

Here's a summary and cheat sheet created from the provided Software Requirements Specification document, including information gleaned from the images:

## **\*\*Summary:\*\***

This document outlines the specifications for a "De-duplication of PL (Price List) Number" system developed using Laravel. It details the system's purpose, scope, functionalities, user characteristics, constraints, and specific requirements. The system aims to streamline the management of PL numbers within an organization by eliminating duplicates, providing a structured management approach, and offering CRUD operations. Key functions include adding, updating, deleting, detecting, and merging PL numbers, with a comprehensive list view. The intended users are internal staff involved in pricing and product catalog management. The system should be easy to use, maintain data integrity, and detect duplicates efficiently. The snapshots provided give a visual representation of the add PL Number function and the view duplicates function.

## **\*\*Cheat Sheet: PL Number De-duplication System\*\***

### **\*\*1. Introduction & Purpose\*\***

- \* **\*\*System Name:\*\*** PL Number De-duplication System
- \* **\*\*Purpose:\*\*** Efficiently manage Price List numbers by adding, updating, deleting, and merging duplicates.
- \* **\*\*Target Users:\*\*** Internal staff for pricing and product catalog management.

### **\*\*2. Scope & Objectives\*\***

- \* **\*\*Scope:\*\*** Standalone Laravel web application for internal use.
- \* **\*\*Objectives:\*\***
  - \* Ensure data consistency and prevent duplication.
  - \* Maintain a clean and optimized database.
  - \* Improve operational efficiency.

### **\*\*3. Definitions\*\***

Term	Definition
PL Number	Price List Number: A unique identifier representing a specific price list in the system
CRUD	Create, Read, Update, Delete: The four basic operations of persistent storage in databases.
UI	User Interface: The visual elements of the application through which the user interacts with the system.

Laravel	A PHP framework used for developing modern, scalable, and secure web applications.	
MVC	Model-View-Controller: A software architectural pattern used to separate the data (Model), UI (View), and control logic (Controller).	
Validation	The process of ensuring that input data meets defined rules and constraints before processing or storing it in the database.	
Timestamp	A record of the time when an entry was created or updated, typically used for tracking changes in the database.	
Merge	The process of combining two or more duplicate PL number entries into a single, consolidated record.	
Soft Delete	A feature where records are not permanently removed from the database but are marked as deleted, allowing for possible recovery.	

#### **\*\*4. System Functionalities\*\***

- \* **\*\*Add PL Numbers:\*\***
  - \* Input: PL number (required), description (optional).
  - \* Validation: Ensure uniqueness.
- \* **\*\*Update PL Numbers:\*\***
  - \* Input: Record ID, updated PL number/description.
  - \* Action: Update record, log timestamp.
- \* **\*\*Delete PL Numbers:\*\***
  - \* Action: Soft-delete (flag as inactive).
- \* **\*\*Detect Duplicates:\*\***
  - \* Mechanism: Exact match, configurable fuzzy logic (Levenshtein, phonetic).
- \* **\*\*Merge Duplicates:\*\***
  - \* Process: User-prompted field conflict resolution, archive/delete redundancies.
- \* **\*\*Audit Trail:\*\***
  - \* Log all critical operations (add, edit, delete, merge, login).

#### **\*\*5. External Interface Requirements\*\***

- \* **\*\*User Interface (UI):\*\***
  - \* Web-based, accessible on modern browsers.
  - \* Intuitive, responsive, and accessible.
  - \* Sections: Login/Authentication, Dashboard, Form for PL number creation/editing, Search/Filter view, Notification system.

#### **\*\*6. Non-Functional Requirements\*\***

- \* **\*\*Performance:\*\***
  - \* Support 50 concurrent users with <2 second response time.
  - \* Duplicate detection on datasets <10,000 records in under 10 seconds.
- \* **\*\*Reliability:\*\*** Graceful recovery from crashes, no data corruption.

- \* **Availability:** 99.5% uptime, notification of downtime exceeding 5 minutes.
- \* **Security:** Secure login (bcrypt), CSRF/XSS/SQL injection protection, session timeout.
- \* **Maintainability:** Laravel coding conventions, modular code, documentation.
- \* **Portability:** LAMP or LEMP stack.

#### **7. Design Constraints**

- \* Laravel framework only.
- \* UI: Laravel Blade (or Vue.js for interactive components).
- \* English language only.

#### **8. Data Flow Diagram (DFD)**

- \* Diagram depicts Users and Admin interacting with "PL Number Management System". The system stores to and retrieves from "PL Number Database".

#### **9. Github Link**

- \* The project repository is located at: ``https://github.com/1amSuraj/PL-Number-DeDuplication``