

Workshop Management System

Software Requirements Specification

Developer:
Saptaparno Chakraborty

Table of Contents

1. INTRODUCTION	1
1.1 PURPOSE.....	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	1
1.4 REFERENCES	1
1.5 OVERVIEW	1
2. GENERAL DESCRIPTION.....	1
2.1 PRODUCT PERSPECTIVE.....	2
2.2 PRODUCT FUNCTIONS.....	2
2.3 USER CHARACTERISTICS.....	2
2.4 GENERAL CONSTRAINTS	2
2.5 ASSUMPTIONS AND DEPENDENCIES.....	2
3. SPECIFIC REQUIREMENTS	3
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	3
3.1.1 <i>User Interfaces</i>	3
3.1.2 <i>Hardware Interfaces</i>	3
3.1.3 <i>Software Interfaces</i>	3
3.1.4 <i>Communications Interfaces</i>	3
3.2 FUNCTIONAL REQUIREMENTS.....	3-4
3.3 NON-FUNCTIONAL REQUIREMENTS	4
3.3.1 <i>Performance</i>	4
3.3.2 <i>Reliability</i>	4
3.3.3 <i>Availability</i>	4
3.3.4 <i>Security</i>	4
3.3.5 <i>Maintainability</i>	4
3.3.6 <i>Portability</i>	4
3.4 DESIGN CONSTRAINTS	3
3.5 OTHER REQUIREMENTS	3
4. ANALYSIS MODELS	5-6
4.1 UML USE CASE DIAGRAM.....	5
4.2 ER DIAGRAM	5
4.3 SYSTEM ARCHITECTURE DIAGRAM	6
5. GITHUB LINK.....	7

1. Introduction

1.1 Purpose

This document provides a detailed Software Requirements Specification (SRS) for the Workshop Management System.

1.2 Scope

The Workshop Management System is a web-based application designed to manage workshops, registrations, attendance, materials, feedback, and certificate generation. The system supports administrators, instructors, and participants. It does not handle payment processing or third-party ticketing.

1.3 Definitions, Acronyms, and Abbreviations

Admin: System administrator

JWT: JSON Web Token

UI: User Interface

API: Application Programming Interface

SRS: Software Requirements Specification

1.4 References

IEEE 830-1998 Recommended Practice for Software Requirements Specifications

MongoDB Documentation

React.js Documentation

Node.js Documentation

1.5 Overview

This document describes the overall system, functional and non-functional requirements, external interfaces, and analysis models used in developing the Workshop Management System.

2. General Description

2.1 Product Perspective

This Workshop Management System web app is a standalone web application built using Node.js with MongoDB for the backend, along with JWT-based authentication, and React.js and Tailwind CSS for the frontend.

2.2 Product Functions

- User authentication and role management
- Workshop creation and management
- Participant registration
- Attendance tracking
- Material sharing
- Feedback collection
- Certificate generation

2.3 User Characteristics

Users are expected to have basic computer literacy and familiarity with web applications.

2.4 General Constraints

The system requires an active internet connection and a modern web browser.

2.5 Assumptions and Dependencies

The system assumes availability of MongoDB and Node.js runtime environment.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface shall be web-based and responsive.

3.1.2 Hardware Interfaces

No specific hardware interfaces are required.

3.1.3 Software Interfaces

The system interfaces with MongoDB and REST APIs.

3.1.4 Communications Interfaces

HTTP/HTTPS protocols are used for communication.

3.2 Functional Requirements

3.2.1 User Authentication

Inputs: User credentials

Processing: Validate credentials and generate JWT

Outputs: Authentication token and user details

Error Handling: Invalid credentials return error messages

3.2.2 Workshop Management

Inputs: Workshop details

Processing: Create, update, or delete workshop records

Outputs: Workshop data

Error Handling: Invalid or missing data handled gracefully

3.2.3 Workshop Registration

Inputs: User ID and Workshop ID

Processing: Register participant

Outputs: Registration confirmation

Error Handling: Duplicate registration prevented

3.2.4 Attendance Tracking

Inputs: Participant and workshop details

Processing: Mark attendance

Outputs: Attendance record

Error Handling: Unauthorized access restricted

3.2.5 Feedback Submission

Inputs: Rating and comments

Processing: Store feedback

Outputs: Feedback confirmation

Error Handling: Invalid inputs rejected

3.2.6 Certificate Generation

Inputs: Attendance completion status

Processing: Generate certificate

Outputs: Downloadable certificate

Error Handling: Certificates generated only for eligible participants

3.3 Non-Functional Requirements

3.3.1 Performance

The system should respond within 2 seconds for standard operations.

3.3.2 Reliability

The system should ensure data consistency and prevent data loss.

3.3.3 Availability

The system should be available 99% of the time.

3.3.4 Security

Passwords shall be encrypted, and access shall be role-based.

3.3.5 Maintainability

The system should follow modular and clean coding practices.

3.3.6 Portability

The system should run on any modern web browser.

3.4 Design Constraints

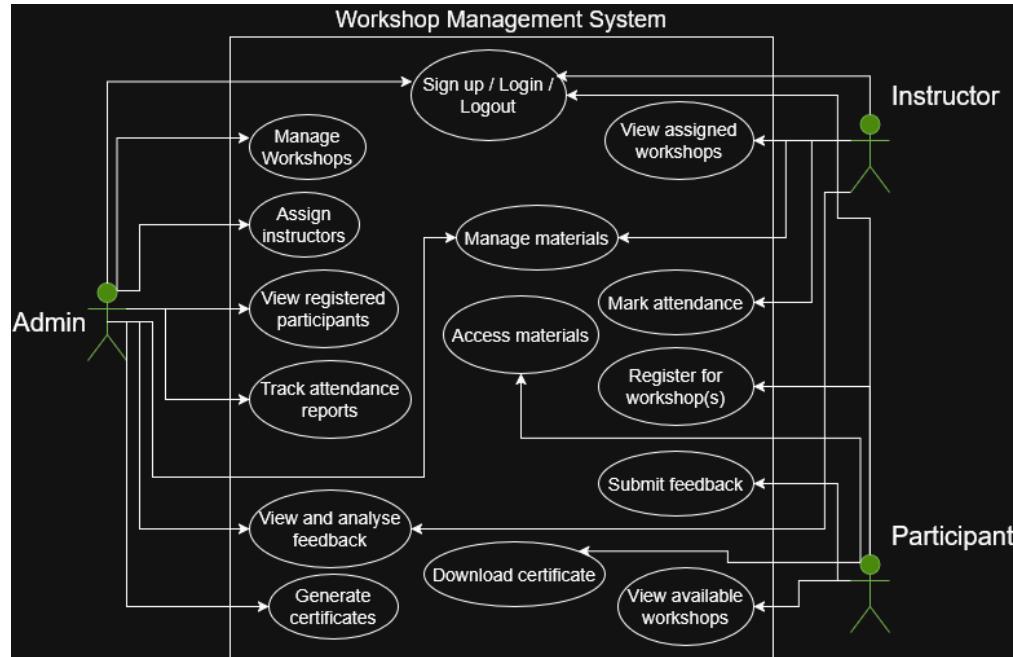
- Technology stack limited to JavaScript-based frameworks
- Browser-based access only

3.5 Other Requirements

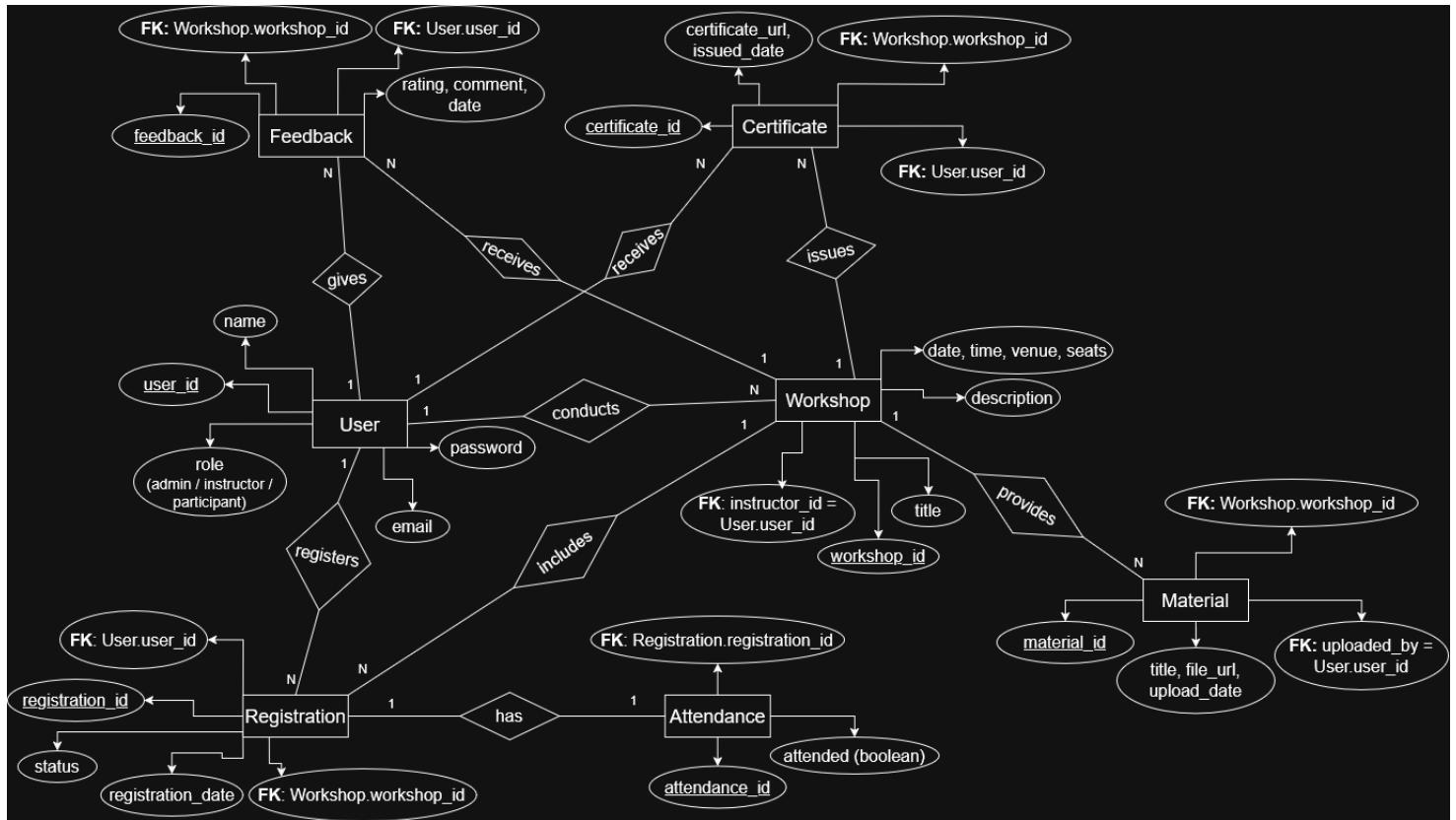
Future enhancements may include analytics and intelligent recommendations.

4. Analysis Models

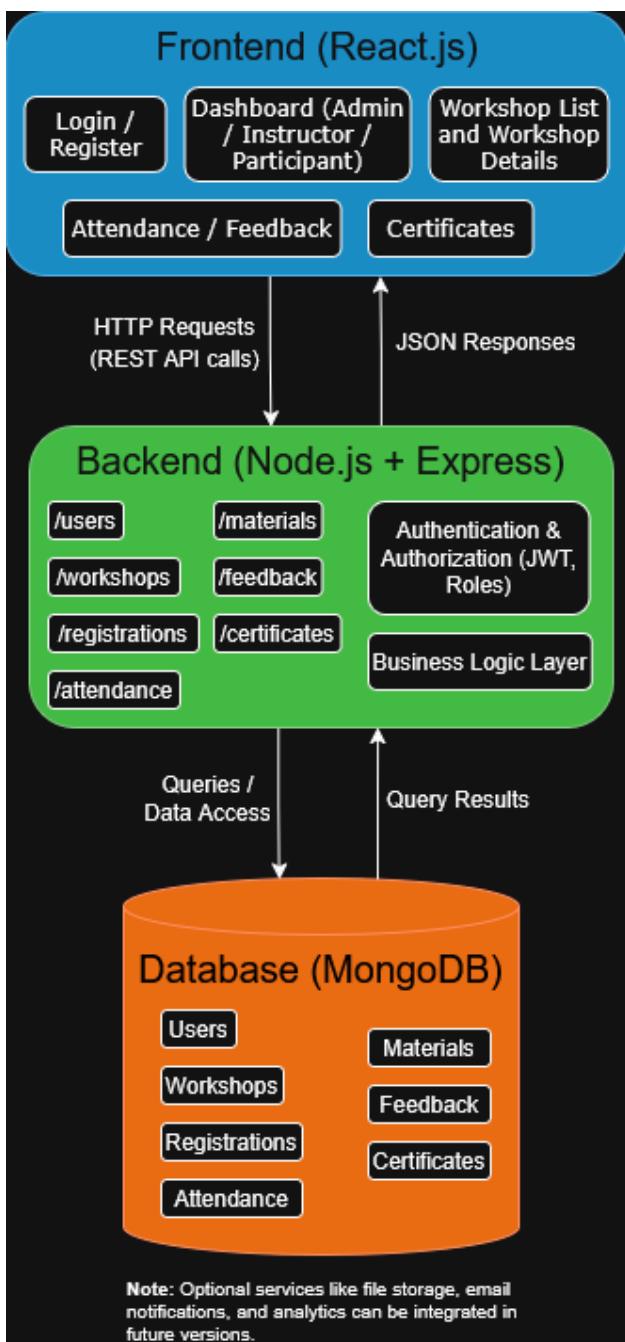
4.1 UML Use Case Diagram



4.2 Entity-Relationship (ER) Diagram



4.3 System Architecture Diagram



5. GitHub Link

<https://github.com/schak04/workshop-management-system>