**REQUIREMENT SPECIFICATION DOCUMENT**

**J&T EXPRESS PARCEL TRACKING SYSTEM WITH GPS MOBILE ACCESS**

A Software Engineering Project Presented to the Faculty of

Datamex College of Saint Adeline in Valenzuela Branch

MR. GABRIEL THOMAS TORNEROS

**By:**ANGELES, ADRIAN R.

BSIT-3B

# INTRODUCTION

This chapter will explain about the purpose, overview and the scope of requirements specification

## Purpose of the document

This document explains the requirements for the parcel tracking system. It describes what the system must do, how it should behave, and the rules it must follow. The document is intended to guide developers, testers, and users, so that everyone clearly understands the objectives and expectations of the system. It also ensures that the system is built correctly, works as intended, and meets the needs of the users. Additionally, it provides a reference for future maintenance and improvements, helping the team keep the system consistent and reliable.

## Overview of the software system

The system is created to help manage and track parcels in a logistics company. Different users have different roles. Customers can check the status and location of their parcels. Branch staff and administrators can update parcel details, manage records, and monitor deliveries. Delivery employees can update parcel locations using barcode scanning or ask branch staff to update them if they do not have a mobile device. The system also keeps a history of parcels and provides notifications to make parcel management easier and faster.

## Scope of the requirements specification

This document lists all the features and rules for the parcel tracking system. It covers what the system must do, how it should perform, how secure it must be, and how it should look and work for users. It also explains limits of the system, such as needing an internet connection and hardware requirements. Optional features like customer account registration are included but not required. The scope helps everyone understand what is included in the system and what is not.

# FUNCTIONAL REQUIREMENTS

These describe what the system should do. They define the features and functions that users expect, such as parcel tracking, status updates, or generating reports.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority | Dependencies | Acceptance Criteria |
| FR-01 | Customer can track a parcel by entering a valid Tracking ID | **Critical** | None | Entering a valid Tracking ID displays parcel status, sender/receiver info, and current location on the map |
| FR-02 | Admin or courier can update parcel status using barcode scanning | **Critical** | Barcode scanning module operational | Scanning the parcel barcode updates its status and location in the database immediately |
| FR-03 | Branch managers can manually update parcel status if courier lacks a device | **High** | FR-02 | Status updates made by the branch manager reflect immediately in the system |
| FR-04 | Display parcel history including past locations and status updates | **Medium** | FR-01, FR-02 | Users can view a chronological list of all parcel updates |
| FR-05 | System differentiates user roles (Customer, Courier, Admin) | **High** | Authentication module | Each role has access only to functionalities allowed for that role |
| FR-06 | Map integration to show parcel’s current location | **Medium** | FR-01 | Map marker displays the parcel’s last recorded coordinates accurately |
| FR-07 (Optional) | Notifications for significant parcel status changes | **Low** | FR-02 | Users receive alerts when parcel moves to a new branch or is delivered |

# NON-FUNCTIONAL REQUIREMENTS

These describe how the system should perform rather than what it does. This includes performance, reliability, usability, security, and scalability. They ensure the system works efficiently, safely, and meets quality standards.

|  |  |  |
| --- | --- | --- |
| Category | Description | Applying in Parcel Tracking System |
| Performance | How fast and efficiently the system responds to user actions | Parcel status appears less than 3 seconds after entering Tracking ID |
| Usability | How easy and intuitive the system is for users | Dashboard and tracking page are clear, readable, and mobile-friendly |
| Reliability | Ability of the system to operate continuously without failure | System available 24/7 with minimal downtime |
| Security | Measures to protect data and control access | Only authorized users can update parcel status; data encrypted |
| Scalability | Ability to handle increased workload | System can handle multiple branches updating parcels simultaneously |
| Maintainability | Ease of updating, fixing bugs, or adding features | Developers can update features without affecting existing functionality |

# USE CASES

Use cases show how users or external systems interact with the system to achieve specific goals.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | Use Case Name | Description | Actors | Preconditions | Postconditions | Alternate Flows |
| UC-01 | Track parcel | Customer enters Tracking ID to view parcel status and location on map. | Customer | Customer has valid Tracking ID. | Parcel status and location displayed. | Invalid Tracking ID then Show error message. |
| UC-02 | Update Parcel Status | Delivery Employee scans parcel barcode to update its status in the system. | Courier | Parcel exists in database; employee authenticated. | Parcel status updated in database. | Barcode scan fails then Employee manually enters Tracking ID. |
| UC-03 | Multi-Branch Access | Branch Manager updates parcel status for parcels delivered at their branch. | Branch Manager | Parcel assigned to the branch. | Updated status reflected for all users. | Branch offline then Status update queued until internet restored. |
| UC-04 | Admin Dashboard Overview | Admin views all parcels, their current status, and branch distribution. | Admin | Admin logged in. | Dashboard displays current parcel data. | Data fetch error then show message and retry option. |
| UC-05 | Feedback Collection | Users submit feedback about parcel delivery. | Customer | Customer logged in or provided email. | Feedback stored in the system. | Feedback submission fails then Notify user and allow retry. |

# DATA REQUIREMENTS

This section details the information the system will use and store. It defines data entities, their attributes, and relationships between them, helping to design the database properly.

|  |  |  |
| --- | --- | --- |
| Data Entity | Attributes | Relationships |
| Branches | id, name, address, city, state, postal\_code, country, contact\_phone, email, is\_hub, is\_active, is\_deleted, latitude, longitude, created\_at, updated\_at | Parcels (origin\_branch\_id & destination\_branch\_id 1:N), Admins (1:N), Couriers (1:N) |
| Customers | id, name, email, phone, address, city, state, postal\_code, country, customer\_type, tax\_id, is\_corporate, is\_deleted, created\_at, updated\_at | Parcels (sender\_id & receiver\_id 1:N) |
| Admins | id, email, password\_hash, name, branch\_id, role, is\_active, is\_deleted, is\_loggedIn, last\_login, created\_at, updated\_at | Branches (N:1), Parcel Updates (optional admin\_notes link) |
| Couriers | id, private\_id, password\_hash, name, phone, email, branch\_id, vehicle\_type, license\_number, is\_active, is\_deleted, last\_login, created\_at, updated\_at | Branches (N:1), Parcel Updates (1:N), Parcels (current\_courier\_id N:1) |
| Parcels | id, tracking\_code, public\_id, secret\_id, sender\_id, receiver\_id, sender\_name, sender\_address, sender\_email, sender\_contact, receiver\_email, receiver\_contact, parcel\_name, latitude, longitude, weight, dimensions, parcel\_type, parcel\_description, declared\_value, insurance\_amount, status, origin\_branch\_id, destination\_branch\_id, current\_courier\_id, expected\_delivery\_date, expected\_delivery\_time, actual\_delivery\_date, special\_instructions, shipping\_cost, is\_deleted, created\_at, updated\_at | Sender & Receiver (N:1), Branches (origin & destination 1:N), Couriers (current\_courier\_id N:1), Parcel Updates (1:N) |
| Parcel Updates | id, parcel\_id, courier\_id, status, location, latitude, longitude, courier\_notes, admin\_notes, signature\_image\_url, created\_at | Parcels (N:1), Couriers (N:1) |

# ASSUMPTIONS AND CONSTRAINTS

Assumptions are conditions considered true for the project, such as users having internet access. Constraints are limits or restrictions, like requiring a barcode scanner or running on a live server.

## Assumptions

* Users (admins, couriers, and customers) have access to a stable internet connection.
* Barcode scanners and mobile devices used by couriers are functional and compatible with the system.
* Branch offices have devices capable of running a modern web browser (Chrome, Edge, or Firefox).
* Supabase cloud services are available and accessible for database operations.
* Admins and couriers have basic training to use the system effectively.

## Constraints

* Only one person is developing this project, which may affect development speed and testing coverage.
* The system must be accessed online; offline operation is not supported.
* Mobile testing for barcode scanning must be completed before using the hardware in real operations.
* The web application must be hosted on a live server for access by all branches.
* Browser compatibility: The system must function correctly on the latest versions of Chrome, Edge, and Firefox.
* Security constraints: Only authorized users can update parcel status or access sensitive information.

# GLOSSARY

A glossary explains key terms used in the document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Admin** | A Branch Manager with full access to manage parcels, branches, and system settings. |
| **Barcode Scanner** | A device used to scan parcel tracking IDs for status updates. |
| **Courier** | Delivery personnel responsible for picking up, transporting, and delivering parcels. |
| **Customer** | The sender or receiver of a parcel. |
| **Database (DB)** | A structured collection of data managed using Supabase in this system. |
| **Parcel** | Any item registered in the system for delivery. |
| **Tracking ID / Tracking Code** | |  | | --- | |  |   Unique identifier assigned to each parcel to monitor its status. |
| **Parcel Status** | |  | | --- | |  |   The current state of a parcel like, Pending, In Transit, Out for Delivery, Delivered. |
| **Parcel Update** | A record of changes in the parcel’s status or location. |
| **RLS (Row Level Security)** | A database feature that restricts access to certain rows based on user permissions. |
| **Supabase** | Backend-as-a-Service platform used for database, authentication, and APIs. |
| **Data Normalization** | Process of organizing database fields to reduce redundancy and improve data integrity. |
| **Encryption** | Security process that protects sensitive data, such as passwords, by converting it into a coded format. |
| **Single Active Session** | A security measure allowing only one active session per user at a time. |
| **Netlify** | Web hosting platform used to deploy the Parcel Tracking System. |
| **Authentication** | Process of verifying a user’s identity before granting access. |
| **CRUD Operations** | Create, Read, Update, delete – the basic operations for managing data. |
| **Password Hash** | A cryptographic representation of a user’s password, stored instead of the actual password for security purposes. |
| **Tailwind CSS** | Utility-first CSS framework used for styling and responsive design. |
| **Term** | **Definition** |
| **Cookie** | Small piece of data stored on the user's device to maintain session state or preferences. |
| **Token** | Unique identifier issued to authenticate a session or API request. |
| **UUID** | Universally Unique Identifier, a 128-bit value used to uniquely identify records, sessions, or objects across a system. |
| **Session** | A temporary interaction between a user and the system, often tied to a token or cookie to track active login activity. |
| **API** | Application Programming Interface, allowing communication between different software components or services. |
| **Admin Panel** | Web interface for administrators to manage parcels, branches, and system users. |
| **CDN (Content Delivery Network)** | A network of servers distributed globally that delivers web content (like images, CSS, and JavaScript) quickly to users based on their location. |
| **SSL (Secure Sockets Layer)** | A security protocol that encrypts data transmitted between a web browser and a server to protect sensitive information. |
| **Serverless** | A cloud computing model where developers run applications without managing servers. |
| **ERD (Entity-Relationship Diagram)** | A visual diagram that shows how database tables (entities) relate to each other, including keys and relationships. |
| **ENUM (Enumeration)** | A data type in programming that defines a set of named values, useful for standardizing choices like “Pending” or “Delivered” in a parcel system. |
| **SDK (Software Development Kit)** | A collection of tools, libraries, and documentation that developers use to create applications for a specific platform or service. |
| **JSON (JavaScript Object Notation)** | A lightweight data format used for storing and exchanging data between a server and a web application in a structured way. |
| **IDE (Integrated Development Environment)** | |  | | --- | |  |  |  | | --- | | A software application that provides tools like a code editor, debugger, and compiler to make software development easier. | |
| **DFD (Data Flow Diagram)** | A diagram that shows how data moves through a system, including inputs, processes, storage, and outputs. |

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| Revision Ver. | Date | Description of Changes |
| 0.5 | **August 26, 2025** | **Initial draft of the requirements specification document.** |