**Asset Management Documentation**



Team Members:

|  |
| --- |
| Team |
| Yash Pawar (Scrum Master) |
| Aditya Sovani |
| Sonali Deshmukh |

Capgemini India Technology Services

Plot 72 & 73, EPIP Zone, Vijayanagar, KIADB Export Promotion Industrial Area, Whitefield, Bengaluru, Karnataka 560066

Table of Contents

1. Introduction ------------------------------------------------------------------------------------
2. Overview-----------------------------------------------------------------------------------------
3. Team Member Work Allocation--------------------------------------------------------------
4. Epic & Stories -----------------------------------------------------------------------------------
5. Use Cases ----------------------------------------------------------------------------------------
   * 1. Asset Management System -----------------------------------------------------------
6. Add an asset ---------------------------------------------------------------------
7. Edit an asset ---------------------------------------------------------------------
8. Delete an asset------------------------------------------------------------------
9. View an asset------------------------------------------------------------
10. Search an asset------------------------------------------------------------------
    * 1. Login Management System -----------------------------------------------------------
11. User Login --------------------------------------------------------------------------
12. User Logout -------------------------------------------------------------------------
    * 1. Employee Management System -----------------------------------------------------------
13. Add an Employee ------------------------------------------------------------------
14. Check an Employee ----------------------------------------------------------------
15. View an Employee -----------------------------------------------------------------
    * 1. Asset Allocation Management System ---------------------------------------------------
16. Allocated Assets --------------------------------------------------------------------
17. Unallocated Assets -----------------------------------------------------------------
18. Pending Assets ---------------------------------------------------------------------
    * 1. Cloud Management ------------------------------------------------------------------------
19. Class Diagram -----------------------------------------------------------------------------------
20. ER Diagram -------------------------------------------------------------------------------------
21. System Requirements --------------------------------------------------------------------------

1. Introduction

Asset Management System is an web based application, which is indispensable for all organizations and with an intention of putting the details about all the machinery working in the different assembly lines of a big factory. The Asset management system is primarily implemented in a plant which has multiple assembly lines with heavy machinery working in that.



Some examples of hardware Assets used in IT Companies

2.Overview

This is an Internet based application that can be accessed throughout the organization and this is a web-based application that can be accessed over the web. This system can be used to search for an asset based on search condition, assign a hardware asset to/from an employee based on request, insert new asset details, modify an existing asset details and display all asset allocation request details. This is an integrated system that contains both the user (Manager) component and the Admin component.

The software helps in an accurate record of all types of assets. Using the software can help in efficient resource planning. It can also reduce the risk of theft of assets. An asset management system will help to monitor the assets located in different locations and departments. You will get to know where the assets are located. You can run reports to know about ownership, service details, and other insightful information.

4. Epic & Stories

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epic | Stories | | As a/an | I want to | So that… |
| Asset Management System | Add an Asset | | Admin | Enter the asset details | A new asset is added to the database |
| Update an Asset | | Admin | Change the details entered for the assets | To update the assets details or to increase the quantity of a particular asset in stock |
| Delete an Asset | | Admin | To delete an asset | Asset will be soft deleted from the database |
| Search an Asset | | User | To search an asset by name | Easy to find an asset from the entire list |
| View an Asset | | User | To view all assets | To look which assets are available |
| Epic | | Stories | As a/an | I want to | So that… |
| Login Management System | | Logged in as Manager/User | User/Manager | Search, view and raise request for assets | User/Manager get to know which assets company are providing |
| Logged in as Admin | Admin | To add, delete and update an Asset | To update the assets details to the users |
|  | | Logout | Admin and User/Manager | To logout from the system | To stop using the system we use logout |
| Epic | | Stories | As a/an | I want to | So that... |
| Employee Management System | | Add an Employee | User | Enter the Employee details | A new employee is added to the database |
| Check an Employee | User | Search employees | To search the employee details |
| View an Employee | User | To display the list of employees | Display the list of the employees |
| Epic | | Stories | As a/an | I want to | So that... |
| Asset Allocation Management System | | Allocated Assets | Admin | Allocate required and available assets to employees | Easy to find the Allocated Assets |
| Unallocated Assets | Admin | Unallocate the Assets | Don’t approve the assets request which are not in need |
| Pending Assets | Admin | Stay on pending to check the requirements later | Check the requirements later that why and for what reason the asset is required |
|  | | Raise Request | User | Raise request for particular assets | The employee gates the assets from the company for his/her work. |

1. Use Cases

5.1 Asset Management System

Overview

Asset management is the process of developing, operating, maintaining, and selling assets in a cost-effective manner. Most commonly used in finance, the term is used in reference to individuals or firms that manage assets on behalf of individuals or other entities.

Prerequisite

Actor must be logged in as Admin to perform the add asset, update asset and delete asset functionalities.

Actor must be logged in as User/Manager to Search asset and view asset functionalities can be performed by any user.

Non-Functional Requirement:

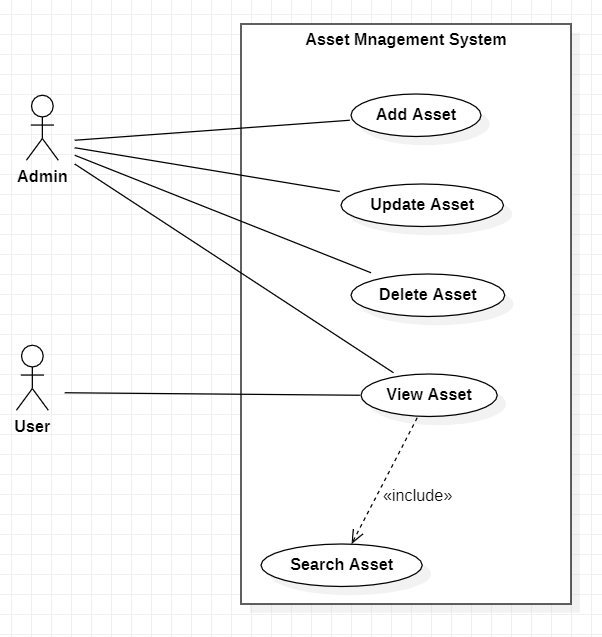
All modules are show designed that the result will be obtained within 10 seconds of request. Proper Loading indicator has been instantiated. Hence once a request been asked no one can further request anything. Authentication has been given a higher priority. All form validations are properly maintained.

Designed and Implemented By:

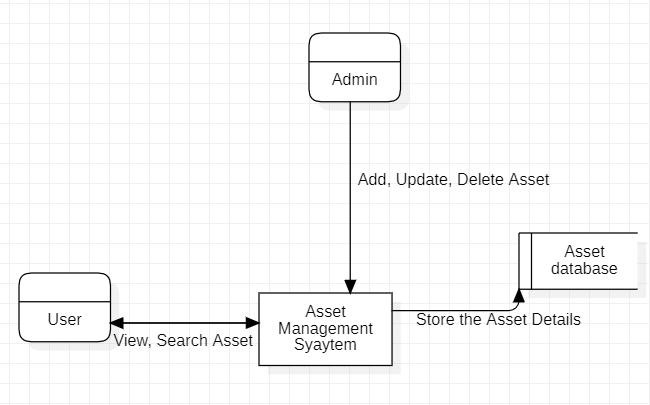
Yash Pawar (Id: JEECloudISGH28)

Designation:

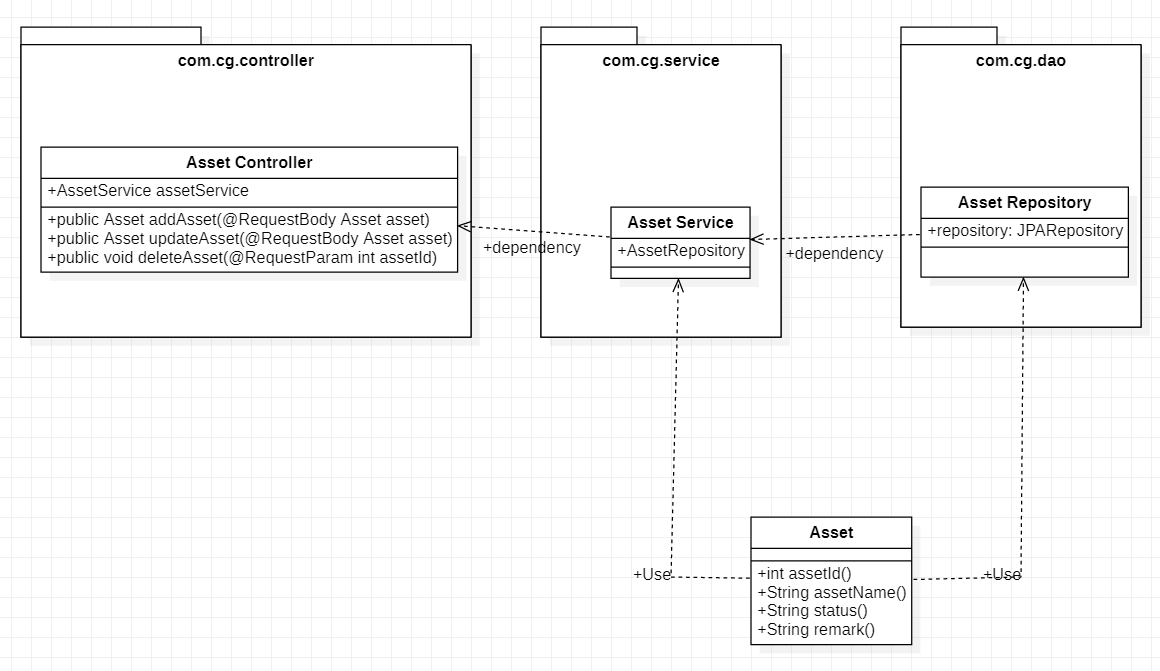
Use Case Diagram for Asset Management System



Asset Management Data Flow Diagram Level 0



Class Diagram for Asset Management



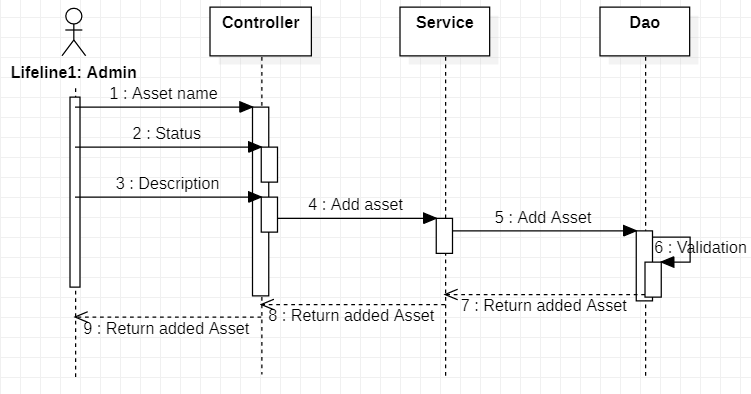
1. Add an Asset

This module has been designed to add an asset in the database. This can be only accessed by Admin. The form is designed with proper validation.

Validation:

All fields are required.

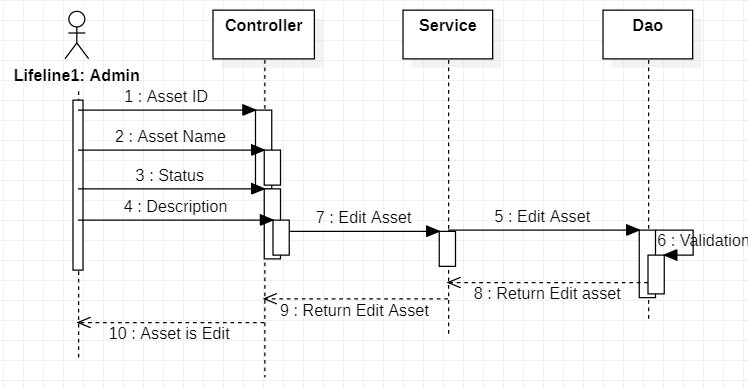
Sequence Diagram for Adding a Product



* 1. Edit an asset

It has been observed that during manual entry for an asset any individual can often make mistakes in entering data. Hence there should always be a scope to update the data which has been once entered. Thus, it is our client requirement to introduce an edit option. Hence, we have implemented a user-friendly edit option for our asset management system. Just on clicking an edit icon over a particular asset the edit form will pop up. Admin has to make changes. More-over it will be treated as a completely new asset.

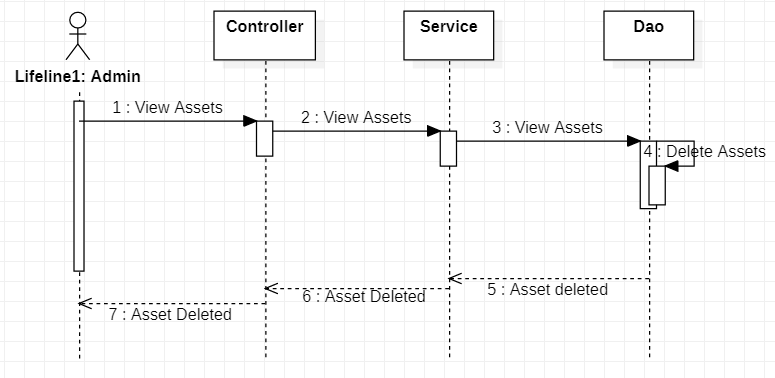
Sequence Diagram for Editing an Asset



* 1. Delete an Asset

Admin needs to click a button to delete an asset. On clicking on the button, the asset will be deleted. It has been found through several researches that there can be nothing more important than data. Hence no one wants to lose any data which has been one generated. Thus, we have implemented soft delete operation for deleting an asset. It will only change the asset present status as false and will not show up in asset page. However, it will be always recorded in the database.

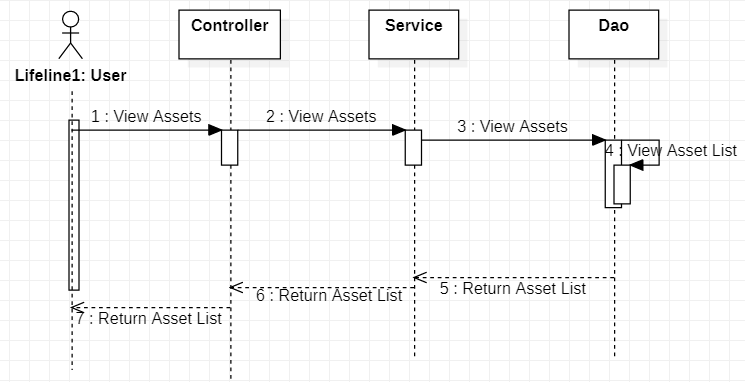
Sequence Diagram for Deleting an Asset



* 1. View an asset

Asset Management System uses a view concept for the protection of authorizations. Therefore, the Asset view authorization object has a special function. The asset view applies especially to user/manager who have only occasional limited contacts with fixed assets. The asset view allows such employees only a limited view of asset data whether or not they formally have access to every master record.

Sequence Diagram for Deleting an Asset



* 1. Search an asset

The entire model has been designed in such a way that we can enter any keyword related to the asset to find the particular asset. Keyword such as asset name, product name or description feature can be used. The entire functionality has been designed in Angular.

Project Progress Status (Asset Management System)

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint# | Task Assigned | Status | Remark |
| Sprint 1 | Creating UML Diagrams, Defining Test Cases and Sequence Diagrams, Implement the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence) | Modules Implemented using Java. Junit test cases are written and successfully tested.  Class Diagrams and Use case Diagrams are modified | Update the Class diagram, include the sequence diagram. Develop Knowledge about micro services architecture, Code Convention is not up to industry standard. Presentation layer is not implemented. |
| Sprint 2 | Spring Boot Implementation | Documentation Started | NA |

5.2Login Management System

Overview

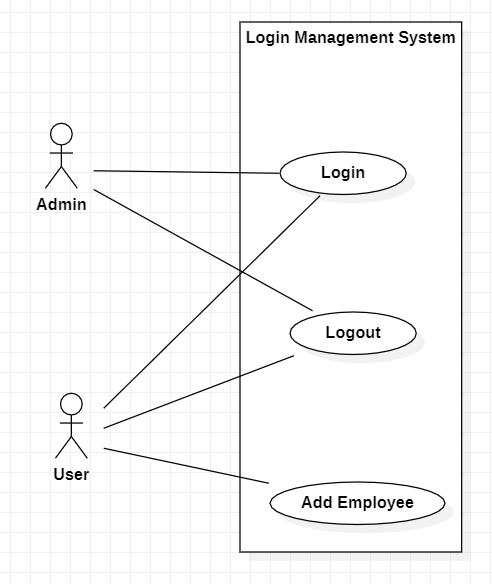
Login management is one of the main aspects for any application. Logging-In as different roles like ADMIN and USER/MANAGER and Logging-out comes under this system. Our Software is made highly secured using authentication guard of angular. Also, Authorization is used to ensure that different types of users can access different types of functionality.

Designed and Implemented By:

Yash Pawar (Id: JEECloudISGH28)

Designation:

Use Case Diagram for User Management System



1. User Login

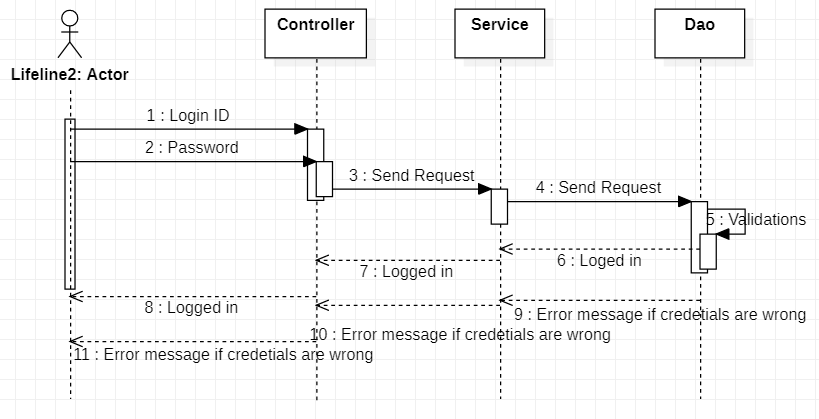
This module has been designed to Log-In an existing Admin or User/Manager. Any type of actor can Log-in from this functionality. The form is designed with proper validation.

***Validation***:

**Password:** Not Applicable.

*\*All fields are required.*

Sequence Diagram for User Registration



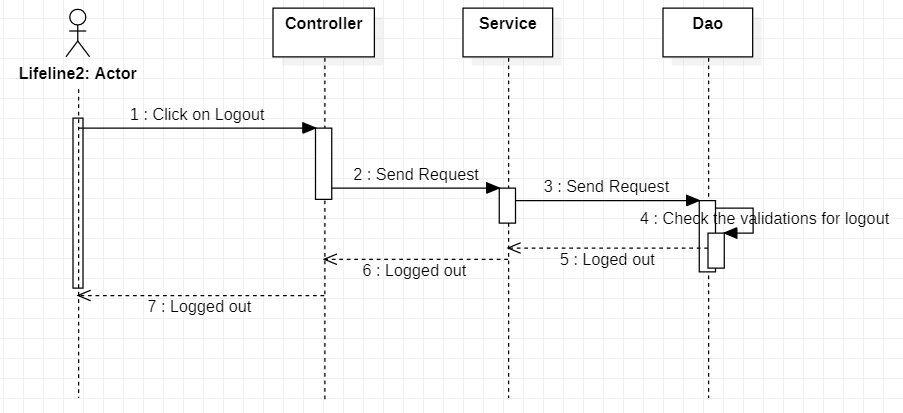
1. User Logout

This module has been designed to Log-Out an already logged-in actor.

***Pre-Requisite***:

User should already be logged-in.

Sequence Diagram for User Logout



* 1. Employee Management System

Overview

Employee management is the process of developing, operating, maintaining, and employee in an effective manner. And is used in every aspects of the system, the term is used in reference to the individual employee that needs assets on behalf other employees.

Prerequisite

Actor must be logged in as User to perform the add employee, view employee and check employee functionalities.

Non-Functional Requirement:

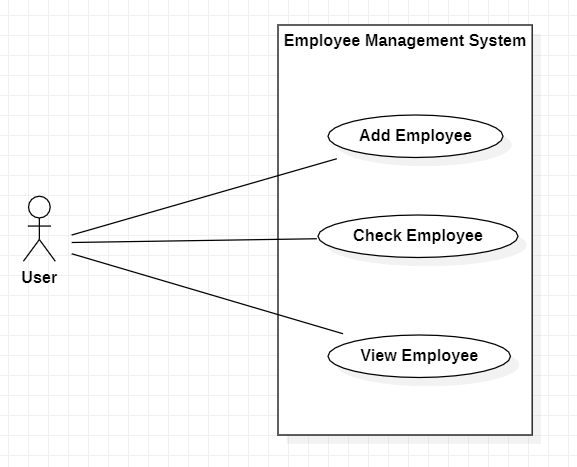
All modules are show designed that the result will be obtained within 10 seconds of request. Proper Loading indicator has been instantiated. Hence once a request been asked no one can further request anything. Authentication has been given a higher priority. All form validations are properly maintained.

Designed and Implemented By:

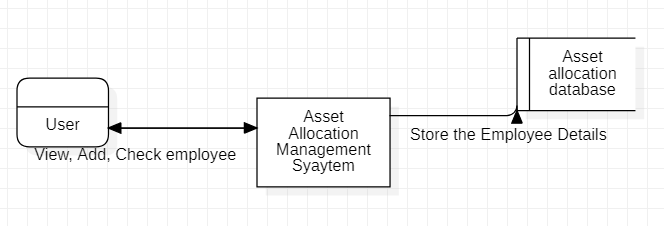
Sonali Deshmukh (Id: JEECloudISGH01)

Designation:

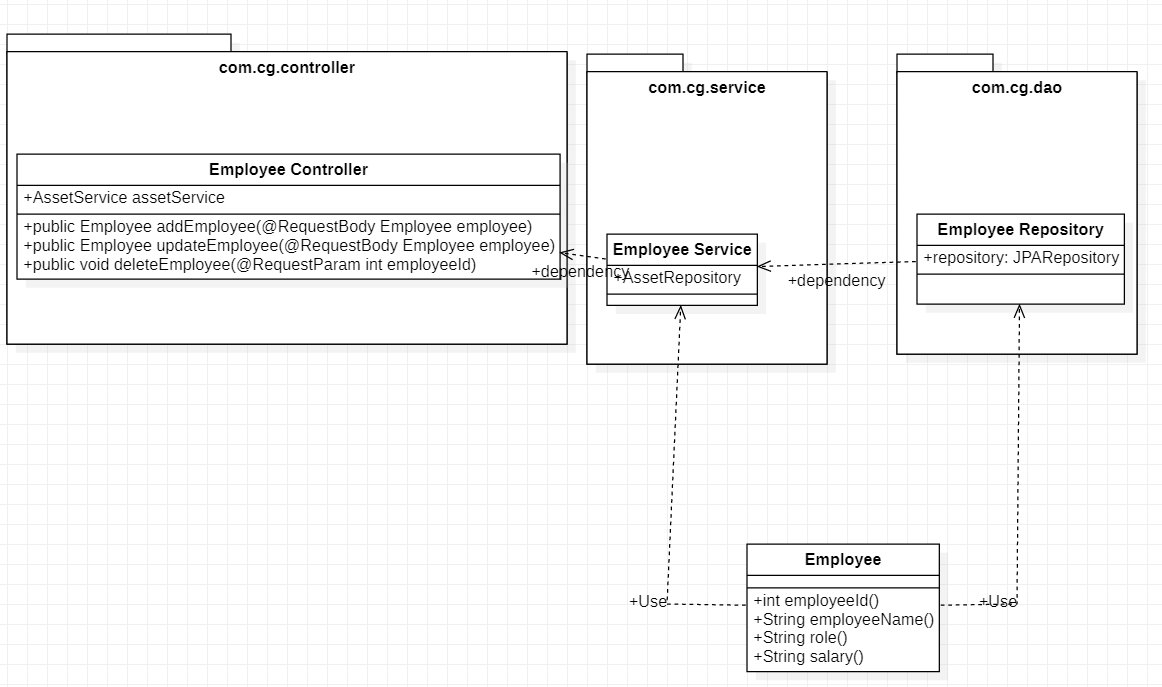
Use Case Diagram for Employee Management System



Employee Management Data Flow Diagram Level 0



Class Diagram for Employee Management



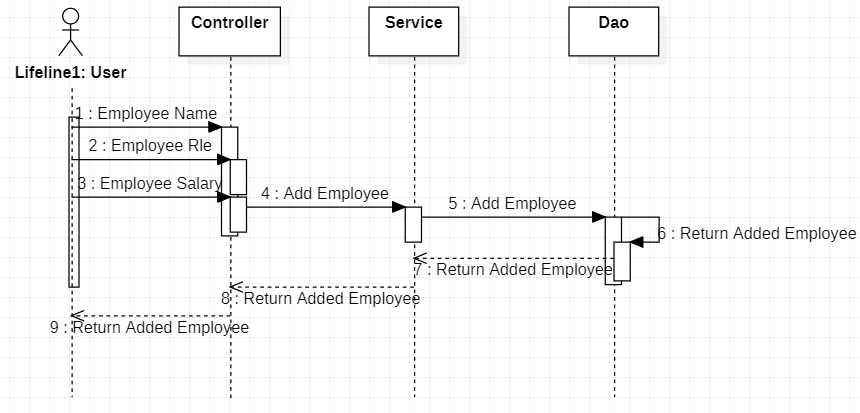
1. Add an Employee

This module has been designed to add an employee in the database. This can be only accessed by User/Manager. The form is designed with proper validation.

Validation:

All fields are required.

Sequence Diagram for Adding an Employee



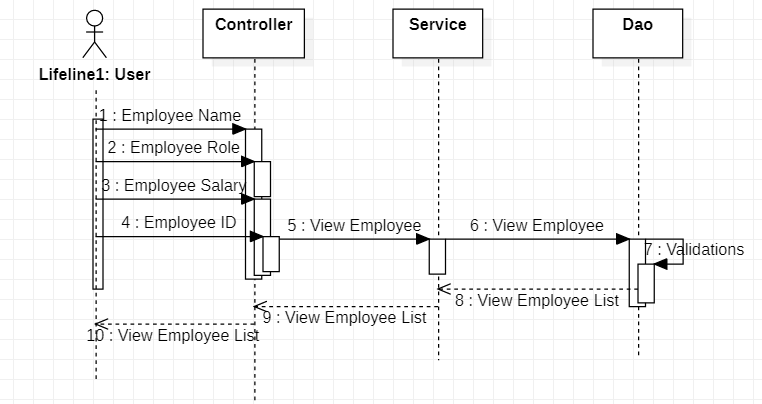
1. Check an Employee

The entire model has been designed in such a way that we can enter any keyword related to the employee to find the particular employee. Keyword such as employee ID, employee name. The entire functionality has been designed in Angular.

1. View an Employee

Employee Management System uses a view concept for the protection of authorizations. Therefore, the Employee view authorization object has a special function. The employee view applies especially to user/manager who have only occasional limited contacts with fixed assets. The employee view allows such employees by using theirs ID, name data whether or not they formally have access to every master record.

Sequence Diagram for Viewing an Employee



Project Progress Status (Employee Management System)

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint# | Task Assigned | Status | Remark |
| Sprint 1 | Creating UML Diagrams, Defining Test Cases and Sequence Diagrams, Implement the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence) | Modules Implemented using Java. Junit test cases are written and successfully tested.  Class Diagrams and Use case Diagrams are modified | Update the Class diagram, include the sequence diagram. Develop Knowledge about micro services architecture, Code Convention is not up to industry standard. Presentation layer is not implemented. |
| Sprint 2 | Spring Boot Implementation | Documentation Started | NA |

* 1. Asset Allocation Management System

Overview

Asset Allocation management is the process of operating and maintaining assets request in an effective manner. And is used by both admin and user/manager of the system, the term is used in reference the request to the individual employee that needs assets on behalf other employees.

Prerequisite

Actor must be logged in as User/Manager, Admin to change the status of request to allocated assets, unallocated assets, raise request and pending requests functionalities.

Non-Functional Requirement:

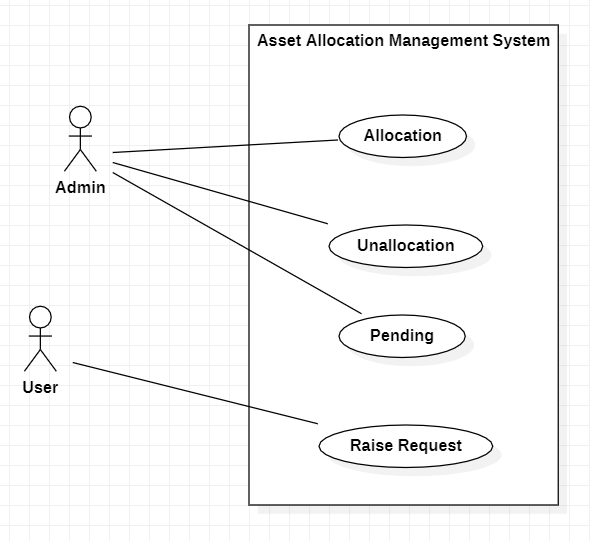
All modules are show designed that the result will be obtained within 10 seconds of request. Proper Loading indicator has been instantiated. Hence once a request been asked no one can further request anything. Authentication has been given a higher priority. All form validations are properly maintained.

Designed and Implemented By:

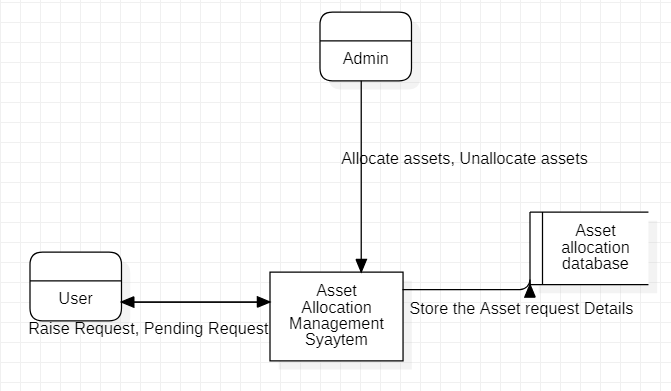
Aditya Sovani (Id: JEECloudISGH10)

Designation:

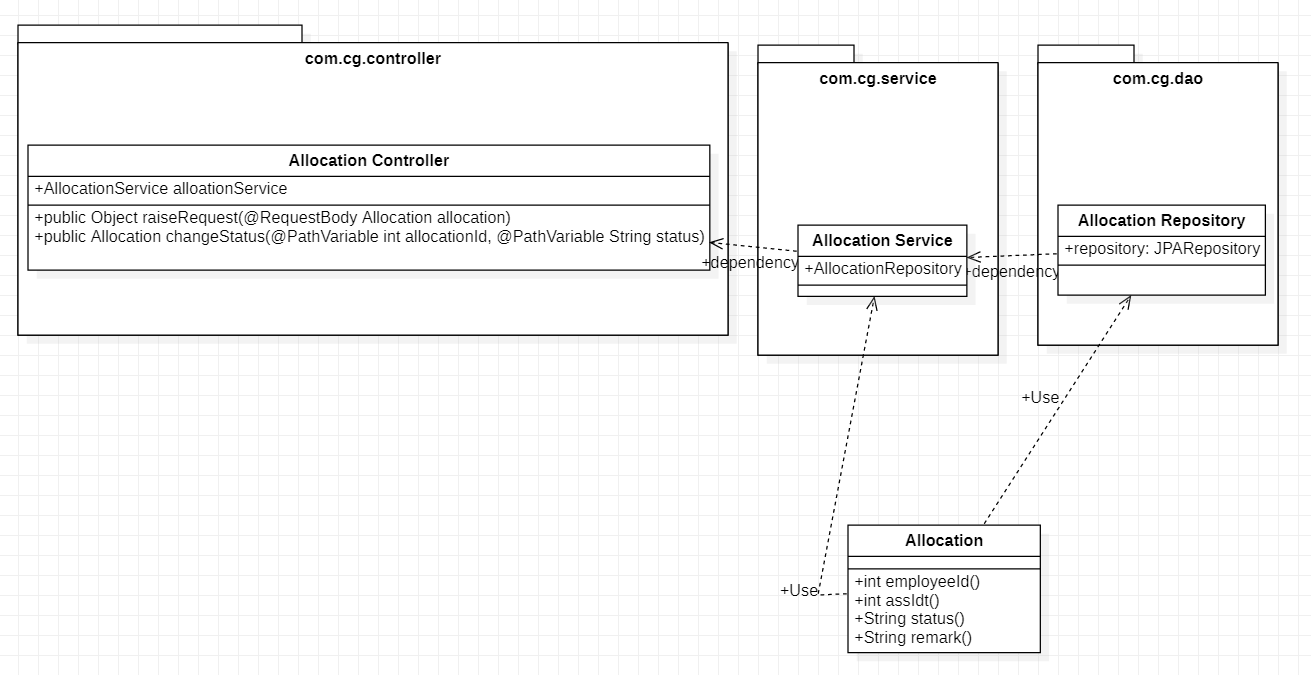
Use Case Diagram for Asset Allocation Management System



Asset Allocation Management Data Flow Diagram Level 0



Class Diagram for Asset Allocation Management



1. Allocated Assets

This module has been designed to allocate an asset in the database. This can be only accessed by Admin. The form is designed with proper validation.

Validation:

The asset should be useful for other employees if it is so then and then only it is allocated.

1. Unallocated Assets

This module has been designed to unallocate an asset in the database. This can be only accessed by Admin. The form is designed with proper validation.

Validation:

The asset should be useful for other employees if it is so then and then only it is allocated otherwise the admin won’t allow for the assets.

1. Pending Assets

This module has been designed to see the pending asset in the database. This can be only accessed by user. The form is designed with proper validation.

Validation:

The asset should be useful for other employees if it is so then and then only it is allocated otherwise the admin won’t allow for the assets, if admin change the mind and keep it for other stage then it goes in the pending assets request state.

Project Progress Status (Asset Management System)

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint# | Task Assigned | Status | Remark |
| Sprint 1 | Creating UML Diagrams, Defining Test Cases and Sequence Diagrams, Implement the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence) | Modules Implemented using Java. Junit test cases are written and successfully tested.  Class Diagrams and Use case Diagrams are modified | Update the Class diagram, include the sequence diagram. Develop Knowledge about micro services architecture, Code Convention is not up to industry standard. Presentation layer is not implemented. |
| Sprint 2 | Spring Boot Implementation | Documentation Started | NA |

* 1. Cloud Management

Front-end is hosted on Firebase. [url: https://asset-management-ui.web.app/]

Instead of running all services locally, we Hosted them on Heroku.

Database used on cloud: Postgress. (via heroku Postgress Addon)

Cloud CI/CD configuration using GitHub actions:

GitHub actions in GitHub’s in-built CI tool. It automatically builds and tests our application. And notifies in case of failure.

On Heroku, I enabled automatic deploys which triggers after every GitHub push. i.e. when I push any app changes to GitHub, after Passing tests of GitHub Actions, Heroku will build and deploy

application.

Thus, our app can run entirely in the cloud. And we can view it in browsers like any other web-app.

8. System Requirements

Below is a list of the minimum Hardware and Software requirements to access Asset Management website.

**Operating System:**

* Windows 7 and above.
* Mac OSX 10.8, 10.9, 10.10 or 10.11
* Android 3 and onwards.

**Hardware:**

* Processor (CPU) with 2 gigahertz (GHz) frequency or above
* A minimum of 4 GB of RAM
* Monitor Resolution 1024 X 768 or higher (For better view)
* A minimum of 5 GB of available space on the hard disk
* Internet Connection Broadband (high-speed) Internet connection with a speed of 2 Mbps or higher
* Keyboard and a Mouse or some other compatible pointing device

**Browsers:**

* Chrome\* 58+
* Microsoft Edge\* 20+
* Mozilla Firefox 40+
* Internet Explorer 11+ (Windows only)
* Safari 6+ (MacOS only)
* Android\* 3+

*\**Google Chrome version 42+ and Microsoft Edge do not support NPAPI-type plug-ins, including Java plug-ins and many media browser plug-in.

*Users using unsupported browsers may experience issues submitting forms, placing orders, purchasing, updating details and transaction management threads.*

**Browser Configuration:**

Your browser must be configured as follows:

* JavaScript must be enabled
* CORS must be configured properly
* Cookies must be enabled.

**Software:**

* Java — to view and interact with all available blackboard applications.
* STS(Spring) — STS workbench was used to run JDK (write, compile and run the code).
* Visual Studio Code — for writing codes for frontend using angular, VS Code was used as a workbench.
* Apache Tomcat — it was used as a server for hosting the website.