|  |  |
| --- | --- |
|  | **Fsoft logo for Words-Small.gif** |

**Design Document**

**For Customer Management System**

*Prepared for*

**Friday, May 08, 20155**

**Version 1.0**

*Prepared by*

**Hoang Thanh Son**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Change reference** |
| 05/07/2015 | Hoang Thanh Son | 1.0 | Create new |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version** | **Position** | **Date** |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc418850088)

[1.1 Purpose 4](#_Toc418850089)

[1.2 Scope 4](#_Toc418850090)

[1.3 Intended Audiences and Document Organization 4](#_Toc418850091)

[1.4 Acronyms and Abbreviations 4](#_Toc418850092)

[1.5 References 5](#_Toc418850093)

[2 Architecture design 5](#_Toc418850094)

[2.1 Application Logical Architecture 5](#_Toc418850095)

[2.1 User case view 5](#_Toc418850096)

[2.1.1 Overview 5](#_Toc418850097)

[2.1.2 Description 6](#_Toc418850098)

[2.2 Architectural Representation 6](#_Toc418850099)

[2.2.1 JSP WEB PAGE Layer 6](#_Toc418850100)

[2.2.2 Data Access Layer 7](#_Toc418850101)

[2.2.3 Data Layer 7](#_Toc418850102)

[2.3 Packages/Components view 7](#_Toc418850103)

[2.2.4 Web 7](#_Toc418850104)

[2.2.5 Model Object ( Entity ) 7](#_Toc418850105)

[2.2.6 Data Access Layer 8](#_Toc418850106)

[2.3 Exception handling mechanism 8](#_Toc418850107)

[3 Database design 8](#_Toc418850108)

[3.1 Entity Relationship Diagram 8](#_Toc418850109)

[3.2 Schema 8](#_Toc418850110)

[3.2.1 User 8](#_Toc418850111)

[4 Application Security 8](#_Toc418850112)

[4.1 . Main functions ‘s security matrix 8](#_Toc418850113)

[5 Interface Design 10](#_Toc418850114)

[5.1 Layout 10](#_Toc418850115)

[5.2 Themes 10](#_Toc418850116)

[6 Details function design 10](#_Toc418850117)

[6.1 Use case 01: Login Logout 10](#_Toc418850118)

[6.1.1 Login 10](#_Toc418850119)

[6.1.2 Logout 12](#_Toc418850120)

[6.2 Use case 02: Maintain Customer List 13](#_Toc418850121)

[6.2.1 Search a customer 13](#_Toc418850122)

[6.2.2 Update a customer 14](#_Toc418850123)

[6.2.3 Delete a customer 15](#_Toc418850124)

[6.2.4 Add a customer 16](#_Toc418850125)

[7 Packaging and Deployment 18](#_Toc418850126)

[8 Appendix 18](#_Toc418850127)

# Introduction

## Purpose

This document contains the detailed design for to be developed application on target platform. It defines, technically, how applications will operate. Developers will base on this document and corresponding SRS to conduct development plan, task assignment and implementation of the new application.

## Scope

This document is prepared for the application Customer Management System in scope of the MOCK project of TCS-Team.

## Intended Audiences and Document Organization

This document is intended for:

* Development team: Developers and Testers
* Rollout Technical Team: Responsible for deploying applications to UAT and Production environments.
* Customer Representatives: Responsible to review & approve the document.

Below are main sections of the document:

* **Introduction** : This section describes the general introduction of this document
* **Architecture Design :** This section describes the high-level technical assessments and decisions for the application.
* **Data Design**: This section describesin detail how data is structured and manipulated in this application.
* **Interface Design:** This section describesin detail how UI is designed in general ( layout , theme ).
* **Application Security**: This section describles security matrix in detail
* **Configuration:** This section describes all configuration needed for the application to function properly.
* **Packaging and Deployment:** This section describles how applications could be packaged and deployed.
* Note:Please refer section 1.4 for all acronyms and abbreviations you may encounter within this document.

## Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| # | Item | Description |
| 1 | JSP | Java Server Page |
| 2 | JVM | Java Virtual Machine |
| 3 | HTTP | Hypertext-Transfer Protocol |
| 4 | MVC | Model View Controller |
| 5 | DAL | Data Access Layer, this kind object is responsible for attaching to a system, extracting some information, based on specific requirements, and creating a model object, update to database |
| 6 | MO | Model Object, Object is description of a real object : customer, user |

## References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Title | Version | File Name / Link | Description |
| 1 | Common Requirements Document | 1.0 |  |  |
| 2 |  |  |  |  |

Table 1.1: List of References

# Architecture design

## Application Logical Architecture

## User case view

### Overview

The following use cases are required to be to be included in the Service Directory



### Description

|  |  |  |
| --- | --- | --- |
| **UC** | **Name** | **Description** |
| UC001 | Login & Logout | Allowing a user to logon to the system |
| UC002 | Maintain User | Maintain the user list, password |
| UC003 | Maintain Customer | Maintain the data for customer |

## Architectural Representation

The following diagram shows the primary tiers in the proposed 3-tier architecture. This diagram shows the main layers in this architecture and the vision of how they fit together.



Figure 1 – 3-tier architecture of CMS System

### JSP WEB PAGE Layer

This layer controls the display to the end user, on the other hand, jsp can have java code so that it have business logic controls, the development framework is based on Jsp model. The framework is responsible for:

* Managing requests/responses from/to the clients.
* Controlling display to the end user.
* Assembling a model that can be presented in a view.
* Performing UI validation.
* Handling exceptions from other layers.
* Handling application business logic and business validation.
* Managing transactions
* Adding flexibility between the presentation and the persistence layer so they do not directly communicate with each other
* Send model object to data acess layer

### Data Access Layer

This layer manages access to persistent storage. The primary reason to separate data access from the rest of the application is that it is easier to switch data sources and share Data Access Layer Object (DALOs) between applications.

This layer manages reading, writing, updating, and deleting stored data.

### Data Layer

In CMS SYSTEM, the storage is managed by a relational database. SQL SERVER 2008 is used for this layer to provide the management of stored data. To connect from Data Access Layer to SQL Server, we use ODBC bridge.

## 2.3 Packages/Components view



### Web

This package includes the implementation for the JSP file proposed to be used in the JSP Web Page Layer to handle the display to the end user.

**Validation**: All validation of incoming requests parameters to the server should be validated using Java code Validation or client side control .

This package includes the implementation of business java code.This layer is used to perform the business operations. The business JAVA code will access the DAL to access database. Transactions should be managed within this layer.

### Model Object ( Entity )

Model object is java class, contains lightweight structures for related business information. These are sometimes referred to as data transfer objects. A value object (VO) is a lightweight, serializable object that structures groups of data items into a single logical construct. .In addition, VOs are useful in communication among all layers of the application.

### Data Access Layer

This package includes the implementation of Data Access Object. Using SQLCLient object here to make the application more flexible to access database. SQLClient object includes basic functions to work with database: *select, insert, update, delete*.

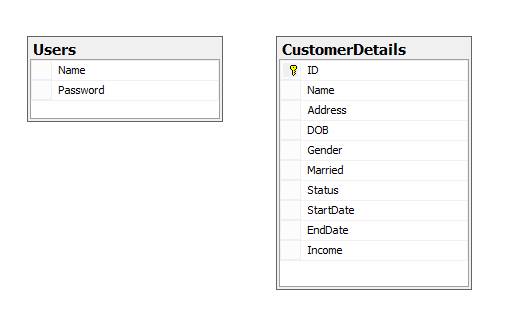
## Exception handling mechanism

The try-catch clauses should be kept to a minimum.

Whenever an exception raise , system will log exception detail to Web message and redirect user to Error page to display standard error message!

# Database design

## Entity Relationship Diagram



## Schema

**Overview**

|  |  |  |
| --- | --- | --- |
| **STT** | **Name** | **Description** |
| 1 | User | List all user that can access to website feature |
| 2 | Customer | List all customer |

### User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
|  | **Field name** | **Type** | **Max Length** | **Descrition** |
| 1 | UserName | nvarchar | 50 | Username of user, primary key |
| 2 | PassWord | nvarchar | 30 | Password of user |

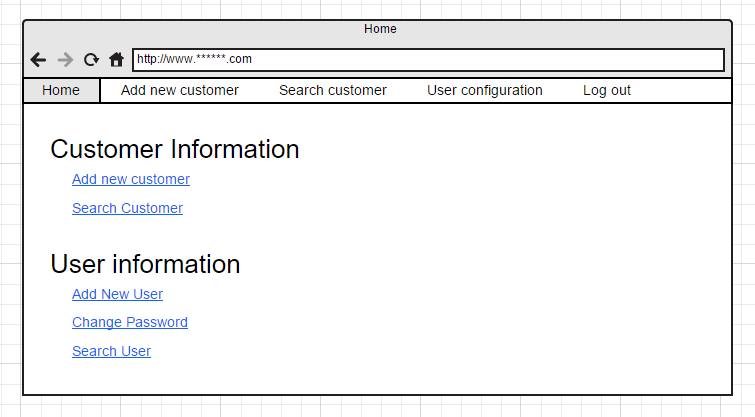
# Application Security

## . Main functions ‘s security matrix

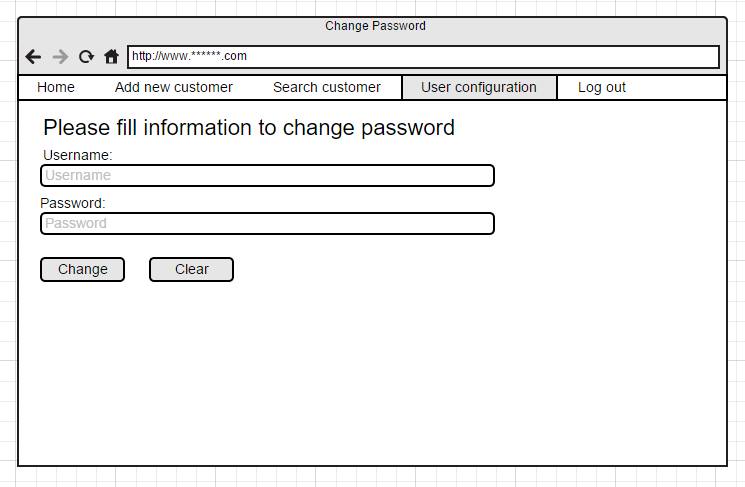
|  |  |  |
| --- | --- | --- |
| **Actor**  **Usercase** |  | **User** |
| Maintain Customer | Create a new Customer | x |
| List all customer | x |
| Delete a customer | x |
| Update info | X |
| Maintain Services | Delete User | x |
| Change Password | x |
| Search user | x |
| Logon & Logout |  | x |

# Interface Design

## Layout



Home page



Change Password

## Themes

This application use Bootstrap to decorate Website, non edited

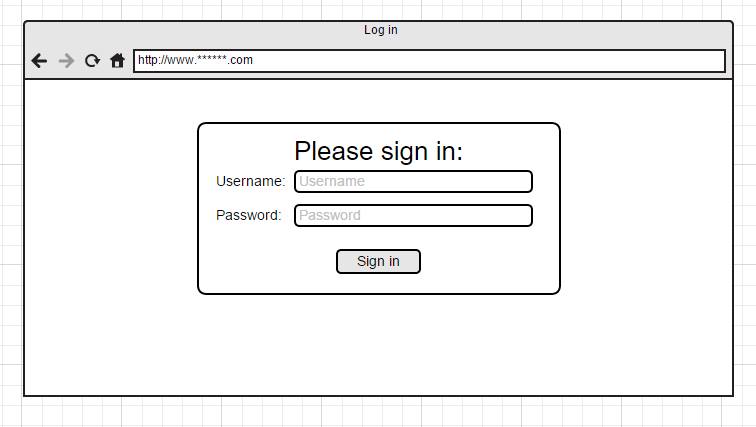
# Details function design

## Use case 01: Login Logout

### Login

|  |  |
| --- | --- |
| **Name** | Login |
| **Description** | This use case allows user to login to web application to use feature |
| **Actor** | Anonymous user |
| **Trigger** | When user fill the login form then clicking on Login button |
| **Pre-condition** | **Logged out for go to Customer manager home page** |
| **Post-condition** | **Go to home page** |

#### Screen Design & Data Description



Login

#### Activities Flow



Figure .:Activities Flow

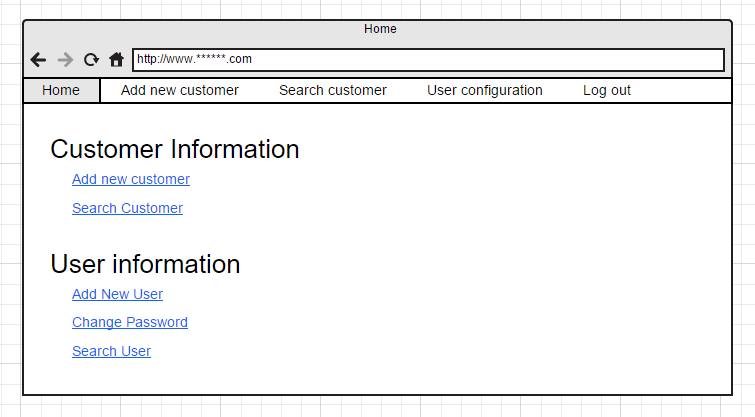
#### Detail Processing

|  |  |  |
| --- | --- | --- |
| Activity | BR Code | Description |
| (2) | BR01 | Loading rules:   * System response login form in login.jsp |
| (4) | BR02 | Data filter rules:   * User fill user name and password in 2 text field * User click on login, web browser send from to server |

### Logout

|  |  |
| --- | --- |
| **Name** | Logout |
| **Description** | This use case allows user to logout |
| **Actor** | Authenticated user |
| **Trigger** | When user clicking on ‘Logout’ button in nav bar from every webpage except for login page |
| **Pre-condition** | **User has been logged in** |
| **Post-condition** | **User logged out from web apps, web app response login page** |

#### Screen Design & Data Description



Home page with log out button in nav bar

#### Activities Flow



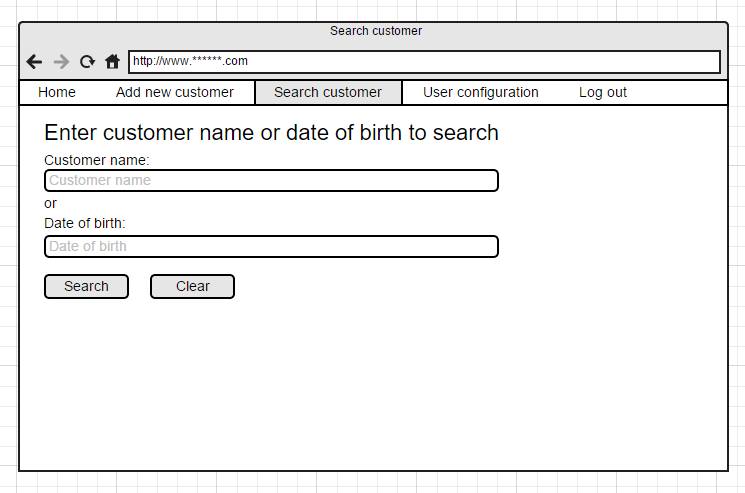
Figure Activities Flow

## Use case 02: Maintain Customer List

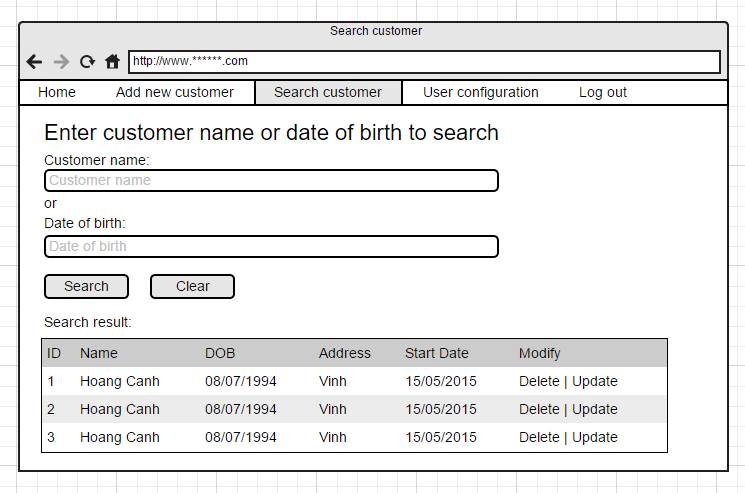
### Search a customer

|  |  |
| --- | --- |
| **Name** | Search customer |
| **Description** | This use case allows user to search customer by name and DOB |
| **Actor** | Authenticated user |
| **Trigger** | When user fill the search form then clicking on Search button |
| **Pre-condition** | **Logged in and click on Search customer in nav bar or menu at home** |
| **Post-condition** | **Show search result** |

#### Screen Design & Data Description



Search page

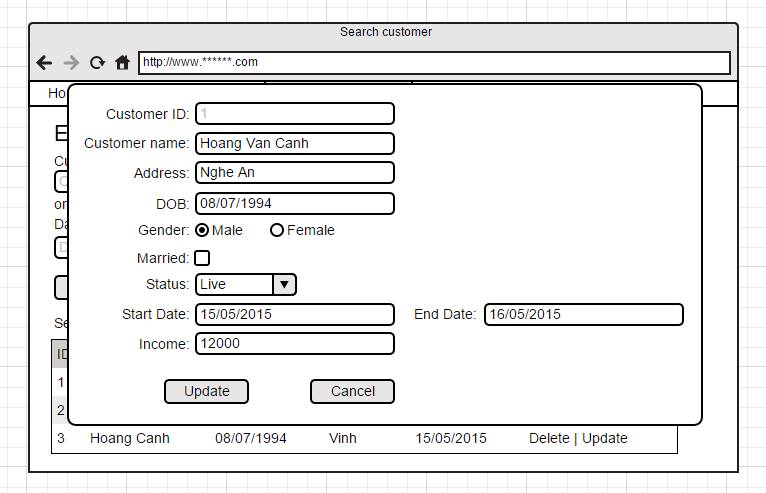


Search result

### Update a customer

|  |  |
| --- | --- |
| **Name** | Update customer |
| **Description** | This use case allows user to update a selected customer’s info |
| **Actor** | Authenticated user |
| **Trigger** | Click on update btn in search result, change info and click on save btn |
| **Pre-condition** | **Have one or more search result** |
| **Post-condition** | **Show dialog customer’s info** |

#### Screen Design & Data Description

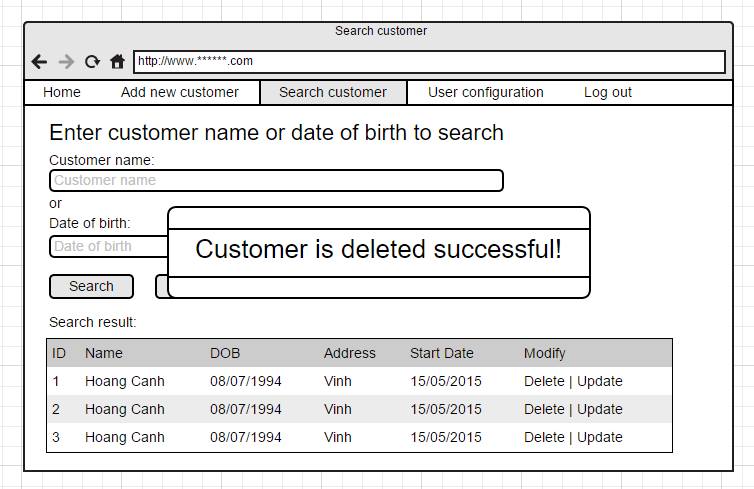


Update customer Info

### Delete a customer

|  |  |
| --- | --- |
| **Name** | Delete customer |
| **Description** | This use case allows user to delet a selected customer’s info |
| **Actor** | Authenticated user |
| **Trigger** | Click on update delete in search result |
| **Pre-condition** | **Have one or more search result** |
| **Post-condition** | **Show dialog delete successfully, update customer result list** |

#### Screen Design & Data Description



Delete a customer

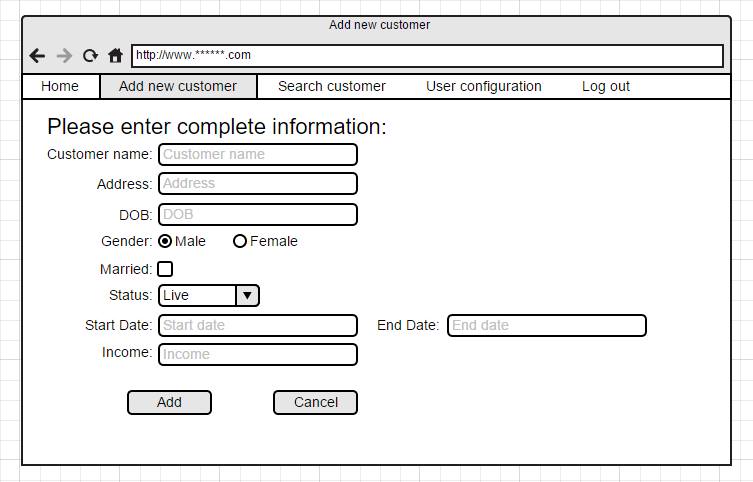
#### Activities Flow



### Add a customer

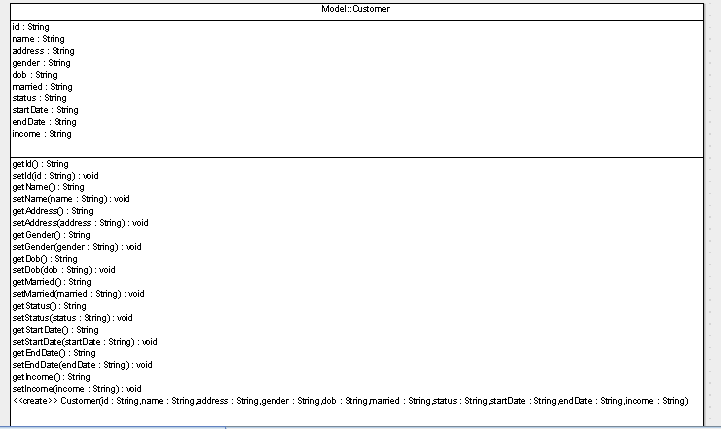
|  |  |
| --- | --- |
| **Name** | Add a customer |
| **Description** | This use case allows user to add a new customer |
| **Actor** | Authenticated user |
| **Trigger** | When user clicking on Add button in nav bar from every webpage except for login page |
| **Pre-condition** | **User has been logged in** |
| **Post-condition** | **Go to add customer page, fill the from and click on Save button** |

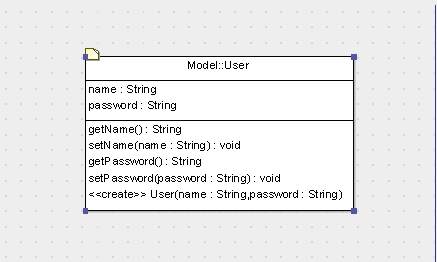
#### Screen Design & Data Description

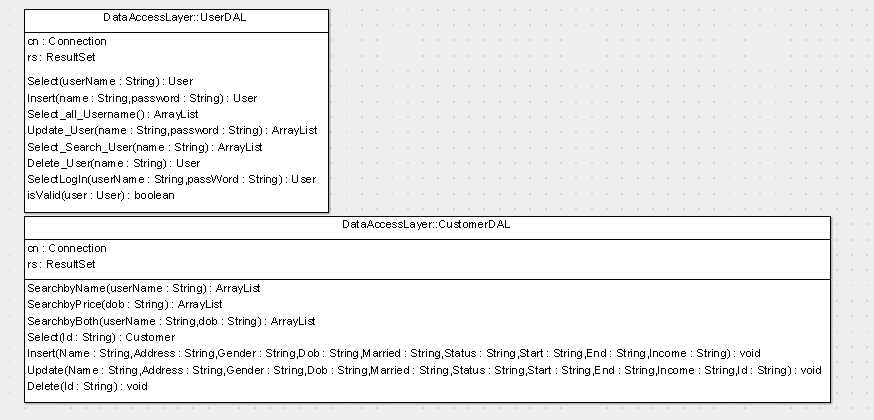


Add a new customer

#### Class diagram







# Packaging and Deployment

Application will be build to a war file. After that, this war file can be deploy to any Apache tomcat server. Access web app by link /CustomerManager

# Appendix