**CEBU INSTITUTE OF TECHNOLOGY**

**UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

AI Foot Traffic Heatmap System

**BALTAZAR**, Rigel L.

**CARBUNGCO**, Louie James F.

**KYONO** JR., Yoshinori L.

**LACANGLACANG**, Nick Carter

**RAVANES**, Jierelle Jane S.

*SAYSTUE10301130*

*February 22, 2025*

Change History

Table of Contents

Change History 2

Table of Contents 3

1. Introduction 4

1.1. Purpose 4

1.2. Scope 4

1.3. Definitions, Acronyms and Abbreviations 4

1.4. References 4

2. Overall Description 5

2.1. Product perspective 5

2.2. User characteristics 5

2.4. Constraints 5

2.5. Assumptions and dependencies 6

3. Specific Requirements 7

3.1. External interface requirements 7

3.1.1. Hardware interfaces 7

3.1.2. Software interfaces 7

3.1.3. Communications interfaces 7

3.2. Functional requirements 7

Module 1 7

Module 2 8

3.4 Non-functional requirements 8

Performance 8

Security 8

Reliability 8

# Executive Summary

**Para: The Commute Companion**

Commuting in urban areas can be challenging, especially when navigating through various jeepney routes and transportation options. PARA: The Commute Companion is a web and mobile application designed to streamline and enhance the commuting experience by providing real-time route navigation, intelligent trip planning, and seamless integration with the Public Utility Vehicle (PUV) system, specifically Public Utility Jeepneys (PUJs).

Through advanced GPS technology, cloud-based data synchronization, and an intuitive user interface, PARA empowers users to find the most efficient routes, track their travel history, and make informed commuting decisions—all in one convenient platform. The app ensures accessibility across both web and mobile devices while incorporating biometric login for enhanced security and ease of access.

## Problem Statement and Proposed Solution

Foot traffic analysis is essential for businesses, malls, transportation hubs, and urban planning to understand movement patterns. Traditional methods rely on manual counting or basic surveillance, which lack accuracy and efficiency.

***PROPOSED SOLUTION***

Our proposed AI Foot Traffic Heatmap System uses computer vision and AI to analyze live video feeds, detect people, and generate real-time heatmaps that visualize high-traffic areas. This solution will enhance business insights, space optimization, and security monitoring.

## Expected Benefits and Impacts

PARA will have their services provided through the web and a mobile application. The developers will be using Java, JavaScript, SQL, Android Kotlin, XML, and CSS to build the frontend and the backend of the system.

PARA will:

* Provide GPS-based route navigation to help users locate their current position and find the best jeepney routes to their destination.
* Suggest the shortest and most efficient routes based on distance.
* Allow users to save frequently used routes for quick access.
* Offer biometric login for mobile users, with an alternative email-password option.
* Enable synchronized data access between web and mobile applications through cloud integration.
* Services are only offered to android mobiles.

PARA will not:

* Provides real-time jeepney arrival tracking, as it does not integrate directly with jeepney operators' tracking systems.
* Guarantee real-time traffic updates, the app will only provide static route data.
* Offer booking or ride-hailing services, as it focuses on route guidance and not direct transportation services.

## Definitions, Acronyms and Abbreviations

**API -** API is the acronym for application programming interface — a software intermediary that allows two applications to talk to each other. APIs are an accessible way to extract and share data within and across organizations.

**Android Kotlin -** Kotlin is a programming language used for Android development. It's simpler and safer than Java while still working well with Java.

**Commuters -** A commuter is someone who has a lengthy trip to work, usually from a suburb to a city.

**Database -** A database is any collection of data, or information, that is specially organized for rapid search and retrieval by a computer.

**GPS -** The Global Positioning System (GPS) is a navigation system using satellites, a receiver, and algorithms to synchronize location, velocity, and time data for air, sea, and land travel.

**HTML/CSS -** HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages.

**Java -** Java is a high-level, class-based, object-oriented programming language.

**JavaScript -** JavaScript is the Programming Language for the Web. JavaScript can update and change both HTML and CSS.

**Jeepney -** A type of public vehicle (PUV) in the Philippines.

**Leaflet -** Leaflet is an open-source JavaScript library for interactive maps. This R package makes it easy to create Leaflet maps from R.

**Mobile Application -** Mobile applications are software programs built for a specific platform like tablets, smartphones, and other touch devices.

**PARA** – Para po is a Filipino word commuters use to tell the jeepney driver that they have arrived at their destination.

**PUJ -** The Philippine public utility jeepney (PUJ), or Jeepneys for short, is an iconic mode of transportation in the country.

**PUV -** Public Utility Vehicles (PUVs) are public utility vehicles. They are made up of a diverse range of transportation modes designed to cater to the general public.

**Route -** A route is a way for travel or movement, the path from point A to point B. A route can also be the method used for achieving a particular result, like going to school and working hard is your route to success.

**SQL -** Structured Query Language (SQL) is a standard language for accessing and manipulating databases.

**Web Application -** A web application is an application software that does not require installation and can instead be accessed from a remote server via a web browser.

**XML** - XML also known as (eXtensible Markup Language) is a markup language used to store and transport data in a structured format.

**OpenStreetMap** - OpenStreetMap is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license.

## References

"About OpenStreetMap," *OpenStreetMap Wiki*. [Online]. Available: <https://wiki.openstreetmap.org/wiki/About_OpenStreetMap>. [Accessed: 31-Jan-2025].

"What is GPS?" *Geotab Blog*, Dec. 6, 2022. [Online]. Available: <https://www.geotab.com/blog/what-is-gps/>. [Accessed: 31-Jan-2025].

"Database," *Encyclopædia Britannica*, [Online]. Available: <https://www.britannica.com/technology/database>. [Accessed: 31-Jan-2025].

"Route," *Vocabulary.com*, [Online]. Available: <https://www.vocabulary.com/dictionary/route>. [Accessed: 31-Jan-2025].

"What is an API?," *MuleSoft*, [Online]. Available: <https://www.mulesoft.com/api/what-is-an-api>. [Accessed: 31-Jan-2025].

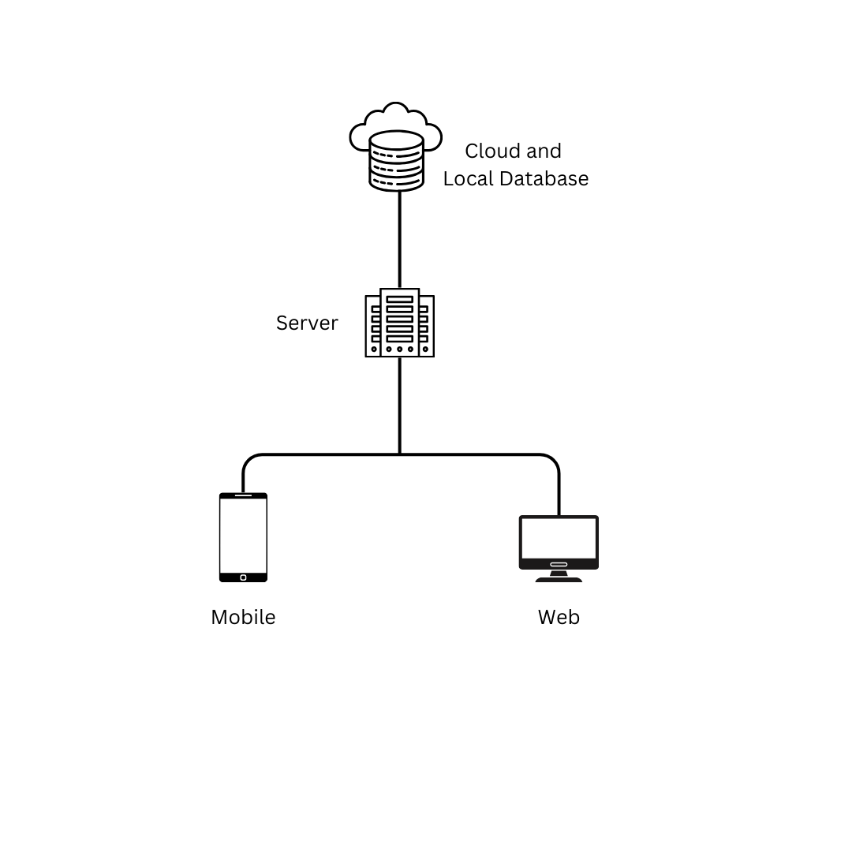
"Commuter," *Vocabulary.com*, [Online]. Available: <https://www.vocabulary.com/dictionary/commuter>. [Accessed: 31-Jan-2025].

"What is a web app?," *Codecademy*, [Online]. Available: <https://www.codecademy.com/article/what-is-a-web-app>. [Accessed: 31-Jan-2025].

"Public Utility Vehicle (PUV)," *LTO Portal*, [Online]. Available: <https://ltoportal.ph/public-utility-vehicle-puv/>. [Accessed: 31-Jan-2025].

# Overall Description

## Product perspective



### Module 1. Route Finder and Navigation Assistance

**Transaction 1.1** Retrieve current location using GPS.

**Transaction 1.2** Allow users to input a desired initial location and destination.

**Transaction 1.3** Fetch and display possible routes using OpenStreetMap and Leaflet

**Transaction 1.4** Provide step-by-step navigation assistance.

### Module 2. Distance-based Route Suggestion

**Transaction 2.1** Calculate and display the shortest route based on distance.

**Transaction 2.2** Highlight the most efficient route in bold colors.

**Transaction 2.3** Suggest alternative routes based on the user's location.

### Module 3. User Management

**Transaction 3.1** Register a new user (via email, social login, or biometric setup)

**Transaction 3.2** Authenticate and log in a user.

**Transaction 3.3** Manage user profile (update name, email, password, and preferences).

**Transaction 3.4** Log out a user.

### Module 4. Cloud-Integrated Mobile and Web Application Module

**Transaction 4.1** Sync user data across mobile and web using Firebase/Azure.

**Transaction 4.2** Store and retrieve user preferences, saved routes, and account details.

**Transaction 4.3** Handle secure database storage for route history and settings.

### Module 5. Saved Routes and Locations ("My Routes")

**Transaction 5.1** Save a searched route to the user’s profile.

**Transaction 5.2** Retrieve and display previously saved routes.

**Transaction 5.3** Delete or edit saved routes.

### Module 6. Biometric Login Module (Mobile Exclusive)

**Transaction 6.1** Enable biometric authentication (fingerprint/face ID).

**Transaction 6.2** Verify biometric credentials.

**Transaction 6.3** Provide a fallback login option via email and password.

## User characteristics

Commuters are the primary users of the app who rely on jeepneys for transportation. They have the following roles and privileges:

**Student**

Users commuting to schools, universities, or training centers.

**Regular Commuter**

Everyday users of jeepneys for work, errands, or daily routines.

Likely to use the service.

**Tourist**

Visitors that are unfamiliar with local jeepney routes and customs.

Can access beginner-friendly guides, tourist destination routes.

**Occasional Commuter**

Users who ride jeepneys only on specific occasions or routes.

May rely on the app for quick route planning.

## 2.4. Constraints

**Regulatory Policies**

* **Compliance with Local Transport Laws**: The developers must adhere to the regulations governing public transportation in the Philippines, ensuring that no claims of real-time tracking or booking are made if not backed by actual services.
* **Data Privacy**: All user data, especially biometric data for login, must comply with local and international data privacy laws (e.g., GDPR, Data Privacy Act of 2012).

**Hardware Limitations**

* **Mobile Device Compatibility**: Since services are only offered to Android mobiles, the application must be compatible with a wide range of Android devices, ensuring proper functionality across various screen sizes, operating system versions, and hardware capabilities.
* **Signal and GPS Quality**: The quality of GPS tracking may vary based on the location, weather, and mobile network conditions, which can affect route suggestions and the accuracy of real-time location data.

**Interfaces to other Applications**

* **Web and Mobile Synchronization**: The application must support data synchronization between the web and mobile versions via cloud integration, ensuring users can access their saved routes and settings across devices.
* **No Integration with Jeepney Operators**: Since PARA does not integrate directly with jeepney operators for real-time tracking, the system will only offer suggested routes based on available data.

**Parallel Operation**

* **Simultaneous Use Restrictions**: The application must support seamless operation between the web and mobile platforms, but ensuring data consistency and avoiding conflicts when used simultaneously can be challenging.

**Audit Functions**

* **Activity Logging Limitations**: Logging user activities, including route searches and saved routes, may require additional system resources and could affect performance if not properly optimized.
* **Data Sync Logs**: Maintaining synchronization logs between platforms introduces overhead in managing and storing large datasets.

**Control Functions**

* **Route Suggestion Constraints**: The accuracy and efficiency of the route suggestion algorithm may be limited by factors such as GPS data quality and the absence of real-time traffic updates.
* **User Account Management**: Implementing biometric authentication and email-password login may be limited by device capabilities and security considerations.

**Reliability Requirements**

* **System Availability**: The application must maintain high availability, especially during peak commute hours, but system uptime may be limited by factors such as server performance, network stability, and data load.

**Criticality of the Application**

* Public Use Dependence: PARA’s reliance on accurate route suggestions for public transport makes it critical for commuters, but the quality of the service may be limited by available data and GPS accuracy.

## 2.5. Assumptions and dependencies

### Assumptions

**Android Mobile Device Availability**

It is assumed that the majority of users will have Android mobile devices, as PARA is only available for Android users. The app will be optimized for a range of Android devices, including various screen sizes, hardware specifications, and Android OS versions.

**Impact:** This assumption impacts the device compatibility requirements and performance considerations for the mobile app, as it should be designed to work across different Android platforms.

**Cloud Infrastructure for Data Synchronization**

It is assumed that a reliable and scalable cloud infrastructure will be available to handle the data synchronization between the web and mobile versions of the app. This includes the storage and management of user data (such as saved routes) and settings.

**Impact**: The cloud infrastructure will directly affect the application’s ability to synchronize user data seamlessly across platforms, and any issues with cloud performance could result in data loss or synchronization delays.

**Access to GPS Functionality**

It is assumed that all users will have access to GPS functionality on their mobile devices. The app relies heavily on GPS data to offer route suggestions, track the user’s current location, and recommend the shortest route.

**Impact**: If the user’s mobile device lacks GPS functionality, it would limit or entirely disable key features of the app. Similarly, poor GPS signal quality or interference (e.g., in indoor environments or areas with weak satellite coverage) could affect the accuracy of route suggestions.

**Stable Internet Connectivity**

It is assumed that users will have a stable internet connection to use the app's features, especially for cloud-based synchronization, GPS data retrieval, and route calculation. The app may not function optimally in offline mode.

**Impact**: Inconsistent or poor network connectivity can hinder real-time features of the app, such as route suggestions and data synchronization. This assumption influences the app’s design to prioritize efficient use of network data while also considering offline use cases.

**User Familiarity with Basic App Functionality**

It is assumed that users will have basic knowledge of using a smartphone app (e.g., navigation, setting up an account). The app’s user interface will be designed to be simple and intuitive to accommodate both tech-savvy and non-tech-savvy users.

**Impact**: The user interface will need to be user-friendly and accessible, with clear instructions and feedback for users who may not be familiar with navigation apps or tech-heavy platforms.

### Dependencies

**Accurate Local Transport Data**

The app’s route suggestions depend on accurate jeepney route data. Since PARA doesn’t integrate with jeepney operators for real-time tracking, it relies on static data.

**Impact**: Inaccurate or outdated data may result in suboptimal route recommendations, affecting the user experience.

**Third-Party APIs and Services**

PARA depends on external services for cloud storage, map routing, and user authentication. These services directly impact the app’s performance.

**Impact**: Downtime or issues with third-party services, such as API changes, can lead to app malfunctions, affecting route calculations, data synchronization, and login features.

**Hardware Capabilities of Devices**

The app’s performance depends on the user’s device, with higher-end devices offering better GPS accuracy and processing speed.

**Impact**: Users with older or less powerful devices may experience delays or reduced accuracy in features like route suggestions.

**Data Privacy and Security Regulations**

PARA must comply with data privacy laws (e.g., GDPR, Data Privacy Act of 2012) to secure user data, including biometric data.

**Impact**: The app must implement secure data handling to avoid legal and reputational issues.

**User Engagement and Data Contribution**

The app assumes users will engage in saving and managing routes, which influences cloud synchronization and route suggestions.

**Impact**: Low user engagement can hinder functionality, limiting opportunities to refine route suggestions.

**Regulatory Restrictions on Real-Time Data**

PARA cannot provide real-time tracking or ride-hailing due to the lack of integration with jeepney operators.

**Impact**: The app’s competitiveness is limited, and users may be disappointed by the lack of real-time updates.

**Availability of Accurate Traffic Data**

PARA’s route suggestions are based on available traffic data but do not offer real-time updates.

**Impact**: Suggested routes may be suboptimal during heavy traffic or unexpected delays, leading to user frustration.

# Specific Requirements

Qualifications are required for the functional requirements to operate and relay its services. Below is the list of specific requirements needed by PARA.

## External interface requirements

### 3.1.1. Hardware interfaces

The PARA app will support the following hardware configurations:

* **Mobile Devices**: Android (minimum API Level 24 - Nougat)
* **Web Browsers**: Chrome, Firefox, Safari, and Edge (latest versions).
* **GPS Hardware**: The app relies on device-embedded GPS sensors to determine location data accurately.
* **Biometric Sensors**: Devices must have fingerprint scanners or facial recognition capabilities for biometric authentication.
* **Server Hardware Requirements**: Cloud-based, scalable servers (Google Firebase / Microsoft Azure) for hosting and real-time data processing.

### 3.1.2. ￼Software interfaces

The PARA app integrates with several software components:

* **Operating Systems**:
  + Mobile: Android and iOS
  + Web: Windows, macOS, and Linux-supported browsers
* **Third-Party APIs and Services**:
  + **Mapping Services**: OpenStreetMap (OSM), Leaflet.js
  + **Cloud Database**: Google Firebase, Microsoft Azure Database
  + **Authentication**: Firebase Authentication, OAuth for social logins
  + **Notification Services**: Firebase Cloud Messaging (FCM) for push notifications
* **Frameworks & Libraries**:
  + React Native for mobile development
  + Node.js/Express.js for backend services
  + PostgreSQL or Firebase Firestore for real-time data storage

### 3.1.3. Communications interfaces

* **Internet Connectivity**: Required for real-time GPS navigation, route searching, and cloud synchronization.
* **Network Protocols**:
  + HTTPS (Secure API requests)
  + WebSocket for real-time data exchange (live location updates, traffic data retrieval)
* **Data Exchange Formats**:
  + JSON for API responses
  + RESTful API endpoints for fetching routes, user data, and saved locations
* **Security Measures**:
  + End-to-end encryption (TLS 1.2/1.3) for secure communication
  + OAuth-based authentication for user access control

## Functional requirements

### For Web and Mobile Application

**Module 1** **Route Finder and Navigation Assistance**

The app uses GPS Service to integrate a mapping service that enables users to locate their current position (or desire initial location) and search for routes to a specific destination, making commuting or travel more convenient. This will be done with the use of several APIs, including but not limited to the following: OpenStreetMap and Leaflet.

**Module 2** **Distance-based Route Suggestion**

The app displays the shortest route to the destination in bold colors. This algorithm enables the user to choose the jeepney services more efficiently.

**Module 3** **User Management**

Users are asked to sign-in or create an account to keep track of their posts and inquiries

**Module 4** **Cloud-Integrated Mobile and Web Application**

Leveraging server and online database integration, the app ensures synchronized and uninterrupted access to user data on both mobile and web platforms. Google Firebase and Microsoft Azure Database will be used to store this information.

**Module 5** **Saved Routes and Locations “My Routes”**

This feature enables users to view previously searched routes, such as jeepney routes they have looked up (e.g., 01K). The app saves the route details, including the map, stops, and directions, allowing users to revisit and review the same route without needing to search again.

### Mobile Exclusive Functionalities

**Module 6** **Biometric Login**

Users can log-in to the application through their own biometric signature. This enables users to open and use the service faster and more efficiently. The app also provides a fallback option for logging in, i.e. email and password.

##### . . .

## Non-functional requirements

* Performance
* The app must respond to user actions (e.g., searching for routes, saving locations) within **2 seconds** under normal network conditions.
* The map should render within **3 seconds** after a location search.
* Biometric login must authenticate users within **1.5 seconds**.
* The system should handle up to **10,000 concurrent users** without significant lag.

### Security

**User Authentication**:

* Supports biometric login (fingerprint, face recognition) with a fallback to email-password authentication.
* Uses **OAuth 2.0** for secure sign-in with Google/Facebook.

**Data Protection**:

* User data is encrypted using **AES-256** before being stored in the cloud.
* Secure communication with **TLS 1.2/1.3** for API requests.

**Access Control**:

* Users can only modify their own saved routes and settings.
* Admin-level users can manage and update route information.

### Reliability

Availability:

* The app should have 99.5% uptime, ensuring minimal service disruptions.
* Offline mode allows users to view saved routes without an internet connection.

Backup & Recovery:

* Data is backed up every 24 hours in cloud storage.
* In case of server failure, automatic failover ensures minimal downtime.

Error Handling:

* Provides descriptive error messages for failed actions (e.g., “Route not found” instead of a generic error).
* Implements retry mechanisms for API failures, especially for fetching routes and login.