Requirements

1. User Management

The system must allow different types of users with varying access levels and permissions.

Requirements:

- **Account Creation and Authentication**:
    - Users must be able to register, log in, and authenticate using secure credentials.
    - Support for role-based access.
    - User authentication should be handled using AWS services such as **AWS Cognito**.
- **Role-based Permissions**:
    - **Admin**: Full write and read access to all assets, users, manage assets, assign assets to users, generate reports and system settings.
    - **Basic User**: Read access to all the assets, the users can view and manage assets assigned to them, request new assets.

2. Asset Management

A core functionality where assets are tracked, monitored, and managed efficiently.

Requirements:

- **Asset Creation and Assignment**:
    - Admins should be able to create, update, rent and delete assets.
    - Assets must be assignable to users and locations.
    - All users must be able to search/scan assets using unique barcode for all the assets
    - Users should see a list of assets assigned to them, including their status (In Use, Available, Rented, etc.).
- **Asset Details**:
    - Each asset must have detailed metadata, such as ID, name, type, serial number, model, location, status, and warranty terms.
    - Assets should have associated documents and images for easier identification.
- **Bulk Asset Import**:
    - The system should support importing assets via multi scan and csv import for large-scale asset onboarding.

3. QR Code and Barcode Functionality

Simplifying asset tracking through scanning.

Requirements:

- **QR Code Generation**:
    - The system will allow admins to generate QR codes for assets.
    - QR codes should be Generated in a batch of 50 or unless specified by the admin ith max value of 100 QR code generation at once
    - Each asset will have a unique QR code for easy identification and tracking.
    - QR codes should be automatically assigned to each asset during asset creation.
- **QR Code Scanning**:
    - Users should be able to scan QR codes to quickly retrieve asset details using a mobile device or scanner.
    - Users can scan assets to view details and perform operations like check-in/check-out or copy asset information into a new asset.
    - Multi-scan functionality should allow batch processing of multiple assets simultaneously, such as bulk check-ins/outs.

4. Inventory Management:

Managing the asset types and inventory effectively.

Requirements:

- **Asset Categorization:**
    - Inventory of assets categorized by type and status (Available, In Use, Rented, etc.).
    - Display inventory levels for each asset type (e.g., Available, In Use, Rented).
- **Asset Status**:
    - Assets must have various statuses such as Available, In Use, Rented, Under Maintenance, and Retired.
    - Allow all users to update the status of each asset.
- **Check-in/Check-out Process**:
    - Assets must be able to be checked in and out by users and assigned accordingly via Transfer, scan or using shortcuts.
    - Ensure tracking of check-in/out history, including user, location, and timestamps.
- **Search and Filters:**
    - Ability to filter/sort assets by type, status and other fields(TBD).
    - Keyword search to quickly find assets.

- **Asset Lifecycle**:
    - Track the entire lifecycle of an asset, including purchase, usage, maintenance, and eventual disposal or retirement.
    - The system should allow all users to create tasks for each asset and assign those tasks to themselves or other users

5. Reporting and Analytics

Provide insights into asset utilization and inventory.

Requirements:

- **Customizable Reports**:
    - Admins should be able to generate reports based on asset usage, status, location, and other attributes.
    - Provide exportable reports in CSV or PDF format.

7. Location Tracking and Mapping

Track and manage assets across multiple locations.

Requirements:

- **Asset Location**:
    - Assets must be assignable to specific locations, such as offices, warehouses, or user homes.
    - The system should show where each asset is located and allow easy reassignment.
- **Map Integration**:
    - Display asset locations on a map for easy identification and tracking.
    - Support viewing asset clusters and movements between locations.

8. Notifications and Alerts

Automatic notifications and alerts for asset status and lifecycle events.

Requirements:

- **Alerts for Maintenance and Warranty**:
    - Notify users when an asset is due for maintenance or if the warranty is nearing expiration.
    - Alerts for overdue check-ins/outs or when an asset requires attention (e.g., maintenance or upgrade).
- **Real-time Notifications**:
    - Support for real-time notifications when assets are checked out or status changes.

9. Security and Compliance

Ensure the security of sensitive data and comply with regulations.

Requirements:

- **Data Security**:
    - Implement security protocols such as SSL encryption for data transmission.
    - Use **AWS IAM** for secure role-based access control.
- **Audit Logs**:
    - Maintain audit logs for all user actions, including asset creation, updates, and deletions.
    - Ensure compliance with company policies and relevant regulations for asset management.

10. Integration with AWS Services

Utilizing AWS services for cloud-based asset management and scalability.

Requirements:

- **AWS Lambda and API Gateway**:
    - Implement serverless functions using **AWS Lambda** for asset-related operations.
    - Use **API Gateway** to handle requests between the mobile app or web interface and the backend services.
- **AWS S3 for Storage**:
    - Use **AWS S3** to store asset-related documents, images, and files.
- **AWS DynamoDB/RDS for Database**:
    - Choose between **DynamoDB** (NoSQL) or **RDS** (Relational Database) for scalable asset storage and management.