### **Software Requirements Specification (SRS)**

#### **for**

**Road Master - Driving Learning Web System**

#### **Version 1.0**

**Prepared by:** Muhammad Abbas Ahmed Naveed

**Organization:** NUCES FAST Islamabad

**Date Created:** 23-03-2025

### **Table of Contents**

1. **Introduction** 1.1 Purpose  
    1.2 Document Conventions  
    1.3 Intended Audience and Reading Suggestions  
    1.4 Product Scope  
    1.5 References
2. **Overall Description** 2.1 Product Perspective  
    2.2 Product Features  
    2.3 User Classes and Characteristics  
    2.4 Operating Environment  
    2.5 Design and Implementation Constraints  
    2.6 User Documentation  
    2.7 Assumptions and Dependencies
3. **System Features**
4. **External Interface Requirements** 4.1 User Interfaces Overview  
    4.2 Hardware Interfaces  
    4.3 Software Interfaces
5. **System Features/Modules**
6. **Nonfunctional Requirements** 6.1 Performance  
    6.2 Security

## **1. Introduction**

### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the **Road Master - Driving Learning Web System**. The system is designed to facilitate the registration of learners and instructors, booking of driving sessions, and management of payments.

### **1.2 Document Conventions**

The following conventions are used in this document:

* Requirements are specified using **"shall"** statements.
* Non-functional requirements are identified by the prefix **NF-**.
* Functional requirements are identified by the prefix **REQ-**.

### **1.3 Intended Audience and Reading Suggestions**

This document is intended for:

* **Developers:** To understand the functional and non-functional requirements.
* **Project Managers:** To plan and track project progress.
* **Testers:** To create test cases based on the requirements.
* **Users:** To understand the system functionalities.

### **1.4 Product Scope**

Road Master is an online driving learning system that provides:

* Registration for learners and instructors.
* Booking and scheduling of driving sessions.
* Processing and management of payments.

### **1.5 References**

* Assignment Document
* Previous Sprint Backlog
* Sequence Diagrams
* Class Diagram
* User Stories

## **2. Overall Description**

### **2.1 Product Perspective**

Road Master is a new web-based platform aimed at streamlining the process of learning how to drive. It allows learners to book sessions with instructors and make payments online.

### **2.2 Product Features**

* User registration and authentication
* Instructor management
* Booking driving lessons
* Payment processing
* Progress tracking
* Report generation

### **2.3 User Classes and Characteristics**

* **Learner:** Registers, books sessions, and tracks progress.
* **Instructor:** Registers and manages driving sessions.
* **Admin:** Manages users, sessions, and payments.

### **2.4 Operating Environment**

* **Frontend:** ReactJS
* **Backend:** Spring Boot
* **Database:** PostgreSQL
* **Operating System:** Windows, Linux, MacOS
* **Browser Support:** Chrome, Firefox, Edge

### **2.5 Design and Implementation Constraints**

* Must use **React** for the frontend and **Spring Boot** for the backend.
* Database must be **PostgreSQL**.
* Must be accessible on both desktop.

### **2.6 User Documentation**

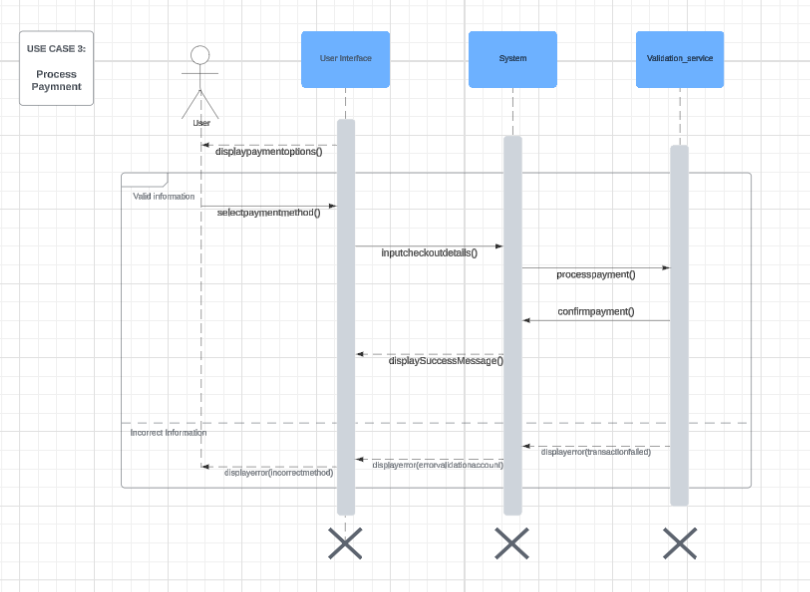
* User Manual
* Online Help and Tutorials
* Administrator Guide

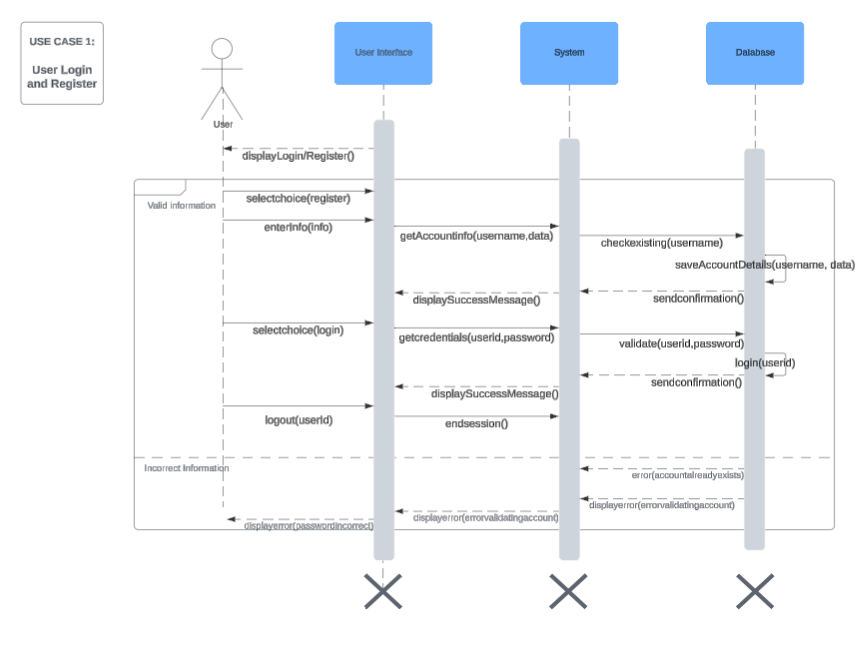
### **2.7 Assumptions and Dependencies**

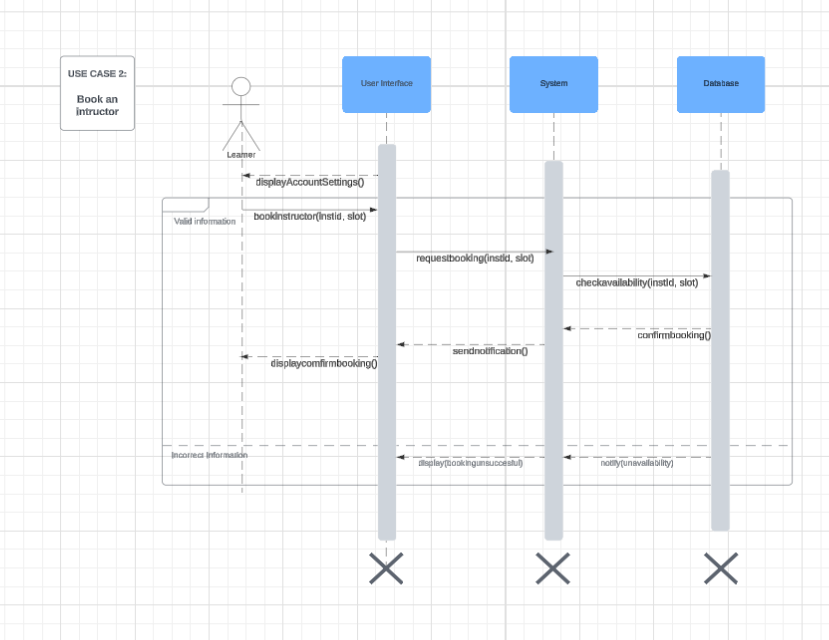
* Users must have a valid email for registration.
* Payment gateways must be operational.
* Reliable internet connection is required.

**2.8 Diagrams**

Sequence Diagrams:

****

****

****

## **3. System Features**

### **3.1 User Registration and Login**

* **Description:** Allows users to register and login as learners or instructors.
* **Priority:** High

### **3.2 Booking Driving Sessions**

* **Description:** Enables learners to book sessions with instructors.
* **Priority:** High

### **3.3 Payment Processing**

* **Description:** Handles payments for booked sessions.
* **Priority:** High

## **4. External Interface Requirements**

### **4.1 User Interfaces Overview**

* **Registration Page:** Allows users to create an account.
* **Login Page:** Allows users to authenticate.
* **Dashboard:** Displays booking options and session history.
* **Payment Gateway:** Allows users to complete payments.

### **4.2 Hardware Interfaces**

* Standard input devices (keyboard and mouse)
* Server hosting with sufficient storage and processing power

### **4.3 Software Interfaces**

* **Database Interface:** PostgreSQL
* **API Interface:** RESTful APIs using Spring Boot
* **Frontend Interface:** ReactJS for user interaction

## **5. System Features/Modules**

### **5.1 User Registration and Login**

#### **5.1.1 Description and Priority**

Users shall be able to register with personal details and login to the system.  
 **Priority:** High

#### **5.1.2 Stimulus/Response Sequences**

* **Stimulus:** User submits registration form  
  + **Response:** System validates input and creates a new user
* **Stimulus:** User attempts to log in  
  + **Response:** System authenticates the user and displays the dashboard

#### **5.1.3 Functional Requirements**

* **REQ-1.1:** The system shall allow users to register with name, email, and password.
* **REQ-1.2:** The system shall validate email format.
* **REQ-1.3:** The system shall authenticate users using email and password.

### **6. Nonfunctional Requirements**

#### **6.1 Performance**

* **NF-1.1:** The system shall respond to login requests within 2 seconds.
* **NF-1.2:** Booking confirmation shall be processed within 1 second.

#### **6.2 Security**

* **NF-2.1:** Passwords shall be encrypted before storage.
* **NF-2.2:** The system shall use HTTPS for secure communication.
* **NF-2.3:** User sessions shall be terminated after 30 minutes of inactivity.