Fixed Asset Management System - Business Requirements Document (BRD)

Project Overview

Project Name:

Fixed Asset Management System

Project Description:

A comprehensive system to manage the lifecycle of fixed assets, ensuring proper tracking, maintenance, compliance, and reporting.

Objectives:

- Efficiently manage and track assets

- Ensure regulatory compliance

- Optimize maintenance schedules

- Provide detailed reporting and analytics

Scope:

This document outlines the business requirements for the Fixed Asset Management System, detailing each module and its microservices.

Stakeholders:

- Project Sponsor

- Asset Managers

- Maintenance Teams

- Financial Department

- IT Department

System Requirements/ Modules:

1. Authentication and Authorization Module

Description:

Handles user authentication and authorization to ensure secure access to the system.

Microservices:

- Authentication Service: Manages user login and authentication.

- Features:

- User login/logout

- Password management (reset, change)

- Multi-factor authentication

- Implementation: Use FastAPI with OAuth or JWT for secure authentication.

- Authorization Service: Manages user roles and permissions.

- Features:

- Role-based access control

- Permission assignment

- Role hierarchy management

- Implementation: Use RBAC (Role-Based Access Control) with a database to store roles and permissions.

Requirements:

- Secure user authentication (e.g., using OAuth or JWT).

- Role-based access control.

- User management (add, update, delete users).

2. Asset Management Module

Description:

Manages the lifecycle of assets from registration to disposal.

Microservices:

- Asset Registration Service: Registers new assets.

- Features:

- Asset addition with metadata

- Unique asset identification

- Implementation: Use FastAPI to create RESTful endpoints for asset registration and store data in a relational database.

- Asset Inventory Service: Tracks current inventory.

- Features:

- Inventory listing

- Status update

- Implementation: Develop RESTful APIs to retrieve and update inventory data.

- Asset Tracking Service: Monitors asset locations.

- Features:

- Real-time location tracking

- Historical location data

- Implementation: Use GPS integration or IoT devices for real-time tracking.

- Asset Metadata Service: Manages asset metadata.

- Features:

- Manage additional metadata (e.g., warranty details)

- Implementation: Create APIs to manage and update asset metadata in the database.

- Asset Categorization Service: Categorizes assets.

- Features:

- Categorization based on type, department, etc.

- Implementation: Use a tagging system or predefined categories stored in the database.

Requirements:

- Register new assets with details.

- Track asset status and location.

- Manage asset metadata (warranty, purchase date, etc.).

- Categorize assets for easy retrieval.

3. Maintenance and Lifecycle Module

Description:

Tracks and manages asset maintenance and lifecycle stages.

Microservices:

- Maintenance Service: Logs maintenance activities.

- Features:

- Schedule maintenance

- Maintenance history

- Implementation: Develop a scheduler for maintenance tasks and endpoints for logging activities.

- Depreciation Calculation Service: Computes depreciation.

- Features:

- Calculate using different methods (straight-line, declining balance)

- Depreciation reports

- Implementation: Implement depreciation algorithms and create APIs to retrieve depreciation data.

- Lifecycle Management Service: Manages lifecycle stages.

- Features:

- Track stages (acquisition, utilization, disposal)

- Lifecycle history

- Implementation: Use state machines to manage lifecycle stages and endpoints for lifecycle transitions.

- Repair Service: Logs repair activities.

- Features:

- Schedule repairs

- Repair history

- Implementation: Create endpoints for logging repair activities and tracking repair history.

- Retirement Service: Manages asset disposal.

- Features:

- Asset retirement procedures

- Disposal documentation

- Implementation: Develop APIs for managing asset retirement and updating inventory status.

Requirements:

- Schedule and log maintenance activities.

- Calculate asset depreciation.

- Track lifecycle stages from acquisition to disposal.

- Log repair activities and manage repairs.

- Handle asset retirement/disposal.

4. Procurement Module

Description:

Manages the procurement process for acquiring new assets.

Microservices:

- Purchase Order Service: Manages purchase orders.

- Features:

- Create and track purchase orders

- Approve/reject orders

- Implementation: Create endpoints for creating, updating, and tracking purchase orders.

- Vendor Management Service: Tracks vendor information.

- Features:

- Vendor listing

- Performance metrics

- Implementation: Develop APIs to manage vendor data and integrate performance tracking metrics.

- Procurement Integration Service: Integrates with procurement systems.

- Features:

- Data synchronization

- Automated acquisition

- Implementation: Use APIs to connect with external procurement systems for data synchronization.

Requirements:

- Create and track purchase orders.

- Manage vendor information and performance.

- Integrate with external procurement systems.

5. Location and Space Management Module

Description:

Manages the physical locations and space utilization of assets.

Microservices:

- Location Tracking Service: Tracks asset locations.

- Features:

- Real-time location updates

- Historical location data

- Implementation: Develop a location management system with APIs to update and retrieve location data.

- Space Utilization Service: Optimizes space usage.

- Features:

- Monitor space usage

- Optimize space allocation

- Implementation: Use analytics to monitor space utilization and create endpoints for space management.

Requirements:

- Track asset locations within buildings and rooms.

- Monitor and optimize space usage.

6. Compliance and Risk Management Module

Description:

Ensures assets comply with regulations and manages risks.

Microservices:

- Regulatory Compliance Service: Ensures regulatory compliance.

- Features:

- Compliance checklists

- Compliance status tracking

- Implementation: Develop compliance checklists and create APIs to track compliance status.

- Risk Assessment Service: Conducts risk assessments.

- Features:

- Risk identification

- Risk reporting

- Implementation: Implement risk assessment algorithms and create endpoints for risk reporting.

Requirements:

- Ensure assets comply with relevant regulations.

- Conduct and report risk assessments.

7. User Management Module

Description:

Manages user profiles and access control.

Microservices:

- User Profile Service: Manages user profiles and preferences.

- Features:

- Profile creation/update

- User settings management

- Implementation: Create endpoints for updating and retrieving user profile data.

- Access Control Service: Controls access to specific assets or functionalities based on user roles.

- Features:

- Role management

- Permission assignment

- Implementation: Implement access control mechanisms using RBAC and create endpoints for role management.

Requirements:

- Create, update, and delete user profiles.

- Manage user preferences and settings.

- Implement access control mechanisms.

8. Reporting and Analytics Module

Description:

Generates reports and provides analytics on asset management.

Microservices:

- Reporting Service: Generates reports on asset utilization, maintenance history, and financial summaries.

- Features:

- Report generation

- Custom report templates

- Implementation: Use reporting tools like JasperReports or BIRT and create endpoints to generate and retrieve reports.

- Analytics Service: Provides data analytics and insights into asset performance and trends.

- Features:

- Data visualization

- Predictive analytics

- Implementation: Use data analytics tools like Pandas or Apache Spark and create APIs to provide analytical insights.

Requirements:

- Develop tools for generating various reports.

- Implement data analytics for asset performance.

9. Cost Management Module

Description:

Manages the costs associated with asset acquisition and maintenance.

Microservices:

- Cost Tracking Service: Tracks costs associated with each asset.

- Features:

- Expense logging

- Cost analysis

- Implementation: Develop endpoints to log and retrieve cost data from the database.

- Budget Management Service: Manages budgets related to asset acquisition and maintenance.

- Features:

- Budget creation

- Budget tracking

- Implementation: Create APIs to set, update, and track budget information.

Requirements:

- Track costs for asset acquisition and maintenance.

- Manage budgets and provide budget tracking tools.

10. Security Module

Description:

Ensures the security of assets and sensitive data.

Microservices:

- Asset Security Service: Implements security measures to protect assets.

- Features:

- Physical security measures

- Security monitoring

- Implementation: Use security protocols like encryption and create endpoints to manage security settings.

- Data Encryption Service: Encrypts sensitive data.

- Features:

- Data encryption

- Encryption key management

- Implementation: Implement data encryption methods and create APIs to manage encryption keys and settings.

Requirements:

- Implement security protocols to protect assets.

- Use encryption to secure sensitive data.

11. Mobile Access Module

Description:

Provides mobile access to the asset management system.

Microservices:

- Mobile Application Service: Provides mobile access to asset management functionalities.

- Features:

- Mobile app development

- Responsive design

- Implementation: Develop mobile apps using frameworks like React Native or Flutter and create APIs for mobile access.

- Offline Sync Service: Allows synchronization of data between mobile devices and the central system.

- Features:

- Offline data storage

- Data synchronization

- Implementation: Implement offline data storage and synchronization mechanisms in the mobile app.

Requirements:

- Develop mobile applications for asset management.

- Implement offline data synchronization.

12. Audit and Compliance Module

Description:

Ensures compliance with regulations and provides audit trails.

Microservices:

- Audit Trail Service: Records all actions within the system for auditing purposes.

- Features:

- Action logging

- Audit reports

- Implementation: Create APIs to log actions and retrieve audit trails from the database.

- Compliance Reporting Service: Prepares reports for compliance audits.

- Features:

- Compliance report generation

- Audit checklist management

- Implementation: Develop tools for generating compliance reports and managing audit checklists.

Requirements:

- Implement logging for audit trails.

- Generate compliance reports.

13. Asset Valuation Module

Description:

Determines the value of assets for financial reporting.

Microservices:

- Valuation Service: Determines the current value of assets.

- Features:

- Valuation methods (e.g., market value, book value)

- Valuation reports

- Implementation: Implement valuation algorithms and create endpoints to retrieve valuation data.

- Appraisal Service: Conducts asset appraisals.

- Features:

- Appraisal scheduling

- Appraisal history

- Implementation: Create APIs to schedule appraisals and log appraisal history.

Requirements:

- Implement valuation methods.

- Conduct appraisals and provide reports.

14. Helpdesk and Support Module

Description:

Provides support for asset-related issues and inquiries.

Microservices:

- Asset Support Service: Provides helpdesk support.

- Features:

- Support ticket creation

- Issue tracking

- Implementation: Develop a ticketing system with APIs to manage and track support tickets.

- Ticket Management Service: Manages support tickets.

- Features:

- Ticket assignment

- Ticket resolution

- Implementation: Create endpoints to assign, update, and resolve support tickets.

Requirements:

- Develop a helpdesk for asset support.

- Implement a ticketing system for managing issues.

15. Calibration and Quality Management Module

Description:

Ensures assets meet quality standards and calibration requirements.

Microservices:

- Calibration Service: Tracks calibration schedules and records.

- Features:

- Calibration scheduling

- Calibration history

- Implementation: Develop endpoints to schedule calibrations and log calibration activities.

- Quality Assurance Service: Ensures assets meet quality standards.

- Features:

- Quality checklists

- Quality audit reports

- Implementation: Implement quality assurance procedures and create APIs to manage and retrieve quality data.

Requirements:

- Track calibration schedules and records.

- Implement quality assurance checks.

16. Knowledge Management Module

Description:

Manages documentation and training related to asset management.

Microservices:

- Documentation Service: Manages documentation related to assets.

- Features:

- Document storage

- Document retrieval

- Implementation: Use a document management system and create APIs to store and retrieve documents.

- Training Service: Provides training materials and courses.

- Features:

- Training material management

- Training progress tracking

- Implementation: Develop a training portal with APIs to manage training materials and track progress.

Requirements:

- Manage documentation for asset specifications.

- Provide training materials and track training progress.

17. Fleet Management Module

Description:

Manages vehicles used as assets.

Microservices:

- Vehicle Tracking Service: Tracks vehicles.

- Features:

- GPS tracking

- Vehicle status updates

- Implementation: Use GPS integration for real-time tracking and create APIs to manage vehicle data.

- Fuel Management Service: Manages fuel usage and expenses.

- Features:

- Fuel consumption tracking

- Fuel expense reports

- Implementation: Develop endpoints to log and retrieve fuel consumption data and generate reports.

Requirements:

- Track vehicles using GPS.

- Manage fuel consumption and costs.

18. Environmental Management Module

Description:

Tracks the environmental impact of assets.

Microservices:

- Environmental Impact Service: Tracks environmental impacts.

- Features:

- Environmental metrics tracking

- Impact reports

- Implementation: Use environmental sensors or data sources and create APIs to log and retrieve environmental data.

- Sustainability Reporting Service: Prepares sustainability reports.

- Features:

- Sustainability metrics

- Reporting tools

- Implementation: Develop tools for generating sustainability reports and managing environmental data.

Requirements:

- Track environmental metrics.

- Generate sustainability reports.

19. Disaster Recovery Module

Description:

Ensures data backup and recovery for assets.

Microservices:

- Backup and Recovery Service: Manages backup and recovery processes.

- Features:

- Data backup scheduling

- Recovery procedures

- Implementation: Implement backup solutions and create APIs to manage and retrieve backup data.

- Disaster Response Service: Manages asset-related actions during disasters.

- Features:

- Disaster response plans

- Emergency procedures

- Implementation: Develop and implement disaster response plans and create endpoints to manage emergency procedures.

Requirements:

- Implement data backup and recovery processes.

- Develop disaster response plans.

20. Integration and API Gateway Module

Description:

Facilitates integration with external systems.

Microservices:

- Integration Service: Facilitates integration with external systems.

- Features:

- Data synchronization

- Integration management

- Implementation: Use integration tools like Apache Camel or MuleSoft and create APIs for data synchronization.

- API Gateway Service: Provides secure access to microservices.

- Features:

- API management

- Security features (rate limiting, throttling)

- Implementation: Implement an API gateway using tools like Kong or AWS API Gateway and create endpoints for managing API access.

Requirements:

- Develop APIs for integration.

- Implement an API gateway for secure access.

Each module contains microservices that interact through well-defined APIs or messaging protocols. This modular architecture supports scalability, flexibility, and ease of maintenance in the fixed asset management system.