Logo

Description automatically generated

Software Design Document- Infrastructure

Software Design Document is a written document that provides a description of a software product in terms of architecture of software with various components with specified functionality.The design specification addresses different aspects of the design model and is completed as the designer refines his representation of the software. These design documents are written by software engineers/designers or project managers and further passed to the software development team to give them an overview of what needs to be built and how.

**Sustainable Outreach And Universal**

**Leadership (SOUL) Limited**

Contents

[A. Introduction 3](#_Toc15724)

[B. Literature review 3](#_Toc3816)

[C. Platform Overview 3](#_Toc20234)

[Platform requirements 3](#_Toc4625)

[i. Software Stack 3](#_Toc6951)

[ii. Hosting requirement 4](#_Toc13331)

[D. Architecture 4](#_Toc3250)

[E. Organization Structure 5](#_Toc6368)

[F. Design Overview 6](#_Toc29098)

[1. Infrastructure Management 6](#_Toc28925)

[1.1 Land 6](#_Toc4319)

[1.2 Buildings 9](#_Toc26331)

[1.3 Floor 12](#_Toc22129)

[1.4 Building Room 14](#_Toc15296)

[1.5 Building type Room 16](#_Toc5521)

[2. Infrastructure Maintenance 19](#_Toc14612)

[2.1 Asset 19](#_Toc4375)

[2.2 Asset Maintenance Team 25](#_Toc281)

[2.3 Asset Maintenance 27](#_Toc18913)

[2.4 Asset Maintenance Log 30](#_Toc25168)

[2.5 Asset Repair 32](#_Toc18716)

[3. Infrastructure Project Management 36](#_Toc11036)

[3.1 Project 36](#_Toc3557)

[3.2 Task 41](#_Toc10078)

[3.3 Project Template 43](#_Toc16613)

[3.4 Project Type 45](#_Toc38)

# **Introduction**

Campus management project for WSC aims at providing an efficient, secure, and user-friendly platform to manage and automate the various processes. It is an integrated system that provides all the necessary tools to manage the various aspects of running a University. This web-based solution enables administrators and other concerned staff to easily track and manage activities related to Infrastructure Management & Maintenance and Project Plan Monitoring.

Scope:

1. Land and Building Database
2. Floor Asset Management
3. Infrastructure Asset Maintenance
4. Project Estimation
5. Ongoing/Completed Project Monitoring

# **Literature review**

The following section first explains the evolution of ERP system and some of its characteristics. It also explores the various modules available in an ERP software. Knowledge-intensive nature of ERP systems and ERP system adoption is also analyzed. Finally it looks into critical success factors for ERP system implementation as well as benefits of ERP implementation.

# **Platform Overview**

**Platform requirements**

1. **Software Stack**

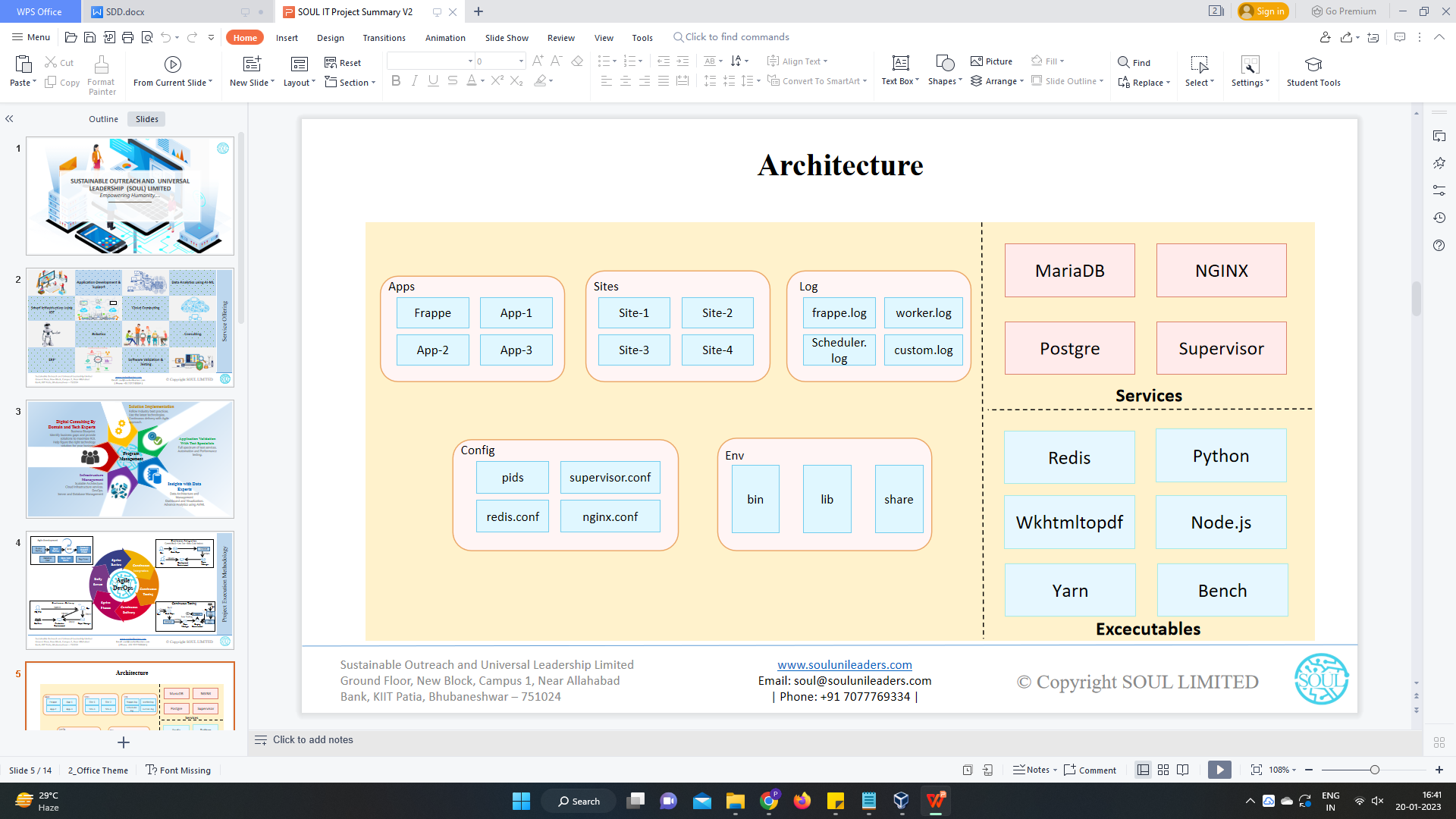
Our application uses Frappe framework as the major software stack which is an open-source full stack web framework written in Python and JavaScript that is used to develop and maintain web applications. It is designed to be simple, easy to use, and fast. It is built on top of the popular Django web framework. Frappe provides a powerful set of tools for building web applications with a focus on scalability and extensibility. It provides a powerful ORM, a powerful templating system, and a comprehensive set of APIs for building web applications. Frappe is also designed for ease of use, with a focus on reducing development time and building an intuitive admin interface. It also features a seamless integration with other open-source projects such as Redis, MongoDB, and MySQL, as well as a host of APIs and other tools.

1. **Hosting requirement**

Our application requires a **Linux**-based hosting environment. It requires at least **1GB** of **RAM**, **2GB** of **disk space**, and preferably an **Ubuntu 16.04** or **higher** **operating system**. It also requires a **web server** such as **Apache** or **Nginx**, **MariaDB 10.1** or **higher**, and **Python 2.7.x** or **higher**. Additionally, it requires an SMTP server to send emails and an SSL certificate to enable HTTPS.

# **Architecture**

Our application is based on a web-based, modular, service-oriented architecture (SOA), which allows for the development of custom applications and services that can be integrated with the core Our application platform. It is built on a combination of JavaScript, Python, and MySQL. The architecture of Our application consists of multiple layers, each with its own set of functions and services. At the highest level are the user interface, user experience, and application logic components. These components provide the user with an interface to the Our application system and the ability to interact with the underlying data and services. Below the user interface layer is the data layer, which is responsible for managing the storage and retrieval of data and services. This layer is powered by a combination of MySQL and PostgreSQL databases. The data layer is also responsible for providing secure access to the data and services. The application layer is responsible for providing access to the data and services, as well as providing the business logic that drives the system. The application layer is powered by a combination of Python and JavaScript, and is responsible for providing the core capabilities of the system, such as authentication, authorization, reporting, and analytics. Finally, the service layer is the layer that connects the application layer to the user interface layer. It provides the ability to integrate the Our application system with external systems and services, such as third-party APIs and web services. The service layer is powered by a combination of web services and web sockets. Following is a diagram showing a representation of the architecture of our application:



**Fig**: Architecture of Our Application

# **Organization Structure**

**Input required form WSC**

# Design Overview

# Infrastructure Management

## Land

**Overview**

The Land screen will hold all the data of any land, like owner details, plot number, land complete address, and relevant documents as attachments.

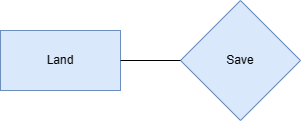
**Description of Problem**

Create a page containing all information about the land, including owner details,plot number, address, land valuation, date of planned development, a table for attaching documents, etc as per the form Elaboration.

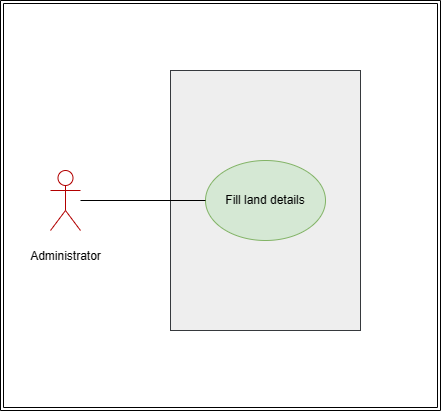
**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Flow Chart**

-

**Use Case Diagram**



**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Name | Text Field |  | Y |  |  |
| 2 | Address | Text Field |  | Y |  |  |
| 3 | Phone | Text Field |  | Y |  |  |
| 4 | Email | Text Field |  | Y |  |  |
| 5 | Property on lease | Select | Yes  No | Y |  |  |
| 6 | Lease | Attach |  |  |  |  |
| 7 | Plot number | Data |  | Y |  |  |
| 8 | Land Complete Address | Small Text Field |  | Y |  |  |
| 9 | Pin code | Text Field |  |  |  |  |
| 10 | Land Size | Text Field |  | Y |  |  |
| 11 | Land Valuation | Currency |  |  |  |  |
| 12 | Start Date | Date |  | Y |  |  |
| 13 | End Date | Date |  | Y |  |  |
| 14 | Land Use | Drop down | Academic  Residential  Others | Y |  |  |
| 15 | Land Use, If others | Text Field |  | Display Depends on Land use drop down selection(others) |  |  |
| 16 | Date of Planned Development | Date |  |  |  |  |
| **17** | **Land Documents** | Table |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Land Documents** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Document Name | Text Field | User Input |  |  |  |
| 2 | Document | Attach Button |  |  |  |  |

**Business Logic**

**land.js:**

To validate that the end date is not before the start date in the UI

**land.py:**

1. def date: To validate if the start date is not after the end date, if it is then throw an error
2. def pincode: Validation for pin code and length, If less than or greater than 6 throw an error.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Building Administrator | | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Floor Supervisor | | No | Yes | No | No | No | NA | NA | NA |
| 3 | Maintenance Department | | No | Yes | No | No | No | NA | NA | NA |
| 4 | Employee | | No | No | No | No | No | No | No | No |

**Path of the Screen**

Home > Infrastructure> Land

## Buildings

**Overview**

All the building information shall be stored here. The buildings can be residential, academic or both. The actions on this screen shall be performed by the administrator only.

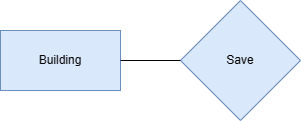
**Description of Problem**

Create a page on which the user can fill in and store the building details. E.g: Building name, building address, number of floors, type of building, etc.

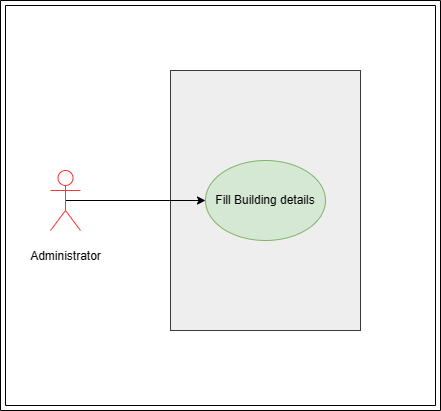
**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Flow Chart**

****

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Land Plot number | Link Field | Fetched from Land master screen |  |  |  |
| 2 | Land Address | Text Field | Auto fetch based on Land Plot Number |  |  |  |
| 3 | Building Name | Text Field | User Input | Y |  |  |
| 4 | Building Address | Text Field |  | Y |  |  |
| 5 | Post Office | Text Field |  | Y |  |  |
| 6 | District | Link Field | Districts | Y |  |  |
| 7 | State | Text Field |  | Y |  |  |
| 8 | Pin Code | Text Field |  | Y |  |  |
| 9 | Building on Lease | Drop down | Yes  No | Y |  |  |
| 10 | Lease | Attach | Display Depends on “Building on Lease” | Y |  |  |
| 11 | Building Type | Drop down | Academic  Administrative  Others | Y |  |  |
| 12 | Building Type, if any other | Text Field |  | Display Depends on “Building Type” drop down selection(others) |  |  |
| 13 | Total Rooms | Integer Field |  |  |  |  |
| 14 | Total Floors | Integer Field |  | Y |  |  |
| 15 | Facing | Drop down | North  South  East  West | Y |  |  |
| 16 | Campus | Text Field |  |  |  |  |
| 17 | Start date | Date picker |  |  |  |  |
| 18 | End date | Date picker |  |  |  |  |
| **19** | **Building Documents** | Attach |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Building Documents** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Document Name | Text Field |  |  |  |  |
| 2 | Document | Attach Button |  |  |  |  |

**Business Logic**

**buildings.js:**

1. To set filter on the Plot number field, so as to fetch only those buildings which are between the start and end date of the land with respect to today’s date.
2. To validate end date is not before the start date in UI

**buildings.py:**

1. def date: To validate if the start date is not after the end date
2. def pincode: Validation for pin code length
3. SQL query to fetch only those buildings which are between the start and end date of the land with respect to today’s date

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Building Administrator | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Floor Supervisor | No | Yes | No | No | No | NA | NA | NA |
| 3 | Maintenance Department | No | Yes | No | No | No | NA | NA | NA |
| 4 | Employee | No | No | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Infrastructure> Building

## Floor

**Overview**

* The Floor Master Screen is a screen which will help the user to manage and track the layout of a building's floors and associated fixed assets. It provides an interface that allows users to view floor plans, assign fixed assets to specific locations, and record important maintenance information.
* One of the key features of the Floor Master Screen is the ability to upload layout drawings, which can be used as a reference for future maintenance and planning. This feature allows users to quickly identify the location of specific assets or areas within the building, which can save time and reduce the risk of errors.
* The Floor Master Screen allows users to record floor-wise details, such as assigning non-movable assets to specific locations. The application keeps this information in MM, but users can access it through the Infrastructure. This allows users to easily view the details of each floor and the assets assigned to it.

**Description of Problem**

Create a page containing all the information about the floor in a building, including the floor number, building name, number of rooms, Floor documents, and Floor Assets.

**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Flow Chart**

**Use Case Diagram**

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Building Name | Link Field | Fetched from Building master screen | Y |  |  |
| 2 | Floor Number | Text Field |  | Y |  |  |
| 3 | Floor Size | Integer Field |  |  |  |  |
| 4 | Number of rooms | Text Field |  | Y |  |  |
| 5 | Floor Plan | Attach |  |  |  |  |
| 6 | Floor Incharge | Text Field |  |  |  |  |
| **7** | **Floor Assets** | Table |  |  |  |  |
| **8** | **Floor Documents** | Table |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Floor Assets** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Item Code | Link Field | Fetched from Item master screen | Y |  |  |
| 2 | Item Name | Text Field | Auto fetch based on Item Code |  |  |  |
| 3 | Location | Link Field | Fetched from Location master screen |  |  |  |
| 4 | Asset Name | Text Field | User Input |  |  |  |
| 5 | Custodian | Link Field | Fetched from Employee master screen | Y |  |  |
| 6 | Department | Link Field | Fetched from Department screen | Y |  |  |

**Business Logic**

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Building Administrator | | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Floor Supervisor | | No | Yes | No | No | No | NA | NA | NA |
| 3 | Maintenance Department | | No | Yes | No | No | No | NA | NA | NA |
| 4 | Employee | | No | No | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Infrastructure> Floor

## Building Room

**Overview**

This screen shall be able to create rooms and store the features of every room while also assigning a room number.

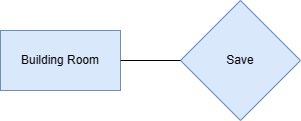
**Description of Problem**

Create a page containing all the information about the room in a building, including type of room, room number, vacancy status, and details about the features like parking and garden availability.

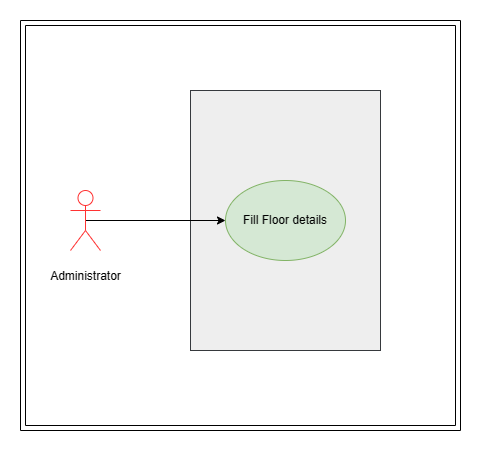
**Dependency**

Dependent on Building screen, as it has a mandatory field for Building details which will be autofetched from the Building screen.

**Flow Chart**

****

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Building Name | Link | Buildings |  |  |  |
| 2 | Building Type | Data |  |  |  |  |
| 3 | Building Address | Data |  | Y |  |  |
| 4 | Building's Land Plot number | Data |  | Y |  |  |
| 5 | Building's Land Address | Data |  | Y |  |  |
| 6 | District | Data |  | Y |  |  |
| 7 | State | Data |  | Y |  |  |
| 8 | Pin Code | Data |  | Y |  |  |
| 9 | Room No. | Data |  | Y |  |  |
| 10 | Floor | Data | Ground  1 ….. |  |  |  |
| 11 | Type of Room | Link | Building type Room | Y |  |  |
| 12 | Type of Residence | Link | Type of Residence | Y |  |  |

**Business Logic**

**building\_room.js:**

* + 1. To filter residence type name in the link field based on selected type of residence
    2. To filter buildings which are currently between start and end date
    3. To validate end date is not before the start date

**building\_room.py:**

1. To filter buildings which are currently between start and end date
2. def duplicate: To check for any duplicate records within a building regarding room no. or room type
3. def dateValidate: To validate if the start date is not after the end date in allotable room type

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Building Administrator | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Floor Supervisor | No | Yes | No | No | No | NA | NA | NA |
| 3 | Maintenance Department | No | Yes | No | No | No | NA | NA | NA |
| 4 | Employee | No | No | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Infrastructure> Building Room

## Building type Room

**Overview**

**This screen helps define the type of the room in a building, for example if it is a classroom or residential or conference hall.**

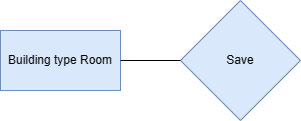
**Description of Problem**

Create a page containing information of types of room eg. Classroom or residential or any other.

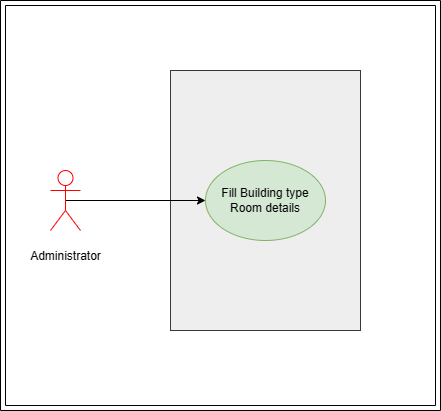
**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Flow Chart**

****

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Type of Room | Data |  | Y |  |  |
| 2 | Allotment Status | Drop down | Allottable  Non Allottable | Y |  |  |

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Building Administrator | | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Floor Supervisor | | No | Yes | No | No | No | NA | NA | NA |
| 3 | Maintenance Department | | No | Yes | No | No | No | NA | NA | NA |
| 4 | Employee | | No | No | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Infrastructure> Building Type Room

# Infrastructure Maintenance

## Asset

**Overview**

In Asset, you can maintain fixed asset records for Company assets like computers, furniture, cars, etc. and manage their depreciation, sale, or disposal. You can track locations of the assets or keep records of employees who are using the asset. You can also manage the maintenance details of the assets.

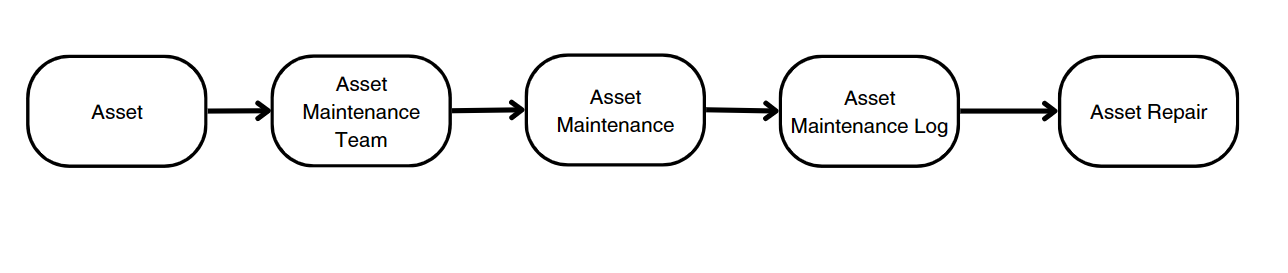
**Description of Problem**

Create a page to enter details about the asset.

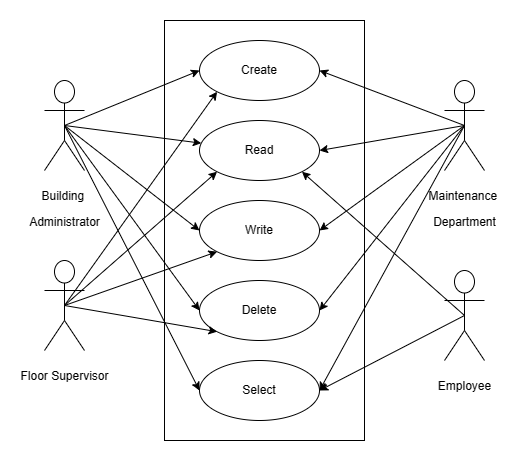
**Dependency**

Dependent on Item, Location, Custodian, Department, Cost Center, Purchase receipt/ Purchase Invoice and Finance Book.

**Flow Chart**



**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Company | Link | Fetched from Company screen | Yes |  |  |
| 2 | Item Code | Link | Fetched from Company screen | Yes |  |  |
| 3 | Item Name | Read Only | Populated based on selection of Item code |  |  |  |
| 4 | Asset Owner | Dropdown | Company/  Supplier /  Customer |  |  |  |
| 5 | Asset Owner Company | Link | Fetched from Company screen |  |  |  |
| 6 | Is Existing Asset | Checkbox | User Input |  |  |  |
| 7 | Supplier | Link | Fetched from Supplier screen |  |  |  |
| 8 | Customer | Link | Fetched from Customer screen |  |  |  |
| 9 | Image | Attach Image | User Input |  |  |  |
| 10 | Naming Series | Dropdown | ACC-ASS-.YYYY.- |  |  |  |
| 11 | Asset Name | Text Field | User Input | Yes |  |  |
| 12 | Asset Category | Link | Fetched from Asset Category screen |  |  |  |
| 13 | Location | Link | Fetched from Location screen | Yes |  |  |
| 14 | Custodian | Link | Fetched from Employee screen |  |  |  |
| 15 | Department | Link | Fetched from Department screen |  |  |  |
| 16 | Disposal Date | Date | User Input |  |  |  |
| 17 | Address | Link | Fetched from Address screen |  |  |  |
| 18 | Cost Center | Link | Fetched from Cost Center screen |  |  |  |
| 19 | Purchase Receipt | Link | Fetched from Purchase Invoice screen |  |  |  |
| 20 | Purchase Invoice | Link | Fetched from Purchase Invoice screen |  |  |  |
| 21 | Available-for-use Date | Date | User Input | Yes |  |  |
| 22 | Gross Purchase Amount | Currency | User Input | Yes |  |  |
| 23 | Asset Quantity | Int | User Input |  |  |  |
| 24 | Purchase Date | Date | User Input | Yes |  |  |
| 25 | Calculate Depreciation | Checkbox | User Input |  |  |  |
| 26 | Opening Accumulated Depreciation | Currency | User Input |  |  |  |
| 27 | Number of Depreciation's Booked | Int | User Input |  |  |  |
| 28 | **Finance Books** | Table | These table is describe below |  |  |  |
| 29 | Depreciation Method | Dropdown | Straight Line /  Double Declining Balance /  Manual |  |  |  |
| 30 | Value After Depreciation | Currency | User Input |  |  |  |
| 31 | Total Number of Depreciation's | Int | User Input |  |  |  |
| 32 | Frequency of Depreciation (Months) | Int | User Input |  |  |  |
| 33 | Next Depreciation Date | Date | User Input |  |  |  |
| 34 | **Depreciation Schedule** | Table | These table is describe below |  |  |  |
| 35 | Policy number | Text Field | User Input |  |  |  |
| 36 | Insurer | Text Field | User Input |  |  |  |
| 37 | Insured value | Text Field | User Input |  |  |  |
| 38 | Insurance Start Date | Date | User Input |  |  |  |
| 39 | Insurance End Date | Date | User Input |  |  |  |
| 40 | Comprehensive Insurance | Text Field | User Input |  |  |  |
| 41 | Maintenance Required | Checkbox | User Input |  |  |  |
| 42 | Status | Dropdown | Draft /  Submitted /  Partially Depreciated /  Fully Depreciated /  Sold /  Scrapped /  In Maintenance /  Out of Order /  Issue /  Receipt /  Capitalized /  Decapitalized |  |  |  |
| 43 | Booked Fixed Asset | Checkbox | User Input |  |  |  |
| 44 | Purchase Receipt Amount | Currency | User Input |  |  |  |
| 45 | Default Finance Book | Link | Fetched from Finance Book screen |  |  |  |
| 46 | Depreciation Entry Posting Status | Dropdown | Successful /  Failed |  |  |  |
| 47 | Manage | Button | Transfer Asset: On click of this button you can transfer Asset from One employee/Location to another |  |  |  |
| Scrap Asset: On click of this button you can scrap Asset |  |  |  |
| Sell Asset: On click of this button you can sell Asset |  |  |  |
| Maintain Asset: On click of this button you can create Maintenance schedule for an Asset |  |  |  |
| Repair Asset: On click of this button you can create repair details for an Asset |  |  |  |
| Split Asset: On click of this you can split Asset |  |  |  |
| Adjust Asset value: On click of this button you can see the value of Asset |  |  |  |
| View Ledger Entry: On click of this button you can view General entry for the Asset |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Finance Books (Child Table)** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Finance Book | Link | Fetched from Finance book screen |  |  |  |
| 2 | Depreciation Method | Dropdown | Straight Line /  Double Declining Balance /  Written Down Value /  Manual | Yes |  |  |
| 3 | Total Number of Depreciation | Int | User Input | Yes |  |  |
| 4 | Frequency of Depreciation (Months) | Int | User Input | Yes |  |  |
| 5 | Depreciation Posting Date | Date | User Input |  |  |  |
| 6 | Expected Value After Useful Life | Currency | User Input |  |  |  |
| 7 | Value After Depreciation | Currency | User Input |  |  |  |
| 8 | Rate of Depreciation | Percent | User Input |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Depreciation Schedule (Child Table)** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Finance Book | Link | Fetched from Finance book screen |  |  |  |
| 2 | Schedule Date | Date | User Input | Yes |  |  |
| 3 | Depreciation Amount | Currency | User Input | Yes |  |  |
| 4 | Accumulated Depreciation Amount | Currency | User Input |  |  |  |
| 5 | Journal Entry | Link | Fetched from Finance book screen |  |  |  |
| 6 | Make Depreciation Entry | Button | On click of this button make depreciation entry for Asset |  |  |  |
| 7 | Finance Book Id | Text Field | User Input |  |  |  |
| 8 | Depreciation Method | DropDown | User Input |  |  |  |

**Business Logic**

1. Go to the Assets list, click on New.
2. Enter a name for the asset.
3. Select the Item Code. Item Name and Asset Category will be fetched automatically.
4. Select the Asset Owner, i.e. Company, Supplier, or Customer.
5. Select the Company/Supplier/Customer.
6. Select the Purchase Receipt/Purchase Invoice. Purchase Date and Gross Purchase Amount will be fetched automatically.
7. Select a Location. Eg: Mumbai Office. This will be fetched automatically if specified in Purchase Receipt items table
8. Set Available-for-use Date. The depreciation will be calculated starting from this date.
9. Click on Save and Submit.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
|  | Building Administrator | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Floor Supervisor | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Maintenance Department | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Employee | Yes | Yes | No | No | No | No | No | No |

**Path of the Screen**

Home > Material Management > Assets > Asset

## Asset Maintenance Team

**Overview**

The Asset Maintenance Team is responsible for carrying out maintenance activities on the asset.

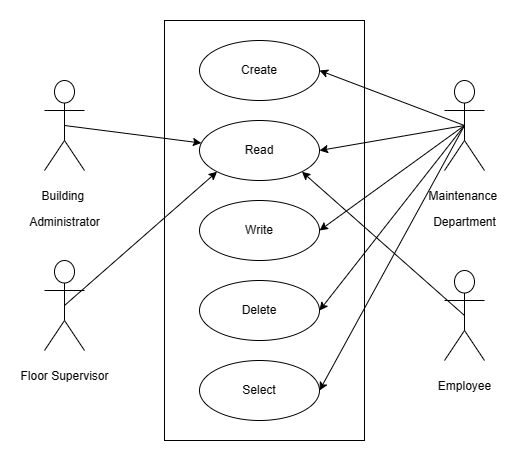
**Description of Problem**

Create a page to form a asset maintenance team required for maintenance of asset.

**Dependency**

Dependent on User, Role, and Company screens.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Maintenance Team Name | Text Field | User Input | Yes |  |  |
| 2 | Maintenance Manager | Link | Fetched from User screen |  |  |  |
| 3 | Maintenance Manager Name | Read Only | Fetched through Maintenance Manager |  |  |  |
| 4 | Company | Link | Fetched from Company screen | Yes |  |  |
| **5** | **Maintenance Team Members** | **Table** |  | Yes | These table is describe below |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Maintenance Team Members** | | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | | **R/N/D** |
| 1 | Team Member | Link | Fetched from User screen | Yes |  |  | |
| 2 | Full Name | Text Field | User Input |  |  |  | |
| 3 | Maintenance Role | Link | Fetched from Role screen | Yes |  |  | |

**Business Logic**

1. Go to the Asset Maintenance Team list, click on New.
2. Enter a name for the team.
3. Select a manager for the team.
4. In the Maintenance Team Members table, add the team members and select their maintenance [roles](https://docs.erpnext.com/docs/v13/user/manual/en/setting-up/users-and-permissions/role-and-role-profile).
5. Save.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
|  | Building Administrator | No | Yes | No | No | No | No | No | No |
|  | Floor Supervisor | No | Yes | No | No | No | No | No | No |
|  | Maintenance Department | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Employee | No | Yes | No | No | No | No | No | No |

**Path of the Screen**

Home > Material Management > Maintenance> Asset Maintenance Team

## Asset Maintenance

**Overview**

Asset Maintenance refers to any activity done on Assets to maintain their performance or condition.

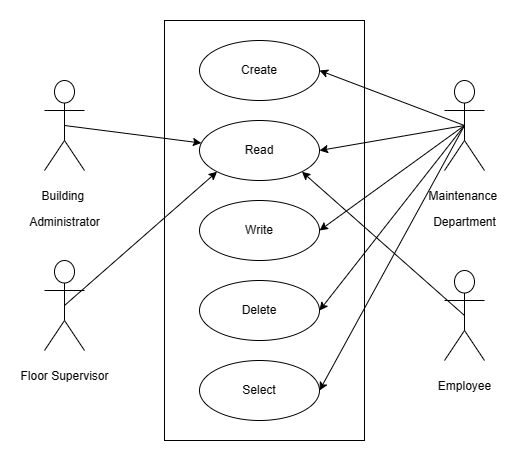
**Description of Problem**

Create a page to view all the details regarding asset maintenance from planning to the actual maintenance process.

**Dependency**

Dependent on Asset, Asset Maintenance Team and Company screen.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Asset Name | Link | Fetched from Asset screen | Yes |  |  |
| 2 | Asset Category | Read Only | Fetched from selected asset |  |  |  |
| 3 | Company | Link | Fetched from Company screen | Yes |  |  |
| 4 | Item Code | Read Only | Fetched from selected asset |  |  |  |
| 5 | Item Name | Read Only | Fetched from selected asset |  |  |  |
| 6 | Maintenance Team | Link | Fetched from Asset Maintenance Team screen | Yes |  |  |
| 7 | Maintenance Manager | Text Field | User Input |  |  |  |
| 8 | Maintenance Manager Name | Read Only | Fetched from selected Asset Maintenance Team |  |  |  |
| 9 | **Maintenance Tasks** | **Table** |  | **Yes** | These table is describe below |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Maintenance Task** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Maintenance Task | Text Field | User Input | Yes |  |  |
| 2 | Maintenance Type | Select | Preventive Maintenance /  Calibration |  |  |  |
| 3 | Maintenance Status | Select | Planned / Overdue / Canceled | Yes |  |  |
| 4 | Start Date | Text Field | User Input | Yes |  |  |
| 5 | Periodicity | Select | Daily / Weekly / Monthly /  Quarterly / Yearly / 2 Yearly | Yes |  |  |
| 6 | End Date | Date | User Input |  |  |  |
| 7 | Certificate Required | Checkbox | User Input |  |  |  |
| 8 | Assign To | Link | Fetched from User screen |  |  |  |
| 9 | Assign to Name | Read Only | Feted from value selected from Assign to value |  |  |  |
| 10 | Next Due Date | Date | User Input |  |  |  |
| 11 | Last Completion Date | Date | User Input |  |  |  |
| 12 | Description | Text Editor | User Input |  |  |  |

**Business Logic**

1. Go to the Asset Maintenance list, click on New.
2. Select the Asset.
3. Select the Asset Maintenance Team.
4. Add Maintenance Tasks in the table.
5. Set the Maintenance Status whether 'Planned', 'Overdue', or 'Canceled'.
6. Select a periodicity for which the task needs to be carried out. The next due date will be calculated.
7. Save.
8. After saving, you can assign the task to a user.
9. If the Item is serialized, the Serial Number can be entered.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
|  | Building Administrator | No | Yes | No | No | No | No | No | No |
|  | Floor Supervisor | No | Yes | No | No | No | No | No | No |
|  | Maintenance Department | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Employee | No | Yes | No | No | No | No | No | No |

**Path of the Screen**

Home > Material Management > Maintenance> Asset Maintenance

## Asset Maintenance Log

**Overview**

Asset Maintenance Log logs the tasks carried out in an Asset Maintenance.

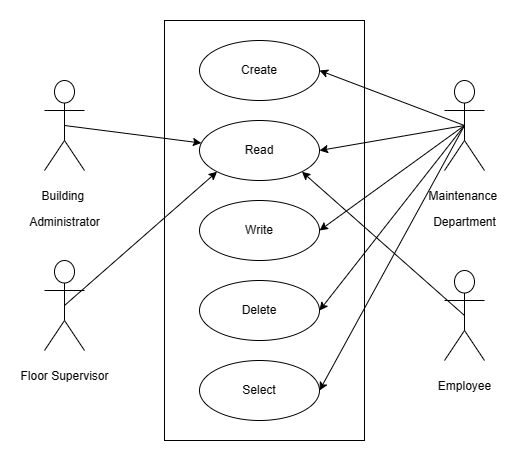
**Description of Problem**

Create a page to enter asset maintenance log.

**Dependency**

Dependent on Asset Maintenance screen.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Asset Maintenance | Link | Fetched from Asset Maintenance screen |  |  |  |
| 2 | Series | Dropdown | ACC-AML-.YYYY.- | Yes |  |  |
| 3 | Asset Name | Read Only | Fetched from Asset Maintenance asset |  |  |  |
| 4 | Item Code | Read Only | Fetched from Asset Maintenance asset |  |  |  |
| 5 | Item Name | Read Only | Fetched from Asset Maintenance asset |  |  |  |
| 6 | Task | Link | Fetched from Task screen |  |  |  |
| 7 | Task Name | Text Field | User Input |  |  |  |
| 8 | Maintenance Type | Read Only | Fetched from Task asset |  |  |  |
| 9 | Periodicity | Text Field | User Input |  |  |  |
| 10 | Has Certificate | Checkbox | User Input |  |  |  |
| 11 | Certificate | Attach | User Input |  |  |  |
| 12 | Maintenance Status | Dropdown | Planned/ Completed /  Cancelled / Overdue | Yes |  |  |
| 13 | Assign To | Read Only | Fetched from Task asset |  |  |  |
| 14 | Due Date | Date | User Input |  |  |  |
| 15 | Completion Date | Date | User Input |  |  |  |
| 16 | Description | Read Only | Fetched from Task asset |  |  |  |
| 17 | Actions performed | Text Editor | User Input |  |  |  |

**Business Logic**

This screen is updated automatically hence no business logic applicable.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
|  | Building Administrator | No | Yes | No | No | No | No | No | No |
|  | Floor Supervisor | No | Yes | No | No | No | No | No | No |
|  | Maintenance Department | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Employee | No | Yes | No | No | No | No | No | No |

**Path of the Screen**

Home > Material Management > Maintenance> Asset Maintenance Log

## Asset Repair

**Overview**

Asset Repair refers to any activity carried to repair a broken Asset to restore full functionality.

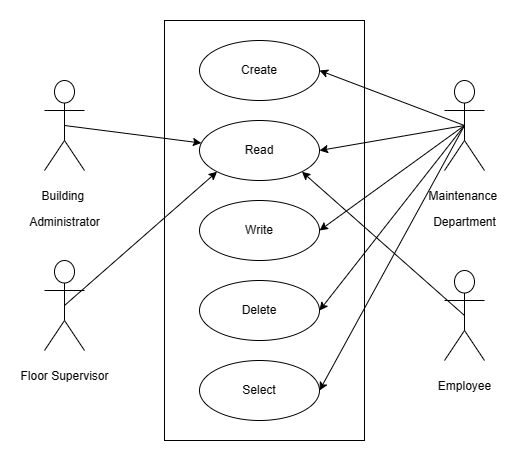
**Description of Problem**

Create a page to enter details about the repaired asset.

**Dependency**

Dependent on Asset screen.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Asset | Link | Fetched from Asset Maintenance screen | Yes |  |  |
| 2 | Company | Link | Fetched from Asset Maintenance screen |  |  |  |
| 3 | Asset Name | Read Only | Fetched from value in Asset field |  |  |  |
| 4 | Series | Dropdown | ACC-ASR-.YYYY.- | Yes |  |  |
| 5 | Failure Date | Datetime | User Input | Yes |  |  |
| 6 | Repair Status | Dropdown | Pending / completed /Cancelled |  |  |  |
| 7 | Completion Date | Datetime | User Input |  |  |  |
| 8 | Address | Link | Fetched from Address screen |  |  |  |
| 9 | Article | Link | Fetched from Article screen |  |  |  |
| 10 | Cost Center | Link | Fetched from Cost center screen |  |  |  |
| 11 | Project | Link | Fetched from Project screen |  |  |  |
| 12 | Repair Cost | Currency | User Input |  |  |  |
| 13 | Capitalize Repair Cost | Checkbox | User Input |  |  |  |
| 14 | Stock Consumed During Repair | Checkbox | User Input |  |  |  |
| 15 | Purchase Invoice | Link | Fetched from Purchase Invoice screen |  |  |  |
| 16 | Warehouse | Link | Fetched from Warehouse screen |  |  |  |
| 17 | **Stock Items** | Table |  |  | These table is describe below |  |
| 18 | Total Repair Cost | Currency | User Input |  |  |  |
| 19 | Stock Entry | Link | Fetched from Stock Entry screen |  |  |  |
| 20 | Increase In Asset Life(Months) | Int | User Input |  |  |  |
| 21 | Error Description | Long Text | User Input |  |  |  |
| 22 | Actions performed | Long Text | User Input |  |  |  |
| 23 | Downtime | Text Field | User Input |  |  |  |
| 24 | View General Ledger | Button | On click of the button you can General ledger for the Asset repair |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Stock Items (Child Table)** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/ Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Item | Link | Fetched from Item screen |  |  |  |
| 2 | Valuation Rate | Currency | User Input |  |  |  |
| 3 | Consumed Quantity | Text Field | User Input |  |  |  |
| 4 | Total Value | Currency | User Input |  |  |  |
| 5 | Serial No | Small Text | User Input |  |  |  |

**Business Logic**

1. Go to the Asset Repair list, click on New.
2. Select the Asset.
3. Select the Failure Date.
4. Enter the Repair Cost.
5. Click on Save.
6. Change the Repair Status from 'Pending' to 'Completed', or 'Canceled'.
7. Select a Purchase Invoice if Repair Cost is greater than zero.
8. Click on Save and Submit.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
|  | Building Administrator | No | Yes | No | No | No | No | No | No |
|  | Floor Supervisor | No | Yes | No | No | No | No | No | No |
|  | Maintenance Department | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|  | Employee | No | Yes | No | No | No | No | No | No |

**Path of the Screen**

Home > Material Management > Maintenance> Asset Repair

# Infrastructure Project Management

## Project

**Overview**

A Project is a planned piece of work that is designed to find information about something, to produce something new, or to improve something.

In this module, project management is task-driven. You can create a Project and divide it into multiple Tasks.

A Project has a broad scope and hence can be divided into tasks. Think of coming up with a new smartphone for the next year as a Project. Then things like designing, prototyping, testing, delivery, etc. become tasks under the project.

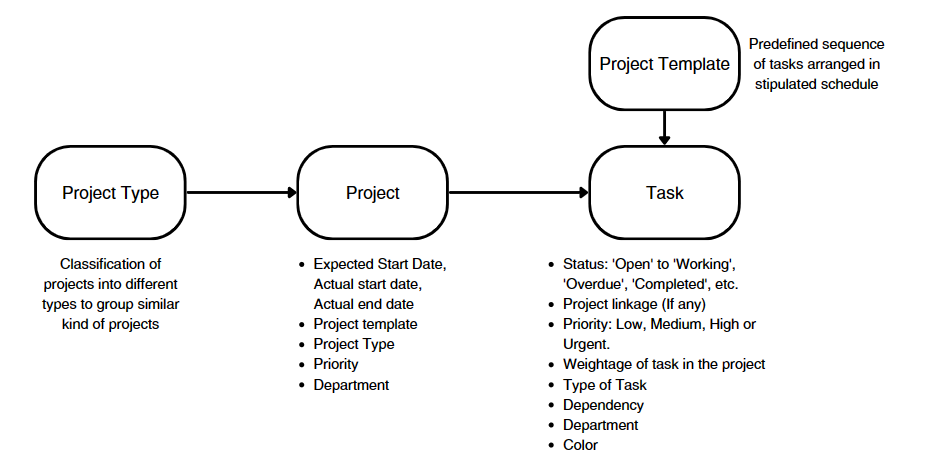
**Description of Problem**

Create a page to maintain the database for projects which will include all the required details regarding a project like duration, pro0ject type, users, priority, etc.

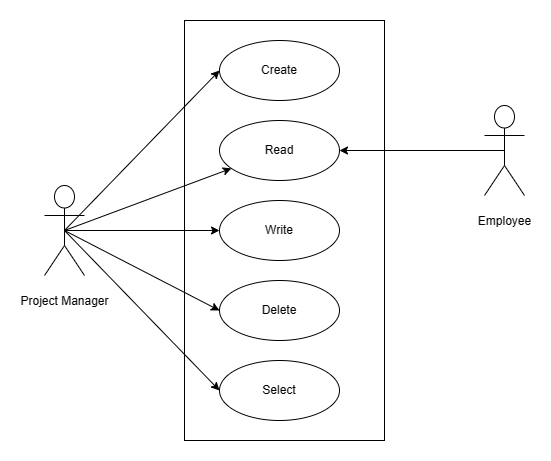
**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Flow Chart**



**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Project Name | Text Field | User Input | Y |  |  |
| 2 | Status | Dropdown | Open  Completed  Cancelled |  |  |  |
| 3 | Project Type | Link | Fetched from Project type screen |  |  |  |
| 4 | Is Active | Dropdown | Yes  No |  |  |  |
| 5 | % Complete Method | Dropdown | Manual  Task Completion  Task Progress  Task Weight |  |  |  |
| 6 | % Completed | Percent |  |  |  |  |
| 7 | From Template | Link | Fetched from Project Template screen |  |  |  |
| 8 | Expected Start Date | Date picker | User Input |  |  |  |
| 9 | Expected End Date | Date picker | User Input |  |  |  |
| 10 | Priority | Dropdown | Medium  Low  High |  |  |  |
| 11 | Department | Link | Fetched from Department screen |  |  |  |
| 12 | Customer | Link | Fetched from Customer screen |  |  |  |
| 13 | Sales Order | Link | Fetched from Sales Order screen |  |  |  |
| 14 | **Project User** | Table | Project User Table is described below |  |  |  |
| 15 | Copied From |  |  |  |  |  |
| 16 | Notes | Text Editor | User Input |  |  |  |
| 17 | Actual Start Date (via Time Sheet) | Date picker | User Input |  |  |  |
| 18 | Actual Time (in Hours via Time Sheet) | Number | User Input |  |  |  |
| 19 | Actual End Date (via Time Sheet) | Date picker | User Input |  |  |  |
| 20 | Estimated Cost | Currency | User Input |  |  |  |
| 21 | Total Costing Amount (via Timesheets) | Currency | User Input |  |  |  |
| 22 | Total Purchase Cost (via Purchase Invoice) | Currency | User Input |  |  |  |
| 23 | Company | Link | Fetched from company screen |  |  |  |
| 24 | Total Sales Amount (via Sales Order) | Currency |  |  |  |  |
| 25 | Total Billable Amount (via Timesheets) | Currency |  |  |  |  |
| 26 | Total Billed Amount (via Sales Invoices) | Currency |  |  |  |  |
| 27 | Total Consumed Material Cost (via Stock Entry) | Currency |  |  |  |  |
| 28 | Default Cost Center | Link | Fetched from Cost Center screen |  |  |  |
| 29 | Gross Margin | Currency | User Input |  |  |  |
| 30 | Gross Margin % | Percent | User Input |  |  |  |
| 31 | Collect Progress | Check box | User Input |  |  |  |
| 32 | Holiday List | Link | Fetched from Holiday List screen |  |  |  |
| 33 | Frequency To Collect Progress | Dropdown | Hourly  Twice Daily  Daily  Weekly |  |  |  |
| 34 | From Time | Time picker | User Input |  |  |  |
| 35 | To Time | Time picker | User Input |  |  |  |
| 36 | First Email | Time picker | User Input |  |  |  |
| 37 | Second Email | Time picker | User Input |  |  |  |
| 38 | Daily Time to send | Time picker | User Input |  |  |  |
| 39 | Day to Send | Dropdown | Monday  Tuesday  Wednesday  Thursday  Friday  Saturday  Sunday |  |  |  |
| 40 | Weekly Time to send | Time picker | User Input |  |  |  |
| 41 | Message | Text Field | User Input |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project User** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | User | Link | Fetched from User screen |  |  |  |
| 2 | Email | Read Only | Auto fetched Based on User Field |  |  |  |
| 3 | Image | Read Only | Auto fetched Based on User Field |  |  |  |
| 4 | Full Name | Read Only | Auto fetched Based on User Field |  |  |  |
| 5 | Welcome email sent | Check Box | User Input |  |  |  |
| 6 | View attachments | Check Box | User Input |  |  |  |
| 7 | Project Status | Text Field |  |  |  |  |

**Business Logic**

1. Go to the Project list and click on New.
2. Add the following details:

**Project Name:** Title of the Project.

**Status:** The default status of a Project is going to be 'Open' which can later be changed to 'Completed' or 'Cancelled'.

**Expected End Date:** Enter the date on which you aim to finish the project.

1. Save

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Project Manager | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Employee | No | Yes | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Projects > Projects > Project

## Task

**Overview**

In project management, a task is an actionable unit or activity which needs to be completed. Whilst each task within a Project can be assigned to an individual or a group of individuals, the assignment can also be done at the project level.

These Tasks can be created from a Project itself or a Task can be created separately as well.

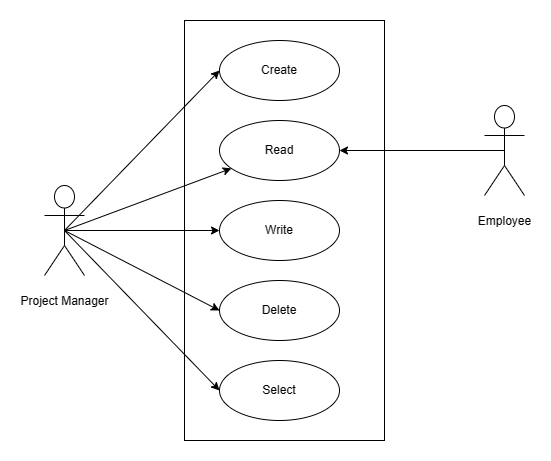
**Description of Problem**

Create a page to maintain and record the project linked task as well as independent tasks, completion method and costing.

**Dependency**

Not dependent on anything other than being filled and saved by the administrator.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Subject | Text Field | User Input | Y |  |  |
| 2 | Project | Link | Fetched from Project screen |  |  |  |
| 3 | Issue | Link | Fetched from Issue screen |  |  |  |
| 4 | Type | Link | Fetched from Task Type screen |  |  |  |
| 5 | Color | Color picker | User Input |  |  |  |
| 6 | Is Group | Check Box | User Input |  |  |  |
| 7 | Is Template | Check Box | User Input |  |  |  |
| 8 | Status | Drop down | Open  Working  Pending Review  Overdue  Template  Completed  Cancelled |  |  |  |
| 9 | Priority | Drop down | Low  Medium  High  Urgent |  |  |  |
| 10 | Weight | Number | User Input |  |  |  |
| 11 | Parent Task | Link | Fetched from Task screen |  |  |  |
| 12 | Completed By | Link | Fetched from User screen |  |  |  |
| 13 | Completed On | Date picker | User Input |  |  |  |
| 14 | Expected Start Date | Date picker | User Input |  |  |  |
| 15 | Expected Time (in hours) | Number | User Input |  |  |  |
| 16 | Begin On (Days) | Number | User Input |  |  |  |
| 17 | Expected End Date | Date picker | User Input |  |  |  |
| 18 | % Progress | Percent | User Input |  |  |  |
| 19 | Duration (Days) | Number | User Input |  |  |  |
| 20 | Is Milestone | Check Box | User Input |  |  |  |
| 21 | Task Description | Text Editor | User Input |  |  |  |
| 22 | **Task Depends On** | Table | Task Depends On Table is described below |  |  |  |
| 23 | Actual Start Date (via Time Sheet) | Date picker | User Input |  |  |  |
| 24 | Actual Time (in Hours via Time Sheet) | Number | User Input |  |  |  |
| 25 | Actual End Date (via Time Sheet) | Date picker | User Input |  |  |  |
| 26 | Total Costing Amount (via Time Sheet) | Currency | User Input |  |  |  |
| 27 | Total Billing Amount (via Time Sheet) | Currency | User Input |  |  |  |
| 28 | Review Date | Date picker | User Input |  |  |  |
| 29 | Closing Date | Date picker | User Input |  |  |  |
| 30 | Department | Link | Fetched from Department screen |  |  |  |
| 31 | Company | Link | Fetched from Company screen |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Depends On** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Task | Link | Fetched from Task screen |  |  |  |
| 2 | Subject | Text Field | User Input |  |  |  |
| 3 | Project | Text Field | User Input |  |  |  |

**Business Logic**

1. Go to the Task List and click on New.
2. Add the subject of the task.
3. Save.

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Project Manager | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Employee | No | Yes | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Projects > Projects > Task

## Project Template

**Overview**

A Project Template is a predefined sequence of tasks arranged in stipulated schedule.

These templates can be pulled for similar kind of Projects and the tasks therein get auto-populated at the time of creation of each new project.

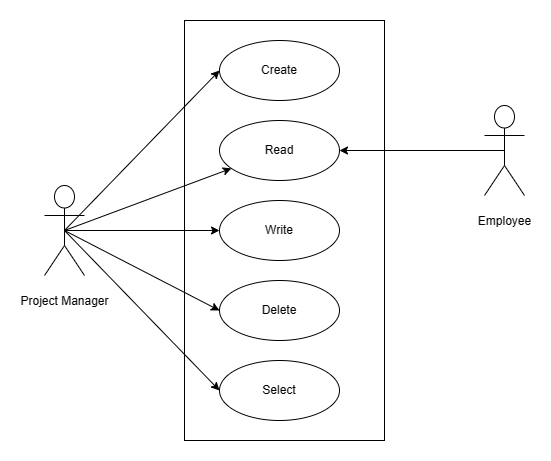
**Description of Problem**

Create a page to maintain a task list so that it can be used by multiple projects as template. This will include a name of the project, project type and tasks.

**Dependency**

Dependent on Task screen.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Name | Text Field | User Input |  |  |  |
| 2 | Project Type | Link | Fetched from Project Type screen |  |  |  |
| 3 | **Project Template Task** | Table | User Input | Y |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Template Task** | | | | | | |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Task | Link | Fetched from Task screen | Y |  |  |
| 2 | Subject |  | Read Only |  |  |  |

**Business Logic**

1. Go to the Project Template List and click on New.
2. Add the following details:

**Project Template Name:** Title of the Project Template

**Project Type:** Project Templates, just like projects can be classified into different project types, e.g., Internal or External.

**Tasks:** Each Project Template will have a set of a predefined sequence of tasks. In this table, you can select the tasks you want for this template

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Project Manager | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Employee | No | Yes | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Projects > Projects > Project Template

## Project Type

**Overview**

A Project Type is the classification of projects into different types to group similar kind of projects.

Internal Projects, External Projects could be examples of Project types and are already created in the system. You can choose to add more Project Types. This comes handy when you are reviewing the Projects, and you want to filter the information based on Project Types.

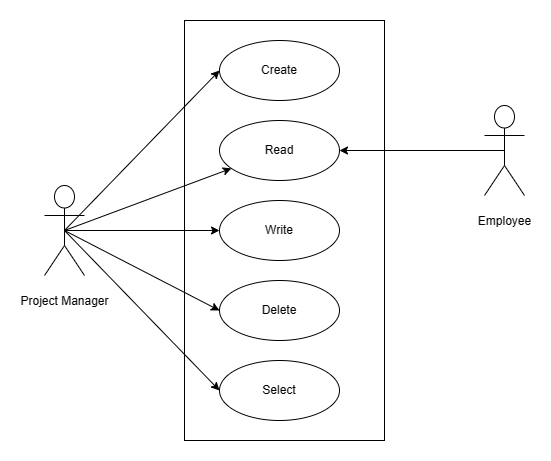
**Description of Problem**

Create a page to maintain project type.

**Dependency**

Not dependent on anything other than being filled and saved.

**Use Case Diagram**

****

**Form Elaboration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Field Label** | **Field Type** | **Validation/Action** | **Mandatory** | **Remarks** | **R/N/D** |
| 1 | Project Type | Text field | User Input | Y |  |  |
| 2 | Description | Text field | User Input |  |  |  |

**Business Logic**

1. Go to the Project Type List and click on New.
2. Add the name of the Project Type and description
3. Save

**Screen Authorization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Select** | **Read** | **Write** | **Create** | **Delete** | **Submit** | **Cancel** | **Amend** |
| 1 | Project Manager | Yes | Yes | Yes | Yes | Yes | NA | NA | NA |
| 2 | Employee | No | Yes | No | No | No | NA | NA | NA |

**Path of the Screen**

Home > Projects > Projects > Project Type