|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  **[BloodBase – A research project]**  **Version: [1]**   |  |  | | --- | --- | | Project Code | F24-63 | | Supervisor | Mr Ali Naseer Shah | | Co Supervisor | Dr Farrukh Shahid | |  |  | | Project Team | Huzaifa K21-3559  Robas Ahmed Shah K21-3613  Yusra Sohail Siddiqui K21-3446 | | Submission Date | 12-12-2024 | |

[Instructions]

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.
* MS-Word Reviewing feature must be used to get the document reviewed by supervisors or co-supervisors.

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1 | Huzaifa | 6-12-2024 | [Document Created] |
| 1 | Huzaifa | 12-12-2024 | [Added User story] |
|  |  |  | [Added UseCase x.x.xx] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Muhammad Ali Naseer Shah | | Supervisor |
| Dr Farrukh Shahid | | Co- Supervisor |
|  | |  |

Document Sign-Off

[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1.](#_heading=h.30j0zll) Introduction 7

[1.1.](#_heading=h.3znysh7) Purpose of Document 7

[1.2.](#_heading=h.2et92p0) Intended Audience 7  
1.3 Abbreviations ………………………………………………………………………………………...7

[1.4.](#_heading=h.tyjcwt) Document Convention 7

[2.](#_heading=h.3dy6vkm) Overall System Description 8

[2.1.](#_heading=h.1t3h5sf) Project Background 8

[2.2.](#_heading=h.4d34og8) Project Scope 8

[2.3.](#_heading=h.2s8eyo1) Not In Scope 8

[2.4.](#_heading=h.17dp8vu) Project Objectives 8

[2.5.](#_heading=h.3rdcrjn) Stakeholders 8

[2.6.](#_heading=h.26in1rg) Operating Environment 8

[2.7.](#_heading=h.35nkun2) System Constraints 8

[2.8.](#_heading=h.1ksv4uv) Assumptions & Dependencies 8

[3.](#_heading=h.2jxsxqh) External Interface Requirements 9

[3.1.](#_heading=h.z337ya) Hardware Interfaces 9

[3.2.](#_heading=h.3j2qqm3) Software Interfaces 9

[3.3.](#_heading=h.1y810tw) Communications Interfaces 9

[4.](#_heading=h.4i7ojhp) Functional Requirements 10

[4.1.](#_heading=h.2xcytpi) Functional Hierarchy 10

[4.2.](#_heading=h.1ci93xb) Use Cases 10

[4.2.1.](#_heading=h.3whwml4) [Title of use case] 10

[5.](#_heading=h.2bn6wsx) Non-functional Requirements 11

[5.1.](#_heading=h.qsh70q) Performance Requirements 11

[5.2.](#_heading=h.3as4poj) Safety Requirements 11

[5.3.](#_heading=h.1pxezwc) Security Requirements 11

[5.4.](#_heading=h.49x2ik5) User Documentation 11

[6.](#_heading=h.147n2zr) References 12

[7.](#_heading=h.3o7alnk) Appendices 13

1. **Introduction**

* 1. **Purpose of Document**

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed and structured outline of the functional and non-functional requirements for the performance management system. This document serves as a comprehensive guide for all stakeholders, including business users, developers, and technical teams, to ensure a shared understanding of the system’s goals, functionalities, and constraints.

* To provide a clear communication framework among stakeholders
* To act as a reference document for developers during the system’s design, implementation, and testing phases.
* To ensure that the system’s design and implementation align with the business objectives
* To establish a foundation for requirement traceability throughout the project lifecycle
  1. **Intended Audience**
* Fast NU
* Jury
* Supervisor (Muhammad Ali Naseer Shah)
* Co-Supervisor (Dr. Farrukh Shahid)
* Students of Fast NU
* Our Team(Designer, Developer, Tester)
* Potential Users of this product

**1.3 Abbreviations**

* HOD = Head Of Department
* KPI = Key Performance Indicators
  1. **Document Convention**
* Font Family = Arial
* Font Size = 12 for headings, 10 for the rest of the content

1. **Overall System Description**
   1. **Project Background**

Blood-Base is a blockchain-based platform designed to improve transparency and security in Pakistan's blood management system. It tracks blood donations from donor registration to transfusion, ensuring real-time data and reducing risks like illegal blood trading. By integrating with healthcare systems and automating processes through smart contracts, Blood-Base enhances the efficiency, accountability, and availability of safe blood supplies.

**Problem Statement:**

* Subjectivity and biases in traditional appraisal systems.
  1. **Project Scope**

This system aims to streamline organizational performance management using AI and blockchain technologies, ensuring transparency, efficiency, and fairness. The primary goal is to provide an integrated platform for employees, HODs, and Admins to manage performance evaluation, attendance, research paper verification, and promotion criteria effectively.

**Included Functionalities:**

* Employee and HOD access to personal and organizational performance data.
* AI-driven evaluation of employees based on performance metrics.
* Verification of research papers using external APIs (e.g., CrossRef).
* Attendance and activity tracking, including active and passive time monitoring.
* Blockchain-based promotion criteria for transparency and accountability on multichain.
* CRUD operations for user management by Admins.

**Excluded Functionalities**:

* **there is a significant shortage of blood donors**

**compared to patients with only 10% of donors being**

**voluntary**

* **The current systems in place are poorly digitized**
* **make manipulation and fabrication of records straightforward**
* **The methods of blood extraction and storage at these banks are of subpar quality**
  1. **Not In Scope**

Creating a working application with real-time data analysis at this moment. Deploying a market ready product with well figured backend set. Becoming a bloodbank in it self and performing blood transfusion

* 1. **Project Objectives**

**Employee Functions**:

* View personal stats, including attendance, activity, and research paper contributions.
* Verify research papers automatically using external APIs.
* Mark attendance and active/passive times automatically.
* Submit and track project statuses.
* Access evaluation scores generated by the AI model.

**HOD Functions**:

* View detailed stats for employees under their supervision.
* Evaluate employees using an AI model with configurable weights.
* Automatically track the number and category of research papers per employee.
* Define promotion criteria via blockchain-based smart contracts.

**Admin Functions**:

* Full CRUD operations for managing users (employees and HODs).
* Ensure system stability and compliance with organizational policies.
  1. **Stakeholders**
* Employees
* HODs
* Software Developer
* HR
* DataBase Administrator
* organization(Fast NU)
  1. **Operating Environment**

The software will operate in an office environment where there is network availability and blockchain availability on the multichain and node is currently ready and running with correct credentials. The system should have xampp, mySQL and our AI model locally loaded and ready and running. Crossref API must be running as it is available open source for free that won't be an issue

* 1. **System Constraints**
* Database Management System: MySQL (v8.0 or later).
* AI model frameworks: TensorFlow (v2.x) or PyTorch (v1.x).
* Blockchain Network: Multichain
* Third-party tools and APIs, such as CrossRef
* Any downtime in the blockchain network may delay promotion criteria updates.
* Real-time attendance and activity tracking depend on uninterrupted communication with external systems like mobile devices or hardware sensors.
* Network dependency for real-time functionalities such as data sync and blockchain interaction. Good internet and connection is necessary
* no cultural constraint as main point of language is English which is widely spoken and understood by our intended audience
* Any AI-based evaluations must avoid bias and adhere to ethical AI guidelines.
* Research paper verification must respect copyright and intellectual property rights.
* The system must operate in potentially noisy environments, such as offices, where audible notifications may not be feasible; silent or vibration-based notifications should be used.
* For employees with limited tech knowledge, the system must provide easy-to-navigate interfaces and detailed help sections.
* The user interface must be intuitive, with more graphical elements and fewer textual descriptions to cater to users with diverse technical proficiencies.
  1. **Assumptions & Dependencies**
* Employees and HODs will input accurate and truthful data for research paper submission and performance tracking.
* Employees will honestly validate that the task they are performing are doen and after that only their status was changed from pending to done as there is no possible way to check if FYP supervised are completed or not although data is sent to the HOD so manual verification can be done by HOD
* Employees know the DOI or Title and Journal name of their research paper when inserting it
* Weights configured by HODs for AI evaluation are logical and align with organizational policies.
* Users (employees, HODs, Admins) have basic knowledge of using web and mobile applications.
  1. **Dependencies**
* **CrossRef API**: Required for verifying research paper details; its availability and API rate limits may impact the system.
* Dependencies on frameworks such as Flask,Php,Mysql,Xammp
* MySQL database availability and efficient query handling are critical for storing and retrieving system data.
* availability of tensor flow for our AI model

1. **External Interface Requirements**

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

* 1. **Hardware Interfaces**
* Desktop devices: Minimum 4GB RAM, dual-core processor, running Windows 10, macOS 10.13, or Linux (kernel 4.x or later).
* any device that has a web browser installed in it with all dependencies and libraries of python installed and ready to run.
* our AI model will be stored locally on the device to reduce latency issues.
  1. **Software Interfaces**
* MySQL (v8.0): Stores employee, HOD, and Admin data, including research papers, attendance, and activity records.
* Data exchange through structured queries using mysql.connector for Python.
* multichain
* AI model Deployed locally using TensorFlow (v2.x) or PyTorch (v1.x).
* Flask: Web framework for backend services.
* **CrossRef API**: For verifying research papers using DOI or title.
* SMTP Servers: For email notifications.

* 1. **Communications Interfaces**
* JSON is used for data exchange between client and server.
* Smart contracts are written in Solidity and deployed on the blockchain.
* Automatic data sync occurs every 15 minutes between the local database and the blockchain to ensure consistency.

1. **Functional Requirements**
   1. **Functional Hierarchy**

**1. User Management (Admin)**1.1 Add Users  
1.2 Edit User Details  
1.3 Delete Users  
1.4 View All Users

**2. Employee Functionality**2.1 View Personal Statistics  
2.1.1 Attendance Records  
2.1.2 Active/Passive Time  
2.1.3 Number of Research Papers  
2.1.4 Research Paper Categories  
2.1.5 FYP Supervision Records  
2.2 Verify Research Papers  
2.3 Mark Attendance Automatically  
2.4 Track Project Status  
2.4.1 Submit New Projects  
2.4.2 View Pending Projects  
2.4.3 View Completed Projects  
2.5 View Evaluation Scores

**3. HOD Functionality**3.1 View Employee Statistics  
3.1.1 Attendance Records  
3.1.2 Activity Logs  
3.1.3 Research Paper Details  
3.2 Evaluate Employees  
3.2.1 Use AI Model for Evaluation  
3.2.2 Configure AI Weights  
3.3 Write Promotion Criteria  
3.3.1 Publish as Smart Contract on Blockchain  
3.3.2 Viewable by All Users

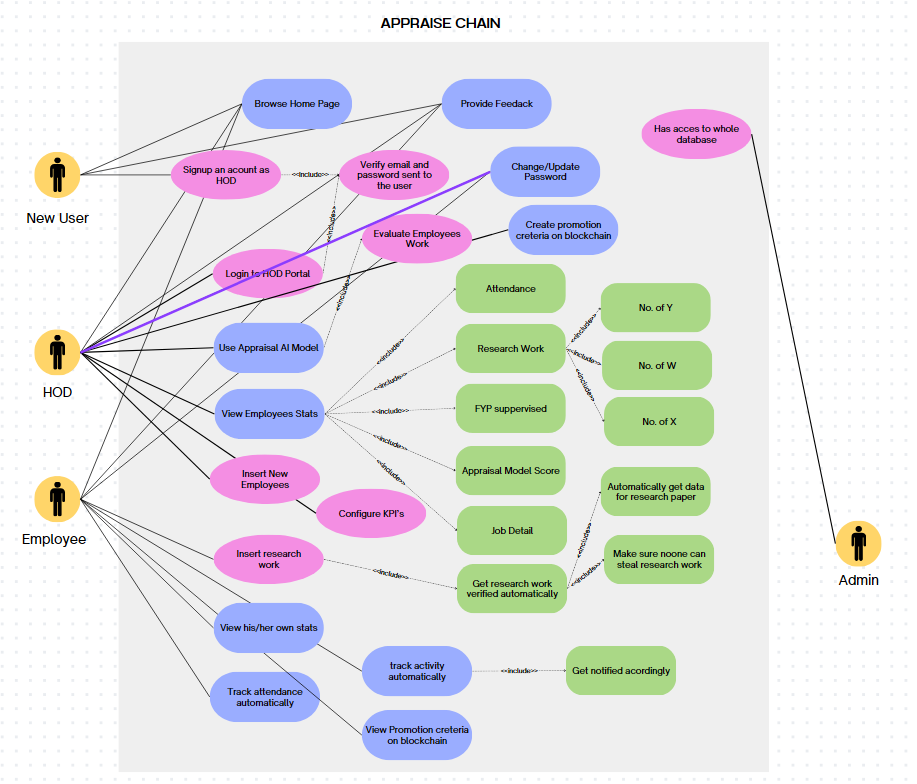
**4. AI Evaluation**4.1 Score Calculation  
4.1.1 Attendance Impact  
4.1.2 Activity Analysis  
4.1.3 Research Paper Quality  
4.2 Weight Configuration

**5. Research Paper Management**5.1 Submission of Research Papers  
5.2 Verification of Research Papers  
5.3 Research Paper Categorization  
5.4 Citation Count Analysis

**6. Blockchain Integration Using Multichain**6.1 Publish Promotion Criteria  
6.2 Ensure Transparency and Immutability  
6.3 Secure Access to Criteria Details

**7. Attendance Management**7.1 Automatic Marking of Attendance  
7.2 Daily Attendance Logging for Employees  
7.3 Calculation of Attendance Percentage

* + 1. **[Use Case Diagram]**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-HOD-01>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **HOD**: Initiates the use case by logging into the system. | | | | |
| **Feature:** View Employee Statistics(attendance,research work,fyps and more) | | | | |
| **Pre-condition:** | | HOD must have a valid account in the system.  HOD must log in with a correct email/CNIC and password.  The system must have employee stats already recorded. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | HOD accesses the login page. | | | Displays the login form. |
| **2.** | HOD enters email/CNIC and password. | | | Validates credentials. |
| **3.** | HOD submits the login form. | | | Authenticates HOD and redirects to the dashboard. |
| **4.** | HOD selects the "View Employee Stats" option. | | | Retrieves and displays a list of employees and their stats. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.**    **2a: If no employee stats are available, the system displays a message: “No data available.”** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | HOD successfully logs into the system. | | | |
|  | Employee statistics are displayed for review. | | | |
|  | HOD can proceed to use other use cases (e.g., evaluate employees). | | | |
| **Use Case Cross referenced** | | | **UC-HOD-02**: HOD evaluates employees using AI.  **UC-HOD-03**: HOD configures weights for the AI model.  **UC-HOD-04**: HOD views the number of research papers written by employees.  **UC-EMP-01**: Employee views their own stats.  **UC-EMP-02**: Employee verifies research papers automatically. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-HOD-02>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **HOD**: Initiates the evaluation. | | | | |
| **Feature:** AI-Based Employee Evaluation | | | | |
| **Pre-condition:** | | HOD must log in successfully.  Employee stats and AI model must be pre-configured. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | HOD selects "Evaluate Employee" from the dashboard. | | | Displays a list of employees. |
| **2.** | HOD clicks "Evaluate." | | | AI model calculates and displays employee scores. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If employee data is incomplete, the system prompts the HOD to add missing information.**    **2a: If no employee stats are available, the system displays a message: “No data available.”** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Employee scores are calculated and displayed. | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | * **UC-HOD-01**: HOD views employee stats. * **UC-HOD-03**: HOD configures weights for the AI model. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-HOD-03>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **HOD**: Configures the weights. | | | | |
| **Feature:** AI Weight Configuration according to our KPIs | | | | |
| **Pre-condition:** | | HOD must log in successfully.  Access to the AI weight configuration page is required. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | HOD navigates to "Configure KPI’s" | | | Displays current weight settings. |
| **2.** | HOD updates weights. | | | Validates and saves the new configuration. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid weights are entered, the system displays an error message and prompts re-entry.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Weights are updated and applied to the AI model. | | | |
|  | Weights are updated and applied to the AI model. | | | |
|  | Weights are updated and applied to the AI model. | | | |
| **Use Case Cross referenced** | | | **UC-HOD-02**: HOD evaluates employees using AI.  **UC-HOD-03**: HOD configures weights for the AI model.  **UC-HOD-04**: HOD views the number of research papers written by employees.  **UC-EMP-01**: Employee views their own stats.  **UC-EMP-02**: Employee verifies research papers automatically. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-01>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Accesses stats. | | | | |
| **Feature:** View Employee Statistics(attendance,research work,fyps and more) | | | | |
| **Pre-condition:** | | Employee must log in successfully. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee logs in and navigates to "My Stats.". | | | Displays employee-specific statistics. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Employee views their attendance, activity, and research stats. | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-02>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Initiates the verification process of research paper automatically | | | | |
| **Feature:** Research Paper Verification automatically | | | | |
| **Pre-condition:** | | Employee must log in successfully. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee selects "Verify Research Paper." | | | System validates and updates status. |
| **2.** | enters DOI or Title. | | | System validates and updates status. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.**    **2a: if author name doesn't match so paper is invalid** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Paper is valid and inserted | | | |
|  | or Paper invalid and not inserted | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | .  **UC-EMP-01**: Employee views their own stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-03>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Logs attendance. | | | | |
| **Feature:** Automatic Attendance | | | | |
| **Pre-condition:** | | Employee must be logged in. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee logs in. | | | Attendance for the day is automatically marked. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | attendance marked automatically | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | **UC-EMP-01**: Employee views their own stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-HOD-04>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **HOD**: Initiates the process. | | | | |
| **Feature:** View Employee Research Papers | | | | |
| **Pre-condition:** | | HOD must have a valid account in the system.  HOD must log in with a correct email/CNIC and password.  The system must have employee stats already recorded. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | HOD selects "View Research Papers" on the dashboard. | | | Displays a list of employees. |
| **2.** | HOD selects "View Research Papers" on the dashboard. | | | Displays the number of research papers written by the employee, categorized by type. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.**    **2a: If no employee stats are available, the system displays a message: “No data available.”** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | HOD can view the total number of research papers written by an employee. | | | |
|  | HOD can view the total number of research papers written by an employee. | | | |
|  | HOD can view the total number of research papers written by an employee. | | | |
| **Use Case Cross referenced** | | | **UC-HOD-01**: HOD views employee stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-HOD-05>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **HOD**: Writes criteria.  **Blockchain**: Stores the criteria for transparency. | | | | |
| **Feature:** Promotion Criteria Management | | | | |
| **Pre-condition:** | | HOD must log in successfully.  Access to the blockchain network must be established. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | HOD navigates to "Promotion Criteria." | | | Displays the input form for criteria. |
| **2.** | HOD writes the promotion criteria. | | | Encrypts and sends the criteria to the blockchain. |
| **3.** | HOD submits the form. | | | Confirms the criteria has been stored on the blockchain. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If invalid credentials are entered, the system displays an error message and prompts re-entry.**    **2a: IIf the blockchain is unavailable, the system displays an error message.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Promotion criteria are securely stored on the blockchain. | | | |
|  | this will be taken in account when our AI model will evaluate employees | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | **UC-HOD-01**: HOD views employee stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-04>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Accesses activity stats. and get them marked automatically | | | | |
| **Feature:** **TRACK** View Activity Statistics | | | | |
| **Pre-condition:** | | Employee must log in successfully.  Active/passive time data must be stored in the system.  Flask code for automatic tracker must be running | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee logs in and selects "View My Activity." | | | Displays active and passive time details. |
| **2.** | employee logs in | | | tracking starts |
| **3.** | if employee is on a break | | | tracking stops |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a: If no activity data is available, the system displays “No activity data recorded.”** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Employee can view their activity stats. | | | |
|  | Employee can tracktheir activity stats. | | | |
|  | . | | | |
| **Use Case Cross referenced** | | | **UC-EMP-01**: EMPviews employee stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-05>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Submits and views project status. | | | | |
| **Feature:** Project Management | | | | |
| **Pre-condition:** | | Employee must log in successfully.  Project management functionality must be active. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee selects "Insert Project" and fills in details. | | | Validates and stores the project data. |
| **2.** | Employee submits the form. | | | Displays a confirmation and the project’s status (Pending/Completed). |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a:If project data is incomplete, the system prompts the employee to fill in the missing details.** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Project is recorded and its status is updated. | | | |
|  | Project is recorded and its status is updated. | | | |
|  | Project is recorded and its status is updated. | | | |
| **Use Case Cross referenced** | | | **UC-EMP-01**: EMP views employee stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-EMP-06>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Employee**: Requests evaluation score. | | | | |
| **Feature:** View Evaluation Score | | | | |
| **Pre-condition:** | | Employee must log in successfully.  AI evaluation score must have been calculated previously. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Employee selects "View Evaluation Score." | | | Displays the score and details about the evaluation. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **1a:If not evaluated no score will be shown** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Employee can view their score and evaluation breakdown. | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | **UC-EMP-01**: EMP views employee stats. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<UC-ADMIN-01>** | | | | |
| **Use case Id:** | | Write use case reference number. | | |
| **Actors:**  **Admin**: Initiates all operations. | | | | |
| **Feature:** User Management and System Maintenance | | | | |
| **Pre-condition:** | | Admin must log in successfully.  Admin must have unrestricted access to all system functionalities. | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Admin selects "Manage Users" from the dashboard. | | | Displays a list of all users with options to add, edit, delete, or view details. |
| **2.** | Admin chooses to add a new user. | | | Displays a form for entering user details (name, role, credentials, etc.). |
| **3.** | Admin submits the new user details. | | | Validates the data and adds the user to the database. Confirms the operation. |
| **4.** | Admin selects an existing user for editing. | | | Displays the user's details for modification. |
| **5.** | Admin submits updated user details. | | | Validates the data and updates the user's record. Confirms the operation. |
| **6.** | |  | | --- | | Admin selects a user for deletion. |  |  | | --- | |  | | | | Prompts for confirmation and deletes the user from the database upon approval. |
| **7.** | Admin monitors system performance. | | | displays system health stats (e.g., database status, server uptime, error logs). |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **1a:**   |  | | --- | | **If the entered user data is invalid, the system displays an error message and highlights issues.** |  |  |  | | --- | --- | | **2a** | **If the user to be deleted has dependencies (e.g., assigned tasks), the system prevents deletion and displays a warning.** |  |  |  | | --- | --- | | **3a** | **If the system detects an issue, it alerts the admin with details for resolution.** | | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
|  | Users are successfully added, updated, or removed as needed. | | | |
|  | System performance is monitored and issues are resolved to maintain smooth operation. | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | UC-HOD-01: HOD views employee stats.  UC-EMP-01: Employee views personal stats. | |

1. **Non-functional Requirements**
   1. **Performance Requirements**

* The system should handle up to more than 1 concurrent users without noticeable lag in processing or response time.
* The system should retrieve and display employee statistics within 2 seconds of a request.
* CRUD operations (Create, Read, Update, Delete) on the database should execute within 1 second on average.
* The AI evaluation process should produce results for an employee within 15 seconds after data submission.
* Blockchain-related actions (e.g., writing promotion criteria) must confirm within 10 seconds, adhering to blockchain network constraints.
* The system must maintain 99.9% uptime to ensure reliability for critical operations like attendance tracking and research paper verification.

* 1. **Safety Requirements**
* The system should prevent unauthorized deletion or modification of critical data (e.g., employee records, research papers).
* A backup mechanism must be implemented to automatically store data snapshots daily, ensuring recovery in case of system failure.
* Error handling must prevent the system from crashing under invalid inputs or unexpected actions by users.
* Safeguards must be in place to prevent accidental overwriting of promotion criteria or AI configuration weights.
* Ensure compliance with workplace safety standards for data handling, avoiding harm caused by incorrect evaluations or statistics.
  1. **Security Requirements**
* All users must authenticate with **multi-factor authentication** (MFA) to access their respective dashboards.
* The system must encrypt sensitive data (e.g., user credentials, employee performance scores) using **AES-256 encryption**.
* Admins, HODs, and employees should have role-based access control (RBAC), ensuring they only access authorized features.
* All communication with the blockchain and external systems must use **encryption**
* Employee data privacy will be optimal
  1. **User Documentation**
* A **User Manual** will guide Admins, HODs, and Employees through system features and functionalities.
* **Online Help** integrated into the system will provide step-by-step instructions for using key features, accessible via home page.
* **Tutorial Videos** will be available for complex operations, such as configuring AI weights or writing promotion criteria on the blockchain.
* **Context-Sensitive Help** will display relevant guidance based on the user’s current action or page.
* A **Troubleshooting Guide** will help users address common issues such as login errors or failed transactions.

1. **References**

[This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. (This section is like the bibliography in a published book).]

1. **Appendices**

[This section should include supporting detail that would be too distracting to include in the main body of the document.]