**Expense Management System**



|  |
| --- |
| **Team Members** |
| **Rajeshwari Bhirud(Scrum Master)** |
| **Prakash Devar** |
| **Swapnali Jadhav** |
| **Deepali Jain** |
|  |

**Table of Contents**

|  |  |
| --- | --- |
| 1. | Introduction |
| 2. | Overview |
| 3. | Epic and User Stories |
| 4. | UML Diagrams |
| 5. | System Requirements |

1. Introduction

This document outlines a case study for Sprint 2 project. The project is to develop an Expense 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.

2. Overview

Expense Management system has following functional requirements:

1. Employee Management
2. Expense Management
3. Project Management
4. Expense Claim Management

The model followed was an agile model. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

The project had to go under 2 sprints. First sprint had design implementation by core java and collections implementations. The assumed data was made static by using collections to enhance and have a practical knowledge on the collections framework. In second sprint we had to convert our monolithic architecture to microservice architecture. In this project we are following CQRS pattern. Implementation of sprint to is with spring with rest services along with Logger implementation and h2 database the front end of this project is designed in AngularJS. Some of DevOps tools are also used such as Jenkins and SonarQube. Herein the entire application is ready to serve a shopping site with the major functionalities.

3. User and Epic Stories

In a sense, stories and epics in agile are similar to stories and epics in film or literature. A story is one simple narrative; a series of related and interdependent stories makes up an epic. The same is true for your work management, where the completion of related stories leads to the completion of an epic. The stories tell the arc of the work completed while the epic shares a high-level view of the unifying objective.

* [**Epics**](https://www.atlassian.com/agile/project-management/epics) are large bodies of work that can be broken down into a number of smaller tasks (called stories).
* [**Stories**](https://www.atlassian.com/agile/project-management/user-stories), also called “user stories,” are short requirements or requests written from the perspective of an end user.

Table 1: User and Epic Stories of Expense Management System

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that… |
| Login | Registration | Admin, Project manager , Expense manager, Employee | Sign up as new user | A new user will be registered |
| Update | Admin, Project manager , Expense manager,  Employee | Change password | A new password will be generated |
| Validate | Admin, Project manager , Expense manager,  Employee | Validate the entered username and password | Check the entered password and username is correct or not |
|  |  |  |  |  |
| Employee Code Module | Add Employee | Admin | Enter employee details | A new Employee can be added to the database |
| Search Employee | Admin | Search employee by employee code | A particular employee can be viewed from employee database |
| Delete Employee | Admin | Delete the employee by employee code | Employee will be deleted from employee database |
| Modify Employee | Admin | Modify employee details | Employee details of selected employee will be modified |
| View all the employees | Admin | View all the employees from database | All the employees can be viewed from employee database |
|  |  |  |  |  |
| Project Code Module | Add Project | Project manager | Enter Project details | A new Project can be added to the project database |
| Search Project | Project manager | Search project by project code | A particular project can be viewed from project database |
| View all projects | Project manager | View all the projects from database | All the projects can be viewed from project database |
| Delete project | Project manager | Delete project by project code | Project will be deleted from project database |
| Update project | Project manager | Update the project | Project details of selected project will be modified |
|  |  |  |  |  |
| Expense Code module | Add Expense | Expense manager | Add the expense details | A new Expense can be added to the database |
| Search Expense | Expense manager | Search an expense by expense code | Easy to find that particular expense from long list of expenses |
| View all the expenses | Expense manager | View all the expenses from database | All the expenses can be viewed |
| Delete an expense | Expense manager | Delete the expense by expense code | Expense will be deleted from Expense database |
| Update an expense | Expense manager | Update an expense | Particular expense will be updated from database |
|  |  |  |  |  |
| Expense claim code Module | Claim an expense | Employee | Add the details of expense to claim | The details of expense to be claim will be added to the database |
| View all the expense claims | Employee | View all the expense claims from database | All the expense claims can be viewed from expense claim database |
| Search Expense Claim | Employee | Search an expense claim by expense claim id | Only particular expense claim will be viewed |
| Delete an expense claim | Employee | Delete the expense claim by expense claim id | Expense claim will be deleted |
| Update an expense claim | Employee | Update an expense claim | Selected expense claim can be modified |
|  | View all projects | Employee | View all projects from database | All projects can be viewed from project database |
|  | View all expense | Employee | View all Expense from database | All expense can be viewed from expense database |

4. UML Diagrams

1. **Pre-requisites**

User must be logged in as an Admin (Project Manager, Admin or Expense Manager) to perform add, update, view all, search or delete employee, projects and expenses

User must be logged in as to Employee to perform claim, update or delete expense claims.

1. **Non-functional requirement**

Proper form validations are maintained. To make it more user friendly proper alert messages are shown wherever required. Any changes in database will only be performed if user gives a confirmation to those pop up warnings.

1. **Employee management**

Use case diagram:

Following is the use case diagram of employee management indicating the cases that are to perform by the admin.

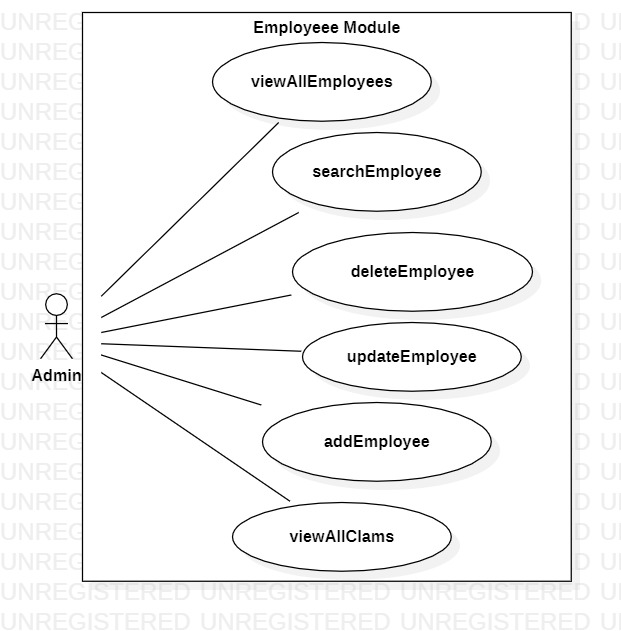


Figure 1: Use Case for Employee Management

Class Diagrams:

Following is the class diagram of employee management:

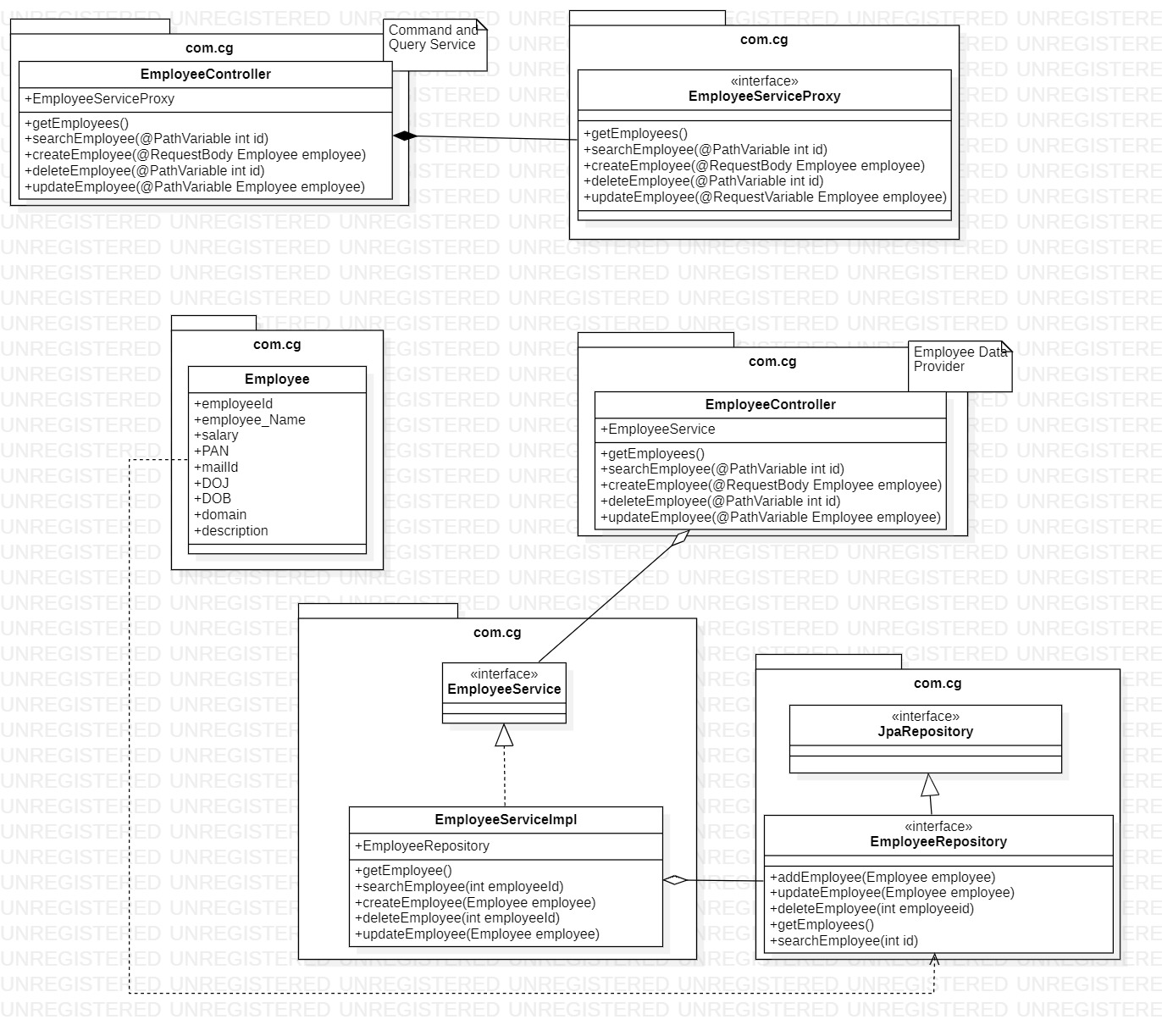


Figure 2: Class Diagram for Employee Management

1. **Project management**

Use Case diagram:

Following is the use case diagram of project management indicating the cases that are to performed by the project manager.

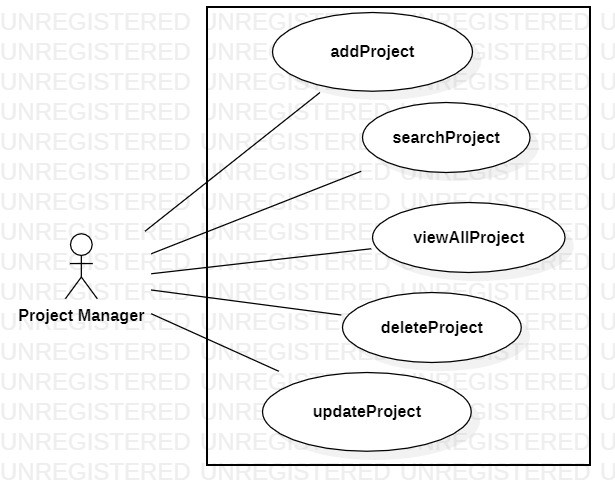


Figure 3: Use Case Diagram for Project Management

Class Diagrams:

Following is the class diagram of project management:

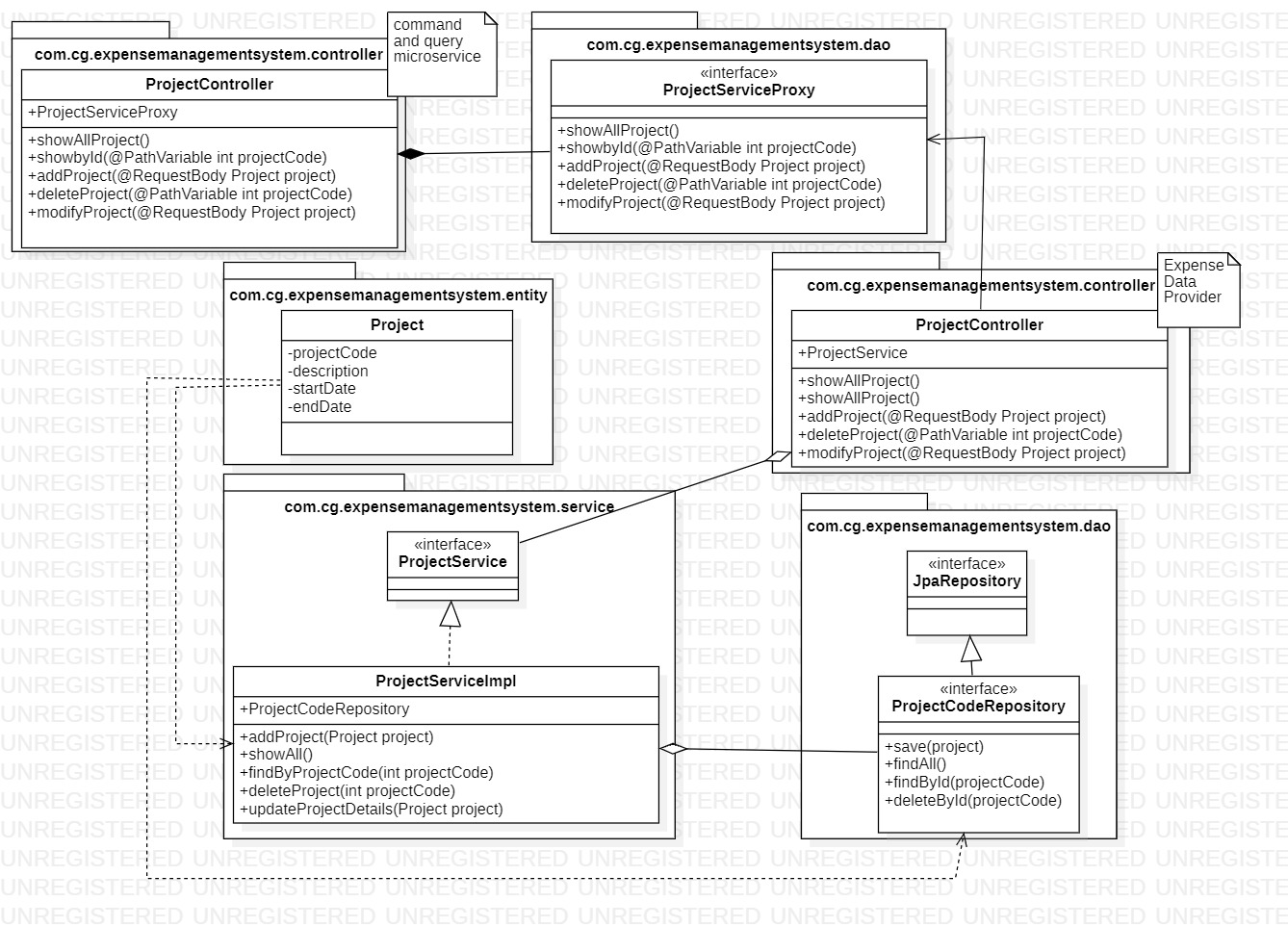


Figure 4: Class Diagram for Project Management

1. **Expense Management**

Use Case Diagram:

Following is the use case diagram of expense management indicating the cases that are to performed by the expense manager.

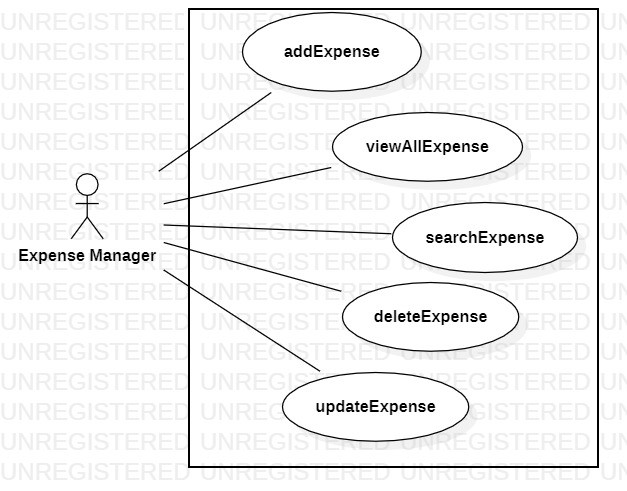


Figure 5: Use Case Diagram for Expense Management

Class Diagrams:

Following is the class diagram of expense management:

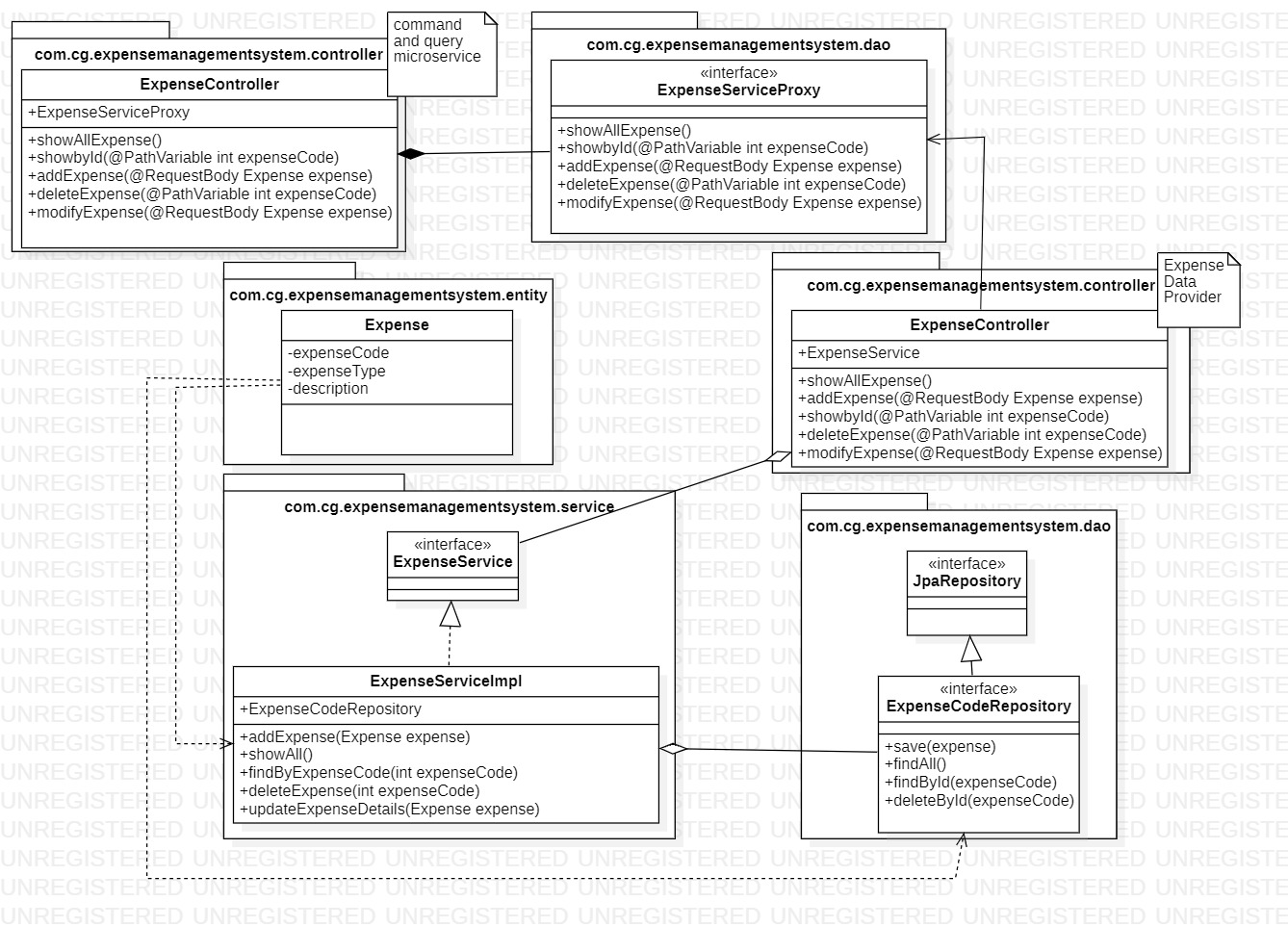


Figure 6: Class Diagram for Expense Management System

1. **Expense Claim management**

Use Case Diagram:

Following is the use case diagram of expense claim management indicating the cases that are to performed by the employee.

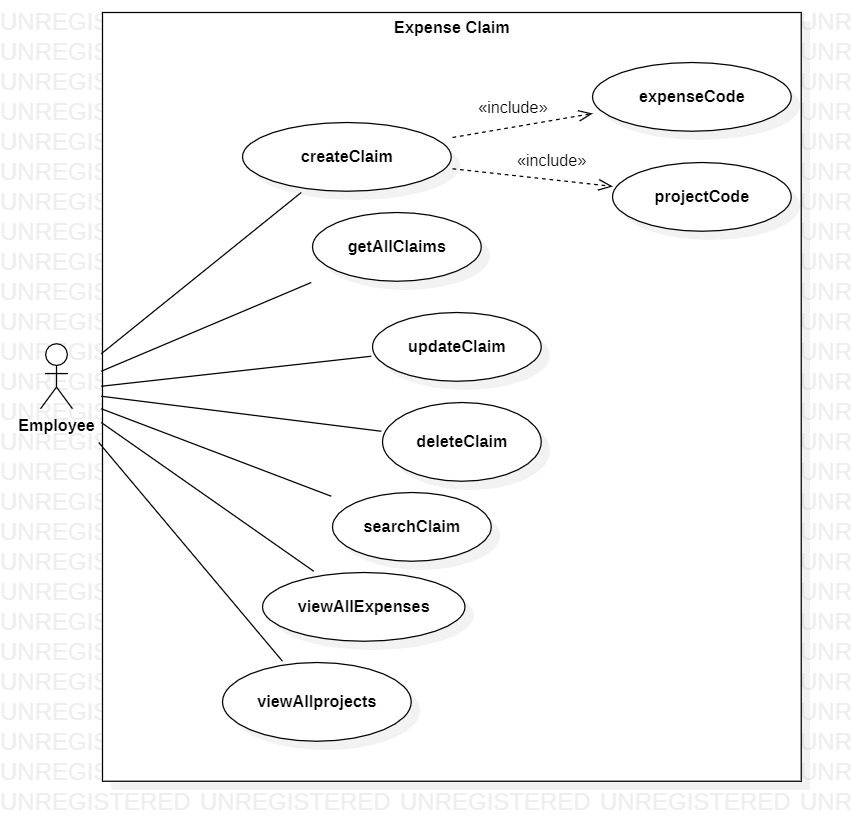


Figure 7: Use Case for Expense Claim Management

Class Diagrams:

Following is the class diagram of expense claim management:

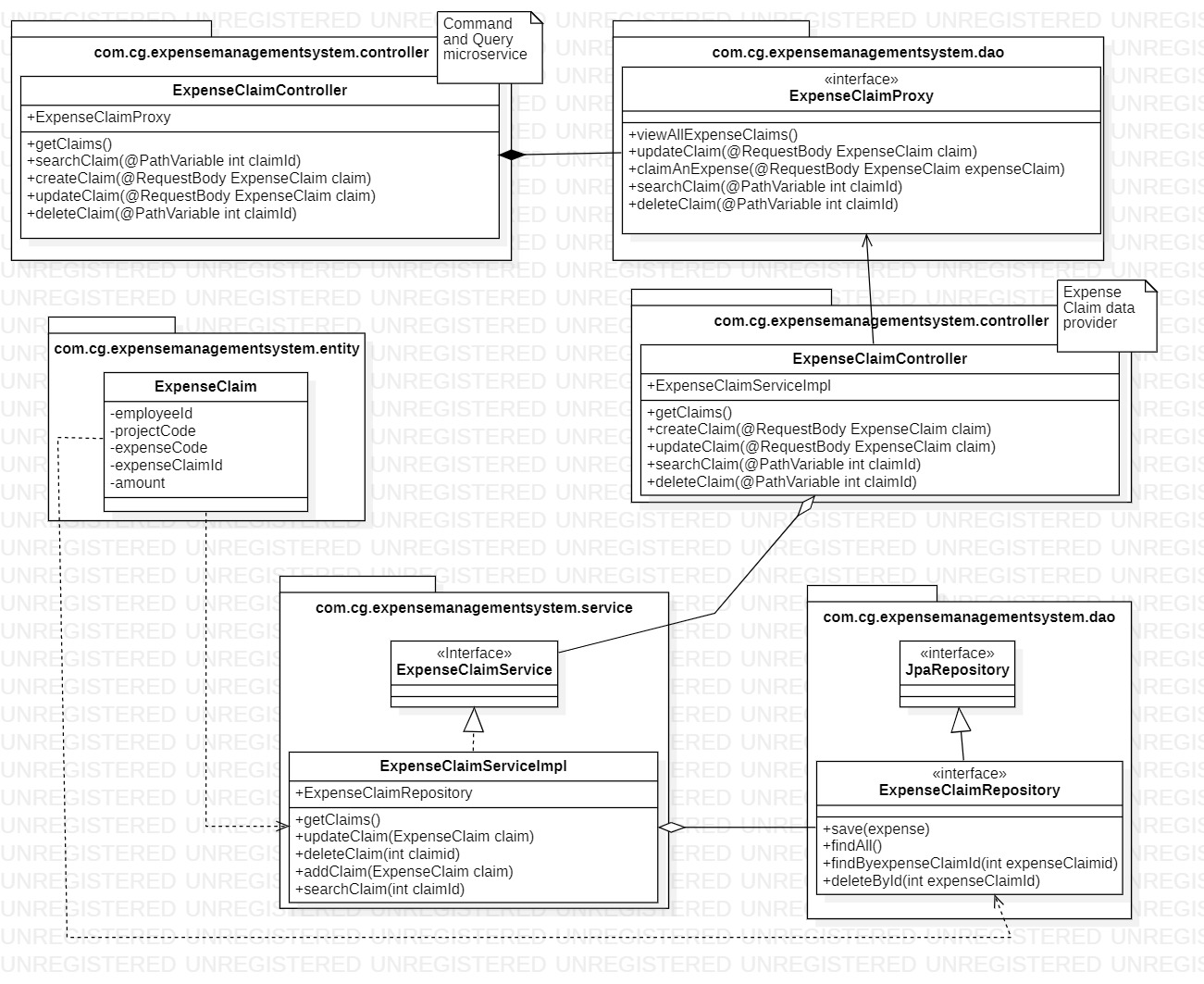
****

Figure 8: Class Diagram for Expense Claim Management

5. System Requirements

1. **Minimum System Requirement**

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows 7 and above
* Memory: 8GB of RAM (8GB or more recommended)
* Internet Explorer 11.0 or higher or Chrome 45 or above

1. **Software/Tool Requirement**

* JDK 8
* IDE-STS(Spring Tool Suite)
* MAVEN
* PostMan Master
* GitHub- Version Control System
* MySql or H2 database
* IDE-Visual Studio Code