| **EMS** | **Software Requirement Specification** |
| --- | --- |
| Version ID Date: 26/07/2023  Document ID: SWD/ 01  Version ID: 1.0 | |
|  | |
|  | |

**Revision History**

| **Doc. Ver.** | **Date** | **Author** | **Reviewer** | **Description of Revision** |
| --- | --- | --- | --- | --- |
| 1.0 | 26/07/22 | Arup Chakraborty | Nani Gopal Barai,  Sarwar Miral |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Contents**

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

[1.1](#_heading=h.z337ya) Purpose 4

[1.2](#_heading=h.3j2qqm3) Scope 4

[1.3](#_heading=h.3znysh7) Intended Stakeholder 4

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

[1.5](#_heading=h.1y810tw) Definitions, Acronyms, and Abbreviations

[**2.**](#_heading=h.4i7ojhp) **Overall Description 5**

[2.1](#_heading=h.1t3h5sf) Overview 5

[2.2](#_heading=h.4d34og8) Technical platform 6

**3.** **Functional Requirements 7**

[3.1](#_heading=h.2xcytpi) Overview 6

3.2 Login/ Registration 8

[3.3.](#_heading=h.3whwml4) Evaluation System 9

3.4 Marking System 10

3.5 User Information 10

[**4.**](#_heading=h.26in1rg) **User Interface 10**

[**5.**](#_heading=h.2bn6wsx) **Non-Functional Requirements 12**

[5.1](#_heading=h.qsh70q) Performance Requirements 13

[5.2](#_heading=h.1ksv4uv) Safety Requirements 13

[5.3](#_heading=h.3as4poj) Security Requirements 15

[**6.**](#_heading=h.1pxezwc) **Design Constraints 17**

[**7.**](#_heading=h.49x2ik5) **Software Quality Attributes 18**

[**8.**](#_heading=h.2p2csry) **User Interface 19**

[**9.**](#_heading=h.147n2zr) **Other Requirements 20**

**Introduction**

The Evaluation Management System is a comprehensive software solution designed to streamline and automate the evaluation process for trainees and trainers in a training program. It provides a centralized platform where trainees, trainers, and admin can register, create tasks, evaluate submissions, and generate final scores. The system aims to enhance efficiency, transparency, and accuracy in evaluating trainees' performance throughout the training program.

The system allows trainees and trainers to register by providing their relevant information. Trainees are assigned to specific batches within the system. Batch information such as batch number, schedule, and trainer details are stored to facilitate better organization and tracking of trainee progress.Trainers can create tasks or assignments within the system by specifying the task title and description.The system employs a weight-based strategy to generate the final score. The details of this strategy are explained in further detail within the system, facilitating the calculation of trainees' overall performance.

.

**Purpose**

The purpose of the Evaluation Management System project is to create a centralized software solution that automates and streamlines the evaluation process for trainees and trainers in a training program. The system aims to enhance the efficiency, transparency, and accuracy of evaluating trainees' performance throughout their training journey.Overall, the purpose of the project is to create a user-friendly and efficient Evaluation Management System that supports trainers, trainees, and managers in conducting fair, accurate, and comprehensive evaluations.

**Scope**

The Evaluation Management System provides a centralized platform for trainees, trainers, and managers to manage and track the evaluation process. It encompasses various modules, including trainee/trainer registration, task creation, submission evaluation, score generation, and performance tracking.

Functional Requirements:

1.Trainees and trainers can register by providing their personal information.

2.The system captures and stores relevant details, such as name, contact information, and role.

3.Trainers can assign trainees to specific batches.

4.Batch information, including batch number, schedule, and trainer details, is stored in the system

5.Trainers can create tasks or assignments on a daily basis.

6.Trainees can access and complete the tasks, with the option to submit file attachments.

7.Trainers evaluate trainees' task submissions manually.

8.The system allows trainers to analyze and provide feedback based on predefined criteria.

9.Trainers manually evaluate trainees' all assessments and all kinds of marking systems.

10.The system facilitates the assessment based on predefined criteria and assigns marks

accordingly.

11.The system generates the final score based on a weight-based strategy.

12.It considers evaluations from various components, such as tasks, projects, manager's

evaluation, aptitude test scores, and CEO office interview scores.

**Intended Stakeholder**

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

**References**

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

**Definitions, Acronyms, and Abbreviations**

| **Term/Acronym** | **Definition** |
| --- | --- |
| APP | Abbreviation of Application |
| API | Application Programming Interface |
| SRS | Software Requirement Specification |
| EMS | The Application Name |
|  |  |

**Overall Description**

The Evaluation Management System is a web-based application that serves as a comprehensive platform for managing and streamlining the evaluation process within a training program. It offers a centralized hub where trainees, trainers, and managers can seamlessly handle evaluation tasks and track trainee progress.The Evaluation Management System encompasses a range of essential functionalities to facilitate a seamless evaluation experience. Trainees and trainers can register and access the system, while trainers can create and assign daily tasks or assignments to trainees. Trainees can submit their completed tasks, and trainers evaluate and provide feedback based on predefined criteria. The system also supports the assessment of mini projects, mid projects, and final projects. Furthermore, trainers upload manager's evaluations, and scores from aptitude tests and CEO office/HR interviews can be included in the overall evaluation process.

**Overview**

This area of the SRS is all about the overall influences on the product and its specifications. This section does not mention particular criteria. Instead, it offers a context for those criteria, which are stated in full in section 3, and makes them simpler to grasp. Include stuff like as:

Product Perspective:

The Evaluation Management System is a web-based application that provides a comprehensive platform

for managing and streamlining the evaluation process in a training program. It serves as a centralized hub

where trainees, trainers, and managers can effectively handle evaluation tasks and track trainee progress.

The system aims to enhance efficiency, transparency, and accuracy throughout the evaluation process.

Product Functions:

The Evaluation Management System offers the following key features:

Trainee and Trainer Registration: Trainees and trainers can register within the system by providing their

necessary information, enabling access to the system's functionalities.

Batch Assignment: Trainers can assign trainees to specific batches, ensuring efficient organization and

management.

Task Creation: Trainers can create daily tasks or assignments for trainees, providing practical exercises to

apply their knowledge and skills.

Task Submission: Trainees can complete tasks and submit them, with the ability to attach files if required.

Evaluation: Trainers manually evaluate trainees' task submissions, providing feedback based on

predefined criteria.

Project Evaluation: Trainers assess trainees' mini projects, mid projects, and final projects based on

predefined criteria, analyzing their deliverables and performance.

Manager's Evaluation: Trainers upload evaluations based on specific parameters, assessing trainees'

professional skills and qualities.

Aptitude Test and Interview Score Upload: Trainers upload scores from trainees' aptitude tests and CEO

office/HR interviews, incorporating these assessments into the overall evaluation process.

Score Generation: The system generates the final score by considering evaluations from various

components, providing an overall assessment of trainees' performance.

User Characteristics:

The Evaluation Management System caters to trainees, trainers, and managers involved in the training

program. Users may have varying levels of technical expertise, ranging from basic computer literacy to

advanced skills. The system is designed to accommodate both tech-savvy users and those less familiar

with using software applications.

Constraints:

The Evaluation Management System development may be subject to the following constraints:

Time and Budget: The project must be completed within the specified time and budget constraints.

Data Availability: The system relies on the availability of relevant data and information required for

evaluation purposes.

Third-Party API Limitations: Live score updates may depend on third-party APIs, and the system must adhere to any limitations imposed by these APIs.

Legal and Ethical Restrictions: The use of player and team data must comply with legal and ethical

regulations.

Assumptions and Dependencies:

The Evaluation Management System operates based on the following assumptions and dependencies:

Users will have access to a web browser and a stable internet connection to use the system.

The system may utilize third-party APIs for certain functionalities, such as live score updates.

Data used within the system will be obtained from reliable sources and maintained with up-to-date

information.The development team possesses the necessary technical expertise and resources to

successfully complete the project.

**Technical platform**

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

Operating System:

The system should support the latest versions of these operating systems and

maintain compatibility with future updates.

Development Environment:

A suitable integrated development environment (IDE) such as Intellij Idea,Visual Studio will be required to

develop and debug the system.

Programming Languages:

The EMS may be developed using languages such as Java or Kotlin, which are commonly used for Android app development.Additional languages such as HTML, CSS, and JavaScript may be utilized for web-based components of the system.

Frameworks and Libraries:

Relevant frameworks and libraries such as Spring Boot, Hibernate, or JPA (Java Persistence API) may be used to support the development and functionality of the EMS.

Database:

An RDBMS such as MySQL, PostgreSQL, or Oracle may be used to store and manage data related to trainees, trainers, tasks, evaluations, and scores within the EMS.The database should be properly designed, normalized, and optimized to ensure efficient data management.

API Integration:

The EMS may require integration with external APIs to access additional functionalities or data sources, such as retrieving live cricket scores or accessing relevant training resources.Integration frameworks or libraries like Retrofit or Spring RestTemplate may be utilized to simplify API integration.

Security:

The EMS should implement appropriate security measures to protect user data and ensure secure communication between the system and users.Techniques such as encryption, secure authentication, and authorization mechanisms should be employed to ensure data privacy and prevent unauthorized access.

User Experience:

The EMS should focus on providing a user-friendly and intuitive interface for trainees, trainers, and managers.The user interface should be designed using technologies such as HTML, CSS, and JavaScript to ensure a responsive and interactive experience.

Scalability and Performance:

The EMS should be designed and optimized for scalability to handle a large number of trainees, tasks, evaluations, and scores.Techniques such as load balancing, caching, and database optimization should be employed to ensure optimal performance.

Testing Frameworks:The EMS should utilize testing frameworks such as JUnit or Mockito to automate testing processes and ensure the quality and reliability of the system.Unit testing, integration testing, and system testing should be performed to validate the functionality and behavior of the EMS.

Deployment Environment:

The EMS should be deployable on cloud platforms like AWS, Azure, or GCP to ensure scalability, availability, and easy access from different devices.Proper configuration and setup of the deployment environment should be performed to ensure the smooth operation of the system.

**Functional Requirements**

The functional requirements of this system are the characteristics and capabilities that the app must provide in order to satisfy the expectations of its user. Among the essential functional needs may be:

* The EMS should provide real-time updates on trainee evaluations, including scores, progress, and feedback.
* Trainers and managers should be able to view and track the evaluation process dynamically.
* Trainers should have the ability to create tasks or assignments and assign them to specific trainees or batches within the system.
* Trainees should be able to access and submit their completed tasks, including file attachments if necessary.
* Trainers should be able to manually evaluate trainees' task submissions and provide feedback based on predefined criteria.
* The system should support the analysis and assessment of mini projects, mid projects, and final projects, allowing trainers to assign marks and provide detailed evaluations.
* The system should calculate and display the overall manager evaluation score based on the provided criteria.
* Trainers should be able to upload scores from trainees' aptitude tests and CEO office/HR interviews.
* The system should incorporate these scores into the overall evaluation process and generate the final score accordingly.
* Trainees, trainers and admin should be able to create and manage their user accounts within the system.
* The system should generate reports and analytics on trainee performance, evaluation trends, and overall program effectiveness.
* Trainers and managers should have access to comprehensive data and visualizations to support decision-making and program improvements.
* The system should define different user roles (trainee, trainer, manager) with specific permissions and access levels to ensure data privacy and security.
* User roles should determine the actions and functionalities available to each user within the system.
* The EMS should have an intuitive and user-friendly interface, allowing users to navigate easily and access the required functionalities without difficulty.
* The interface should be responsive and optimized for both web and mobile devices.

These are some examples of the functional criteria that the web app must achieve in order to give its user with a complete and engaging experience.

**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 | Trainee/Trainer/Admin Registration | Users can register as admin trainees or trainers by providing their information.. |
| 2 | Assign Trainees to Batches | Admin can assign trainees to specific batches for organized management. |
| 3 | Assignment/Daily Task Creation | Trainers can create daily tasks or assignments for trainees to complete. |
| 4 | Daily Task Evaluation | Trainers can manually evaluate trainees' task submissions and provide feedback.. |
| 5 | Mini Project Evaluation | Trainers can evaluate trainees' mini project submissions based on predefined criteria.. |
| 6 | Mid Project Evaluation | Trainers can evaluate trainees' mid project submissions based on predefined criteria. |
| 7 | Final Project Evaluation | Trainers can conduct a detailed evaluation of trainees' final projects based on specific criteria. |
| 8 | Manager's Evaluation | Trainers can upload evaluations on various parameters for manager assessment. |
| 9 | Aptitude Test and Interview Score Upload | Trainers can upload scores from trainees' aptitude tests and CEO office/HR interviews. |
| 10 | Final Score Generation | The system generates the final score based on a weighted strategy. |

**Login/ Registration**

The Systems Login/Registration function is an essential component that enables users to access their individual information and preferences. Typically, the procedure consists of the following steps:

User registration: A user can create a new account by providing basic information such as their name, email address, and password.

Login: Once a user has created an account, they can log in using their email address and password.

Password recovery: If a user forgets their password, they can request a password reset via email.

User profile: After logging in, users can access their personal profile, which displays information such as saved teams and players, notification settings, and recent activity.

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

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_001 | A user can create a new account by providing basic information such as their name, email address, and password. | Essential | Server might not be available | TC\_001 |
| EMS\_002 | Once a user has created an account, they can log in using their email address and password. | Essential | Server might not be available | TC\_002 |
| EMS\_003 | If a user forget their password then they should be able to recover the password | Essential | User may not be registered. | TC\_003 |
| EMS\_004 | Users can log out of their account at any time by clicking the logout button in the app's menu. | Essential | User may not be logged in. | TC\_004 |

**Evaluation Management System**

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_005 | Trainers should be able to assign trainees to specific batches within the system for organized management. | Essential | None | TC\_005 |
| EMS\_006 | Trainers should be able to create and assign daily tasks or assignments to trainees for completion. | Essential | None | TC\_006 |
| EMS\_007 | Trainers should be able to manually evaluate trainees' task submissions and provide feedback based on predefined criteria | Essential | None | TC\_007 |
| EMS\_008 | Trainers should be able to evaluate trainees' mini project submissions based on predefined criteria. | Essential | None | TC\_008 |
| EMS\_009 | Trainers should be able to evaluate trainees' mid project submissions based on predefined criteria. | Essential | None | TC\_009 |

**Evaluation Management System**

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_010 | Trainers should be able to conduct a detailed evaluation of trainees' final projects based on specific criteria. | Essential | None | TC\_010 |
| EMS\_011 | Trainers should be able to upload evaluations on various parameters for manager assessment | Essential | None | TC\_011 |
| EMS\_012 | Admin should be able to upload scores from trainees' aptitude tests and CEO office/HR interviews as part of the overall evaluation process. | Essential | None | TC\_012 |
| EMS\_013 | Admin also Submitted Quiz mark’s based on their weighted strategy individually | Essential | None | TC\_013 |
| EMS\_014 | The system should generate the final score based on a weighted strategy. | Essential | None | TC\_014 |
| EMS\_015 | Trainer also submitted comments based on the trainee.How was the trainee? | Essential | None | TC\_015 |

**View Score**

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_016 | Trainee,Trainer,admin can view the score individually and combinedly. | Essential | None | TC\_016 |
| EMS\_017 | There will be a remarks option.Admin and trainee also remarks trainees marks. | Essential | None | TC\_017 |

**Users Profile**

###### **Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_018 | The user profile should contain essential details about the user, such as their full name, email address, contact number, and trainee/trainer ID.. | Essential | None | TC\_018 |
| EMS\_019 | The profile should clearly indicate the role or designation of the user, such as trainee, trainer, or admin. This information helps define user access privileges and determines the actions they can perform within the system. | Essential | None | TC\_019 |
| EMS\_020 | The profile can showcase the user's evaluation history, including the tasks, projects, or evaluations they have completed. This helps users review their past performance and identify areas of improvement. | Essential | None | TC\_020 |
| EMS\_021 | The user profile should include options for privacy and security settings.Needs to define who can access which option. | Essential | May require regular updates to ensure accuracy of information | TC\_021 |

**User Interface**

| **UI No.** | **UI Name** | **Related Function Requirement ID** | **Description** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_UI\_001 | Home Screen | EMS\_Function\_001 | The main screen of the system displays the latest scores, news, and analysis of recent evaluations. | TC\_022 |
| EMS\_UI\_002 | Trainee/Trainer Dashboard | EMS\_Function\_002 | A dashboard providing an overview of the trainee's/trainer's assigned tasks, evaluation progress, and other relevant information. | TC\_023 |
| EMS\_UI\_003 | Batch Management | EMS\_Function\_003 | A screen allowing admin to manage batches, including creating new batches, assigning trainees, and viewing batch details. | TC\_024 |
| EMS\_UI\_004 | Task/Assignment Creation | EMS\_Function\_004 | A screen where trainers can create tasks or assignments, specifying details such as title, description, and submission deadline. | TC\_025 |
| EMS\_UI\_005 | Task/Assignment Submission | EMS\_Function\_005 | A screen enabling trainees to submit their tasks or assignments with file attachments. | TC\_026 |
| EMS\_UI\_006 | Task Evaluation | EMS\_Function\_006 | A screen for trainers to evaluate trainees' task submissions, analyzing and assigning scores based on predefined evaluation criteria. | TC\_027 |
| EMS\_UI\_007 | Final Score Display | EMS\_Function\_007 | A screen displaying the final scores generated for trainees, reflecting their overall performance and progress throughout the evaluation process | TC\_028 |
| EMS\_UI\_008 | Manager's Evaluation | EMS\_Function\_008 | A screen where trainers can upload the manager's evaluation, providing scores and feedback on various parameters such as communication skills and work ethic. | TC\_29 |
| EMS\_UI\_009 | User Account | EMS\_Function\_009 | A screen allowing users to create and manage their accounts, including profile information, preferences, and notification settings. | TC\_30 |

**Non-Functional Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_022 | The system should have a fast and responsive user interface with minimal latency. The average response time should not exceed 2 seconds, and the maximum response time should not exceed 5 seconds. | Essential | May require optimization of the code and the use of efficient algorithms to ensure good performance | TC\_31 |
| EMS\_023 | The system should use appropriate encryption techniques to protect user data. It should also have a secure login system to prevent unauthorized access. | Essential | May require regular security audits and updates to ensure protection against emerging threats | TC\_32 |
| EMS\_024 | The web app should have a user-friendly interface with clear navigation and intuitive controls. It should be easy to use for users of all ages and skill levels. | Essential | May require regular user testing and feedback to improve the user experience | TC\_33 |
| EMS\_025 | The web app should be compatible with a range of popular web browsers and should work seamlessly on different screen sizes and resolutions. | Essential | Regular testing is performed on different browsers, screen sizes, and resolutions to ensure compatibility and address any limitations or constraints. | TC\_34 |

**Performance Requirements**

Response Time:

Average response time of transactions should be less than 2 seconds.

Maximum response time of transactions should not exceed 5 seconds.

Throughput:

The application must be able to process at least 50 requests per second.

Capacity:

The application must support a minimum of 5,000 concurrent users.

Degradation Modes:

In the event of a degraded network connection, the application should convert to offline mode while still

allowing users to read scores and headlines.

Capital Utilization:

Memory usage should not exceed 250 MB.

Optimizing disk utilization will reduce storage use.

The use of communications should be improved to reduce data consumption.

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

Reliability:

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

Security:

Encryption and secure authentication measures should be implemented to protect the web application

against unauthorized access and data breaches.

User data should be securely stored and transmitted using appropriate encryption protocols.

Scalability:

The web application should be designed to handle an increase in the number of users and request/response

without a significant decrease in performance.

The architecture and infrastructure should be scalable to accommodate future growth and additional

functionalities.

Compatibility:

The web application should be compatible with a wide range of web browsers and devices, ensuring

consistent performance and functionality.

It should support various screen sizes and resolutions for a responsive and adaptive user experience.

Support:

Sufficient user assistance should be provided, including documentation, FAQs, and a method for reporting

problems and difficulties.

The application should be regularly reviewed and updated to maintain optimal performance and address any

issues that may arise.

Usability:

The user interface (UI) of the web application should be intuitive, user-friendly, and easy to navigate.

Accessibility features, such as support for screen readers and high-contrast mode, should be implemented to

ensure accessibility for users with impairments.

Customizability:

Users should have the ability to personalize their experience by customizing settings such as themes, text

size,and notification preferences.

Reporting and Analytics:

The web application should provide information and insights into usage trends, user behavior, and

performance indicators.

Real-time analytics and reporting features should be available to help identify performance issues and

improve the user experience.

Localization:

The web application should support multiple languages to cater to a diverse user base.

Integration:

The application should be able to integrate with third-party services and platforms, such as social networking

and advertising, to enhance the user experience and provide additional functionalities.

**Safety Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_026 | The web app should not expose users to any potential harm or danger, including physical harm or psychological harm. | Essential | May require regular user testing and feedback to ensure safety | TC\_035 |
| EMS\_027 | The web app should protect the privacy and security of user data, including personal information and login credentials. | Essential | May require regular security audits and updates to ensure data safety | TC\_036 |
| EMS\_028 | The web app should not cause damage to the device or equipment being used to access the app. | Essential | May require regular testing and compatibility checks to ensure equipment safety | TC\_037 |

**Security Requirements**

To prevent unauthorized or malicious access, use, modification, destruction, or disclosure of software, the following security measures may be implemented:

Utilize Encryption: Encrypt sensitive data such as user passwords, financial transactions, and other secret information to prevent unwanted access. A powerful encryption method, such as AES or RSA, should be used by the application to guarantee data security.

Access Control: The application should have an authentication method to guarantee that only authorized users may access sensitive data. This may consist of a login procedure, password rules, and multi-factor authentication.

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.

Auditing and Logging: The application should preserve logs and historical data sets to monitor user activity and identify unusual conduct. These logs should be checked and analyzed on a regular basis to detect possible security concerns.

Communications between the web app and the server should be encrypted using secure protocols such as SSL/TLS for network security. In order to prevent unwanted access to critical data, the web app should additionally limit communications between certain software components.

Integrity of Data: The application should verify the integrity of crucial variables to guarantee that data is not changed or damaged during transmission. SHA-256 and other hashing algorithms may be used to verify the integrity of data.

Regular Security Updates: The application should be routinely updated to patch any security flaws and maintain the safety of sensitive data.

These are some of the precautions that may be taken to prevent unauthorized or malicious access, use, modification, destruction, or disclosure of software. Depending on the type and scope of the app and the intended audience, the particular security needs may vary.

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_029 | The web app should use appropriate encryption techniques to protect user data, such as passwords and personal information. | Essential | May require regular security audits and updates to ensure data encryption remains secure | TC\_038 |
| EMS\_030 | The web app should have a secure login system to prevent unauthorized access to user accounts. | Essential | May require regular security audits and updates to ensure the login system remains secure | TC\_039 |
| EMS\_031 | The web app should check the integrity of critical data to ensure that it has not been modified or tampered with. | Essential | May require regular security audits and updates to ensure data integrity | TC\_040 |
| EMS\_032 | The web app should have mechanisms in place to detect and respond to potential security threats, such as malware or unauthorized access attempts. | Essential | May require regular security audits and updates to ensure threat detection remains effective | TC\_041 |

**Design Constraints**

The design constrains for this app are as bellow:

The software languages for the Evaluation Management System may include Java or SpringBoot for the main application code and XML for the user interface design. The development tools that can be used may include Visual Studio, a popular Integrated Development Environment (IDE) for Web application development.

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

The design constraints for the web app have been outlined in the previous answer and may include considerations such as platform compatibility, user experience, performance, data management, scalability, security, and compliance.

Purchased components may include third-party libraries or APIs for specific functionalities, such as accessing live cricket scores or news feeds. The use of purchased components should be carefully evaluated to ensure that they meet the performance, security, and compatibility requirements of the app.

Some other constraints can be like :

Platform Compatibility: The web app should be compatible with the latest version of the Android operating system and should support a range of device types and screen sizes.

User Experience: The web app should provide an engaging and intuitive user experience, with clear navigation and accessible features. The design should be visually appealing and consistent throughout the app.

Performance: The web app should have a fast and responsive user interface, with minimal lag or delay. The app should also have optimized performance, with efficient use of memory, processing power, and network resources.

Data Management: The web app should have efficient data management, with the ability to handle large amounts of data and provide fast access to critical information. The app should also have the ability to store and retrieve data offline, where necessary.

Scalability: The web app should be scalable to accommodate a growing user base and increased usage. The design should take into account the possibility of adding new features and functionality in the future.

Security: The web app should meet the security requirements outlined in the previous answer, including the protection of sensitive information and the prevention of unauthorized access.

Compliance: The web app should comply with relevant legal and regulatory requirements, such as data privacy laws and industry standards.

These are some of the design constraints for the Evaluation Management System. The specific constraints may vary based on the nature and scope of the web app and the target audience. The design should take into account these constraints to ensure a high-quality and secure web app that meets the needs of users.

**Software Quality Attributes**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_033 | The web app should be easy to use and intuitive for users, with a clear and straightforward interface. | Essential | May require regular user testing and feedback to refine usability | TC\_042 |
| EMS\_034 | The web app should have a fast and responsive interface, with minimal lag or delay in interactions. | Essential | May require regular performance testing and optimization to ensure high performance | TC\_043 |
| EMS\_035 | The web app should be able to handle an increasing number of users and transactions without degradation in performance. | Essential | May require regular testing and scalability improvements to ensure scalability | TC\_044 |
| EMS\_036 | The web app should have high availability and minimal downtime, with the ability to recover from failures and errors. | Essential | May require regular testing and reliability improvements to ensure reliability | TC\_045 |

**User Interface**

| **UI No.** | **UI Name** | **Related Info No.** | **Notes** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS |  |  |  | TC\_046 |
|  |  |  |  | TC\_047 |
|  |  |  |  | TC\_048 |
|  |  |  |  | TC\_049 |

**Other Requirements**

| **REQUIREMENT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| EMS\_037 |  |  |  | TC\_050 |
| EMS\_038 |  |  |  | TC\_051 |
| EMS\_039 |  |  |  | TC\_052 |
| EMS\_040 |  |  |  | TC\_053 |