|  |  |
| --- | --- |
| Training Management System  (TMS) | Software Requirement Specification |
| Version ID Date: 01/02/2023  Document ID: SWD/ 01  Version ID: 1.0 | |
|  | |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc. Ver. | Date | Author | Reviewer | Description of Revision |
| 1.0 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Revision History**

Contents

[1. Introduction 4](#_Toc139539503)

[1.1 Purpose 4](#_Toc139539504)

[1.2 Scope 4](#_Toc139539505)

[1.3 References 5](#_Toc139539506)

[1.4 Definitions, Acronyms and Abbreviations 5](#_Toc139539507)

[2. Overall Description 5](#_Toc139539508)

[2.1 Overview 5](#_Toc139539509)

[2.2 Technical platform 7](#_Toc139539510)

[Functional Requirements: 7](#_Toc139539511)

[User Interface 8](#_Toc139539512)

[Non Functional Requirements 10](#_Toc139539513)

[Performance Requirements 10](#_Toc139539514)

[Security Requirements (SRs) 10](#_Toc139539515)

[Software Quality Attributes 11](#_Toc139539516)

[6.1 Hard Drive Space 11](#_Toc139539517)

[6.2 Application Memory Usage 11](#_Toc139539518)

# 1. Introduction

The Training Management System (TMS) Software Requirements Specification (SRS) defines the requirements for the development and implementation of a comprehensive training management system. The TMS aims to streamline and automate the management of trainers, trainees, courses, batches, assignments, and communication within a training environment. This document provides an overview of the system, its objectives, and the intended audience.

## 1.1 Purpose

The purpose of the TMS (Training Management System) project is to provide a comprehensive platform for managing training programs, trainers, and trainees. It aims to streamline the registration process, create and manage batches and courses, assign trainers and trainees to batches, schedule training programs, facilitate assignment creation and submission, provide a collaborative classroom environment, and enable communication through group chat. The TMS project aims to enhance the efficiency and effectiveness of training management, ensuring smooth coordination between trainers and trainees.

## 1.2 Scope

The Training Management System (TMS) project aims to provide a comprehensive platform for managing training programs, trainers, and trainees. The project's purpose is to enhance the efficiency and effectiveness of training management by streamlining various processes. The scope of the TMS project includes the registration of trainers and trainees, allowing the admin to capture their relevant information such as personal details, contact information, educational background, and expertise. Admin can create batches with specific start and end dates, assign trainers and trainees to these batches using a dynamic user interface, and schedule training programs within each batch. The system enables trainers to create assignments and daily tasks, set deadlines, and track trainees' progress. A collaborative classroom environment is provided where trainers can upload messages, files, and comments in a date-wise order. Trainees can search and filter the data, comment on trainer's posts, and access a notice board managed by trainers. Trainees can submit assignments electronically in various formats, and trainers can review and evaluate the submissions. Additionally, a group chat feature facilitates communication among participants of a specific batch, enabling textual communication and the sharing of images and files. Overall, the TMS project aims to serve as a centralized platform for efficient training management, enhancing collaboration and effectiveness throughout the training process.

## 1.3 References

|  |  |
| --- | --- |
| Reference | Location |
| Requirement Specification |  |

## 1.4 Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| TMS | Training Management System |
| User | Someone who interacts with the application. |
| FR | Functional Requirement |
| NFR | Non-Functional Requirement |
| UI | User Interface |
| API | Application Programming Interface |

# 2. Overall Description

## 2.1 Overview

**Product Perspective:** The Training Management System (TMS) is composed of two main components: the frontend and the backend. These components work together to provide a seamless user experience through a REST API-based connection. The frontend relies on the backend architecture for implementation and dynamically receives resource files from the backend.

**User Characteristics:** The TMS system caters to three different user roles: Admin, Trainer, and Trainee. Each user can have one or multiple roles within the system.

**Admin:** The Admin role holds the highest level of authority in the system. Administrators have the ability to add, modify, and delete information, as well as assign roles to other users. They have complete control over system administration and configuration.

**Trainer:** Trainers are responsible for conducting training programs and managing course content. They have specific permissions and capabilities within the system that allow them to create assignments, set deadlines, and provide guidance to trainees. Trainers may also have access to certain administrative functions related to their training programs.

**Trainee:** Trainees are the individuals participating in the training programs. They have view permissions to access relevant training materials, schedules, and assignments. Trainees can interact with the system to submit their assignments and track their progress. They have limited modification permissions, primarily pertaining to their own personal data.

The TMS system is designed to cater to the specific needs of each user role, ensuring that administrators have the necessary control, trainers can effectively manage training programs, and trainees can access and engage with the training materials and assignments provided to them.

## 2.2 Technical platform

The technical platform for the project includes:

**Backend:**

* Spring Boot for Java application development
* JWT for secure authentication
* JPA for object-relational mapping
* Relational database (MySQL or PostgreSQL) for data persistence

**Frontend:**

* React.js for dynamic and interactive user interfaces
* RESTful APIs for communication with the backend

**Security:**

* JWT token-based authentication
* Authorization and access control mechanisms

**Development Environment:**

* IDE like IntelliJ IDEA
* Package managers like npm

**User Experience:**

Responsive design for a user-friendly experience on different devices.

# Functional Requirements:

|  |  |  |
| --- | --- | --- |
| Requirement ID | Main Features | Requirement Description |
| FR1 | Login | Users with any role can log in with their email and password |
| FR2 | Trainers and Trainee Registration | Admin can register trainers and trainees |
| FR3 | Create Batches | Admin can create new batches with necessary fields, including starting date and ending date |
| FR4 | Create Courses | Admin can create courses and assign trainers to those courses |
| FR5 | Assign Trainers/Trainees to Batches | Admin can assign trainers and trainees to specific batches |
| FR6 | Batch Scheduling | Admin can create and manage the schedule for each batch, including assigning training programs |
| FR7 | Assignment/Daily Task Creation | Trainers can create assignments and daily tasks, assign them to batches, and set deadlines for submission |
| FR8 | Classroom Features with a Dashboard | Trainers can upload messages, files, and comments in the classroom |
| FR9 | Classroom Features with a Notice Board | Trainers can manage a notice board in the classroom |
| FR10 | Submit Assignments | Trainees can submit their assignments, and trainers can view and assess the submissions |
| FR11 | Group Chat Feature | Participants of a specific batch can communicate through a group chat feature |

# User Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **UI No.** | **UI Name** | **Related Function Req ID** | **Description** |
| UI1 | Login Screen | FR1 | A screen where users can enter their credentials to log into the system |
| UI2 | Registration Form | FR2 | A form for trainers and trainees to provide their personal information for registration |
| UI3 | Batch Creation Form | FR3 | A form for the admin to create new batches, including fields for starting date, ending date, and other details |
| UI4 | Course Creation Form | FR4 | A form for the admin to create new courses, including fields for course name, description, and trainer |
| UI5 | Assign Trainers/Trainees | FR5 | A form for the admin to assign trainers and trainees to specific batches |
| UI6 | Batch Schedule Management | FR6 | A screen or component for the admin to create and manage the schedule for each batch |
| UI7 | Assignment Creation Form | FR6 | A form for trainers to create assignments, including fields for assignment title, description, and deadline |
| UI8 | Classroom Dashboard | FR7 | A dashboard where trainers can upload messages, files, and comments for trainees |
| UI9 | Notice Board Management | FR8 | A component for trainers to manage the notice board, including adding, editing, and deleting notices |
| UI10 | Assignment Submission | FR9 | A feature for trainees to submit their assignments, with the ability to upload files in various formats |
| UI11 | Group Chat Feature | FR10 | A chat component or screen for participants of a specific batch to communicate with each other |

# Non Functional Requirements

## Performance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Acceptability/Completion Criteria | Limitations/Constraints |
| NFR1 | Usage of the result in the list view | The results displayed should be user-friendly and clear | None |
| NFR2 | Response Time | Response time should be ≤ 2 seconds (100% of the time) | None |
|  |  | Response time should be ≤ 1 second (100% of the time) |  |
| NFR3 | Application Portability | The application should run on any browser | None |

## Security Requirements (SRs)

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Acceptability/Completion Criteria | Limitations/Constraints |
| SR1 | Front End Layer Role Check | Users should only view options associated with their role | Can be manipulated with JavaScript |
| SR2 | Always check if the token is expired | Smooth auto login functionality | None |
| SR3 | Always filter token for any API request | Secure authentication for API requests | None |
| SR4 | Verify role from token to return response | Protection of sensitive data | None |

Software Quality Attributes

## 6.1 Hard Drive Space

**ID:** QR1

**Title:** Hard Drive Space

**Description:** The application's need for hard drive space.

**Scale:** The amount of hard drive space required by the application.

Measurement: Megabytes (MB).

**Must:** The application should not occupy more than 20 MB of hard drive space.

**Plan:** The target is to occupy no more than 15 MB of hard drive space.

**Wish:** Ideally, the application should occupy no more than 10 MB of hard drive space.

**MB:** Megabyte (defined unit of measurement)

## 6.2 Application Memory Usage

**ID:** QR6

**Title:** Application Memory Usage

**Description:** The amount of operating system memory occupied by the application.

**Scale:** Megabytes (MB).

**Measurement:** Observations obtained from the performance log during testing.

**Must:** The application's memory usage should not exceed 20 MB.

**Plan:** The target is to keep the application's memory usage below 16 MB.

**Wish:** Ideally, the application's memory usage should be below 10 MB.

Top of Form