Software Requirements Specification

for

Go-VYA

Version 1.0

Prepared by

Ashwin Mittal-220953128

Yogesh Prashant Rane-220953290

Vansh yadav-220953306

Manipal Institute of Technology, Manipal

2 March 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

*The objective of the **"Cargo Management Software"** is to act as an intermediary between the various customers in need of moving cargo of any size and the Packing and Moving companies .This SRS describes the initial release of this software(Version 1.0).The scope of this SRS covers the full functionality of our project.*

1.2 Document Conventions

Times New Roman is the font used throughout the document, with headings in Bold. Sections and Subsections are numbered to give a clear way of understanding the project's structure and development. These conventions enhance readability, organization, and clarity.

1.3 Intended Audience and Reading Suggestions

The SRS is intended for getting a comprehensive idea about the functionality of the software for consumers and developers alike. Special care has been taken to simplify the working of the various subsystems for ease in grading and gauging quality of the project by our respective examiners.

1.4 Product Scope

The Cargo Management Software streamlines cargo transportation by connecting customers with packing and moving companies. Its goals include simplifying the booking process for customers, expanding business opportunities for providers, and ensuring efficient system management. By achieving these objectives, the software aligns with corporate goals of enhancing customer satisfaction and operational efficiency in the cargo management industry.

2. Overall Description

2.1 Product Perspective

The Cargo Management Software specified in this SRS is an independent solution created to streamline cargo transportation processes. It is not part of a product family or a replacement for existing systems. Instead, it functions as a self-contained product, facilitating interactions between customers and packing/moving companies. This software operates autonomously, without relying on external systems or components.

2.2 Product Functions

The Cargo Management Services project comprises key system features to facilitate the interaction between users and manage cargo movement efficiently. The system begins with a robust user authentication and authorization mechanism, allowing users to securely register, log in, and access the platform based on their roles—whether as Shipment Requesters, Providers, or Administrators. Shipment Requesters can initiate new cargo movement orders by specifying details such as source, destination, weight, and delivery speed. Shipment Providers, in turn, can efficiently manage incoming orders, accepting and providing quotes. Admins have access to oversee and manage system entries, validate users, and monitor aggregate user statistics. Additionally, a profile update feature is implemented for both Shipment Requesters and Providers, allowing them to maintain accurate information such as name, phone number, and location. These collectively ensure a seamless and secure environment for cargo management services.

2.3 User Classes and Characteristics

We anticipate 2 major user classes for this software , namely end consumers paying for the services and major businesses providing the services. A broad classification may be made between individual customers and other businesses themselves availing the service among the consumers. The major business may also have among them freelancers who provide transport services on a smaller scale. The functionality aims to satisfy end consumers who tend to make occasional requests for sundry transport which would provide the base for customer acquisition.

2.4 Operating Environment

The database application will be set up on an SQL server. It will be accessed using a React.js application as front end. The react application will implement a HTML website and CSS and Tailwind for aesthetics and utilize JavaScript to handle requests. Node.js/Django will be implemented for the API endpoints to connect to the backend SQL server using the Node.js/Django SQL library.

2.5 Design and Implementation Constraints

While Industry standards and software design conventions have been followed closely for this project, complete functionality and deployment will be taken up in the later versions. Deployment of the website and database along with debugging support have not been included in this iteration. Consumers and services alike will have an encryption-less login, which will be improved further. The guidelines explicitly dictate usage of MySQL database systems, therefore NoSQL systems such as MongoDB haven't been explored.

2.6 User Documentation

A standard documentation underlining usage of the MySQL database from their official website along with a comprehensive GitHub repository will be provided to help users with the functionality of the website.

2.7 Assumptions and Dependencies

The project is designed using modern industry level applications which are tried and tested and regularly maintained, hence we do not take into account faults on a fundamental level to affect the project. The linking of the various platforms, however, are subject to constant updates in terms of libraries and functions used, under whose scrutiny our projects bugs and effectiveness depends on. Other miscellaneous details such as presence of tools and basic dependencies on the client's end may also have an effect on the user experience.

3. External Interface Requirements

3.1 User Interfaces

The user interface is designed with a landing page providing access to login and register pages. After login, Requesters can efficiently place orders, view order history, check statuses, and update profiles. Providers, on the other hand, manage incoming orders, update statuses, create quotes, and adjust estimated multipliers. Admins access a dashboard to view all database entries, primarily focusing on validating new registrations. Standard GUI elements ensure consistency, error messages follow conventions, and a separate specification details the comprehensive user interface design for a streamlined experience.

3.2 Hardware Interfaces

The website is accessible and compatible with various hardware devices, including desktop computers, laptops, tablets, and smartphones to access the interface across all the devices

3.3 Software Interfaces

***Database System**: MySQL (version 8.0.3): The website utilizes MySQL as its primary database system to store user profiles, orders etc.*

***Backend Framework**: The website utilizes Django/Node.js as the primary backend framework, providing a robust and scalable platform for developing web applications.*

***Frontend Framework** React.js (version 18) + Vite : The user interface is developed using React.js for a dynamic webpage and the design is done through Tailwind CSS.*

***Data Items**: User profiles, Business details,order requests,orders placed,validation requests etc.*

3.4 Communications Interfaces

***Communication Protocols**: The application uses standard HTTP/HTTPS protocols for communication between the front-end and back-end components.*

***Security**: Communication is secured with use of JWT or CSRF tokens to ensure the confidentiality and integrity of data during transmission.*

***Error Messages**: Clear and concise error messages are displayed to users in case of communication failures or other issues, aiding in troubleshooting.*

***Synchronization Mechanisms**: Real-time updates and synchronization are implemented to ensure consistent data across all connected devices.*

4. System Features

4.1 User Authentication

4.1.1 Description and Priority

The User Authentication feature is designed to enable secure user access to the Cargo Management Services platform. It is of high priority as it forms the foundation for personalized user interactions.

4.1.2 Stimulus/Response Sequences

- *Stimulus: User clicks on the "Sign Up" button.*
  Response: System prompts the user to provide valid details and password for account creation. Users can decide the type of account they want to register as and need fulfill appropriate form fields which include:
  - *Name of Logistics Provider*
  - *Address of Logistics Provider*
  - *GSTIN and Pan Number*
  - *Contact Details:Name,Phone,Mail*
  - *Confirmation/ Undertaking by the Logistics Provider that they have the necessary documents as required by the Government of India Vehicle Registration, Route Permits, Vehicle Fitness Certificates etc.*

  Appropriate User Ids will be auto generated and allotted to the user.
- *Stimulus: User enters login credentials.*
  Response: System verifies credentials; successful login redirects the user to their respective dashboards based on their user-ids determined by roles.

4.1.3 Functional Requirements

REQ-1: Users can only register using valid information in the form fields.
REQ-2: Passwords must be securely hashed and stored.
REQ-3: Successful login redirects users to role-specific dashboards.
REQ-4: Appropriate prompts and error messages will be provided while filling in registration forms and well as on the login page.

4.2 Cargo Movement Order Placement

4.2.1 Description and Priority

The Cargo Movement Order Placement feature allows Shipment Requesters to seamlessly initiate new cargo movement orders. This feature holds high priority due to its central role in facilitating user interactions.

4.2.2 Stimulus/Response Sequences

- *Stimulus: Requester clicks "Place Order."*
  Response: System prompts the user to input order details such as source, destination, weight, and delivery speed, and sends this to the providers.

- *Stimulus: Provider clicks on "View Requests"*
  Response: Provider receives a list of all the orders requested and the provider can input their price and estimated delivery time.

- *Stimulus: Requester selects a shipment provider.*
  Response: System displays responses and quotes from the chosen provider and notifies the providers with order details. Customers can wait and accept the best provided offer.

4.2.3 Functional Requirements

REQ-5: Requesters provide order details including source, destination, weight, and delivery speed.
REQ-6: The system presents a list of available shipment providers with estimated rates and an option to notify specific providers about the order.
REQ-7: Requesters can confirm and finalize orders.
REQ-8: Providers can reply back to the request regarding any other specifics if required, upon which requesters can make their final choice.

4.3 Shipment Provider Order Management

4.3.1 Description and Priority
The Shipment Provider Order Management feature empowers providers to efficiently handle incoming cargo movement orders. Given its significance, this feature is classified as medium priority.

4.3.2 Stimulus/Response Sequences
Stimulus: Shipment Provider logs in and navigates to order management.
Response: System prompts the provider to update the order status.

Stimulus: Requester navigates to the Order status page.
Response: Order status as entered by the Provider is displayed.

4.3.3 Functional Requirements

REQ-9: Shipment Providers can view incoming orders.
REQ-10:Providers can accept or reject orders based on availability.

REQ-11:Providers provide quotes for accepted orders.

4.4 Admin Access

4.4.1 Description and Priority

The Admin Access feature provides administrative users with the capability to oversee and manage system entries, validate users, and access aggregate user statistics. This feature holds high priority for maintaining system integrity and monitoring user activities.

4.4.2 Stimulus/Response Sequences

Stimulus: Admin logs in using credentials.
Response: System validates admin credentials and redirects to the admin dashboard.

Stimulus: Admin navigates to view all entries.
Response: System displays a comprehensive list of all system entries.

Stimulus: Admin validates user accounts.
Response: System updates user account status and notifies the affected users.

Stimulus: Admin views total users present.
Response: System provides real-time statistics on the total number of users.

4.4.3 Functional Requirements

REQ-11: Admins log in using valid credentials.
REQ-12: Admins are redirected to an admin-specific dashboard upon successful. login.
REQ-13: Admins can view and manage all system entries.
REQ-14: Admins can validate user accounts, updating their status.
REQ-15: Admins can access real-time statistics on the total number of users.

4.5 Shipment Provider Order Management

4.5.1 Description and Priority

The Profile Update feature allows both Shipment Requesters and Providers to update their profile details. Low priority is assigned to this feature.

4.5.2 Stimulus/Response Sequences

Stimulus: User clicks on "Update Profile" in the dashboard.
Response: System presents a form for updating user information.

Stimulus: User updates profile details.
Response: System validates and updates the user profile information.

4.5.3 Functional Requirements

REQ-16: Users can access the "Update Profile" feature from their dashboards.

REQ-17: Users provide updated information such as name, phone number, and location.

REQ-18: Providers can update multipliers for estimated prices.

REQ-19: System validates and securely stores updated profile information.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response Time*: The website's endpoints must respond to requests within a short duration on average, around 300 ms under normal operating conditions. Quick response times are required for maintaining interactivity in the web application, so that users have a smooth experience with the platform.*

Error Handling*: The platform shall handle errors smoothly to minimize disruptions. Poor handling of errors causes inconveniences while using the application and may even lead to security issues.*

5.2 Safety Requirements

Actions Prevented*: Unauthorized access to user data, data breaches, unauthorized sharing of user information with third parties.*

Data Encryption: *Rationale: To protect sensitive user and business data during transmission.*

5.3 Security Requirements

Secure Communication*: Use secure communication protocols such as HTTPS to encrypt data transmitted between the app and its servers. This helps prevent eavesdropping and man-in-the-middle attacks*

User Authentication*: Implement secure user authentication mechanisms to ensure that only authorized users can access the app.*

Prevent Cyber Attacks: *Use strict measures to prevent unauthorized access using attacks like SQL Injection, XSS Attack etc.*

5.4 Software Quality Attributes

***Usability**: Users should be able to navigate the website easily and perform tasks intuitively. This could be measured by the average time it takes for a user to complete common tasks like creating a request, checking status etc.*

***Security**: Users' personal data should be protected from unauthorized access or breaches .*

***Maintainability**: The codebase should be well-organized and documented, making it easier for developers to make updates or fix bugs.*

5.5 Business Rules

Order Placement:

Rule: Only registered users (Requesters) can place cargo movement orders.

Implication: User authentication and authorization mechanisms must be in place.

Provider Quote Submission:

Rule: Shipment Providers must submit quotes within 24 hours of receiving an order.

Implication: System should include automated reminders for Providers to submit timely quotes.