项目文档

# Functional Requirement

2. Functional Requirements  
2.1 Asset Registration  
Input: Asset name, category, description, purchase date, cost, location, owner.  
Output: Confirmation of asset registration with unique asset ID.  
Description: Users can register new assets into the system. Only authenticated users can perform this action.  
2.2 Asset Usage Tracking  
Input: Asset ID, user ID, start time, end time.  
Output: Updated usage history record.  
Description: Users can log when they start and end using an asset. This data is used for tracking and reporting.  
2.3 Asset Transfer  
Input: Source user ID, destination user ID, asset ID, reason for transfer.  
Output: Updated ownership and location of the asset.  
Description: Assets can be transferred between users within the same organization. Requires administrative approval.  
2.4 Asset Return  
Input: Asset ID, user ID, condition upon return.  
Output: Updated asset status to "available".  
Description: Users can return assets after use. The system logs the return event and updates the asset’s availability.  
2.5 Report Generation  
Input: Date range, asset type, user ID.  
Output: Exportable reports in PDF/CSV format.  
Description: Administrators can generate reports on asset usage, transfers, and inventory levels.  
2.6 Permission Allocation  
Input: User role, permissions to assign (e.g., edit, delete, approve).  
Output: Updated user permissions.  
Description: System administrators can assign different levels of access to users based on their roles.  
2.7 Approval Workflow  
Input: Request type (e.g., transfer, deletion), requester ID, asset ID.  
Output: Status update (approved/rejected).  
Description: Certain actions require multi-level approvals. The workflow must be configurable and traceable.  
2.8 Data Import and Export  
Input: File (CSV/XML), format options.  
Output: Imported asset records / Exported data file.  
Description: Supports bulk import/export of asset data in standard formats.  
3. External Interfaces  
3.1 Authentication System Integration  
Interface Type: API  
Description: The system integrates with a centralized authentication system (e.g., LDAP, OAuth 2.0, SSO) to verify user identities and manage access control.  
3.2 Email Notification System  
Interface Type: SMTP / REST API  
Description: Sends automated emails for approval requests, asset alerts, and system notifications.  
3.3 Database Interface  
Interface Type: SQL (e.g., PostgreSQL, MySQL)  
Description: Stores all asset data, user information, logs, and audit trails.  
3.4 File Storage Interface  
Interface Type: Local or Cloud (e.g., AWS S3)  
Description: Used for storing uploaded files (e.g., reports, images).

# External Description

3. External Interfaces  
3.1 Authentication System Integration  
Interface Type: API  
Description: The system integrates with a centralized authentication system (e.g., LDAP, OAuth 2.0, SSO) to verify user identities and manage access control.  
3.2 Email Notification System  
Interface Type: SMTP / REST API  
Description: Sends automated emails for approval requests, asset alerts, and system notifications.  
3.3 Database Interface  
Interface Type: SQL (e.g., PostgreSQL, MySQL)  
Description: Stores all asset data, user information, logs, and audit trails.  
3.4 File Storage Interface  
Interface Type: Local or Cloud (e.g., AWS S3)  
Description: Used for storing uploaded files (e.g., reports, images).