**Software Requirements Specification for Asset Management System**

**1. Introduction**

**1.1 Purpose**

The Asset Management System (AMS) is designed to help organizations track, manage, and maintain assets such as laptops, printers, scanners, etc. It enables asset tagging using QR codes, maintenance scheduling, and issue reporting via a web and mobile application.

**1.2 Scope**

* **Asset Tracking**: Add, update, and track assets in real-time.
* **QR Code Integration**: Generate & scan QR codes for asset identification.
* **Maintenance Management**: Schedule, monitor, and track asset maintenance.
* **Reporting System**: Employees can report faulty assets via a mobile app.
* **Role-based Access**: Admin and Employee have different permissions.

**1.3 References**

* React.js documentation: <https://reactjs.org/docs/getting-started.html>
* Node.js documentation: <https://nodejs.org/en/docs/>
* AWS documentation: <https://docs.aws.amazon.com/>

**2. Overall Description**

**2.1 Product Perspective**

The AMS is an independent system designed to integrate with existing enterprise systems via APIs for seamless data exchange.

**2.2 User Classes and Characteristics**

* **Admin**: Manages all assets, users, and system settings.
* **Employee**: Views assets assigned to them, reports issues.

**2.3 Operating Environment**

* **Web Application**: Compatible with all web browsers.
* **Mobile Application**: Available on iOS and Android platforms.

**3. Functional Requirements**

**3.1 Asset Management**

* Admin can add, update, or delete assets.
* Users can view asset details and history.

**3.2 QR Code Integration**

* System generates a unique QR code for each asset.
* Mobile app can scan QR codes to retrieve asset info.

**3.3 Maintenance Scheduling**

* IT Manager can schedule regular maintenance.
* System sends automated alerts for upcoming maintenance tasks.

**3.4 Reporting System**

* Employees can report asset issues via mobile app.
* Admin reviews and assigns reported issues.

**3.5 Request for extra equipment**

* Employee can request for any extra requirements for his asset.
* Admin looks after the request and provides with the necessary requirements.

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* System should load pages within 2 seconds.
* QR code scanning should process within 1 second.

**4.2 Security Requirements**

* Data encryption for sensitive information.
* Role-based access control to restrict unauthorized access.

**4.3 Usability Requirements**

* User-friendly interface with intuitive navigation.
* Comprehensive user guide and FAQs available.

**5. Supporting Information**

**5.1 Glossary**

* **Asset**: Any item owned or managed by the organization.
* **QR Code**: A machine-readable code used for asset tagging.

**5.2 Technologies Used**

* **Frontend**: React.js (Web), React Native (Mobile)
* **Backend**: Node.js with Express
* **Database**: MongoDB
* **QR Code**: JavaScript or Python (qrcode library)
* **Hosting**: AWS / Firebase

This document serves as a comprehensive guide for stakeholders to understand the system’s functionalities, design, and requirements

***Flow Chart for asset management system***

