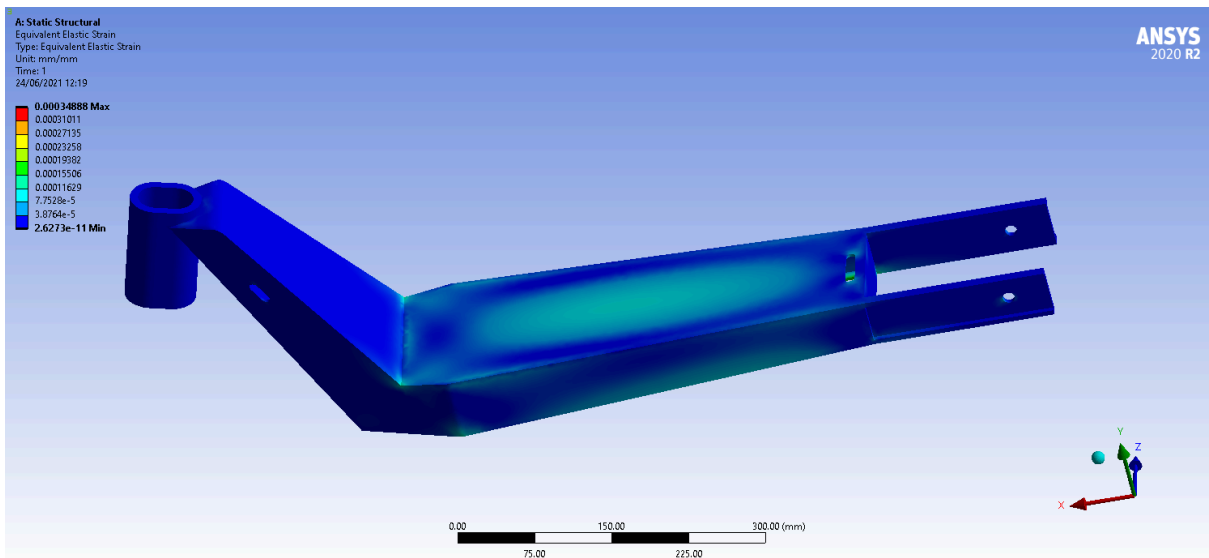


Figure 5.6 Equivalent Elastic Strain analysis

Table 5-6 Static Structural Data

Time [s]	Minimum [mm/mm]	Maximum [mm/mm]	Average [mm/mm]
1.	2.6273e-011	3.4888e-004	2.7552e-005

A total equivalent strain is calculated by summing equivalent elastic, plastic, creep (and thermal) equivalent strains shown in figure 5.7.

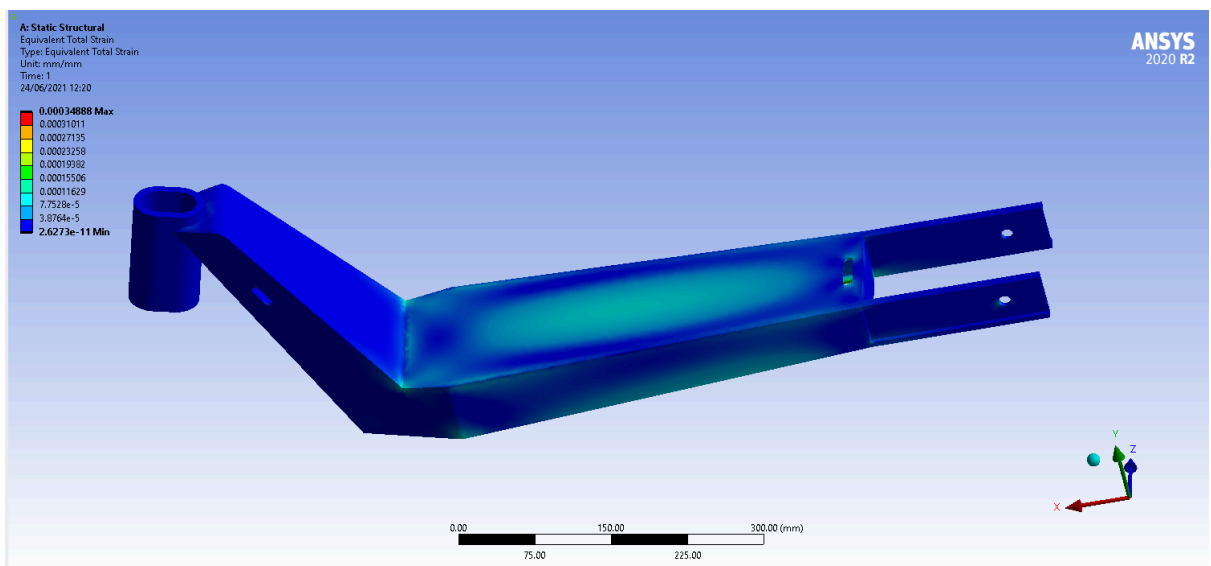


Figure 5.7 Equivalent Total Strain analysis

Figure 5.8 represents the total deformation of our model. The maximum total deformation is approximately 0.03 mm, which is perfectly acceptable for 1300 N applied force.

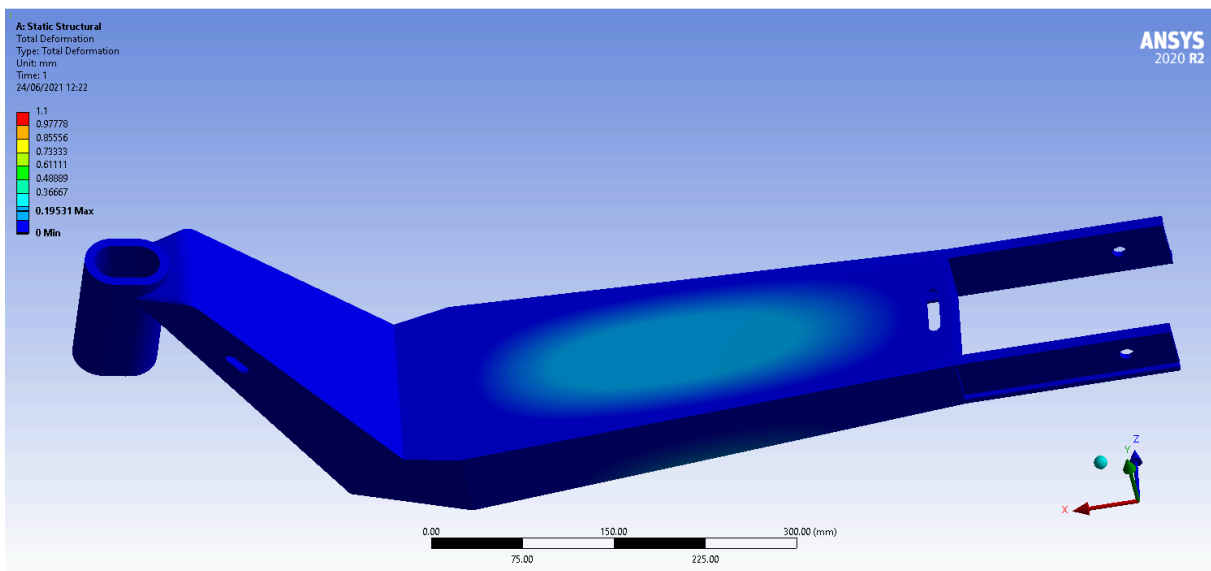


Figure 5.8 Total Deformation analysis

5.2 Cruise Control Simulink & VESC

After simulation results were obtained, the controller design was investigated. In our project,

BLDC Hub Motor was used as an actuator. In addition, a primary PID controller was developed in the Simulink environment to control the speed output, as seen in the Figure below:

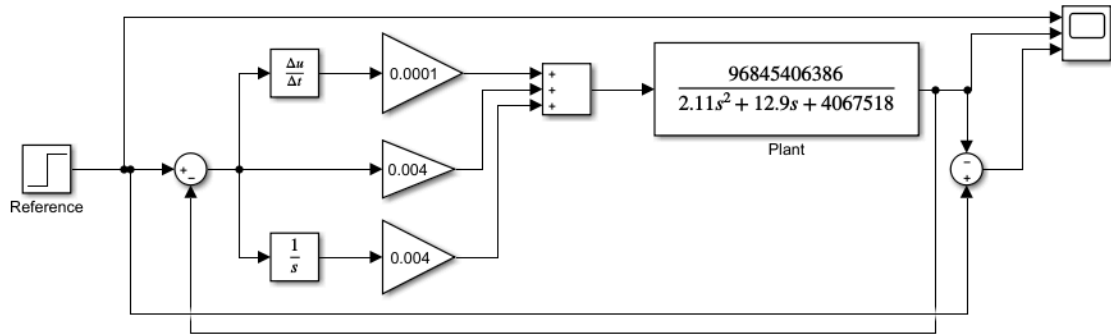


Figure 5.9 Simulation blocks for speed control

Some assumptions have been made in this controller design. Furthermore, some physical parameters (such as rotor moment of inertia, motor torque constant) must be known precisely to design the controller's plant part. Unfortunately, however, there was no information about any of these parameters on the manufacturer's side. So, instead, we made our design by taking references from some publications.

The Coefficient of Viscous Friction (B)	0.05
Motor Resistance (R_a)	0.102 ohm
Motor Inductance (L_a)	0.09161 H
Total Inertia (J)	0.01 kg.m ²
Motor Constant (K_e)	0.042

This PID controller's parameter was chosen. Parameters for our system like this:

$$K_p = 0.004, K_i = 0.004, K_d = 0.00010$$

As seen from the above model, the reference speed, the output speed, and the error between input and reference were plotted in the same scope. This scope could be seen in the Figure below. The real-world test results are also shown in the Figure below, where Graphs are presented in the VESC toolbox as ERPM values.

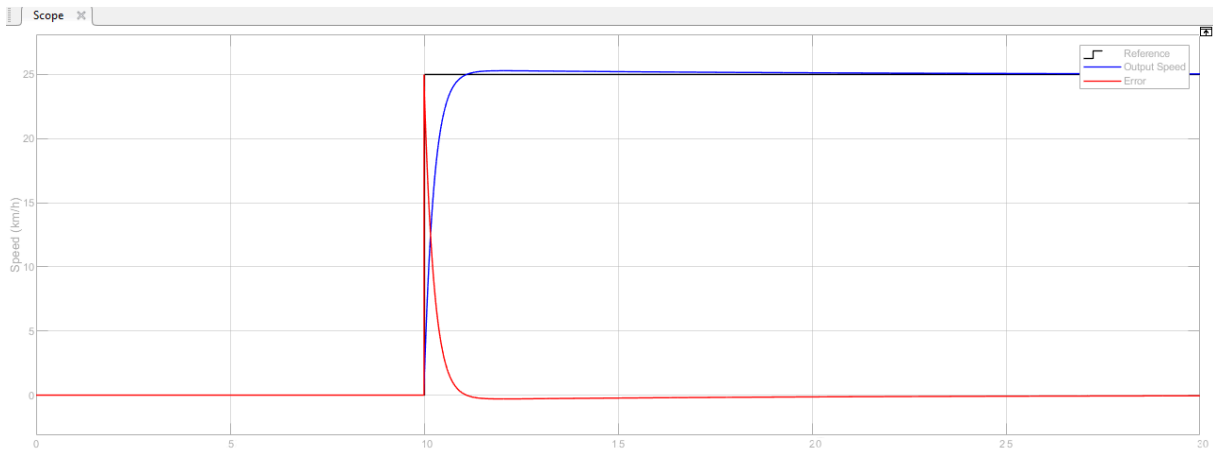


Figure 5.10 Comparison of input, output signals, and the error Simulink

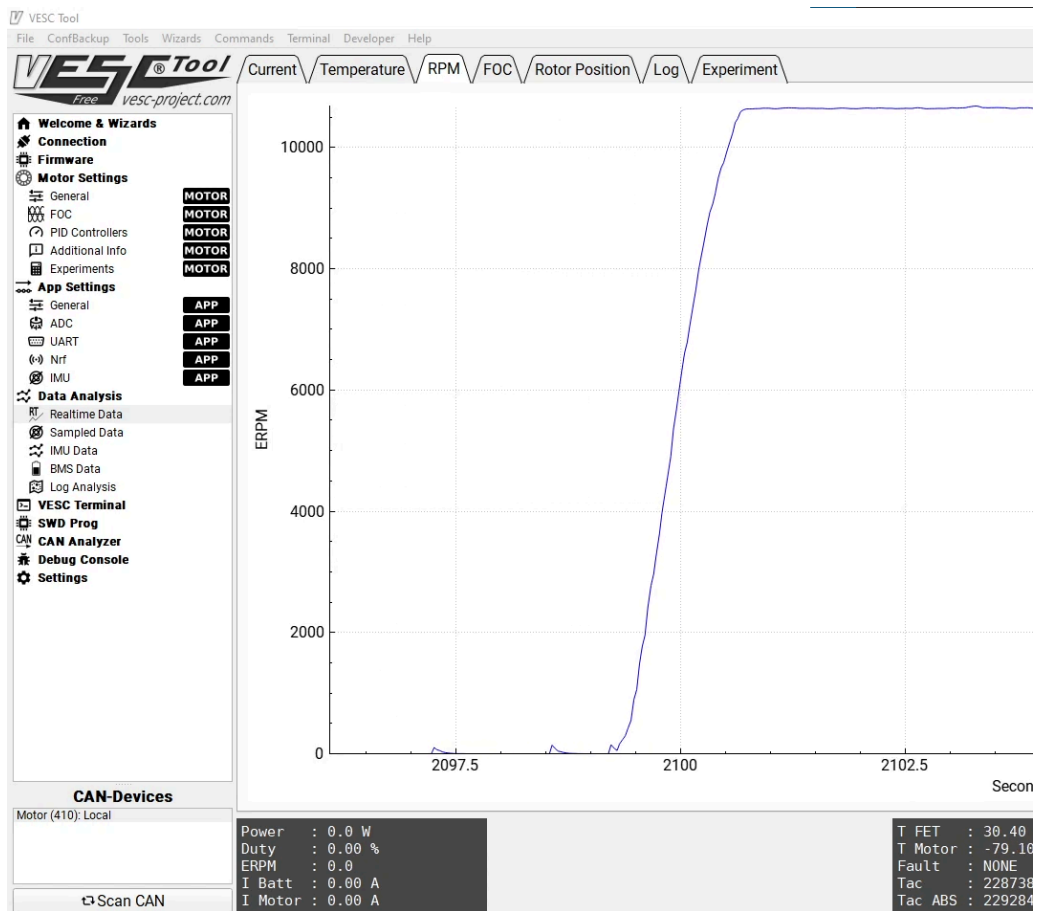


Figure 5.11 VESC ERPM Graph

5.3 Sharing System Software Tests

The Sharing system is made of complex architecture consisting of three different programming languages integrated; That is why separate unit tests and integration tests are necessary to test the system.

Manual unit test functions were written to test each module separately. These unit tests cover all of the functions in Sharing System and Motor Controller. In Python programming language, the standard Unit testing framework was used. The software was updated until the program passed all the test cases.

```

Overpathed 3 meter
Velocity= 4.535898384862246 , distance= 3.559580141545193 , rel_vel= -3.559580141545193 , ds= (2.6216073085876563, 6.654237164444989)
Velocity= 4.241229516856366 , distance= 4.346175865097999 , rel_vel= -4.346175865097999 , ds= (2.2920524738345676, 7.253636217543123)
Velocity= 4.060768987951169 , distance= 5.164961394239614 , rel_vel= -5.164961394239614 , ds= (2.1011524940756834, 7.943033008839336)
Velocity= 4.0 , distance= 5.999999999999997 , rel_vel= -5.999999999999997 , ds= (2.038735983690112, 8.708013546667118)
Velocity= 4.0607689879511675 , distance= 6.835038605760391 , rel_vel= -6.835038605760391 , ds= (2.1011524940756816, 9.533124923690613)
Velocity= 4.2412295168563645 , distance= 7.653824134901985 , rel_vel= -7.653824134901985 , ds= (2.2920524738345653, 10.402535851074877)
Velocity= 4.535898384862243 , distance= 8.440419858454792 , rel_vel= -8.440419858454792 , ds= (2.6216073085876532, 11.30042252956737)
Velocity= 4.935822227524085 , distance= 9.179515585399232 , rel_vel= -9.179515585399232 , ds= (3.1042738355913375, 12.211056241506345)
Velocity= 5.428849561253838 , distance= 9.856725658119238 , rel_vel= -9.856725658119238 , ds= (3.7554036134971946, 13.118622439622676)
Velocity= 5.999999999999994 , distance= 10.45886895286436 , rel_vel= -10.45886895286436 , ds= (4.587155963302743, 14.00684976908327)
Velocity= 6.631919426697317 , distance= 10.974225435330247 , rel_vel= -10.974225435330247 , ds= (5.604275647579673, 14.85856428442813)
Velocity= 7.30540728933227 , distance= 11.392764277795 , rel_vel= -11.392764277795 , ds= (6.8003281935562, 15.655303600565695)
.....
-----
Ran 36 tests in 0.004s
OK

```

Figure 5.12 Python Successful UnitTests

The third-party software The Postman was used to test the Backend API. All of the six endpoints of the API were tested and passed cases successfully. Additionally, integrations tested were performed manually by experimental methods.

The screenshot shows a Postman API test for a GET request to `https://api.mirkeet.com/vehicles/`. The response is a JSON array of two vehicle objects. The first object has an id of 2, a battery level of 75, and is located at latitude 41.069967299999995 and longitude 28.9929652. The second object has an id of 1, a battery level of 80, and is located at latitude 40.987357 and longitude 29.049642. Both vehicles have a distance of 0.0 and are not rented.

```

1  {
2    "id": 2,
3    "battery": 75,
4    "lat": 41.069967299999995,
5    "lng": 28.9929652,
6    "distance": 0.0,
7    "rented": false
8  },
9  {
10   "id": 1,
11   "battery": 80,
12   "lat": 40.987357,
13   "lng": 29.049642,
14   "distance": 0.0,
15   "rented": false
16  }
17 }
18

```

Figure 5.13 Postman API Tests

5.4 Safety Algorithm Simulation

A lidar-based safety system has been simulated in Python to verify the algorithm mentioned in the chapter **Safety System**.

A random sinusoidal data generator was used to simulate lidar's response data, as shown in figure 5.14 below:

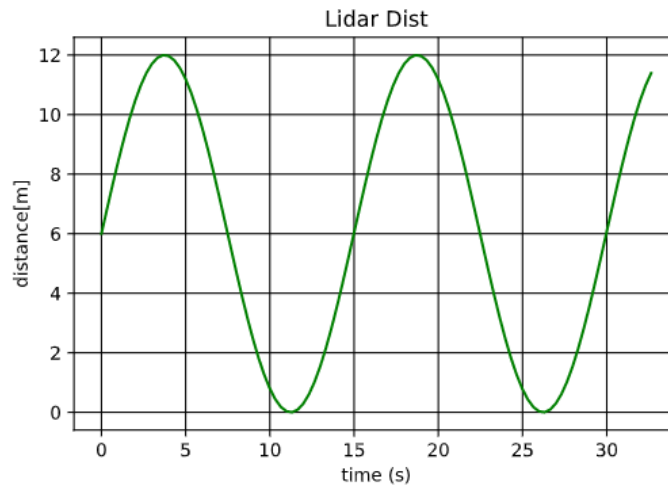


Figure 5.14 Lidar Distance simulation plot

The car's velocity, which can be calculated with the mentioned method before, can be seen in figure 5.15:

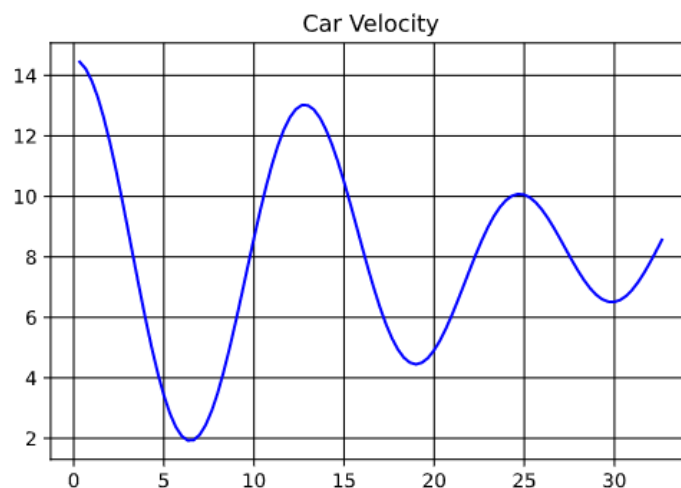


Figure 5.15 Front car's velocity simulation plot

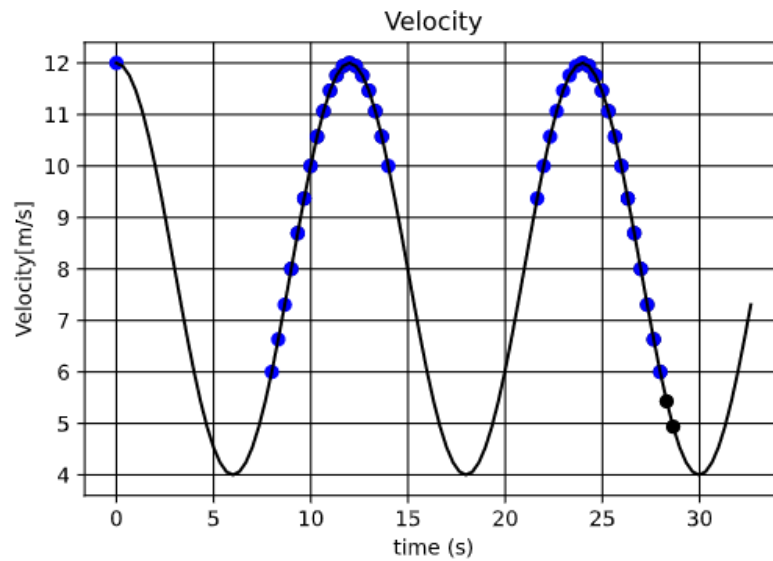


Figure 5.16 Results of the Front safety system

Figure 5.16 indicates the electric scooter's velocity, and blue dots indicate the critical situations where the auto-braking system must activate. Due to simplicity, the simulation does not account for reducing speed and only shows the critical situations.

6. ASSEMBLY AND PRODUCTION STAGES

Unlike the simple parts to manufacture and can be easily obtained from abroad, our piece named deck was carried out directly in the workshop due to its unique features and design. Furthermore, as we mentioned before, due to the production conditions problem caused by the Covid 19 pandemic, carbon steel was preferred instead of Aluminum 6061 alloy for the prototype.

In order to carry out the production, our raw material, 3mm thick carbon steel sheet C75S (CK75), was supplied in 620x2000 dimensions. The production of the part was carried out in 6 stages and explained with pictures.



Figure 6.1 Cutting the sheet in the dimensions determined and bending



Figure 6.2 Welding the fork and optimizing the wheel arch



Figure 6.3 Bending the neck of the main part



Figure 6.4 Welding to the main part



Figure 6.5 Welding the Handlebar stem seat to the 'Neck'



Figure 6.6 5th Adding screw slots and other details



Figure 6.7 Testing by assembling the produced part

After the production was made and all the mechanical parts were assembled and tested, the part was disassembled again and prepared for the painting process for visual improvement. In order to decide on the color to be painted, the RAL catalog was examined and deemed appropriate to use the RAL 5013 (dark blue) color. Electrostatic powder paint was used for painting the piece.



Figure 6.8 Painted Deck

After completing all these processes, all the main parts of the Mirkeet Electric scooter were assembled, and the first driving tests were carried out.



Figure 6.9 Prototype's Mechanical Assembly

After the harmonious and successful operation of the main components has been confirmed, the assembly and electronic connections of the other components (lidar, display, camera, GPS, etc.) that help the driving have been made and tested. One can see the photos of the final product in the Figure below.



Figure 6.10 Final Product

7. FINAL SYSTEM TESTING

After performing the system assembly, we prepared the electric prototype vehicle. First, both batteries were charged, and final electrical cables and connections were checked. Then, the first Tests were performed while the system was fixed to the stationary table, as shown in Figure 7.1 below.



Figure 7.1 Pre-Test Vehicle Position Preparation

To acquire data from VESC, Flipsky Bluetooth Low Energy Module was connected to VESC. Then, from Android App VESC Tool, mobile phone and VESC were paired. With This data logging tool, the following information can be analyzed later with the Log Analysis program:

- Electric Scooter's Speed, Time of Day
- Time, the distance of a trip
- Motor's current, battery's current
- Motor's power

- ERPM, Duty Cycle
- Fault Codes
- MOSFET & Motor Temperatures
- Ah Used, Ah Charged
- Wh Used, Wh Charged

During testing, Motor parameters were constantly updated from the Mobile App to reflect the best ride quality.

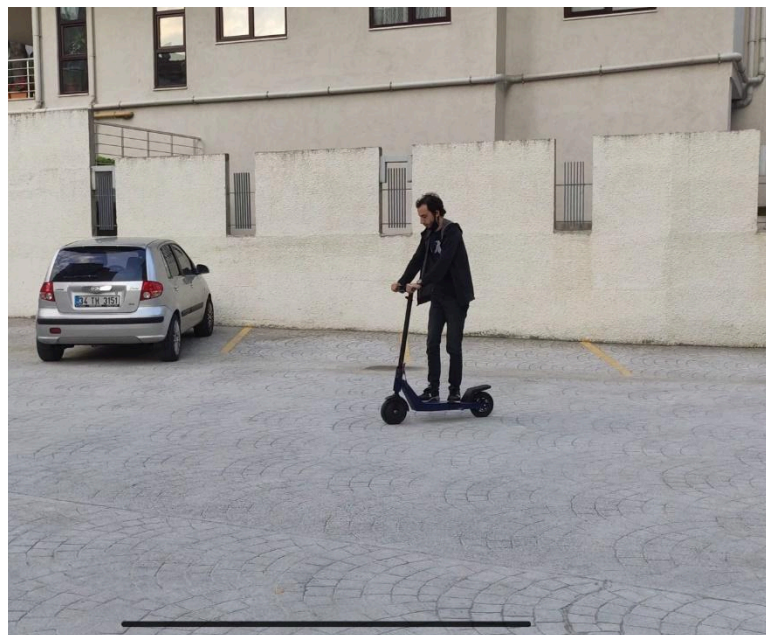


Figure 7.2 Rider testing the Scooter

After the ride, necessary data was analyzed in the VESC tool.

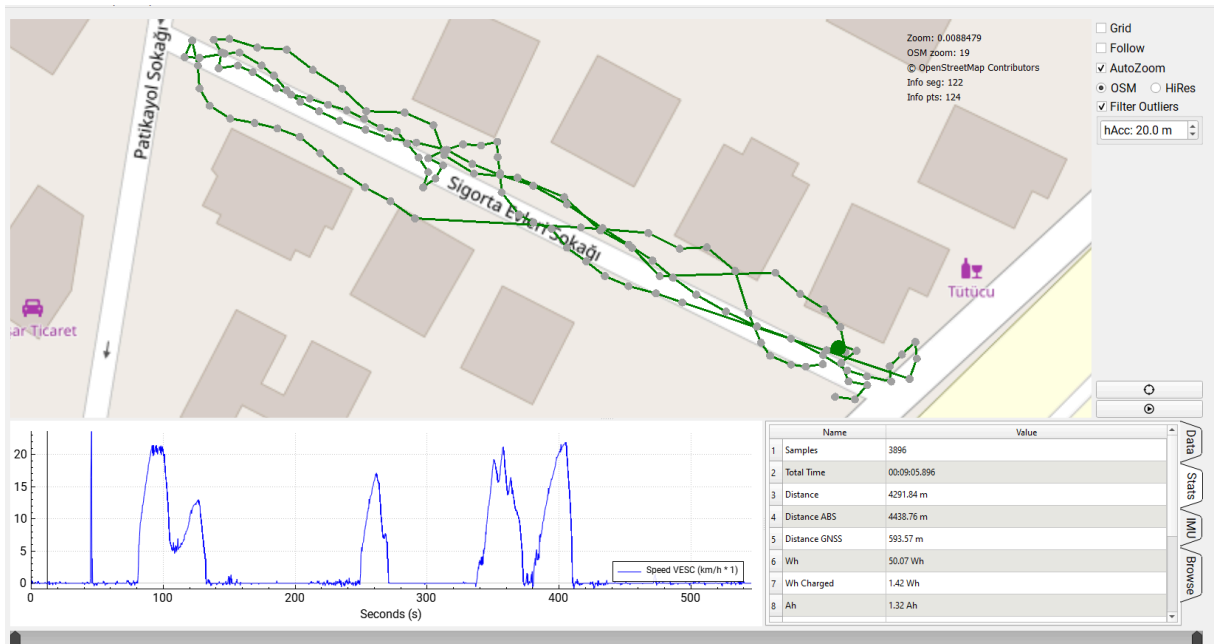


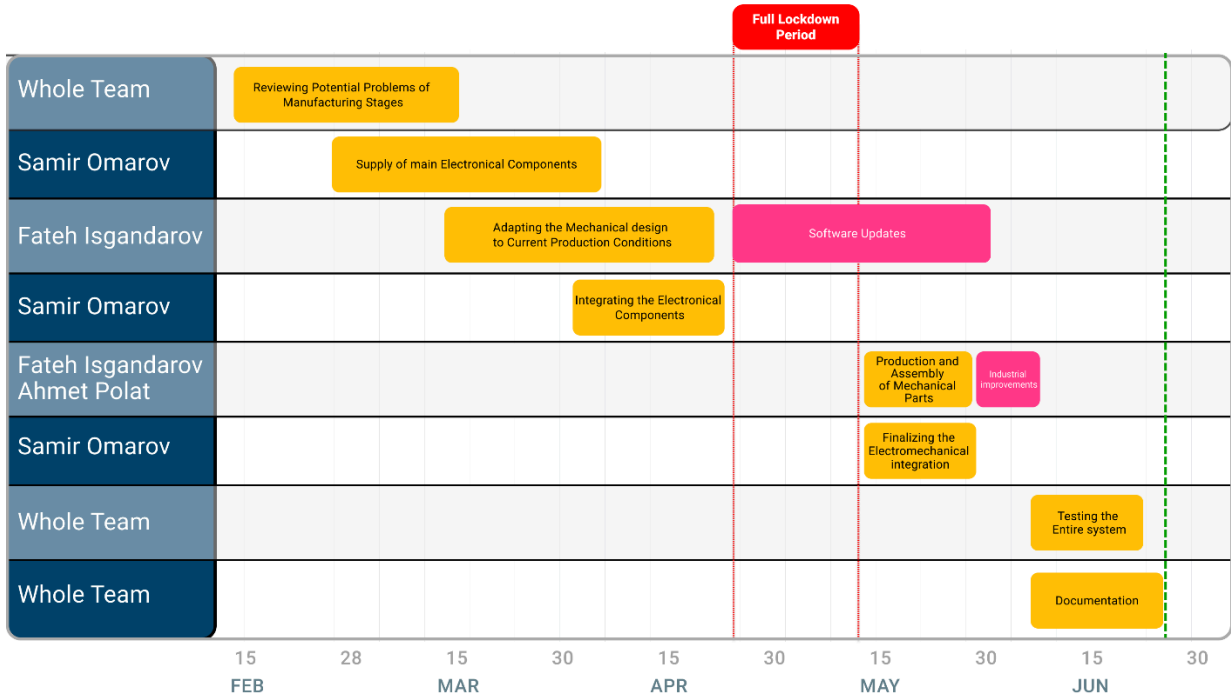
Figure 7.3 VESC Log Analysis

8. WORKING SCHEDULE

Product Roadmap 2021 Lockdown Updated V2

2021

Electrical Scooter Design

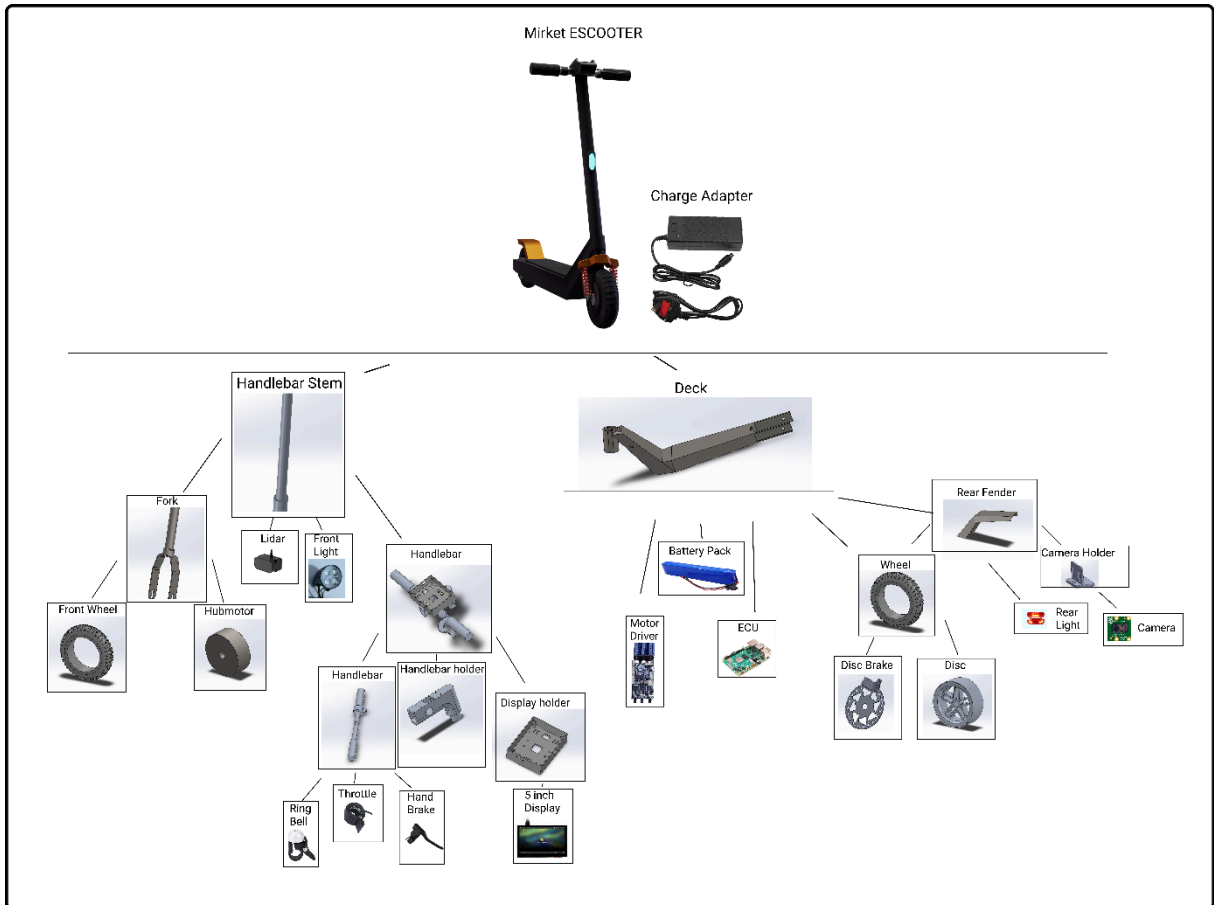


9. BUDGET

9.1 List of Materials

Product Name	Model	Quantity	Cost
Hub Motor and Front Wheel	Volta H.VT1 Motor	1	900
Rear Wheel	Volta VT1 Pneumatic Wheel	1	180
Battery Pack	Xiomi 36V 7.5 AH li-ion Battery Pack	1	700
Motor Driver	Flipsky FSESC 4.12	1	400
Handlebar Set	Xiaomi mi 365 Handlebar	1	650
ECU	Raspberry Pi4 2gb	1	395
ADC converter	ADS 1115	1	30
GPS & GPRS	Sim 808	1	220
Front Lidar	TF Mini Plus lidar	1	429
Back Camera	Raspberry Pi (Camera) Module V2	1	350
Display	Raspberry Pi 5 Inch HDMI Lcd Ekran 800×480	1	443.47
Disc Brake Set	QS Motor Brake Kit	1	234
ECU Power Supply	Philips 10000 Mah Dlp 1710CW/97	1	87
Raw Material	Carbon Steel	10 kg	416
Lights	Speed Light	2	53
Ring Bell	Bianchi Plus	1	18
Total			5505 TRY

9.2 Product Tree



10. PRODUCT BACKLOG ITEMS

Sprint	User Story Role	User Story Name	Tasks
1	Project Team Member Project Manager	Preparing Work Plan	Determining Full Time Chart
			Determining the distribution of tasks
			Preparing Work Plan
2	Project Team Member Mechanical Designer	Designing the Overall Mechanical System	Making a Concept Design
			Determine the mechanical components to be used
			Making technical drawings
2	Project Team Member Electrical System Designer	Designing the Overall Electronical System	Determining the basic electrical components to be used
			Define the electrical power Requirements
			Making technical drawings
2	Project Team Member Software Engineer	Developing Mobile App REST API, MQTT, Web Dashboard	A Mobile application with a secure, bug-free, user-friendly interface needs to be developed.
			Unit Testing, Interface Testing
3	Project Team Member Electrical System Designer	Specify Electrical Components	Determining motor specifications
			Choosing the motor to be used
			Determine the motor drive to be used
			Determining safety sensor specifications
4	Project Team Member Mechanical Designer	Mechanical Analysis	Identifying the parts that will be subject to hard strain
			Determining the method of analysis
			Preparing the determined parts for analysis
			Performing simulation at ANSYS
4	Project Team Member Software Engineer	Safety System	Develop Lidar based algorithm
			Test Lidar based algorithm
			Deploy control system to Scooter ECU
4	Project Team Member Electrical System Designer	Electrical Simulations	Performing speed cruise control PID simulation

11. CONCLUSION

In this study, the problems arising from the inefficient use of vehicles were examined, the deficiencies of the existing micro-mobility vehicles were determined, and a new generation transportation vehicle was designed and produced and attempted to eliminate these deficiencies and to prevent the accidents that may occur during the use of these vehicles, which may cause human health risks.

The concept design of the project was made using the Blender 3D modeling program. Technical drawings were created using the Solidworks CAD program. After selecting the material to be used, this material's durability tests were made using the Ansys Workbench program with the finite element method. The design problems were solved according to these outputs. Electromechanical parts were selected according to the outputs obtained at the end of this design process. The power and required power source of the vehicle determined the appropriate technical requirements. The required electrical components were selected based on market requirements. The mobile user application with QR code integration and web admin dashboard was developed for the electric scooter sharing system. We believe that by carrying out this research, we will significantly impact the technological development of short-distance vehicles, making life easier and safer with intelligent vehicles and the widespread use of environmentally friendly dockless vehicles in the future.

Batteries, which are the power sources to be used within the project's scope, can be improved. Furthermore, with the lithium production facility opened in our country, the design cost can be further reduced if the battery production is produced domestically.

Another development that can be made for the future is to enable the E-scooter service to communicate with smartwatches. Operations such as renting, dropping, locking, etc., can be done with these devices.

During the final system tests, it was unfortunate that low efficiency has occurred on rough roads due to the camera shaking and the resulting deterioration of vision on display. In order to prevent this, it is possible to improve the images taken with software updates and mechanical anti-shake parts.

During the project realization, the system control and server communication parts were

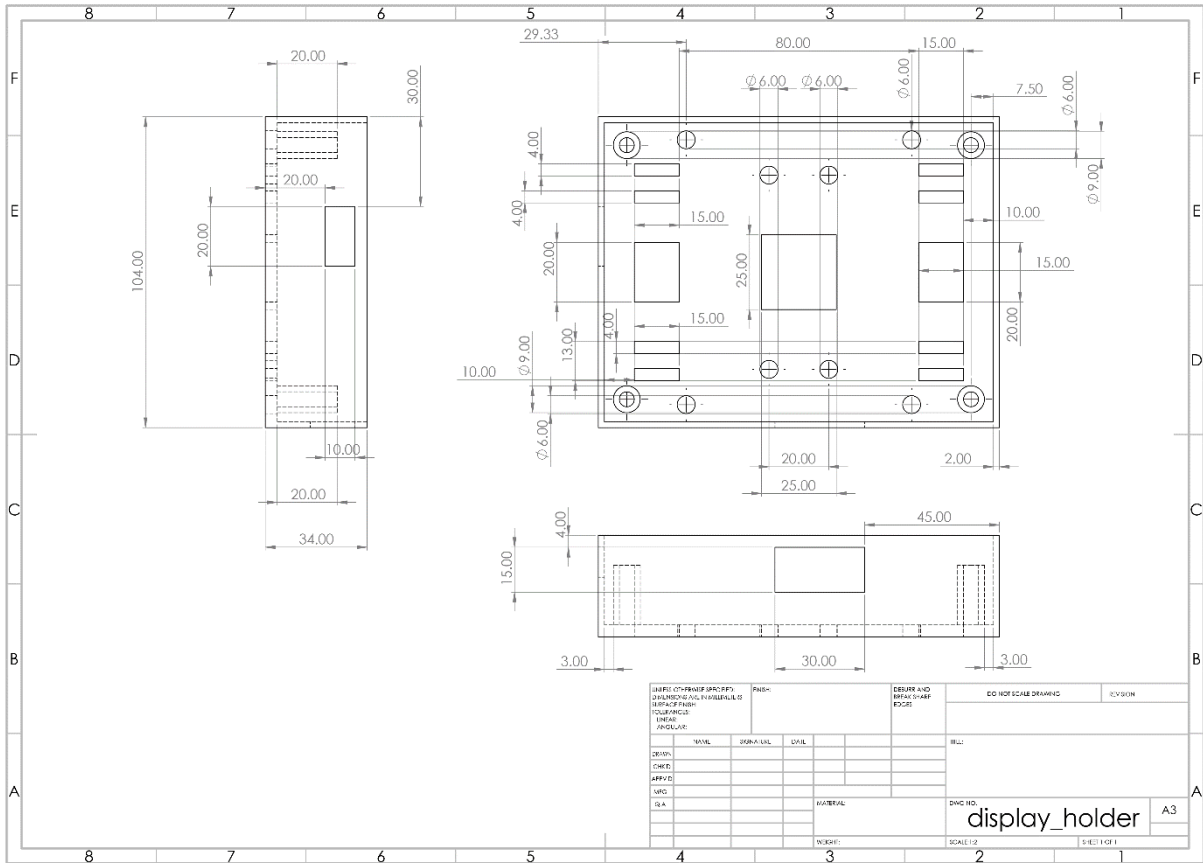
cooperatively designed with MIRKEET Bike Branch. As a result, the controlling communication algorithms principles were common, and the collective work realized the server's database structure.

REFERENCES

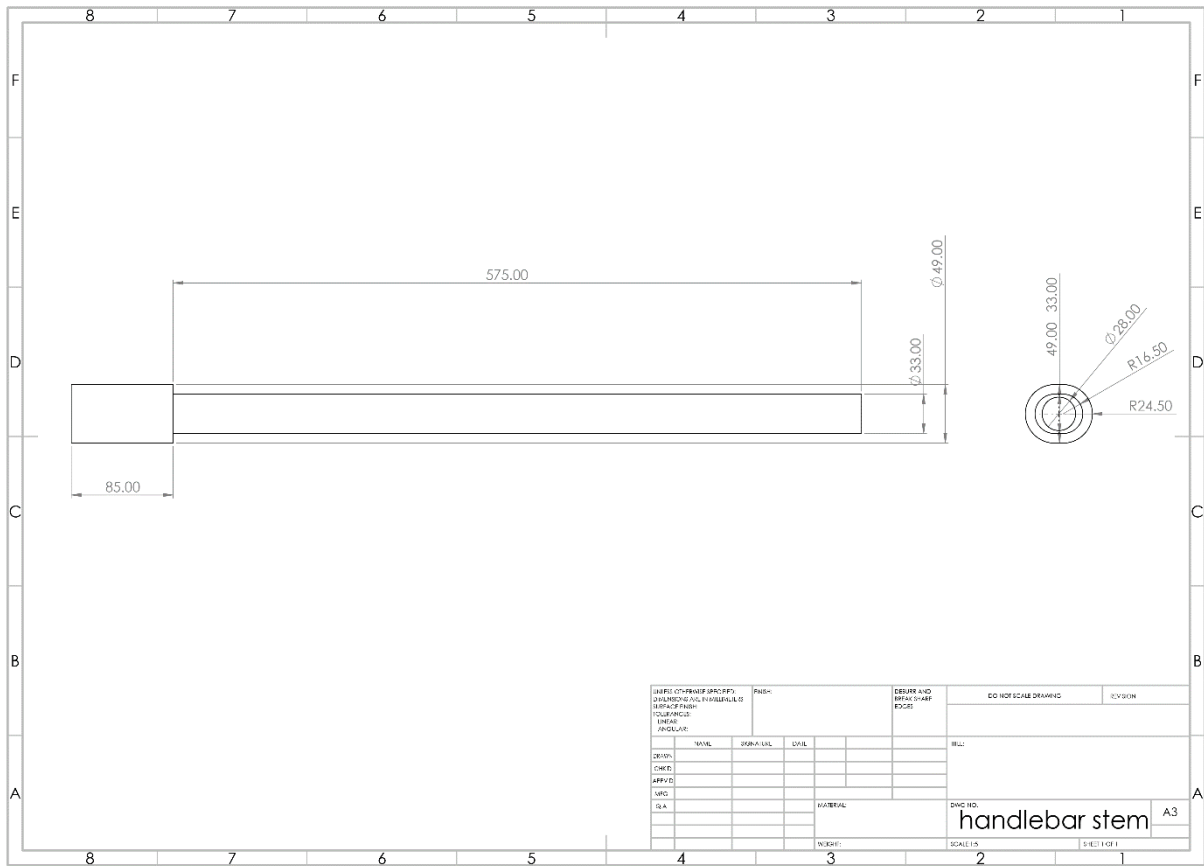
- [1] Haerri, V.V. and Madawala, Udaya and Zabkar, Daniel, "An Electric Micro-Scooter with a Super-capacitor Energy Buffer," Conference Paper, Lucerne, 2008.
- [2] Mahesh S. Khande, Akshay S. Pati, Gaurav C. Andhale, Rohan S. Shirsat, "Design and Development of Electric scooter," *International Research Journal of Engineering and Technology (IRJET)*, vol. 07, no. 05, pp. 359-364, 2020.
- [3] O. S. Burheim, "Secondary Batteries," in *Engineering Energy Storage*, Academic Press, 2018, pp. 111-145.
- [4] Trivedi, T. K., Liu, C., Antonio, A., Wheaton, N., Kreger, V., Yap, A., Schriger, D., & Elmore, J. G, "Injuries Associated With Standing Electric Scooter Use," *JAMA Netw Open*, 2019.
- [5] Royo, Santiago & Ballesta-Garcia, Maria, "An Overview of Lidar Imaging Systems for Autonomous Vehicles," *Applied Sciences.*, vol. 9, no. 19, 2019.
- [6] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine, 2000.
- [7] "What Is Finite Element Analysis?," [Online]. Available: <https://www.mathworks.com/discovery/finite-element-analysis.html>. [Accessed 23 January 2021].
- [8] "Sheet Metal Forming," [Online]. Available: <https://www.custompartnet.com/wu/sheet-metal-forming>. [Accessed 25 February 2021].
- [9] A. J. Hawkins, "Bird's electric scooters are getting more rugged to handle heavy use," *The Verge*, 24 09 2018. [Online]. Available: <https://www.theverge.com/2018/9/24/17886390/bird-electric-scooter-share-ceo-travis-van-derzanden>. [Accessed 4 01 2021].

- [10 Pawar, Amey & Shenoy, Ashwin & Wartika, Samson & Thavre, Srikant & Sasane, Kamlesh, "Design and Development of Electric Scooter with Regenerative Braking.," 2019.
- [11 K. S. SUSHANTA, "Sengerandu's Tutorials," April 2013. [Online]. Available: <https://sengerandu.wordpress.com/tutorials/automobile-engineering/resistance-to-motion/#:~:text=The%20component%20along%20the%20road,hard%20surfaces%20produces%20a%20friction..> [Accessed 15 January 2021].
- [12 Dawood, Younis & Mahmood, Ali & A. Ibrahim, Muhammed, "Comparison of PID, GA and Fuzzy Logic Controllers for Cruise Control System," *International Journal of Computing and Digital Systems.* , vol. 7, no. 10, pp. 311-319, 2018.
- [13 FASTAPI, "FastAPI," May 2020. [Online]. Available: <https://fastapi.tiangolo.com>. [Accessed 11 May 2021].
- [14 escooterguide, "Technical Guide: Electric Scooter Brakes," 21 12 2020. [Online]. Available: <https://electric-scooter.guide/guides/electric-scooter-brakes/>.
- [15 Weinert, Jonathan and Ogden, Joan and Sperling, Daniel and Burke, Andy, "The future of electric two-wheelers and electric vehicles in China," *Institute of Transportation Studies, UC Davis, Institute of Transportation Studies, Working Paper Series*, vol. 36, 2008.

APPENDIX 3



APPENDIX 4



APPENDIX 5

