**VINOTHINI S**

Codebase implemetation

Table of Contents

[**1.** **Technical Overview** 2](#_Toc109250882)

[**1.1.** **Migration Report** 2](#_Toc109250883)

[**1.2.** **Architecture Context Diagram** 2](#_Toc109250884)

[**1.3.** **Package Structure** 3](#_Toc109250885)

[**1.4.** **File Control** 5](#_Toc109250886)

[**1.4.1.** **User Interface** 5](#_Toc109250887)

[**1.4.2.** **Code Flow Implementation** 5](#_Toc109250888)

[**1.4.3.** **Functionality** 6](#_Toc109250889)

[**1.5.** **Table Maintenance Tab** 7](#_Toc109250890)

[**1.5.1.** **User Interface** 7](#_Toc109250891)

[**1.6.** **Demote File Tab** 7](#_Toc109250892)

[**1.6.1.** **User Interface** 8](#_Toc109250893)

[**1.6.2.** **Code Flow Implementation** 8](#_Toc109250894)

**Treasury Receipt Station**

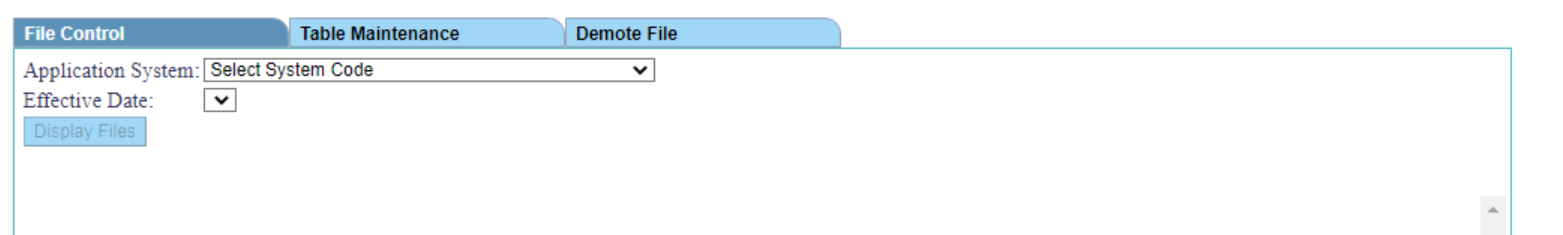
# **Technical Overview**

## **Migration Report**

* IBM Binary App Scanner is used to generate the Migration Report by uploading the project EAR file.
* It was decided to migrate the whole application since we had the similar JSF Migration issue in March 2021 for another application.



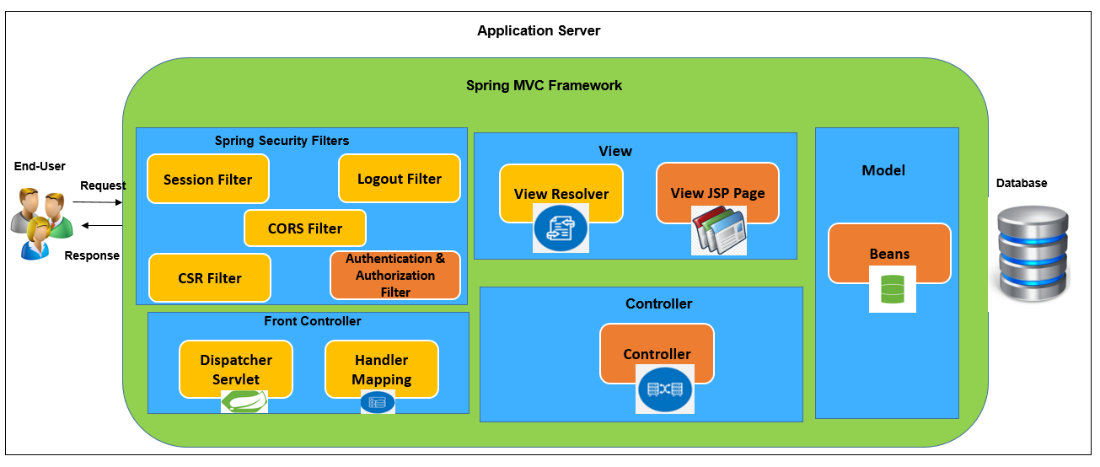
* Followed the same approach of Migrating JSF to Spring MVC to design this application.
  + Treasury Receipt Station is a small-scale application used only by business.
  + Has 3 screens with multiple functionalities.
    - File Control
    - Table Maintenance
    - Demote File



* I’ve personally worked on *developing 2 screens (File Control, Demote File) + one functionality in Table maintenance screen (Retrieval of Dynamic Table Data).*

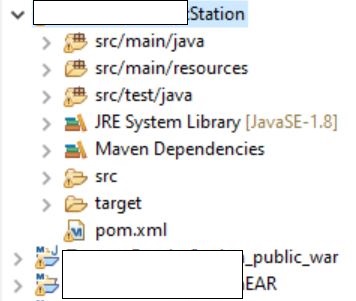
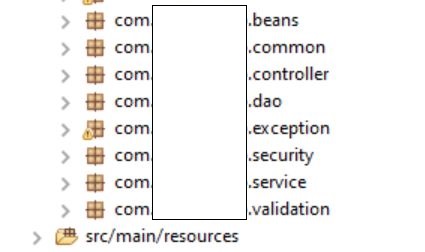
## **Architecture Context Diagram**

* This application is built using the standard Spring Web MVC framework.
* Spring Security is introduced in this application for centralized security layer



## **Package Structure**

* This application follows the standard Maven project structure.
  + src/main/java
  + src/main/resources
  + src/test/java
  + src/test/resources
  + pom.xml

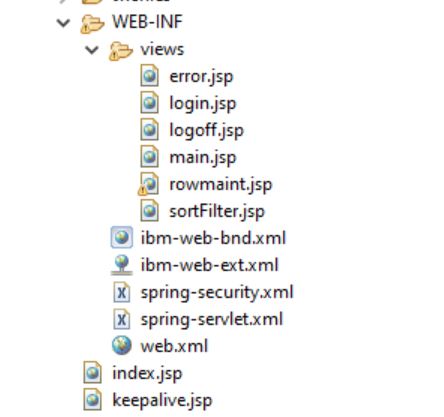
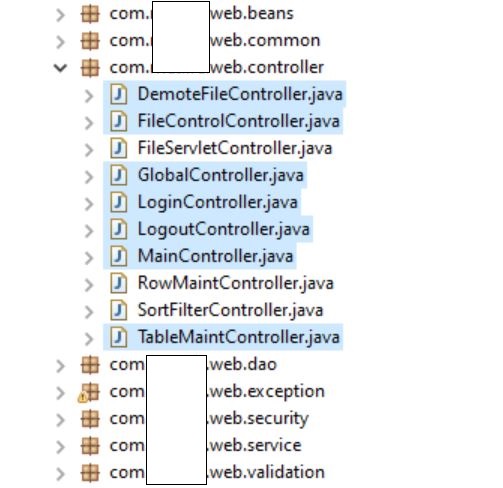
**Package Explanation**

* ***beans*** – Model Bean
* ***common*** – acts as a util package where the common functionalities are implemented for code reusability
* ***controller*** – url mapping
* ***service*** – business logic (Interface and Implementation)
* ***DAO*** – Database Layer (Interface and Implementation)
* ***Security*** – Centralized Security layer
* ***Exception*** – Custom Exception

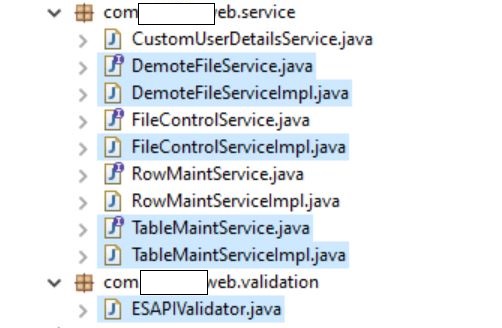
I’ve highlighted the class files that I’ve implemented or developed a part of it.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Validation/Exception** | **Spring Security** | **Controller** | **View** | **Service** | **Helper/Constants** | **Database Model** |
| ErrorCodeHandler | AccessControl | TableMaintController | Main.jsp | CustomUserDetailsService | Constants | DemoteFileDAO |
| TRSExceptionHandler | AppRole | RowMaintController | RowMain.jsp | DemoteFileService | LogoutActivity | DemoteFileDAOImpl |
| ESAPIValidator | Role | FileContController | SortFilter.jsp | DemoteFileServiceImpl | NumberUtils | FileControlDAO |
|  | TrsUserDetails | DemoteController | Login/logoff.jsp | FileControlService | PaginationUtil | FileControlDAOImpl |
|  |  | SortFilterController | Error.jsp | FileControlServiceImpl | SaveAs | RowMaintDAO |
|  |  | MainController |  | RowMaintService | SQLConstants | RowMaintDAOImpl |
|  |  | LogoutController | Main.js | RowMaintServiceImpl | StringUtils | TableMaintDAO |
|  |  | LoginController | RowMain.js | TableMaintService |  | TableMaintDAOImpl |
|  |  | GlobalController | sortFilter.js | TableMaintServiceImpl |  | UserDetailsDAO |
|  |  | FileServletController |  |  |  | UserDetailsDAOImpl |
|  |  |  | RowMaint.css |  |  |  |
|  |  |  | Master.css |  |  |  |
|  |  |  | Main.css |  |  |  |

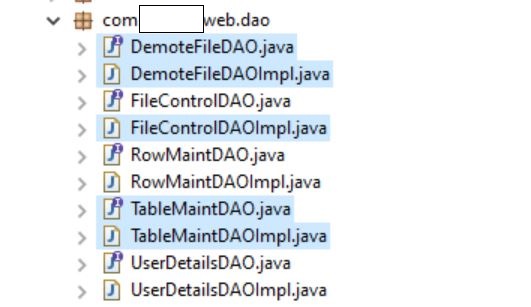
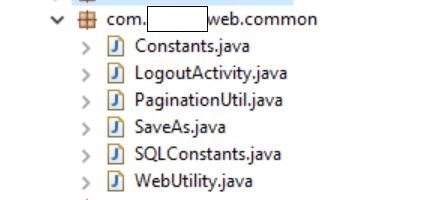
**JSP and XML in webapp folder Controller classes for each functionality**

**Service Layer (Interface and Implementation)**



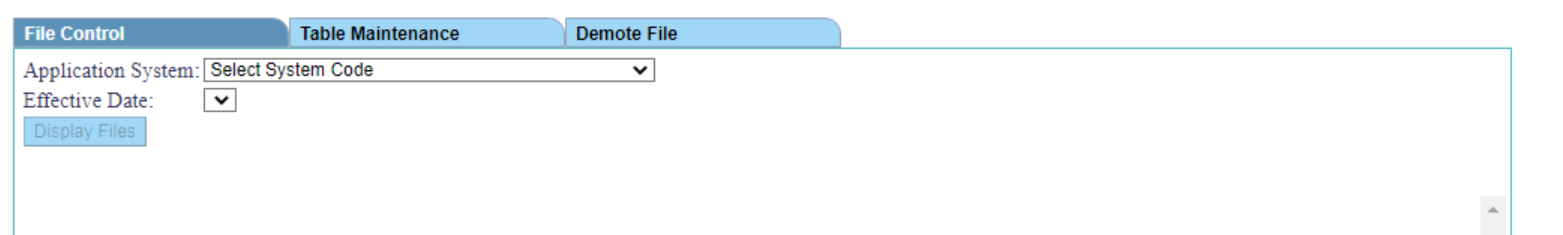
**Database Layer (Interface and Implementation) Util package for common methods and Constants**

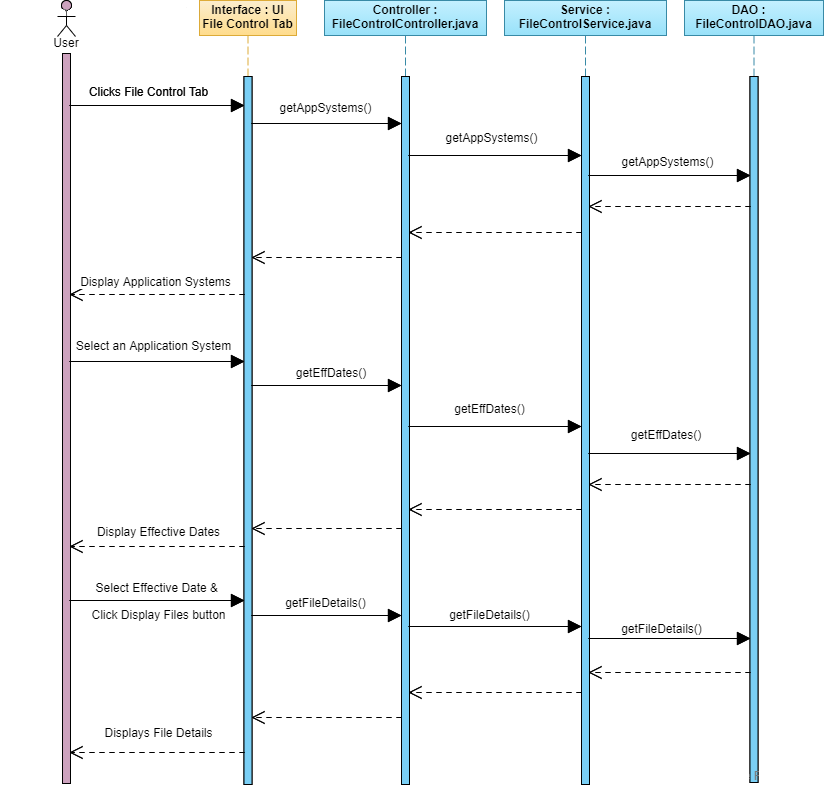
## **File Control**

* A simple retrieval functionality by accessing 3 different tables from the database. (**SELECT queries + JOIN**)
* The control goes from *login.jsp 🡪 LoginController 🡪 MainController 🡪 FileControlController 🡪 FileControlService 🡪 FileControlDAO*

### **User Interface**

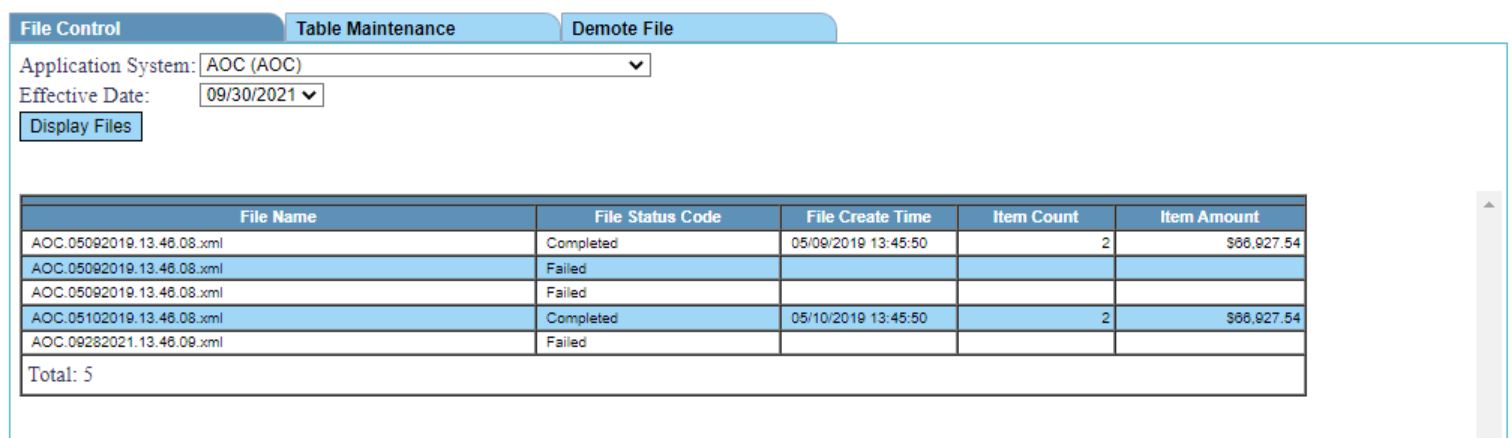


### **Code Flow Implementation**



### **Functionality**

1. User lands on the File Control tab after the successful login. It has the following fields
   1. *Application System – Dropdown*
   2. *Effective Date – Dropdown (Disabled)*
   3. *Display Files – Button (Disabled)*
2. User selects Application System from the dropdown,
   1. *Effective Date dropdown is enabled and populated with the first value in drop down*
   2. *Display Files button will be enabled*
3. User clicks on ‘**Display Files’** button
4. File details will be displayed in a table format with the below columns.
   1. *File Name*
   2. *File Status Code*
   3. *File Create Time*
   4. *Item Count*
   5. *Item Amount*



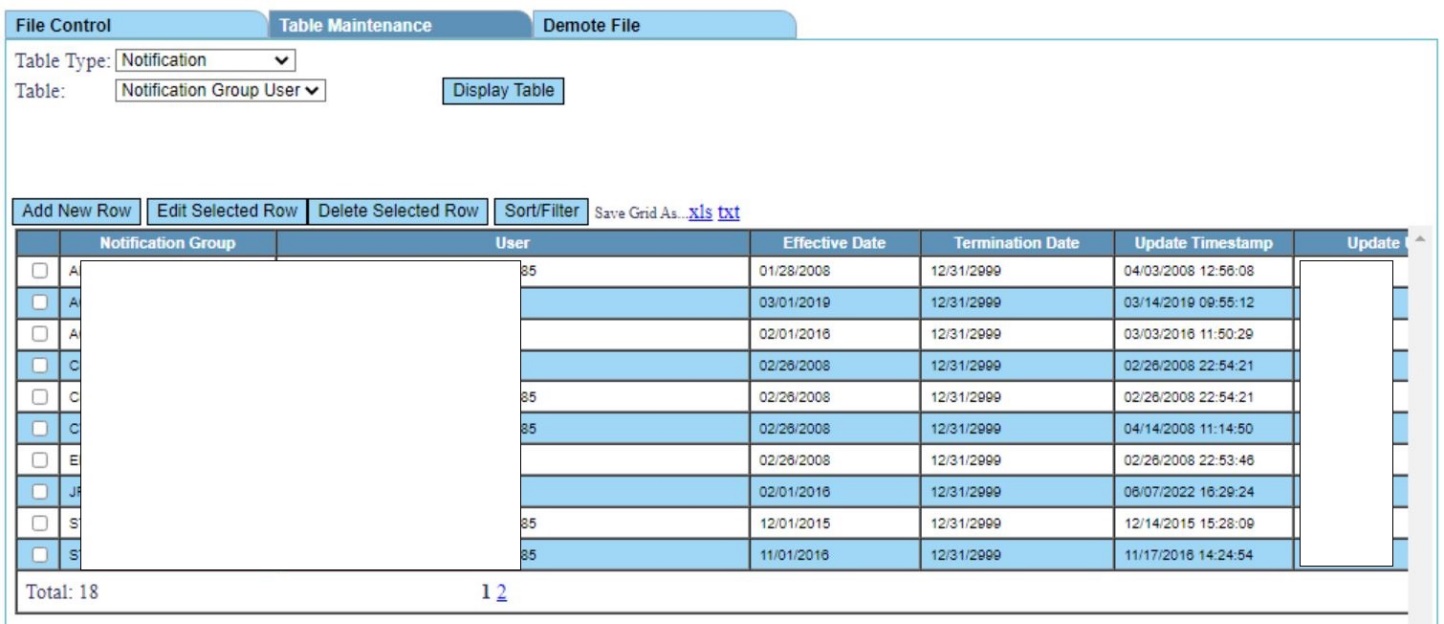
## **Table Maintenance Tab**

* The control goes from login.jsp 🡪 LoginController 🡪 MainController 🡪 TableMaintController 🡪 TableMaintService 🡪 TableMaintDAO
* Retrieval functionality is used to display the table details. (SELECT queries + JOIN)

### **User Interface**



* DML queries are used to add/update/delete rows from the selected tables
* Result is also uploaded as a file to user system (.txt or .xls format)

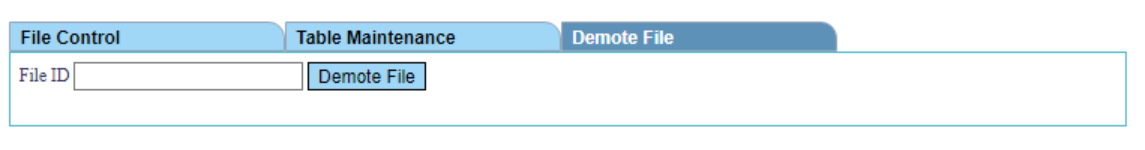


*I would like to inform you that I’ve not worked on Insert/Update/Delete records functionality in this particular screen.*

## **Demote File Tab**

* To demote or delete the record from the database. All the related child table details are also deleted **(SELECT query + Subqueries + DELETE queries)**
* The control goes from login.jsp 🡪 LoginController 🡪 MainController 🡪 DemoteFileController 🡪 DemoteFileService 🡪 DemoteFileDAO

### **User Interface**



### **Code Flow Implementation**

