Software Requirements Specification (SRS) Document

Asset Management System

Team 3

Members
Apeksha Chandak - 2023101126
Arnav Deshmukh-2023101114
Bhavya Ahuja - 2023111035
Mrudani Pimpalkhare- 2023101133
Nidhish Jain - 2023101071

Brief problem statement

Building an **Asset Management System** in which all the assets owned by an organization can be tracked. The system should be able to manage different programs, projects and locations as specified by the organization. The system should handle changes in asset locations and track employee assignments in assets. Additionally, the system must maintain the status of maintenance, repairs, and costs.

System requirements

1. **IDE:** VS Code, WebStorm, or any text editor with JavaScript support
2. **Node.js Version:** 22..x or later(latest)
3. **Frontend** - React based Framework
4. **Browser:** Any browser (Preferable-Chrome/Firefox (for React development))
5. **Web Server-** To be provided by the client
6. **Authentication** - JSON Web Tokens (JWT)
7. **Styling** - Tailwind CSS
8. **Version Control** - Github
9. **Hosting / Deployment** - To be discussed yet (local MongoDB setup.)
10. **Postman** (for API testing)
11. **Nodemon** (for automatic server restart during development)
12. **MongoDB**

Users profile

- <u>Superusers</u> :Responsible for creating new programs as per the organization's structure. Also
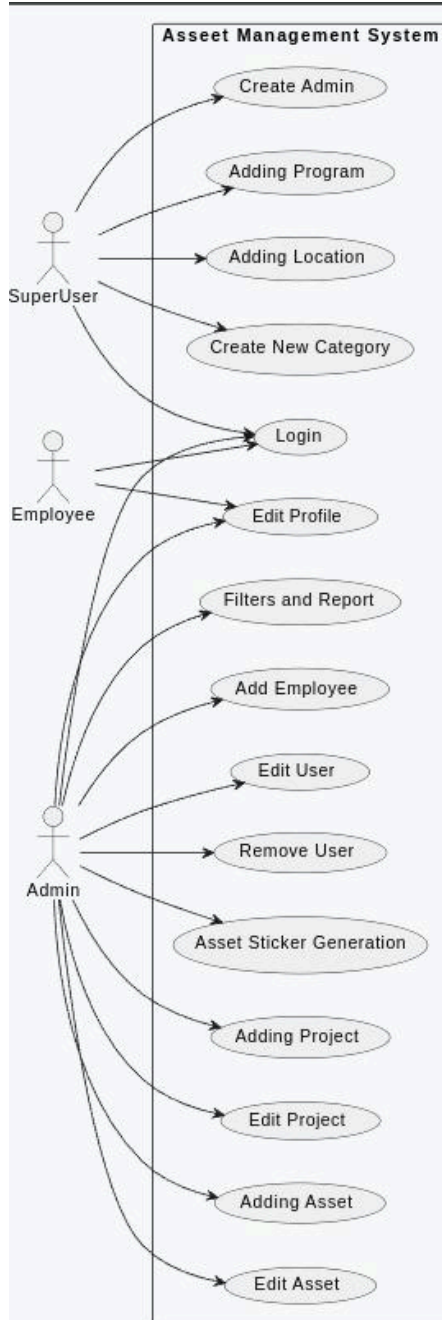
responsible for adding new categories of assets. Responsible for adding new admins.

- <u>Admin-</u> Responsible for creating new employees and adding new assets in the DB of the current categories present. Responsible for creating new projects.Also will generate asset reports by applying different filters.
- <u>Employees-</u> Just have read access. They can see what assets they own and they may not have advanced technical skills hence, the interface for them should be simple.

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Login	User is authenticated via login and can use the system depending on the privileges he/she has	R1
2.	Adding a program	Superuser adds a program to the organization.	R1
3.	Adding a Project	Admin adds a project to the organization with a parent Program	R1
4.	Adding an Asset	Admin adds an asset to the database and configures its properties	R1
5.	Adding a Location	Superuser adds a location for the organization	R1
6.	Creating New Categories	Superuser can add new categories for new assets.	R1
7.	Adding New Employees	Admin can add new Employees to the organization linked to a specific location	R1
8.	Creating New Admins	Superuser can add new Admins linked to a specific location	R1
9.	Editing a Project	Admin can update details about a project like location, program, employees on it, Assets under the project, description of the project, Project Name, Status	R2
10.	Editing an Asset	Admin can update details about an Asset like name, assigned location, assigned project, status, description	R2
11.	Applying filters for report generation	Admin can apply various filters to the different assets present and then can download reports.	R2
12.	Removing users from organisation.	Admin can remove users under them in cases of leaving the organisation.	R2
13.	Asset Sticker Generation	When adding an asset, admin can input an existing sticker ID or the system will auto-generate one in the format: EKL/[Location]/[Category]/[Month]/[Year]/[Random 4 Digits]	R2
14.	Edit User	Admin can change the user details like location (in cases of transfer).	R2
15.	Edit user profile	Users can change their name, email, password, phone number in profile.	R2

Use case diagram

Asseet Management System

- Create Admin
- Adding Program
- Adding Location
- Create New Category
- Login
- Edit Profile
- Filters and Report
- Add Employee
- Edit User
- Remove User
- Asset Sticker Generation
- Adding Project
- Edit Project
- Adding Asset
- Edit Asset

Actors:
- SuperUser
- Employee
- Admin

Use case description

Use Case Number	UC-01
Use Case Name:	Login
Overview:	User is authenticated via login
Actors:	Superusers, Admins, Employees
Pre condition:	The user must be connected to the organization beforehand
Flow:	1. The user accesses the website, via the organization 2. The user tries to login via the authentication system
Alternate Flows:	1. **Wrong Email ID or password** - Invalid Username or Password is displayed. 2. **Removed User Trying to login** - Invalid Username or Password is displayed.
Post Condition:	1. If the user is identified as a **superuser**, then they must be able to access the system with said privileges after successfully logging in. 2. If the user has been added as an **admin** by the **superuser**, the admin will be able to login successfully, and access the system with admin privileges after successfully logging in. 3. If the user is an **employee** added by the **admin**, then, on successful login, the user will be able to access the system with read-only privileges. 4. If neither of 1., 2, 3. are true, depending on the circumstance, the user is either notified about the wrong/unverified credentials

Use Case Number:	UC-02
Use Case Name:	Adding a new program
Overview:	There are multiple programs under which different projects are being executed (eg - CEE,TEOA,etc). Functionality to add new programs has to be provided.
Actors:	Superuser
Pre condition:	The superuser should be authenticated.
Flow:	Main (success) Flow: 1. Superuser login to the Asset Management System. 2. On login superuser will reach the dashboard where Add a new program button would be present.

	3. On Clicking a form would come up where a new program name and a description for it has to be added. 4. System validates the details entered. 5. On validation the new program details are updated on DB. 6. System displays a confirmation message that the details have been added.
	Alternate Flows: 1. **Incomplete details** - If some of the details in form would be missing then the superuser could not submit it and would be able to submit only after completing all the fields. 2. **Duplicate name -** There should not be the same name for two programs. If the same name is added then the user is asked to edit the name.
Post Condition:	The new program is added to the database along with its details and is made available to use by the projects.

Use Case Number	UC-03
Use Case Name:	Adding a project
Overview:	Admin adds a project to the organization
Actors:	Admin
Pre condition:	The user must be a verified system admin
Flow:	1. The admin navigates to the project view, and clicks on the "add a new project" link. 2. The admin adds project details - the name, location, project head, deadline and description. 3. The admin saves and adds the project, and navigates to the view projects page
	Alternate Flows: 1. **Incomplete Details :** If either or any of the project name, head, location are not specified, the admin is prompted to specify them before adding the project, 2. **System error:** If at any point the system fails, the error is logged and displayed, and the admin is prompted to try adding the project again.
Post Condition:	A new project is created in the database and successfully shown on the view-project page.

Post Condition:	5. If the user is identified as a **superuser**, then they must be able to access the system with said privileges after successfully logging in. 6. If the user has been added as an **admin** by the **superuser**, the admin will be able to login successfully, and access the system with admin privileges after successfully logging in. 7. If the user is an **employee** added by the **admin**, then, on successful login, the user will be able to access the system with read-only privileges. 8. If neither of 1., 2, 3. are true, depending on the circumstance, the user is either notified about the wrong/unverified credentials

Use Case Number:	UC-04
Use Case Name:	Adding an asset
Overview:	This use case describes the process of adding a new asset to the asset management system. The system allows users to input relevant asset details and store them in the database for tracking and management.
Actors:	Admin
Pre condition:	The admin must be authenticated and authorized to add new assets.
Flow:	Main (success) Flow: 1. The admin logs into the asset management system. 2. The admin navigates to the "Add Asset" section. 3. The system displays an asset entry form. 4. The admin enters the required asset details 5. The admin submits the form. 6. The system validates the entered details. 7. The system stores the asset details in the database.
	Alternate Flows: AF-01: Missing/invalid data If the user enters incomplete or invalid data, the system prompts the user to correct the errors and prevents submission.
Post Condition:	The asset is successfully added to the system and available for tracking and management.

Use Case Number	UC-05
Use Case Name:	Adding a location
Overview:	Allows to add new location to the organization
Actors:	Superuser
Pre condition:	Superuser is authenticated.

Flow:	Main (success) Flow:
	  1. The superuser navigates to the "Add Location" section in the dashboard.
	  2. The system displays the "Add Location" form.
	  3. The superuser enters the location details (e.g., name, address, contact information).
	  4. The superuser submits the form.
	  5. The system validates the entered details.
	  6. The system saves the new location in the database.
	  7. The system displays a confirmation message indicating that the location has been added successfully.

	Alternate Flows: **1. Invalid Input Data** ● If the superuser enters invalid data (e.g., missing required fields or incorrect format), the system displays an error message and prompts for correction or the entered location already exists. ● **Post Condition:** The location is not added, and the system remains on the "Add Location" form until valid data is provided. ● **Duplicate Location name :** The system displays an error message **2. System Error** ● If there is a database error or connectivity issue, the system displays an error message and logs the issue. ● **Post Condition:** The location is not added, and the system prompts the superuser to try again later
Post Condition:	The new location is successfully saved in the database and is available for use within the organization..

Use Case Number	UC-06
Use Case Name:	Creating new categories
Overview:	To can add new categories for new assets.
Actors:	Superusers
Pre condition:	1. The user must be connected to the organization beforehand 2. The Superuser must be authenticated
Flow:	1. Superuser navigates to add a new category option in the dashboard. 2. The system displays the Add New Category form 3. The superuser enters details like(category name ,estimated life,depreciation rate,etc) 4. The superuser submits the form 5. The system validates the entered details. 6. The system creates and saves the new category for the assets in the system.
	Alternate Flows **1. Invalid Input Data** ● If the superuser enters invalid data (e.g., missing required fields or incorrect format(like negative estimated years)), the system displays an error message and prompts for correction or the entered category name already exists. ● **Post Condition:** The category is not added, and the system remains on the "Add New Category " form until valid data is provided **2. System Error** ● If there is a database error or connectivity issue, the system displays an error message and logs the issue. ● **Post Condition:** The category is not added, and the system prompts the superuser to try again later
Post Condition:	The new category is successfully saved in the database and is available for use within the organization..

Use Case Number	UC-07
Use Case Name:	Adding new employees
Overview:	To add new Employees to the organization linked to a specific location

Actors:	Admin
Pre condition:	1. The user must be connected to the organization beforehand 2. The Admin must be authenticated
Flow:	1. Admin navigates to add a new employee option in the dashboard. 2. The system displays the Add New Employee form 3. The admin enters details like(name,email, id ,etc) 4. The admin submits the form 5. The system validates the entered details. 6. The system creates and saves the new employee .
	Alternate Flows **1. Invalid Input Data** ● If the admin enters invalid data (e.g., missing required fields or incorrect format) or the entered employee already exists. ● the system displays an error message and prompts for correction ● **Post Condition:** The category is not added, and the system remains on the "Add New Employee " form until valid data is provided **2. System Error** ● If there is a database error or connectivity issue, the system displays an error message and logs the issue. ● **Post Condition:** The Employee is not added, and the system prompts the admin to try again later
Post Condition:	The new employee is successfully saved in the database and is registered with the organisation

Use Case Number:	UC-08
Use Case Name:	Creating new admins
Overview:	Organisation might need to add new admins as when new locations are made or when more admins are needed to ensure smooth operations.
Actors:	Superuser
Pre condition:	The superuser should be authenticated.

Use Case Number:	UC-09
Use Case Name:	Editing a Project
Overview:	Project name, description, location, or other details are subject to change over time, hence, an editing project option is needed.
Actors:	Admin
Pre condition:	1. Admin must be authenticated. 2. Project must exist in the organization
Flow:	Main (success) Flow: 1. Admin logs into the AMS. 2. Admin navigates to the "Projects" Section. 3. Admin selects the project to edit. 4. System displays the project details in an editable form. 5. Admin edits the Project details 6. Admin submits the updated project details 7. System validates the updated details 8. System updates the project details in the database 9. System displays the confirmation details

	Alternate Flows: **1. Incomplete details** - If some of the details in form would be missing then the Admin could not submit it and would be able to submit only after completing all the fields. **2. Duplicate Project Name -** There should not be the same details in project name to be added **3. Admin Cancels Editing** - Admin clicks the cancel button and the system will discard any changes made.
Post Condition:	The new project details are updated to the database and added to the system according to the admin's edits.

Use Case Number:	UC-10
Use Case Name:	Editing an Asset
Overview:	Asset name, description, location, or other details are subject to change over time, hence, an editing asset option is needed.
Actors:	Admin
Pre condition:	1. Admin must be authenticated. 2. Asset must exist in the organization

Flow:	Main (success) Flow:
	  1. Admin logs into the AMS.
	  2. Admin navigates to the "Assets" Section.
	  3. Admin selects the asset to edit.
	  4. System displays the asset details in an editable form.
	  5. Admin edits the asset details
	  6. Admin submits the updated asset details
	  7. System validates the updated details
	  8. System updates the asset details in the database
	  9. System displays the confirmation details

	Alternate Flows:
	  1. **Incomplete details** - If some of the details in form would be missing then the Admin could not submit it and would be able to submit only after completing all the fields.
	  2. **Admin Cancels Editing** - Admin clicks the cancel button and the system will discard any changes made.

Post Condition:	The new asset details are updated to the database and added to the system according to the admin's edits.

Use Case Number:	UC-11
Use Case Name:	Applying filters for asset reports and then downloading it.
Overview:	We can apply various filters like in stock, issued, not serviceable(damaged), repairing.
Actors:	Admin
Pre condition:	1. Admin must be authenticated.
Flow:	Main (success) Flow: 1. Admin logs into the AMS. 2. Admin navigates to the "Reports" Section. 3. Admin selects the category(ies), status, assignment status ,location(s). 4. If some field is missing then we can do a select all query and then show details for the things given. 5. Then the admin will select the filters like in stock, issued , not disposed,etc. 6. After selecting a submit button would be clicked and the filters and assets would be filtered out and displayed on the screen. 7. Based on the report on screen the admin can do changes in the search query by changing the different details filled in order to get the desired result. 8. If the search is complete then it can be downloaded in excel format. 9. The excel should include the name, the assigned employee(if any),the depreciation rate, current value and other basic details.
	Alternate Flows: 1. **No assets as per the combination given -** If assets are not present then nothing would be displayed.
Post Condition:	After clicking the submit button database is searched to get the correct assets and then reports are generated.

Use Case Number:	UC-12
Use Case Name:	Removing users from organisation
Overview:	In case of a user leaving the organisation, the admin can remove the user if all the assets assigned to the user are returned and no dues are submitted.
Actors:	Admin
Pre condition:	1. Admin must be authenticated. 2. The user should exist in the organisation and should be under the admins administration.
Flow:	Main (success) Flow: 1. Admin logs into the AMS. 2. Admin navigates to the "Users" Sections. 3. Admin search the user. 4. Admin click on the profile of the user and click on remove user option. 5. User is shown a pop up which shows all the assets assigned to him, and a message that all these assets will be unassigned and he will removed to all his projects. 6. Admin is asked for confirmation of removal. (regardless of whether he had assets or was a part of projects) 7. After confirming, user will be removed from the organization and turned inactive.
Post Condition:	After clicking the submit button, the database is updated and the user's active field is made inactive. We cannot remove the user as their usage of assets etc might need to be tracked in future even after leaving the organisation.

Use Case Number:	UC-13
Use Case Name:	Asset Sticker Generation
Overview:	When adding an asset, each asset needs a unique sticker ID. This can be manually entered if an existing sticker is available, or automatically generated by the system following a specific format.
Actors:	Admin
Pre condition:	- Admin must be authenticated - The admin must be in the process of adding a new asset (UC-04)

Flow:	Main (success) Flow:
	1. During asset addition process, the system displays a field for "Sticker ID"
	2. The admin can either:
	    a. Enter an existing sticker ID manually, or
	    b. Leave the field empty for auto-generation
	3. If left empty, upon form submission, the system generates a sticker ID in the format: EKL/[Location]/[Category]/[Month]/[Year]/[Random 4 Digits]
	4. The sticker ID is associated with the asset in the database

	Alternate flows - None
Post Condition:	The asset is successfully associated with a sticker ID that is stored in the database and can be used for identification and tracking.

Use Case Number:	UC-14
Use Case Name:	Edit User
Overview:	In cases of transfer admin need to handle the cases of transfer by changing the location of the user.
Actors:	Admin
Pre condition:	- Admin must be authenticated
Flow:	Main(success) flow - 1. Admin navigate to the view users page. 2. Admin searches the name or the email id of the user whom they want to edit. 3. Admin can view the details of the user. 4. Admin is given all the locations in the organisation so that they can select the location at which the user has to be transferred. 5. In case of no assets the users location is updated. 6. Click on update user button so that the details get updated.
Alternate flow:	Assets are present with the user. 1. In case of changing the location, admin is asked which assets to be kept with the user and which should be unassigned and kept at the location. The user has to tick those assets which will go along with the user being transferred.
Post Condition:	The user database is updated with the new location. The assets status and locations are also updated accordingly.

Use Case Number:	UC-15
Use Case Name:	Edit profile.
Overview:	This use case allows users (Superusers, Admins, and Users) to update their personal information in their profile section, including name, email, password, and phone number
Actors:	Admin, Superuser, Users.
Pre condition:	- The user must be authenticated and logged into the system - The user must have an existing profile in the system
Flow:	Main (success) Flow: 1. User logs into the Asset Management System 2. User navigates to the "View Profile" section 3. System displays the user's current profile information in an editable form 4. User modifies desired fields (name, email, password, phone number) 5. User submits the updated information 6. System validates the entered details 7. System updates the user's profile information in the database 8. System displays a confirmation message indicating successful update
	Alternate Flows: 1. **Invalid Input Data** If the user enters invalid data, the system displays an error message and prompts for correction **Post Condition**: The profile is not updated, and the system remains on the edit profile form until valid data is provided 2. **Duplicate Email** If the user attempts to change their email to one that already exists in the system, an error message is displayed **Post Condition**: The profile update is rejected, and the user must provide a unique email 3. **Password Confirmation Mismatch** If the user is changing their password and the confirmation doesn't match, an error message is displayed **Post Condition**: The password is not updated, and the user must correctly confirm the new password
Post Condition:	The user's profile information is successfully updated in the database and reflected in the system.