*Asset Management System*

*SPRINT-2*

Table of Contents

[1 Introduction 4](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528891)

[1.1 Setup Checklist for Mini Project 4](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528892)

[2 Problem Statement 5](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528894)

[2.1 Objective 5](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528895)

[2.2 Abstract of the project 5](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528896)

[2.3 Functional components of the project 5](file:///C:\Users\Sonali%20Deshmukh\Downloads\Sprint1-Miniproject\Sprint1-Miniproject\Airline%20Reservation%20System.doc#_Toc360528897)

# Introduction

This document outlines a case study for Sprint 2 project. The aim of this project is to develop an Asset Management System as integration of all independent microservices. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules.

## **Setup Checklist for Mini Project**

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7
* Memory: 32MB of RAM (64MB or more recommended)
* Internet Explorer 6.0 or higher
* STS
* Angular

**Software/Tool Requirement**

* JDK 8
* Nodejs
* IDE- STS (Spring Tool Suite)
* MAVEN
* Post-Man Master
* GitHub- Version Control System
* MySQL or H2 database
* IDE-Visual Studio Code

# Problem Statement

## **Objective**

Development of Asset Management System used for maintenance of an asset details in an organization.

## **Abstract of the project**

This project is aimed at developing Online Asset Management System. This is an Intranet 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.

## **Functional components of the project**

In this project, the independent functions are divided into microservices and description of those microservices is as follows:

**Asset Code Module**: Following is a list of functionalities of the system. There is an Admin who can add, delete, update, search and view the assets details.

There are one microservices named asset-service which contains all the CRUD operations which are to be perform in asset code module and one Database Provider Service to store the data.

**1.asset-service:**

* + - * 1. Add Assets:
* Asset Id is AUTO generated.
* Asset name contains the name with the combination of UPPER\_CASE letters and LOWER\_CASE alphabets. (not to be EMPTY)
* Add Asset also give status whether it is allocated to the employee or unallocated.
* Add Asset also give description about the assets that are available in companies.
  + - * 1. Update Assets:
* Update Asset ID is required to know which asset you have to update (not to be EMPTY)
* Update Asset requires Asset Name to update.
* Update asset needs the status whether it is change to allocated or unallocated.
* Update asset requires the description also for updating asset.
  + - * 1. Delete Assets:
* Delete Asset will be soft delete the assets form the database.
  + - * 1. Search Assets:
* Search Asset gives the result based on ASSET NAME.
  + - * 1. View Assets:
* View Asset is to display the Asset Id, Asset Name, Description, Status, Action fields on the UI.

**Login Code Module**: Functionality of this module is to identify the entered login credentials is of ADMIN or USER/ MANAGER. If the entered details are wrong it gives the ERROR of enter the proper values.

**Employee Code Module**: Following is a list of functionalities of the system. There is a user/manager who can add, check and view the employee details.

There are one microservice named employee-service which contains all the CRUD operations which are to be perform in employee code module and one Database Provider Service to store the data.

**1.employee-service:**

(a) Add Employee:

* Employee Name contains the combination of UPPER\_CASE, LOWER\_CASE alphabets (not to be EMPTY).
* Employee Role contains the combination of UPPER\_CASE, LOWER\_CASE alphabets (not to be EMPTY).
* Employee Salary contains the combination of UPPER\_CASE, LOWER\_CASE alphabets and numbers (not to be EMPTY).

(b) Check Employee:

* This method contains two ways to check employee:

By ID:

* After giving employee ID the exist employee name should be visible.
* If that employee ID doesn’t exist then it gives the ERROR.

By Name:

* After giving employee name the existing employee name should be visible.
* If that employee name doesn’t exist then it gives ERROR.

(c) View Employee:

* View Employee is to display the Employee Id, Employee Name, Employee Role, Employee Salary, Action fields on the UI.

**Asset Allocation Code Module**: Following is a list of functionalities of the system. There is an Admin who can allocate or unallocate the assets to the employee.

There are one microservice named allocation-service which contains all the operations which are to be perform in asset allocation code module and one Database Provider Service to store the data.

**1.allocation-service:**

(a) Allocated:

* When Admin approves the request raised by the employee for assets the status change to allocated.

(b) Unallocated:

* When Admin decline the request raised by the employee for assets the status remains same as unallocated.

(c) Raise Request:

* When User/Manager want any kind of assets from the company they raise request for that asset and then that request seen by the admin and admin decide whether to approve that request or not.