



Employee Management System

Document Version / Details : Ver. 1.30/

28-Jul-2020

**Technical Design Document**

**Record of Release**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version No. | Modified By | Reviewed By | Authorized By | Release Date | Modifications Done |
| 1 | Roshan Zameer Ali S | Hasan Ali DoulatGanjWala,  Anjum Fathima | Hasan Ali DoulatGanjWala,  Anjum Fathima | 16-02-2021 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[1. Introduction 4](#_Toc46250930)

[1.1 Document Reference 4](#_Toc46250931)

[1.2 Document Approval 4](#_Toc46250932)

[2. Technical Overview 5](#_Toc46250933)

[2.1 Integration Solution Design 5](#_Toc46250934)

[2.2 Assumptions 5](#_Toc46250935)

[2.3 Dependencies & Constraints 5](#_Toc46250936)

[2.4 Source and Target Applications 5](#_Toc46250937)

[3. Integration Component Details 7](#_Toc46250938)

[3.1 homepage Component 7](#_Toc64276603)

[3.1.1 Component details 7](#_Toc64276604)

[3.1.2 Services and Components 8](#_Toc64276605)

[3.2 register Component 8](#_Toc64276606)

[3.2.1 Component details 8](#_Toc64276607)

[3.2.2 Services and Components 9](#_Toc64276608)

[3.3 login Component 1](#_Toc64276609)0

[3.3.1 Component details 1](#_Toc64276610)0

[3.3.2 Services and Components](#_Toc64276611) 11

[3.4 employee dashboard Component](#_Toc64276612) 11

[3.4.1 Component details](#_Toc64276613) 11

[3.4.2 Services and Components 12](#_Toc64276614)

[3.5 admin dashboard Component 1](#_Toc64276615)3

[3.5.1 Component details 1](#_Toc64276616)3

[3.5.2 Services and Components](#_Toc64276617) 14

[3.6 Log Out Component](#_Toc64276618) 15

[3.6.1 Component details 1](#_Toc64276619)5

[3.6.2 Services and Components..................................................................................1](#_Toc64276620)6

[4. Error Handling Notification and Reporting 1](#_Toc46250944)7

[4.1 Technical & Business error 1](#_Toc46250945)7

[6. Glossary 1](#_Toc46250948)8

# Introduction

1. The introduction of the **Requirements Specification** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the document.
2. The **Requirments Specification** captures the complete software requirements for the system, or a portion of the system. Following is a typical outline for a project **using use case modeling**. This artifact consists of a package containing use cases of the use case model and applicable Supplemental Specifications and other supporting information.
3. This subsection describes what the rest of the **Requirements Specification** contains and explains how the document is organized.

Employee Management System will allow an organisations’ employees to effectively View their profile and update the details and also it will be able to help the Admin to manage the Employees by updating their details, deleting and View their Profiles. The application has five major components HomePage, Register, Login, Employee Dashboard, Admin Dashboard. The application built using Angular, Asp.net WebApi and database used is MS SQL. This is a database first application.

## Document Reference

| Document Name | Comments |
| --- | --- |
|  |  |
| Use Case Specification Document | This document contains the list of use cases that is used to develop the application |

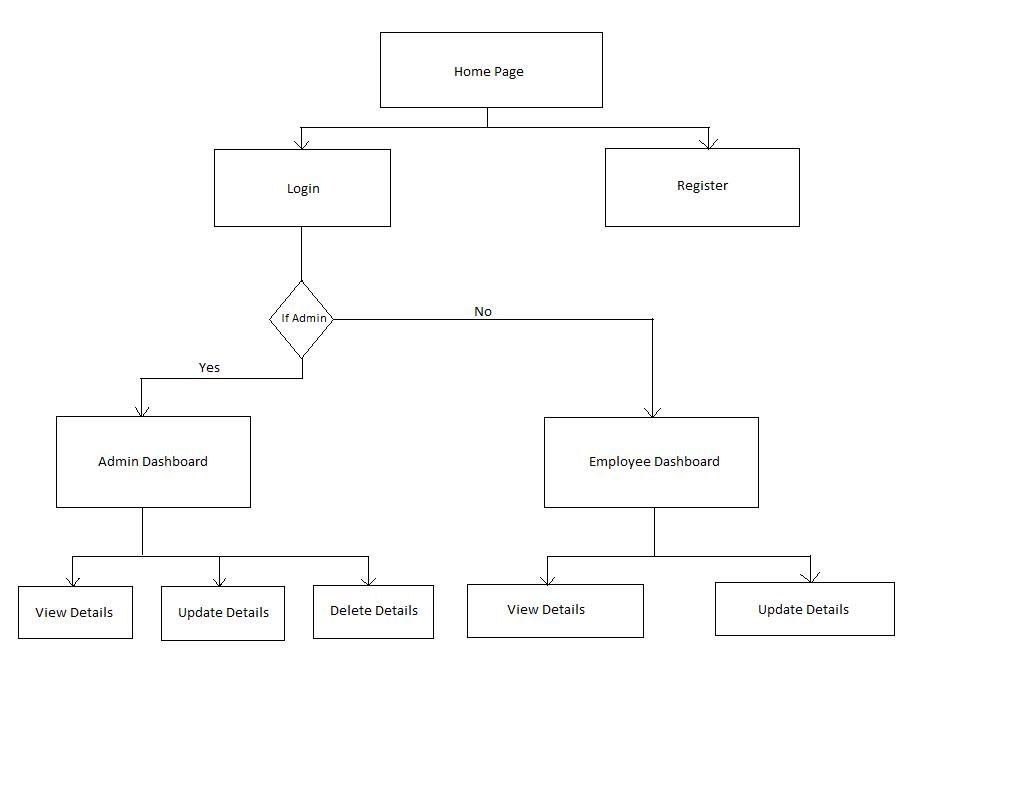
## Document Approval

**Approvers Only – Sign-Off Required**

| Date of Approval | Version # | Document Approver | Role from Sign-off Matrix |
| --- | --- | --- | --- |
|  |  |  |  |
| <Date of approval> | <Ver. Of doc> | <Approver name> | <Role of approval> |
|  |  |  |  |

# Technical Overview

## Integration Solution Design



The application has five major components HomePage, Register, Login, Employee Dashboard, Admin Dashboard. The application built using Angular, Asp.net WebApi and database used is MS SQL. This is a database first application.

## Assumptions

- When User logins to the portal, the data entered during registration is already present in the database.

- Employee will be able to edit only his EmailID, Phone Number and Address.

- Admin will be able to edit the Department and Business Details of Employee.

## Dependencies & Constraints

## Source and Target Applications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | System/  Application Name | System/  Application Details | Version | Operating System |
| 1. | Microsoft SQL Server | Sql Server is used to create the components of the BackEnd Part of the application such as Database, Tables and Procedures. | V12.0.2000.8 | Windows Professional single language |
| 2 | C# ASP.NET WebApi | C# based ASP.NET WebApi is used to regulate the Data transfers between FrontEnd part and BackEnd part of the application. | V2.2 | Windows Professional single language |
| 3 | Angular | Angular framework is used to build the FrontEnd part of the application. It is used to create a very User-Friendly UI to the User. | v8.0 | Windows Professional single language |

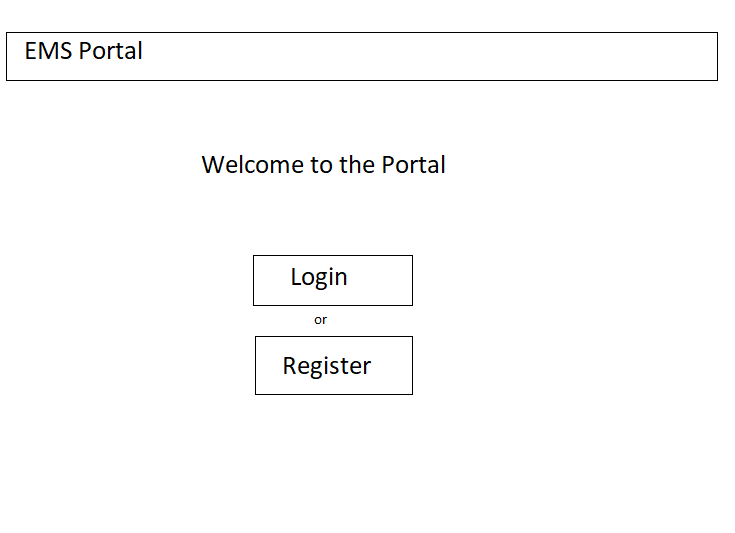
# Integration Component Details

## HomePage Component

When the user first enters the portal, he will see the homepage where he will be able to either login to the portal or register himself into the portal by clicking on the respective buttons.

### Component details

|  |  |
| --- | --- |
| Component Name | HomePage |
| Service Implementation | Display the options to login or register |
| Description | When the user first enters the portal, he will see the homepage where he will be able to either login to the portal or register himself into the portal by clicking on the respective buttons. |
| Input Message | Button Click on Register or Login |
| Response Message | Login Component or Register Component |
| Type of component | Functional |
| Error Message | ER5 |



### Services and Components

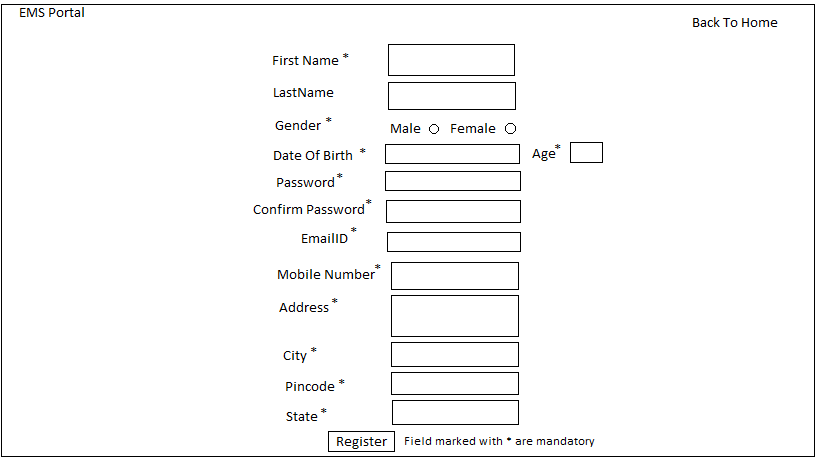
|  |  |  |
| --- | --- | --- |
| Name | Version and location | Description |
| Homepage.component.ts | Inside register component folder in app folder in angular app (EMSPortal) | It will redirect the route to Login Component or Register Component |
| Homepage.component.html | Inside register component folder in app folder in angular app (EMSPortal) | This file will display the options to either login or register. |

## Register Component

User will enter the details like Name, Mobile Number, Email ID, Password, Confirm Password, Employee ID, Address Details like City, State, Pin Code. While the user enters the Password, the system asks him to Enter a Password whose length is not less than 8, and Should contain a Number and Capital Letter. While the user enters other details like Mobile Number,Gender, Age, Email ID, the System should check whether those are in correct format.

### Component details

|  |  |
| --- | --- |
| Component Name | Register |
| Service Implementation | Adding new User as an Employee to the Database. |
| Description | The component will help the user to enter his details It will validate all his details while he enters the details and sends it to the Database. |
| Input Message | First Name, Last Name, Mobile Number, Email ID, Password, Confirm Password, Address, City, Pincode, State. |
| Response Message | Employee Registers Successfully |
| Type of component | Authentication |
| Error Message | ER1,ER2 |



### Services and Components

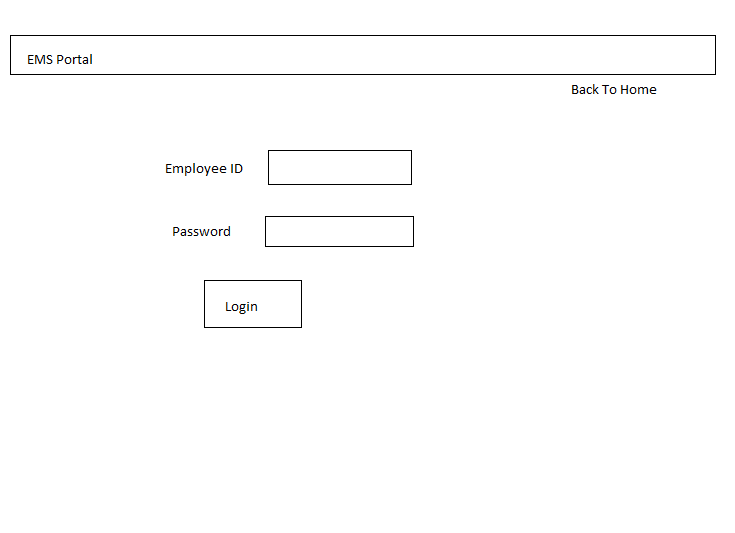
|  |  |  |
| --- | --- | --- |
| Name | Version and location | Description |
| Register.component.ts | Inside register component folder in app folder in angular app (EMSPortal) | It will collect all the details from the user as a model and will call the register service to send it to the WebApi. |
| Register.service.ts | Inside services folder in app folder in EMSPortal Angular project | This Service will send the details to the WebApi Part of the Application. |
| RegisterController.cs | Inside Controllers folder in WebApi Application “EmployeeManagementSystem” | This file will receive the user details from the Angular Part and will store the details in the Database. This file will also will throw error if the user is already registered. |
| register.component.html | Inside register component folder in app folder in angular app (EMSPortal) | This file will show all the form fields to enter the details of the user and validate it. This file also will show whether the error has occurred during validation |

## Login Component

This Component will allow users to enter Employee Id, Password and login as Employee or Admin. The Component will check the System whether the Employee is an admin or not. If the Employee is an Admin, the Component will redirect to the Admin DashBoard. If the Employee is not an Admin, the Component will redirect to the Employee Dashboard.

### Component details

|  |  |
| --- | --- |
| Component Name | Login |
| Service Implementation | Authentication of the User. |
| Description | The Component will allow the user to Login into the portal using Employee Id and Password and will redirect to the respective dashboards based on the Role of the Employee. |
| Input Message | Employee Id, Password |
| Response Message | Display of the Employee/Admin Dashboard |
| Type of component | Authentication |
| Error Message | ER3 |



### Services and Components

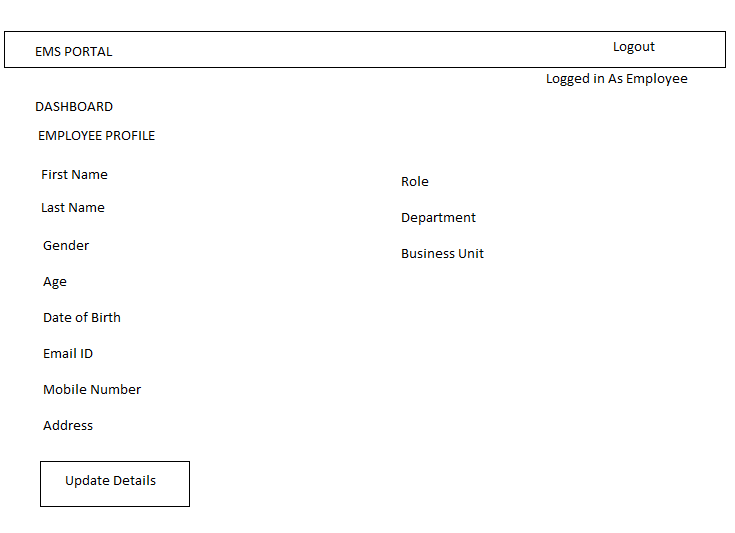
|  |  |  |
| --- | --- | --- |
| Name | Version and location | Description |
| login.component.ts | Inside login component folder in app folder in angular app (EMSPortal) | It will collect all the details from the user as a model and will call the login service to send it to the WebApi. |
| login.service.ts | Inside services folder in app folder in EMSPortal Angular project | This Service will send the details to the WebApi Part of the Application. |
| LoginController.cs | Inside Controllers folder in WebApi Application “EmployeeManagementSystem” | This file will receive the user details from the Angular Part and will store the details in the Database. This file will also will throw error if the user is not registered. |
| login.component.html | Inside logincomponent folder in app folder in angular app (EMSPortal) | This file will show all the form fields to enter the details of the user and validate it. This file also will show whether the error has occurred during validation |

## Employee DashBoard Component

This Component will allow the Employee to view his Details like Employee Id, Name, Address, EmailID, Age, Gender, Phone Number, Role, Department Name, Business Unit Name and will also allow the Employee to edit details like Email ID, Phone Number and Address.

### Component details

|  |  |
| --- | --- |
| Component Name | Employee Dashboard |
| Service Implementation | View Employee Details and Update certain Details. |
| Description | This Component will allow the Employee to view his Details like Employee Id, Name, Address, EmailID, Age, Gender, Phone Number, Role, Department Name, Business Unit Name and will also allow the Employee to edit details like Email ID, Phone Number and Address. |
| Input Message | Email ID, Phone Number and Address. |
| Response Message | Display of the Employee/Admin Dashboard and If correct details are provided for updation, the component will display “Update credentials succssfully” |
| Type of component | Functional |
| Error Message | ER4 |



### Services and Components

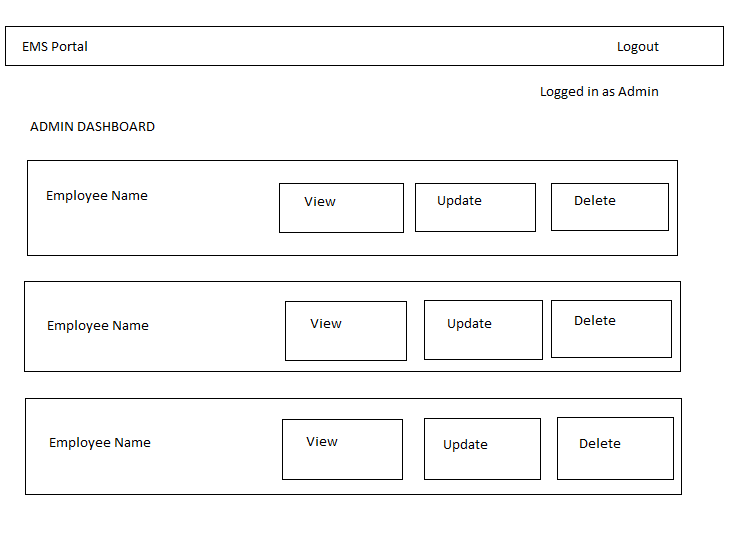
|  |  |  |
| --- | --- | --- |
| Name | Version and location | Description |
| employeedashboard.component.ts | Inside employeedashboard component folder in app folder in angular app (EMSPortal) | It will call the service which will get all the user details from the WebApi and store it in the object. |
| employeedashboard.service.ts | Inside services folder in app folder in EMSPortal Angular project | This Service will send a Get Request to the WebApi and acquire the HttpResponse Message from the WebApi. |
| EmployeeDashboardController.cs | Inside Controllers folder in WebApi Application “EmployeeManagementSystem” | This file will get the details of the Employee from the Database and return as HttpResponseMessage.  This file also get the details from the Angular App and update |
| employeedashboard.component.html | Inside logincomponent folder in app folder in angular app (EMSPortal) | This file will unpack the model and display the details of the employee to the user in a readable manner |
| Updatedetails.service.ts | Inside services folder in app folder in EMSPortal Angular project | This file will send all the details entered by the user in the Dashboard and send it to webapi for updation in the database. |

## Admin DashBoard Component

This Component will allow the Admin to view all the Employees’ Details like Employee Id, Name, Address, EmailID, Age, Gender, Phone Number, Role, Department Name, Business Unit Name and will also allow the Employee to edit details like Role, Department name and Business Unit Name and also delete an Employee’s entire data.

### Component details

|  |  |
| --- | --- |
| Component Name | Admin Dashboard |
| Service Implementation | View Employee Details and Update certain Details. |
| Description | This Component will allow the Admin to view Employees’ Details like Employee Id, Name, Address, EmailID, Age, Gender, Phone Number, Role, Department Name, Business Unit Name and will also allow the Employee to edit details like Role, Department name and Business Unit Name and also delete an Employee’s entire data. |
| Input Message | Role, Department name and Business Unit Name and Delete Employee |
| Response Message | Display of the Employee/Admin Dashboard and If correct details are provided for updation, the component will display “Update credentials succssfully”. If delete button is clicked for an Employee, the component will ask for a confirmation messag to delete the data. If Yes is clicked, the data will be deleted. If No is clicked, the data will remain as such and the operation of deletion will be rolled back. |
| Type of component | Functional |
| Error Message | ER3 |



### Services and Components

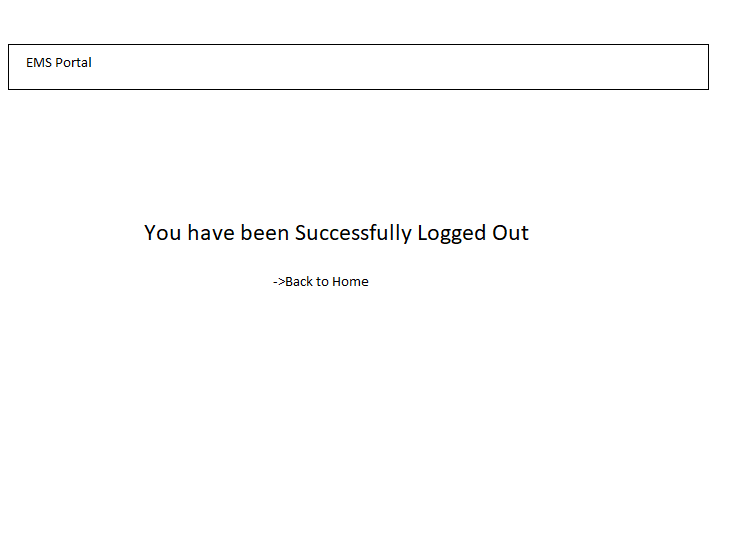
|  |  |  |
| --- | --- | --- |
| Name | Version and location | Description |
| admindashboard.component.ts | Inside admindashboardcomponent folder in app folder in angular app (EMSPortal) | It will call the service which will get all the user details from the WebApi and store it in the object. |
| admindashboard.service.ts | Inside services folder in app folder in EMSPortal Angular project | This Service will send a Get Request to the WebApi and acquire the HttpResponse Message from the WebApi. |
| AdminDashboardController.cs | Inside Controllers folder in WebApi Application “EmployeeManagementSystem” | This file will get the details of the Employee from the Database and return as HttpResponseMessage.  This file also get the details from the Angular App and update |
| admindashboard.component.html | Inside logincomponent folder in app folder in angular app (EMSPortal) | This file will unpack the model and display the details of the employee to the user in a readable manner |
| Updatedetails.service.ts | Inside services folder in app folder in EMSPortal Angular project | This file will send all the details entered by the user in the Dashboard and send it to webapi for updation in the database. |
| Deletedetails.service.cs | Inside services folder in app folder in EMSPortal Angular project | This file will send the Employee id to the WebApi for which the details have to deleted . |

## Log Out Component

Logout component helps users to logout of their account and get redirected to logout successful page. When user clicks on the Logout button, the component action will be performed.

### Component details

|  |  |
| --- | --- |
| Component Name | Logout |
| Service Implementation | Logs out of the application |
| Description | Logged in user can logout of the application if user wants using the functional button provided in the application |
| Input values | Log Out Event |
| Response Message | Redirected to Logout page where a message will be shown as “Logout Successfull” |
| Type of component | Authentication |
| Error Message | ER4 |



### Services and Components

|  |  |  |
| --- | --- | --- |
| Name | location | Description |
| Logout.component.html | Inside logout component folder which is in the app folder within the project folder. | Displays a message as “Logout Successful” and link to return to the home page. |

1. The intent of this section is to record the Testing requirements. Reference the Testing Requirements (TE.005).
2. The intent of this section is to include the Services Portfolio. Individual SOA Services may be referenced by the Use Cases. Reference the **Service Portfolio (**RA.025).

# Error Handling Notification and Reporting

## Technical & Business error

|  |  |  |
| --- | --- | --- |
| **NO** | **Error** | **Message** |
| **ER1** | Password error | Password does not meet the criteria of lngth greater that 8, contain a Capital Letter and a number. Password and Confirm Password does not matc |
| **ER2** | Employee already registered | Already Registered? Login |
| **ER3** | Invalid Employee No & password | Please Enter the correct Employee Id or Password |
| **ER4** | Updation Failed | Please enter correct values. |
| **ER5** | Unexpected Error | Some problem occurred while processing your request |

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| XSD | XML Schema Definition |
| XML | Extensible Markup Language |
| XSL | Extensible Style sheet Language |
| WSDL | Web Service Definition Language |
| FMW | Fusion Middleware |
| BPEL | Business Process Execution Language |
| OSB | Oracle Service Bus |
| EBS | Oracle E-business Suite |