Software Requirements Specification

for

Smart Jeepney Monitoring: A GPS Integrated Tracking and Utilization Platform for SPTGTSC

Version 1.3

Prepared by

Group Name: OUTCODE

Amor, Richard Gabriel	2022-130806	amorre@nu-fairview.edu.ph
Chico, Christopher Jake	2022-131282	chicocr@nu-fairview.edu.ph
Culilap, David Orlan	2022-132195	culilapds@nu-fairview.edu.ph
Minas, Brylle Justin	2022-132781	minasbc1@nu-fairview.edu.ph
Quiambao, Lance Harvey	2022-132704	quiambaolr@nu-fairview.edu.ph

Instructor: Kris P. Garcia

Course: BSIT - MI

Date: October 2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	Richard Gabriel Amor	Initial draft for the SRS during midterm phase.	08/20/25
Version 1.1	Richard Gabriel Amor	Post-consultation/defense revisions	09/10/25
Version 1.2	Richard Gabriel Amor	Overhaul of SRS for adjustments of the system	09/20/25
Version 1.3	Richard Gabriel Amor	Complete version of the SRS	10/02/25

1 Introduction

This chapter provides an overview of the “Smart Jeepney Monitoring: A GPS Integrated Tracking and Utilization Platform for SPTGTSC”, outlining the purpose and scope of the system, its intended users, and the structure of this document. It introduces the key definitions, acronyms, and conventions used throughout the paper, as well as references and acknowledgments relevant to the project.

1.1 Document Purpose

The purpose of this document is to define the general requirements for the "Smart Jeepney Monitoring: A GPS Integrated Tracking and Utilization Platform for SPTGTSC” system project, developed by the OUTCODE team. This document also specifies the features, functions, and constraints of the system for its initial release version, serving as a guide for both development and evaluation. It ensures that all stakeholders, including the Sapang Palay Tungko Grotto Transport Services Cooperative (SPTGTSC), development team, and future maintainers, share a common understanding of the system’s objectives and requirements. In addition, this document sets the foundation for later sections that describe the overall system design, specific functional details, and non-functional requirements, ensuring consistency and alignment throughout the project.

1.2 Product Scope

The project is a technology solution designed to optimize jeepney operations and commuting experience of passengers with the cooperative’s fleet of modernized jeepney. The complete system integrates hardware components (GPS and Camera Module) with software platforms (Mobile App and Website Platform). The system is intended for four key user groups: passengers, drivers/conductors, dispatchers, and administrators. It is designed for the Sapang Palay Tungko Grotto Transport Services Cooperative (SPTGTSC) located in Bulacan.

The following are the benefits and general objectives of the Smart Jeepney Monitoring System for the various stakeholders**:**

1. **Passengers -** Benefits from reduced waiting time, since the system provides visibility on vehicles and seat occupancy. This helps make their daily commute more predictable.
2. **Dispatchers -** More accurate tools for monitoring and managing trips, improving dispatch efficiency and reducing errors in scheduling. This leads to smoother fleet coordination.
3. **Drivers/Conductors -** More organized workflow, as trip details and updates are streamlined. This minimizes confusion in operations and allows them to focus on providing better service.
4. **Administrators -** Benefit from transparent access to digitalized reports on trips and overall operations. This improves decision-making and helps ensure resources are effectively used.
5. **SPTGTSC -** Gains business benefits by standing out from competitors using modern technology and attract more passengers with improved services. On the technical side, it supports the digitalization which fosters innovative development by integrating new tools.

While the system shares similarities with existing transportation platforms such as ride-hailing apps (e.g., Grab, Angkas, Joyride, Move It, etc.) or fleet management solutions used in large-scale programs like the LTFRB’s PUV Modernization initiatives, it is narrower in scope as it is tailored specifically for modernized jeepneys within a cooperative setting. Its primary focus is on improving monitoring and utilization rather than handling complex features like fare integration, automated scheduling, or predictive analytics. However, the system is designed with the potential for future expansion, including route optimization, digital fare collection, and AI-driven analytics, which can further enhance its role in modern public transport management.

1.3 Intended Audience and Document Overview

This document is intended for the following groups of readers: the Sapang Palay Tungko Grotto Transport Services Cooperative (SPTGTSC) as the project client, the OUTCODE development team, future maintainers, and the academic panel.

1. **Introduction –** Gives a general picture of the system, what it is made for, what it can do, the limits of how it will work, and the assumptions made in planning the system.
2. **Overall Description –** Gives a general picture of the system, what it is made for, what it can do, the limits of how it will work, and the assumptions made in planning the system.
3. **Specific Requirements –** Lists the details of what the system should do, how users will use it, and the main functions it needs to perform.
4. **Other Non-Functional Requirements –** Describes qualities the system should have, such as how fast it should respond, how secure it should be, and how reliable it is for daily use.
5. **Appendices -** Contains supporting materials such as the Data Dictionary/ERD, the Group Log of activities and meetings, and the Test Plan/Test Cases.

READER TYPE	**RECOMMENDED SEQUENCE**
Client (SPTGTSC)	S1 → S2 → S4 → Appendices
Development Team	S2 → S3 → S4 → Appendix A
Future Maintainers	S1 → S2 → S3 → S4 → Appendix A
Academic Panel	S1 → S2 → S3 → S4 → Appendices

S1	Introduction
S2	Overall Description
S3	Functional Req
S4	Non-Functional Req

1.4 Definitions, Acronyms and Abbreviations

Term	Explanations
GPS	(Global Positioning System) Real-time location tracking of vehicles [1].
UI	(User Interface) What users directly interact with in the system [2].
RPI4	(Raspberry PI 4) A single-board computer used for embedded systems [3].
UART	(Universal Asynchronous Receiver-Transmitter) A hardware communication protocol for serial data transmission between devices [4]
NoSQL	(Not Only SQL) A non-relational database system designed for flexible storage of structured and unstructured data [5].
USB	(Universal Serial Bus) An industry standard for cables and connectors that enable data transfer and power supply between devices [6].
CSI	(Camera Serial Interface) A standard interface defined by MIPI for connecting cameras to processors [7].
OS	(Operating System) – Software that manages computer hardware and provides services for application programs [8].
MIPI	(Mobile Industry Processor Interface) – A global standard for mobile and embedded device interfaces, such as camera and display connections [0].
JPEG	(Joint Photographic Experts Group) – A commonly used image compression standard that reduces file size while maintaining quality [10].
NMEA Protocol	(National Marine Electronics Association Protocol) – A standard protocol used by GPS devices to transmit data like location, time, and speed [11].
BCrypt	A password-hashing function designed for secure storage of passwords by applying multiple rounds of encryption [12].

1.5 Document Conventions

1. Formatting Conventions

- Body Text – Arial, Size 11, Justified, Double-spacing
- Section Titles – Arial, Size 14, Bold, numbered (e.g., 1, 2, 3), Left-aligned
- Subsection Titles – Arial, Size 12, Bold, Hierarchical Numbering (e.g., 2.1, 2.1.1)

2. Naming Conventions

- Functional requirements - labeled as FR-1, FR-2, FR-3, and so on.
- Non-functional requirements - labeled as NFR-1, NFR-2, NFR-3, and so on.
- Performance requirements - labeled as PR-1, PR-2, etc.

1.6 References and Acknowledgments

[1] "Global Positioning System (GPS)," GPS.gov. https://www.gps.gov/

[2] "User Interface (UI)," Techopedia. https://www.techopedia.com/definition/704/user-interface-ui

[3] "Raspberry Pi 4 Model B," https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[4] National Instruments. https://www.ni.com/en-us/innovations/uart.html

[5] MongoDB. https://www.mongodb.com/nosql-explained

[6] "Universal Serial Bus (USB)," USB.org. https://www.usb.org/

[7] MIPI Alliance. https://www.mipi.org/specifications/csi-2

[8] Britannica. https://www.britannica.com/technology/operating-system

[9] MIPI Alliance. https://www.mipi.org/

[10] Joint Photographic Experts Group. https://jpeg.org

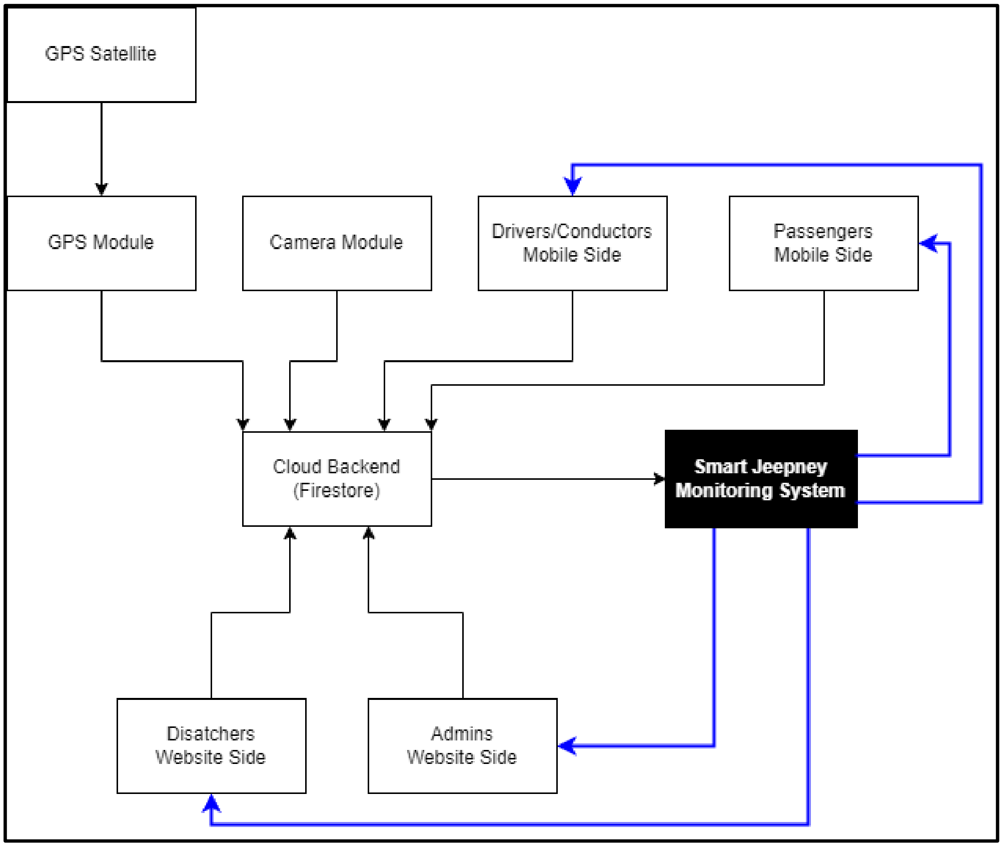
[11] National Marine Electronics Association. https://www.nmea.org/

[12] "npm: bcrypt," *Npm*. https://www.npmjs.com/package/bcrypt

2 Overall Description

2.1 Product Overview

The Smart Jeepney Monitoring System is a technology solution designed to improve both the daily operations of the SPTGTSC and the commuting experience of its passengers. This system is not part of any previous system or direct upgrade, but a completely new setup that replaces the cooperative's current manual. The system addresses common issues such as lack of transparency in unit availability, inefficiencies in dispatching, and limited tools for administrative monitoring of the operation's data. By integrating digital monitoring and reporting into day-to-day transport operations, the system provides SPTGTSC with a tailored solution rather than serving as a replacement or add-on to an existing platform.



The diagram above illustrates how the system operates and how its components interact. Each jeepney is equipped with a GPS and camera module that sends real-time location and occupancy data to the central system. The backend processes this information and distributes it to users through their respective interfaces. The mobile app allows passengers to locate nearby jeepneys and check seat availability, while drivers and dispatchers can access relevant trip data. The website enables dispatchers to manage and coordinate jeepney operations and allows administrators to monitor fleet activity, track usage, and make informed operational decisions. Overall, the system streamlines daily transport operations, enhances commuter service, and helps the cooperative manage its fleet more efficiently.

2.2 Product Functionality

The Smart Jeepney Monitoring Platform aims to meet the needs of both passengers and transport operators by offering real-time visibility and data-based insights into modernized jeepney operations. This system brings together GPS tracking, occupancy monitoring, and performance logging to improve transport services and enhance commuter experience. The main features include:

1. **Real-Time Tracking -** Display real-time GPS locations of all active jeepney units to the users via mobile app and website.
2. **Passenger Visibility -** Allow passengers to share and view their location for better trip coordination.
3. **Seat Availability Monitoring -** Show current passenger occupancy status inside each jeepney using camera-based monitoring.
4. **Estimated Arrival Time -** Provide estimated arrival times of nearby jeepneys based on current location and route data.

5. **Fare Cost Estimate -** Each route or stop has a predetermined fare based on the distance traveled and standard rates set by the cooperative.
6. **Passenger Visibility for Drivers/Conductors -** Enable drivers and conductors to see incoming passengers wanting to board for smoother operations.
7. **Fleet Management -** Allow dispatchers to monitor jeepney locations through a dashboard for better dispatching and route control.
8. **Data Reporting -** Provide administrators with reports on fleet usage and operations.
9. **Accounts and Jeepney Units Management -** Allows administrators to manage all system accounts and registered jeepney units.

The major functions listed above provide a high-level summary of what the Smart Jeepney Monitoring System is designed to do. They highlight the core capabilities for each user type (passengers, drivers/conductors, dispatchers, and administrators) without going into technical or implementation details. The specific and detailed functional requirements, including precise workflows, inputs, outputs, and system behavior, are provided in Section 3 of this document, where each function is described in full to guide development and ensure all user needs are met.

2.3 Design and Implementation Constraints

This section describes the constraints and limitations that affect the design and implementation of the Smart Jeepney Monitoring System. It covers hardware limitations, the technologies and tools used, database selection, and security considerations, all which guide development decisions.

Hardware Limitations

1. **Raspberry Pi 4 (RPI4) -** Serves as the primary edge device. It has limited CPU and memory resources compared to traditional servers, therefore:
    - Image processing must be minimal (e.g., capture and upload only).
    - Background services must be lightweight to ensure smooth performance.
2. **RPI Camera Module and GPS Module** - Captures images and update unit locations respectively, at set intervals (every 10 seconds) to minimize processing load and network.
3. **The GPS Module *(NEO-6M)* -** Must interface via serial/UART and provide accurate location data for the cloud backend. This allows proper data reflection on the users' end.

Technologies, Tools, and Databases

1. **Programming Languages**
    - **Python** - Used to develop scripts that run on the Raspberry Pi, handling tasks such as reading data from the modules, and sending updates to the cloud server.
    - **Java** - Used for developing the Android mobile application, which provides the user interface for passengers, drivers/conductors, and dispatchers.
2. **Database**
    - **Cloud Firestore (NoSQL)** - Used for storing structured and semi-structured data such as user accounts, jeepney units, and trip logs.
3. **Deployment**
    - **Android Studio** - Used for Android mobile application
    - **Visual Studio Code** - Used for writing Raspberry Piand other supporting code.

Security Considerations

1. Data transmitted between the Raspberry Pi and Firebase will be encrypted via HTTPS.

2. Sensitive data such as account password will be secured using the BCrypt [1] hashing algorithm. This prevents passwords from being stored in plaintext, reducing the risk of credential leaks in case of a data breach.

2.4 Assumptions and Dependencies

This section outlines the key assumptions and dependencies that could significantly impact the requirements, design, and implementation of the system. These assumptions are not guaranteed, and any changes may require revision of project plans.

Assumptions

1. **Hardware Availability** - It is assumed that all participating jeepney units will be equipped with functioning GPS and CCTV modules before full deployment. The system assumes continued operation of these devices throughout tracking periods.

2. **Data Connectivity** - The project assumes that there is stable internet coverage throughout the jeepney routes to enable real-time data transmission between the devices, cloud servers, and mobile applications.

3. **User Adoption and Access** - Commuters are assumed to have compatible smartphones and sufficient internet data to use the application. They are also assumed to be open to using the new system once launched.

4. **Fixed Project Scope and Budget** - The project assumes that no additional scope elements will be added beyond what is documented and that the approved ₱520,924.00 budget will be sufficient for full development and deployment.

Dependencies

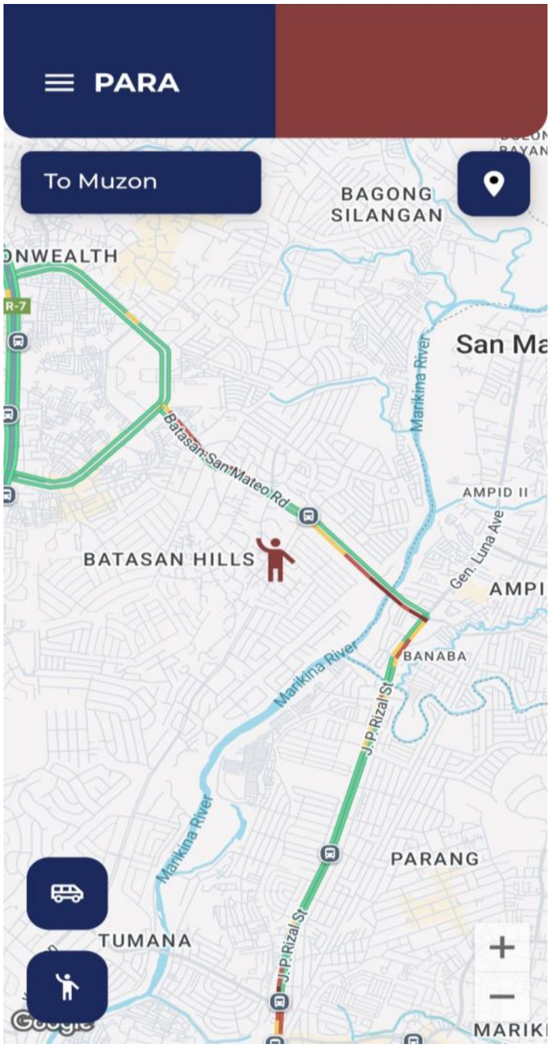
1. **Third-Party Components** - The project depends on the successful integration of off-the-shelf hardware components (e.g., Raspberry Pi, GPS modules, camera modules) and open-source software libraries for implementation.
2. **Cloud Services** - The system backend relies on cloud infrastructure to store and manage data. Its reliability and uptime directly affect the system's functionality.
3. **Jeepney Unit Access** - Timely access to jeepney units for hardware installation, calibration, and field testing is critical and is dependent on coordination with SPTGTSC's operational schedules.

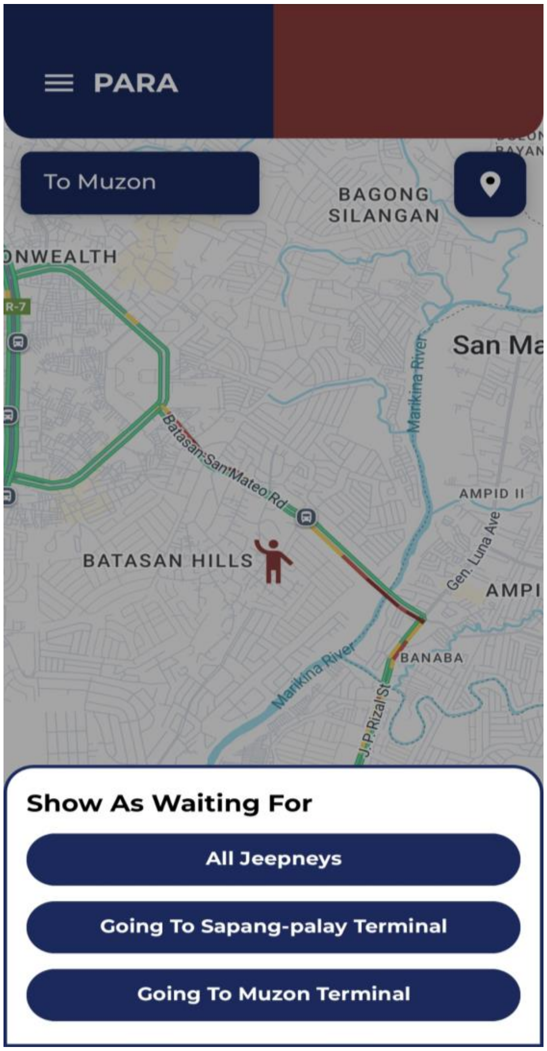
The outlined assumptions and dependencies serve as the foundation for the project's development and deployment. While they provide a framework for planning and execution, any deviations such as hardware malfunctions, unstable connectivity, or limited user adoption may require adjustments to the project scope, schedule, or resources. Likewise, dependencies on third-party components, cloud services, and jeepney access emphasize the importance of external factors that are beyond direct project control. Recognizing these conditions early ensures that risks can be anticipated and mitigated to support the successful implementation of the system.

3 Specific Requirements

3.1 External Interface Requirements

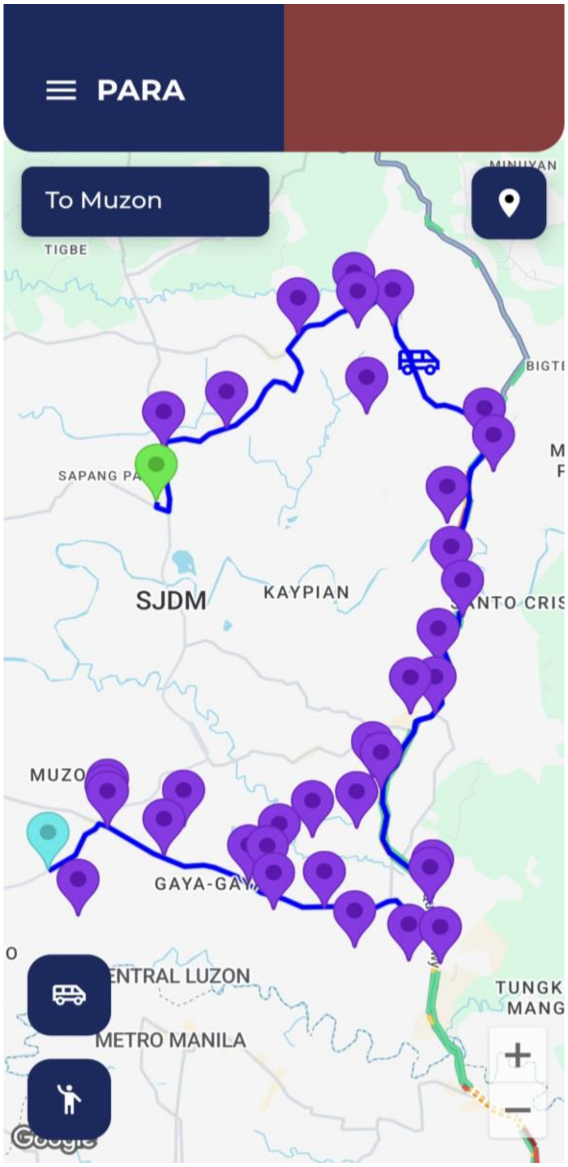
3.1.1 User Interfaces

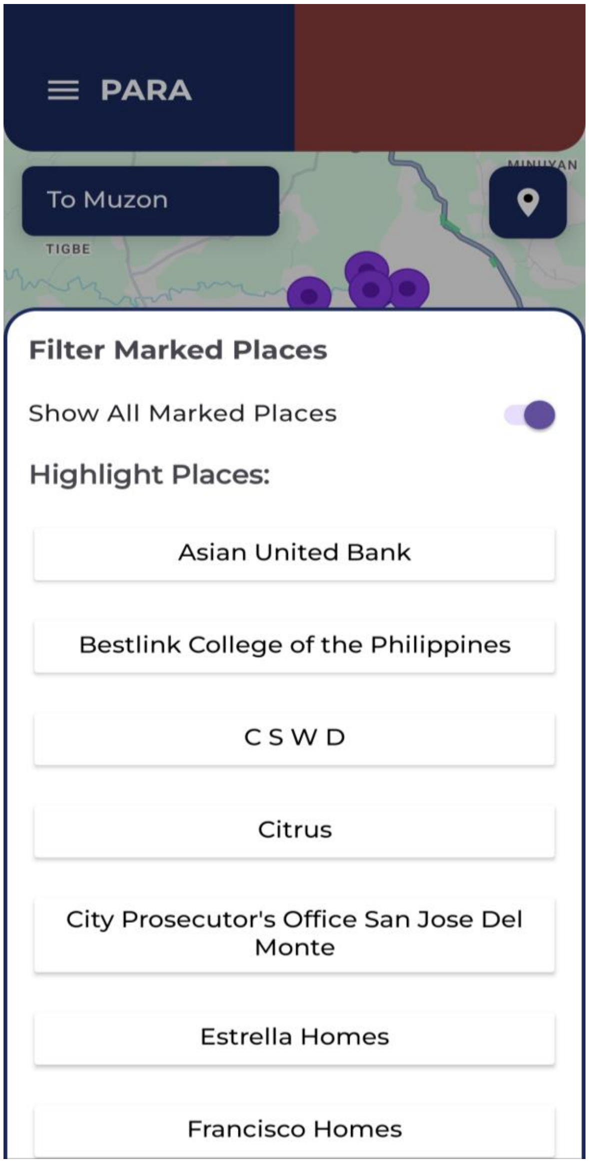




- This is the Commuter Map page; This Is what the user would see after the user have logged in. The user would see their real time location with the use of the users' phone GPS being represented by a red person marker.

- This is what the user would see after they press the button with the text "Show as Waiting", They would see three choices being "All Jeepneys", "Going To Sapang-palay Terminal", "Going To Muzon Terminal".





Filter Marked Places

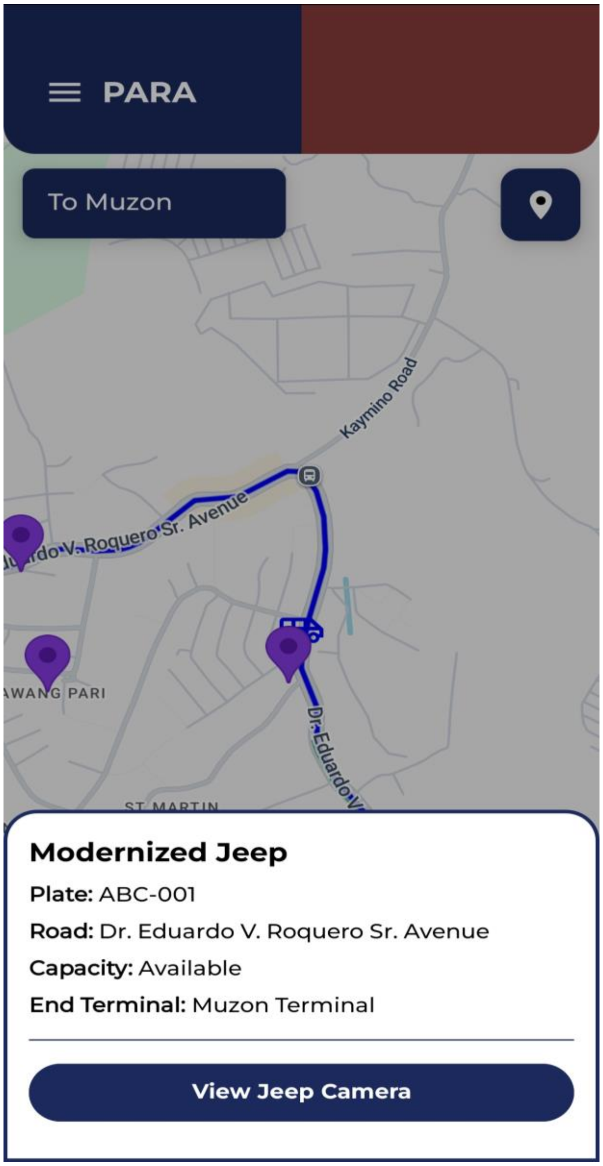
Show All Marked Places ●

Highlight Places:

- Asian United Bank
- Bestlink College of the Philippines
- C S W D
- Citrus
- City Prosecutor's Office San Jose Del Monte
- Estrella Homes
- Francisco Homes

- This the route line for jeepneys that could be changed by clicking the top right button, the user would see many different colored markers, Purple representing the places the jeepneys would go too, Green representing the terminal the jeepney started from, Cyan representing the end terminal and a Jeep Marker representing the jeepneys.

- This is what the user would see if they press the top left button with the icon of a marker, it's a filer; The user could disable all the marked places (Purple Markers) or highlight a specific place. A highlighted place would turn into pink.



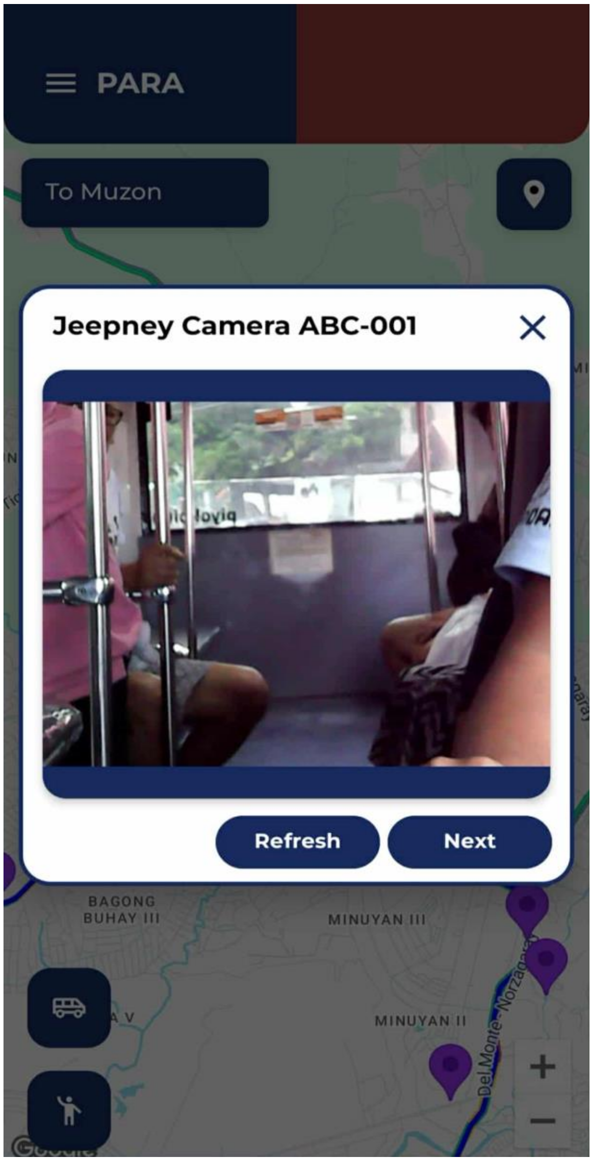
Modernized Jeep

Plate: ABC-001

Road: Dr. Eduardo V. Roquero Sr. Avenue

Capacity: Available

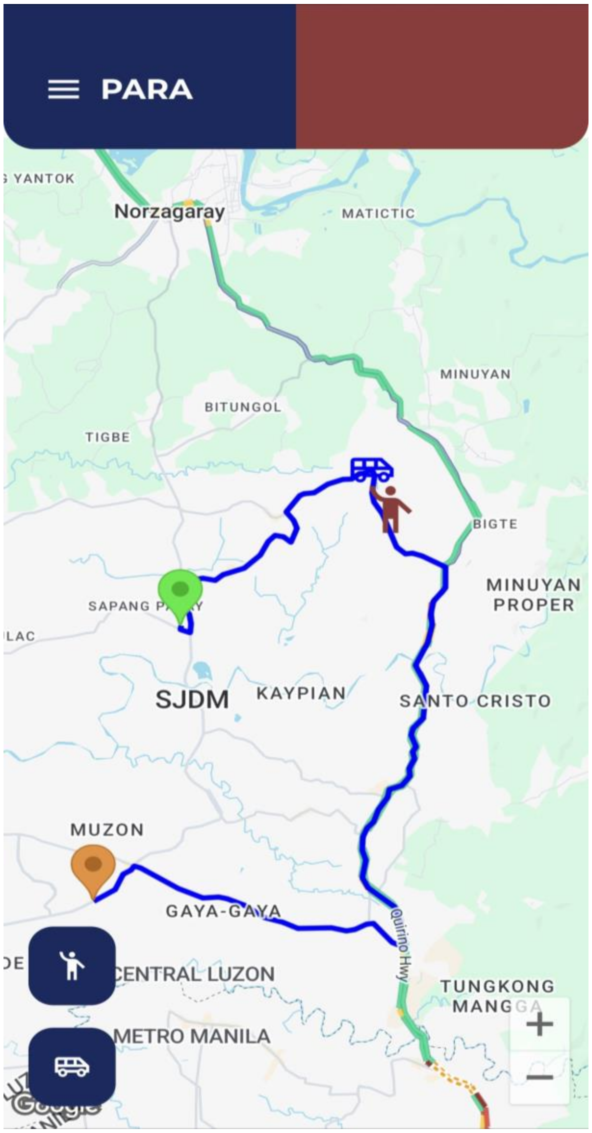
End Terminal: Muzon Terminal

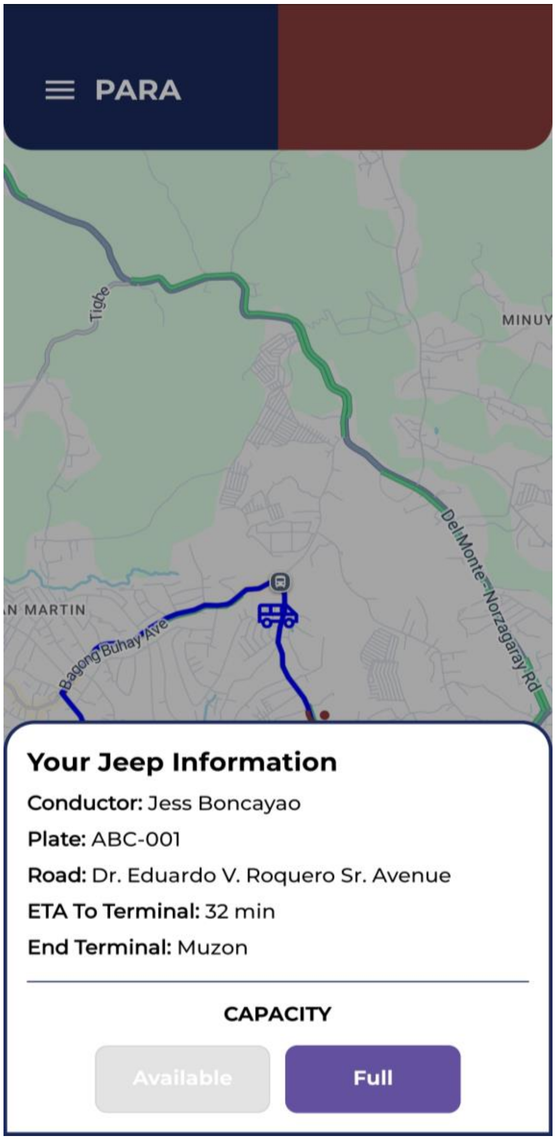


Jeepney Camera ABC-001

- This what would happen if the user pressed the jeepney marker. The user would see the jeepneys information the jeepneys plate number, the road the jeepney are in, the capacity being available or full, the end terminal, and a button to view the two cameras.

- This is what the user would see after the press the button called “View Jeep Camera” in the jeepney information. There would be two buttons to being a refresh button and a next button.





Your Jeep Information

Conductor: Jess Boncayao

Plate: ABC-001

Road: Dr. Eduardo V. Roquero Sr. Avenue

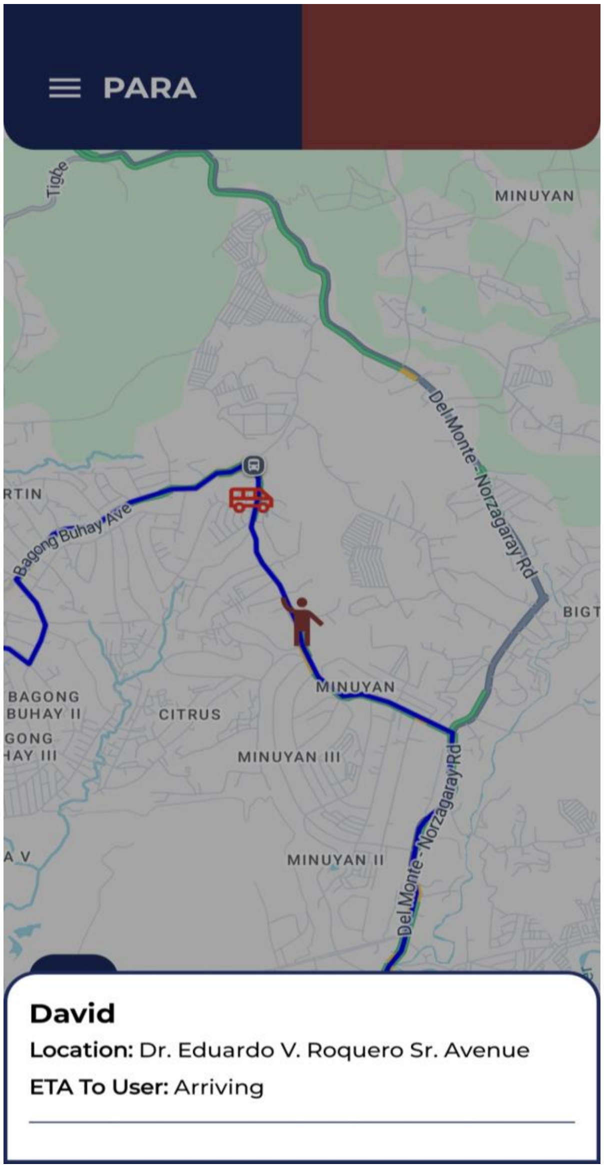
ETA To Terminal: 32 min

End Terminal: Muzon

CAPACITY

- This is Driver/Conductor map page. They would see two markers being Green the terminal the user came from and the end terminal. There also two buttons on the bottom left, the button with the icon of a person would cycle to every user that is waiting and the button with the icon a jeep would go to their jeepney.

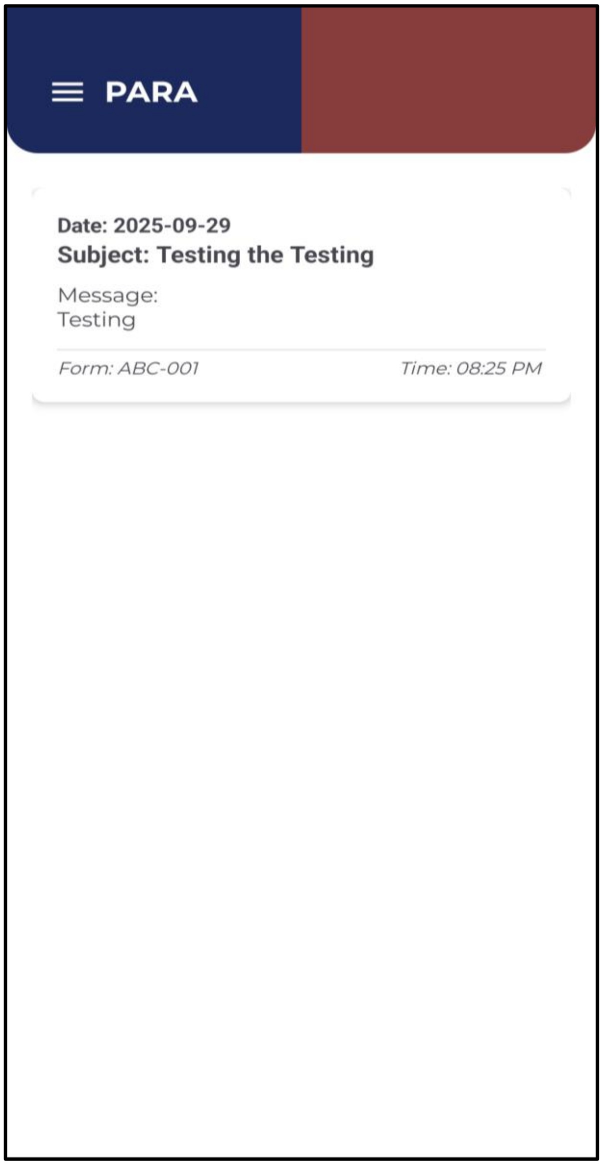
- This is what would happen when you click the Jeep Marker you would see your jeepneys information such as the Conductor, the Plate Number, the Road you are in, ETA to terminal, the terminal, and the button to change the Capacity its defaulted to Available (Blue Jeep Marker) meaning there is still space on the jeepney and Full (Red Jeep Marker) if there's no more space.



David

Location: Dr. Eduardo V. Roquero Sr. Avenue

ETA To User: Arriving



Date: 2025-09-29

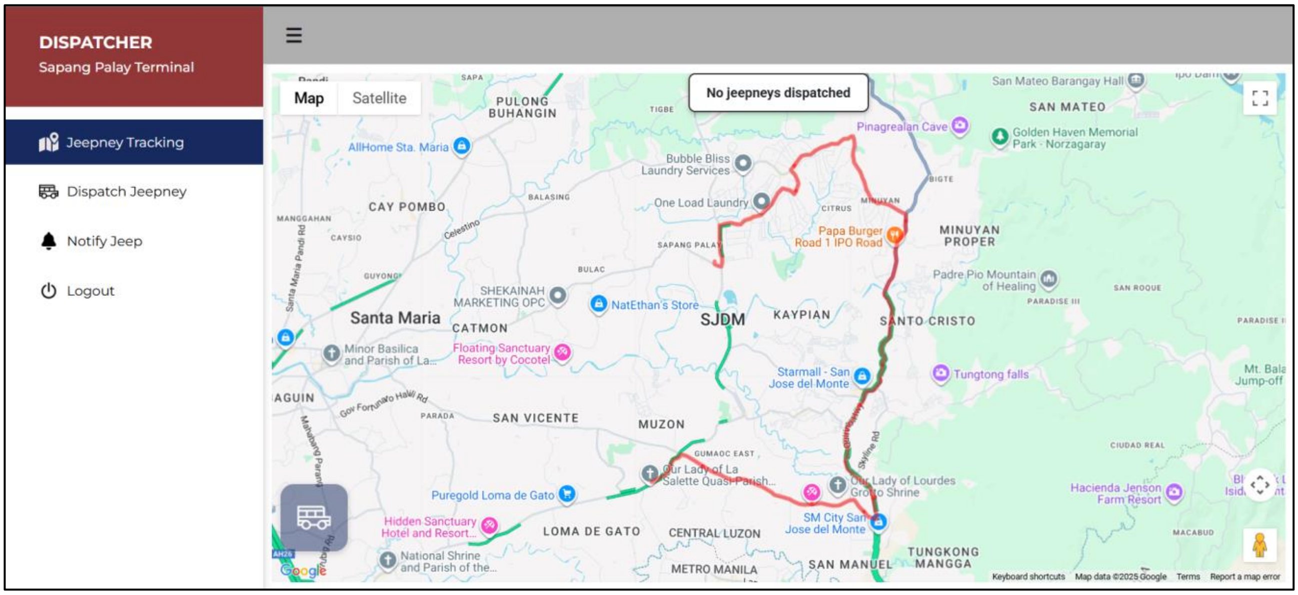
Subject: Testing the Testing

Message:
Testing

Form: ABC-001 *Time: 08:25 PM*

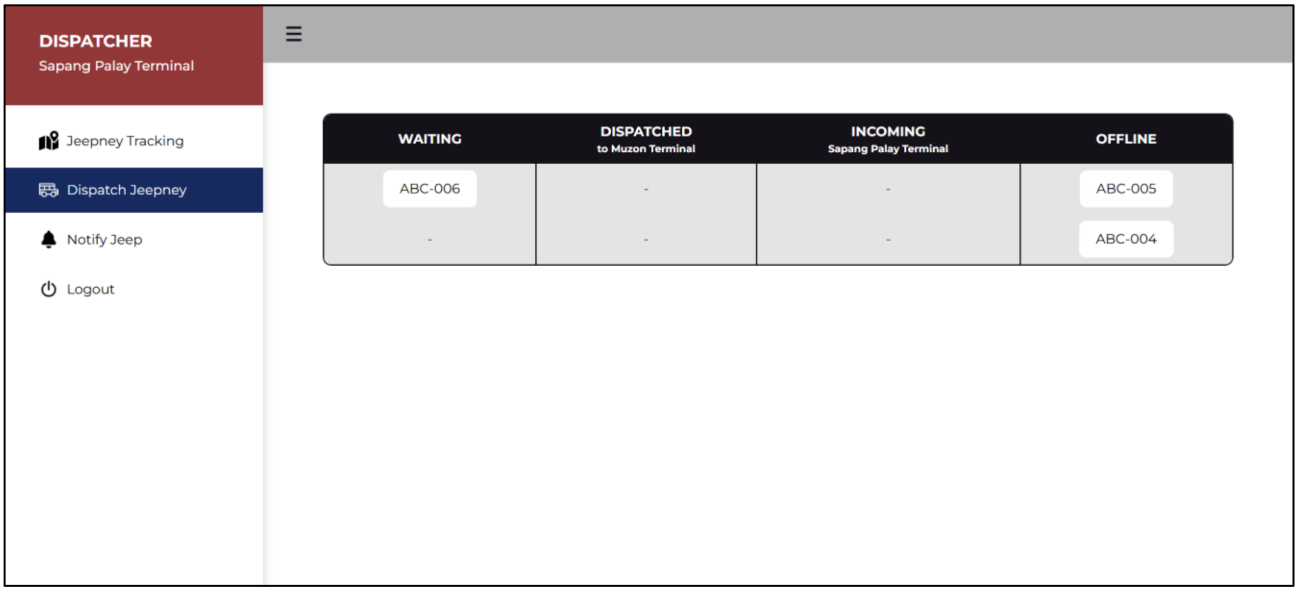
- This is the commuter's information that are waiting, it would show the location and the ETA to the user.

- This is the Announcements page from the Driver/Conductor here they would see the Date, Subject, Message, Form if its either for all jeepneys or just for their jeep, and the time.



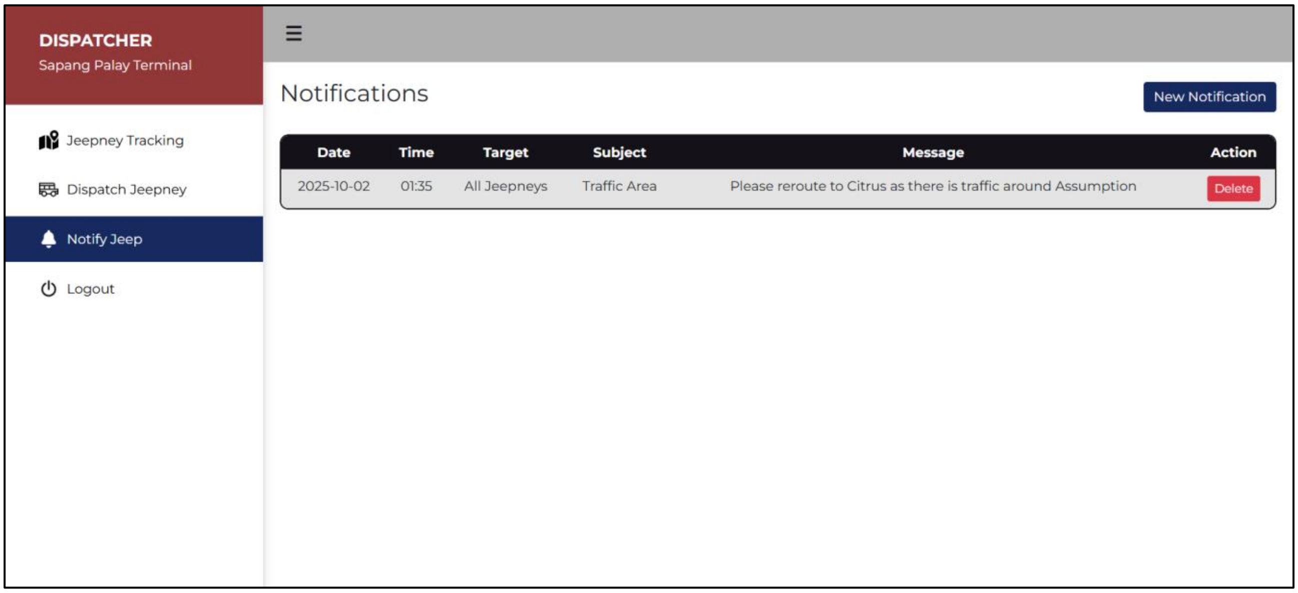
No jeepneys dispatched

- The jeepney tracking module serves as a tracking page for all dispatched jeepneys by SPTGTSC, enabling them to view the current location of a specific jeepney on the company's set routes.



WAITING	DISPATCHED to Muzon Terminal	INCOMING Sapang Palay Terminal	OFFLINE
ABC-006	-	-	ABC-005
-	-	-	ABC-004

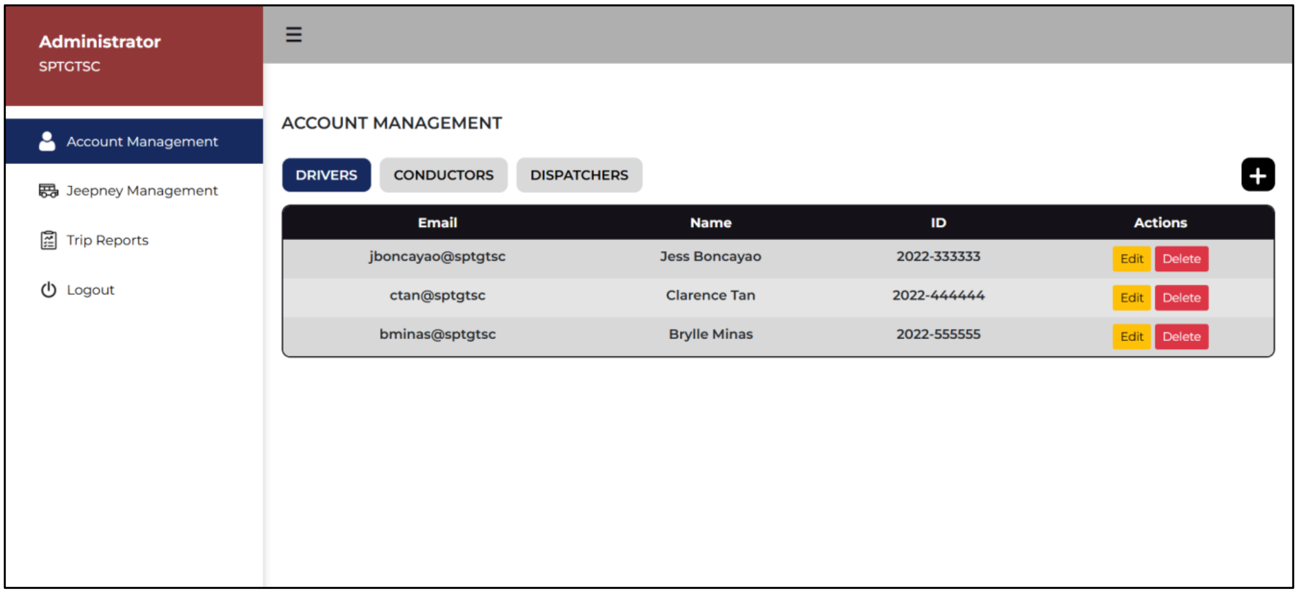
- The dispatch jeepney module displays the different jeepneys along with their corresponding status (waiting, dispatched, incoming, offline). This module is where the dispatcher can assign drivers and conductors on a waiting jeepney, as well as stop or abort dispatched units.



Notifications

Date	Time	Target	Subject	Message	Action
2025-10-02	01:35	All Jeepneys	Traffic Area	Please reroute to Citrus as there is traffic around Assumption	Delete

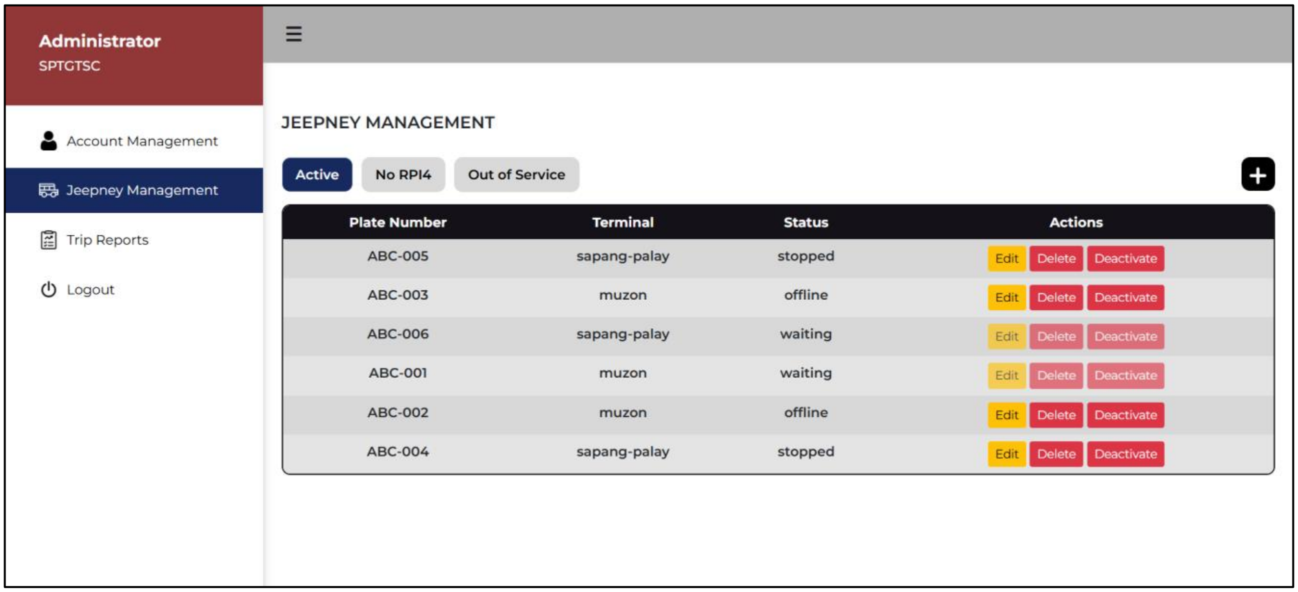
- The Notify Jeep module is designed for dispatchers to alert all or specified jeepneys on a certain situation such as sudden traffic or roadblocks which require immediate action (e.g. rerouting). Any notifications made from this module would reflect on the mobile app of drivers and conductors that are currently assigned on a dispatched jeepney.



ACCOUNT MANAGEMENT

Email	Name	ID	Actions
jboncayao@sptgtsc	Jess Boncayao	2022-333333	Edit Delete
ctan@sptgtsc	Clarence Tan	2022-444444	Edit Delete
bminas@sptgtsc	Brylle Minas	2022-555555	Edit Delete

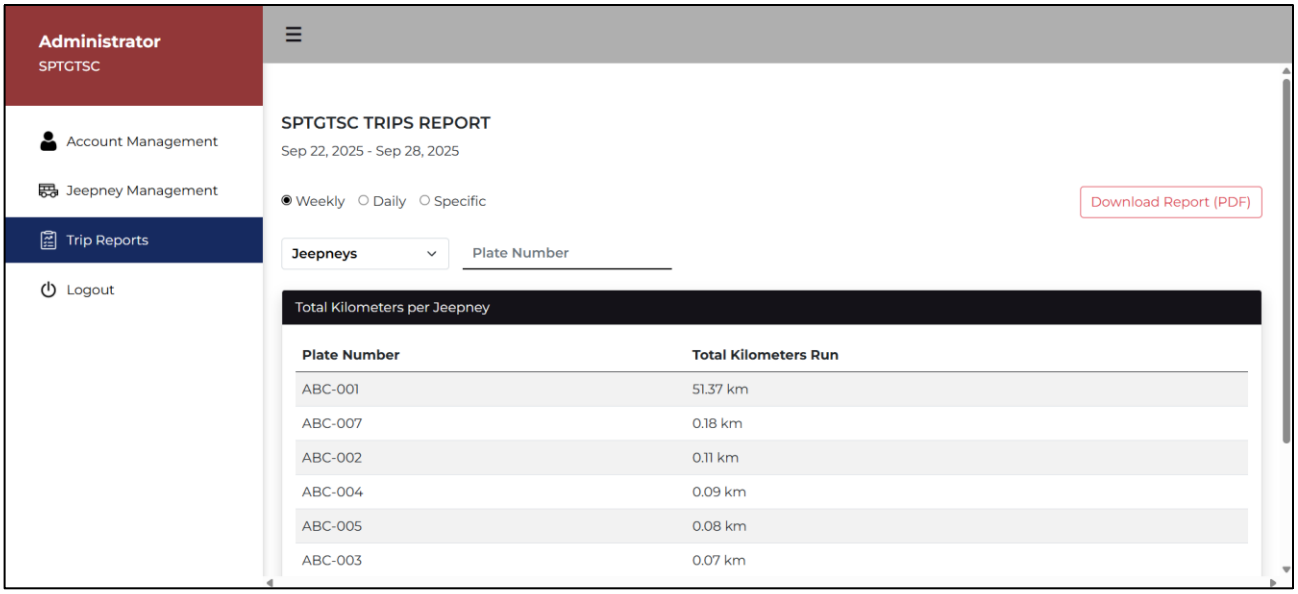
- The Account Management module handles the adding, editing, and deleting of drivers, conductors, and dispatchers.



JEEPNEY MANAGEMENT

Plate Number	Terminal	Status	Actions
ABC-005	sapang-palay	stopped	Edit Delete Deactivate
ABC-003	muzon	offline	Edit Delete Deactivate
ABC-006	sapang-palay	waiting	Edit Delete Deactivate
ABC-001	muzon	waiting	Edit Delete Deactivate
ABC-002	muzon	offline	Edit Delete Deactivate
ABC-004	sapang-palay	stopped	Edit Delete Deactivate

- The Jeepney Management module adds, edits, and deletes jeepney units. Additionally, it also enables jeepney deactivation if some jeepneys need to go under maintenance. Lastly, RPI4 devices can be attached/detached on jeepney units.



SPTGTSC TRIPS REPORT

Sep 22, 2025 - Sep 28, 2025

● Weekly ○ Daily ○ Specific

Jeepneys | Plate Number

Total Kilometers per Jeepney

Plate Number	Total Kilometers Run
ABC-001	51.37 km
ABC-007	0.18 km
ABC-002	0.11 km
ABC-004	0.09 km
ABC-005	0.08 km
ABC-003	0.07 km

- The Trips Report module allows the administrator to check for reports of drivers, conductors, and drivers on the past week. It also enables the administrator to conveniently access specific date ranges on a specific entity (jeepney, driver, conductor) and view their specific reports. They also have the option to download the reports of their query.

3.1.2 Hardware Interfaces

1. Raspberry Pi 4

- **Type:** Main computing unit / microcontroller
- **Physical Interface:** GPIO pins, USB ports, CSI camera port
- **Logical Interface:**
  - Acts as the host controller for all other connected modules.
  - Runs Raspberry Pi OS 64-bit (Debian Bookworm).
  - Executes Python and Bash scripts to handle data acquisition, processing, and data transmission from database and hardware modules.
- **Data & Control:**
  - Controls camera and GPS modules via attached ports.
  - Sends data to external servers (Firebase, GoDaddy webserver).

2. Raspberry Pi Camera Module

- **Type:** Image sensor
- **Physical Interface:** 2-Lane MIPI (Mobile Industry Processor Interface) CSI (Camera Serial Interface) Camera Port
- **Logical Interface:**
  - Controlled via raspistill for image capture.
  - Provides JPEG image output to be stored in the GoDaddy webserver.

- **Data & Control:**
    - Command to capture image.
    - Output is a file (e.g., image.jpg) that is processed and sent to the webserver.

3. NEO-6M GPS Module

- **Type:** Location sensor
- **Physical Interface:** UART (via GPIO)
- **Logical Interface:**
    - Uses NMEA protocol over serial connection.
    - Read interface via /dev/ttyAMA0 using the pyserial Python library.
- **Data & Control:**
    - Provides real-time location data (latitude, longitude, distance covered).
    - Software reads location periodically and logs or sends to Firebase.

4. Power Supply System

- **Type:** Jeepney-provided power via onboard USB port.
- **Physical Interface:** Connected to the Raspberry Pi 4 via USB-C power input.

3.1.3 Software Interfaces

The Smart Jeepney Monitoring System interfaces with the commuter mobile application and an admin dashboard to enable real-time data exchange and monitoring. These software interfaces are described as follows:

1. **Mobile Application (Passenger Side)** - The mobile app connects to the cloud backend to fetch live GPS data and video/image feeds from each jeepney unit. It displays:
    - Image capture of the jeepney seats using a camera.
    - Showing nearby jeepneys and estimated arrival times.
    - Allows passengers to make their location visible to drivers/conductors.
2. **Mobile Application (Driver/Conductor Side)** - Provides drivers and conductors with tools to assist in managing trips and passengers. Features include:
    - Viewing passenger location visibility (commuters who choose to share their location).
    - Checking route assignments and active trip schedules.
    - Monitoring occupancy levels in real time to prevent overloading.
3. **Website Dashboard (Dispatcher Side)** - Designed for dispatchers to manage daily operations efficiently. Features include:
    - Monitoring real-time location and status of all jeepneys.
    - Assign Jeepney units with Drivers and Conductors.
    - View all Jeepney units with status Waiting, Inc

4. **Website Dashboard (Administrator Side)** - This web-based or application-based interface allows administrators to:
    - Monitor location and status of all jeepneys.
    - View route utilization and number of trips per unit.
    - Generate analytics reports such as daily mileage and route-based earnings.
    - Identify idle or underperforming units for dispatch optimization.

5. **Website Dashboard (Administrator Side)** - This web-based or application-based interface allows administrators to:
    - Monitor location and status of all jeepneys.
    - View route utilization and number of trips per unit.
    - Generate analytics reports such as daily mileage and route-based earnings.
    - Identify idle or underperforming units for dispatch optimization.

Together, these interfaces form the core interaction of the Smart Jeepney Monitoring System, ensuring that each user type: passengers, drivers/conductors, dispatchers, and administrators has access to the specific tools and data they need. By tailoring features to each role, the system promotes efficiency, improves commuter experience, streamlines trip management, and supports data-driven decision-making for the cooperative.

3.2 Functional Requirements

This section defines the specific capabilities and behaviors that the system must perform to fulfill its intended purpose. The functional requirements are described as follows:

3.2.1 **F1:** The system shall display the real-time GPS location of all active modernized jeepney units to commuters via the mobile application.

3.2.2 **F2**: The system shall transmit GPS data from each modernized jeepney to the cloud backend at regular intervals.

3.2.3 **F3**: The system shall compute and display estimated arrival times of modernized jeepneys based on current GPS location and route data**.**

3.2.4 **F4**: The system shall determine and display the current passenger occupancy status of each modernized jeepney using camera-based monitoring.

3.2.5 **F5**: The system shall allow commuters to share their current location so that drivers and conductors can identify commuters who are ready to board.

3.2.6 **F6**: The system shall allow drivers and conductors to view all commuters within the jeepney's route who are waiting to board.

3.2.7 **F7:** The system shall allow dispatchers to assign a jeepney unit for deployment to a terminal.

3.2.8 **F8:** The system shall allow dispatchers to assign drivers and conductors to specific jeepney units prior to dispatch.

3.2.9 **F9:** The system shall record and store the daily distance traveled by each modernized jeepney unit.

3.2.10 **F11:** The system shall track the performance of drivers and conductors, including frequency of trips completed.

3.2.11 **F12:** The system shall allow administrators to add, edit, and delete user accounts for drivers, conductors, and dispatchers.

3.2.12 **F13:** The system shall allow administrators to add, edit, and delete jeepney units.

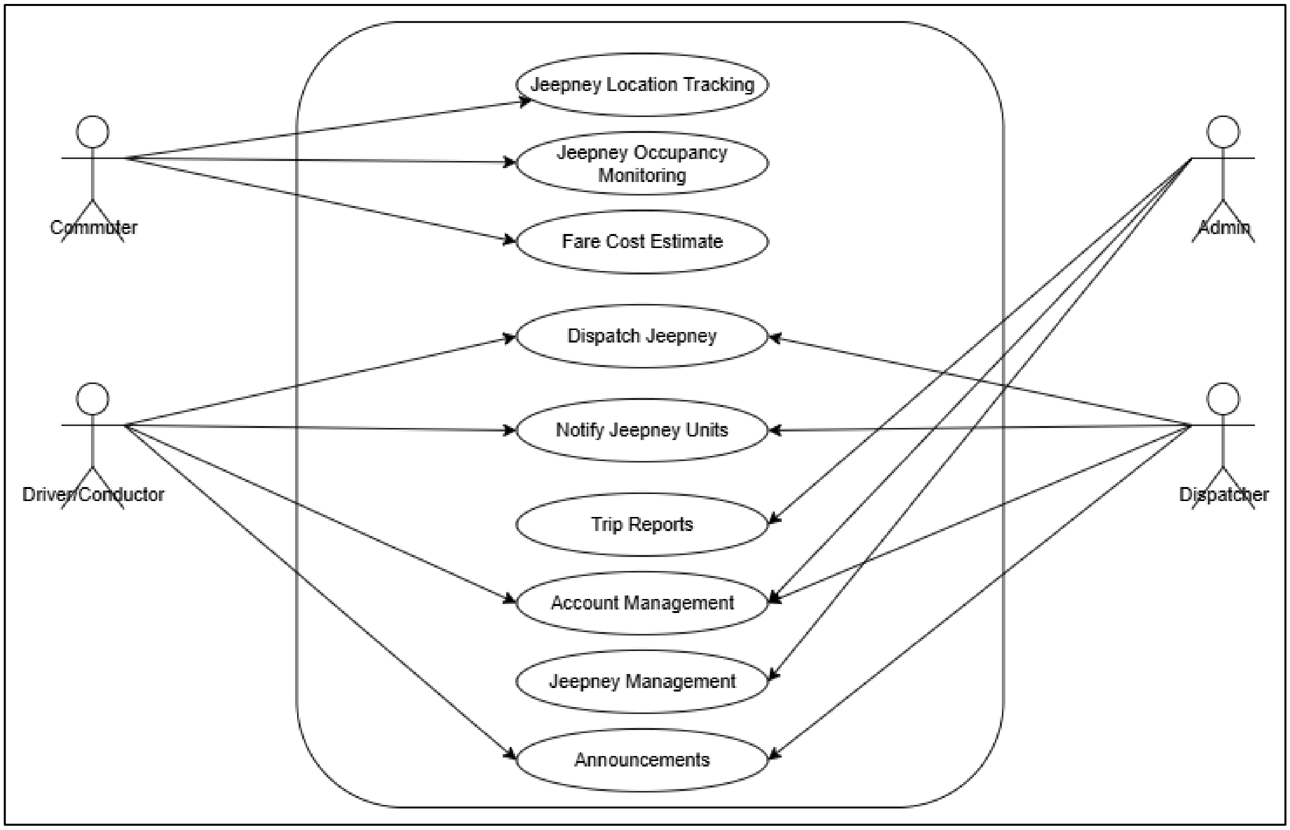
3.2.13 **F14:** The system shall allow administrators to deactivate jeepneys that are under maintenance or unavailable.

3.2.14 **F15:** The system shall allow administrators to detach and reassign the RPi4 device to another jeepney unit when necessary.

3.2.15 **F16:** The system shall display estimated fare costs for commuters based on the fare matrix provided by the cooperative.

3.3 Use Case Model

In this Use Case Model, the "Smart Jeepney Monitoring: A GPS-Integrated Tracking and Utilization Platform for SPTGTSC" system's functional connection to users is depicted. To manage jeepney operations, access data, and maximize commuter convenience and vehicle efficiency, commuters, drivers, and administrators will be the main users of this system. A detailed representation use case is broken down more thoroughly below.



3.3.1 Use Case #1 (Jeepney Location Tracking – U1)

<table>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Jeepney Location Tracking</td></tr>
<tr><td><b>Author:</b></td><td colspan="3">OutCode</td></tr>
<tr><td><b>Purpose:</b></td><td colspan="3">To allow commuters to view the real-time location and estimated arrival time of available jeepney units via the mobile application.</td></tr>
<tr><td><b>Requirements Traceability:</b></td><td colspan="3">F1, F2, F3, F5</td></tr>
<tr><td><b>Priority:</b></td><td colspan="3">High</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">• GPS modules are properly installed and activated on each jeepney.
• The commuter has internet access and a mobile device with the application installed.</td></tr>
<tr><td><b>Post Conditions:</b></td><td colspan="3">• The commuter can see the location of nearby jeepney units in real-time.
• Estimated time of arrival (ETA) is provided.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">• Commuter
• GPS Module
• Mobile Application</td></tr>
<tr><td rowspan="2"><b>Flow of Events:</b></td><td><b>Basic Flow</b></td><td><b>Alternative Flow</b></td><td><b>Exceptions</b></td></tr>
<tr><td>• Commuter opens the mobile application.
• The system fetches the GPS data from the cloud database.
• Jeepney locations are displayed on the map.
• Commuter selects a unit to see ETA.</td><td>• If the commuter’s location is turned off, the app asks for location permission and centers on a default area.</td><td>• GPS data is unavailable: The system notifies the commuter that data cannot be retrieved.</td></tr>
<tr><td><b>Notes/Issues:</b></td><td colspan="3">• Network reliability might affect real-time updates.
• GPS drift may cause slight location inaccuracies.</td></tr>
</table>

3.3.2 Use Case #2 (Jeepney Occupancy Monitoring – U2)

Use Case Name:	Jeepney Occupancy Monitoring		
Author:	OutCode		
Purpose:	To allow commuters to view real-time jeepney seat availability using camera feeds.		
Requirements Traceability:	F4, F6		
Priority:	High		
Preconditions:	• Jeepney must have a functioning camera module installed. • The commuter has internet access.		
Post Conditions:	• Commuters can see seat availability before boarding. • Admins can monitor jeepney occupancy remotely.		
Actors:	• Commuter • Camera Module • Backend System		
Flow of Events:	**Basic Flow**	**Alternative Flow**	**Exceptions**
	• Commuter opens the app. • Selects a jeepney unit. • System shows a live picture from the camera. • Commuter checks seat availability.	If camera feed is delayed, system shows the last available image snapshot.	Poor lighting or obstructed view.
Notes/Issues:	Explore fallback options when the camera is unavailable.		

3.3.3 Use Case #3 (Fare Cost Estimate – U3)

<table>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Fare Cost Estimate</td></tr>
<tr><td><b>Author:</b></td><td colspan="3">OutCode</td></tr>
<tr><td><b>Purpose:</b></td><td colspan="3">To allow commuters to view the estimated fare for their trip based on standard rates set by the cooperative.</td></tr>
<tr><td><b>Requirements Traceability:</b></td><td colspan="3">F17</td></tr>
<tr><td><b>Priority:</b></td><td colspan="3">Medium</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">• The system has updated fare rates for each route.
• GPS data for distance calculation is available.</td></tr>
<tr><td><b>Post Conditions:</b></td><td colspan="3">• The commuter can view the fare cost estimate before boarding.
• The system provides consistent fare estimates aligned with cooperative standards.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">• Commuter
• Mobile Application
• GPS Module</td></tr>
<tr><td rowspan="2"><b>Flow of Events:</b></td><td><b>Basic Flow</b></td><td><b>Alternative Flow</b></td><td><b>Exceptions</b></td></tr>
<tr><td>• Commuter opens the mobile application.
• Commuter selects a jeepney unit or destination.
• System calculates estimated distance based on route data.
• System displays the fare cost estimate.</td><td>If the commuter selects a different route or destination, the system changes the fare based on the route or destination provided.</td><td>None</td></tr>
<tr><td><b>Notes/Issues:</b></td><td colspan="3">• Fare estimates must comply with government-regulated fare matrices.
• Possible future integration with automated fare payment systems.</td></tr>
</table>

3.3.4 Use Case #4 (Dispatch Jeepney – U4)

<table>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Dispatch Jeepney</td></tr>
<tr><td><b>Author:</b></td><td colspan="3">OutCode</td></tr>
<tr><td><b>Purpose:</b></td><td colspan="3">To allow the dispatcher to assign jeepney units to a terminal and allocate drivers and conductors.</td></tr>
<tr><td><b>Requirements Traceability:</b></td><td colspan="3">F7, F8, F9</td></tr>
<tr><td><b>Priority:</b></td><td colspan="3">High</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">• Jeepney is available and active in the system.
• Drivers and conductors are registered and available for assignment.</td></tr>
<tr><td><b>Post Conditions:</b></td><td colspan="3">• Jeepney is successfully dispatched with assigned crew.
• Fleet records are updated.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">• Dispatcher
• Jeepney Unit
• Driver
• Conductor</td></tr>
<tr><td rowspan="2"><b>Flow of Events:</b></td><td><b>Basic Flow</b></td><td><b>Alternative Flow</b></td><td><b>Exceptions</b></td></tr>
<tr><td>• Dispatcher logs into the system.
• Dispatcher selects an available jeepney.
• Dispatcher assigns driver and conductor.</td><td>If assigned driver/conductor is unavailable, dispatcher chooses another available staff.</td><td>Jeepney marked under maintenance or no RPI4: Dispatch cannot proceed.</td></tr>
<tr><td><b>Notes/Issues:</b></td><td colspan="3">• Dispatch delays may occur if crew allocation is not automated.
• Must prevent duplicate assignments of the same driver.</td></tr>
</table>

3.3.5 Use Case #5 (Notify Jeepney – U5)

<table>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Notify Jeepney</td></tr>
<tr><td><b>Author:</b></td><td colspan="3">OutCode</td></tr>
<tr><td><b>Purpose:</b></td><td colspan="3">To allow the dispatcher to send alerts or notifications to one or multiple jeepney units about important events such as traffic rerouting or emergencies.</td></tr>
<tr><td><b>Requirements Traceability:</b></td><td colspan="3">F10</td></tr>
<tr><td><b>Priority:</b></td><td colspan="3">Medium</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">• Dispatcher must be logged into the system.
• Jeepney units must be online and connected to the network.</td></tr>
<tr><td><b>Post Conditions:</b></td><td colspan="3">• Drivers and conductors receive notifications on their devices.
• Jeepney units can take immediate corrective actions.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">• Dispatcher
• Driver
• Conductor
• Jeepney Unit Device</td></tr>
<tr><td rowspan="2"><b>Flow of Events:</b></td><td><b>Basic Flow</b></td><td><b>Alternative Flow</b></td><td><b>Exceptions</b></td></tr>
<tr><td>• Dispatcher selects jeepney(s)
• Dispatcher composes an alert message.
• System delivers notif to specified jeep devices.
• Driver and conductor acknowledge the alert.</td><td>If only a subset of jeepneys is selected, only those units receive the alert.</td><td>• Notification fails to deliver: System retries and alerts dispatcher of failed attempts.</td></tr>
<tr><td><b>Notes/Issues:</b></td><td colspan="3">• Requires stable internet to ensure timely notifications.
• Acknowledgment tracking may be required for critical alerts.</td></tr>
</table>

3.3.6 Use Case #6 (Trip Reports – U6)

Use Case Name:	Trip Reports		
Author:	OutCode		
Purpose:	To generate reports on jeepney utilization, including distance traveled and driver/conductor performance.		
Requirements Traceability:	F11, F12		
Priority:	High		
Preconditions:	• GPS and tracking devices must be functioning. • All trips must be properly recorded in the database.		
Post Conditions:	• Admins can review trip performance metrics. • Drivers and conductors can be evaluated for consistency.		
Actors:	• Admin • GPS Module • Database		
Flow of Events:	**Basic Flow**	**Alternative Flow**	**Exceptions**
	• Admin logs into the system. • System compiles distance traveled and trip frequency. • Report is displayed and can be exported.	Admin filters reports by date range, driver, or jeepney unit	• Database error: Report generation fails with an error message.
Notes/Issues:	• Weekly reporting cycles may need automation. • Data privacy for driver and conductor performance must be maintained.		

3.3.7 Use Case #7 (Account Management – U7)

Use Case Name:	Account Management		
Author:	OutCode		
Purpose:	To allow admins to add, edit, and delete accounts for drivers, conductors, and dispatchers.		
Requirements Traceability:	F13		
Priority:	High		
Preconditions:	• Admin must be logged into the system with proper authorization.		
Post Conditions:	• Accounts are updated in the system and synced across all modules.		
Actors:	• Admin • Driver • Conductor • Dispatcher		
Flow of Events:	**Basic Flow**	**Alternative Flow**	**Exceptions**
	• Admin logs into the system. • Admin chooses to add, edit, or delete an account. • System updates the database. • Changes are reflected in real-time.	When editing, only personal details may be updated.	• Attempting to delete an account tied to active trips: System prevents deletion. • Invalid data entry: System prompts admin to correct fields.
Notes/Issues:	• Must ensure strict access control for account changes. • Audit logs may be needed for accountability.		

3.3.8 Use Case #8 (Jeepney Management – U8)

<table>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Jeepney Management</td></tr>
<tr><td><b>Author:</b></td><td colspan="3">OutCode</td></tr>
<tr><td><b>Purpose:</b></td><td colspan="3">To allow admins to add, edit, delete, or deactivate jeepney units, and manage hardware devices attached to them.</td></tr>
<tr><td><b>Requirements Traceability:</b></td><td colspan="3">F14, F15, F16</td></tr>
<tr><td><b>Priority:</b></td><td colspan="3">High</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">• Jeepney details (plate number, route, etc.) are available.
• Hardware device (e.g., RPi4) is registered in the system.</td></tr>
<tr><td><b>Post Conditions:</b></td><td colspan="3">• Jeepney records are updated in the database.
• Hardware can be reassigned as necessary.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">• Admin
• Jeepney Unit
• RPi4 Device</td></tr>
<tr><td><b>Extends:</b></td><td colspan="3">None</td></tr>
<tr><td rowspan="2"><b>Flow of Events:</b></td><td><b>Basic Flow</b></td><td><b>Alternative Flow</b></td><td><b>Exceptions</b></td></tr>
<tr><td>• Admin logs into the system.
• Admin adds, edits, deletes, or deactivates a jeepney.
• System updates fleet records.</td><td>Admin detaches RPi4 from a unit and reassigns to another jeepney.</td><td>• Jeepney under active trip: System blocks deletion or deactivation.</td></tr>
<tr><td><b>Includes:</b></td><td colspan="3">None</td></tr>
<tr><td><b>Notes/Issues:</b></td><td colspan="3">• Accurate recordkeeping is required for fleet compliance.
• Must prevent duplicate jeepney entries.</td></tr>
</table>

3.3.9 Use Case #9 (Announcements – U9)

Use Case Name:	Announcements		
Author:	OutCode		
Purpose:	To allow dispatchers to create and broadcast announcements to drivers and conductors.		
Requirements Traceability:	F10		
Priority:	Medium		
Preconditions:	• Dispatcher must be logged into the system.		
Post Conditions:	• Announcements are posted on the mobile application.		
Actors:	• Drivers • Conductors • Dispatchers		
Flow of Events:	**Basic Flow**	**Alternative Flow**	**Exceptions**
	• Dispatcher creates an announcement. • System broadcasts the announcement. • Drivers and Conductors receive notifications in the app.	Announcement may include attachments.	• Network issue: Announcement delivery is delayed. • User device offline: Notification is queued until reconnection.
Notes/Issues:	• Critical announcements may require confirmation of receipt. • Expired announcements should be archived automatically.		

4 Other Non-functional Requirements

4.1 Performance Requirements

The following performance requirements ensure that the system operates efficiently under real-time conditions and supports smooth usage by both commuters and administrators. These requirements are derived from the system's functional and non-functional goals, technical architecture, and stakeholder expectations:

P1. GPS Update Frequency

- The system shall update the location of each jeepney unit every 10 seconds.
- *Rationale:* Enables tracking for both commuters and dispatchers, allowing timely decisions on route planning and unit availability.

P2. Occupancy Status Uptime

- The integrated camera module for monitoring passenger occupancy should be operational during the system’s active hours (6:00 AM to 10:00 PM).
- *Rationale:* Ensures commuters can reliably view seat availability and improve user satisfaction.

P3. Image Transmission Latency

- The system shall transmit captured occupancy images to the backend within 5 seconds of capture.
- *Rationale:* Maintains periodic updated of occupancy information to commuters.

P5. Mobile App Data Refresh

- The commuter mobile application shall refresh jeepney location and occupancy data within 3 seconds after receiving updates from the server.
- *Rationale:* Ensures the user interface reflects the latest data for optimal route and boarding decisions.

P6. Network Failover

- The onboard hardware device (e.g., Raspberry Pi 4) shall automatically attempt to reconnect and resume data transmission within 10 seconds if the primary network connection fails.
- *Rationale:* Ensures continuous GPS and occupancy data reporting by minimizing service interruptions caused by connectivity issues.

4.2 Safety and Security Requirements

This section describes the required precautions and protections to ensure that the Smart Jeepney Monitoring System operates securely, avoids harm, and does not violate privacy.

1. **Secure Data Transmission**
    - All data transmitted between jeepney devices, the central server, the mobile app, and the admin dashboard must use HTTPS or other secure communication protocols.
2. **User Authentication**
    - All admin and drivers/conductors must log in using a unique username and password.
    - Passwords must be stored using secure hashing methods on the server.

3. **Role-Based Access**

   - Different users should have access only to features and data relevant to their role.

4. **Data Privacy Compliance**

   - The system must comply with the Data Privacy Act of the Philippines.
   - No personally identifiable information of commuters shall be collected or stored.

5. **Driver Safety Measures**

   - The driver interface must be non-interactive during active driving or limited to the
   - Any driver-visible UI must be simple, minimal, and non-distracting.

6. **Protection Against Unauthorized Access**

   - The system must have session timeouts for inactive admin and driver accounts.

4.3 Software Quality Attributes

The system shall meet specific software quality attributes to ensure reliability, usability, maintainability and adaptability. These attributes are designed to guarantee smooth operations, long-term sustainability, and flexibility for future improvements.

4.3.1 Reliability

The system shall maintain dependable operations even during connectivity or sensor issues by implementing backup and recovery mechanism.

- Enable automatic synchronization of data once internet connection is restored.
- Add error-handling routines to detect and mitigate inaccurate sensor readings.
- Ensure data redundancy and backup to protect against data loss in case crashes.

4.3.2 Usability

The system shall provide a simple, user-friendly interface to minimize confusion and reduce the need for technical testing.

- Design a clean interface with icons and labels
- Ensure mobile-friendly design for easy access on smartphones.
- Provide immediate updates of occupancy and GPS data to reduce delays.
- Maintain consistent design to reduce learning curve.

4.3.3 Maintainability

The system shall be easy to update, debug, and extent without requiring full redesigns.

- Ensure proper documentation of the system, including developer guidelines and API references, to simplify knowledge transfer.
- Utilize version control (e.g., Git) and enforce coding standards to maintain consistency across updates.
- Document common troubleshooting scenarios like GPS signal drop or camera failure and recommended fixes.

4.3.4 Adaptability

The system shall be designed to support future changes, new technologies, and cooperative growth without major changes.

- Ensure scalability by allowing more modernized jeepney units to be added as the cooperative expands.
- Allow easy adjustments in system settings and configurations without extensive redevelopment

Appendix A – Data Dictionary/ERD

Jeepneys Collection

Field_name	Field_type	Description
conductor_id	String/Text	Unique identifier assigned to the conductor of the jeepney
driver_id	String/Text	Unique identifier assigned to the driver of the jeepney
destination	String/Text	The jeepney's route endpoint or target location
plate_number	String/Text	Official license plate number of the jeepney
status	String/Text	Current operational status

Notifications Collection

Field_name	Field_type	Description
date	Date	The calendar date when the notification was created or sent
from_admin	Boolean	The calendar date when the notification was created or sent
message	String/Text	The content or body of the notification
subject	String/Text	The title or subject line summarizing the notification
time	Time	The specific time the notification was issued

Trips Collection

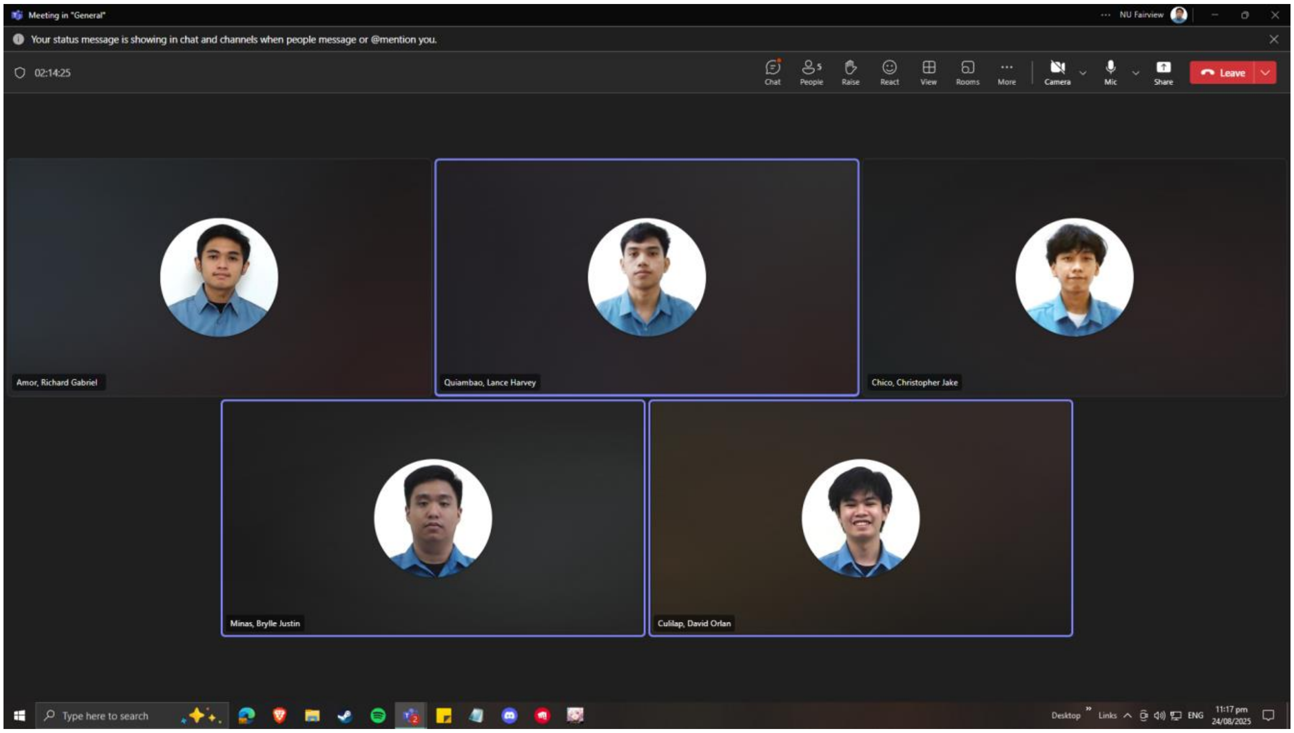
Field_name	Field_type	Description
conductor_id	String/Text	Unique identifier assigned to the conductor
conductor_name	String/Text	Full name of the conductor
date	Date	The date of the trip or record entry
driver_id	String/Text	Unique identifier assigned to the driver
driver_name	String/Text	Full name of the driver
last_updated	String/Text	The last recorded update for this entry
latitude	Number	GPS latitude coordinate of the jeepney's location
longtitude	Number	GPS longitude coordinate of the jeepney's location
meters_run	Number	Total distance traveled by the jeepney in meters
plate_number	String/Text	Official license plate number of the jeepney

Users Collection		
Field_name	**Field_type**	**Description**
waiting	Boolean	Indicates if the user/driver is currently waiting
email	String/Text	Email address of the user
first_name	String/Text	First name of the user
last_name	String/Text	Last name of the user
driver_name	String/Text	Full name of the driver
password	String/Text	User’s account password
role	String/Text	User’s role identifier
uid	Number	Unique identifier for the user record
waitingMZ	Boolean	Indicates if the user/driver is waiting at a specific terminal/zone
waitingSP	Boolean	Indicates if the user/driver is waiting at a specific terminal/zone

RPI Collection		
Field_name	**Field_type**	**Description**
rpi_id	String/Text	Unique identifier assigned to the Raspberry Pi device installed in the jeepney
vacant	Boolean	Indicates if there are available passenger seats

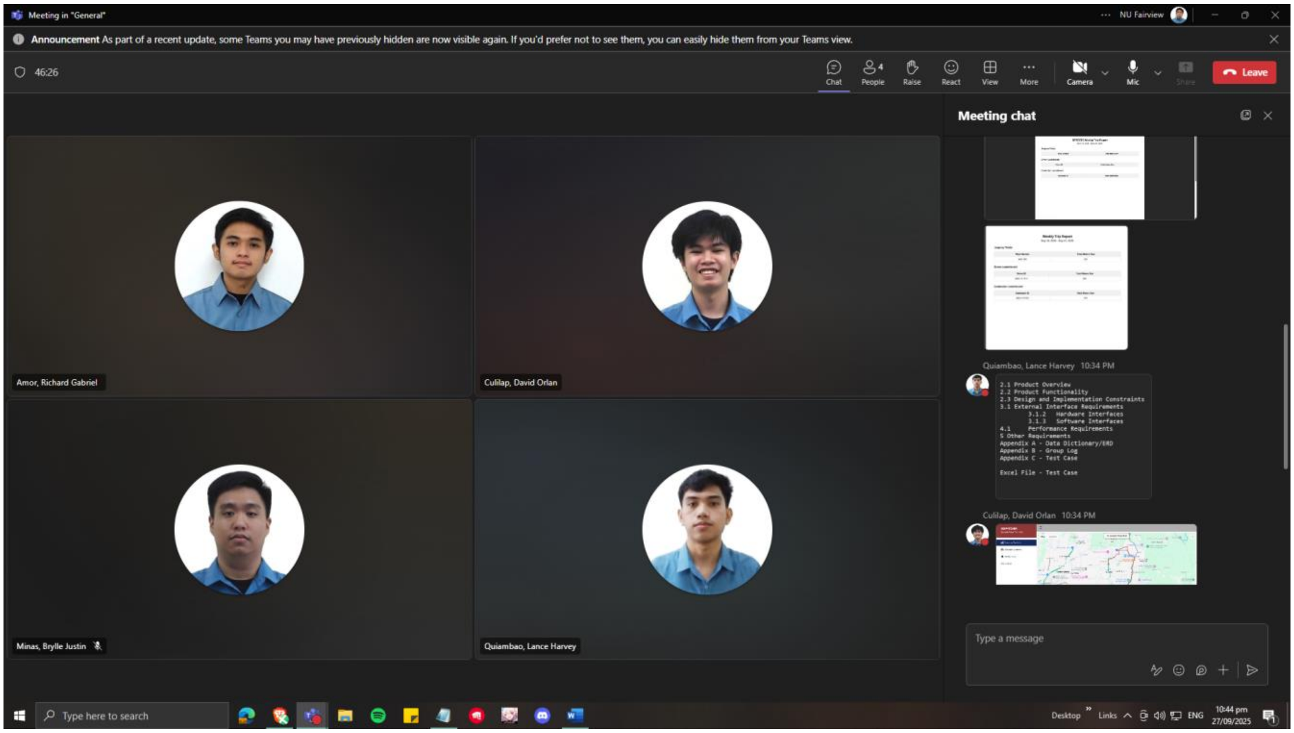
Terminals Collection		
Field_name	**Field_type**	**Description**
dispatches_to	String/Text	The destination or terminal where jeepneys from this terminal are dispatched
terminal_name	String/Text	The official name of the terminal

Appendix B - Group Log



Meeting in "General"

Your status message is showing in chat and channels when people message or @mention you.



Meeting in "General"

Announcement As part of a recent update, some Teams you may have previously hidden are now visible again. If you'd prefer not to see them, you can easily hide them from your Teams view.

Meeting chat

Quiambao, Lance Harvey 10:34 PM

```
2.1 Product Overview
2.2 Product Functionality
2.3 Design and Implementation Constraints
3.1 External Interface Requirements
        3.1.2   Hardware Interfaces
        3.1.3   Software Interfaces
4.1     Performance Requirements
5 Other Requirements
Appendix A - Data Dictionary/ERD
Appendix B - Group Log
Appendix C - Test Case

Excel File - Test Case
```

Culilap, David Orlan 10:34 PM

Appendix C – Test Plan/Test Cases

Test Case#	Test Title	Test Summary	Test Steps	Test Data	Expected Results	Post-condition	Actual Result	Status
TC001	Verify App Launch	Ensure the app launches without errors.	1. Install app. 2. Tap app icon.	Installed APK.	Splash screen loads, then home screen appears.	App is running.	Succe ss	Pass
TC002	Verify User Login with Valid Credentials	Check login flow works with correct credentials.	1. Enter valid username & password. 2. Tap login.	Valid credential s.	User logs in and lands on home screen.	Session active.	Succe ss	Pass
TC003	Verify User Login with Invalid Credentials	Ensure invalid login shows error message.	1. Enter wrong password. 2. Tap login.	Invalid credential s.	Error message “Invalid username/pa ssword.”	User not logged in.	Succe ss	Pass
TC004	Verify Logout Functionality	User should be logged out successfully.	1. Log in. 2. Tap logout. 3. Confirm logout.	Active user session.	User is redirected to login screen.	Session ended.	Succe ss	Pass
TC005	Verify Navigation Drawer Items	Ensure drawer items open correct pages.	1. Open navigation drawer. 2. Tap each menu item.	User logged in.	Each menu opens correct screen.	Navigation works.	Succe ss	Pass
TC006	Verify App Runs Without Internet	App should handle offline gracefully.	1. Disable Wi-Fi/data. 2. Launch app.	No internet.	App shows “No connection” but doesn’t crash.	App running.	Succe ss	Pass
TC007	Verify Push Notification Received	Confirm push notifications appear.	1. Trigger server push. 2. Observe device.	Push enabled device.	Notification appears in tray.	Notificatio n visible.	Succe ss	Pass
TC008	Verify Push Notification Tap Action	Ensure tapping notification opens correct screen.	1. Tap incoming notification.	Notificatio n data.	User navigates to relevant screen.	Screen loaded.	Succe ss	Pass
TC009	Verify App Permissions Request	App must request needed permissions.	1. Fresh install. 2. Launch app.	No permissio ns granted.	Prompts for location, storage, etc.	User chooses.	Succe ss	Pass
TC010	Verify Location Services Disabled	App should handle when GPS is off.	1. Disable GPS. 2. Launch app.	Location OFF.	App shows “Enable Location” prompt.	User can enable GPS.	Succe ss	Pass

TC011	Verify Background to Foreground Resume	App should resume correctly.	1. Launch app. 2. Minimize. 3. Reopen.	Active session.	App restores previous state.	State preserved.	Success	Pass
TC012	Verify Waiting Status Toggle Updates Firestore and Notification	Ensure commuter waiting status updates Firestore and shows notification.	Steps as in earlier case.	Valid Firebase user.	Firestore Waiting = true, notification shown; reset on stop.	Waiting reset.	Success	Pass
TC013	Verify Default Route Loads on First Launch	Check if the default route (Muzon) is loaded at first app start.	1. Install fresh app. 2. Launch app. 3. Observe route spinner and map.	New install, no prefs.	Spinner shows “To Muzon”. Map loads default route.	Default route ready.	Success	Pass
TC014	Verify Route Switching	Ensure selecting Sapang Palay updates waypoints and redraws route.	1. Open spinner. 2. Select “To Sapang Palay”. 3. Observe route redraw.	Existing user.	Waypoints switch to alternate set.	CurrentRouteWaypoints updated.	Success	Pass
TC015	Verify Last Selected Route Persistence	Route preference should persist between sessions.	1. Select “To Sapang Palay”. 2. Close app. 3. Relaunch app.	Saved prefs.	Spinner opens with “To Sapang Palay” selected.	Last selection retained.	Success	Pass
TC016	Verify Nearest Jeep FAB with Jeeps Present	FAB cycles through nearest jeep markers.	1. Add active jeep markers in Firestore. 2. Tap nearest jeep FAB repeatedly.	Driver trips exist.	Map pans to nearest jeep, cycles through all.	SortedJeepList updated.	Success	Pass
TC017	Verify Nearest Jeep FAB with No Jeeps	FAB should show toast if no jeeps available.	1. Ensure no drivers active. 2. Tap nearest jeep FAB.	No trip docs.	Toast “No active jeeps currently on the map.”	No change.	Success	Pass
TC018	Verify Refresh Button Rotates and Updates UI	Manual refresh triggers marker and sheet updates.	1. Tap refresh button. 2. Observe rotation animation and UI.	Valid session.	Refresh log, markers update, button animates.	UI refreshed.	Success	Pass

TC019	Verify Auto Refresh Runs Every Second	Ensure background refresh happens on timer.	Wait >5s with app open.	Session active.	Logs show “Auto-refresh triggered” every second.	Auto refresh running.	Success	Pass
T20	*Verify User Bottom Sheet Opens on Tap*	*User sheet opens with correct details.*	*1. Tap user marker.* *2. Observe bottom sheet.*	*Logged-in user.*	*User sheet shows name, waiting status button.*	*Sheet open.*	Success	Pass
T21	Verify Jeep Bottom Sheet Opens	Jeep sheet shows jeep details when tapped.	1. Tap jeep marker. 2. Observe bottom sheet.	Active driver marker.	Sheet shows jeep plate, capacity, distance.	Sheet open.	Success	Pass
T22	Verify Dispatcher/Admin Login	Check if respective user logs in successfully	1. Enter login credentials 2. Click submit	Valid credentials	User is redirected to the first module of their respective role	Session active	Success	Pass
T23	Add user account	Check if respective user is being added	1. Go to Account Management 2. Click the add icon	User information	Added user is reflected on the table in its respective role	Table updated	Success	Pass
T24	Delete user account	Check if respective user is being deleted	1. Go to Account Management 2. Click the delete button	User information	Deleted user is removed on the table in its respective role	Table updated	Success	Pass
T25	Edit user account	Check if respective user is being edited	1. Go to Account Management 2. Click the edit button	User information	Edited user information is updated on the table in its respective row	Table updated	Success	Pass
T26	Add jeepney	Check if jeepney is being added	1. Go to Jeepney Management 2. Click the add icon	Jeepney information	Added jeepney is reflected on the table	Table updated	Success	Pass
T27	Delete jeepney	Check if jeepney is being deleted	1. Go to Jeepney Management 2. Click the delete button	Jeepney information	Deleted jeepney is removed on the table	Table updated	Success	Pass
T28	Edit jeepney	Check if jeepney is being edited	1. Go to Jeepney Management	Jeepney information	Edited jeepney is	Table updated	Success	Pass

			2. Click the edit button		updated on the table			
T29	Detach RPI on jeepney	Check if the RPI of jeepney is removed	1. Go to Jeepney Management 2. Click the edit button 3. Click detach RPI 4. Confirm detachment	RPI information	Jeepney is moved to ‘No RPI4’ section	Table updated	Success	Pass
T30	Attach RPI on jeepney	Check if the RPI of jeepney is attached	1. Go to Jeepney Management 2. Go to ‘No RPI4’ section 3. Click the edit button 4. Click detach RPI 5. Confirm detachment	RPI information	Jeepney is moved to ‘Active’ section	Table updated	Success	Pass
T31	Filter trip reports	Check if the trip reports is displaying properly	1. Trip reports 2. Click the ‘Specific’ radio button 3. Enter input details for filtering 4. Click submit	Trips information	Trip details are filtered based on the specified input data	Table updated	Success	Pass
T32	Download PDF of trip reports	Check if the PDF download for trip reports is displaying the expected output	1. Go to Trip Reports 2. Click ‘Download Report (PDF)’	Trips information	Trip details are reflected properly on the PDF	PDF downloaded	Success	Pass
T33	View dispatched jeepneys	Check if the dispatched jeepneys are reflected on the map	1. Go to Jeepney Dispatch and check for dispatched jeepneys, create one if none. 2. Go to Jeepney Tracking 3. Look for the	Jeepney Data	Dispatched jeepneys are being reflected on the map	Map page updated	Success	Pass

			dispatched unit					
T34	Dispatch jeepney	Check if jeepney is being dispatched	1. Go to Jeepney Dispatch and look for waiting jeepneys. 2. Assign a driver and conductor. 3. Click dispatch	Jeepney Data	Dispatched jeepney is moved on the 'Dispatched' column	Table updated	Success	Pass
T35	Stop/Abort Jeepney	Check if the jeepney is being stopped/aborted	1. Go to Jeepney Dispatch and look for dispatched/incoming jeepneys. 2. Click the jeepney. 3. Click stop/abort	Jeepney Data	Stopped/Aborted jeepney is moved on the 'Dispatched' column	Table updated	Success	Pass
T36	Notify Jeepney	Check if notifications are being sent.	1. Go to Notify Jeep 2. Click New Notification 3. Enter required inputs.	Message Data	Table is updated with new message and is reflected on the jeepney's driver/conductor's mobile app	Table and UI updated	Success	Pass