# *Software Engineering*
# *Software Requirements Specification (SRS) Document*

*Amit G. Patil*
*16/02/2019*

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|
| Final Draft | Amit G. Patil | All sections being Filled | 16/02/19 |

# Contents

# 1. Introduction

## 1.1 Introduction

The purpose of this document is to define and describe the requirements of the project and to spell out the system's functionality and its constraints.

## 1.2 Scope of this Document

A web based asset management system having three users namely Admin, Manager and Employee. The system requests the users to log in to use the system. After authentication, the system allows the user to do some operations. The system allows the Admin to add, edit and delete the assets, manager and employees in the database (DB). Admin also need to allocate the assets to manager and employees. The manager can view the allocated assets to his team and his own assets too. Employees can view allocated assets information.

The system restricts the users from accessing all the above mentioned features depending on the user's level of permissions.

## 1.3 Overview

This document is written according to the standards for Software Requirements Specifications explained in "IEEE Recommended Practice for Software Requirements Specifications".

This document, in Section 2, states the general descriptions which includes product functions, user characteristics, problem statement and objectives. The Functional requirements in Section 3, gives the details of the requirements for storing and accessing data from database. Remaining sections includes Performance requirements, other non- functional requirements, Operational scenario, Use case models, schedule followed by appendices.

# 2. General Description

## 2.1 Product Functions

The system performs the following functions. The functions depend on the user's level and permission package, as explained in the user characteristics.

### 2.2.1 Login

This function allows the user to enter into the application. The user is required to provide username and password. After authentication user will have access to main menu. Availability of menu functions depends on user's level.

### 2.2.2 View/add/edit/delete records

This function allows the admin to view/add/edit/delete records in the appropriate categories of the DB. System has four main databases: "Employees", "Assets", "Manager", and "Assets_Allocation".

### 2.2.3 Create new type of asset

This function allows the admin with to create a new type of asset.

### 2.2.4 Assign asset/employee to employee/manager

This function allows the user with appropriate permissions to assign:

- asset(s) to employee
- employee(s) to manager

### 2.2.5 Add new request

This function can be done by employee or manger. The users can request a specific assets which they are needed. After submission, the requests will forwarded to admin for approval/rejection.

### 2.2.6 View list of all requests in the system

This function allows the admin to browse list of all requests existing in the system.

### 2.2.7 Approve/reject request

This function allows the admin to approve/reject any submitted request. He/she can either approve or reject the request. While approving the request system will check whether sufficient number of assets are available or not. If not available then the request is decline by the system. Admin when rejects request shall provide reason for rejection.

### 2.2.8 View "My profile"

This function allows all the users to view list of all assets assigned to him/her, and own details with name of manager.

### 2.2.9 View "Team"

This function allows the manager to view list of all team members with assets assigned to them.

### 2.2.10 View "Allocated Assets"

This function allows all the admin to view list of all assets assigned to which employee.

### 2.2.11 Search

This function enables the admin to perform search operation with admin types text to search for specific assets or employee.

### 2.2.12 Logout

This function can be done by all users. It terminates the user session. The system can also do this function automatically if the session is left unused for half an hour.

## 2.2 User Characteristics

The users include the employees, manager and admin. For this system, the user is required to know the basic usability of a very base level understanding of access, which hopefully will be facilitated by the software team through training.

## 2.3 User Problem Statement

The users system, currently, is slow and inefficient as it relates to the manual asset allocation. Employee may wait hours to get the asset they requested.

## 2.4 User Objectives

The user wants a database that will store information of asset allocated to employees. The program must faciliatate the speed and ease of input.

## 2.5 General Constraints

Constraints include an easy to use interface for the program through forms, a Windows platform. Our system is implemented in Java. To install and execute the asset management system, JVM is required.

# 3. Functional Requirements

1. **Assets/Employee records shall be stored in the Access Database.**
    1. Assets/Employee shall be stored on the database and have complete fields.
    2. Very high criticality
    3. This requirement is the basis of the project; all other aspects depend on it.
2. **The items shall be accessible via queries.**
    1. Users of the database should be able to run queries on the data that has been put into the database.
    2. Very high criticality
    3. We do not foresee any technical issues preventing the implementation of this.
    4. Given the capabilities of Access, this requirement is able to be satisfied.
    5. This requirement depends on requirement number one.
3. **The data stored should be able to be manipulated through forms.**
    1. Items and other data should be able to be added and updated through the use of forms.
    2. Very high criticality
    3. We do not foresee any technical risks involved in this requirement.
    4. The only factor we can encounter here is the user of the system not being able to use it correctly. We will overcome this by training those who will be using it.
    5. This requirement is dependent on requirement one.

# 4. Interface Requirements

## 4.1 User Interfaces

- **4.1.1 GUI**
  The user interface for this program is the interface provided by our system. It includes forms for the users to query and organize data to suit their needs. The UI should be easy to manipulate without any additional training.
- **4.1.2 CLI**
  There is no command line interface
- **4.1.3 API**
  There is no API for the product
- **4.1.4 Diagnostics or ROM**
  There is a troubleshooting and help section provided.

### 4.2 Hardware Interfaces

The system uses the hard disk. Access to the hard drive and other hardware is managed by the operating system.

### 4.3 Communications Interfaces

The web based UI is the only means of communication between user and server. The system accessible to all popular web browsers. If we decide to implement an Ad Hoc network for a shared database, the operating system will handle those connections.

### 4.4 Software Interfaces

Our system used to import and export data with MySQL. This functionality is built in to the user interface.

## 5. Performance Requirements

The database is designed to be operated through MySQL, thus no additional system requirements exist beyond those required to run MySQL, except for a negligible amount of hard drive space to store the database.

Microsoft lists the requirements for MySQL as follows:
500 MHz processor or higher
256MB RAM or higher
1.5GB Available Hard Drive Space
Windows XP SP2 or later operating system.
MySQL 5.5

## 6. Other non-functional attributes

### 6.1 Security

The system shall be designed with a level of security appropriate for the sensitivity of information enclosed in the database. More interaction is needed with client about the volatility of the information. Since there is no obvious information that is of a high security level such as credit card information, the only requirements that could be implemented are encrypting the database and/or making the database password-protected, by user's request.

### 6.2 Binary Compatibility

This system will be compatible with any computer that has MySQL 5.5 or later installed, and will be designed with more than one computer in mind.

### 6.3 Reliability

Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested to ensure reliability.

### 6.4 Maintainability

The document should be easy for the users who execute the system day to day, for the developers who wish to edit or develop further, and for the personnel who is in charge of the maintenance.

### 6.5 Portability

The system should support new versions of the related browsers. The administrative and server technologies should be standard and supported by most platforms.

### 6.6 Extensibility

The system shall be designed and documented in such a way that anybody with an understanding of Java shall be able to extend the system to fit their needs with the team's basic instructions.

### 6.7 Reusability

The system should be designed in a way that allows the database to be re-used regularly for the various asset requested generated by employees.

### 6.8 Application Affinity/Compatibility

This system requires the MySQL 5.5 or later and Java 7.

### 6.9 Resource Utilization

The resources used in the creation of this system include the server space and the internet or intranet when used in ad-hoc network.

### 6.10 Serviceability

The maintenance of the system should be able to be sufficiently performed by any person with a basic understanding of Java and MySQL.

# 7. Operational Scenarios

## Scenario A: Initial Item Definitions

The admin shall enter the information about the items into the database for its initial construction and evolution. The fields will be completed via a form that will manipulate the data.

## Scenario B: Database Maintenance

The admin may want to alter/delete information after the project is over, In this case they will need to be able to remove the data that has been entered.

# 8. Preliminary Use Case Models and Sequence Diagrams

This section presents a list of the fundamental sequence diagrams and use cases that satisfy the system's requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

## 8.1 Use Case Model

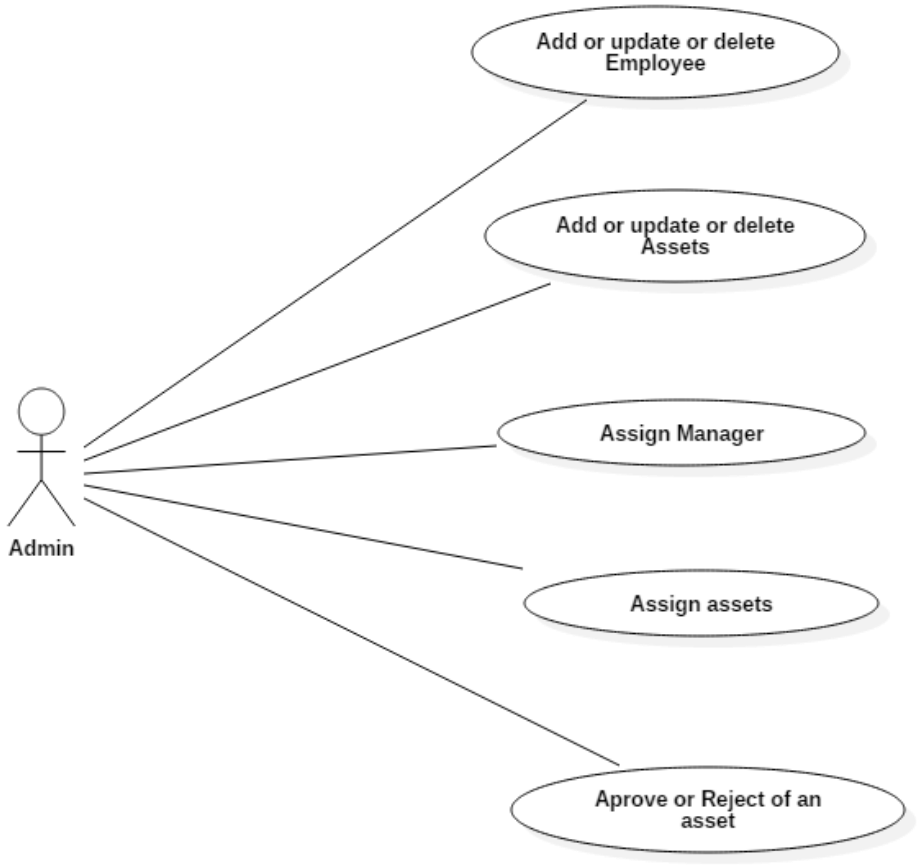


Figure 8.1: Use Case Diagram for Admin

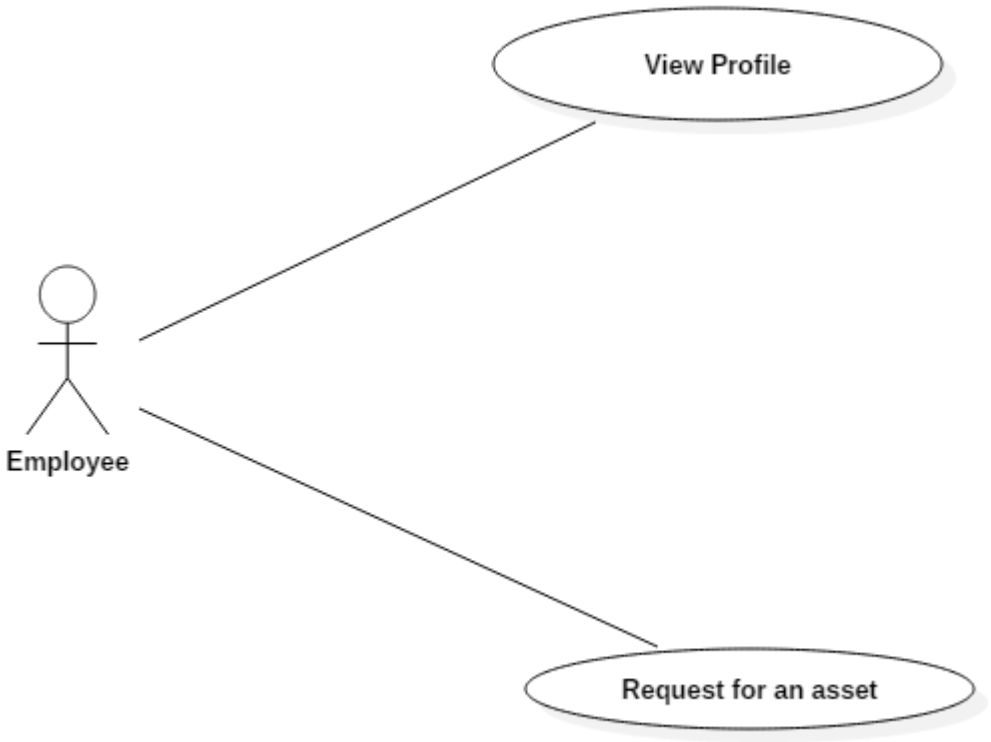


Figure 8.2: Use Case Diagram for Employee

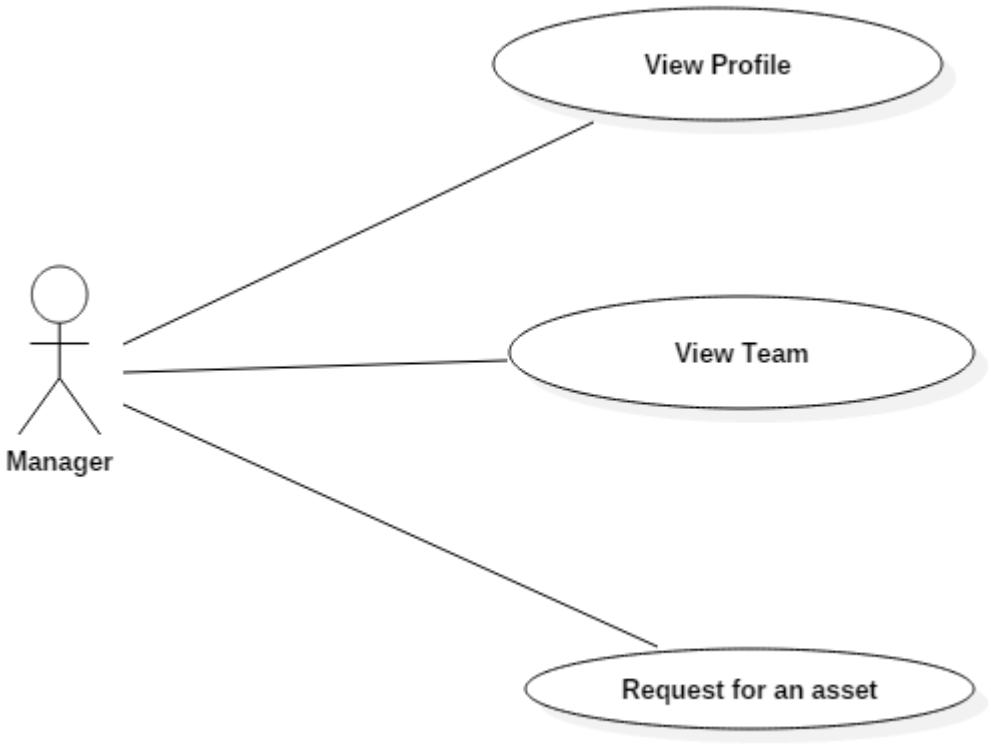


Figure 8.3: Use Case Diagram for Manager

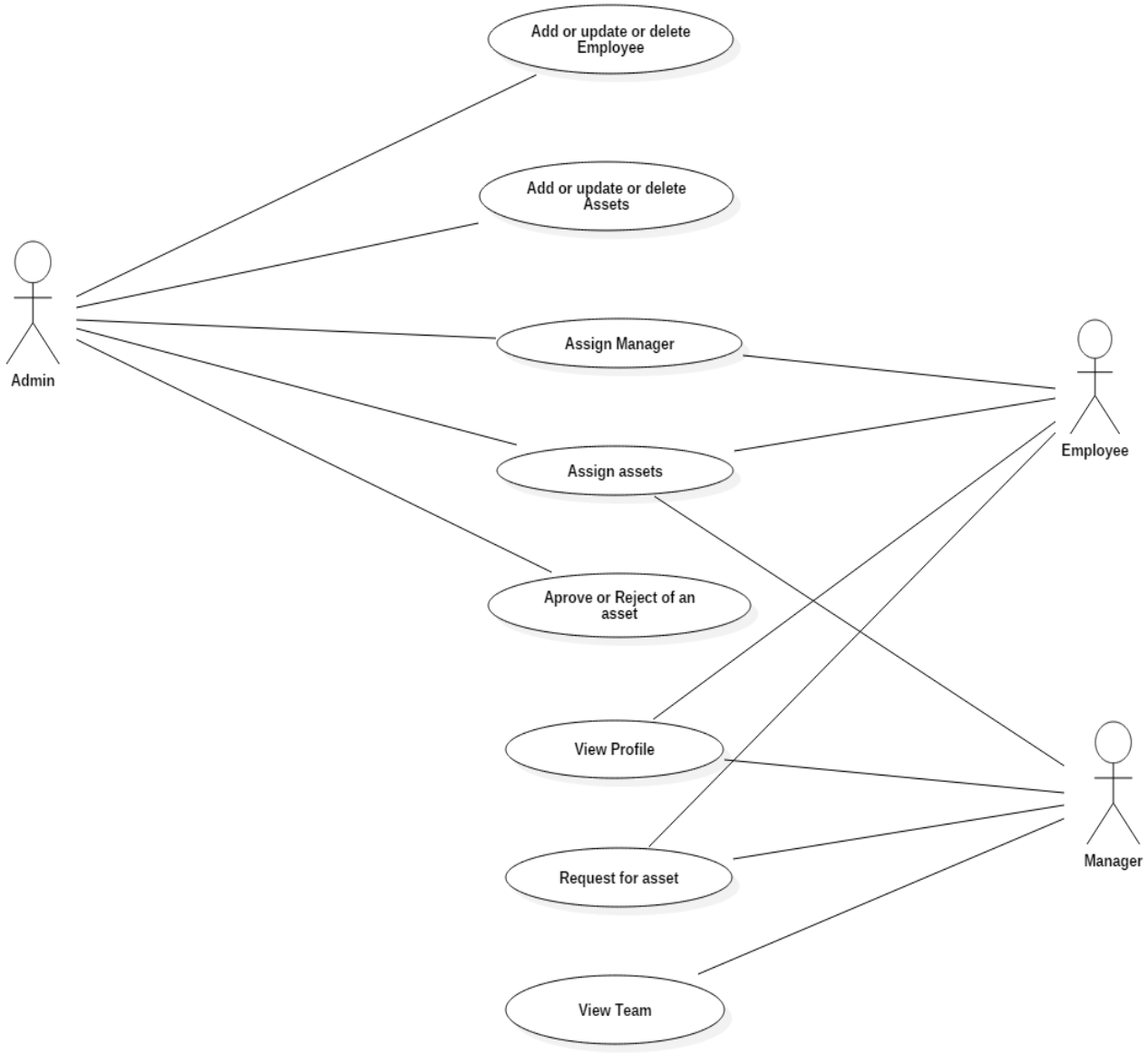


Figure 8.4: Use Case Diagram for Asset Management System

## 8.2 Sequence Diagrams

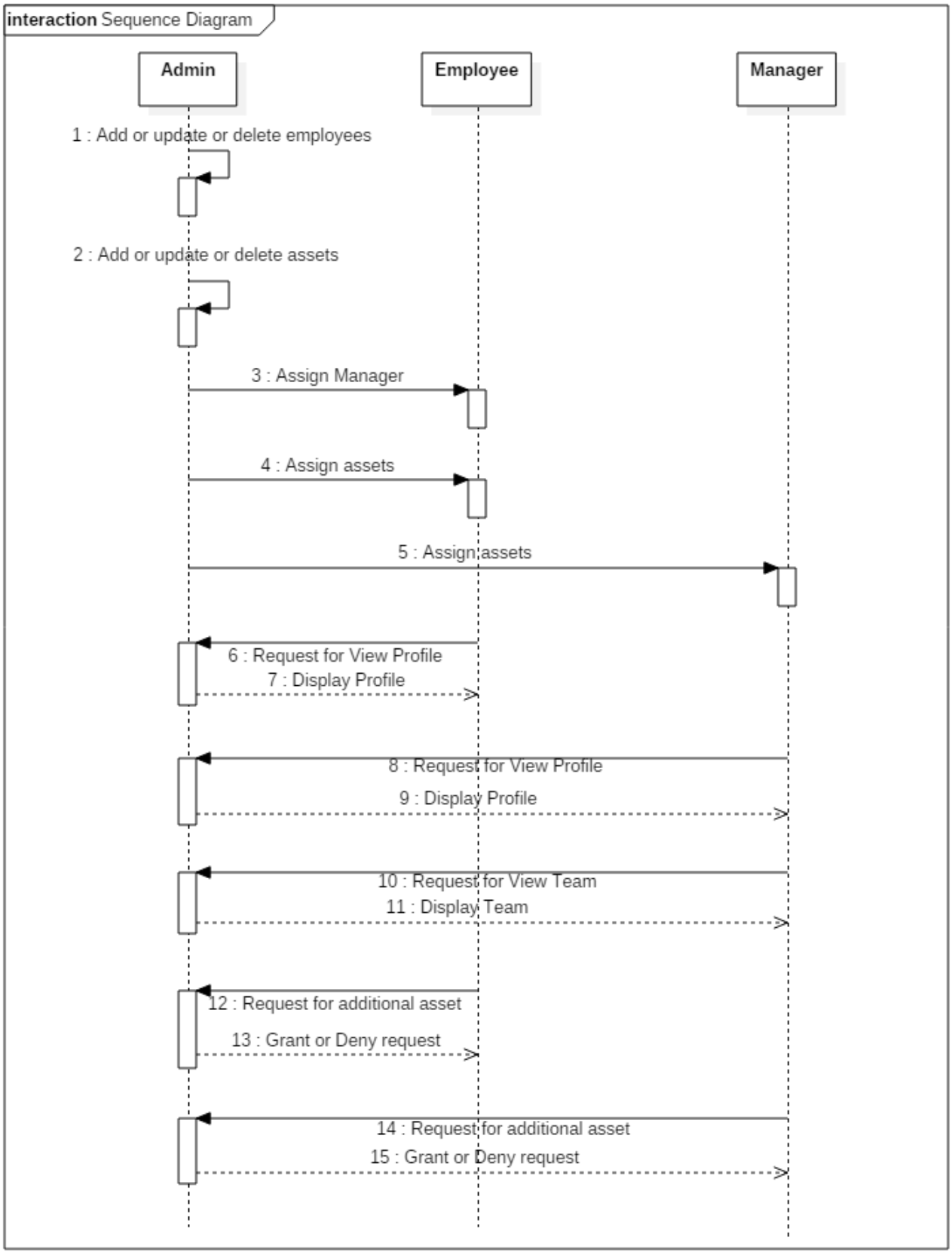


Figure 8.5: Sequence Diagram for Asset Management System

## 8.3 Class Diagrams

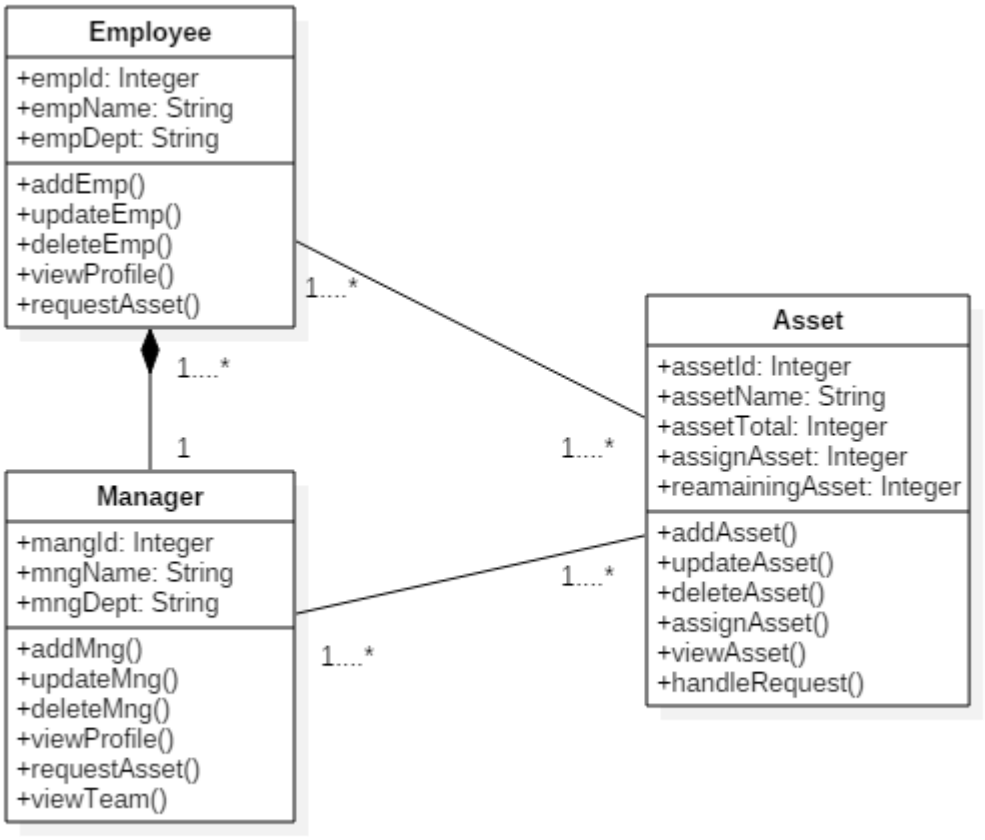


Figure 8.6: Class Diagram for Asset Management System

# 9. Database Schema

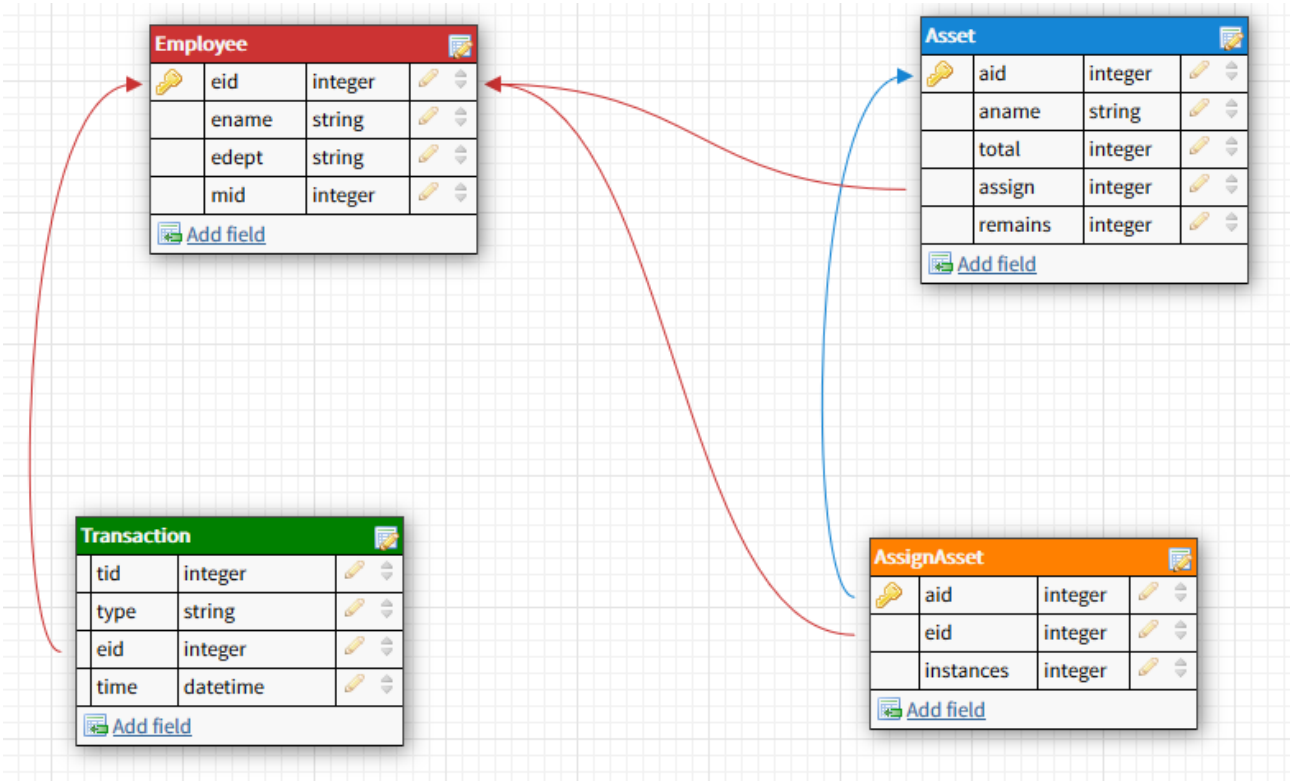


Figure 9.1: Database Schema for Asset Management System