Software Requirements Specification (SRS) Document

Project Name : M&E Dashboard
Team Number: 35

Team Members:

Nikhil - 2023101084
Viswas - 2023101008
Gautam - 2023101032
Anvithraj - 2023101023
Saubhanik - 2023115004

Brief problem statement

The goal of the project is to build a dashboard that allows clients to see the crucial details like milestones, progress and the status of the project. The dashboard contains visualizations which shows the real time comparison and progress of the project. The web app should also allow the clients to see their project reports and be able to upload relevant documents and provide messaging and feedback feature to give a direct connection between clients and the project (Working) team.

System requirements

- **Hardware Requirements:**

- **Software Requirements:**
  - Node.js (LTS version)
  - npm (for Package Management)
  - MongoDB (local installation for development and testing)
- **Technologies:**
  - **Database :** MongoDB
    - Local for testing
    - Atlas Model for production
  - **Backend :** Express (Node.js FrameWork)
  - **Frontend :**
    - React
    - D3.js (for Data Visualizations)
  - **Testing :**
    - Frontend : Jest and React Testing Library
    - Backend : Supertest

Users profile

Admins :

- **Users :** Members of the Anusandhan Team.
- **Mode of Use :** They will have full access to each and every project and be able to update the progress, add new projects, update documents and visualizations, update

relevant data of the project.

- **Technical Familiarity :** High, familiar with project management tools and data handling for visualizations and reports.

Field Worker :

- **Users :** Temporary users assigned by Admins to input field data.
- **Mode of Use :** They will have access to add the data from the field of the existing projects. The access will be temporary and can be removed by Admins. Admin will provide the credentials.
- **Technical Familiarity :** Low to Moderate, familiar with data of the project assigned and limited experience with the UI and the software.

Clients:

- **Users :** Companies funding the projects.
- **Mode of Use :** They can monitor project progress, view reports, upload relevant documents, and provide feedback(comments) on specific tasks.
- **Technical Familiarity :** Moderate, familiar with their project and have basic knowledge in understanding the reports and visualizations. basic understanding of the sofware

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Role Based Authentication	The users will be authenticated based on their roles.They will be given different views,controls according to their role.	R1
2.	Admin Management	Admins shall be able to add or remove users,assign them roles and set user permissions.	R1
3.	Project Management	Admins shall be able to create or delete projects. Admins shall be able to update the project status and can give access for field workers to edit some specific information.	R1
4.	Field Worker Access	Field workers only get temporary access to update project progress(like upload images,mark tasks according to progress etc.,)	R1
5.	Feedback or Comments	Clients shall be able to leave comments on the project page which will be visible for admins and field workers(if given permission).	R1
6.	Client Project View	Clients shall be able to see all the information about their projects like progress,status,budget spent,milestones,impact etc.,	R1
7.	Document Upload	The clients shall be able to upload any documents if necessary directly to the project page.	R2
8.	Customisable Visualisations	Admins shall be able to create and customise visualisations to display project data in client-friendly way.	R1
9.	Downloadable Reports	The clients shall be able to download necessary reports in suitable formats like .csv,.pdf etc.,	R2
10.	Multi-Project Overview	Clients with multiple projects can see a consolidated overview of all projects including status,progress and deadlines.	R1
11.	Role-Based Dashboard views	Each user role shall see a customised dashboard with relevant data and functionalities	R1
12.	Success Story Upload	Admins shall be able to upload success stories related to projects (if any).	R2

Use case diagram:

Use case description

Note:

In alternative flow , the flows are represented as < flow_name (step_number) > here, step_number is the step in the main flow where the alternate flow diverges from.

Use Case Number:	UC-01
Use Case Name:	Role-Based Authentication
Overview:	The purpose of this use case is to authenticate users based on their roles and provide them with appropriate controls.
Actors:	1. Admins 2. Field Workers 3. Clients
Pre condition:	1. The user must have a valid account registered in the system. 2. The user must be assigned a role(Admin,Field Worker,Client)
Flow:	Main (success) Flow: 1. The user will navigate to the login page. 2. The user will enter their credentials(username,password). 3. The system validates credentials. 4. The system identifies the user's role. 5. The system redirects users to respective views based on the role a. **Admin:** Full access to user management,project management and visualisations b. **Client:** Access to project progress,status and feedback features. c. **Field Worker :** Temporary access to update the project progress 6. The user successfully logs in and access their dashboard.
	Alternate Flows: **Invalid Credentials (3) :** 1. At step 3, if the system detects invalid credentials, it displays an error message:"Invalid credentials." 2. The user is redirected back to the login page. **Post Condition :** The user stays in the login page. **Unassigned Role (4) :** 1. At step 4, if the system detects that the user has no assigned role,it displays error message : "Your not assigned any role.Please contact the Admin " 2. The user is redirected to the login page. **Post Condition :** The user remains on the login page.
Post Condition:	1. The user is successfully authenticated and redirected to their respective view. 2. The user will now have access to features and controls based on their role.

Use Case Number:	UC-02
Use Case Name:	Admin Management
Overview:	The purpose of this use case is to allow Admins to manage users by adding, removing,assigning roles and setting permissions for users
Actors:	Admin
Pre condition:	1. The admin must be logged into the system 2. The system must be operational and connected to the database..
Flow:	Main (success) Flow: 1. The admin navigates to the “User Management” section of the dashboard. 2. The admin selects the option to add,remove or edit a user **Add User:** 1. The user enters the user details(username, password,role). 2. The system validates the details and creates a new user account. 3. The system sends a notification to the user with login credentials. **Remove User:** 1. The admin selects the user to be deleted from the current existing list of users. 2. The admins confirm the deletion. 3. The system removes the user and related information from the database. **Edit User :** 1. The admin selects the user whose details are to be edited from the current list of existing users, 2. The admin updates the necessary fields of the user , generally the user's role or permissions. 3. The system saves the changes and updates the database with relevant information. 3. The system updates the database and displays a success message.
	Alternate Flows: **Invalid User Details (2a) :** 1. At step 2(Add user), if the admin enters invalid details, the system displays an error message : “Invalid user details.Please check and try again”. 2. The admin will be prompted to re-enter the details. **Post Condition :** The user is not added , and the system remains on the “Add User” Page. **Unauthorized Action (2b) :** 1. If the admin tries to add,remove or edit a user with higher permissions(Admins), the system displays an error message : “You don’t have permission to perform this action”. 2. The system redirects admin to the user management page. **Post Condition :** No changes are made to the user account. **Network/Database Error (3) :** 1. If the system encounters a network or database error during any of the operations, it displays an error message:”An error occurred.Try again

	later." **Post Condition :** The system remains in its previous state, and no changes are made.
Post Condition:	1. The user database will be updated with the changes made by the admin. 2. The admin receives a confirmation message indicating success of operation.

Use Case Number:	UC-03
Use Case Name:	Project Management
Overview:	This use case allows admins to create,delete and update projects.Admins can also assign field workers to specific projects and grant them access to edit project-related information(eg., progress updates,task completion).
Actors:	1. Admin 2. Field Workers
Pre condition:	1. The admin must be logged in to the system. 2. The system must be operational and connected to the database. 3. The field worker must have access to the specific project. 4. The field worker must be logged in to the system.
Flow:	Main (success) Flow: 1. The admin navigates to the "Project Management" section from the dashboard. 2. The system displays a list of all existing projects with their details 3. The admin selects an action: **Create Project:** 1. The user enters the project details (name,description,budget,milestones etc.,) 2. The system validates and creates the project. **Delete Project:** 1. The admin selects the project to be deleted from the list of existing projects 2. The admin confirms the deletion. 3. The system removes the project and related information from the database. **Update Project Status:** 1. The admin selects the project to be updated from the list of existing projects 2. The admin enters the necessary fields of the project , generally the progress,milestones and status. 3. The system updates the project and related information into the database. **Assign Field Worker:** 1. The admin selects the project to which the field worker is to be assigned from the list of existing projects. 2. The admin then allocates a time period for which the field worker should have access to the selected project. 3. The system updates the project with the permissions into the

	database. 4. The system updates the database and displays a success message.
	Alternate Flows: **Invalid Project Details (3a) :** 1. If the admin enters invalid details (e.g., missing fields etc.,), the system displays an error message : “Invalid project details.Try again later”. 2. The admin will be prompted to re-enter the details. **Post Conditions :** The project will not be created and the system remains on the “Add Project” page. **Field Worker Unavailable (3b) :** 1. If there are no field workers in the system, then the system displays an error message : “No field workers are present in the system. Please add field workers first.” **Post Condition :** The project will not have any field workers assigned.The system remains on the same page where field workers are to be added. **Network/Database Error (4) :** 1. If the system encounters a network or database error during any of the operations, it displays an error message:”An error occurred.Try again later.” **Post Condition :** The system remains in its previous state, and no changes are made.
Post Condition:	1. The project list is updated with the changes made by the admin. 2. The admin receives a confirmation message indicating the success of the operation 3. If a field worker is assigned, they receive a notification about their new project areas.

Use Case Number:	UC–04
Use Case Name:	Field Worker Access
Overview:	This use case allows field workers to temporarily update project progress, including uploading images, marking tasks, and providing field observations.
Actors:	1. Admin 2. Field Workers
Pre condition:	1. The admin must be logged in to the system. 2. Admin should give credentials to the field worker 3. Admin should assign the field worker to a project 4. field worker must be logged into the system
Flow:	Main (success) Flow: 1. The field worker logs into the system.

	2. The system verifies the credentials and role. 3. The field worker navigates to the assigned project. 4. The system displays the project details with editable fields (if granted permission). 5. The field worker updates project progress (e.g., adding status updates, uploading images). 6. The system validates and saves the data to the database. 7. The field worker receives a success message.
	Alternate Flows: **Invalid Credentials (2) :** The system rejects invalid login credentials and redirects the user to the login page. **Unauthorized Access (4) :** If the field worker tries to edit an unassigned project, the system denies access with an error message. **Network/Database Error (7) :** If an error occurs during data submission, the system notifies the field worker and prompts a retry.
Post Condition:	1. The project progress is updated. 2. Admins receive a notification about the update.

Use Case Number:	UC-05
Use Case Name:	Feedback or Comments
Overview:	This use case allows clients to leave comments and feedback on specific projects, which will be visible to admins.
Actors:	1. Clients 2. Admins
Pre condition:	1. The client must be logged into the system. 2. The project must exist. 3. The system must be operational.
Flow:	Main (success) Flow: 1. The client navigates to their project page. 2. The system displays the project details and comment section at each task. 3. The client writes a comment and submits it. 4. The system validates and stores the comment in the database. 5. Admins receive notifications of the comment. 6. The system displays a success message.
	Alternate Flows: **Invalid Comment (3) :** If the comment is empty, the system prompts the client that the comment was empty. **Network Error (4) :**

	If the system fails to store the comment, it prompts a retry.
Post Condition:	The comment is stored and visible to authorized users and Admins..

Use Case Number:	UC-06
Use Case Name:	Client Project View
Overview:	This use case allows clients to view project details, including progress, milestones, budget usage, and reports.
Actors:	Clients
Pre condition:	1. The client must be logged into the system 2. The project must exist 3. The system must be operational.
Flow:	Main (success) Flow: 1. The client logs in and navigates to the project dashboard. 2. The system displays a list of assigned projects. 3. The client selects a project to view details. 4. The system retrieves and displays project data, including visualizations.
	Alternate Flows: **Network/Database Error (4) :** If project data retrieval fails, the system displays an error message.
Post Condition:	The client views real-time project progress, visualizations and reports.

Use Case Number:	UC-07
Use Case Name:	Document Upload
Overview:	This use case allows clients to upload relevant documents directly to the project page.
Actors:	Clients
Pre condition:	1. The client must be logged in. 2. The project must exist. 3. The system must be operational.
Flow:	Main (success) Flow: 1. The client navigates to the project page. 2. The clients navigates to the document upload section 3. The client selects a file and uploads it.

	4. The system validates the file format and size. 5. The system stores the file and links it to the project. 6. A success message is displayed.
	Alternate Flows: **Invalid File Format/Size (4) :** The system rejects unsupported files and prompts a retry. **Network Error (5) :** If upload fails, the system prompts a retry.
Post Condition:	The document is stored and linked to the project.

Use Case Number:	UC-08
Use Case Name:	Customizable Visualizations
Overview:	This use case allows admins to create and customize visualizations for displaying project data in a client-friendly format.
Actors:	Admins
Pre condition:	1. The admin must be logged in. 2. The system must be operational and connected to the database.
Flow:	Main (success) Flow: 1. Admin navigates to a project page 2. Admin navigates to the visualization dashboard 3. Admin selects an action : **Add Visualization:** 1. Admin Uploads the data file with the correct format. 2. Admin selects a the type of visualization and it's attributes 3. Admin saves the visualization for client viewing. **Delete Visualization** 1. Admin selects an existing visualization and clicks on the delete icon. 2. System prompts success message **Edit Visualization :** 1. Admin selects an existing visualization and clicks on the delete icon. 2. Admin can change the data file, type of visualization and the attributes. 3. Admin saves the visualization for client viewing. 4. System prompts success message
	Alternate Flows: **Invalid Data Selection (3a) :** The system prompts the admin to modify the selection. **Network/Database Error (3b):** If visualization fails, the system prompts the admin to try again.
Post Condition:	The new visualization or updated visualization is saved and available for clients.

Use Case	UC-09

Number:	
Use Case Name:	Downloadable Reports
Overview:	This use case allows clients to download project reports in different formats (e.g., CSV, PDF).
Actors:	Clients
Pre condition:	1. The client must be logged in. 2. The project must exist. 3. The system must be operational.
Flow:	Main (success) Flow: 1. The client navigates to the project page. 2. The client clicks on the download report button. 3. System gives the client to select the format of the report file. 4. The system generates and downloads the report.
	Alternate Flows: **File Generation Error (4) :** The system notifies the client and prompts a retry.
Post Condition:	The report is downloaded successfully.

Use Case Number:	UC-10
Use Case Name:	Multi-Project Overview
Overview:	This use case allows clients with multiple projects to see a consolidated overview of all projects.
Actors:	Clients
Pre condition:	1. The client must be logged in. 2. The client should have at least 1 project. 3. The system must be operational.
Flow:	Main (success) Flow: 1. As the client logged in, the client will land on home page which is also the Multi-Project overview page 2. The system retrieves and displays the summary of all project in form of visualizations , progress bar and reports
	Alternate Flows: **Customisable Time-Line (2) :** The client can customize the timeline in which he wants to see the overall progress and reports in multi-project overview
Post Condition:	The client views all projects in one dashboard in a particular timeline.

Use Case Number:	UC-11

Use Case Name:	Role-Based Dashboard Views
Overview:	This use case ensures that users see different dashboards based on their role.
Actors:	1. Admins 2. Field Workers 3. Clients
Pre condition:	1. The user must be logged in. 2. The system must be operational
Flow:	Main (success) Flow: 1. As the user logs in , The system identifies the user's role 2. The system loads the appropriate dashboard layout. 3. The dashboard appears for **Admins:** 1. The whole list of the clients and on clicking on a client it reveals whole list of projects under the client 2. The system also provides to filter based on some categories **Field Worker:** 1. The list of projects that are assigned by the admins will appear. **Clients:** 1. The multi-page overview will be appeared with all projects under the the client
	Alternate Flows: **Network Error (1) :** If the user login fails then the system prompts to try login again.
Post Condition:	The user will see their respective dashboard.

Use Case Number:	UC-12
Use Case Name:	Success Story Upload
Overview:	This use case allows admins to upload success stories (Blogs) related to completed projects.
Actors:	Admins
Pre condition:	1. The admin must be logged in. 2. The project must exist. 3. The system must be operational.
Flow:	Main (success) Flow: 1. The admins navigates to the project page 2. Admin navigates to the success story section. 3. Admin adds a news success story with some images(if any) and clicks on submit. 4. The system validates & saves the story in databases and links to the project.

	Alternate Flows: **Invalid Content (3) :** If the story is empty and the admin tries to upload it then the system prompts that it was an empty story and will not svae it.
Post Condition:	The success story is published.