项目文档

# Functional Requirement

# 1. Functional Requirements   
  
## 1.1 Asset Registration Function   
\*\*Function ID\*\*: FR-01   
\*\*Description\*\*: The system allows an Administrator to register new assets into the database. The registration includes essential attributes such as name, description, category, status, purchase date, and location.   
\*\*Input\*\*: Asset name, description, category, status, purchase date, and location provided by the Administrator.   
\*\*Output\*\*: A new asset record stored in the database with the specified attributes.   
  
## 1.2 Asset Modification Function   
\*\*Function ID\*\*: FR-02   
\*\*Description\*\*: The system allows an Administrator to modify the information of an existing asset in the database. Only valid and authenticated administrators can perform this action.   
\*\*Input\*\*: Selected asset record and updated fields (e.g., name, location, status) provided by the Administrator.   
\*\*Output\*\*: Updated asset record stored in the database.   
  
## 1.3 Asset Deletion Function   
\*\*Function ID\*\*: FR-03   
\*\*Description\*\*: The system allows an Administrator to delete an asset from the database after confirmation.   
\*\*Input\*\*: Selected asset record and confirmation from the Administrator.   
\*\*Output\*\*: Deleted asset record from the database and a confirmation message to the Administrator.   
  
## 1.4 Asset Lifecycle Management Function   
\*\*Function ID\*\*: FR-04   
\*\*Description\*\*: The system allows an Administrator to update the lifecycle stage of an asset (e.g., from "In Use" to "Retired") and logs the change.   
\*\*Input\*\*: Selected asset and new lifecycle stage provided by the Administrator.   
\*\*Output\*\*: Updated lifecycle stage stored in the database and a confirmation message to the Administrator.   
  
## 1.5 Asset Usage Tracking Function   
\*\*Function ID\*\*: FR-05   
\*\*Description\*\*: The system records and tracks the usage of an asset by a user, including the usage date, duration, and purpose.   
\*\*Input\*\*: Selected asset, associated user, usage date, duration, and purpose provided by the Administrator.   
\*\*Output\*\*: A new asset usage record stored in the database.   
  
## 1.6 Asset Allocation Function   
\*\*Function ID\*\*: FR-06   
\*\*Description\*\*: The system enables an Administrator to allocate an asset to a user, recording the allocation date, expected return date, and user association.   
\*\*Input\*\*: Selected asset, associated user, and allocation parameters (e.g., allocation date, expected return date) provided by the Administrator.   
\*\*Output\*\*: A new asset allocation record stored in the database and an updated asset status.   
  
## 1.7 Asset Release Function   
\*\*Function ID\*\*: FR-07   
\*\*Description\*\*: The system allows an Administrator to release an asset from a user, recording the release date and reason for release.   
\*\*Input\*\*: Selected asset, associated user, and release parameters (e.g., release date, reason) provided by the Administrator.   
\*\*Output\*\*: A new asset release record stored in the database and an updated asset status.   
  
## 1.8 Asset Status Change Tracking Function   
\*\*Function ID\*\*: FR-08   
\*\*Description\*\*: The system logs changes in asset status (e.g., from "Available" to "In Use") including the timestamp and previous status.   
\*\*Input\*\*: Selected asset, previous status, new status, and timestamp (if provided) from the system.   
\*\*Output\*\*: A new asset status change record stored in the database.   
  
## 1.9 Asset Availability Monitoring Function   
\*\*Function ID\*\*: FR-09   
\*\*Description\*\*: The system monitors the availability status of assets, recording the check date and any notes related to the status.   
\*\*Input\*\*: Asset to be monitored and system-generated availability status.   
\*\*Output\*\*: A new asset availability monitor record stored in the database.   
  
## 1.10 Asset Allocation Record Management Function   
\*\*Function ID\*\*: FR-10   
\*\*Description\*\*: The system allows an Administrator to modify or update an existing asset allocation record, such as changing the return date or allocation status.   
\*\*Input\*\*: Selected asset allocation record and updated parameters provided by the Administrator.   
\*\*Output\*\*: Updated asset allocation record stored in the database.   
  
## 1.11 Asset Release Record Management Function   
\*\*Function ID\*\*: FR-11   
\*\*Description\*\*: The system allows an Administrator to modify or update an existing asset release record, such as changing the release date or reason.   
\*\*Input\*\*: Selected asset release record and updated parameters provided by the Administrator.   
\*\*Output\*\*: Updated asset release record stored in the database.   
  
## 1.12 Asset Status Log Management Function   
\*\*Function ID\*\*: FR-12   
\*\*Description\*\*: The system allows an Administrator to update or modify an asset status log entry, including timestamps and notes about the change.   
\*\*Input\*\*: Selected asset status log entry and updated parameters provided by the Administrator.   
\*\*Output\*\*: Updated asset status log entry stored in the database.   
  
## 1.13 Asset Notification Management Function   
\*\*Function ID\*\*: FR-13   
\*\*Description\*\*: The system allows an Administrator to send notifications related to assets to users via the configured email system and logs the notification event.   
\*\*Input\*\*: Selected asset, recipient(s), message content, and send request from the Administrator.   
\*\*Output\*\*: A new notification record stored in the database and the notification message sent via the email system.   
  
## 1.14 Asset History Audit Function   
\*\*Function ID\*\*: FR-14   
\*\*Description\*\*: The system enables an Administrator to audit the history of an asset, retrieving all lifecycle and status changes for review.   
\*\*Input\*\*: Selected asset from the asset list.   
\*\*Output\*\*: Asset history data retrieved from the database and displayed to the Administrator.   
  
## 1.15 Asset Permission Assignment Function   
\*\*Function ID\*\*: FR-15   
\*\*Description\*\*: The system allows an Administrator to assign specific permissions (e.g., view, edit, manage) to users or roles for a given asset.   
\*\*Input\*\*: Selected asset, user or role, and permission level provided by the Administrator.   
\*\*Output\*\*: A new user permission assignment record stored in the database.   
  
## 1.16 Asset Permission Update Function   
\*\*Function ID\*\*: FR-16   
\*\*Description\*\*: The system allows an Administrator to update existing asset permissions for users or roles.   
\*\*Input\*\*: Selected asset and updated permission settings provided by the Administrator.   
\*\*Output\*\*: Updated user permission assignment record stored in the database.   
  
## 1.17 Asset Permission Revocation Function   
\*\*Function ID\*\*: FR-17   
\*\*Description\*\*: The system allows an Administrator to revoke permissions for users or roles on a specific asset.   
\*\*Input\*\*: Selected asset and user or role to revoke permissions from, provided by the Administrator.   
\*\*Output\*\*: Updated or removed user permission assignment record stored in the database.   
  
## 1.18 User Permission Assignment Management Function   
\*\*Function ID\*\*: FR-18   
\*\*Description\*\*: The system allows an Administrator to modify or update an existing user permission assignment record, such as changing the assigned permission or status.   
\*\*Input\*\*: Selected user permission assignment record and updated parameters provided by the Administrator.   
\*\*Output\*\*: Updated user permission assignment record stored in the database.   
  
## 1.19 Asset Usage Analysis Report Generation Function   
\*\*Function ID\*\*: FR-19   
\*\*Description\*\*: The system allows an Administrator to generate a report analyzing asset usage, including filters such as time range, asset category, and location.   
\*\*Input\*\*: Report parameters (e.g., time range, asset category) provided by the Administrator.   
\*\*Output\*\*: An asset usage analysis report displayed or exported to the Administrator.   
  
## 1.20 Asset Usage Data Analysis Function   
\*\*Function ID\*\*: FR-20   
\*\*Description\*\*: The system allows an Administrator to analyze asset usage data based on selected criteria (e.g., time range, usage frequency).   
\*\*Input\*\*: Analysis criteria provided by the Administrator.   
\*\*Output\*\*: Analysis results (e.g., usage trends, underutilized assets) displayed to the Administrator.   
  
## 1.21 External Database Integration Function   
\*\*Function ID\*\*: FR-21   
\*\*Description\*\*: The system allows an Administrator to configure and integrate with an external database for asset data exchange.   
\*\*Input\*\*: External database configuration details (e.g., hostname, port, username, password) provided by the Administrator.   
\*\*Output\*\*: Integration configuration stored in the internal database and confirmation message to the Administrator.   
  
## 1.22 Asset Search Function   
\*\*Function ID\*\*: FR-22   
\*\*Description\*\*: The system allows an Administrator to search for asset records based on filters such as asset name, category, status, and location.   
\*\*Input\*\*: Search criteria (e.g., name, status) provided by the Administrator.   
\*\*Output\*\*: Matching asset records retrieved from the database and displayed to the Administrator.

# External Description

# 2. External Interfaces   
  
This chapter outlines the external interfaces required for the system to function effectively. These interfaces include user interfaces, hardware interfaces, software interfaces, and communication interfaces. Each interface is described in terms of its role, method of interaction, and any relevant input/output requirements.   
  
## 2.1 User Interface Output   
  
The system interacts with the \*\*Administrator\*\* through a \*\*web-based user interface\*\* that supports asset management tasks. The interface provides a structured and intuitive way for Administrators to perform operations such as registration, modification, deletion, and auditing of assets, as well as managing permissions and generating reports.   
  
- \*\*Role\*\*: The user interface is the primary means of interaction for Administrators to manage asset data and perform system operations.   
- \*\*Interaction Method\*\*: Web-based graphical user interface (GUI) with forms, tables, and buttons for input and output.   
 - \*\*Input\*\*: Asset attributes (name, description, category, status, purchase date, location), user selections, confirmation actions, and report filters.   
 - \*\*Output\*\*: Screen displays of asset records, confirmation messages, status updates, asset history data, and usage analysis reports.   
  
## 2.2 Hardware Interface Output   
  
The system does not directly interact with any external hardware devices. All data collection and processing are performed internally via the user interface and software components. Therefore, no specific hardware interfaces are defined in this system.   
  
- \*\*Role\*\*: None   
- \*\*Interaction Method\*\*: N/A   
  
## 2.3 Software Interface Output   
  
The system interacts with an \*\*internal database\*\* to store and retrieve asset records, status changes, allocation records, release records, and permission assignments. Additionally, it integrates with an \*\*external database\*\* for data exchange and has a \*\*configured email system\*\* for sending notifications.   
  
### 2.3.1 Internal Database   
- \*\*Role\*\*: The internal database is used to store all asset-related data, including asset records, status logs, allocation and release records, and user permissions.   
- \*\*Interaction Method\*\*:   
 - \*\*Input\*\*: Data from the user interface, such as new asset records, updated asset information, allocation and release records, and permission assignments.   
 - \*\*Output\*\*: Storage of records and retrieval of data for display, analysis, or modification.   
  
### 2.3.2 External Database   
- \*\*Role\*\*: The external database is used for asset data exchange and integration with other systems.   
- \*\*Interaction Method\*\*:   
 - \*\*Input\*\*: Configuration details (hostname, port, username, password) provided by the Administrator to establish the connection.   
 - \*\*Output\*\*: Confirmation of integration and data exchange between the internal and external databases.   
  
### 2.3.3 Email System   
- \*\*Role\*\*: The email system is used to send notifications to users regarding asset-related events.   
- \*\*Interaction Method\*\*:   
 - \*\*Input\*\*: Notification content, recipient information, and send request from the system.   
 - \*\*Output\*\*: Notification message sent via the configured email system.   
  
## 2.4 Communication Interface Output   
  
The system supports communication via \*\*email\*\* to send notifications and updates to users.   
  
- \*\*Role\*\*: The communication interface is used to inform users about asset allocation, release, status changes, and other relevant updates.   
- \*\*Interaction Method\*\*:   
 - \*\*Input\*\*: Message content, recipient details, and send request.   
 - \*\*Output\*\*: Notification message sent to the specified email address.   
  
---  
  
This section ensures that all external data sources and interfaces referenced in the functional requirements are clearly defined and categorized. It provides a clear understanding of how the system interacts with its environment, both in terms of data storage/retrieval and communication.

# Use Case

Use Case Name: Asset Lifecycle Management   
Use Case ID: UC-01   
Actors: Administrator, Email System, Database   
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset lifecycle information.   
2. The system is connected to the database where asset data is stored.   
3. The email system is configured and available for sending notifications.   
  
Postconditions:   
1. The asset lifecycle information is updated in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the email system.   
  
Main Flow:   
1. The Administrator opens the Asset Lifecycle Management module in the system.   
2. The system retrieves the current asset data from the database.   
3. The Administrator selects an asset to update its lifecycle status.   
4. The Administrator modifies the asset’s lifecycle status (e.g., from "In Use" to "Retired").   
5. The system validates the changes and updates the asset lifecycle data in the database.   
6. The system logs the update and sends a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset data from the database, it displays an error message to the Administrator.   
2. If the Administrator enters an invalid lifecycle status, the system prompts for a valid status.   
3. If the database update fails, the system logs the error and sends a notification to the Administrator via the email system.  
  
Use Case Name: Register Asset   
Use Case ID: UC-02   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to register new assets.   
2. The system is connected to the database for storing asset records.   
  
Postconditions:   
1. The new asset is successfully stored in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Register Asset" module in the system.   
2. The system displays an asset registration form.   
3. The Administrator fills in the required asset information (e.g., name, type, location, status).   
4. The Administrator submits the form.   
5. The system validates the input data.   
6. The system stores the new asset record in the database.   
7. The system logs the registration and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the Administrator leaves mandatory fields empty, the system prompts for their completion.   
2. If the input data is invalid (e.g., incorrect format), the system displays an error message and asks the Administrator to correct the input.   
3. If the database fails to store the new asset record, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Modify Asset Information   
Use Case ID: UC-03   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to modify asset information.   
2. The system is connected to the database where asset records are stored.   
  
Postconditions:   
1. The asset information is updated in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Modify Asset Information" module in the system.   
2. The system displays a list of existing assets retrieved from the database.   
3. The Administrator selects an asset to modify.   
4. The system shows the asset’s current information in an editable form.   
5. The Administrator updates the necessary fields (e.g., name, location, status).   
6. The Administrator submits the updated form.   
7. The system validates the input data.   
8. The system updates the asset record in the database.   
9. The system logs the modification and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset information, it displays an error message to the Administrator.   
2. If the Administrator leaves mandatory fields empty, the system prompts for their completion.   
3. If the input data is invalid (e.g., incorrect format), the system displays an error message and asks the Administrator to correct the input.   
4. If the database fails to update the asset record, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: View Asset Details   
Use Case ID: UC-04   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to view asset details.   
2. The system is connected to the database where asset records are stored.   
  
Postconditions:   
1. The asset details are displayed to the Administrator.   
2. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "View Asset Details" module in the system.   
2. The system displays a list of assets retrieved from the database.   
3. The Administrator selects an asset to view its detailed information.   
4. The system retrieves the selected asset’s data from the database.   
5. The system displays the asset details (e.g., name, type, status, usage history).   
  
Alternative Flow:   
1. If the system fails to retrieve the asset data, it displays an error message to the Administrator.   
2. If the selected asset does not exist, the system informs the Administrator and returns to the asset list.  
  
Use Case Name: Delete Asset   
Use Case ID: UC-05   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to delete assets.   
2. The system is connected to the database where asset records are stored.   
  
Postconditions:   
1. The selected asset is removed from the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Delete Asset" module in the system.   
2. The system displays a list of existing assets retrieved from the database.   
3. The Administrator selects an asset to delete.   
4. The system confirms the deletion with the Administrator.   
5. The Administrator confirms the deletion.   
6. The system deletes the selected asset record from the database.   
7. The system logs the deletion and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset list, it displays an error message to the Administrator.   
2. If the selected asset does not exist, the system informs the Administrator and returns to the asset list.   
3. If the Administrator cancels the deletion confirmation, the system returns to the asset list without making any changes.   
4. If the database fails to delete the asset record, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Generate Asset Usage Analysis Report   
Use Case ID: UC-06   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to generate asset usage analysis reports.   
2. The system is connected to the database where asset usage data is stored.   
3. The email system is configured and available for sending reports if required.   
  
Postconditions:   
1. The asset usage analysis report is generated and displayed or exported.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the report is sent to the Administrator via the email system.   
  
Main Flow:   
1. The Administrator navigates to the "Generate Asset Usage Analysis Report" module in the system.   
2. The system displays options for filtering the report (e.g., time range, asset type, location).   
3. The Administrator selects the desired filters and confirms the report parameters.   
4. The system retrieves the relevant asset usage data from the database.   
5. The system generates the analysis report based on the retrieved data.   
6. The system displays the report to the Administrator and offers options to export it (e.g., PDF, Excel).   
7. The system logs the report generation and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset usage data, it displays an error message to the Administrator.   
2. If the selected filters result in no data, the system informs the Administrator and offers to adjust the filters.   
3. If the report generation fails, the system logs the error and displays a notification to the Administrator.   
4. If the Administrator chooses to send the report via email, the system sends it using the email system.   
5. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Analyze Asset Usage Data   
Use Case ID: UC-07   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to analyze asset usage data.   
2. The system is connected to the database where asset usage records are stored.   
3. The email system is configured and available if the Administrator chooses to send the analysis results.   
  
Postconditions:   
1. The asset usage analysis is performed and results are displayed or exported.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the analysis results are sent to the Administrator via the email system.   
  
Main Flow:   
1. The Administrator navigates to the "Analyze Asset Usage Data" module in the system.   
2. The system displays options for selecting the analysis criteria (e.g., time range, asset category, usage frequency).   
3. The Administrator selects the criteria and confirms the analysis parameters.   
4. The system retrieves the relevant asset usage data from the database.   
5. The system processes the data and generates the analysis results (e.g., usage trends, underutilized assets).   
6. The system displays the analysis results to the Administrator and offers options to export the findings.   
7. The system logs the analysis and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset usage data, it displays an error message to the Administrator.   
2. If the selected criteria result in no data, the system informs the Administrator and offers to adjust the criteria.   
3. If the analysis processing fails, the system logs the error and displays a notification to the Administrator.   
4. If the Administrator chooses to send the analysis results via email, the system sends them using the email system.   
5. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Assign Asset Permissions   
Use Case ID: UC-08   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to assign or modify asset permissions.   
2. The system is connected to the database where asset and user permission data is stored.   
3. The email system is configured and available for sending notifications if required.   
  
Postconditions:   
1. The asset permissions are updated in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the system sends an email notification to the affected users regarding permission changes.   
  
Main Flow:   
1. The Administrator navigates to the "Assign Asset Permissions" module in the system.   
2. The system displays a list of assets and their current permissions retrieved from the database.   
3. The Administrator selects an asset and chooses the user(s) or role(s) to assign or modify permissions for.   
4. The Administrator specifies the level of access (e.g., view, edit, manage).   
5. The Administrator confirms the permission assignment.   
6. The system validates the input and updates the asset permissions in the database.   
7. The system logs the permission change and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset or user data, it displays an error message to the Administrator.   
2. If the Administrator selects invalid users or roles, the system prompts for valid selections.   
3. If the permission update fails in the database, the system logs the error and displays a notification to the Administrator.   
4. If the Administrator chooses to send an email notification, the system sends it using the email system.   
5. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Update Asset Permissions   
Use Case ID: UC-09   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to update asset permissions.   
2. The system is connected to the database where asset and user permission data is stored.   
3. The email system is configured and available for sending notifications if required.   
  
Postconditions:   
1. The asset permissions are updated in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the system sends an email notification to the affected users regarding the permission changes.   
  
Main Flow:   
1. The Administrator navigates to the "Update Asset Permissions" module in the system.   
2. The system displays a list of assets and their associated permissions retrieved from the database.   
3. The Administrator selects an asset to update its permissions.   
4. The system presents the current permissions settings for the selected asset.   
5. The Administrator modifies the permissions (e.g., grants or revokes access for specific users or roles).   
6. The Administrator confirms the changes.   
7. The system validates the input and updates the asset permissions in the database.   
8. The system logs the permission update and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset or permission data, it displays an error message to the Administrator.   
2. If the Administrator selects invalid users or roles, the system prompts for valid selections.   
3. If the permission update fails in the database, the system logs the error and displays a notification to the Administrator.   
4. If the Administrator chooses to notify users via email, the system sends the notification using the email system.   
5. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Revoke Asset Permissions   
Use Case ID: UC-10   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to revoke asset permissions.   
2. The system is connected to the database where asset and user permission data is stored.   
3. The email system is configured and available for sending notifications if required.   
  
Postconditions:   
1. The asset permissions are revoked and updated in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the system sends an email notification to the affected users regarding the revoked permissions.   
  
Main Flow:   
1. The Administrator navigates to the "Revoke Asset Permissions" module in the system.   
2. The system displays a list of assets and their current permissions retrieved from the database.   
3. The Administrator selects an asset and chooses the user(s) or role(s) to revoke permissions from.   
4. The Administrator confirms the permission revocation.   
5. The system validates the input and updates the database to remove the selected permissions.   
6. The system logs the revocation and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset or permission data, it displays an error message to the Administrator.   
2. If the Administrator selects invalid users or roles, the system prompts for valid selections.   
3. If the permission revocation fails in the database, the system logs the error and displays a notification to the Administrator.   
4. If the Administrator chooses to notify users via email, the system sends the notification using the email system.   
5. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Send Asset Notification via Email   
Use Case ID: UC-11   
Actors: Administrator, Email System, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to send asset notifications.   
2. The system is connected to the database where asset information is stored.   
3. The email system is configured and available for sending notifications.   
4. The asset for which a notification is to be sent exists in the database.   
  
Postconditions:   
1. The asset notification is sent to the designated recipient(s) via the email system.   
2. The notification details are logged in the system.   
3. The Administrator receives a confirmation message.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Send Asset Notification" module in the system.   
2. The system retrieves the list of assets from the database.   
3. The Administrator selects an asset and specifies the notification message and recipients.   
4. The Administrator confirms the notification details.   
5. The system validates the recipients and message content.   
6. The system sends the notification via the email system.   
7. The system logs the notification event and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset list, it displays an error message to the Administrator.   
2. If the selected asset does not exist, the system informs the Administrator and returns to the asset list.   
3. If the notification message is empty or invalid, the system prompts the Administrator to provide valid content.   
4. If the email system is unavailable, the system logs the error and displays a notification to the Administrator.   
5. If the notification fails to send, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Integrate with External Database   
Use Case ID: UC-12   
Actors: Administrator, External Database, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to configure external database integrations.   
2. The system has access to the internal database for storing integration configurations.   
3. The external database is accessible and compatible with the system's integration protocols.   
  
Postconditions:   
1. The external database is successfully integrated with the system.   
2. Integration details are stored in the internal database.   
3. The Administrator receives a confirmation message.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Integrate with External Database" module in the system.   
2. The system displays a configuration form for the external database (e.g., hostname, port, username, password).   
3. The Administrator fills in the required connection details for the external database.   
4. The Administrator confirms the configuration.   
5. The system validates the connection to the external database.   
6. The system saves the integration configuration in the internal database.   
7. The system logs the integration and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the Administrator leaves mandatory fields empty, the system prompts for their completion.   
2. If the connection to the external database fails, the system displays an error message and logs the failure.   
3. If the internal database fails to store the integration configuration, the system logs the error and notifies the Administrator.   
4. If the Administrator chooses to cancel the integration, the system discards the configuration and returns to the previous screen.  
  
Use Case Name: Audit Asset History   
Use Case ID: UC-13   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to audit asset history.   
2. The system is connected to the database where asset history records are stored.   
  
Postconditions:   
1. The asset history is retrieved and displayed to the Administrator.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Audit Asset History" module in the system.   
2. The system displays a list of assets retrieved from the database.   
3. The Administrator selects an asset to view its history.   
4. The system retrieves the asset's historical data (e.g., lifecycle changes, usage records, permission modifications).   
5. The system displays the audit history in a structured format to the Administrator.   
6. The system logs the audit request and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset list, it displays an error message to the Administrator.   
2. If the selected asset does not exist, the system informs the Administrator and returns to the asset list.   
3. If the system fails to retrieve the asset history, it displays an error message to the Administrator.   
4. If the Administrator cancels the audit request, the system returns to the previous screen without logging the action.  
  
Use Case Name: Track Asset Status Changes   
Use Case ID: UC-14   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to track asset status changes.   
2. The system is connected to the database where asset status data is stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset to be tracked exists in the database.   
  
Postconditions:   
1. The asset's status change is recorded in the database.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the system sends an email notification to relevant users about the status change.   
  
Main Flow:   
1. The Administrator navigates to the "Track Asset Status Changes" module in the system.   
2. The system displays a list of assets retrieved from the database.   
3. The Administrator selects an asset to monitor for status changes.   
4. The system retrieves the asset’s current status and history from the database.   
5. The Administrator sets up tracking parameters (e.g., alerts, frequency, users to notify).   
6. The system saves the tracking configuration in the database.   
7. The system logs the tracking setup and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset list, it displays an error message to the Administrator.   
2. If the selected asset does not exist, the system informs the Administrator and returns to the asset list.   
3. If the tracking parameters are invalid, the system prompts the Administrator for valid input.   
4. If the database fails to store the tracking configuration, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to send a notification via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Allocate Asset to User   
Use Case ID: UC-15   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to allocate assets to users.   
2. The system is connected to the database where asset and user data is stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset to be allocated exists in the database and is available for allocation.   
  
Postconditions:   
1. The asset is successfully allocated to the specified user and the status is updated in the database.   
2. The system logs the allocation event and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user about the allocation.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Allocate Asset to User" module in the system.   
2. The system displays a list of available assets retrieved from the database.   
3. The Administrator selects an asset to allocate.   
4. The system displays a list of users or roles to assign the asset to.   
5. The Administrator selects a user or role and confirms the allocation.   
6. The system updates the asset’s status and ownership in the database.   
7. The system logs the allocation and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset list, it displays an error message to the Administrator.   
2. If the selected asset is not available or does not exist, the system informs the Administrator and returns to the asset list.   
3. If the user or role selected is invalid, the system prompts the Administrator for a valid selection.   
4. If the database fails to update the asset allocation, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to notify the user via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Release Asset from User   
Use Case ID: UC-16   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to release assets from users.   
2. The system is connected to the database where asset and user allocation data is stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset to be released is currently allocated to a user and exists in the database.   
  
Postconditions:   
1. The asset is successfully released from the user and its status is updated in the database.   
2. The system logs the release event and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user about the release.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Release Asset from User" module in the system.   
2. The system displays a list of allocated assets retrieved from the database.   
3. The Administrator selects an asset to be released from a user.   
4. The system confirms the current allocation and displays the user to whom the asset is assigned.   
5. The Administrator confirms the release action.   
6. The system updates the asset’s status to "Available" and removes the user association in the database.   
7. The system logs the release and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the list of allocated assets, it displays an error message to the Administrator.   
2. If the selected asset is not allocated or does not exist, the system informs the Administrator and returns to the asset list.   
3. If the Administrator cancels the release confirmation, the system returns to the asset list without making any changes.   
4. If the database fails to update the asset status, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to notify the user via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Monitor Asset Availability   
Use Case ID: UC-17   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to monitor asset availability.   
2. The system is connected to the database where asset status and availability data are stored.   
3. The email system is configured and available for sending notifications if required.   
  
Postconditions:   
1. The asset availability status is checked and displayed to the Administrator.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
4. If configured, the system sends an email notification to the Administrator about assets that are unavailable or require attention.   
  
Main Flow:   
1. The Administrator navigates to the "Monitor Asset Availability" module in the system.   
2. The system retrieves the list of all assets and their current availability status from the database.   
3. The system displays the asset availability information in a structured format (e.g., status, last used date, allocated to).   
4. The Administrator reviews the availability status of assets and identifies any assets that are unavailable or need maintenance.   
5. The system logs the monitoring activity and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset availability data, it displays an error message to the Administrator.   
2. If no assets are found in the database, the system informs the Administrator and offers to return to the main menu.   
3. If the Administrator chooses to receive email notifications and the email system is unavailable, the system logs the error and displays a notification to the Administrator.   
4. If the system detects that an asset is unavailable and the Administrator is not configured to receive alerts, no action is taken beyond logging the status.  
  
Use Case Name: Search Asset Records   
Use Case ID: UC-18   
Actors: Administrator, Database   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to search for asset records.   
2. The system is connected to the database where asset records are stored.   
  
Postconditions:   
1. The search results are displayed to the Administrator.   
2. The Administrator receives a confirmation message.   
3. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Search Asset Records" module in the system.   
2. The system displays a search interface with options to filter by asset name, type, status, or location.   
3. The Administrator enters search criteria and confirms the search request.   
4. The system queries the database for matching asset records.   
5. The system displays the search results in a list format to the Administrator.   
6. The system logs the search activity and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve asset records, it displays an error message to the Administrator.   
2. If no assets match the search criteria, the system informs the Administrator and offers to adjust the search terms.   
3. If the search criteria are invalid, the system prompts the Administrator to provide valid input.   
4. If the Administrator cancels the search, the system returns to the previous screen without logging the action.  
  
Use Case Name: Manage Asset Allocation Record   
Use Case ID: UC-19   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset allocation records.   
2. The system is connected to the database where asset allocation records are stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset allocation record to be managed exists in the database.   
  
Postconditions:   
1. The asset allocation record is updated or modified in the database.   
2. The system logs the action and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user or relevant parties about the allocation change.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Asset Allocation Record" module in the system.   
2. The system displays a list of existing asset allocation records retrieved from the database.   
3. The Administrator selects an allocation record to modify or update.   
4. The system presents the current allocation details (e.g., asset ID, user ID, allocation date, return date, status).   
5. The Administrator modifies the necessary fields (e.g., updates the return date or changes the allocation status).   
6. The Administrator confirms the changes.   
7. The system validates the input and updates the asset allocation record in the database.   
8. The system logs the modification and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset allocation records, it displays an error message to the Administrator.   
2. If the selected allocation record does not exist, the system informs the Administrator and returns to the allocation record list.   
3. If the Administrator enters invalid data (e.g., incorrect date format or invalid status), the system prompts for valid input.   
4. If the database fails to update the allocation record, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to send a notification via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Manage Asset Release Record   
Use Case ID: UC-20   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset release records.   
2. The system is connected to the database where asset release records are stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset release record to be managed exists in the database.   
  
Postconditions:   
1. The asset release record is updated or modified in the database.   
2. The system logs the action and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user or relevant parties about the release change.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Asset Release Record" module in the system.   
2. The system displays a list of existing asset release records retrieved from the database.   
3. The Administrator selects a release record to modify or update.   
4. The system presents the current release details (e.g., asset ID, user ID, release date, reason for release, status).   
5. The Administrator modifies the necessary fields (e.g., updates the release date or changes the status to "Completed").   
6. The Administrator confirms the changes.   
7. The system validates the input and updates the asset release record in the database.   
8. The system logs the modification and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset release records, it displays an error message to the Administrator.   
2. If the selected release record does not exist, the system informs the Administrator and returns to the release record list.   
3. If the Administrator enters invalid data (e.g., incorrect date format or invalid status), the system prompts for valid input.   
4. If the database fails to update the release record, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to send a notification via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Manage Asset Status Log   
Use Case ID: UC-21   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage asset status logs.   
2. The system is connected to the database where asset status logs are stored.   
3. The email system is configured and available for sending notifications if required.   
4. The asset status log to be managed exists in the database.   
  
Postconditions:   
1. The asset status log is updated or modified in the database.   
2. The system logs the action and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user or relevant parties about the status log change.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Manage Asset Status Log" module in the system.   
2. The system displays a list of existing asset status logs retrieved from the database.   
3. The Administrator selects a status log entry to modify or update.   
4. The system presents the current status log details (e.g., asset ID, previous status, new status, timestamp, responsible user).   
5. The Administrator modifies the necessary fields (e.g., updates the timestamp or adds a comment to the log entry).   
6. The Administrator confirms the changes.   
7. The system validates the input and updates the asset status log record in the database.   
8. The system logs the modification and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the asset status logs, it displays an error message to the Administrator.   
2. If the selected status log entry does not exist, the system informs the Administrator and returns to the status log list.   
3. If the Administrator enters invalid data (e.g., incorrect timestamp format or invalid status), the system prompts for valid input.   
4. If the database fails to update the status log, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to send a notification via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.  
  
Use Case Name: Manage User Permission Assignment   
Use Case ID: UC-22   
Actors: Administrator, Database, Email System   
  
Preconditions:   
1. The Administrator is authenticated and has the necessary permissions to manage user permission assignments.   
2. The system is connected to the database where user and permission data is stored.   
3. The email system is configured and available for sending notifications if required.   
4. The user permission assignment to be managed exists in the database.   
  
Postconditions:   
1. The user permission assignment is updated or modified in the database.   
2. The system logs the action and provides a confirmation message to the Administrator.   
3. If configured, the system sends an email notification to the user or relevant parties about the permission assignment change.   
4. If an error occurs, the Administrator is notified via the system interface.   
  
Main Flow:   
1. The Administrator navigates to the "Manage User Permission Assignment" module in the system.   
2. The system displays a list of existing user permission assignments retrieved from the database.   
3. The Administrator selects a user permission assignment record to modify or update.   
4. The system presents the current permission assignment details (e.g., user ID, asset ID, assigned permissions, assignment date, status).   
5. The Administrator modifies the necessary fields (e.g., updates the assigned permissions or changes the status to "Revoked").   
6. The Administrator confirms the changes.   
7. The system validates the input and updates the user permission assignment record in the database.   
8. The system logs the modification and provides a confirmation message to the Administrator.   
  
Alternative Flow:   
1. If the system fails to retrieve the user permission assignments, it displays an error message to the Administrator.   
2. If the selected permission assignment does not exist, the system informs the Administrator and returns to the permission assignment list.   
3. If the Administrator enters invalid data (e.g., incorrect permission level or invalid date format), the system prompts for valid input.   
4. If the database fails to update the permission assignment, the system logs the error and notifies the Administrator.   
5. If the Administrator chooses to send a notification via email and the email system is unavailable, the system logs the error and displays a notification to the Administrator.