

PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004
DEPARTMENT OF COMPUTER APPLICATIONS
Academic Year 2024-2025 MCA G1 & G2 Sem 3
23MX37 Mini project
Software Requirement Specification (SRS) Document

1) Title of the Mini project

Asset management and tracking system

2) Team Detail

Batch No.	Roll No.	Name of the student	Name of the Faculty Guide
10	23MX322	Rahul G	Dr Subathra M

3) Project Goal

The goal of the project is to design and develop an asset management and tracking system for the organization

4) Project Abstract

The “Asset management and tracking system” project aims to create a robust and simple online platform to keep track of existing assets as well as add new assets. Built on a technology stack comprising of HTML5, CSS3, JavaScript and Bootstrap for responsive front-end design, Django for server-side functionality and PostgreSQL for efficient data storage, the portal provides a versatile environment to easily manage and track assets.

Users register using their official credentials, gaining access to role-specific functionalities tailored for administrators and other users like students, teaching staff, etc. The portal allows all users to view the assets available and their specifications including location, status and many more. These allow the administrators to easily view and identify the assets instead of looking them up manually.

Added features such as filtering of assets based on their characteristics, displaying a log of the upgradation of each asset along with its date and the upgrades performed are also included.

The portal also integrates the ability to download a list of the assets in excel or pdf format for official purposes in and outside the department.

By promoting maintenance and lookup of available resources and assets, the asset management and tracking system aims to digitize the traditional way of managing assets. It serves as a centralized hub for all assets.

5) Existing System/ Literature Review

The existing systems for asset management include:

- **Asset panda (<https://www.assetpanda.com/>)**

Asset Panda is an IT management software that places asset lifecycle management as a top priority. It is designed to empower organizations to efficiently track, organize, and optimize their assets across the entire organization, ensuring that each asset is utilized to its fullest potential.

- **ManageEngine (<https://www.manageengine.com/>)**

ManageEngine AssetExplorer offers comprehensive IT Asset Management (ITAM) solutions, guiding businesses through the entire asset lifecycle, from planning to disposal. With a user-friendly interface, AssetExplorer offers diverse methods for discovering all organization assets, simplifying the process.

- **SolarWinds (<https://www.solarwinds.com/>)**

SolarWinds Service Desk is an all-encompassing web-based platform designed to maintain a consistent record of the organization's IT assets, ensuring efficient tracking, management, and control.

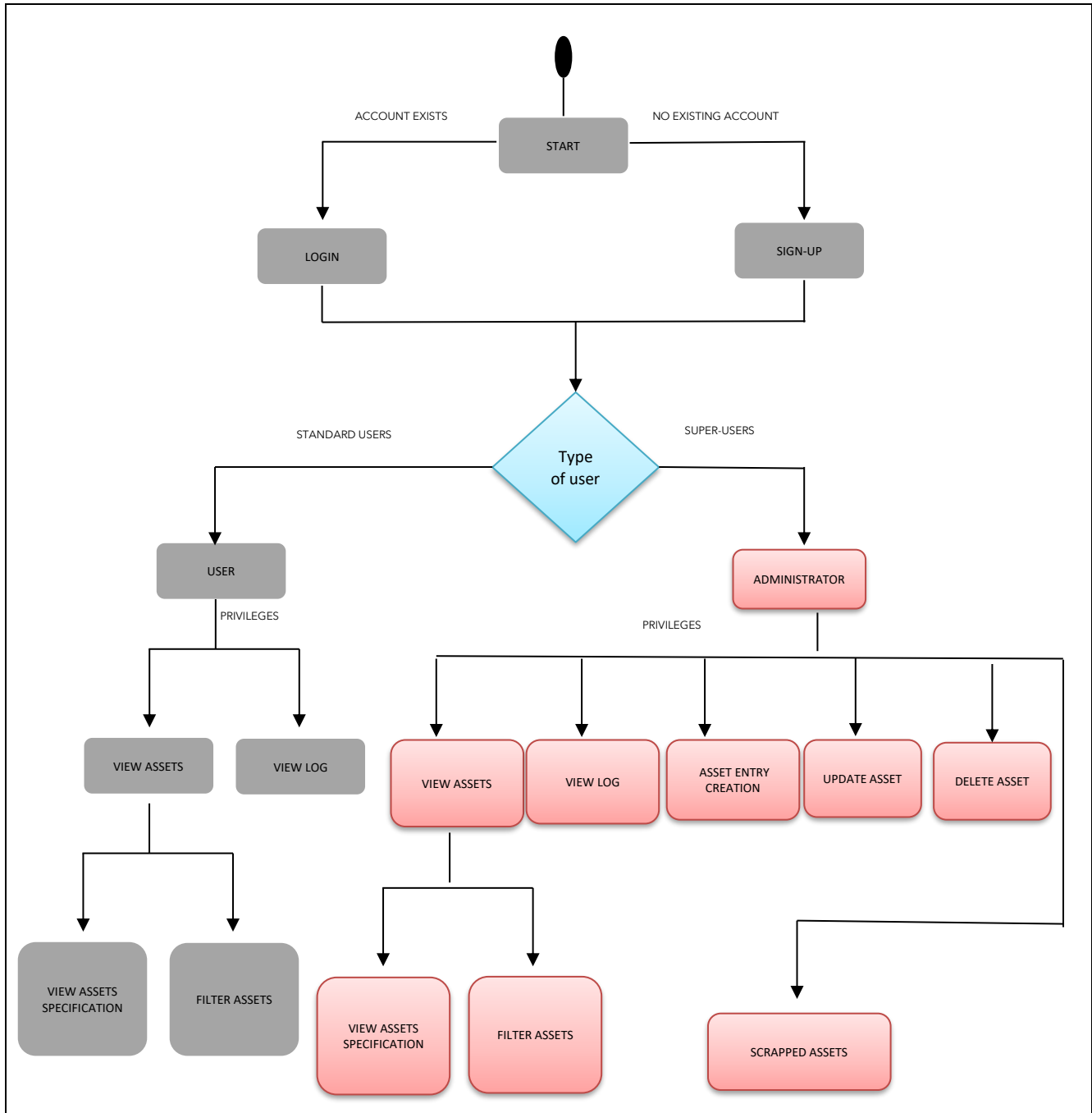
- **Syam Software (<https://syamsoftware.in/simplified-intelligent-asset-management/>)**

Syam Software is an intelligent IT asset management software that helps the organization manage and track their assets. It provides a real-time report and updates about any changes in the hardware and software of the client.

- **Asset Sonar (<https://ezo.io/assetsonar/>)**

Asset Sonar is an advanced cloud-based solution for IT hardware and software management, designed to streamline asset maintenance and asset lifecycle management across diverse locations.

6) Project Flow Analysis



7) Outcome of the Proposed Project

The outcome of the asset management project will be a web application that enables efficient management and viewing of various assets for each of the organizations divisions

Key benefits of the project include:

- User Authentication: Secure sign-up and login processes for users and administrators.

- **Role-Based Access:** Differentiation between administrator and user functionalities, enhancing security and usability.
- **Asset Categorization:** Organized display of assets by categories such as computer labs, classrooms, seminar hall etc.
- **Detailed Asset Specifications:** Users can view detailed information about each asset, facilitating better resource management.
- **Scalability:** The system is designed for future expansion with additional features or assets as needed.

8) Modules of the Proposed Project- Functional Requirement

The modules identified in the Project are:

- User authentication module
- Dashboard module
- Upgradation log module
- Input form / update form module
- Asset Filtering module

9) Non- functional requirements of the Project

The non-functional requirements identified for this project are:

- **Performance:** The system should provide fast response times and handle multiple concurrent users efficiently.
- **Security:** User data must be protected with authentication.
- **Usability:** The interface should be intuitive and require minimal training for end-users
- **Scalability:** The system must support horizontal scaling to accommodate increased load and asset growth.

- Availability: Ensure 99.9% uptime, accessible 24/7 with minimal scheduled downtime.

10) Project Deliverables

A Minimum Viable Product (MVP) for an asset management system should cover the basic needs without overcomplicating things. It should let users add new assets with details like type, serial number, purchase date, and value. Users should also be able to update the information if something changes, delete assets when they're no longer needed, and view all the details easily. An important feature is keeping a log of all updates so there's a record of any changes made. This will ensure the system is functional and practical, meeting the essential needs of managing assets.

11) Technology to be used

The technologies/Frameworks used for mini-project are:

- Frontend: HTML5, CSS3, JavaScript, Bootstrap

HTML5, CSS3 and JavaScript form the basis of web development and are the building blocks of any web application. The use of Bootstrap, a CSS framework helps developers to easily design and develop websites.

Bootstrap offers a wide range of pre-built components, such as navigation bars, forms, buttons, and more, saving you time and effort in coding from scratch. It provides a responsive grid system that automatically adjusts the responsive layout based on the screen size.

JavaScript helps in creating interactive websites that enrich the user experience and make it more immersive.

- Backend: Django

Django is a python-based framework which is mainly designed for speedy development of apps. Django Architecture follows the MVT structure. In MVT, M stands for Model, V stands for views, and T stands for Templates.

Model is the structure of storing the data in the database, the view is a python function used to handle the web request, and the template contains static content like HTML, CSS, and JavaScript.

The Object-Relational Mapper (ORM) is one of Django's most powerful features, allowing you to interact with your database in the same way that you would with SQL.

- Database: PostgreSQL

Postgres is categorized as an ORDBMS (Object-Relational Database Management System) due to its ability to handle complex data types and store and retrieve objects.

ORDBMSs are better suited to handling complex data structures and have more advanced query capabilities. NoSQL databases often prioritize scalability, availability, and high-speed data access over data consistency, whereas ORDBMSs are designed to prioritize data consistency while still providing some flexibility in data modelling.

The Postgres support for JSON and other semi-structured data types make it a viable option for applications that require some NoSQL capabilities. Postgres also has strong support for indexing, allowing for efficient retrieval of data even when dealing with large and complex data structures.

12) Timeline of Activities planned/completed

SPRINT NO.	ACTIVITY	PERIOD
1	Intimation of Project details, Project title selection and submission of project abstract	28/06/2024 (Friday) - 10/07/2024 (Wednesday)
2	Submission of SRS document	11/07/2024 (Thursday) - 16/07/2024 (Tuesday)
3	Design and coding of user authentication and dashboard module	17/07/2024 (Wednesday) - 25/07/2024 (Thursday)
4	Implementation of input/update form and update log modules	29/07/2024 (Monday) – 12/08/2024 (Monday)
5	Design and implementation of Asset filtering and pagination module	13/08/2024 (Tuesday)- 09/09/2024 (Monday)
6	Testing	14/09/2024 (Saturday)- 26/09/2024 (Thursday)
7	Deployment	27/09/2024 (Friday)- 29/09/2024 (Sunday)
8	Project Report submission	30/09/2024 (Monday)- 01/10/2024 (Tuesday)

RAHUL G (23MX322)

Name & Signature of the student

Signature of the Guide