./

Mini Project Report :

Java



|  |  |  |
| --- | --- | --- |
| **Name** | **PS Number** | **Email ID** |
|  |  |  |
|  |  |  |
| Shrikant Barkale | 99002458 | shrikant.barkale@ltts.com |

**Details**

# 

# V – Model Document

# Employee Database Management System

1.3 Definition of the Product:

Employee Management system which does the CRUD operation on the Employee Data

1. Create Employee
2. Retrieve Employee
3. Update Employee
4. Delete Employee

1.4 SWOT Analysis:

|  |  |
| --- | --- |
| Strength   1. Scalability 2. Flexibility 3. option for every Employee to choose different action. 4. Easy to Access | Weakness   1. Only provides 2. High Cost. |
| Opportunities   1. Very Portable. 2. Available to all communities and eaters. | Threats   1. When scouting locations, be sure to research crime statistics and avoid areas that have high incidents of theft and vandalism. 2. Damage or theft will quickly eat up your profits. 3. Mix age of the Variety rice’s. 4. Dispersion the raw materials. |

1.4 Requirements:

1.4.1 High Level Requirements:

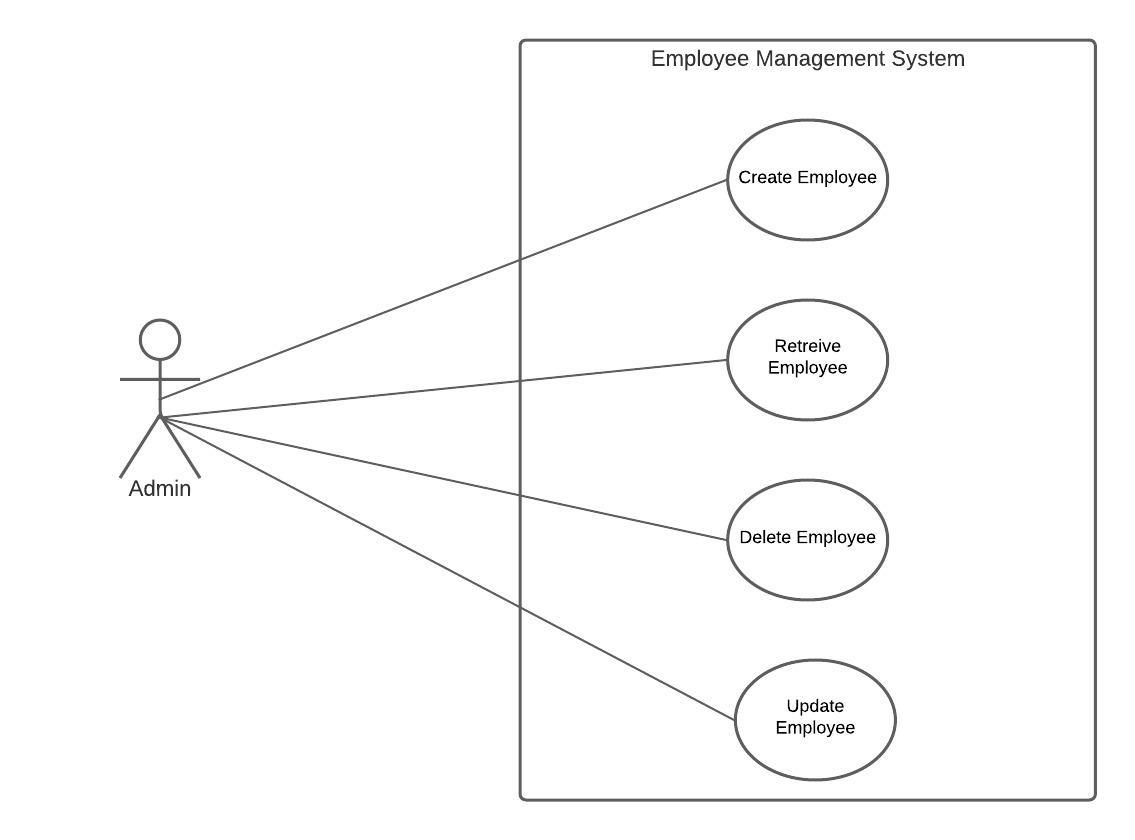
|  |  |
| --- | --- |
| **ID** | **Description** |
| H1 | Able to Login using username and password and send to Front page |
| H2 | Able to add Employee Details |
| H3 | Able to Update Employee Details |
| H4 | Able to Search Employee Details using empId |
| H5 | Able to View Employee Details using empId |
| H6 | Able to Delete Employee Details using empId |

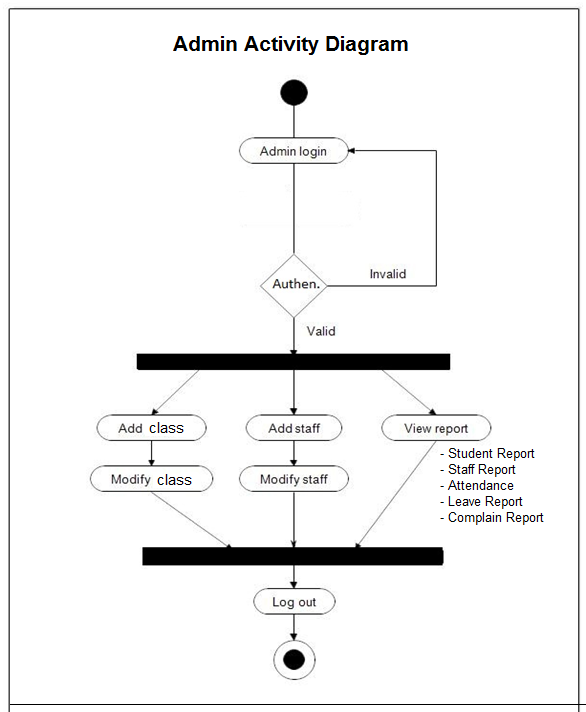
|  |  |
| --- | --- |
| **ID** | **Description** |
| L1 | Able to Call login function |
| L2 | Able to call add Employee function |
| L3 | Able to call update Employee function |
| L4 | Able to call search Employee function |
| L5 | Able to call View Employee function |
| L6 | Able to call delete Employee function |
| L7 | Should go to forget password page if needed |
| L8 | Able to Establish database connection |

1.4.2 Low Level Requirements:

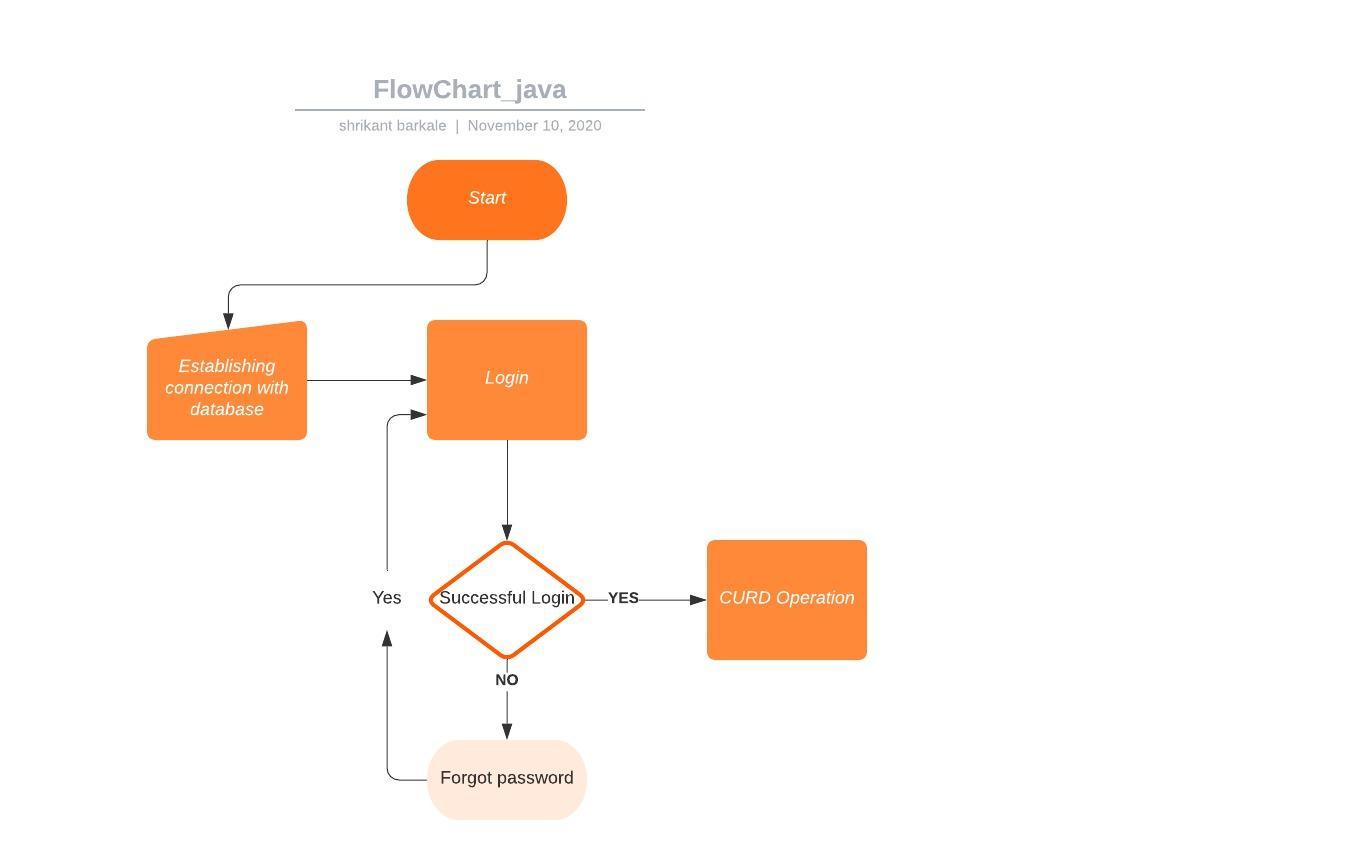
1.5 Design:

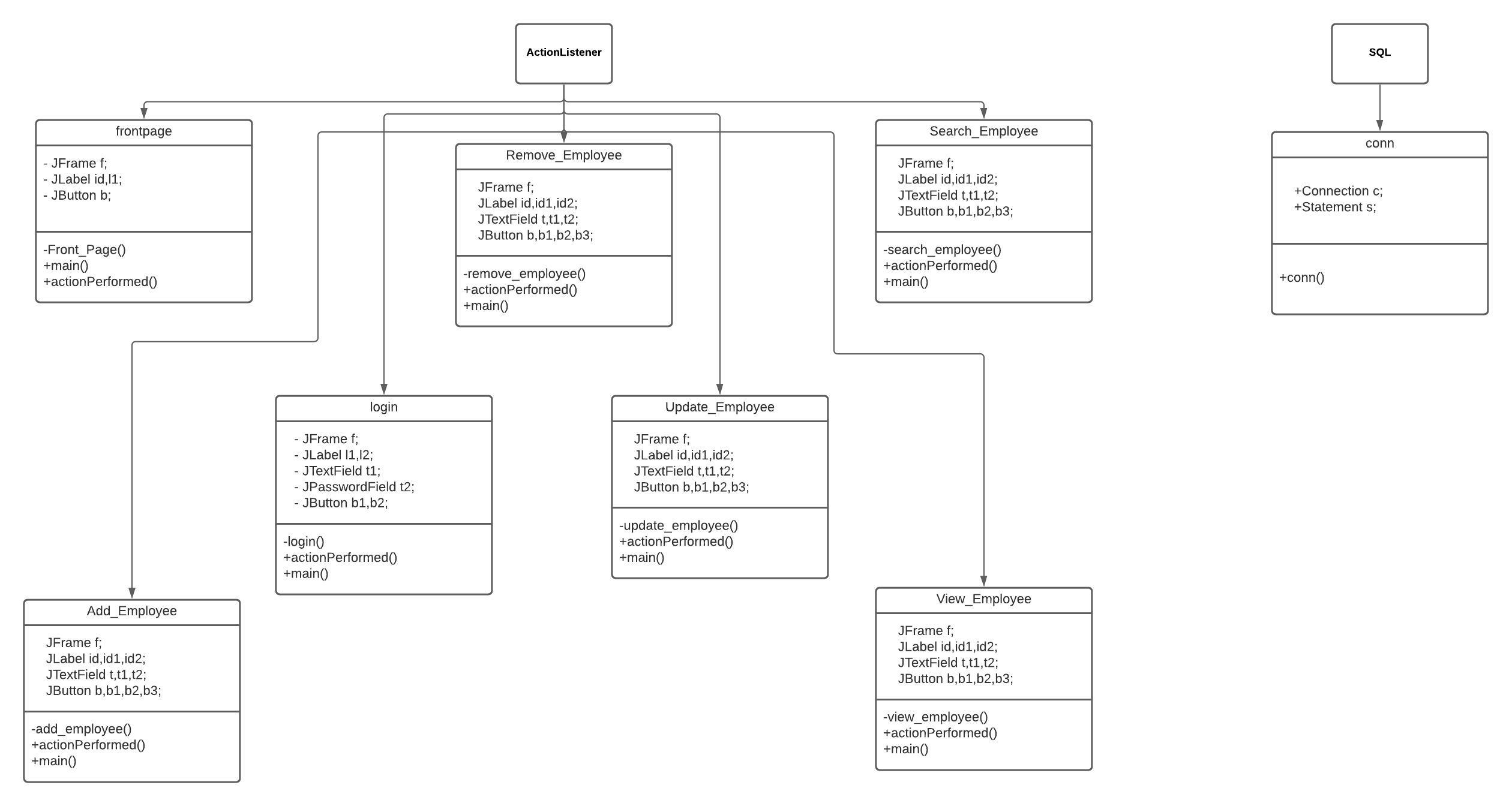
1.5.1 System Design





1.5.2 Unit Design





1.6 Test Plan:

1.6.1 Unit Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| **U1** | Testing the Insertion of the token. | Empty token port | Successful insertion of the token | Display the menu after the insertion of the token | Display the menu after the insertion of the token |
| **U2** | Displaying the menu of Quantities and the variety of rice’s. | Insertion of the correct token. | Input from the customer of the required dish | Display the dish and quantity after the insertion of the token | Display the dish and quantity after the insertion of the token |
| **U3** | Serving the food hot. | Insertion of the token and the selection of the food in the menu. | Wait for 5 minutes for dispensing | Get the dish hot and clean | Get the dish hot and clean |
| **U4** | Closing the process and waiting for the next input. | Supplied the food to the previous customer. | NA | Cleaned tray and ready for next. | Cleaned tray and ready for next. |

1.6.2 Integration Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| **I1** | Display the menu with correct menu and appropriate images. | NA | NA | Display the right food images and menu. | Display the right food images and menu. |
| **I2** | Able to mix the food with the right powder provided by the customer | The dish selected by the customer | The boiled rice and the powder | Correct texture and the temperature of the food. | Correct texture and the temperature of the food. |
| **I3** | Ready to convey the correct amount referenced by the client | The quantity mentioned during the start. | The right quantity of the rice and powder. | The correct quantity served. | The correct quantity served. |
| **I4** | Monitor the correct token | NA | The right token from the customer. | Displaying the Menu. | Displaying the Menu. |

1. Activity Diagram