# Trainee Management System	# Software Requirement Specification

Version ID Date: 06/07/2023

Document ID: SWD/01

Version ID: 1.0

Revision History

Doc. Ver.	Dat	Author	Reviewer	Description of Revision
1.0		Kazi Md Nazmul Hoque ID:00-30059	Nani Gopal Barai, Sarwar Miral, Rezaur Rahaman, Raihan Uddin	

Contents

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the "Training Management System" (TMS) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

The scope of the Training Management System encompasses various modules, including login/registration, batch management, course management, batch scheduling, assignment management, classroom module, and group chat. Each module contributes to the overall functionality of the system, supporting trainers and trainees in managing and participating in training programs effectively.

1. Login/Registration Module:

- Users with ADMIN credential can register and create accounts for the system users.
- Users can log in to access the system.
- User roles and permissions are defined to differentiate between trainers and trainees.
- User can create their own profiles.

2. Batch Management Module:

- Admin users can create new batches.
- Batch details such as batch name, start date, end date, and additional information can be entered.
- Admin users can view, edit, and delete existing batches.
- Trainees can be assigned to specific batches.

3. Course Management Module:

- Admin users can create new courses.

- Course details such as course name, description, and assigned trainers can be entered.
- Admin users can assign trainers to courses.
- Admin users can view, edit, and delete existing courses.
- Trainer users can see their assigned courses.

4. Batch Scheduling Module:

- Admin users can schedule training programs within a batch.
- Training program details such as program name, start time, and end time can be entered.
- Admin users can assign courses to specific training programs/batches.
- Admin users can view, edit, and delete scheduled training programs.

5. Assignment Management Module:

- Trainers can create assignments or daily tasks for batches.
- Assignment details such as title, description, and submission deadline can be entered.
- Trainers can assign assignments to specific batches.
- Trainees can submit their assignments.
- Trainers can view and evaluate trainee submissions.

6. Classroom Module:

- Trainers can upload messages, files, and comments to the classroom.
- Messages and files are displayed in date-wise order.
- Trainees can search and filter the classroom data.
- Trainees can comment on trainer posts.
- Trainers manage a notice board within the classroom.

7. Group Chat Feature:

- Participants of a specific batch can communicate through a group chat.
- Textual communication, image sharing, and file sharing are supported.
- Participants can view chat history.
- The group chat feature facilitates communication among participants.

The Training Management System aims to provide a comprehensive platform for managing training programs, supporting trainers in delivering courses, facilitating assignment management, and promoting effective communication among participants. The system offers functionalities for batch and course management, scheduling, assignment creation and submission, classroom collaboration, and group chat, providing a seamless and efficient training experience for trainers and trainees.

1.3 Intended Stakeholder

The BJIT Academy is the main Stack Holder of the project

1.4 References

Reference	Location
Requirement Specification	

1.5 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
TMS	Training Management System
User	Someone who interacts with the application.
SRS	Software Requirement Specification
API	Application Programming Interface
JWT	JSON Web Token
CM	Classroom Module

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Overview

the Training Management System provides a comprehensive solution for managing training programs, supporting trainers in course delivery, facilitating assignment management, and promoting effective communication between trainers and trainees. The system aims to streamline administrative tasks, enhance communication, and provide a user-friendly platform for a seamless training experience.

1. The Login/Registration module allows users to create accounts and log in to access the system. User roles and permissions differentiate between trainers and trainees, ensuring appropriate access levels.

2. The Batch Management module enables admin users to create new batches by entering details such as batch name, start date, end date, and additional information. Admin users have the ability to view, edit, and delete existing batches. Trainees can be assigned to specific batches for streamlined organization.

3. The Course Management module empowers admin users to create new courses and assign trainers to those courses. Course details such as course name, description, and assigned trainers can be entered and managed. Admin users can view, edit, and delete existing courses, ensuring effective course administration.

4. The Batch Scheduling module allows admin users to schedule training programs within specific batches. Details such as program name, start time, end time, and venue can be assigned, and courses can be associated with each training program. Admin users can also view, edit, and delete scheduled training programs, ensuring efficient management of the training schedule.

5. The Assignment Management module enables trainers to create assignments or daily tasks for batches. Trainers can enter assignment details such as title, description, and submission deadline. Assignments can be assigned to specific batches, and trainees can submit their completed assignments. Trainers have the ability to view and evaluate trainee submissions, facilitating effective assignment management.

6. The Classroom Module provides a platform for trainers to upload messages, files, and comments for trainees. All data is displayed in a date-wise order, and trainees can search and filter the information. Trainees can also comment on trainer posts, promoting collaboration and interaction. Trainers manage a notice board within the classroom, facilitating important announcements and updates.

7. The Group Chat Feature allows participants of specific batches to communicate through a group chat interface. Participants can engage in textual communication, share images, and exchange files. The chat history is accessible, allowing participants to refer back to previous conversations. The group chat feature promotes communication and collaboration among batch participants.

2.2 Technical platform

The technical platform for the Evaluation Management System, when implemented as a web application, may include the following components:

Backend:

- Framework: Spring Boot
- Language: Java
- Database: Specify the preferred database technology (e.g., MySQL, PostgreSQL, MongoDB)
- API Integration: Use relevant libraries or frameworks to integrate with external APIs (if required)
- Security: Implement authentication and authorization using Spring Security
- Deployment: Deploy the backend services on a suitable server or cloud platform (e.g., AWS, Azure, Heroku)

Frontend:

- Framework: React.js
- Language: JavaScript (ES6+)
- UI Library: Utilize popular UI libraries such as React Bootstrap, Material-UI, or Ant Design for efficient frontend development
- State Management: Use Redux or React Context for managing application state
- CSS Preprocessor: Employ Sass or LESS for enhanced CSS styling
- Browser Compatibility: Ensure the application is compatible with major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge
- Responsive Design: Implement responsive design principles to ensure optimal viewing experience across different devices and screen sizes
- API Integration: Communicate with backend APIs using Fetch API, Axios, or other suitable libraries
- Form Validation: Implement client-side form validation using libraries like Formik or React Hook Form
- Internationalization (i18n): Consider incorporating i18n support for multi-language capabilities
- Testing: Utilize testing frameworks like Jest and Enzyme or React Testing Library for unit and integration testing

Deployment:

- Backend: Deploy the Spring Boot backend on a suitable server or cloud platform, such as AWS Elastic Beanstalk or Azure App Service.
- Frontend: Build the React.js application and deploy it as a static website using platforms like AWS S3, Netlify, or Vercel.

API Documentation:

- Generate API documentation using tools like Swagger or Springfox to provide a clear understanding of available endpoints, request/response formats, and authentication requirements.

Version Control:

- Use a version control system (e.g., Git) to manage code repositories and enable collaboration among team members.

Monitoring and Logging:

- Employ logging frameworks (e.g., Log4j, SLF4J) for logging application events and errors.
- Consider implementing monitoring solutions (e.g., Prometheus, Grafana) to track system metrics, performance, and errors.

These are some of the key technical components that may be part of the technical platform for the Android cricket app. The technical platform should be chosen and implemented in a way that ensures the app meets the needs and requirements of the end-users and stakeholders.

3. Functional Requirements

3.1 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	Login/Registration	Users should be able to log in to the system using their credentials. User with Admin can register new accounts for the System.
2	Batch Management	The system should allow the creation, modification, and deletion of batches. Batches are used to group trainees and assign training programs. This feature ensures efficient organization and management of training activities.
3	Course Management	The admin should be able to create courses and assign trainers to those courses. Course management enables the offering of various training programs and ensures proper allocation of trainers to each course.
4	Batch Scheduling	The admin can create training programs within a batch and schedule them. Scheduling includes specifying the start time, end time, and venue for each program. Batch scheduling ensures proper planning and coordination of training sessions.

5	Assignment Management	Trainers can create assignments or daily tasks for specific batches, assign deadlines, and track trainee progress. This feature facilitates the effective management and evaluation of trainee assignments, allowing trainers to provide timely feedback and monitor trainee performance.
6	Classroom Module with a Dashboard	Trainers can upload messages, files, and comments in the classroom module. The module displays data in date-wise order, allowing trainees to search, filter, and comment on trainer posts. A notice board managed by trainers serves as a central location for important announcements and updates. The classroom module promotes effective communication and collaboration between trainers and trainees.
7	Group Chat Feature	Participants of a specific batch can engage in group chat discussions. The feature supports textual communication and enables the sharing of images and files. The group chat facilitates real-time interaction and collaboration among batch members, promoting knowledge sharing and team cohesion.

3.1.1. Login/Registration

This section provides a detailed overview of Login/Registration.

Login: Once a user has created an account, they can log in using their email address and password. A JWT is produced by the system. Once a User requests for information, this token will be filtered and the User role will be verified. The user will be redirected to its intended page based on his credentials.

Registration: User with role Trainer and Trainee will be registered by the user with role Admin.

Logout: Users can log out of their account at any time by clicking the logout button in the system's menu.

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_001	User should be able to login using their email and password	Essential	Server might not be available	TC-001
TMS_002	The system should handle a large number of registrations without performance degradation. The registration process should have appropriate validation to ensure data integrity. The system should enforce unique email addresses for trainees.The system should enforce unique contact numbers for trainers and trainees.	Essential	Server might not be available	TC_002
TMS_003	If a user forget their password then they should be able to recover the password	Essential	User might not be registered	TC_003
TMS_004	User should be able to log out	Essential	User may not be logged in	TC_004

3.1.2. Batch management

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_005	The admin should be able to create a new batch by providing the necessary details, such as batch name, starting date, and ending date.	Essential	Need Admin access	TC_005
TMS_006	The admin should be able to modify an existing batch by updating its details, such as the batch name, starting date, or ending date.	Essential	Need Admin access	TC_006
TMS_007	The admin should be able to assign trainees in a batch. Can assign multiple trainees at a time.	Essential	Need admin access	TC_007
TMS_008	The admin should be able to see all the batches with their starting_time, ending_time, and name.	Essential	Need admin access	TC_008
TMS_009	User with trainee credential can see their assigned batch details	Essential	Batch has to be created before displaying to the trainees	TC_009

3.1.3. Course management

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_010	The admin should be able to create a new course by providing the necessary details, such as course name, description,.	Essential	Need admin access	TC_010
TMS_011	Get the courses	Essential	Need admin access	TC_011
TMS_012	Trainer should be able to see the courses they are assigned to	Optional	Course needs to be created	TC_012
TMS_013	Admin should be able to modify courses	Essential	Need admin access	TC_013

3.1.4. Batch scheduling

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_014	The admin should be able to create a new schedule within a specific batch. This includes providing details such as schedule name, start time, end time.	Essential	Need admin access	TC_014

TMS_015	Assign course and batch name to the schedule.	Essential	Need admin access. One course can be assigned to a schedule	TC_015
TMS_016	Assign trainer to a schedule	Essential	Only one trainer can be assigned to a schedule	TC_016
TMS_017	Update schedule timing	Essential	Need admin access	TC_017
TMS_018	User(admin, trainer, trainee) can view schedules.	Essential	Batch name is essential	TC_018

3.1.5. Assignment management

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier

TMS_019	Trainers should be able to create assignments or daily tasks for specific batches. The assignment creation process should include providing a title, description, and deadline for submission.	Essential	One assignment can be assigned	TC_019
TMS_020	Trainer should assign the assignment to a specific batch and a schedule	Essential	One assignment can be assigned	TC_020
TMS_021	Trainers should be able to assign file for the assignment	Essential	One assignment can be assigned	TC_021
TMS_022	Trainees should be able to view assignment details	Essential	Assignment needs to be assigned to a batch	TC_022
TMS_023	Trainees should be able to submit assignment answer to an assignment. The submission process should allow trainees to upload files in various formats, such as PDF, DOC	Essential	None	TC_023
TMS_024	Trainers should be able to put marks on an assignment answer.	Not Essential	Has to implement a separate module for mark evaluation	TC_024

3.1.6. Classroom management

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_025	Trainer should be able to create a classroom for a batch with specific course.	Essential	One trainer can be associated with one classroom	TC_025
TMS_026	Trainers should be able to upload messages, files, and comments within the classroom module.	Essential	File needs to be in PDF or DOC format	TC_026
TMS_027	Trainers should be able to create multiple notice for a classroom.	Essential	None	TC_027
TMS_028	Trainees should be able to view the data in the classroom.	Essential	None	TC_028
TMS_029	Trainees can search/filter data.	Essential	Can not modify information	TC_029
TMS_030	Trainees should be able to create reply for the attachments to clarify questions.	Essential	None	TC_030

3.1.7. Group chat feature

Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_031	Participants of a specific batch should have access to a group chat interface where they can engage in a textual communication	Optional	Group needs to be created before	TC_031
TMS_032	Participants should be able to engage in textual communication within the group chat.	Essential	Participants need to be added to the group.	TC_032
TMS_033	The system should maintain a history of group chat conversations for participants to refer to previous messages.	Not Essential	Database mechanism to limit the history of chat	TC_033
TMS_034	Participants should have the option to send private messages to specific individuals within the group chat or send messages to the entire group.	Not Essential	Complex Security mechanism	TC_034

4. User Interface

UI No.	UI Name	Related Function Req ID	Description	Test case Identifier

TMS_UI_001	Login page	TMS_001	The login page should provide a user interface for users to enter their login credentials, such as email address and password.	TMS_UI_TC_001
TMS_UI_002	Registration Page	TMS_002	Registration page provides admin to register trainee and trainer to the system	TMS_UI_TC_002
TMS_UI_003	Batch Creation Page	TMS_005	The batch creation page should provide a user interface for admins to enter the necessary details, such as batch name, starting date, and ending date, to create a new batch.	TMS_UI_TC_003
TMS_UI_004	Assign trainees to batch	TMS_007	This page should provide a an admin to assign trainees to a batch	TMS_UI_TC_004
TMS_UI_005	Batch Information	TMS_008	This page displays all the batch specific information in a detailed manner	TMS_UI_TC_005
TMS_UI_006	Course Creation page	TMS_010,TMS_011	The course creation page should provide a user interface for admins to enter the necessary details, such as course name, description, and assigned trainers, to create a new course.	TMS_UI_TC_006
TMS_UI_007	View courses	TMS_014	Admin will be able to see all courses. Trainer can see the courses assigned to them	TMS_UI_TC_007
TMS_UI_008	Batch Schedule Page	TMS_015, TMS_016, TMS_017	The batch schedule page should provide a user interface for admins to schedule training programs within a batch by entering the program details, such as program name, start time, end time, trainer email, course name.	TMS_UI_TC_008
TMS_UI_009	Schedule display page	TMS_019	Trainees can see the schedules assigned to their batches. Trainer can see the schedules assigned to them.	TMS_UI_TC_009

TMS_UI_010	Assignment Creation Page	TMS_020, TMS_021, TMS_022	The assignment creation page should provide a user interface for trainers to create assignments or daily tasks by entering the assignment details, such as title, description, and deadline for submission.	TMS_UI_TC_010
TMS_UI_TC_011	Assignment view page	TMS_023	Trainees should be able to view the assignment assigned to them in this page.	TMS_UI_TC_011
TMS_UI_TC_012	Assignment submission page	TMS_024	Trainees should be able to submit their assignment answer.	TMS_UI_TC_012
TMS_UI_TC_013	Classroom creation page	TMS_026	Trainer can create classroom by providing information to the field in this page's form.	TMS_UI_TC_013
TMS_UI_TC_014	Classroom Module Dashboard	TMS_029, TMS_030	The classroom module dashboard should provide a user interface for trainers and trainees to view messages, files, comments, notices, and other relevant information in a structured and organized manner. It should display data in date-wise order, allow trainees to search and filter the data, enable trainees to leave comments on trainer posts, and provide a notice board managed by trainers.	TMS_UI_TC_015
TMS_UI_TC_015	Classroom attachment creation page	TMS_027	Trainer can create attachments/comment/message in this page which will be shown in Trainee classroom dashboard.	TMS_UI_TC_015
TMS_UI_TC_016	Notice creation page	TMS_028	Trainer can create notice for the batch in the classroom	TMS_UI_TC_016

TMS_UI_TC_017	Group Chat Interface	TMS_032, TMS_033, TMS_034, TMS_035	The group chat interface should provide a user interface for participants of a specific batch to engage in textual communication, share images and files, and view chat history. It should display messages in real-time, support text formatting options, allow participants to customize their chat experience, and provide options for navigating the chat history.	TMS_UI_TC_017

5. Non-Functional Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TMS_NFR_001	The user interface should have an intuitive design and provide a user-friendly experience. It should be visually appealing, easy to navigate, and consistent across different screens and devices.	User should be able to get the information, can create necessary information for other pages to display.	The system should load and display the user interface within acceptable response times to provide a smooth and seamless user experience and follow UI best practices.	TMS_NFR_TC_001

TMS_NFR_002	The user interface should be responsive and compatible with different screen sizes, resolutions, and devices, including desktops, laptops, tablets, and mobile phones. It should adapt and display properly on various browsers and operating systems.	Show information in a readable manner in every window sizes	The user interface should be tested on different devices, browsers, and operating systems to ensure compatibility and responsiveness.	TMS_NFR_TC_002
TMS_NFR_003	The user interface should deliver a fast and responsive performance to provide users with a smooth and seamless experience.	Response should be under 2-3 seconds.	The system should be optimized to minimize loading times and deliver fast response times for user interactions.	TMS_NFR_TC_003
TMS_NFR_004	The user interface should implement appropriate security measures to protect user data and ensure the confidentiality, integrity, and availability of the system.	Data should be secured following industry best practices	The system should follow data protection regulations and guidelines, such as General Data Protection Regulation (GDPR) or relevant privacy laws.	TMS_NFR_TC_004
TMS_NFR_005	Sclability/flexibility	The user interface should be scalable and flexible to accommodate future growth and changes in system requirements.	The system should be able to handle a growing number of concurrent users without significant performance degradation.	TMS_NFR_TC_005

5.1 Performance Requirements

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.

2. Throughput:

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

3. Reliability:

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

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

4. Security:

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

5. 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.

6. Usability:

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

   b) With adequate support for screen readers, high-contrast mode, and other accessibility features, the application should be accessible to people with impairments.

7. Capital Utilization:

   a) Memory usage should not above 250 MB.

   b) Optimizing disk utilization will reduce storage use

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

   d) The application should be built to reduce battery use and prevent excessive heat production.

5.2 Safety Requirements

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier

SAFETY_001	Data Privacy and Confidentiality.	The system should ensure the privacy and confidentiality of user data.	The system should comply with relevant data protection laws and regulations, such as the General Data Protection Regulation (GDPR) or other applicable privacy standards	SAFETY_TC_001
SAFETY_002	System Backup and Disaster Recovery	The system should implement backup and disaster recovery mechanisms to ensure the availability and integrity of data. It should regularly backup data and have procedures in place to recover data in case of system failures, natural disasters, or other unforeseen events.	The system should perform regular backups of data to prevent data loss in the event of system failures or disasters.	SAFETY_TC_002

SAFETY_003	User Safety and System Reliability	The system should prioritize user safety and ensure the reliability of its operations. It should be designed and implemented with fail-safe mechanisms to prevent accidents or risks to users. Additionally, the system should have high availability and minimize downtime to provide a reliable user experience.	The system should undergo regular testing and quality assurance processes to identify and fix any issues that may compromise user safety or system reliability.	SAFETY_TC_003
SAFETY_004	Secure Authentication and Authorization	The system should ensure secure authentication and authorization mechanisms to protect user accounts and prevent unauthorized access. It should enforce strong password policies, implement multi-factor authentication (MFA), and verify the identity of users before granting access to sensitive functionalities or data.	The system should handle cases where users forget their passwords or encounter account lockouts and provide secure account recovery mechanisms.	SAFETY_TC_004

SAFETY_005	Error Handling and User Notifications	The system should handle errors gracefully and provide clear error messages to users. It should notify users about system status, maintenance periods, or any other relevant information that may impact their interaction with the system.	The system should notify users about scheduled maintenance or system downtime in advance to minimize user inconvenience.	SAFETY_TC_005

5.3 Security Requirements

<Specify the factors that protect the software from accidental or malicious access, use, modification, destruction or disclosure. Specific requirements may include the need to

- utilize certain cryptographic techniques
- Keep specific log or history data sets.
- assign certain functions to different modules
- restrict communications between some areas of the program
- check data integrity for critical variables>

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
SR_001	User Authentication and Authorization	The system should enforce secure user authentication and authorization mechanisms to ensure that only authorized users can access the system and perform specific actions.	The system should enforce password policies, such as complexity requirements and password expiration, to enhance security.	SR_TC_001

SR_002	Data Encryption and Confidentiality	The system should encrypt sensitive data to ensure its confidentiality and protect against unauthorized access. It should use strong encryption algorithms and secure key management practices to safeguard data at rest and in transit.	The system should comply with relevant data protection regulations and guidelines, such as GDPR or HIPAA, when handling and storing sensitive data.	SR_TC_002
SR_003	Secure Communication Channels	The system should ensure secure communication channels to protect sensitive data during transmission. It should use secure protocols, such as HTTPS, for data exchange between the client and server.	The system should prevent man-in-the-middle (MITM) attacks by validating server certificates and using secure connections.	SR_TC_003
SR_004	Security Testing and Vulnerability Assessments	The system should undergo regular security testing and vulnerability assessments to identify and address potential security vulnerabilities. It should employ industry-standard security practices, such as penetration testing and code reviews, to ensure the robustness of the system.	Security tests should cover various aspects, including network security, application security, and secure coding practices.	SR_TC_004
SR_005	Protection against Cross-site Scripting attacks	User input should be properly validated.	Encoding of user input	SR_TC_005

SR_006	Prevention against SQL injection attacks	Data input by users should not be directly associated to repository data	Parameterized queries and specifications	SR_TC_006

6. Design Constraints

The design constrains for this app are as below:

1. Technology Stack Limitations: The Training Management System must adhere to the limitations and constraints imposed by the chosen technology stack, including Spring Boot for the backend and React.js for the frontend. It is essential to consider the compatibility and supported features of these frameworks and libraries when designing and implementing the system.

2. Database Constraints: The system must work within the limitations of the selected database technology (e.g., MySQL, PostgreSQL, MongoDB). This includes understanding the maximum data storage capacity, data indexing capabilities, and any performance constraints specific to the chosen database technology.

3. Performance Constraints: The system should be designed to perform efficiently and provide a responsive user experience. However, certain performance constraints may arise due to limitations in server resources, network bandwidth, or processing power. It is important to consider these constraints and optimize system components accordingly to ensure optimal performance.

4. Scalability Constraints: The system should be designed to handle growing user bases and increased usage over time. However, scalability constraints may arise due to limitations in server capacity or network infrastructure. Consideration should be given to scalability mechanisms such as load balancing, horizontal scaling, and cloud services to accommodate future growth and ensure the system can handle increased workloads.

5. Security Constraints: The system must adhere to security constraints to protect sensitive data and prevent unauthorized access. This includes implementing proper authentication and authorization mechanisms, secure communication protocols (e.g., HTTPS), and data encryption. Compliance with industry standards and regulations (e.g., GDPR, HIPAA) should also be considered when designing security measures.

6. Integration Constraints: The system may need to integrate with external systems or APIs for specific functionalities. Integration constraints may include compatibility issues, data format requirements, and access restrictions imposed by the external systems. It is important to understand these constraints and design appropriate integration mechanisms to ensure seamless communication and data exchange.

7. User Interface Constraints: The system's user interface (UI) should adhere to design constraints, including platform-specific guidelines (e.g., Material Design for Android), responsive design principles, and accessibility standards. Consideration should be given to supporting various screen sizes,

resolutions, and user interaction patterns to provide a consistent and user-friendly experience across different devices and platforms.

8. Time and Resource Constraints: The system development process should consider time and resource limitations, including project deadlines, budget constraints, and available development resources. Efficient project planning, resource allocation, and prioritization of features and functionalities are crucial to meet project objectives within the given constraints.

7. Software Quality Attributes

REQUIREMNT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
QA_001	Functionality	The software should fulfill the functional requirements and provide all the specified features and capabilities. It should accurately perform the intended tasks and operations.	The software should handle various scenarios, edge cases, and input combinations effectively.	QA_TC_001
QA_002	Reliability	The software should consistently perform its intended functions without failures or errors. It should be reliable and stable, with minimal downtime or disruptions.	The software should be thoroughly tested for stability and reliability to ensure it can handle expected workloads without failures or crashes.	QA_TC_002

QA_003	Usability	The software should be user-friendly and easy to use. It should have an intuitive interface and provide a pleasant user experience. Users should be able to accomplish tasks efficiently and effectively.	The software should undergo usability testing to ensure it meets user expectations and is easy to learn and navigate. The software should provide clear and concise instructions, labels, and tool tips to guide users through the application.	QA_TC_003
QA_004	Performance	The software should perform efficiently and respond quickly to user interactions. It should handle expected workloads without significant delays or performance bottlenecks.	The software should be tested under different load conditions to ensure it can handle expected user concurrency and data volume.	QA_TC_004
QA_005	Scalability	The system should be designed to handle increasing amounts of data, users, and concurrent requests without significant degradation in performance.	The system should be scalable by utilizing technologies such as load balancing, clustering, or cloud services to accommodate growing user bases and increased usage	QA_TC_005

QA_006	Maintainability	The system's code-base should be well-organized, modular, and maintainable. It should follow best practices and coding standards, making it easy for developers to understand and modify the code when necessary.	The system's architecture and code structure should follow industry best practices and design patterns to promote maintainability.	QA_TC_006
QA_007	Extensibility	The system should be designed to accommodate future enhancements and new features. It should have a flexible architecture that allows for easy integration of additional modules or functionality without significant code modifications.	The system's APIs or interfaces should be well-defined and designed to support future extensions and integration. The system should support the integration of third-party tools or services for enhanced functionality.	QA_TC_007

8. Other Requirements

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