Low Level Design (add Asset module)

|  |  |
| --- | --- |
| **Project Code** | 2\_AMS |
| **Project Name** | Asset Management System |

|  |  |  |
| --- | --- | --- |
| **Prepared/Modified by** | **Role** | **Date of Preparation** |
| Chitranjan Trivedi | Software Engineer | 25/11/2016 |
| **Reviewed by** | **Role** | **Date of Review** |
| Khushboo Kumar | Software Engineer | 25/11/2016 |
| **Approved by** | **Role** | **Date of Approval** |
| Mrs. Vaishali Srivastava |  |  |
| **Circulation List** | vaishali.srivastava@capgemini.com  khushboo.kumar@capgemini.com,  chitranjan.trivedi@capgemini.com,  divakar.rana@capgemini.com,  nidhi.barnwal@capgemini.com,  girjesh.soni@capgemini.com,  sneha.pandey@capgemini.com,  aman.jogi@capgemini.com, | **Version Number of the document:1.0** |
| **Version Number** | 1.0 |  |

Table of Contents

[1. Description Of The Unit 3](#_Toc306116213)

[2. References 4](#_Toc306116214)

[3. Dependencies / Assumptions 5](#_Toc306116215)

[4. Class Diagrams 6](#_Toc306116216)

[5. Sequence Diagrams 7](#_Toc306116217)

[6. Design Components 8](#_Toc306116218)

[6.1 List Of UI Components 8](#_Toc306116219)

[6.2 List Of Design Classes 8](#_Toc306116220)

[7. Re-Usable Routines/Components 9](#_Toc306116221)

[8. Messages 10](#_Toc306116222)

[9. Configuration Files 11](#_Toc306116223)

[10. Crud Matrix Table 12](#_Toc306116224)

[11. Data Mapping Matrix 13](#_Toc306116225)

[12. Action Flow Matrix 14](#_Toc306116226)

[13. Stored Procedures 15](#_Toc306116227)

[14. Program Specification / Pseudo Code 16](#_Toc306116228)

[REVISION HISTORY OF THE WORK PRODUCT 17](#_Toc306116229)

# Description Of The Unit

The login module allows the admin and the manager of the application to login. This module can be used only by the admin and the manager of the application.

On clicking on the login button on the index page, the admin is redirected to its home page to add asset, update asset and display request. The admin enters all the details of the asset like asset name, asset description, asset quantity and asset status. If the entered product details are invalid, appropriate error messages are displayed.

And for the manager, on clicking login button on the index page, the manager is redirected to its home page to view asset, raise request and view request status.

# References

| **Reference** | **Description** |
| --- | --- |
| Repository reference | https://github.com/ThreadPoll/FinalServletJspProjects/blob/master/AssetManagement.txt |
| Class Diagram | /Project Documents/Design Documents/HLD/Class Diagrams |
| URS | /Project Documents/Requirements/Case Study/Group 2-Asset Management System.doc |
| UTP | /Project Documents/Test Plan/UTP/Test Cases/Login /UTP\_LoginModule.xlsm |
| DB Design | /Project Documents/Design Documents/HLD/DB Design/AssetManagement\_GROUP2\_DBDESIGN.xls |

# Dependencies / Assumptions

**Dependencies:**

Only admin and manager can login and for new user, he/she should register as an employee to login.

**Assumptions:**

The admin details are stored in the database.

The admin will not have any manager.

A person who is a manager is his own manager.

# Class Diagrams

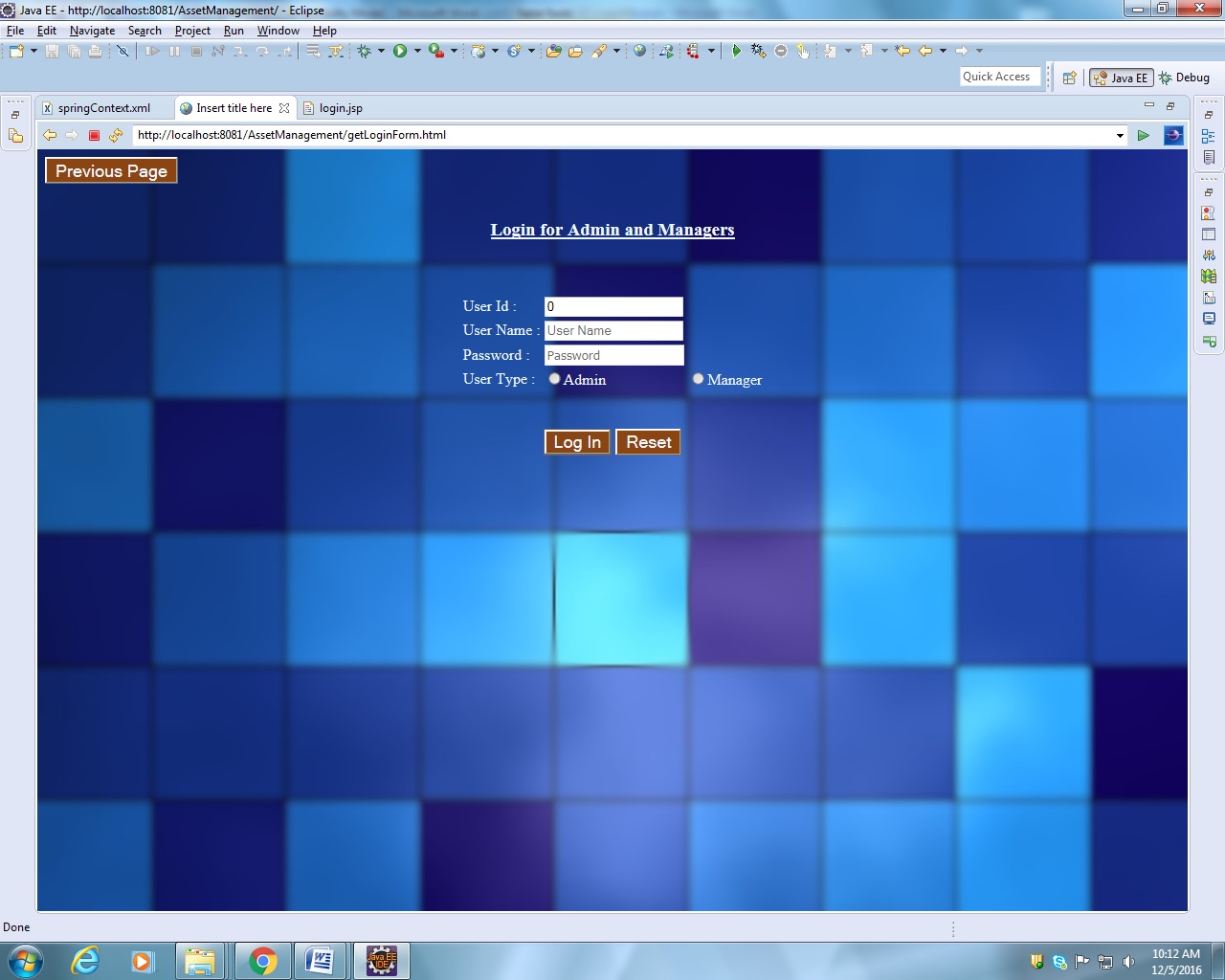
[Class Diagram](file:///D:\Users\HLD\Class%20Diagrams)

# Sequence Diagrams

|  |
| --- |
| **Sequence Diagram** |
| NA |

# Design Components

## List Of UI Components



## List Of Design Classes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **LLD Name** | **Class Name** | **Method Name** | **Method Description** |
| 1 | Login Module | User and  Employee | Appropriate getters and setters methods | Getters retrieve the property values and setters set the property values |
| 2 | Login Module | AssetFrontController | getLoginForm(Model model): String  login(@ModelAttribute User user,Model model , HttpServletRequest request):String | Sets the model attribute.  Calls isValidUser() of AssetUserServiceImpl class, which in turn calls isValidUser() of AssetUserDAOImpl and if user is present in user\_master in database, it redirects to appropriate Pages based on the role of the user. |
| 3 | Login Module | AssetUserServiceImpl | isValidUser():user | Calls isValidUser() of AssetUserDAOImpl class and returns the result to controller. |
| 4 | Login Module | AssetUserDAOImpl | isValidUser():user | Validates userId and password as per the database and returns the result to AssetUserServiceImpl . |

# Re-Usable Routines/Components

Re-Usable Routines/Components

* CSS files
* Javascript files
* DataAccessException class
* Database properties files
* Logger properties files

# Messages

|  |  |  |
| --- | --- | --- |
| **#.** | **Type** | **Message Text** |
| M1. | Error | Invalid User Name/ Password |

# Configuration Files

|  |  |  |
| --- | --- | --- |
| File Name | Path | Description |
| springContext.xml | /WebContent/WEB-INF/ config | To configure database connection.  To configure application context. |
| Web.xml | /WebContent/WEB-INF | To deploy the webapplication to any Java Servlet container. |
| servlet-config.xml | /WebContent/WEB-INF/ config | To configure the request mapping and view resolver. |

# Crud Matrix Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Table Name** | **Create** | **Read** | **Update** | **Delete** |
| 1 | User | N | Y | N | N |

# Data Mapping Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Screen Label** | **Field Source** | **Field Type** | **Data Type** | **Max Length** | **Data Flow** | **Mandatory** |
| 1 | User Id | NA | Label | NA | NA | UI | NA |
| 2 | userId | User. userId | TextBox | Varchar2 | 50 | UI🡪D | Yes |
| 3 | User Name | NA | Label | NA | NA | UI | NA |
| 4 | userName | User. userName | TextBox | Varchar2 | 50 | UI🡪D | Yes |
| 5 | Password | NA | Label | NA | NA | UI | NA |
| 6 | userPassword | User. userPassword | TextBox | Varchar2 | 50 | UI🡪D | Yes |
| 7 | User Type | NA | Label | NA | NA | UI | NA |
| 8 | userType | User. userType | TextBox | Varchar2 | 50 | UI🡪D | Yes |
| 13 | Login | NA | Button | NA | NA | UI | NA |
| 14 | Reset | NA | Button | NA | NA | UI | NA |

**Legend**

D 🡪 UI (Database to User Interface)

UI 🡪 D (User Interface to Database)

UI (User Interface Only)

# Action Flow Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Action** | **Description** | **Process** |
| 1 | On click of Previous Button in login page | Redirected to the home page | The user clicks on the previous button in the login page and gets redirected to the home page. |
| 2 | On click of Login button in add page | The user is directed to the appropriate page if the user is a valid user. | Data entered in the fields by the user are checked in the database when login button is pressed and the user is redirected to the appropriate page if the user is a valid user. |
| 3 | On click of Reset button in Add page | The form data is cleared. | The user on clicking on the reset button in login page gets redirected to the same page with no entries. |

# Stored Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Stored Procedure Name**  **(with Description)** | **Input/Out Parameters** | **Type** | **Size** |
| 1. | NA | NA | NA | NA |

# Program Specification / Pseudo Code

## <ACTION :ONPAGE\_LOAD >

**Pseudo Code:**

1. **Client Side Scripting:**

None

1. **Server Side Scripting:**

**BEGIN**

Call loginPage in Controller class

Navigate to Login Page.

**END**

**14. 2 <ACTION: LOGIN\_CLICK>**

**Pseudo Code:**

1. **Client Side Scripting:**

IF UserId isEmpty

Print Error Message M1

IF Password isEmpty

Print Error Message M1

IF invalid UserId

Print Error Message M1

IF invalid Password

Print Error Message M1

IF length of UserId > 10 characters

Print Error Message M1

IF length of Password > 16 characters

Print Error Message M1

1. **Server Side Scripting:**

**BEGIN**

Call checkLogin(@ModelAttribute(value=” username”)ValidUser user,BindingResult result,Model model) in Controller class in Presentation Layer

**BEGIN**

Call isValidUser(User user) in AssetServiceImpl class in Service Layer

**BEGIN**

Call isValidUser(User user) in AssetDAOImpl class in Service Layer

**BEGIN**

Create query to search if the user exists or not

**END**

Return int to AssetServiceImpl

**END**

Return int to Controller class

**END**

If return==1

**BEGIN**

Call searchUser(User user) in AssetServiceImpl class in Service Layer

**BEGIN**

Call searchUser(User user) in AssetDAOImpl class in Service Layer

**BEGIN**

Create query to retrieve the role of the user

**END**

Return role to AssetServiceImpl

**END**

Return role to Controller class

Navigate to respective pages based on their roles

Else navigate to login page

**END**

**14. 2 <ACTION: RESET\_CLICK>**

**Pseudo Code:**

**1. Client Side Scripting:**

None

1. **Server Side Scripting:**

None

REVISION HISTORY OF THE WORK PRODUCT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Date | Version # | Section Changed | Details of changes made | Approved By |
| 1 | 25/11/2016 | 1.0 | LLD prepared | N/A |  |