**Infosys Limited**

Package Tracking System

**Requirements Specification Document**

28-02-2025

**REVISION LIST**

| Ver.Rev | Date | Authors | Description |
| --- | --- | --- | --- |
| V1.0 | 28-02-2025 | Rajamanickam.n | High level user stories |

***TABLE OF CONTENTS***

1. PROJECT DESCRIPTION 3

2. USER STORIES 3

3. FRONTEND COMPONENTS 7

4. ENVIRONMENT REQUIREMENTS 11

# PROJECT DESCRIPTION

This internship project focuses on developing a comprehensive package tracking system that leverages distribution center data to monitor shipments throughout the delivery network. The system will optimize package handling, provide accurate delivery estimates.

# USER STORIES

|  |  |
| --- | --- |
| **US 01** | Priority: Must Have |
| **User Story Description** | |
| As a customer, I want to track my package by entering a tracking number so that I can see its current status and location. | |
| **Acceptance Criteria** | |
| * System accepts alphanumeric tracking numbers * Current package status is clearly displayed * Last known location is shown with timestamp * Estimated delivery date is provided * Package history shows all scan points | |
|
|
|
|
|

|  |  |
| --- | --- |
| **US 02** | Priority: Must Have |
| **User Story Description** | |
| As a distribution center manager, I want to see real-time package volumes and status distributions so that I can allocate resources effectively. | |
| **Acceptance Criteria** | |
| * Dashboard displays current package count by status * Filtering by package type, destination, and priority * Alerts for unusual volume spikes | |
|
|
|

|  |  |
| --- | --- |
| **US 03** | Priority: Must Have |
| **User Story Description** | |
| As a logistics coordinator, I want to flag delayed packages automatically so that I can prioritize them for expedited handling. | |
| **Acceptance Criteria** | |
| * System identifies packages that deviate from expected delivery timeline * Delay severity is categorized (minor, moderate, severe) * Delayed packages are highlighted in sorting queue | |
|
|
|

|  |  |
| --- | --- |
| **US 04** | Priority: Must Have |
| **User Story Description** | |
| As a distribution center supervisor, I want to track incoming package volumes so that I can plan staffing needs in advance. | |
| **Acceptance Criteria** | |
| * Forecast of incoming packages for next 24, 48, and 72 hours * Historical comparison to identify trends * Breakdown by package size, type, and destination * Alerts for unexpected volume changes | |
|
|
|

|  |  |
| --- | --- |
| **US 05** | Priority: Must Have |
| **User Story Description** | |
| As a sorting facility operator, I want the system to assign optimal routing paths for each package so that processing is efficient. | |
| **Acceptance Criteria** | |
| * Routing based on package destination, priority, and current distribution center capacity * Visual indicators for sorting staff showing correct bin/conveyor * Optimization for batching similar destinations * Real-time updates if routes need to change * Performance metrics tracked for route efficiency | |
|
|
|

|  |  |
| --- | --- |
| **US 06** | Priority: Must Have |
| **User Story Description** | |
| As a distribution center manager, I want to view center capacity metrics so that I can prevent bottlenecks. | |
| **Acceptance Criteria** | |
| * Real-time visualization of current capacity utilization * Breakdown by processing area (intake, sorting, outbound) * Threshold alerts when approaching capacity limits * Historical data comparison * Recommendations for load balancing between centers | |
|
|
|

|  |  |
| --- | --- |
| **US 07** | Priority: Must Have |
| **User Story Description** | |
| As a logistics planner, I want to see inter-center transfer schedules so that I can optimize transportation resources. | |
| **Acceptance Criteria** | |
| * Calendar view of all scheduled transfers * Package volume estimates for each transfer * Transportation capacity utilization metrics * Ability to adjust schedules based on volume or delays | |
|
|
|

|  |  |
| --- | --- |
| **US 08** | Priority: Must Have |
| **User Story Description** | |
| As a customer service representative, I want to look up detailed package information so that I can resolve customer inquiries effectively. | |
| **Acceptance Criteria** | |
| * Search by tracking number, customer name, or address * Complete tracking history with timestamps * Access to all customer communications about the package * Ability to add notes to package record * Options to initiate exception processes (investigation, refund, reroute) | |
|
|
|

|  |  |
| --- | --- |
| **US 09** | Priority: Must Have |
| **User Story Description** | |
| As a returns processor, I want to process and track returned packages so that they're properly accounted for in the system. | |
| **Acceptance Criteria** | |
| * Link to original outbound tracking information * Routing instructions for returned items * Metrics on return volumes and reasons | |
|
|
|

|  |  |
| --- | --- |
| **US 10** | Priority: Must Have |
| **User Story Description** | |
| As a quality assurance manager, I want to track damaged package incidents so that I can identify and address systematic issues. | |
| **Acceptance Criteria** | |
| * Record damaged package details * Classification of damage types and severity * Reporting on damage by route, handler, or packaging type * Trend analysis to identify problem areas | |
|
|
|

|  |  |
| --- | --- |
| **US 11** | Priority: Must Have |
| **User Story Description** | |
| As a distribution center supervisor, I want to handle exception cases for oversized or special handling packages so that they receive appropriate processing. | |
| **Acceptance Criteria** | |
| * Flagging of packages that exceed standard dimensions or weight * Special routing paths for non-standard items * Extra fee calculation for special handling cases * Performance tracking for special handling items | |
|
|
|

|  |  |
| --- | --- |
| **US 12** | Priority: Must Have |
| **User Story Description** | |
| As a security officer, I want to flag suspicious packages so that they can be diverted for additional screening. | |
| **Acceptance Criteria** | |
| * Criteria-based identification of packages requiring inspection * Secure holding area assignment * Chain of custody tracking | |
|
|
|

|  |  |
| --- | --- |
| **US 13** | Priority: Must Have |
| **User Story Description** | |
| As an operations manager, I want to generate performance reports so that I can measure KPIs across distribution centers. | |
| **Acceptance Criteria** | |
| * Reports on processing times, accuracy, and volume * Comparison across centers and against targets * Customizable reporting periods * Scheduled report generation and distribution | |
|
|
|

|  |  |
| --- | --- |
| **US 13** | Priority: Must Have |
| **User Story Description** | |
| As an operations manager, I want to generate performance reports so that I can measure KPIs across distribution centers. | |
| **Acceptance Criteria** | |
| * Reports on processing times, accuracy, and volume * Comparison across centers and against targets * Customizable reporting periods * Scheduled report generation and distribution | |
|
|
|

|  |  |
| --- | --- |
| **US 14** | Priority: Must Have |
| **User Story Description** | |
| As a customer, I want to report a problem with my delivery so that it can be investigated and resolved. | |
| **Acceptance Criteria** | |
| * Issue reporting form accessible from tracking page * Selection of common issue types * Auto-creation of case in support system * Acknowledgment and case number provided immediately | |
|
|
|

|  |  |
| --- | --- |
| **US 15** | Priority: Must Have |
| **User Story Description** | |
| As a finance manager, I want the system to calculate shipping costs based on package characteristics and routing so that billing is accurate and transparent. | |
| **Acceptance Criteria** | |
| * Automatic fee calculation based on weight, dimensions, distance, and service level * Support for surcharges and special handling fees * Detailed cost breakdown available to authorized users * Bulk pricing for high-volume customers | |
|
|
|

# FRONTEND COMPONENTS

**Customer Tracking Portal**

Description: A user-friendly interface for customers to track their packages by entering tracking numbers.

Features:

- Tracking number input field with validation for alphanumeric formats

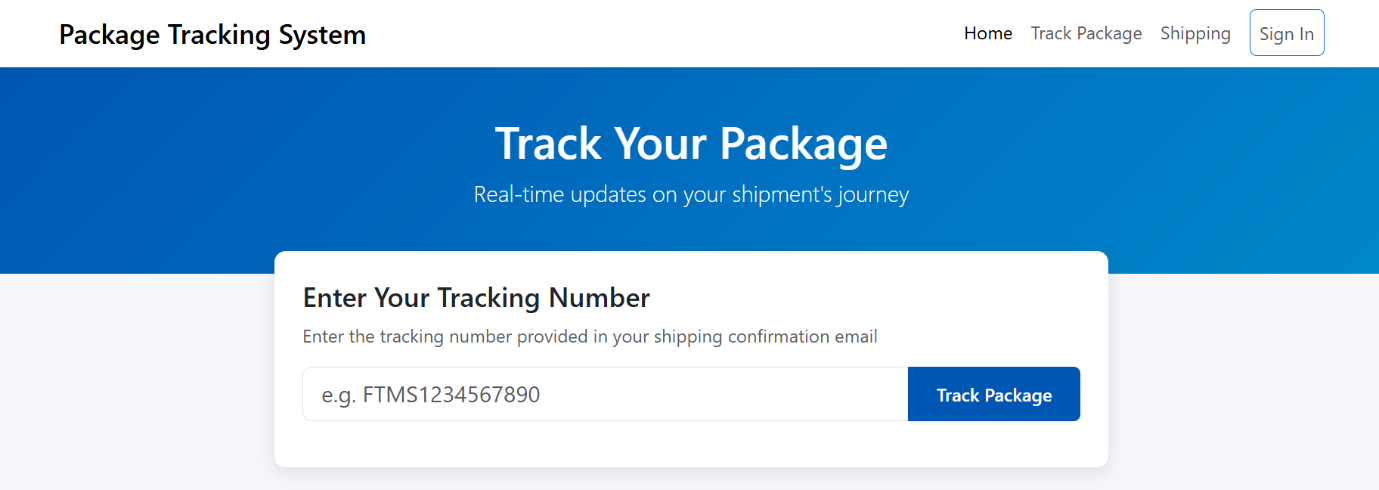
- Package status card displaying:

- Current status with clear visual indicators

- Last known location with timestamp

- Estimated delivery date

- Interactive timeline showing complete package history



**Exception Management Console**

Description: Specialized interface for handling delayed, damaged, oversized, or suspicious packages.

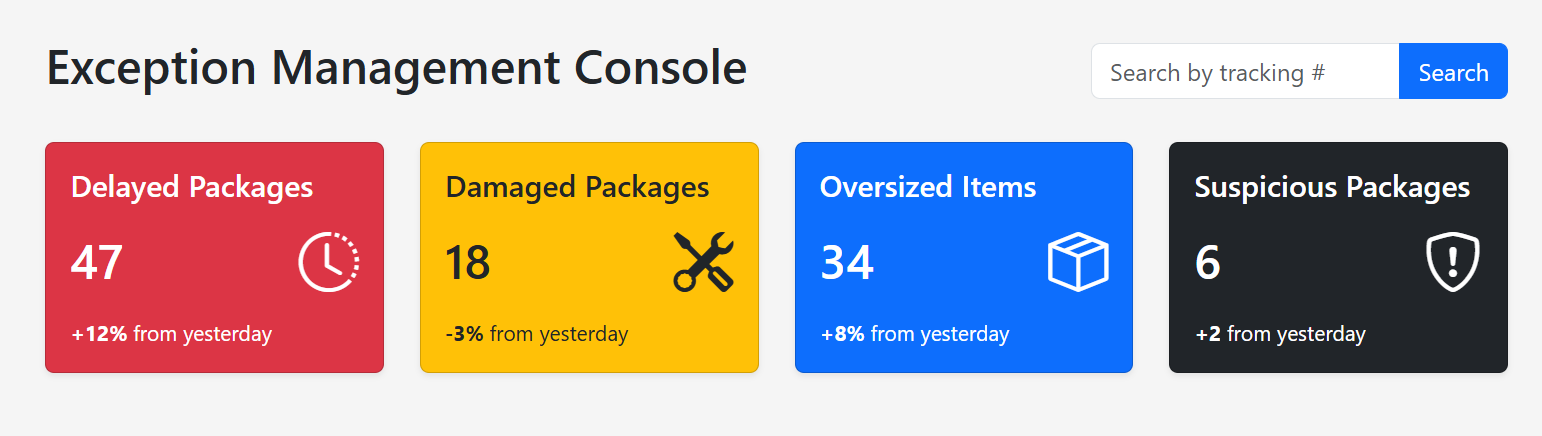
Features:

- Delayed package queue with severity indicators

- Damage reporting form with classification system

- Special handling workflow for oversized items

- Suspicious package flagging and inspection tracking



**Customer Service Representative Portal**

Description: Comprehensive information access tool for CSRs to quickly resolve customer inquiries.

Features:

- Multi-parameter search (tracking number, customer name, address)

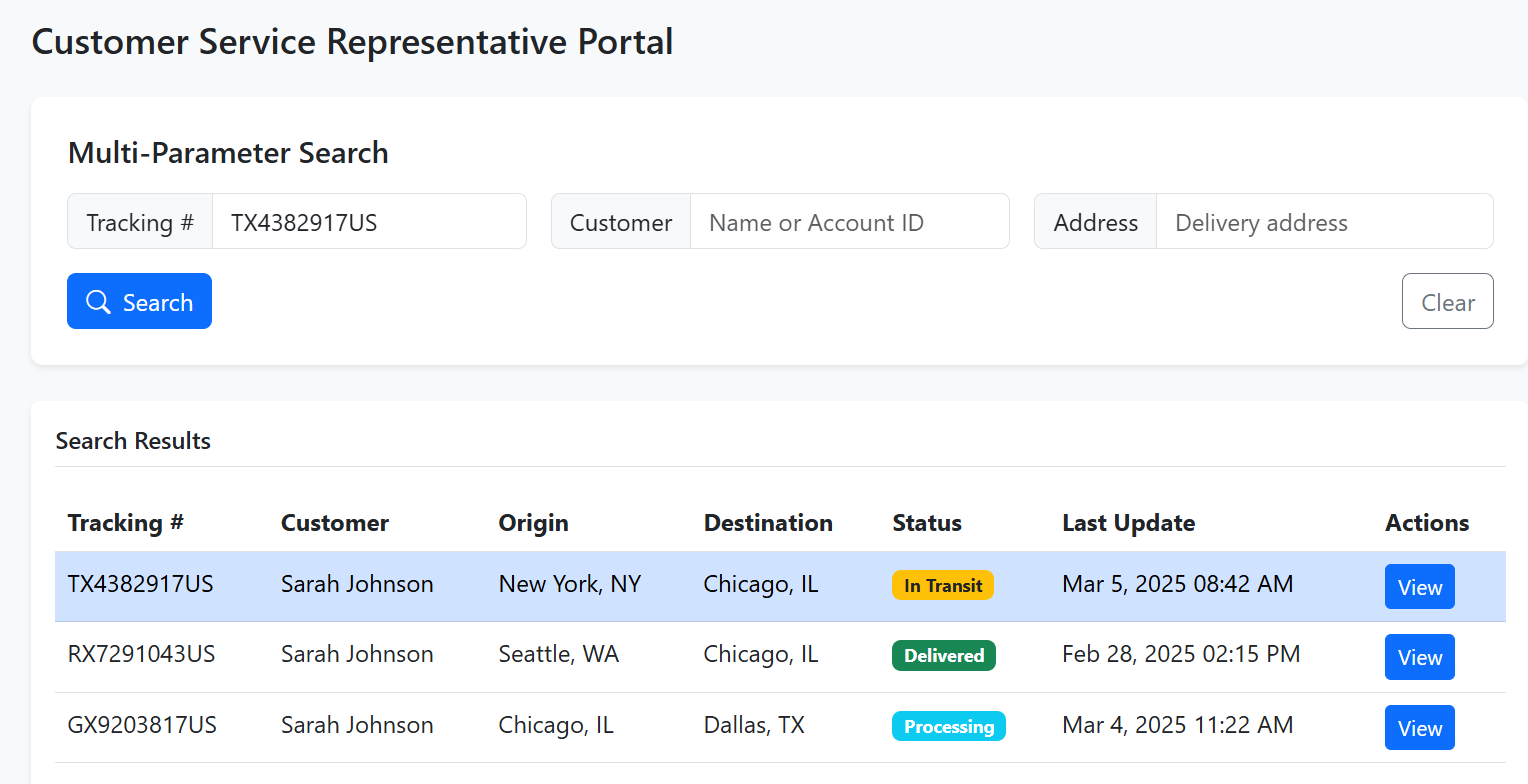
- Complete package information display

- Timeline visualization of all tracking events

- Customer communication history related to package

- Case management for investigations and resolutions

- Quick action buttons (initiate refund, reroute package, escalate)



**Returns Processing Station**

Description: Specialized interface for handling and tracking returned packages.

Features:

- Return authorization entry and validation

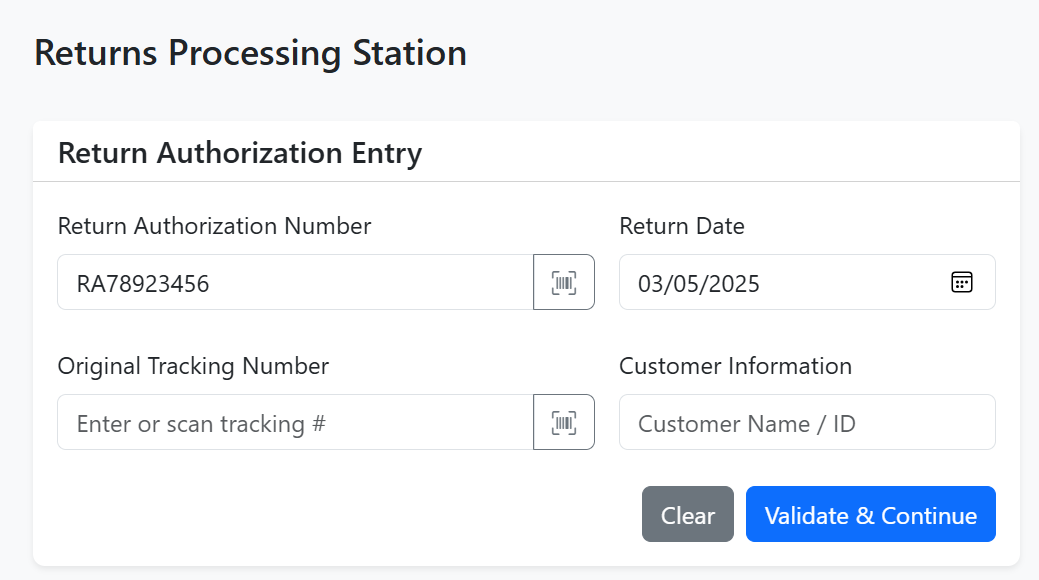
- Original order lookup and linkage

- Return reason categorization

- Disposition decision support

- Returns metrics dashboard

- Automated routing instructions for returned items



**Financial Management Interface**

Description: Cost calculation and billing management system for shipping operations.

Features:

- Shipping cost calculator based on package characteristics

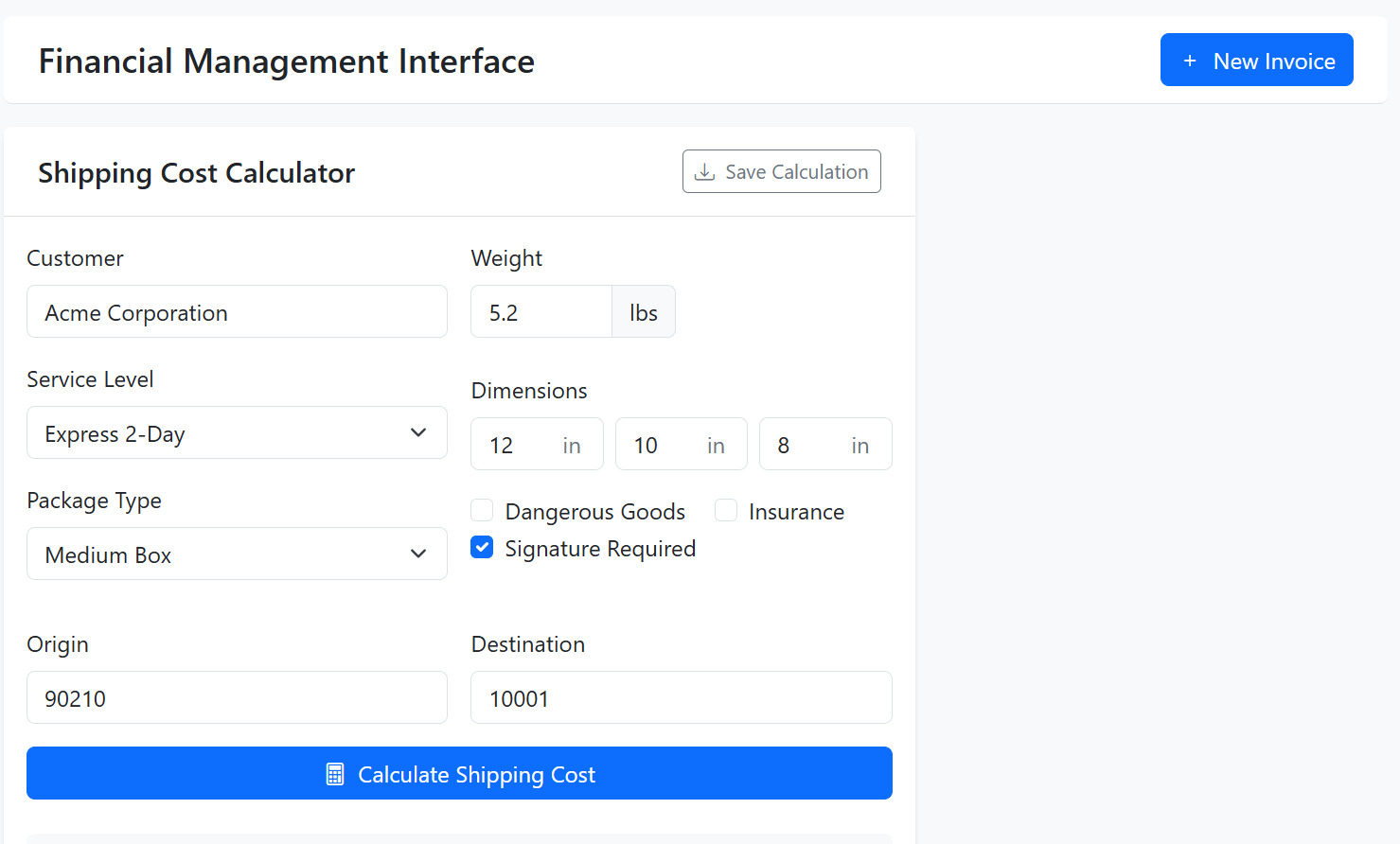
- Surcharge and special handling fee management

- Detailed cost breakdown visualizations

- Bulk pricing configuration for high-volume customers

- Billing history and report generation

- Invoice creation and management



**Reporting & Analytics Hub**

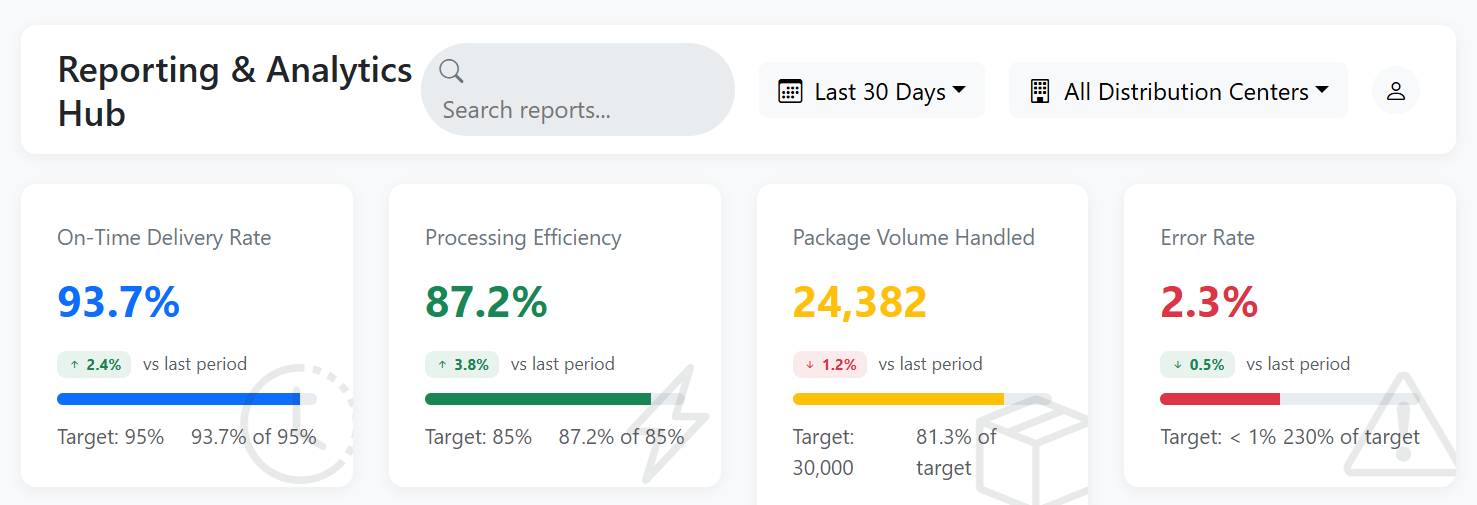
Description: Comprehensive reporting tool for operations managers to track KPIs and performance metrics.

Features:

- Scheduled report generation and distribution

- Trend analysis and forecasting tools

- Performance comparisons against targets



# ENVIRONMENT REQUIREMENTS

**Technology Stack:**

* **Backend:** Spring Boot, Spring Data, Spring REST, Spring Cloud Consul
* **Frontend**: Angular/React, Typescript/JavaScript HTML5, CSS3, Bootstrap
* **Database**: MySQL/NOSQL
* **Security** - OAuth/Spring Security

**Non-Functional Requirements:**

* Performance: The application should be fast and responsive, with quick load times and minimal lag or delay.
* Scalability: The application should be able to handle large amounts of traffic and scale as needed to accommodate growth.
* Security: The application should be secure and protect user data from unauthorized access, with measures such as encryption and secure authentication. Secure password storage using hashing. Protect against common web vulnerabilities (e.g., SQL injection, XSS).
* Reliability: The application should be reliable and available, with minimal downtime or outages.
* Usability: The application should be easy to use and intuitive, with a clear and consistent user interface.
* Maintainability: The application should be easy to maintain and update, with clear and well-organized code that is easy to understand and modify.
* Compatibility: The application should be compatible with a wide range of devices and browsers, with support for different screen sizes and resolutions.
* Implement caching mechanisms to improve performance.
* Session Management: Implement secure session management techniques to prevent unauthorized access. Consider using short-lived sessions with automatic timeouts.
* Lazy Loading: Implement lazy loading techniques for images and other content, ensuring faster initial page loads and smoother user experience.
* User Interface (UI) and User Experience (UX) Design: Design an intuitive user interface that is easy to navigate and provides a seamless user experience. Prioritize responsiveness and clarity across all devices (desktop, mobile, tablets).

**Final Deliverables:**

* Application archive (.jar ) with source code
* Database DDL Script
* Complete Source code
* Sample screenshots of important screens

**Implementation Guidelines:**

* Utilize Lombok for model class creation and logging.
* Implement logic using Lambdas and Streams.
* Organize code into logical layers (controller, service, DTO, entity, etc).
* Use meaningful package names reflecting the domain.
* Employ Spring dependency injection for effective component management.

**General Guidelines:**  
**Error Handling:**

* Handle cases where the user attempts to access without authorization.
* Implement a custom exception to throw user-defined messages.
* Centralized exception handling mechanism should be used to capture exceptions and translate them into HTTP response.
* Implement proper error handling mechanisms to catch any issues and log all the service exceptions using LoggingAspect
* Handle success and error responses appropriately with proper messages.

**DTOs and Mapping:**

* Data Transfer Objects (DTOs) should be used for API request and response.
* Entities should not be exposed directly to APIs.
* Use ModelMapper to convert from entities to dto and vice versa.

**Database Interaction:**

* Spring Data Repository should be used for database operation.
* Add appropriate properties specific to the application like database properties and table generation strategies.

**API Design:**

* All the api should be mapped with the base URI
* Set server port number of your choice.
* Use appropriate HTTP methods and status code for all rest end points.
* Use Swagger for generating the API documentation.

**Microservice Communication:**

* Circuit Breaker pattern should be implemented for all the critical services with a timeout of 3 seconds and error threshold of 50%. The circuit breaker should open after 10 consecutive failures and remain open for 60 seconds. Upon circuit breaker opening, a fallback mechanism should return a generic message to the client.
* Fallback behaviors should be defined for all critical operations.
* Create two instances for any Microservice and implement load balancing.

**Validation:**

* Bean validation should be used for all inputs throughout the process.
* Custom Validators should be used for complex validation.
* All request parameters must be validated for null or empty values. For any such invalid values, ‘Please provide a valid <attribute name>’ should be the error message.
* Any date and time value should not start with zero.

**Testing:**

* Write JUnit test cases for all service methods using Mockito.
* 80% code coverage should be achieved.

**Code Quality:** To ensure adherence to coding standards, the project should be analyzed using SonarQube. The following minimum acceptable values must be met:

|  |  |
| --- | --- |
| **SonarQube Metrics** | **Minimum Acceptable Value** |
| Security | A |
| Reliability | A |
| Issues | <= 5 |
| Coverage | >= 80% |
| Duplications | <=3% |
| Security Hotspots | A |

**Postman:** Once done with implemen­­ting the requirements, use Postman or Swagger to test whether the REST endpoints are working fine