| **Trainee Management System** | **Software Requirement Specification** |
| --- | --- |
| Version ID Date: 3/7/2023  Document ID: SWD/1  Version ID: 1.0 | |
|  | |
|  | |

**Revision History**

| **Doc. Ver.** | **Date** | **Author** | **Reviewer** | **Description of Revision** |
| --- | --- | --- | --- | --- |
| 1.0 | 3/07/2023 | Zawadul Islam | Sarwar Miral  Nani Gopal |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Contents**

[**1.**](#_heading=h.gjdgxs) **Introduction 4**

[1.1](#_heading=h.4i7ojhp) Purpose 4

[1.2](#_heading=h.2xcytpi) Scope 4

[1.3](#_heading=h.1ci93xb) Intended Stakeholder 4

[1.4](#_heading=h.2et92p0) References 4

[1.5](#_heading=h.3whwml4) Definitions, Acronyms, and Abbreviations 4

[**2.**](#_heading=h.3dy6vkm) **Overall Description 5**

[2.1](#_heading=h.2bn6wsx) Overview 5

[2.2](#_heading=h.4d34og8) Technical platform 5

[**3.**](#_heading=h.2s8eyo1) **Functional Requirements 5**

[3.1](#_heading=h.17dp8vu) Overview 5

[3.1.1.](#_heading=h.qsh70q) Feature/Function 1 6

[3.1.2.](#_heading=h.3rdcrjn) Feature/Function 2 6

[**4.**](#_heading=h.26in1rg) **User Interface 7**

[**5.**](#_heading=h.lnxbz9) **Non-Functional Requirements 7**

[5.1](#_heading=h.3as4poj) Performance Requirements 7

[5.2](#_heading=h.1pxezwc) Safety Requirements 8

[5.3](#_heading=h.44sinio) Security Requirements 8

[**6.**](#_heading=h.49x2ik5) **Design Constraints 8**

[**7.**](#_heading=h.z337ya) **Software Quality Attributes 9**

[**8.**](#_heading=h.3j2qqm3) **User Interface 9**

[**9.**](#_heading=h.1y810tw) **Other Requirements 9**

1. **Introduction**

The Training Management System (TMS) is a sophisticated software solution designed to streamline and automate the management of training programs for BJIT Academy. This software will save a lot of time and reduce the workload of the managers. Additionally, the trainers and trainees will be benefited while conducting any training session.

This Software Requirements Specification (SRS) document serves as a comprehensive guide, outlining the functional and non-functional requirements of the TMS. By providing a clear understanding of the system's scope, objectives, and key features, this document facilitates effective communication and collaboration among stakeholders.

* 1. **Purpose**

The purpose of this document is to define the requirements and specifications of the TMS, acting as a roadmap for the development team. By clearly articulating the desired functionality and behavior of the system, this document ensures that all stakeholders share a common understanding of its capabilities and limitations. It serves as a reference point throughout the development life-cycle, guiding the design, implementation, and testing of the TMS.

* 1. **Scope**

The scope of TMS is determined by the project's unique objectives and goals. Supporting end-to-end management of trainers and trainees, course creation and assignment, batch scheduling, assignment/task creation, submission management, and classroom features are included in the scope of this system. It provides administrators, trainers, and trainees with a centralized platform to efficiently carry out their respective roles, thereby improving the overall training management process within the organization. The TMS encompasses a wide range of functionalities, enabling seamless coordination and collaboration among stakeholders involved in the training programs.

* 1. **Intended Stakeholder**

The BJIT Academy is the main Stack Holder of the project.

* 1. **References**

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

* 1. **Definitions, Acronyms, and Abbreviations**

| **Term/Acronym** | **Definition** |
| --- | --- |
| TMS | Trainee Management System |
| API | Application Programming Interface |
|  |  |

1. **Overall Description**

The subsequent sections of this document delve into the detailed description of the functional and non-functional requirements of the TMS. These requirements encompass every aspect of training management, covering the entire life-cycle from the registration and on-boarding of trainers and trainees to batch creation, course assignment, scheduling, assignment/task creation, submission management, and classroom features. By addressing these requirements comprehensively, the TMS aims to optimize the training management process and enhance the learning experience for all stakeholders involved.

It is important to note that this document is a dynamic artifact that will evolve throughout the development process. As the project progresses, updates and revisions to the SRS will be documented, tracked, and communicated to relevant stakeholders. This ensures that the development team can effectively build and deliver a robust Training Management System that aligns with the organization's objectives and meets the needs of its training programs.

1. **Overview**

This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in section 3, and makes them easier to understand. Include such items as:

1. Product perspective : The Training Management System (TMS) is an integral component of the organization's training infrastructure. It is designed to seamlessly integrate with existing systems and processes related to training management. The TMS will establish interfaces with other systems, such as HR management or learning management systems, to ensure efficient data exchange and streamline training operations.
2. Product functions : The TMS will provide a comprehensive set of functionalities to support the management of trainers and trainees. Key functions include trainer and trainee registration, batch creation, course management, scheduling, assignment creation, submission, and other essential training-related activities. The system aims to automate manual processes, enhance communication, and simplify administrative tasks for improved training program effectiveness.
3. User characteristics : The TMS caters to different user roles with distinct characteristics. Administrators will oversee system configuration, manage user access, and perform administrative tasks. Trainers will utilize the system to create training programs, manage assignments, and track trainee progress. Trainees, on the other hand, will register for training, access course materials, submit assignments, and communicate with trainers.
4. Constraints : The TMS development and implementation are subject to certain constraints. These may include technical limitations such as compatibility with specific browsers or operating systems. Organizational constraints like budgetary constraints, time limitations, and resource availability may also influence the project's scope and timeline. The database system may also face some limitations while accessing it.
5. Assumptions and dependencies : As the project is implemented using java spring boot, the server where it will be hosted needs to have jdk and jre to run the project. For accessing the frontend part, the machine should have npm and react libraries. This service should be accessible from any browser and operating system while having a stable internet connection.

1. **Technical platform**

The technical platform for the Trainee Management System may include the following components:

1. **Operating System:** No particular operating system is needed. Any latest web browser will be enough for accessing the application.
2. **Development Environment:** A suitable integrated development environment (IDE) such as IntelliJ Idea or Eclipse will be required to develop and debug the app. JDK and JRE are prerequisite for running the IDE. On the other hand, VS code will be used as editor and nodejs needs to be installed.
3. **Programming Languages:** Java will be used as the primary programming language for the backend part. For the frontend part we will use javascript.
4. **Frameworks and Libraries:** As it is a large application, we will need framework and library support for several functionalities. That’s why we will use Spring Boot framework for java and reactjs library for javascript.
5. **Database:** A database such as MySQL will be used to store and retrieve data for the application, such as user information, batch schedule and course information.
6. **API Integration:** For creating API, we will use REST controller. Later we will show data in the UI using these APIS.
7. **Security:** The app will need to implement appropriate security measures such as encryption, authentication, and authorization to protect sensitive user data and ensure privacy.
8. **User Experience:** The app will be designed with a focus on user experience, including intuitive navigation, clear and concise user interface, and responsive performance.
9. **Mobile Device Compatibility:** The app will need to be tested and optimized for different types of mobile devices, including smartphones and tablets, to ensure compatibility and optimal performance.

1. **Functional Requirements**
2. **Overview**

This section sums up in the below table the main functionalities or services provided by the sub-system, which will be detailed in the following subsections. A use case diagram could be also used to list the main functionalities.

| **Serial No** | **Main Features** | **Description** |
| --- | --- | --- |
| 1 | Registration and Login | Trainers and trainees will register with their personal information and then the admin will provide a username and password for the trainers and trainees. Inputting the username and password, the user will be able to access the service based on their role |
| 2 | Create Batches | Admin creates batches with start and end dates for training programs. |
| 3 | Create Courses | Admin creates courses and assigns trainers to them, such as Domain Specific Training (J2EE), Soft Skill Training, SCRUM Training, etc. |
| 4 | Assign Trainers/Trainees to Batches | Admin assigns trainees to specific batches. |
| 5 | Batch Scheduling | Admin sets the schedule of each batch and course. |
| 6 | Assignment or Daily Task Creation | Trainers create assignments and daily tasks, assign them to batches, and set submission deadlines. |
| 7 | Classroom | Trainers upload messages, files, and comments in a classroom module. Data is displayed chronologically, and trainees can search, filter, and comment on trainer posts. A notice board is managed by trainers. |
| 8 | Assignment Submission | Trainees submit assignments in various formats (PDF, Docs, Word, PNG). Trainers can view trainee submissions. |

* + 1. **Login and Registration**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_001 | Users should be able to create an account in the TMS by providing necessary information such as full name, email, password, and additional required fields. | Essential | Server might not be available | TC\_001 |
| TMS\_002 | Admin should be able to provide username and password. | Essential | None | TC\_002 |
| TMS\_003 | Users should be able to log in to the TMS using their credentials. Upon successful login, users should be redirected to their respective roles and granted access to the system's functionalities. | Essential | Server might not be available | TC\_003 |
| TMS\_004 | Users should have an option to reset their password if they forget it. | Essential | User may not be registered | TC\_004 |
| TMS\_005 | Users should be able to update their account information. | Essential | User may not be registered | TC\_005 |
| TMS\_006 | Users can log out of their account at any time by clicking the logout button in the application’s menu. | Essential | User may not be logged in | TC\_006 |

* + 1. **Create Batches**

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_007 | Admin should be able to create a new batch in the TMS. | Essential | None | TC\_007 |
| TMS\_008 | Admin should be able to manage batch details, such as updating the starting and ending dates | Essential | None | TC\_008 |
| TMS\_009 | Admin should be able to view a list of all created batches. | Essential | None | TC\_009 |

* + 1. **Create Courses**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_010 | Admin should be able to create courses in the TMS. | Essential | None | TC\_010 |
| TMS\_11 | Admin should be able to manage course details, such as updating the course name or description. | Essential | None | TC\_011 |
| TMS\_12 | Admin should be able to view a list of all created courses. | Essential | None | TC\_012 |
| TMS\_13 | Courses should be assigned to each batch | Essential | None | TC\_013 |

* + 1. **Assign Trainers/Trainees to Batches**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_014 | Admin should be able to assign trainers to batches in the TMS. | Essential | None | TC\_014 |
| TMS\_015 | Admin should be able to assign trainees to batches in the TMS. | Essential | None | TC\_015 |
| TMS\_016 | Admin should be able to view and manage trainer and trainee assignments for each batch. | Essential | None | TC\_016 |
| TMS\_017 | The system should prevent double assignments of trainees to the same batch. | Essential | None | TC\_017 |

* + 1. **Batch Scheduling**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_018 | Admin should be able to create and assign training programs to batches in the TMS. | Essential | None | TC\_018 |
| TMS\_019 | The trainees and trainers should be assigned to batches. | Essential | None | TC\_019 |
| TMS\_020 | Admin should be able to manage the training programs and schedules assigned to batches. | Essential | None | TC\_020 |
| TMS\_021 | The system should handle conflicts and overlaps in batch scheduling. | Essential | None | TC\_021 |

* + 1. **Assignment/Daily Task Creation**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_022 | Trainers should be able to create assignments or daily tasks for a batch in the TMS. | Essential | None | TC\_022 |
| TMS\_023 | Trainees should be able to view and access assigned assignments or daily tasks. | Essential | None | TC\_023 |
| TMS\_024 | The system should handle task submission and track the submission status. | Essential | None | TC\_024 |

* + 1. **Classroom Feature**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_025 | Trainers should be able to upload messages, files, and comments in the classroom. | Essential | None | TC\_025 |
| TMS\_026 | Classroom data should be displayed in date-wise order. | Essential | None | TC\_026 |
| TMS\_027 | Trainees should be able to search and filter the classroom data. | Essential | None | TC\_027 |
| TMS\_028 | Trainees should be able to comment on the posts of trainers. | Essential | None | TC\_028 |
| TMS\_029 | Trainers should be able to manage a notice board in the classroom. | Essential | None | TC\_029 |

* + 1. **Submit assignment**

###### Requirements

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_030 | Trainees should be able to submit their assignments in various file formats. | Essential | None | TC\_30 |
| TMS\_031 | Trainers should be able to view and access the submitted assignments. | Essential | None | TC\_31 |
| TMS\_032 | The system should track the submission status of assignments. | Essential | None | TC\_32 |
| TMS\_033 | Trainees should receive confirmation or acknowledgement upon successful assignment submission. | Essential | None | TC\_33 |

1. **User Interface**

| **UI No.** | **UI Name** | **Related Function Req ID** | **Description** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| TMS\_UI\_001 | Login Screen |  | The initial screen where users can enter their credentials (e.g., username and password) to log into the TMS. It should provide a secure and user-friendly interface for authentication. | TC\_034 |
| TMS\_UI\_002 | Registration Screen |  | A screen where new users can create their accounts by providing required information such as full name, email, contact number, and other necessary details. It should guide users through the registration process and validate the entered information. | TC\_035 |
| TMS\_UI\_003 | Dashboard |  | The main screen after successful login, displaying an overview of important information, such as upcoming batches, assigned courses, and notifications. It should provide a clear and organized view to users, allowing easy navigation to different sections and functionalities of the TMS. There also should be a notice to display important information. | TC\_036 |
| TMS\_UI\_004 | Batches Management |  | A screen where administrators can create and manage batches. It should allow them to input necessary details like batch name, starting date, ending date, add trainees and other relevant information. Administrators should be able to view and edit existing batches. | TC\_037 |
| TMS\_UI\_005 | Courses Management |  | A screen where administrators can create and manage courses. It should provide options to assign trainers to specific courses and define the course details such as name, duration, and description. Administrators should be able to view, edit, and delete existing courses. | TC\_038 |
| TMS\_UI\_006 | Assign Trainers/Trainees to Batches |  | A screen allowing administrators to assign trainers and trainees to specific batches. It should provide a user-friendly interface to search and select trainers/trainees and associate them with the desired batches. | TC\_039 |
| TMS\_UI\_007 | Batch Scheduling |  | A screen where administrators can schedule training programs within batches. It should allow them to create a schedule, assign courses to specific dates and times, and define the duration of each training session. | TC\_040 |
| TMS\_UI\_008 | Assignment/Daily Task Creation |  | A screen where trainers can create assignments or daily tasks for trainees within a batch. It should provide options to set deadlines, attach necessary files or instructions, and specify any additional details. | TC\_041 |
| TMS\_UI\_009 | Classroom Interface |  | A screen representing the virtual classroom where trainers and trainees can interact and share information. It should display messages, files, comments, and notices in a chronological order. Trainees should be able to search, filter, and comment on the posts made by trainers. | TC\_042 |
| TMS\_UI\_010 | Assignment Submission |  | A screen where trainees can submit their completed assignments. It should allow them to upload the assignment files in various formats (e.g., PDF, Docs, Word, PNG) and provide a confirmation upon successful submission. | TC\_043 |

1. **Non-Functional Requirements**

**5.1** **Performance Requirements**

**5.1.1. Response Time:**

a) A transaction's average response time should be less than 2 seconds.

b) A transaction's maximum response time should not exceed 5 seconds.

**5.1.2. Throughput:**

a) The application must be able to process at least 50 transactions per second.

**5.1.3. Capacity:**

a) The application must support a minimum of 10,000 concurrent users.

**5.1.4. Degradation Modes:**

In the event of a degraded network connection, the application should show a warning and save wrok temporarily so that the user doesn’t have to start over.

**5.1.5. Capital Utilization:**

a) Bandwidth usage shouldn’t be too high.

b) Optimizing disk utilization will reduce storage use

c) The use of communications should be improved to reduce data consumption

**5.1.6. Reliability:**

a) The application should be available at least 99.5% of the time.

b) The application should feature error-handling capabilities to maintain stability and reduce accidents.

**5.1.7. Security:**

Encryption and secure authentication measures should be used to safeguard the application against unwanted access and data breaches.

**5.1.8. Scalability:**

a) The application should be able to accommodate an increase in the number of users and transactions without seeing a noticeable decrease in performance.

b) Future additions and functions should be readily included into the application's design.

**5.1.9. Compatibility:**

The application should be compatible with a broad variety of browsers and systems.

**5.1.10. Support:**

The application should be regularly reviewed and updated in order to maintain optimum performance and handle any problems that may occur

**5.1.11. Usability:**

The application's UI should be straightforward and user-friendly, with easy navigation and accessible functions.

**5.2** **Safety Requirements**

**5.2.1.** The app should not expose users to any potential harm or danger, including physical harm or psychological harm.

**5.2.2.** The app should protect the privacy and security of user data, including personal information and login credentials.

**5.2.3.** The app should not cause damage to the device or equipment being used to access the app.

**5.3** **Security Requirements**

**5.3.1. Data Security:** The TMS should employ robust data security measures to safeguard personal and sensitive information. This includes encryption of data at rest and in transit, secure storage mechanisms, and protection against unauthorized access, data breaches, and data loss.

**5.3.2. User Authentication and Authorization:** The system should have a secure login mechanism to verify the identity of users and grant appropriate access privileges based on their roles. It should implement strong password policies, password hashing, and account lockout mechanisms to prevent unauthorized access. The JWT token should be used for token based authentication.

**5.3.3. Privacy Protection:** The TMS should comply with applicable privacy regulations. It should obtain user consent for data processing, provide transparency about data handling practices, and offer options for users to manage their privacy preferences.

**5.3.4. Role-based Access:** Access should be controlled based on the user's role and permissions. Modules should be allocated functions depending on the user's role and permissions. This will aid in preventing unauthorized access to and alteration of sensitive data.

**5.3.5. Audit Trail and Logging:** The system should maintain detailed audit logs, capturing user activities, system events, and changes made to critical data. This allows for traceability, monitoring, and investigation of any suspicious or unauthorized activities.

**5.3.6. Backup and Disaster Recovery:** The TMS should have a robust backup strategy in place to regularly back up data and ensure its recoverability in case of system failures, natural disasters, or other unforeseen events. It should have mechanisms for data restoration and business continuity planning.

**5.3.7. Vulnerability Management:** Regular security assessments and vulnerability scanning should be performed to identify and address any potential security weaknesses or vulnerabilities in the system. This includes patch management, software updates, and proactive measures to mitigate security risks.

**5.3.8. Training and Awareness:** Proper training and awareness programs should be conducted to educate system administrators, trainers, and trainees about best security practices, data handling guidelines, and potential risks. This helps to foster a security-conscious culture and minimize human-related security vulnerabilities.

1. **Design Constraints**

The design constraints for this app are as below:

6.1. The software languages for the TMS should include Java for the main application code and reactjs for the user interface design. The development tools that can be used may include IntelliJ Idea or Eclipse as IDE.

6.2. The architectural constraints for the app may include the use of a Model-View-Controller (MVC) architecture, which provides a clear separation between the data, user interface, and business logic of the system. This can help ensure maintainable and scalable code.

6.3. The system design may need to adhere to certain hardware or software constraints, such as limited processing power, memory, or storage capacity. Compatibility with specific operating systems, browsers, or devices may also be a constraint that needs to be considered.

6.4. The TMS may have performance constraints that require efficient handling of large volumes of data, quick response times, and minimal system resource utilization. The design should aim to meet performance expectations within the given constraints.

6.5. The TMS may have operational constraints that impact the system's availability, maintenance, and support. These constraints could include scheduled maintenance windows, downtime limitations, or requirements for backup and disaster recovery procedures.

1. **Software Quality Attributes**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |