**System Design Document**

**For**

**Resource Allocation System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| **Authors** | | Sadhna MJS  Dinesh Chandradasa | **Date** | 17.11.2014 | |
| **Organization** | | SimCentric Technologies (pvt) Ltd | **Reported to** | SusanthaAndradi | |

Contents

[**1.** **Introduction** 3](#_Toc403635243)

[1.1. **Purpose** 3](#_Toc403635244)

[1.2. **Scope** 3](#_Toc403635245)

[**2.** **Use Cases** 3](#_Toc403635246)

[2.1. **Actors** 3](#_Toc403635247)

[2.2. **Use Cases** 4](#_Toc403635248)

[2.3. **Use Case Elaboration** 5](#_Toc403635249)

[**3.** **Design Overview** 8](#_Toc403635250)

[3.1. **System Architecture** 8](#_Toc403635251)

[3.2. **System Interfaces** 8](#_Toc403635252)

[**4.** **Data Design** 9](#_Toc403635253)

[4.1. **Entity Relationship Diagram** 9](#_Toc403635254)

[**5.** **Dynamic Models** 9](#_Toc403635255)

[5.1. **Sequence diagram** 9](#_Toc403635256)

[5.2. **Data flow Diagram** 11](#_Toc403635257)

[**6.** **Technologies** 12](#_Toc403635258)

[6.1. **Technologies going to use to create this System** 12](#_Toc403635259)

[**7.** **References** 12](#_Toc403635260)

1. **Introduction**
   1. **Purpose**

The main objective of this document is to provide an architectural overview and conceptual design of “Resource Allocation System”. This document presents different types of abstraction it aims to provide stakeholders a clear understanding of the system.

* 1. **Scope**

The scope is to define high level design and technology decisions of the “Resource Allocation System”.

1. **Use Cases**
   1. **Actors**
      1. Employees

Employees are the lowest level of users. They can add their basic profile information and edit profile details.This category consist with two types,

* Developers
* Quality Assurance Engineers
  + 1. Line Mangers/Product Owners

Current Product owners (Line managers) are the middle level of management. They have the capability of adding employee to the pool, getting employee from the pool and lock employees.

* + 1. Higher Management

Higher Management is the highest level user. They can view reports, add data to the system, add project, add employee to the pool and confirm or reject the line managers’ requests.

* 1. **Use Cases**

The following figures display the use cases of “Resource Allocation System”.

These diagrams emphasize what system functions performed by which actor.

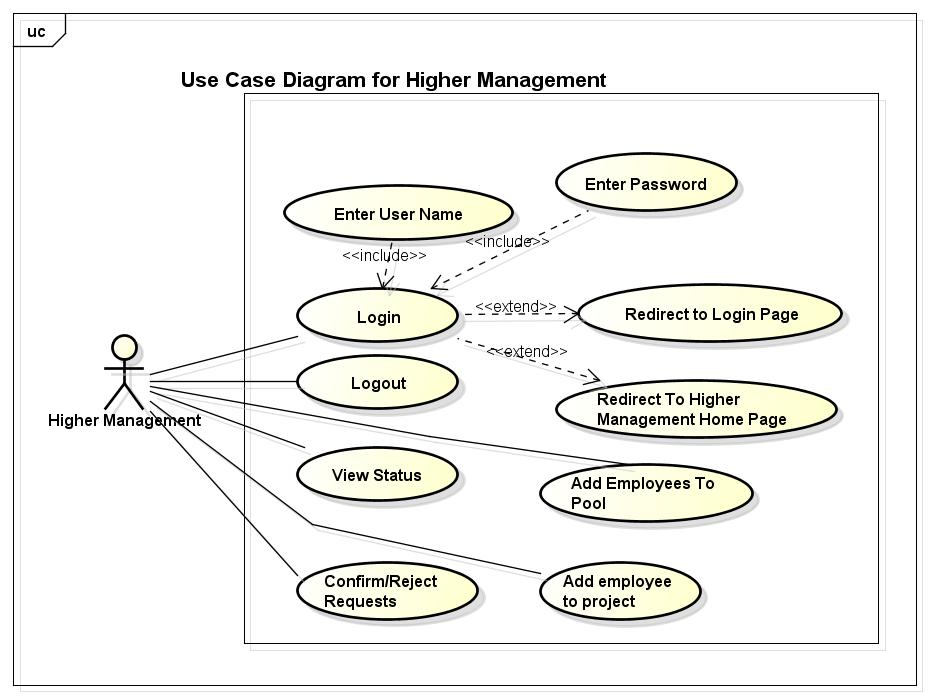
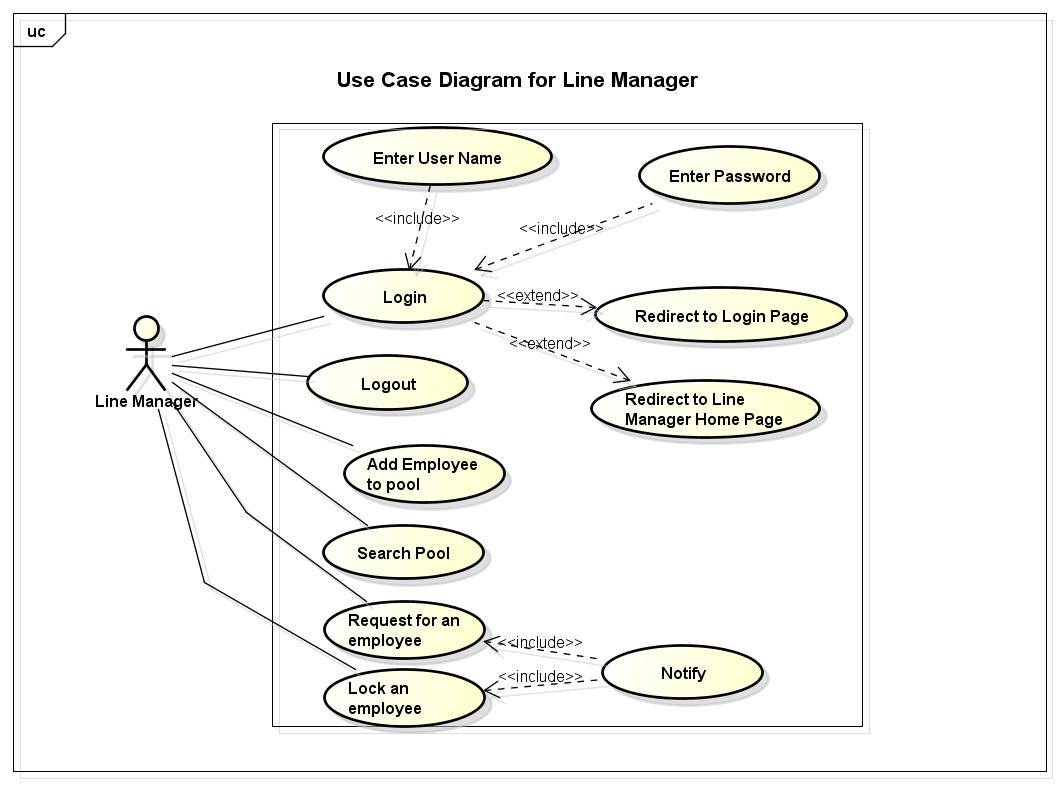


Figure01: Use case diagram for Higher Management

Figure02 : Use case diagram for product/line manager

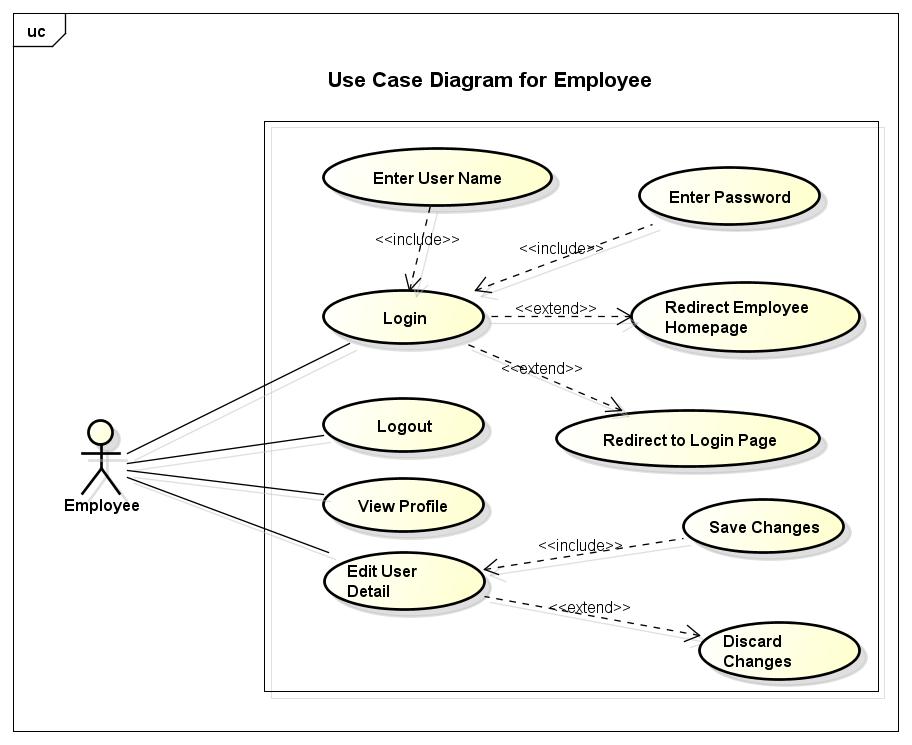
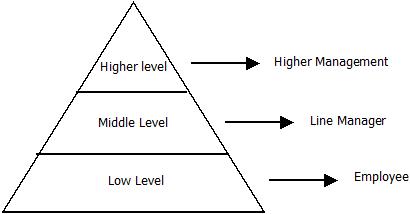


Figure03: Use case diagram for an Employee

* 1. **Use Case Elaboration**

The above three figures describe the behaviors and functions of each actors. When it comes to user levels, higher management has the responsibility of all controls in the system. Following figure illustrates the user hierarchy of the system.

Figure 04: User hierarchy of the system

|  |  |
| --- | --- |
| *Use case name* | **Login** |
| *Participating actors* | User (Employee, line Manager, higher management) |
| *Flow of events* | 1. User enters username & password. 2. If valid username & password, user can enter to the system. |
| *Entry Condition* | Only if username and password are valid. |

|  |  |
| --- | --- |
| *Use case name* | **Add employee to the system** |
| *Participating actors* | Higher Management |
| *Flow of events* | 1. Create new profile to employee 2. Assign into a project. 3. Send notification about allocation and username, password. |
| *Entry Condition* | If new employee comes. |
| *Exit condition* | If allocation done correctly. |

|  |  |
| --- | --- |
| *Use case name* | **Add employee to pool** |
| *Participating actors* | Line manager |
| *Entry Condition* | If there is employee going to expire. |
| *Exit condition* | Added employee detail has been stored correctly. |

|  |  |
| --- | --- |
| *Use case name* | **Lock an employee** |
| *Participating actors* | Line Manager |
| *Flow of event* | 1. Search an employee from the pool and lock. 2. Send notification alone with the locking details to all line managers, locked person. |
| *Entry Condition* | User should login and pool should not be empty. |
| *Exit condition* | Locking has been stored correctly or user has cancelled the locking. |

|  |  |
| --- | --- |
| *Use case name* | **Request an employee** |
| *Participating actors* | Line Manager |
| *Flow of events* | 1. Search an employee from the pool and send request. 2. Request is notified to particular employee, all line managers and higher Management. |
| *Entry Condition* | User should log in and if match the suitable employee. |
| *Exit condition* | Locking has been stored correctly or user has cancelled the locking. |

|  |  |
| --- | --- |
| *Use case name* | **Confirm/ Reject a request** |
| *Participating actors* | Higher management |
| *Flow of events* | 1. See the notification about the request. 2. Check the details. 3. Clicking confirm or reject button. |
| *Entry Condition* | When a request arises then only this function will work. |
| *Exit condition* | Confirm/reject has been stored correctly. |

|  |  |
| --- | --- |
| *Use case name* | **Edit personal information.** |
| *Participating actors* | Employee |
| *Flow of events* | 1. View the profile page. 2. Clicking edit button user can edit personal information. 3. Click save changes. |
| *Entry Condition* | User should log in. |
| *Exit condition* | Edited information has been stored or user discards changes. |

1. **Design Overview**
   1. **System Architecture**

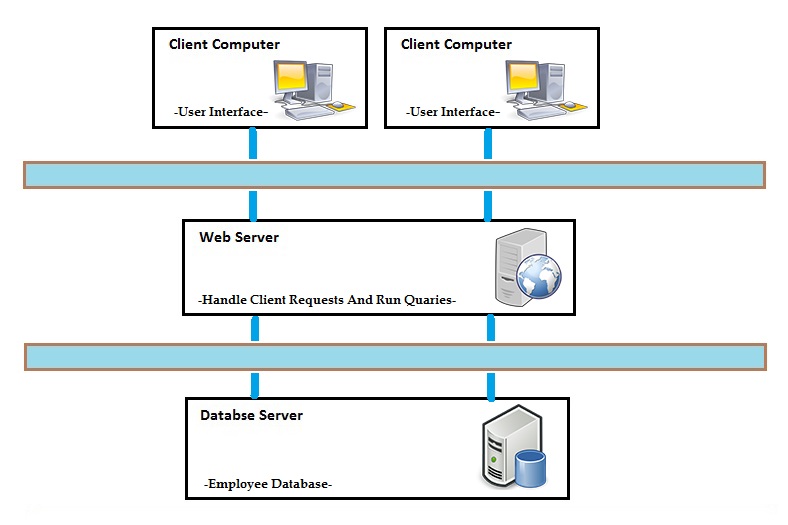
****

Figure 05: System Architecture

* 1. **System Interfaces**

An interface is a boundary across which two separate components of [computer system](http://en.wikipedia.org/wiki/Computer_system) to exchange information. The system we are going to implement has all theseUser Interfaces.

* System has a **login interface** which enables users to inputusername, password to enter into the system, from this interface user authentication is achieved, so authorized users can onlyview the information and can make changes into the system.
* User authentication is done by database which consists of employees’ information. **Username and password are verified from the database.**
* Depending on the user levels each level has an own interface.
* Employee has a GUI interface for view, edit profile information.
* Line managerhas a GUI forsearch, lock, request an employee from the employee pool database.
* Higher management user has a GUI interface for view each team details, add employee to the system, view summaries and add project to a team. Other than this higher management has a GUI interface for confirm, reject requests.
* All users have a GUI interface for change password and notification GUI for view notification associated to him/her.

1. **Data Design**
   1. **Entity Relationship Diagram**

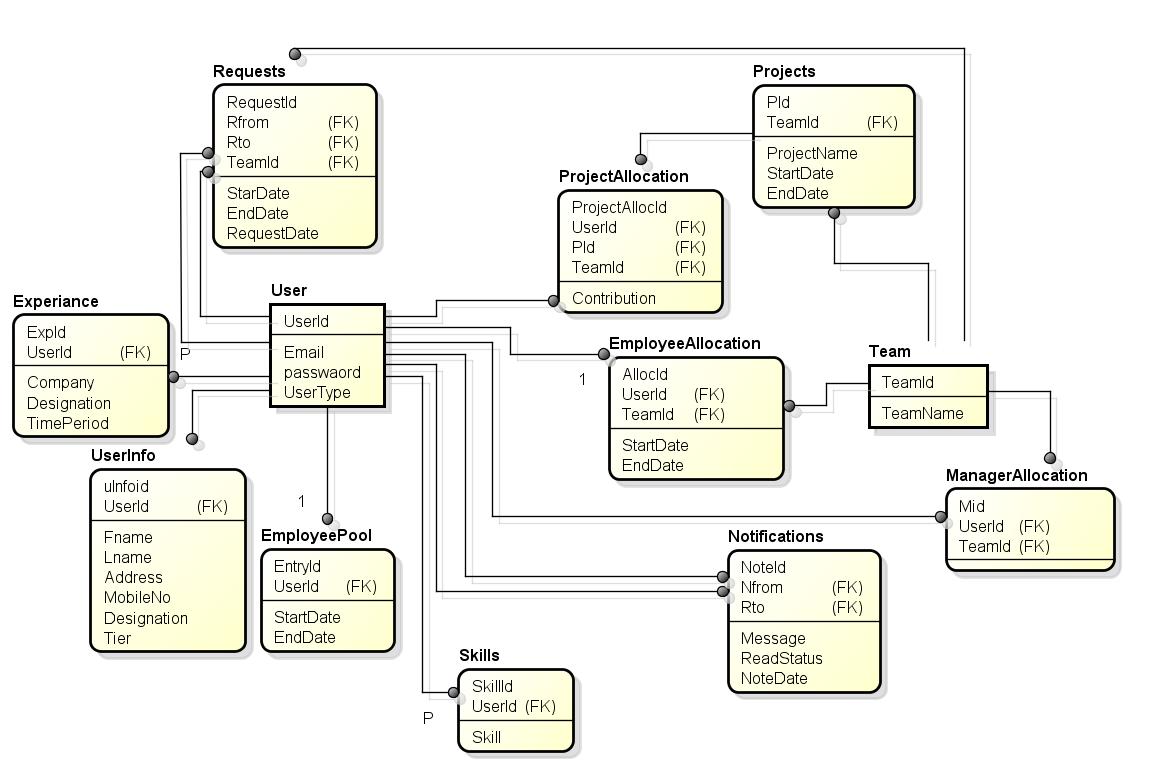
****

Figure 06: ER Diagram of the system

1. **Dynamic Models**
   1. **Sequence diagram**
      1. Login

Below Sequence diagram represents login structure of the system. First user view to main page where user gets a login window to authenticate him.Properly logged-in user will be lead to a page depend on the type of user level (Higher management/Linemanager/Employee).

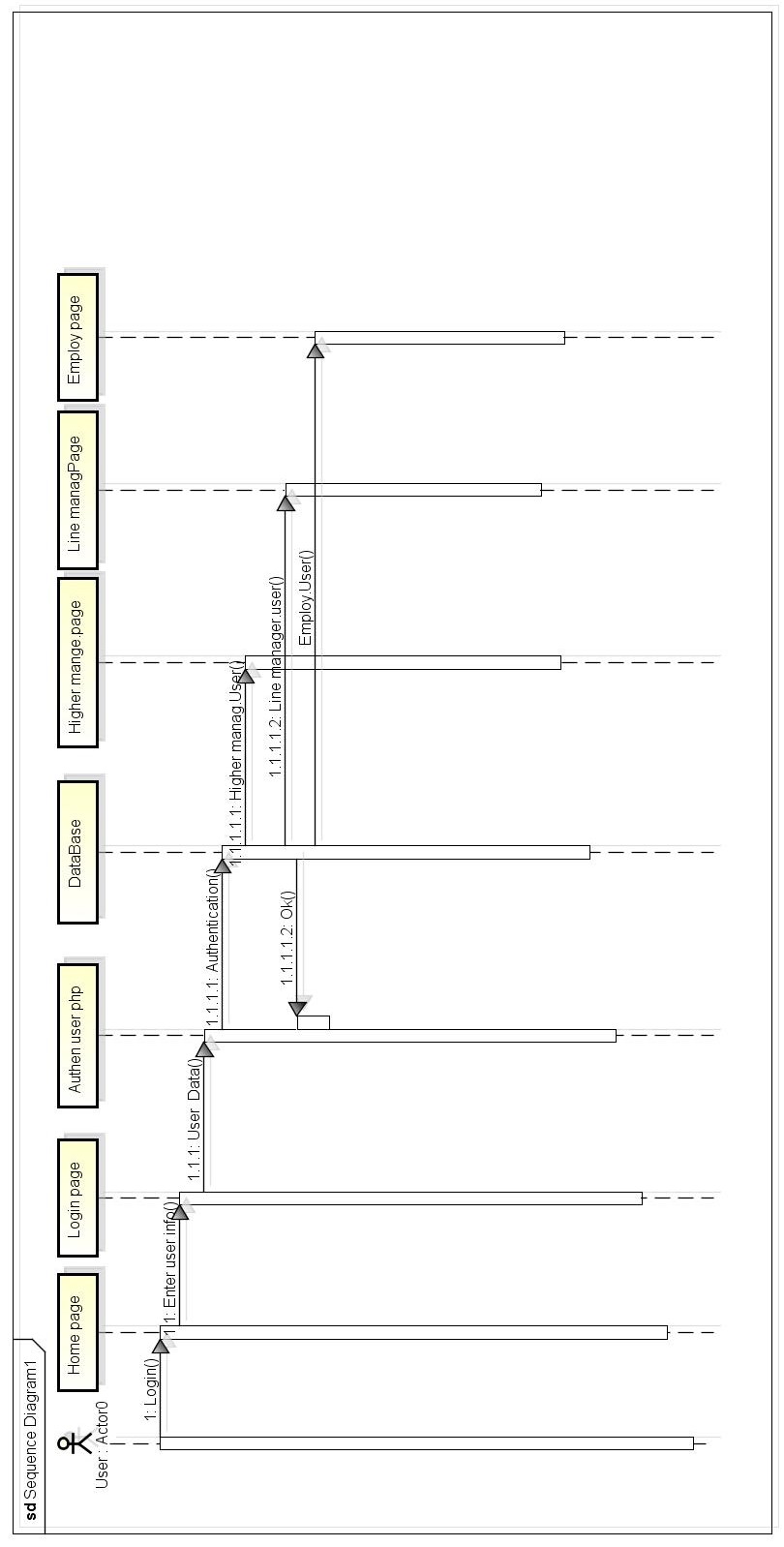
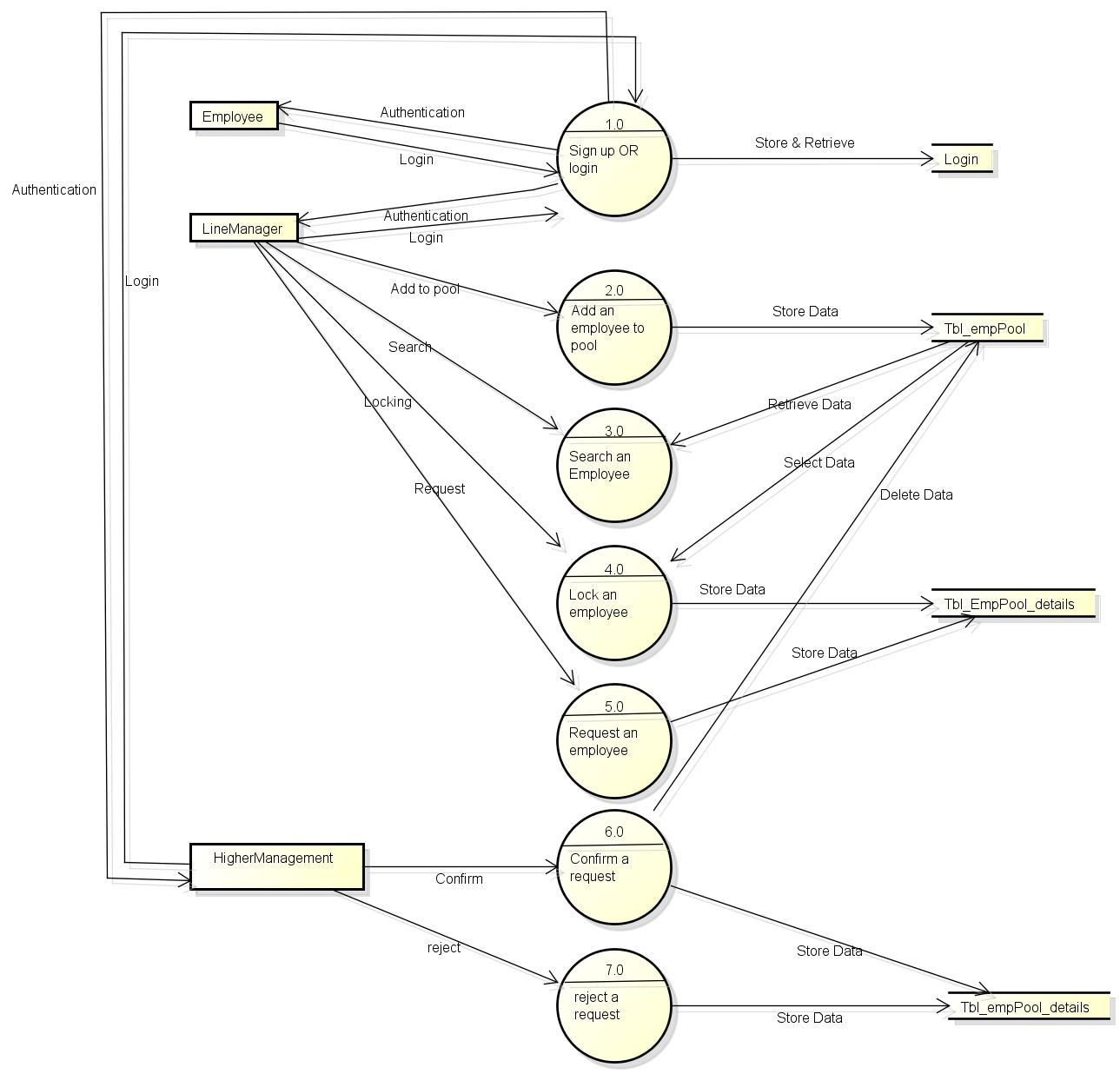


Figure 07: Sequence Diagram of Successful log-in to the system

* 1. **Data flow Diagram**

****Figure 08: Data Flow Diagram of the system

1. **Technologies**
   1. **Technologies going to use to create this System**
      1. HTML

**Hypertext Markup Language**, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

* + 1. CSS

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in HTML.

* + 1. PHP

We use this as server side scripting language.

* + 1. JavaScript

It is a scripting language that used to make web pages interactive.

* + 1. Apache

Free available web server. This allows MySQL and PHP to run on it.

* + 1. MYSQL

**MySQL** is an open source relational database management system (RDBMS) based on Structured Query Language (SQL).

* + 1. Bootstrap

**Bootstrap** is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

* + 1. CodeIgniter

**CodeIgniter** is a powerful open-source PHP framework with a very small footprint.

1. **References**

* <http://www.cardisoft.gr>
* <https://www.oasis-open.org>
* <http://en.wikibooks.org/wiki/Alevel_Computing/CIE/Computer_systems,_commuications_and_software/System_software/User_interfaces>
* <http://www.wikipedia.org>

**Approval**

|  |  |  |
| --- | --- | --- |
| Prepared by | Signature | Date |
| Sadhna MJS  Dinesh Chandradasa |  | 17.11.2014 |
| Approved by | Signature | Date |
|  |  |  |