项目文档

# Functional Requirement

# 1. Functional Requirements  
  
## 1.1 Asset Registration Function   
Function ID: FR-01   
Description: Administrators can register new assets in the system by providing details such as name, description, category, and department. The system generates a unique asset ID and stores the asset record.   
Input: Asset details including name, description, category, and department.   
Output: New asset record with a unique AssetID stored in the database.  
  
## 1.2 Asset Categorization Function   
Function ID: FR-02   
Description: Administrators can create and manage asset categories to classify assets. A unique category ID is generated and stored.   
Input: Category details including name, description, and relevant attributes.   
Output: New or updated AssetCategory record with a unique CategoryID stored in the database.  
  
## 1.3 Asset Allocation Function   
Function ID: FR-03   
Description: Administrators can allocate an asset to a specific user, specifying the allocation date and any additional notes. A record is stored in the system.   
Input: Asset and user selection, allocation date, and notes.   
Output: Updated Asset and UsageRecord entries, along with a PermissionAssignment record in the database.  
  
## 1.4 Asset Usage Tracking Function   
Function ID: FR-04   
Description: Administrators can record the usage of an asset by a user, including usage dates, activity type, and notes.   
Input: Selected asset and user, usage start and end times, activity type, and notes.   
Output: New UsageRecord entry stored in the database, with an AuditLog generated.  
  
## 1.5 Asset Return Function   
Function ID: FR-05   
Description: Administrators can process the return of an allocated asset. The system updates the asset status and closes or updates the usage record.   
Input: Selected asset and return confirmation with notes.   
Output: Updated Asset and UsageRecord entries, with an AuditLog generated.  
  
## 1.6 Asset Disposal Function   
Function ID: FR-06   
Description: Administrators can mark an asset as disposed, updating its status and recording the disposal action.   
Input: Selected asset and disposal confirmation with notes.   
Output: Updated Asset record with a Disposed status and an AuditLog generated.  
  
## 1.7 View Asset Information Function   
Function ID: FR-07   
Description: Administrators can view the details of a registered asset, including its category, department, status, and usage history.   
Input: Selected asset ID or name.   
Output: Displayed asset details retrieved from the database.  
  
## 1.8 Modify Asset Details Function   
Function ID: FR-08   
Description: Administrators can update the information of an asset, such as name, description, or department.   
Input: Selected asset and updated information fields.   
Output: Updated Asset record in the database, with an AuditLog generated.  
  
## 1.9 Manage Asset Inventory Function   
Function ID: FR-09   
Description: Administrators can manage the inventory of assets by adding, removing, or modifying quantity, location, and last updated time.   
Input: Selected asset and inventory action (add, remove, or modify).   
Output: Updated AssetInventory record in the database, with an AuditLog generated.  
  
## 1.10 View Usage Records Function   
Function ID: FR-10   
Description: Administrators can view the usage history of an asset, including all associated UsageRecord entries.   
Input: Selected asset ID or name.   
Output: Displayed list of usage records retrieved from the database.  
  
## 1.11 Generate Usage Report Function   
Function ID: FR-11   
Description: Administrators can generate reports based on usage records, filtered by date range, user, or asset category.   
Input: Date range, user, or asset category for report filtering.   
Output: Formatted usage report displayed or downloaded by the Administrator.  
  
## 1.12 View Audit Logs Function   
Function ID: FR-12   
Description: Administrators can view all audit logs, including timestamp, action, and affected entity.   
Input: Filtering criteria such as date range, user, or action type.   
Output: Displayed list of audit logs retrieved from the database.  
  
## 1.13 Export Audit Logs Function   
Function ID: FR-13   
Description: Administrators can export audit logs in a selected format (e.g., CSV, PDF) based on filtering criteria.   
Input: Filtering criteria and selected export format.   
Output: Exported audit logs in the selected format, with an AuditLog entry for the export action.  
  
## 1.14 Send Email Notifications Function   
Function ID: FR-14   
Description: The system automatically sends email notifications to users and administrators based on predefined events.   
Input: Event details (e.g., asset allocation, return, or disposal) and recipient email address.   
Output: EmailNotification record stored in the database, and an email sent to the recipient.  
  
## 1.15 Configure Email Settings Function   
Function ID: FR-15   
Description: Administrators can configure the system's email notification settings, including SMTP host, port, and sender email.   
Input: Email server configuration details (e.g., SMTP host, port, authentication credentials, and sender email).   
Output: Updated EmailNotification settings stored in the database, with an AuditLog generated.  
  
## 1.16 Assign Permissions Function   
Function ID: FR-16   
Description: Administrators can assign specific permissions to users for managing assets.   
Input: Selected user and one or more permissions to assign.   
Output: Updated PermissionAssignment record in the database, with an AuditLog generated.  
  
## 1.17 Revoke Permissions Function   
Function ID: FR-17   
Description: Administrators can revoke previously assigned permissions from users.   
Input: Selected user and one or more permissions to revoke.   
Output: Updated PermissionAssignment record in the database, with an AuditLog generated.  
  
## 1.18 View Permission Settings Function   
Function ID: FR-18   
Description: Administrators can view the permission assignments for a selected user.   
Input: Selected user ID or name.   
Output: Displayed list of assigned permissions retrieved from the database.  
  
## 1.19 Manage Permission Assignment Function   
Function ID: FR-19   
Description: Administrators can manage permission assignments, including assigning, revoking, or modifying permissions for users.   
Input: Selected user and action (assign, revoke, or modify).   
Output: Updated PermissionAssignment record in the database, with an AuditLog generated.  
  
## 1.20 User Registration Function   
Function ID: FR-20   
Description: Administrators can register new users in the system by providing details such as name, email, department, and role.   
Input: User details including name, email, department, and role.   
Output: New User record with a unique UserID stored in the database.  
  
## 1.21 User Login Function   
Function ID: FR-21   
Description: Users can log in to the system using their username and password. The system verifies credentials and creates a session.   
Input: Username and password.   
Output: Authenticated session for the user with a redirect to the appropriate dashboard.  
  
## 1.22 Department Setup Function   
Function ID: FR-22   
Description: Administrators can create new departments in the system by providing details such as name, description, and manager.   
Input: Department details including name, description, and manager information.   
Output: New Department record with a unique DepartmentID stored in the database.  
  
## 1.23 Update Department Information Function   
Function ID: FR-23   
Description: Administrators can update the information of an existing department, including name, description, and manager.   
Input: Selected department and updated information fields.   
Output: Updated Department record in the database, with an AuditLog generated.  
  
## 1.24 Delete Department Function   
Function ID: FR-24   
Description: Administrators can delete a department from the system, provided it has no associated assets or users.   
Input: Selected department and deletion confirmation.   
Output: Deleted Department record from the database, with an AuditLog generated.  
  
## 1.25 Manage Asset Status Function   
Function ID: FR-25   
Description: Administrators can update the status of an asset (e.g., Active, Inactive, Maintenance, Disposed, Allocated).   
Input: Selected asset and new status.   
Output: Updated Asset and AssetStatus records in the database, with an AuditLog generated.  
  
## 1.26 View Asset Status Function   
Function ID: FR-26   
Description: Administrators can view the current status of an asset, including the timestamp of the last status change and any related notes.   
Input: Selected asset ID or name.   
Output: Displayed asset status and related information retrieved from the database.  
  
## 1.27 Manage Usage Record Function   
Function ID: FR-27   
Description: Administrators can update or delete an existing usage record.   
Input: Selected usage record and modification or deletion request.   
Output: Updated or deleted UsageRecord entry in the database, with an AuditLog generated.

# External Description

# 2. External Interfaces  
  
## 2.1 User Interface Output   
The system interacts with users through a web-based graphical user interface (GUI), allowing administrators and users to perform actions and view information. The interface is designed to be intuitive and user-friendly, with the following components and interactions:   
  
- \*\*Asset Registration Form\*\*: Administrators enter asset details (name, description, category, and department) via a form. Upon submission, the system displays a confirmation message with the new asset ID.   
- \*\*Asset Categorization Form\*\*: Administrators create or modify asset categories by inputting category name, description, and attributes. A confirmation message with the new or updated category ID is displayed.   
- \*\*Asset Allocation Interface\*\*: Administrators select an asset and a user, input the allocation date and notes, and submit the allocation. The system displays a confirmation and updates the UI to reflect the allocation.   
- \*\*Asset Usage Tracking Interface\*\*: Administrators record usage by selecting an asset and user, specifying usage start and end times, activity type, and notes. A confirmation is displayed, and the usage history is updated on the screen.   
- \*\*Asset Return Confirmation\*\*: Administrators confirm the return of an allocated asset, and the system updates the asset status and usage record accordingly in the UI.   
- \*\*Asset Disposal Confirmation\*\*: Administrators confirm the disposal of an asset, and the system updates the asset status to "Disposed" and logs the action in the UI.   
- \*\*Asset Information View\*\*: Displays detailed asset information including category, department, status, and usage history based on a selected asset ID or name.   
- \*\*Modify Asset Details Form\*\*: Allows administrators to update asset information (e.g., name, description, department). After submission, the system displays the updated asset details and logs the action.   
- \*\*Manage Asset Inventory Interface\*\*: Enables administrators to add, remove, or modify inventory details such as quantity, location, and last updated time. A confirmation message is displayed upon completion.   
- \*\*View Usage Records Interface\*\*: Displays a list of usage records for a selected asset, including timestamps, activity types, and notes.   
- \*\*Generate Usage Report Interface\*\*: Provides a form to select report filters (date range, user, asset category) and generates a formatted usage report for display or download.   
- \*\*View Audit Logs Interface\*\*: Displays a list of audit logs with timestamps, actions, and affected entities, filtered by specified criteria (e.g., date range, user, action type).   
- \*\*Export Audit Logs Interface\*\*: Allows administrators to select audit logs and an export format (e.g., CSV, PDF) and provides a downloadable file of the logs.   
- \*\*Email Notification Interface\*\*: Displays a record of emails sent (e.g., asset allocation, return, or disposal) and allows viewing event details and recipient information.   
- \*\*Configure Email Settings Form\*\*: Provides a form to input and update email server configuration details (e.g., SMTP host, port, authentication credentials, and sender email). The system displays confirmation of the updated settings.   
- \*\*Permission Assignment Interface\*\*: Enables administrators to assign, revoke, or modify permissions for users. A confirmation message is displayed, and the system updates the UI to reflect permission changes.   
- \*\*View Permission Settings Interface\*\*: Displays a list of permissions assigned to a selected user, including details such as permission type and timestamp.   
- \*\*User Registration Form\*\*: Administrators register new users by inputting user details (name, email, department, and role). A confirmation message with the new user ID is displayed.   
- \*\*User Login Interface\*\*: Users enter their username and password to log in. Upon successful authentication, the system redirects them to the appropriate dashboard.   
- \*\*Department Setup Form\*\*: Administrators create new departments by inputting department name, description, and manager information. A confirmation message with the new department ID is displayed.   
- \*\*Update Department Information Form\*\*: Allows administrators to update department details. A confirmation message is displayed, and the system reflects the changes in the UI.   
- \*\*Delete Department Interface\*\*: Provides a confirmation prompt for deleting a department. If successful, the system removes the department and updates the UI.   
- \*\*Manage Asset Status Interface\*\*: Enables administrators to update an asset's status (e.g., Active, Inactive, Maintenance, Disposed, Allocated). The system displays the new status and logs the action.   
- \*\*View Asset Status Interface\*\*: Displays the current status of an asset, along with the timestamp of the last status change and related notes.   
- \*\*Manage Usage Record Interface\*\*: Allows administrators to update or delete existing usage records. A confirmation message is displayed, and the system updates the UI accordingly.  
  
## 2.2 Hardware Interface Output   
The system does not have direct interaction with external hardware devices. Therefore, no hardware interfaces are defined in this document.   
  
## 2.3 Software Interface Output   
The system interacts with several software components and databases to store, retrieve, and manage asset and user data. These interfaces are described as follows:   
  
### 2.3.1 Asset Database Interface   
- \*\*Description\*\*: Stores and retrieves asset records, including asset ID, name, description, category, department, status, and other relevant details.   
- \*\*Interaction Method\*\*: CRUD (Create, Read, Update, Delete) operations via the system's internal database layer.   
- \*\*Inputs\*\*: Asset details (name, description, category, department, status, etc.), modification requests (e.g., update or delete).   
- \*\*Outputs\*\*: Asset records (new or updated), error messages in case of invalid data or constraints.   
  
### 2.3.2 AssetCategory Database Interface   
- \*\*Description\*\*: Stores and retrieves asset category records, including category ID, name, description, and relevant attributes.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: Category details (name, description, attributes), modification requests (e.g., update or delete).   
- \*\*Outputs\*\*: AssetCategory records (new or updated), error messages for invalid data or constraints.   
  
### 2.3.3 UsageRecord Database Interface   
- \*\*Description\*\*: Stores and retrieves records of asset usage, including timestamps, user ID, asset ID, activity type, and notes.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: Asset and user selection, usage start and end times, activity type, and notes.   
- \*\*Outputs\*\*: UsageRecord entries (new, updated, or deleted), error messages for invalid usage data.   
  
### 2.3.4 PermissionAssignment Database Interface   
- \*\*Description\*\*: Stores and retrieves user permission assignments, including user ID, permission type, and timestamp of assignment.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: User selection, permission type, and action (assign, revoke, or modify).   
- \*\*Outputs\*\*: PermissionAssignment records (updated or deleted), error messages for invalid permission configurations.   
  
### 2.3.5 AuditLog Database Interface   
- \*\*Description\*\*: Logs all system actions (e.g., asset allocation, modification, disposal, and permission changes) with timestamps, action type, and affected entity.   
- \*\*Interaction Method\*\*: Write-only operations via the system's internal database layer.   
- \*\*Inputs\*\*: Action details (e.g., type, timestamp, entity ID).   
- \*\*Outputs\*\*: AuditLog entries (new or updated), no direct output to user interface.   
  
### 2.3.6 EmailNotification Database Interface   
- \*\*Description\*\*: Stores records of email notifications sent by the system, including event type, recipient email address, timestamp, and status (e.g., sent, failed).   
- \*\*Interaction Method\*\*: Write-only operations via the system's internal database layer.   
- \*\*Inputs\*\*: Event details (e.g., asset allocation, return, or disposal) and recipient email address.   
- \*\*Outputs\*\*: EmailNotification records (new), no direct output to user interface.   
  
### 2.3.7 Email Server Interface   
- \*\*Description\*\*: Configures and sends email notifications using an external email server (e.g., SMTP server).   
- \*\*Interaction Method\*\*: Uses email server configuration (e.g., host, port, authentication credentials) to send emails to recipients.   
- \*\*Inputs\*\*: Email server settings (host, port, username, password, sender email), recipient email addresses, and message content.   
- \*\*Outputs\*\*: Emails sent to users/administrators, and EmailNotification records stored in the database.   
  
### 2.3.8 Department Database Interface   
- \*\*Description\*\*: Stores and retrieves department records, including department ID, name, description, and manager information.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: Department details (name, description, manager), modification or deletion requests.   
- \*\*Outputs\*\*: Department records (new, updated, or deleted), error messages for invalid department data.   
  
### 2.3.9 User Database Interface   
- \*\*Description\*\*: Stores and retrieves user records, including user ID, name, email, department, and role.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: User details (name, email, department, role), modification or deletion requests.   
- \*\*Outputs\*\*: User records (new, updated, or deleted), error messages for invalid user data.   
  
### 2.3.10 AssetStatus Database Interface   
- \*\*Description\*\*: Tracks and stores asset status changes, including status type (e.g., Active, Inactive, Maintenance, Disposed, Allocated) and timestamps.   
- \*\*Interaction Method\*\*: Write-only operations via the system's internal database layer.   
- \*\*Inputs\*\*: Asset ID and new status.   
- \*\*Outputs\*\*: AssetStatus records (updated), no direct output to user interface.   
  
### 2.3.11 AssetInventory Database Interface   
- \*\*Description\*\*: Stores and retrieves inventory data for assets, such as quantity, location, and last updated time.   
- \*\*Interaction Method\*\*: CRUD operations via the system's internal database layer.   
- \*\*Inputs\*\*: Inventory action (add, remove, or modify) and relevant asset details.   
- \*\*Outputs\*\*: Updated AssetInventory records, error messages for invalid inventory actions.   
  
## 2.4 Communication Interface Output   
The system communicates with external entities via the following interfaces:   
  
### 2.4.1 Email Notification Communication Interface   
- \*\*Description\*\*: Sends email notifications to users and administrators based on predefined events such as asset allocation, return, or disposal.   
- \*\*Interaction Method\*\*: Uses the configured email server (e.g., SMTP) to send emails in plain text or HTML format.   
- \*\*Inputs\*\*: Recipient email address, event type, and message content.   
- \*\*Outputs\*\*: Email sent to the recipient, and a record stored in the EmailNotification database.   
  
### 2.4.2 Audit Log Communication Interface   
- \*\*Description\*\*: Logs all system actions and changes in the AuditLog database for transparency and tracking.   
- \*\*Interaction Method\*\*: Internal logging mechanism that writes to the AuditLog table.   
- \*\*Inputs\*\*: Action type, timestamp, user ID, and affected entity ID.   
- \*\*Outputs\*\*: AuditLog entry stored in the database.   
  
### 2.4.3 Export Audit Logs Communication Interface   
- \*\*Description\*\*: Exports audit logs in a selected format (e.g., CSV, PDF) based on filtering criteria such as date range, user, or action type.   
- \*\*Interaction Method\*\*: Generates files in the specified format and provides a download link or file to the user.   
- \*\*Inputs\*\*: Filtering criteria and export format selection.   
- \*\*Outputs\*\*: Exported audit logs file (e.g., CSV, PDF), and an AuditLog entry for the export action.   
  
### 2.4.4 Web Session Communication Interface   
- \*\*Description\*\*: Manages user authentication and session creation after a successful login.   
- \*\*Interaction Method\*\*: Validates user credentials against the User database and creates a session token for subsequent interactions.   
- \*\*Inputs\*\*: Username and password.   
- \*\*Outputs\*\*: Authenticated session token and a redirect to the appropriate dashboard.   
  
---  
  
This section defines all external interfaces necessary for the system to interact with users, databases, and external communication systems. Each interface is clearly described with its role, interaction method, and relevant inputs and outputs to ensure clarity and ease of implementation.

# Use Case

Use Case Name: Asset Registration   
Use Case ID: UC-01   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to register assets.   
2. The asset category database is populated with valid categories.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. A new asset record is successfully created in the system.   
2. The asset information is stored in the database.   
3. An audit log is generated to record the registration activity.   
4. The Administrator is notified of the successful registration.   
  
Main Flow:   
1. The Administrator navigates to the Asset Registration interface in the system.   
2. The Administrator selects an asset category from the dropdown list.   
3. The Administrator enters the asset details, including name, description, location, and any additional relevant information.   
4. The Administrator submits the asset registration request.   
5. The system validates the input data and checks for any missing or invalid fields.   
6. The system generates a unique asset ID and stores the asset information in the database.   
7. The system creates an audit log entry to record the registration action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the asset registration.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the registration request.   
3. If the selected asset category is not valid or does not exist, the system displays an error message and prompts the Administrator to select a valid category.   
4. If the system is unable to generate a unique asset ID, it displays an error message and prevents the registration.   
5. If the email notification fails to send, the system logs the error and displays a message to the Administrator indicating that the notification could not be sent, but the asset registration was still successful.  
  
Use Case Name: Asset Categorization   
Use Case ID: UC-02   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset categories.   
2. The system is operational and accessible to the Administrator.   
3. The asset category database is initialized and ready for updates.   
  
Postconditions:   
1. A new asset category is successfully added to the system.   
2. The asset category information is stored in the database.   
3. An audit log is generated to record the categorization activity.   
4. The Administrator is notified of the successful asset category creation.   
  
Main Flow:   
1. The Administrator navigates to the Asset Categorization interface in the system.   
2. The Administrator enters the details of the new asset category, including name, description, and any relevant attributes.   
3. The Administrator submits the asset category creation request.   
4. The system validates the input data and checks for any missing or invalid fields.   
5. The system generates a unique category ID and stores the category information in the database.   
6. The system creates an audit log entry to record the categorization action.   
7. The system displays a confirmation message to the Administrator.   
8. The system sends an email notification to the Administrator confirming the asset category creation.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the category creation request.   
3. If the system is unable to generate a unique category ID, it displays an error message and prevents the category from being added.   
4. If the entered category name already exists, the system displays an error message and prompts the Administrator to enter a unique name.   
5. If the email notification fails to send, the system logs the error and displays a message to the Administrator indicating that the notification could not be sent, but the asset category creation was still successful.  
  
Use Case Name: Asset Allocation   
Use Case ID: UC-03   
Actors: Administrator, User, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to allocate assets.   
2. The asset database contains at least one registered asset.   
3. The user database contains valid user records.   
4. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. An asset is successfully allocated to a user.   
2. The allocation information is stored in the database.   
3. An audit log is generated to record the allocation activity.   
4. The Administrator and User are notified of the successful allocation.   
  
Main Flow:   
1. The Administrator navigates to the Asset Allocation interface in the system.   
2. The Administrator selects an asset from the available list.   
3. The Administrator selects a user to whom the asset will be allocated.   
4. The Administrator enters the allocation date and any additional notes.   
5. The Administrator submits the allocation request.   
6. The system validates the selected asset and user, ensuring they are available for allocation.   
7. The system updates the asset record to reflect the allocation and stores the information in the database.   
8. The system creates an audit log entry to record the allocation action.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the Administrator and the User confirming the allocation.   
  
Alternative Flow:   
1. If the selected asset is already allocated, the system displays an error message and prompts the Administrator to choose an available asset.   
2. If the selected user does not exist or is invalid, the system displays an error message and prompts the Administrator to choose a valid user.   
3. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the allocation request.   
5. If the email notification fails to send, the system logs the error and displays a message to the Administrator indicating that the notification could not be sent, but the allocation was still successful.  
  
Use Case Name: Asset Usage Tracking   
Use Case ID: UC-04   
Actors: Administrator, User, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to track asset usage.   
2. The asset database contains at least one allocated asset.   
3. The user database contains valid user records.   
4. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset usage record is successfully created and stored in the system.   
2. An audit log is generated to record the usage tracking activity.   
3. The User is notified of the usage tracking.   
4. The Administrator is informed that the usage has been recorded.   
  
Main Flow:   
1. The Administrator navigates to the Asset Usage Tracking interface.   
2. The Administrator selects an allocated asset from the list.   
3. The Administrator chooses the user associated with the asset.   
4. The Administrator enters the usage details, including start and end times, activity type, and notes.   
5. The Administrator submits the usage tracking request.   
6. The system validates the asset status and user information.   
7. The system creates a new usage record and stores it in the database.   
8. The system generates an audit log entry for the usage tracking.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the User confirming the recorded usage.   
  
Alternative Flow:   
1. If the selected asset is not allocated, the system displays an error message and prompts the Administrator to select an allocated asset.   
2. If the selected user is not associated with the asset, the system displays an error message and asks the Administrator to correct the selection.   
3. If the usage time range is invalid (e.g., end time before start time), the system shows an error and requires correction.   
4. If the input data is incomplete or invalid, the system displays an error message specifying the missing or incorrect fields.   
5. The Administrator corrects the input data and resubmits the request.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the usage record was successfully created.  
  
Use Case Name: Asset Return   
Use Case ID: UC-05   
Actors: Administrator, User, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to process asset returns.   
2. The asset database contains at least one allocated asset.   
3. The user database contains valid user records.   
4. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset is successfully marked as returned and updated in the database.   
2. The usage record is updated or closed in the system.   
3. An audit log is generated to record the return activity.   
4. The Administrator and User are notified of the successful return.   
  
Main Flow:   
1. The Administrator navigates to the Asset Return interface in the system.   
2. The Administrator selects an allocated asset from the list.   
3. The Administrator confirms the asset is to be returned and enters any relevant notes.   
4. The Administrator submits the return request.   
5. The system validates the asset status and confirms it is currently allocated.   
6. The system updates the asset record to reflect the return status and stores the change in the database.   
7. The system updates or closes the associated usage record.   
8. The system creates an audit log entry to record the return action.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the Administrator and the User confirming the asset return.   
  
Alternative Flow:   
1. If the selected asset is not allocated, the system displays an error message and prompts the Administrator to select an allocated asset.   
2. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
3. The Administrator corrects the input data and resubmits the return request.   
4. If the system fails to update the asset record, it displays an error message and prevents the return process.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the asset return was still processed.  
  
Use Case Name: Asset Disposal   
Use Case ID: UC-06   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to dispose of assets.   
2. The asset database contains at least one registered asset.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset is successfully marked as disposed and updated in the database.   
2. An audit log is generated to record the disposal activity.   
3. The Administrator is notified of the successful disposal.   
  
Main Flow:   
1. The Administrator navigates to the Asset Disposal interface in the system.   
2. The Administrator selects an asset to be disposed of from the list.   
3. The Administrator confirms the disposal and enters any relevant notes or reasons for disposal.   
4. The Administrator submits the disposal request.   
5. The system validates the selected asset and confirms it is eligible for disposal (e.g., not in active use).   
6. The system updates the asset record to reflect the disposal status and stores the change in the database.   
7. The system creates an audit log entry to record the disposal action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the asset disposal.   
  
Alternative Flow:   
1. If the selected asset is currently allocated, the system displays an error message and prompts the Administrator to return the asset first.   
2. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
3. The Administrator corrects the input data and resubmits the disposal request.   
4. If the system fails to update the asset record, it displays an error message and prevents the disposal process.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the disposal was still processed.  
  
Use Case Name: View Asset Information   
Use Case ID: UC-07   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view asset information.   
2. The asset database contains at least one registered asset.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset information is displayed to the Administrator.   
2. An audit log is generated to record the view activity.   
3. The Administrator is informed that the information has been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Asset Information interface in the system.   
2. The Administrator selects an asset from the list of registered assets.   
3. The system retrieves the selected asset's information from the database.   
4. The system displays the asset details to the Administrator, including its category, allocation status, and usage history.   
5. The system creates an audit log entry to record the viewing action.   
6. The system displays a confirmation message to the Administrator that the information has been successfully retrieved.   
  
Alternative Flow:   
1. If the selected asset does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid asset.   
2. If the asset information cannot be retrieved due to database issues, the system displays an error message and logs the failure in the audit log.   
3. If the Administrator does not have permission to view the asset information, the system denies access and displays an appropriate error message.  
  
Use Case Name: Modify Asset Details   
Use Case ID: UC-08   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to modify asset details.   
2. The asset database contains at least one registered asset.   
3. The selected asset is not currently in use or allocated.   
4. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset details are successfully updated in the system.   
2. The updated asset information is stored in the database.   
3. An audit log is generated to record the modification activity.   
4. The Administrator is notified of the successful modification.   
  
Main Flow:   
1. The Administrator navigates to the Asset Details interface in the system.   
2. The Administrator selects an asset from the list to modify.   
3. The system displays the current asset details.   
4. The Administrator makes the necessary changes to the asset information.   
5. The Administrator submits the modification request.   
6. The system validates the updated data and checks for any missing or invalid fields.   
7. The system updates the asset record in the database.   
8. The system creates an audit log entry to record the modification action.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the Administrator confirming the asset details have been modified.   
  
Alternative Flow:   
1. If the selected asset is currently allocated or in use, the system displays an error message and prompts the Administrator to return or unallocate the asset before modification.   
2. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
3. The Administrator corrects the input data and resubmits the modification request.   
4. If the system fails to update the asset record, it displays an error message and prevents the modification.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the asset modification was still processed.  
  
Use Case Name: Manage Asset Inventory   
Use Case ID: UC-09   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset inventory.   
2. The asset database contains registered assets.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The asset inventory is updated based on the Administrator's actions (e.g., adding, removing, or modifying inventory items).   
2. The system stores the updated inventory information in the database.   
3. An audit log is generated to record the inventory management activity.   
4. The Administrator is notified of the successful inventory update.   
  
Main Flow:   
1. The Administrator navigates to the Asset Inventory Management interface in the system.   
2. The Administrator selects an action to perform (e.g., add, remove, or modify an inventory item).   
3. The Administrator enters or modifies the relevant asset information (e.g., quantity, status, or location).   
4. The Administrator submits the inventory management request.   
5. The system validates the input data and checks for any missing or invalid fields.   
6. The system updates the asset inventory records in the database.   
7. The system creates an audit log entry to document the inventory change.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the inventory update.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the inventory management request.   
3. If the selected asset does not exist in the database, the system displays an error message and prompts the Administrator to select a valid asset.   
4. If the system fails to update the inventory records, it displays an error message and prevents the change.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the inventory update was still processed.  
  
Use Case Name: User Registration   
Use Case ID: UC-10   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to register users.   
2. The user database is initialized and ready for updates.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. A new user record is successfully created in the system.   
2. The user information is stored in the database.   
3. An audit log is generated to record the registration activity.   
4. The Administrator is notified of the successful registration.   
  
Main Flow:   
1. The Administrator navigates to the User Registration interface in the system.   
2. The Administrator enters the user details, including name, email, department, and role.   
3. The Administrator selects the appropriate permissions for the user.   
4. The Administrator submits the user registration request.   
5. The system validates the input data and checks for any missing or invalid fields.   
6. The system generates a unique user ID and stores the user information in the database.   
7. The system creates an audit log entry to record the registration action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the user registration.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the registration request.   
3. If the system is unable to generate a unique user ID, it displays an error message and prevents the registration.   
4. If the entered email is already registered, the system displays an error message and prompts the Administrator to enter a new email.   
5. If the email notification fails to send, the system logs the error and displays a message to the Administrator indicating that the notification could not be sent, but the user registration was still successful.  
  
Use Case Name: User Login   
Use Case ID: UC-11   
Actors: User, System   
Preconditions:   
1. The system is operational and accessible to the user.   
2. The user has a valid user account in the system.   
3. The user is not currently logged in.   
  
Postconditions:   
1. The user is successfully authenticated and logged into the system.   
2. The system updates the user session status.   
3. An audit log is generated to record the login activity.   
4. The user is redirected to the appropriate dashboard based on their role and permissions.   
  
Main Flow:   
1. The user opens the login interface of the system.   
2. The user enters their username and password.   
3. The user clicks the "Login" button.   
4. The system verifies the entered credentials against the user database.   
5. If the credentials are valid, the system authenticates the user and creates a session.   
6. The system updates the user's login status in the database.   
7. The system creates an audit log entry to record the successful login.   
8. The system displays the user's dashboard or main interface.   
  
Alternative Flow:   
1. If the username or password is incorrect, the system displays an error message and prompts the user to re-enter their credentials.   
2. If the user account is locked or disabled, the system displays an error message and denies access.   
3. If the system is unable to create a session, it displays an error message and prevents login.   
4. If the audit log cannot be generated, the system logs the failure internally and informs the user that the login was successful but the audit could not be recorded.  
  
Use Case Name: Assign Permissions   
Use Case ID: UC-12   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to assign permissions.   
2. The system contains valid user records.   
3. The system contains valid permission definitions.   
4. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The permissions are successfully assigned to the selected user.   
2. The permission assignment is stored in the database.   
3. An audit log is generated to record the permission assignment activity.   
4. The Administrator is notified of the successful permission assignment.   
  
Main Flow:   
1. The Administrator navigates to the Permission Assignment interface in the system.   
2. The Administrator selects a user from the user list.   
3. The Administrator chooses one or more permissions to assign to the user.   
4. The Administrator submits the permission assignment request.   
5. The system validates the selected user and permissions.   
6. The system updates the user's permission records in the database.   
7. The system creates an audit log entry to document the permission assignment action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the permission assignment.   
  
Alternative Flow:   
1. If the selected user does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid user.   
2. If the selected permissions are not valid or do not exist, the system displays an error message and asks the Administrator to choose valid permissions.   
3. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the permission assignment request.   
5. If the system fails to update the permission records, it displays an error message and prevents the assignment.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the permission assignment was still processed.  
  
Use Case Name: Revoke Permissions   
Use Case ID: UC-13   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to revoke user permissions.   
2. The system contains valid user records with assigned permissions.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The selected permissions are successfully revoked from the user.   
2. The system updates the permission records in the database.   
3. An audit log is generated to record the revocation activity.   
4. The Administrator is notified of the successful revocation.   
  
Main Flow:   
1. The Administrator navigates to the Permission Revocation interface in the system.   
2. The Administrator selects a user from the list whose permissions will be revoked.   
3. The Administrator chooses one or more permissions to revoke.   
4. The Administrator submits the revocation request.   
5. The system validates the selected user and permissions.   
6. The system updates the user's permission records in the database to reflect the revocation.   
7. The system creates an audit log entry to document the revocation action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the permission revocation.   
  
Alternative Flow:   
1. If the selected user does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid user.   
2. If the selected permissions are not assigned to the user, the system displays an error message and asks the Administrator to choose valid permissions to revoke.   
3. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the revocation request.   
5. If the system fails to update the permission records, it displays an error message and prevents the revocation.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the permission revocation was still processed.  
  
Use Case Name: View Permission Settings   
Use Case ID: UC-14   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view permission settings.   
2. The system contains at least one user with assigned permissions.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The permission settings for the selected user are displayed to the Administrator.   
2. An audit log is generated to record the viewing of permission settings.   
3. The Administrator is informed that the permission settings have been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Permission Settings interface in the system.   
2. The Administrator selects a user from the user list to view their permissions.   
3. The system retrieves the permission assignments for the selected user from the database.   
4. The system displays the permission settings to the Administrator, including permission name, status, and scope.   
5. The system creates an audit log entry to record the viewing action.   
6. The system displays a confirmation message to the Administrator that the permission settings have been successfully retrieved.   
  
Alternative Flow:   
1. If the selected user does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid user.   
2. If the system cannot retrieve the permission settings due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the Administrator does not have permission to view permission settings, the system denies access and displays an appropriate error message.  
  
Use Case Name: Department Setup   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to set up departments.   
2. The system is operational and accessible to the Administrator.   
3. The department database is initialized and ready for updates.   
  
Postconditions:   
1. A new department record is successfully created in the system.   
2. The department information is stored in the database.   
3. An audit log is generated to record the department setup activity.   
4. The Administrator is notified of the successful department setup.   
  
Main Flow:   
1. The Administrator navigates to the Department Setup interface in the system.   
2. The Administrator enters the department details, including name, description, and any relevant attributes.   
3. The Administrator submits the department setup request.   
4. The system validates the input data and checks for any missing or invalid fields.   
5. The system generates a unique department ID and stores the department information in the database.   
6. The system creates an audit log entry to record the department setup action.   
7. The system displays a confirmation message to the Administrator.   
8. The system sends an email notification to the Administrator confirming the department setup.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the department setup request.   
3. If the system is unable to generate a unique department ID, it displays an error message and prevents the department from being added.   
4. If the entered department name already exists, the system displays an error message and prompts the Administrator to enter a unique name.   
5. If the email notification fails to send, the system logs the error and displays a message to the Administrator indicating that the notification could not be sent, but the department setup was still successful.  
  
Use Case Name: Update Department Information   
Use Case ID: UC-16   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to update department information.   
2. The department database contains at least one registered department.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The department information is successfully updated in the system.   
2. The updated department record is stored in the database.   
3. An audit log is generated to record the update activity.   
4. The Administrator is notified of the successful update.   
  
Main Flow:   
1. The Administrator navigates to the Department Information interface in the system.   
2. The Administrator selects a department from the list of registered departments.   
3. The system displays the current department details.   
4. The Administrator makes the necessary changes to the department information.   
5. The Administrator submits the update request.   
6. The system validates the updated data and checks for any missing or invalid fields.   
7. The system updates the department record in the database.   
8. The system creates an audit log entry to document the update action.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the Administrator confirming the department information has been updated.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the update request.   
3. If the selected department does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid department.   
4. If the system fails to update the department record, it displays an error message and prevents the change.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the department update was still processed.  
  
Use Case Name: Delete Department   
Use Case ID: UC-17   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to delete departments.   
2. The department database contains at least one registered department.   
3. The system is operational and accessible to the Administrator.   
4. The selected department has no assets or users currently associated with it.   
  
Postconditions:   
1. The selected department is successfully deleted from the system.   
2. An audit log is generated to record the deletion activity.   
3. The Administrator is notified of the successful deletion.   
  
Main Flow:   
1. The Administrator navigates to the Department Management interface in the system.   
2. The Administrator selects a department from the list of registered departments to delete.   
3. The Administrator confirms the deletion of the selected department.   
4. The system validates the department status and confirms there are no assets or users linked to it.   
5. The system deletes the selected department record from the database.   
6. The system creates an audit log entry to document the deletion action.   
7. The system displays a confirmation message to the Administrator.   
8. The system sends an email notification to the Administrator confirming the department has been deleted.   
  
Alternative Flow:   
1. If the selected department does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid department.   
2. If the department has assets or users associated with it, the system displays an error message and prevents the deletion.   
3. If the system fails to delete the department record, it displays an error message and prevents the deletion.   
4. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the department deletion was still processed.  
  
Use Case Name: View Asset Categories   
Use Case ID: UC-18   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view asset categories.   
2. The asset category database contains at least one registered category.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The list of asset categories is displayed to the Administrator.   
2. An audit log is generated to record the viewing activity.   
3. The Administrator is informed that the category list has been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Asset Categories interface in the system.   
2. The Administrator requests to view the list of asset categories.   
3. The system retrieves the asset categories from the database.   
4. The system displays the list of categories to the Administrator, including category name, description, and ID.   
5. The system creates an audit log entry to record the viewing action.   
6. The system displays a confirmation message to the Administrator that the category list has been successfully retrieved.   
  
Alternative Flow:   
1. If the asset category database is empty or unavailable, the system displays an error message and informs the Administrator that no categories are available.   
2. If the system cannot retrieve the asset categories due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the Administrator does not have permission to view asset categories, the system denies access and displays an appropriate error message.  
  
Use Case Name: Update Asset Category   
Use Case ID: UC-19   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to update asset categories.   
2. The asset category database contains at least one registered category.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The selected asset category is successfully updated in the system.   
2. The updated category information is stored in the database.   
3. An audit log is generated to record the update activity.   
4. The Administrator is notified of the successful update.   
  
Main Flow:   
1. The Administrator navigates to the Asset Category Management interface in the system.   
2. The Administrator selects an existing asset category to update.   
3. The system displays the current category details (name, description, and attributes).   
4. The Administrator modifies the necessary information for the category.   
5. The Administrator submits the update request.   
6. The system validates the updated data and checks for any missing or invalid fields.   
7. The system updates the asset category record in the database.   
8. The system creates an audit log entry to document the update action.   
9. The system displays a confirmation message to the Administrator.   
10. The system sends an email notification to the Administrator confirming the asset category has been updated.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the update request.   
3. If the selected category does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid category.   
4. If the system fails to update the category record, it displays an error message and prevents the change.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the asset category update was still processed.  
  
Use Case Name: Add Asset Category   
Use Case ID: UC-02   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset categories.   
2. The system is operational and accessible to the Administrator.   
3. The asset category database is initialized and ready for updates.   
  
Postconditions:   
1. A new asset category is successfully added to the system.   
2. The category information is stored in the database.   
3. An audit log is generated to record the addition of the asset category.   
4. The Administrator is notified of the successful addition.   
  
Main Flow:   
1. The Administrator navigates to the Asset Category Management interface in the system.   
2. The Administrator selects the option to add a new asset category.   
3. The Administrator enters the category details, including name, description, and any relevant attributes.   
4. The Administrator submits the request to add the category.   
5. The system validates the input data and checks for missing or invalid fields.   
6. The system generates a unique category ID and stores the category in the database.   
7. The system creates an audit log entry to record the addition of the asset category.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the addition of the asset category.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the request.   
3. If the category name already exists in the database, the system displays an error message and prompts the Administrator to enter a unique name.   
4. If the system is unable to generate a unique category ID, it displays an error message and prevents the category from being added.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the category was successfully added.  
  
Use Case Name: View Usage Records   
Use Case ID: UC-10   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view usage records.   
2. The asset database contains at least one asset with associated usage records.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The usage records are displayed to the Administrator.   
2. An audit log is generated to record the viewing activity.   
3. The Administrator is informed that the usage records have been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Usage Records interface in the system.   
2. The Administrator selects an asset from the list of available assets.   
3. The system retrieves the usage records associated with the selected asset from the database.   
4. The system displays the usage records to the Administrator, including details such as start and end times, activity type, and user information.   
5. The system creates an audit log entry to document the viewing action.   
6. The system displays a confirmation message to the Administrator that the usage records have been successfully retrieved.   
  
Alternative Flow:   
1. If the selected asset does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid asset.   
2. If the asset has no associated usage records, the system displays a message indicating that no records are available.   
3. If the system cannot retrieve the usage records due to database issues, it displays an error message and logs the failure in the audit log.   
4. If the Administrator does not have permission to view the usage records, the system denies access and displays an appropriate error message.  
  
Use Case Name: Generate Usage Report   
Use Case ID: UC-11   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to generate usage reports.   
2. The asset database contains at least one asset with associated usage records.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The usage report is successfully generated and displayed to the Administrator.   
2. The system stores the report generation activity in the audit log.   
3. The Administrator is informed that the report has been successfully generated.   
  
Main Flow:   
1. The Administrator navigates to the Usage Report Generation interface in the system.   
2. The Administrator selects a date range or specific time period for the report.   
3. The Administrator chooses the type of report (e.g., by user, by department, by asset category).   
4. The Administrator clicks the "Generate Report" button.   
5. The system retrieves the relevant usage records from the database based on the specified criteria.   
6. The system compiles the usage data into a formatted report.   
7. The system displays the report to the Administrator for viewing or downloading.   
8. The system creates an audit log entry to record the report generation action.   
9. The system displays a confirmation message to the Administrator that the report has been generated.   
  
Alternative Flow:   
1. If no usage records match the specified criteria, the system displays a message indicating that no data is available for the report.   
2. If the system is unable to retrieve the usage records due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the selected time range is invalid (e.g., end date before start date), the system displays an error message and prompts the Administrator to correct the date input.   
4. If the Administrator does not have permission to generate the report, the system denies access and displays an appropriate error message.  
  
Use Case Name: View Audit Logs   
Use Case ID: UC-12   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view audit logs.   
2. The audit log database contains at least one log entry.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The audit logs are displayed to the Administrator.   
2. An audit log is generated to record the viewing activity.   
3. The Administrator is informed that the logs have been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Audit Log interface in the system.   
2. The Administrator selects the criteria for viewing audit logs (e.g., date range, user, or action type).   
3. The system retrieves the relevant audit logs from the database based on the selected criteria.   
4. The system displays the audit logs to the Administrator, including details such as timestamp, user, action performed, and affected entity.   
5. The system creates an audit log entry to document the viewing action.   
6. The system displays a confirmation message to the Administrator that the audit logs have been successfully retrieved.   
  
Alternative Flow:   
1. If the selected criteria result in no matching audit logs, the system displays a message indicating that no logs are available for the given conditions.   
2. If the system cannot retrieve the audit logs due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the date range is invalid (e.g., end date before start date), the system displays an error message and prompts the Administrator to correct the input.   
4. If the Administrator does not have permission to view audit logs, the system denies access and displays an appropriate error message.  
  
Use Case Name: Export Audit Logs   
Use Case ID: UC-13   
Actors: Administrator, System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to export audit logs.   
2. The audit log database contains at least one log entry.   
3. The system is operational and accessible to the Administrator.   
  
Postconditions:   
1. The audit logs are successfully exported in the requested format (e.g., CSV, PDF).   
2. An audit log is generated to record the export activity.   
3. The Administrator is informed that the export has been completed.   
  
Main Flow:   
1. The Administrator navigates to the Audit Log Export interface in the system.   
2. The Administrator selects the criteria for exporting audit logs (e.g., date range, user, or action type).   
3. The Administrator chooses the file format for the export (e.g., CSV, PDF).   
4. The Administrator clicks the "Export" button to initiate the export.   
5. The system retrieves the relevant audit logs from the database based on the selected criteria.   
6. The system formats the audit logs into the selected file format.   
7. The system generates the export file and provides it for download.   
8. The system creates an audit log entry to record the export action.   
9. The system displays a confirmation message to the Administrator that the audit logs have been exported.   
  
Alternative Flow:   
1. If the selected criteria result in no matching audit logs, the system displays a message indicating that no logs are available for the given conditions.   
2. If the system cannot retrieve the audit logs due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the date range is invalid (e.g., end date before start date), the system displays an error message and prompts the Administrator to correct the input.   
4. If the selected file format is not supported, the system displays an error message and asks the Administrator to choose a valid format.   
5. If the Administrator does not have permission to export audit logs, the system denies access and displays an appropriate error message.  
  
Use Case Name: Send Email Notifications   
Use Case ID: UC-14   
Actors: System, Administrator, User   
Preconditions:   
1. The system is configured with a valid email notification service.   
2. The system has access to the email addresses of the Administrator and/or User.   
3. The system is operational and accessible.   
4. The event triggering the email notification has occurred (e.g., asset registration, allocation, return, etc.).   
  
Postconditions:   
1. An email notification is successfully sent to the specified recipient(s).   
2. The system logs the email notification attempt in the audit log.   
3. If the notification fails, the system logs the error and informs the relevant user.   
  
Main Flow:   
1. The system detects an event that requires sending an email notification (e.g., asset allocation confirmation, user registration success).   
2. The system retrieves the email address of the recipient(s) from the database.   
3. The system generates an email notification with the relevant message and details of the event.   
4. The system sends the email notification through the configured email service.   
5. The system logs the successful email notification in the audit log.   
6. The system displays a confirmation message to the Administrator if applicable.   
  
Alternative Flow:   
1. If the system is unable to retrieve the recipient's email address, it logs the failure and displays an error message to the Administrator.   
2. If the email notification fails to send, the system logs the error in the audit log and displays a message to the Administrator indicating the failure.   
3. If the email notification service is unavailable, the system logs the issue and displays a message to the Administrator that the notification could not be sent.   
4. If the system does not have permission to send email notifications, it denies the action and logs the event in the audit log.  
  
Use Case Name: Configure Email Settings   
Use Case ID: UC-15   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to configure email settings.   
2. The system is operational and accessible to the Administrator.   
3. The email notification service is available for configuration.   
  
Postconditions:   
1. The email settings are successfully configured and updated in the system.   
2. The system stores the updated email configuration in the database.   
3. An audit log is generated to record the configuration activity.   
4. The Administrator is notified of the successful configuration.   
  
Main Flow:   
1. The Administrator navigates to the Email Configuration interface in the system.   
2. The Administrator selects the option to configure email settings.   
3. The Administrator enters or modifies the email server details, such as SMTP host, port, authentication credentials, and sender email address.   
4. The Administrator submits the configuration request.   
5. The system validates the input data and checks for any missing or invalid fields.   
6. The system updates the email settings in the database.   
7. The system creates an audit log entry to document the configuration action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends a test email notification to confirm the settings are functional.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the configuration request.   
3. If the test email notification fails to send, the system logs the error and displays a message to the Administrator, prompting them to verify the email server details.   
4. If the system fails to update the email settings in the database, it displays an error message and prevents the change.   
5. If the email notification service is unavailable during testing, the system logs the error and informs the Administrator that the settings could not be verified.  
  
Use Case Name: Manage Permission Assignment   
Use Case ID: UC-16   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage permission assignments.   
2. The system contains valid user records and permission definitions.   
3. The system is operational and accessible to the Administrator.   
4. The Permission Assignment database is initialized and ready for updates.   
  
Postconditions:   
1. The permission assignments are successfully managed (assigned, revoked, or modified).   
2. The updated permission assignment information is stored in the database.   
3. An audit log is generated to record the permission assignment activity.   
4. The Administrator is notified of the successful management of permission assignments.   
  
Main Flow:   
1. The Administrator navigates to the Permission Assignment Management interface in the system.   
2. The Administrator selects a user from the list of valid user records.   
3. The system displays the current permission assignments for the selected user.   
4. The Administrator chooses to either assign new permissions, revoke existing ones, or modify the current permission settings.   
5. The Administrator enters or selects the relevant permissions and confirms the action.   
6. The system validates the selected user and permissions to ensure they are valid and consistent.   
7. The system updates the permission assignment records in the database to reflect the changes.   
8. The system creates an audit log entry to document the permission assignment management action.   
9. The system displays a confirmation message to the Administrator indicating the changes have been applied.   
10. The system sends an email notification to the Administrator confirming the permission assignment has been updated.   
  
Alternative Flow:   
1. If the selected user does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid user.   
2. If the selected permissions are not valid or do not exist, the system displays an error message and asks the Administrator to choose valid permissions.   
3. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the permission assignment request.   
5. If the system fails to update the permission assignment records, it displays an error message and prevents the changes from being applied.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the permission assignment changes were still processed.  
  
Use Case Name: Manage Asset Inventory   
Use Case ID: UC-17   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage the asset inventory.   
2. The asset database contains at least one registered asset.   
3. The system is operational and accessible to the Administrator.   
4. The inventory management interface is available for interaction.   
  
Postconditions:   
1. The asset inventory is successfully updated based on the Administrator's actions (e.g., adding, removing, or modifying inventory items).   
2. The updated inventory information is stored in the database.   
3. An audit log is generated to record the inventory management activity.   
4. The Administrator is notified of the successful inventory update.   
  
Main Flow:   
1. The Administrator navigates to the Asset Inventory Management interface in the system.   
2. The Administrator selects an action to perform (e.g., add, remove, or modify an asset in the inventory).   
3. The Administrator enters or modifies the relevant asset information (e.g., quantity, status, or location).   
4. The Administrator submits the inventory management request.   
5. The system validates the input data and checks for any missing or invalid fields.   
6. The system updates the asset inventory records in the database to reflect the changes.   
7. The system creates an audit log entry to document the inventory management action.   
8. The system displays a confirmation message to the Administrator.   
9. The system sends an email notification to the Administrator confirming the asset inventory has been updated.   
  
Alternative Flow:   
1. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
2. The Administrator corrects the input data and resubmits the inventory management request.   
3. If the selected asset does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid asset.   
4. If the system fails to update the inventory records, it displays an error message and prevents the change.   
5. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the inventory update was still processed.  
  
Use Case Name: Manage Asset Status   
Use Case ID: UC-18   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset statuses.   
2. The asset database contains at least one registered asset.   
3. The system is operational and accessible to the Administrator.   
4. The asset status definitions (e.g., Active, Inactive, Maintenance, Disposed) are already configured in the system.   
  
Postconditions:   
1. The asset status is successfully updated in the system.   
2. The updated asset status is stored in the database.   
3. An audit log is generated to record the status change activity.   
4. The Administrator is notified of the successful status update.   
  
Main Flow:   
1. The Administrator navigates to the Asset Status Management interface in the system.   
2. The Administrator selects an asset from the list of registered assets.   
3. The system displays the current status and other relevant details of the selected asset.   
4. The Administrator chooses a new status from the predefined status options (e.g., Active, Inactive, Maintenance, Disposed).   
5. The Administrator enters any relevant notes or comments to document the reason for the status change.   
6. The Administrator submits the status update request.   
7. The system validates the selected asset and the new status to ensure the update is permissible (e.g., an asset in use cannot be marked as Disposed).   
8. The system updates the asset record in the database with the new status.   
9. The system creates an audit log entry to document the status change action.   
10. The system displays a confirmation message to the Administrator.   
11. The system sends an email notification to the Administrator confirming the asset status has been updated.   
  
Alternative Flow:   
1. If the selected asset is not valid or does not exist in the database, the system displays an error message and prompts the Administrator to select a valid asset.   
2. If the new status is not a valid option for the selected asset (e.g., attempting to mark an asset as Disposed while it is still allocated), the system displays an error message and prevents the update.   
3. If the input data is incomplete or invalid, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the status update request.   
5. If the system fails to update the asset status in the database, it displays an error message and prevents the change.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the asset status update was still processed.  
  
Use Case Name: Manage Usage Record   
Use Case ID: UC-19   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage usage records.   
2. The asset database contains at least one asset with associated usage records.   
3. The system is operational and accessible to the Administrator.   
4. The Usage Record database is initialized and ready for updates.   
  
Postconditions:   
1. The selected usage record is successfully managed (updated or deleted) in the system.   
2. The updated or deleted usage record information is stored or removed from the database.   
3. An audit log is generated to record the management activity.   
4. The Administrator is notified of the successful management of the usage record.   
  
Main Flow:   
1. The Administrator navigates to the Usage Record Management interface in the system.   
2. The Administrator selects a usage record from the list of existing records.   
3. The system displays the details of the selected usage record, including start and end times, activity type, and associated user and asset information.   
4. The Administrator chooses to either update or delete the usage record.   
5. If updating, the Administrator modifies the necessary fields (e.g., start/end time, activity type, or notes).   
6. If deleting, the Administrator confirms the deletion of the selected usage record.   
7. The Administrator submits the management request.   
8. The system validates the selected usage record and ensures it is associated with a valid asset and user.   
9. If updating, the system updates the usage record in the database.   
10. If deleting, the system removes the usage record from the database.   
11. The system creates an audit log entry to document the management action.   
12. The system displays a confirmation message to the Administrator indicating the changes have been applied.   
13. The system sends an email notification to the Administrator confirming the usage record has been updated or deleted.   
  
Alternative Flow:   
1. If the selected usage record does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid record.   
2. If the usage record is associated with an asset that is currently in use or under specific constraints, the system displays an error message and prevents the deletion or modification.   
3. If the input data is incomplete or invalid during an update, the system displays an error message indicating the missing or incorrect fields.   
4. The Administrator corrects the input data and resubmits the management request.   
5. If the system fails to update or delete the usage record in the database, it displays an error message and prevents the change.   
6. If the email notification fails to send, the system logs the error and informs the Administrator that the notification could not be sent, but the usage record management was still processed.  
  
Use Case Name: View Asset Status   
Use Case ID: UC-20   
Actors: Administrator, System   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view asset statuses.   
2. The asset database contains at least one registered asset.   
3. The system is operational and accessible to the Administrator.   
4. The asset status definitions (e.g., Active, Inactive, Maintenance, Disposed, Allocated) are already configured in the system.   
  
Postconditions:   
1. The asset status is displayed to the Administrator.   
2. An audit log is generated to record the viewing activity.   
3. The Administrator is informed that the asset status has been successfully retrieved.   
  
Main Flow:   
1. The Administrator navigates to the Asset Status interface in the system.   
2. The Administrator selects an asset from the list of registered assets.   
3. The system retrieves the current status and related information (e.g., allocation status, maintenance history, or disposal status) of the selected asset from the database.   
4. The system displays the asset status details to the Administrator, including the timestamp of the last status change and any associated notes.   
5. The system creates an audit log entry to record the viewing action.   
6. The system displays a confirmation message to the Administrator that the asset status has been successfully retrieved.   
  
Alternative Flow:   
1. If the selected asset does not exist or is invalid, the system displays an error message and prompts the Administrator to select a valid asset.   
2. If the system cannot retrieve the asset status due to database issues, it displays an error message and logs the failure in the audit log.   
3. If the Administrator does not have permission to view the asset status, the system denies access and displays an appropriate error message.   
4. If the system fails to create an audit log entry, it logs the failure internally and informs the Administrator that the asset status has been retrieved but the activity could not be recorded.