**Technical Design Document**

**Web Application**

**Document Version: 1.0**

Contents

[1. DOCUMENT CONTROL 3](#_Toc109141527)

[1.1 REVISION HISTORY 3](#_Toc109141528)

[1.2 REQUIRED APPROVERS 3](#_Toc109141529)

[2. INTRODUCTION 4](#_Toc109141530)

[3. APPLICATION OVERVIEW 4](#_Toc109141531)

[4. TECHNICAL ASSUMPTIONS, DEPENDENCIES, CONSTRAINTS AND RISKS 4](#_Toc109141532)

[5. TECHNICAL DESIGN OVERVIEW 5](#_Toc109141533)

[5.1 SERVER MIGRATION TO LIBERTY 5](#_Toc109141534)

[5.2 SPRING MIGRATION 5](#_Toc109141535)

[5.3 SAVEAS GRID FUNCTIONALITY 6](#_Toc109141536)

[5.4 JAVA 8 UPGRADE 6](#_Toc109141537)

[5.5 SYSTEM FRAMEWORK FLOW DIAGRAM 6](#_Toc109141538)

[5.5.1 CURRENT FLOW 6](#_Toc109141539)

[5.5.2 FUTURE FLOW 7](#_Toc109141540)

[5.5.3 SEQUENCE DIAGRAM 9](#_Toc109141541)

[5.5.4 USER INTERFACES 14](#_Toc109141542)

[5.6 OBJECT-ORIENTED DESIGN APPROACH 17](#_Toc109141543)

[5.6.1 UI CHANGES 17](#_Toc109141544)

[5.6.2 JAVA CHANGES 18](#_Toc109141545)

[5.6.3 CONFIG FILES 20](#_Toc109141546)

[5. TRACEABILITY 20](#_Toc109141547)

[6. APPENDIX 20](#_Toc109141548)

[6.1 DEFINITIONS, ACRONYMS, ABBREVIATIONS: 20](#_Toc109141549)

# DOCUMENT CONTROL

## REVISION HISTORY

| Revision History | | | | |
| --- | --- | --- | --- | --- |
| Version  No. | Details of Change | Changed Sections | Prepared by | Date |
|  |  |  |  |  |
|  |  |  |  |  |

## REQUIRED APPROVERS

|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# INTRODUCTION

Web application comes under the scope of Global Currency Factory program for the following migration/updrage for web & batch component

1.Migrating the server from Websphere Application server 8.5.5.13 to Liberty 20

2.Migrating the MVC framework from JSF to Spring MVC (only for Web component)

3.Upgrade the application from Java 6 to Java 8  
4.Modify the Save Grid As functionality. Rewrite the code from JXL and f1j11swing API to Apache POI API

The purpose of the document is to cover the challenges and detailed information of Migration /Upgrade of the Web application from technical and functional aspects

# APPLICATION OVERVIEW

This is a middleware application used in the processing of payments between different source systems and banks. It is mainly used to transfer funds from source (Partner Bank) to destination (Beneficiary). This application acts a middleware between App2 and Bank. All payments come from App2 to App1 and then it goes out to the Bank. It is supported for multiple country and currencies. It is a legacy application and has both web and batch component.

Web component has the following UI screens

* Table Maintenance Screen-Maintain DB lookup data
* Report-Connect to cognos site to retrive data for reports

App1 Batch Component have the following Upstream system

* ACE
* TWS
* AOC
* SSB

# TECHNICAL ASSUMPTIONS, DEPENDENCIES, CONSTRAINTS AND RISKS

|  |  |
| --- | --- |
| Assumptions | *1.In future, Web Application will be compatible only in Internet Explorer version 10 onwards preferably IE 11.*  *2.Swift Message Application screen will be completely removed as it is confirmed by business that the functionality is not used in production.* |
| Dependencies | *1.APPD configurations will be provided by techops team*  *2.SSL Configuration will be provided by techops team*  *3.SiteMinder & SSO configuration will be provided by techops team* |
| Constraints |  |
| Risks | *1.Since the Web UI has JSF component to handle each and every field in the UI the look and feel of the screen might differ slightly from current world* |

# TECHNICAL DESIGN OVERVIEW

## 5.1 SERVER MIGRATION TO LIBERTY

Migration of application to liberty involve the following steps /procedures

* Execute the IBM Migration Toolkit for the application to evaluate the compatibility of the application in the liberty server
* Attached the IBM Migration Toolkit scan report for both web and batch component in the Appendix
* Batch component can be migrated to Liberty just configuration change without any code changes complication
* Web component has issues with the following jars and relevant code change

|  |  |  |  |
| --- | --- | --- | --- |
| **Sno** | **Current Jars** | **Functional Purpose** | **Mitigation Plan** |
| 1 | Myfaces-api-2.0.24 | Java Server Faces Specification | Spring Framework Migration |
| 2 | Myfaces-bundle-2.0.24 | Java Server Faces Specification |
| 3 | Myfaces-impl-2.0.24 | JSF Component Rendering |
| 4 | Richfaces-ui-3.3.3.Final | UI fields Style & Display |
| 5 | Richfaces-api-3.3.3.Final | UI fields Rendering |
| 6 | Richfaces-impl-3.3.3.Final | UI fields Rendering |
| 7 | Jsf-ibm-1.0.0 | JSF Widget Library providing input, output & navigation components in JSP |
| 8 | Odc-jsf-1.0.0 | JSF Widget Library providing input, output & navigation components in JSP |
| 9 | Jxl-2.6.12 | Read and write to excel | Rewrite excel handling with Apache POI API |
| 10 | f1j11swing-1.0.0 | Use of JBook class to copy from one excel to another | Rewrite excel handling with Apache POI API |

* For Liberty server, all the server related details are configured in the following files
* **Server.xml** - JDBC Confirgurations, Server Features, Logging, EAR mapping, SSL & Keystore configuration are defined
* **Server.env** – Environment variables are specified
* **jvm.options** – JVM Options/Arguments are specified
* **bootstrap.properties** – Server Startup variables/arguments are defined

## 5.2 SPRING MIGRATION

* JSF framework being an outdated technology and upgrading JSF leads to various challenges like compatibility issue. Web application alone will be migrated to Spring framework of 5.3 version.
* Web application is used only by business and has 2 major screens with multiple functionalities as below
* Along with Spring Migrations, the following activities will be implemented
* Remove the unused functionalities -Swift screen and its impacted code,
* Convert Main Screen tab functionality as tabledriven. In current world, tab switch is controlled in code

## 5.3 SAVEAS GRID FUNCTIONALITY

In current application, F1J11swing jar is used to copy data from temporary excel file to final excel file due to read issue in JXL jar. After migration, the usage of f1j11swing jar and jxl jar will be completely removed and Apache POI will be used for reading and writing to excel file.

## 5.4 JAVA 8 UPGRADE

No changes needed in Java 8 upgrade of TMW batch application. However, in TMW web component, we will be using Java 8 features while migrating to Spring framework.

## 5.5 SYSTEM FRAMEWORK FLOW DIAGRAM

### 5.5.1 CURRENT FLOW

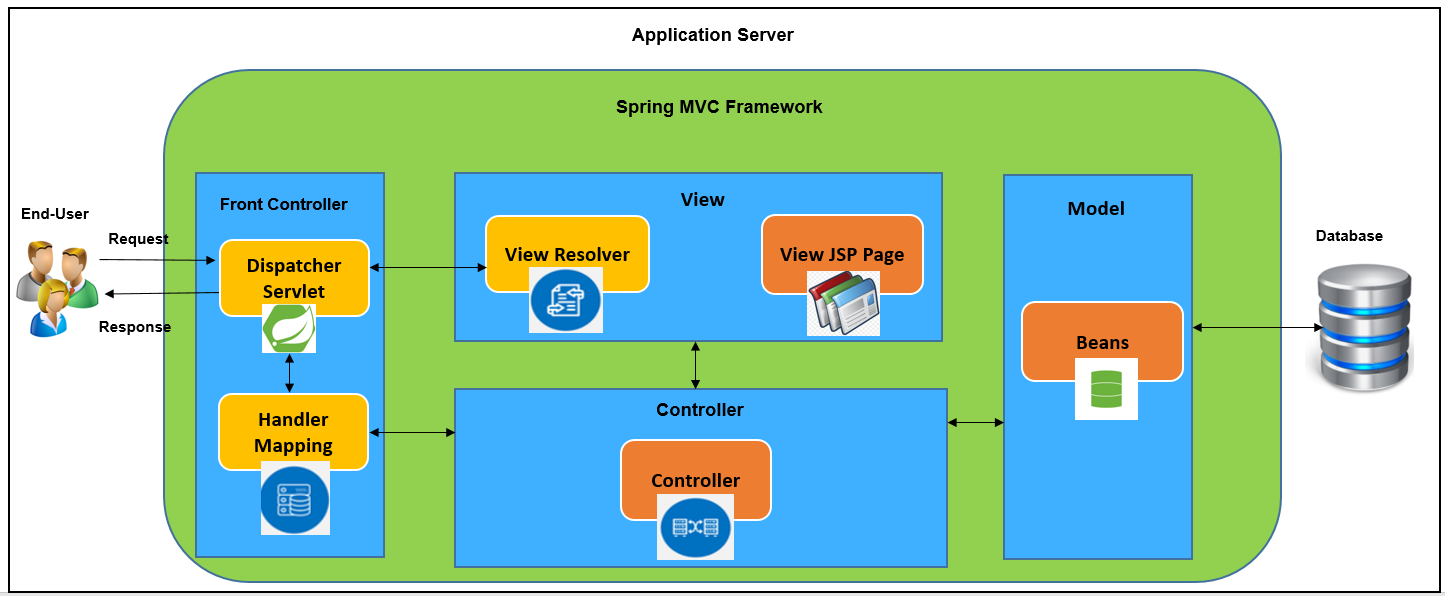
* Currently web application runs with JSF 1.2 [Java Sun RI Implementation] in Websphere 8.5.5.13. To run the application in liberty/websphere 9, the application needs to be completely migrated to JSF 2.0 [Myfaces 2.0 implementation]
* In web component, the following mix of jars are used as part of JSF implementation:
  + IBM JWL [Java Widget Library includes jsf-ibm-1.0.0.jar and odc-jsf-1.0.0.jar]
  + Richfaces [richfaces-ui-3.3.3.Final.jar, richfaces-impl-3.3.3.Final.jar, richfaces-api-3.3.3.Final.jar]
  + Apache Myfaces [myfaces-2.0.24].

These jars currently follow JSF 1.2 implementation. So, these must be upgraded to support JSF 2.0 implementation.

* But there is a challenge in upgrading these jars due to the below dependency issue:

Richfaces must be upgraded to its latest version 4.0 to support JSF 2.0, but we need to migrate all jsp to facelets because richfaces 4.0 does not support jsp. On the other hand, IBM JWL is deprecated and does not support facelets.

### 5.5.2 FUTURE FLOW



**Note:**

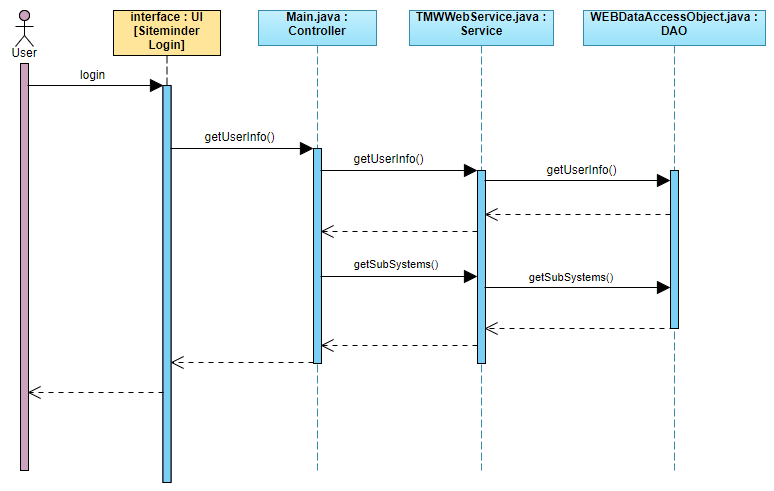
**User Defined Classes**

**Spring classes**

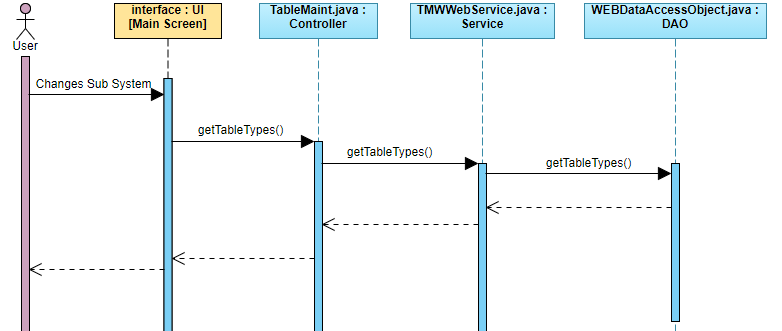
* Spring dependency jars will be included with the ear file
* Spring-webmvc-5.3.8
* Spring-jdbc-5.3.8
* Jstl-1.2
* javax.servlet-api-4.0.1
* Register a DispatcherServlet which is the Front Controller and also place the servlet mappings in the web.xml
* Also remove the existing param configs for IBM JSF jars and richfaces jars from web.xml.
* Remove the faces-config.xml and create a spring-servlet.xml to configure the beans and datasource.
* Modify the below classes to act as Spring MVC Controller Classes with annotations [@Controller] to handle the requests for different functionalities in web application as follows:
* Main.java
* TableMaint.java
* Report.java
* RowMaint.java
* Modify existing DAO class to use the Spring framework JDBCTemplate to query the database.
* Create an interface for DAO class and make the existing WEBDataAccessObject to implement it. This interface will contain all the methods to perform DB operations in Web.
* Create a Service class –WebService to act as a Service layer between the Controller and DAO.
* Modify existing JSP by removing the usage of jsf jars and replace them with spring & jstl jars.
* All the jsp pages need to be rebuilt as currently the components are built in using the jsf jars.
* Use JSTL expression language to capture the field values in the jsp.
* Use Jquery to perform client-side scripting.
* Integrate Jquery Ajax functionality to get details on the web page without page refresh.
* Add the below Jackson jars to convert objects to json and pass it in ajax response.
* jackson-core - 2.12.3
* jackson-databind - 2.12.3
* Remove the usage of Jexcel and f1j11swing jars from Save Grid As functionality by using the apache-poi jar to read and write to excel sheet

### 5.5.3 SEQUENCE DIAGRAM

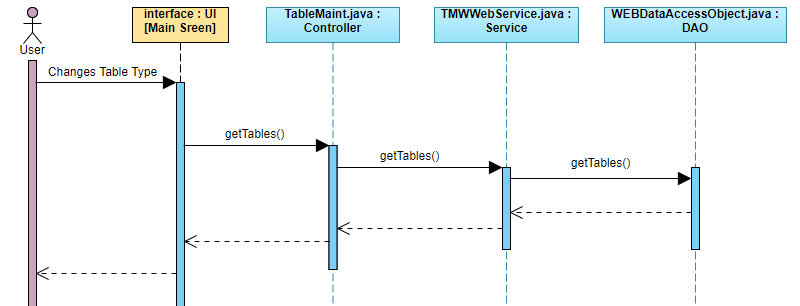
**Login:**



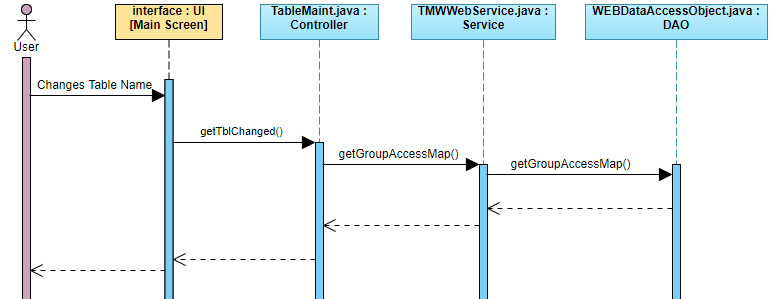
**SubSystem Selection:**



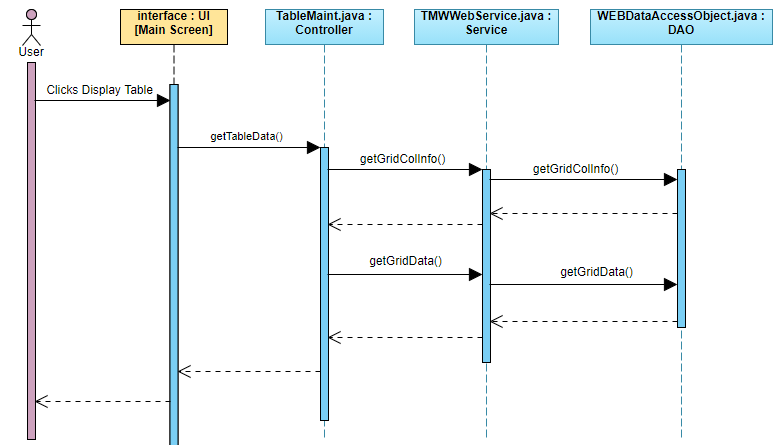
**Table Type Selection:**



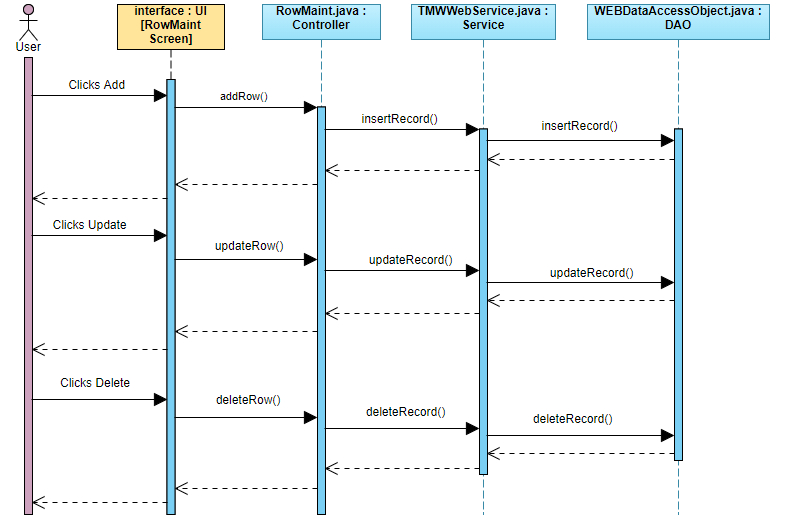
**Table Selection:**



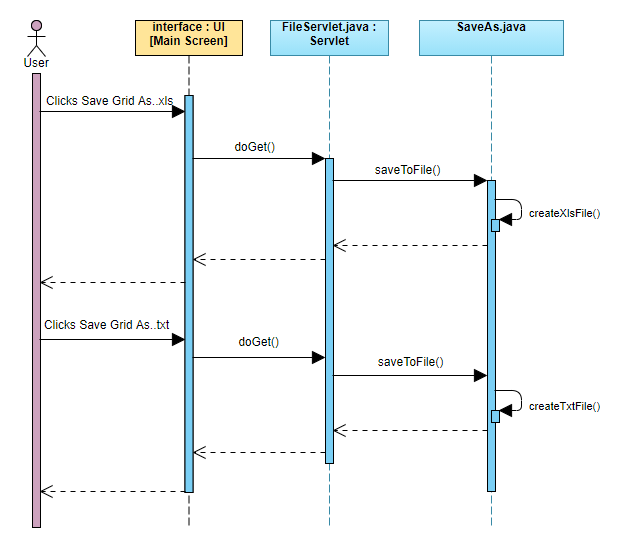
**Display Table:**



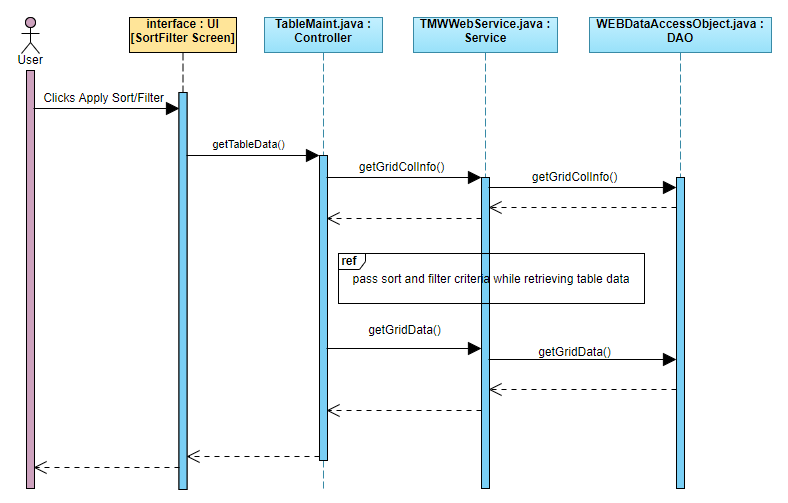
**Add/Edit/Delete Actions:**

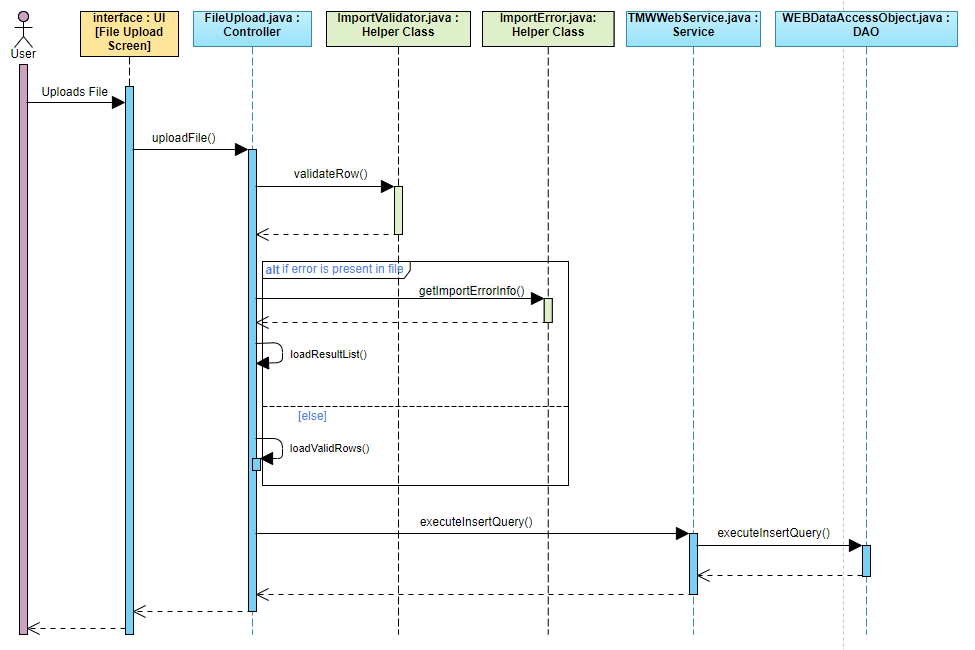


**Save Grid As:**

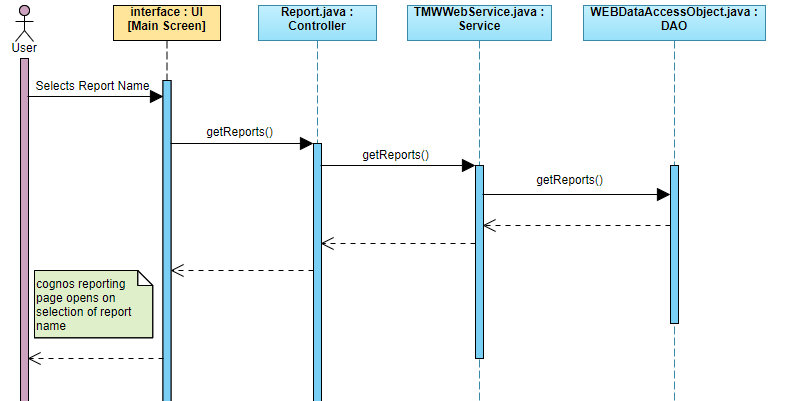


**Sort/Filter:**



**File Import:**

**Report:**



**Note:**

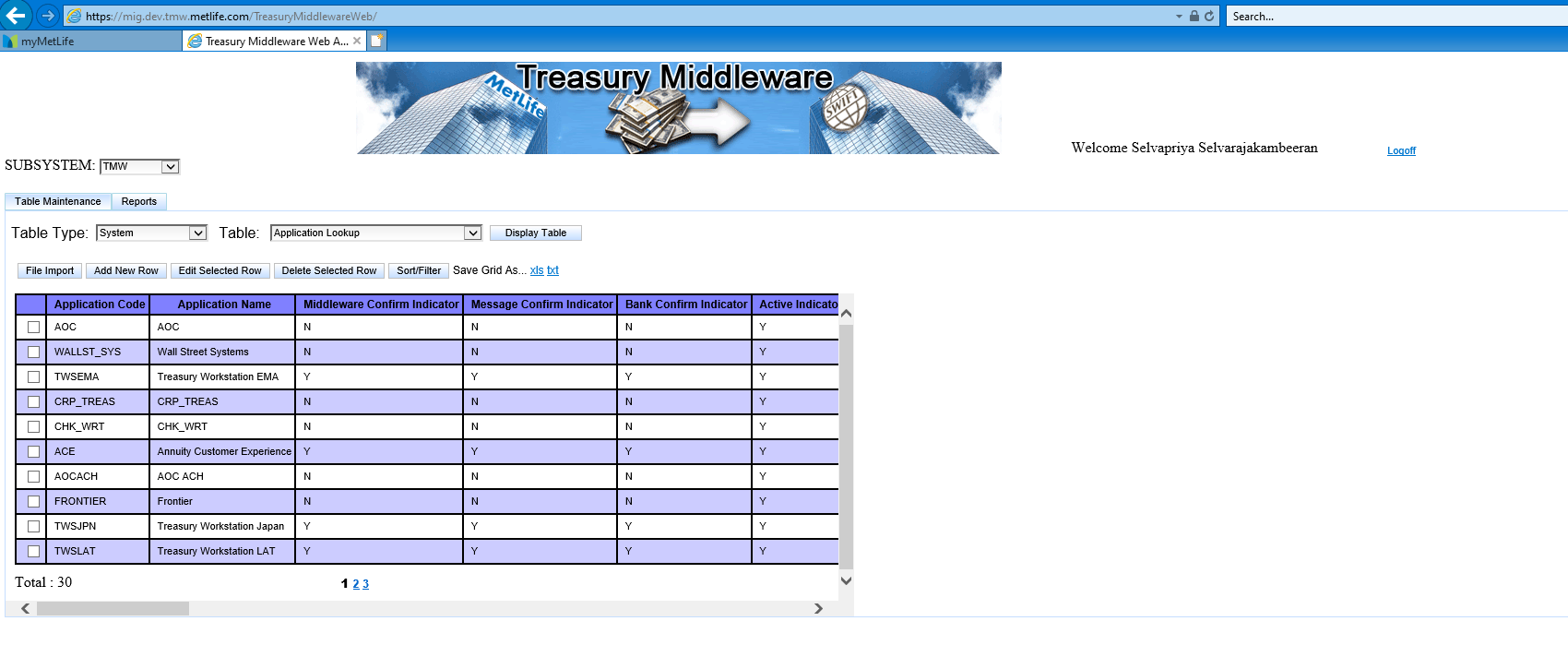
Denotes Interface

Denotes Classes to be reused

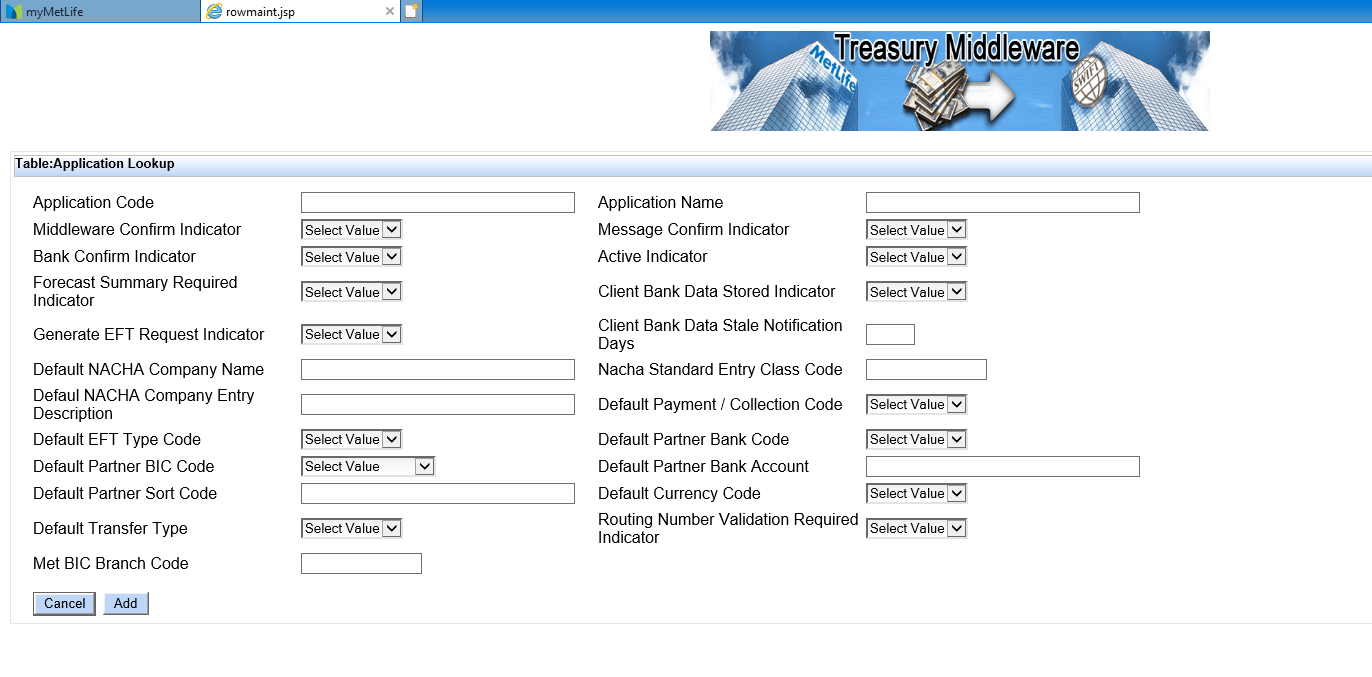
Denotes Classes to be modified as part of Spring migration

### 5.5.4 USER INTERFACES

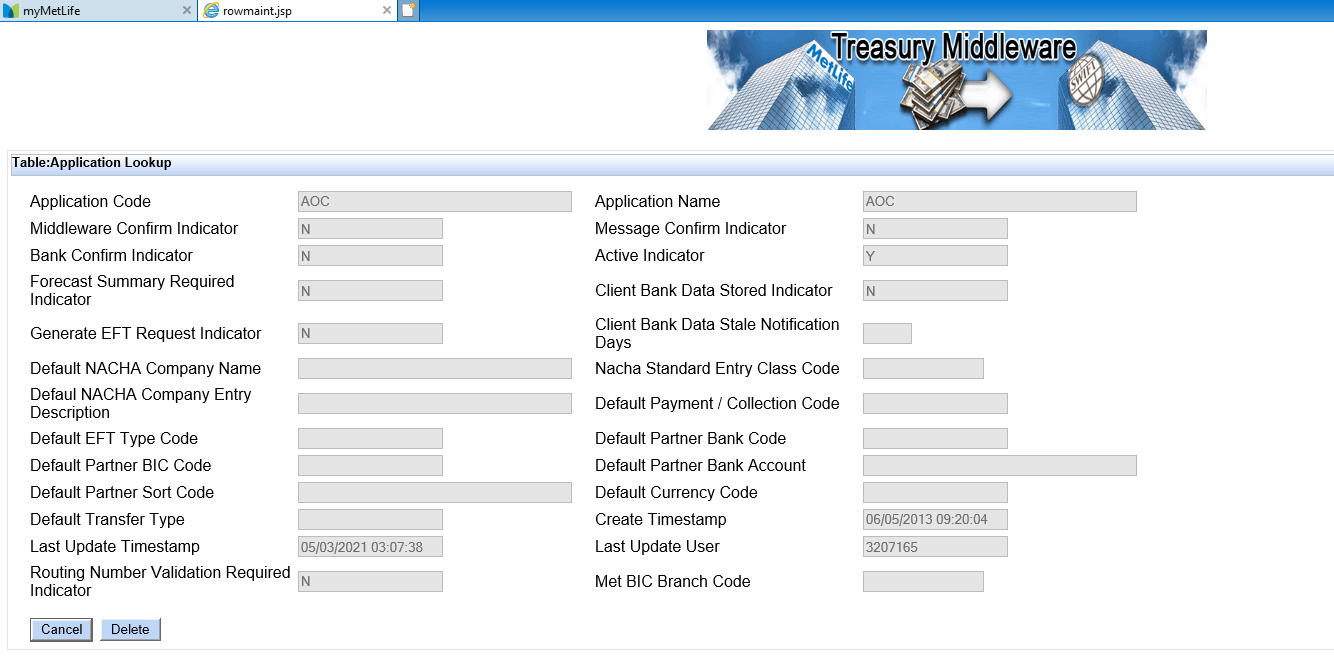
**Main Screen:**



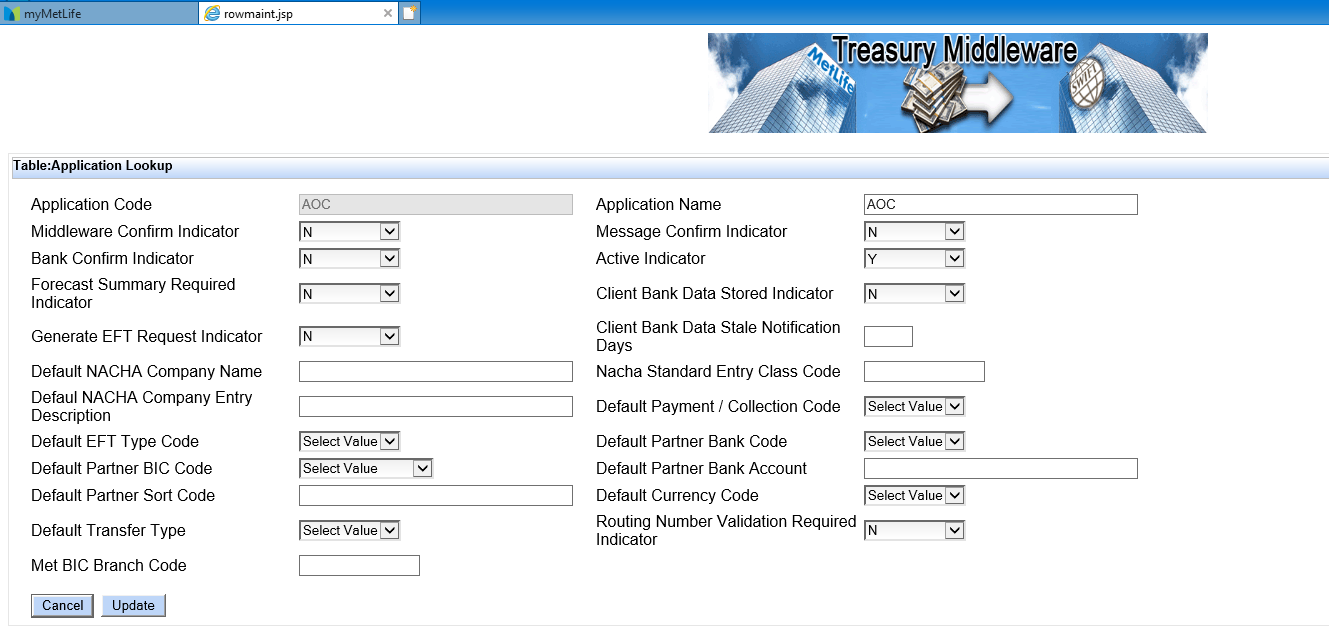
**Add New Row:**



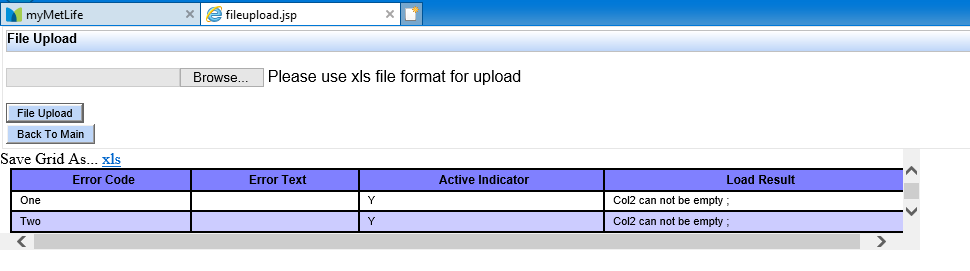
**Delete Row:**



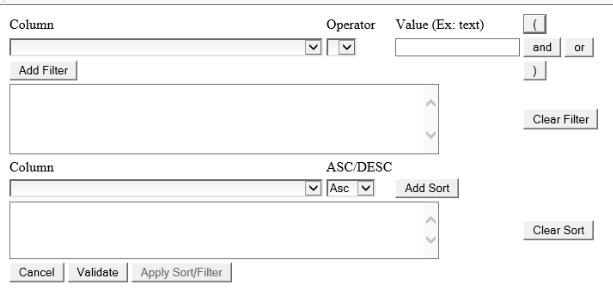
**Update Row:**



**File Import:**



**Sort/Filter:**



## 5.6 OBJECT-ORIENTED DESIGN APPROACH

### 5.6.1 UI CHANGES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sno** | **FileType** | **FileNmaes** | **Functionality** | **Code Implementation** |
| 1 | JSP | fileupload.jsp | Spring Migration | Modfiy the jsp by removing the usage of jsf tags and replace with spring, jstl & jsp tags. Try to maintain the look and feel of the components as same as possible in current jsp |
| 2 | JSP | error.jsp | Spring Migration | Modfiy the jsp by removing the usage of jsf tags and replace with html tags |
| 3 | JSP | main.jsp | Spring Migration | Modify all the elements in the jsp like tabs, dropdowns, tables and buttons etc. by removing the usage of jsf tags with spring, jstl & jsp tags. Try to maintain the look and feel of the components as same as possible in current jsp |
| 4 | JSP | noresult.jsp | Unused Functionality Removal | Remove this jsp as we are going to remove the Swift Messaging functionality |
| 5 | JSP | result.jsp | Unused Functionality Removal | Remove this jsp as we are going to remove the Swift Messaging functionality |
| 6 | JSP | rowmaint.jsp | Spring Migration | Modify all the elements in the jsp like tabs, dropdowns, tables and buttons etc. by removing the usage of jsf tags with spring,jstl & jsp tags. Try to maintain the look and feel of the components as same as possible in current jsp |
| 7 | JSP | sortFilter.jsp | Spring Migration | Modify all the elements in the jsp like tabs, dropdowns, tables and buttons etc. by removing the usage of jsf tags with spring,jstl & jsp tags. Try to maintain the look and feel of the components as same as possible in current jsp |
| 8 | JSP | tablemaint.jsp | Unused Functionality Removal | Remove this jsp as it is not used in any existing functionality |
| 9 | JSP | logoff.jsp | Spring Migration | Modfiy the jsp by removing the usage of jsf tags and replace with html/jsp tags |
| 10 | JSP | index.jsp | Spring Migration | Modify and replace existing jsf page to jsp page |
| 11 | JSP | snoop.jsp |  | No change |
| 12 | JSP | logoff.jsf | Spring Migration | Modfiy the jsp by removing the usage of jsf tags and replace with html/jsp tags |
| 13 | JSP | keepalive.jsp | Unused Functionality Removal | No change |
| 14 | CSS | Main.css | Spring Migration | Newly added stylesheet file for Spring migration. This has the styles for all the components in the main screen |
| 15 | CSS | Master.css |  | Reuse |
| 16 | CSS | stylesheet.css |  | Remove these files as they were used for JSF components. Not needed in Spring migration |
| 17 | CSS | swift.css |
| 18 | CSS | tablegrid.css |
| 19 | CSS | tabpanel.css |
| 20 | JS | fileupload.js |  | Newly added script file for FileUpload functionality |
| 21 | JS | main.js |  | Newly added script file for Main Pge functionality |
| 22 | JS | rowmaint.js |  | Newly added script file for Add/Edit/Delete functionality |
| 23 | JS | sessiontimeout.js |  | Newly added script file for Session Timeout functionality |
| 24 | JS | sortfilter.js |  | Newly added script file for SortFilter functionality |

### 5.6.2 JAVA CHANGES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sno** | **FileType** | **FileNames** | **Functionality** | **Code Implementation** |
| 1 | Java | FileUploadController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 2 | Java | MainController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 3 | Java | ReportsController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 4 | Java | RowMaintController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 5 | Java | SortFilterController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 6 | Java | TableMaintController.java | Spring Migration | Rename the existing file as per Spring Standards by adding suffix Controller. Modfiy this file as a spring controller and rewrite the actions/methods removing the usage of JSF classes and modfiy it according to spring and java 8. |
| 7 | Java | DAOBean.java | Spring Migration | Modify it by removing the usage of faces classes and replacing with java & spring classes |
| 8 | Java | Error.java | Spring Migration | Modify it by removing the usage of faces classes and replacing with java & spring classes. This class will handle the 400/401/404/500 errors in the application |
| 9 | Java | Constants.java | Spring Migration | Remove it as the functionality |
| 10 | Java | FileServlet.java | Spring Migration | Reuse and modify it based on Spring migration and perform code cleanup |
| 11 | Java | SessionLogoffFilter.java | Spring Migration | Remove it as the functionality |
| 12 | Java | SessionTimeoutFilter.java | Spring Migration | Remove it. Session Timeout will be handled in javascript |
| 13 | Java | SessionListener.java | Spring Migration | Remove it as the functionality will be handled by a single SessionFilter in spring migration |
| 14 | Java | TMWConstants.java | Spring Migration | Reuse and add new constants when needed while migrating to Spring framework |
| 15 | Java | AppProps.java | Spring Migration | Remove the JNDI datasource configuration as this configuration will be moved to spring-servlet.xml as part of spring migration. Also remove the changes done to show a UI for the user to change the datasource and logger properties on the fly. |
| 16 | Java | WEBDataAccessObject.java | Spring Migration | Modify all the methods and rewrite the database calls using spring jdbctemplate instead of JDBCManager. |
| 17 | New Java class | TMWWebServicejava | Spring Migration | Inject the WEBDataAccessObject.java bean and invoke the dao calls to Database. |
| 18 | New Java class | IWebDAO.java | Spring Migration | New Interface for DAO |
| 19 | New Java class | IWebService.java | Spring Migration | New Interface for Service |
| 20 | Bean Java | All the bean clases | Spring Migration | Spring Annotaion will be included if required |
| 21 | Java | SaveAs.java | Apache POI API | Remove the usage of existing f1j11Swing jar and replace the usage of JXL jar with apache POI for writing data to excel. |
| 22 | Java | SwiftMessage.java | Unused Functionality Removal | Remove this class as Swift Messging functionality is going to be removed completely from the application |
| 23 | Java | JDBCManager.java | Unused Functionality Removal | Remove it as we are going to use Spring JDBCTemplate for database operations |
| 24 | Java | DataSourceProvider.java | Unused Functionality Removal | Remove it as we are going to use Spring JDBCTemplate for database operations |
| 25 | Java | FtpUtility.java | Unused Functionality Removal | This class is used in Swift Messaging functionality. Since we are going to remove this Swift message, we can remove this class also |
| 26 | Java | PageCodeBase.java | Unused Functionality Removal | Remove it as it is no longer needed as the project is going to be migrated to spring framework |
| 27 | Java | JSFComponentBuilder.java | Unused Functionality Removal | Remove it as it is no longer needed as the project is going to be migrated to spring framework |
| 28 | Java | SessionManagement.java | Unused Functionality Removal | Remove this class as we will not be using JSF FacesContext session after migrating to spring |
| 29 | Java | NextProcessDate.java | Unused Functionality Removal | Remove it as it is not used any existing functionality |
| 30 | Java | CacheControlPhaseListener.java | Unused Functionality Removal | Remove it as it is no longer needed as the project is going to be migrated to spring framework |
| 31 | Java | ButtonListener.java | Unused Functionality Removal | Remove it as it is no longer needed as the project is going to be migrated to spring framework |
| 32 | Java | SkinBean.java | Unused Functionality Removal | Remove it as it is no longer needed as the project is going to be migrated to spring framework |
| 33 | Java | SessionPanelBean.java | Unused Functionality Removal | Remove it as it is not used any existing functionality |
| 34 | Java | ProcessDateBean.java | Unused Functionality Removal | Remove it as it is not used any existing functionality |
| 35 | Java | GenEftRqstDataBean.java | Unused Functionality Removal | Remove it as it is not used any existing functionality |
| 36 | Java | EftRqstDtlBean.java | Unused Functionality Removal | Remove it as it is not used any existing functionality |
| 37 | Java | Approval.java | Unused Functionality Removal | Remove this file as this functionality is not used. |
| 38 | Java | GenerateEftRequest.java | Unused Functionality Removal | Remove this file as this functionality is not used. |
| 39 | Java | OFACOverrideClient.java | Unused Functionality Removal | Remove this file as this functionality is not used. |
| 40 | Java | OFACOverrideEftRqst.java | Unused Functionality Removal | Remove this file as this functionality is not used. |
| 41 | Java | TWSExternalDAO.java | Unused Functionality Removal | Remove this file as this functionality is not used. |
| 42 | New Java File | TableNameBean.java | Spring Migration | New POJO class added for displaying list of table names in Spring migration |
| 43 | Java | ImportErrorInfo.java | Spring Migration | Reuse |
| 44 | Java | ImportErrors.java | Spring Migration | Reuse |
| 45 | Java | ImportExportColumnInfo.java | Spring Migration | Reuse |
| 46 | Java | ImportValidator.java | Spring Migration | Reuse |
| 47 | Java | PropertiesManager.java | Spring Migration | Reuse |
| 48 | Java | StringUtils.java | Spring Migration | Reuse and added methods for esapi encoding |

### 5.6.3 CONFIG FILES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sno** | **FileType** | **FileNmaes** | **Functionality** | **Code Implementation** |
| 1 | Config File | web.xml | Spring Migration | Remove JSF Configuration and add Spring Configuration |
| 3 | Config File | tmwweb-web/pom.xml | Spring Migration, Apache POI | Remove JSF Configuration, Swing jar and add Spring Configuration jars |
| 4 | Config File | tmwweb-public/pom.xml | Spring Migration | Remove JSF Configuration jar |
| 5 | Config File | ibm-web-bnd.xml |  | No Change |
| 6 | Config File | ibm-web-ext.xml |  | No Change |
| 7 | Config File | faces-config.xml | Unused Functionality Removal | Remove this file as it is used for JSF |
| 8 | Config File | Spring-servlet.xml | Spring Migration | New configuration file to map beans, resources and datasource in spring |
| 9 | Config File | Esapi.properties | Spring Migration | Newly added for performing Esapi encoding |
| 10 | Config File | Log4j2.properties | Spring Migration | Log4j2 Configuration file newly added for logging |

# TRACEABILITY

# APPENDIX

## DEFINITIONS, ACRONYMS, ABBREVIATIONS: