Delivery Service

WEB APPLICATION

SOFTWARE REQUIREMENTS SPECIFICATIONS

Contents

1. Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Delivery Service Web Application (DSWA). This SRS will allow for a complete understanding of what is to be expected of the DSWA to be constructed. The clear understanding of the DSWA and its functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the DSWA can be designed, constructed, and finally tested.

This SRS will be used by the software engineers constructing the DSWA and the delivery service end users. The software engineers will use the SRS to fully understand the expectations of this DSWA to construct the appropriate software. The delivery service end users will be able to use this SRS as a "test" to see if the software engineers will be constructing the system to their expectations. If it is not to their expectations the end users can specify how it is not to their liking and the software engineers will change the SRS to fit the end users' needs.

1.2 Scope

Need to mention

1.3 Definitions, Acronyms, and Abbreviations

SRS – Software Requirements Specifications

DSWA – Delivery Service Web Application

Subjective satisfaction – The overall satisfaction of the system

End Users – The people who will be using the system

1.4 Overview

The SRS is organized into two main sections. The first is The Overall Description and the second is the Specific Requirements. The Overall Description will describe the requirements of the DSWA from a general high-level perspective. The Specific Requirements section will describe in detail the requirements of the system.

2. The Overall Description

2.1 Product Perspective

The Delivery Service Web Application (DSWA) as a separate independent system. It is fully self-contained.

2.1.1 Hardware Interfaces

The DSWA will be hosted on regular servers and can be accessed via common web browsers from devices such as desktops, laptops or tablets/ smartphones. The system will interface to simple hardware like keyboards and mice monitors.

2.1.2 Software Interfaces

Need to mention

2.2 Product Functions

Need to mention

2.3 User Characteristics

- Educational Level: Low. But user should be capable of handling web applications

- Experience: No special training is needed for the basic operations, but some training may be required for the administrative tasks.

- Technical Expertise: Low; this is because the system is developed in a manner that can easily be understood and used by any lay person.

2.4 Apportioning of Requirements

Audio and video, other options can be considered as insignificant right now and thus are likely to be implemented later along with improvements.

2.4 Assumptions and Dependencies

Assumptions

- User Connectivity: Assumes customers and drivers have reliable internet access for using the web application.

- Device Compatibility: Assumes compatibility across various devices (e.g., smartphones, tablets, desktops) for seamless user experience.

- Data Security: Assumes implementation of robust security measures (e.g., HTTPS, encryption) to protect user data and maintain privacy.

- User Competence: Assumes users (customers, drivers, and administrators) are competent in using basic digital tools required for registration, delivery requests, and accessing features.

- Delivery Service Provider Availability: Assumes delivery service providers maintain accurate availability information for parcel tracking.

Dependencies

- Email Services: Dependency on reliable email services for account verification and password recovery functionalities.

- Internet Service Providers: Dependency on ISPs for stable internet connectivity for both users and administrators.

- Database Management: Dependency on efficient database management systems for storing and retrieving user information, delivery request details, and feedback.

3. Specific Requirements

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

3.1 External Interfaces

The E-Channeling Web Application will use standard input/output devices typically available on personal computers and mobile devices. This includes the following:

- Keyboard
- Mouse
- Monitor
- Touchscreen (for mobile devices)

3.1.1 User Interfaces

The User Interface Screens are described in table 1.

Table 1. Delivery Service Web Application User Interfaces Screens

Screen Name	Description
Login	Log into the system as a customer, driver, admin, and super admin
User Registration	Create new account as a customer, driver, admin, and super admin
Password Recovery	Recover or reset account password
Delivery Request	Request a parcel delivery
Track Parcel	Track parcels
Feedback	Give feedbacks
User Profile Management	View and update personal information
Driver Dashboard	Manage delivery request updates
Admin Dashboard	Manage customers, drivers, delivery requests

3.1.2 Software Interfaces

The system shall interface with a MySQL database for storing user information, appointment details, and schedules.

3.1.3 Hardware Interfaces

The system shall run on various platforms, including personal computers and mobile devices, and be accessible through modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

3.1.4 Communication Interfaces

The system shall support internet-based communication for functionalities like email verification, password recovery, etc. It will use standard protocols such as HTTP/HTTPS for data transmission

3.2 Functional Requirements

Functional requirements define the fundamental actions that a system must perform. The functional requirements for the system are divided into three main categories, Customers, Drivers, and Management/System Admin. For further details, refer to the use cases.

3.3 Non-Functional Requirements

Non-Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

3.3.1 Performance Requirements

Ensure the application responds quickly to user interactions and handles requests efficiently.

3.3.2 Logical Database Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

Need to mention

3.3.3 Design Constraints

The Delivery Service Web Application shall be a stand-alone system running in a Windows environment. The system shall be developed using Java and a MySQL database.

3.3.4 Standards Compliance

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

3.3.5 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

3.3.6 Availability

The system shall be available during 24 hours.

3.3.7 Security

Need to mention

3.3.8 Maintainability

The Delivery Service Web Application is being developed in PHP (Laravel). PHP is an object-oriented programming language and shall be easy to maintain.

3.3.9 Portability

Need to mention

3.3.10 Accessibility

Ensure the application is accessible to users with disabilities, complying with relevant accessibility standards.

3.3.11 Compatibility

Ensure the application works seamlessly across different devices, browsers, and operating systems.

3.3.12 Usability

Provide an intuitive and user-friendly interface that simplifies the requesting and tracking process for all users.

4. Change Management Process

Changes to this document require approval from the project manager and the client.

5. Document Approvals

5.1 Team One Approval

________________ ________________

Client Date

5.2 Team Two Approval

________________ ________________

Backend Developer Date

________________ ________________

Frontend Developer Date

6. Supporting Information

A system context diagram as well as use cases and use case descriptions that have been developed is as follows.

6.1 System Context Diagram

Below shows the system context diagram created for the DSWA

Need to mention

6.2 Use cases of DSWA

6.3 Use case diagram