Technical Specification Document

Document Control Guidelines

Version: 1.0

Date: 11-Feb-2022

Version control			
Version Number	**Date issued**	**Author**	**Update information**
V1.0	11-02-2022	Umer Farooque	First published version

Disclaimer

The information contained in this document is confidential, privileged and only for the information of the intended recipient and may not be used, published or redistributed without the prior written consent of **{Company_Name}**

Table of Contents

1. Introduction 5
2. Requirements 5
3. Solution 6
3.1 Database (Oracle) 6
3.2 Web API / Middleware / Business Logic 6
3.3 Web Pages / Frontend / Presentation Layer 7
3.4 Background Worker / Windows Service / .NET Worker 11

1. Introduction

An application to connect the oracle database (express edition), read files from directory and save into oracle data table. Project is divided into three parts.

1 – Web API / Backend

2 – Web MVC / Frontend

3 – Worker service / Background worker

2. Requirements

1. **Develop .net web API with following functionalities:-**
   - Look to the specific path for txt file
   - Upload .txt file to the oracle table along with file name, created date and time as per the attached file
   - Verify file type which to accept.txt only. Error message to be displayed if file format is not .txt
   - Process file content in the DB by inserting each line in another oracle table
   - Log status for each function performed in a log file

2. **Develop .net service which will do the following:-**
   a. Call the above .net web API every x minutes (for example 1 minute) and perform its functionalities.

3. **Develop ASP.net web page with following functionalities:**
   a. Display the processing of the .net web API log file
   b. Refresh page every x minutes (for example 1 minute)
   c. Web page should use bootstrap

3. Solution

Solution developed in .NET 6 with Oracle database for storing data.

3.1 Database (Oracle)

Application is connect with the Oracle Database Express Edition (XE) using Oracle.ManagedDataAccess.Core library.

All the script are attached in solution repository (AdditionalFiles/Db Script). Scripts are divided into three parts.

1 - new-db-user-script.sql

After connecting Oracle Database with SYS user, this script will help with creating a new user which will be later used by application to connect the Oracle Database.

2 - tables-sps-script.sql

Script contains are table creation script along with all stored procedures.

3 - db-script.sql

Sample select and insert statements for querying database.

3.2 Web API / Middleware / Business Logic

A web API developed in .NET 6 Web API to perform project business functionalities. It supports the dependency injection (DI) software design pattern, which is a technique for achieving Inversion of Control (IoC) between classes and their dependencies. A detailed API specification is document is also wrote separately. API have following functions:

1 – Check Files in defined directory and process them and load them into Oracle Database

2 – Get the content of log file

3 – Download the log file

4 – Get the list of files from Oracle Database

5 – Get the file content from Oracle Database for particular file

Configurations:

All major configurations are stored in appsettings.json file.

1 – Connection String -> Oracle connection string is stored under “ConnectionStrings:OracleConnection”

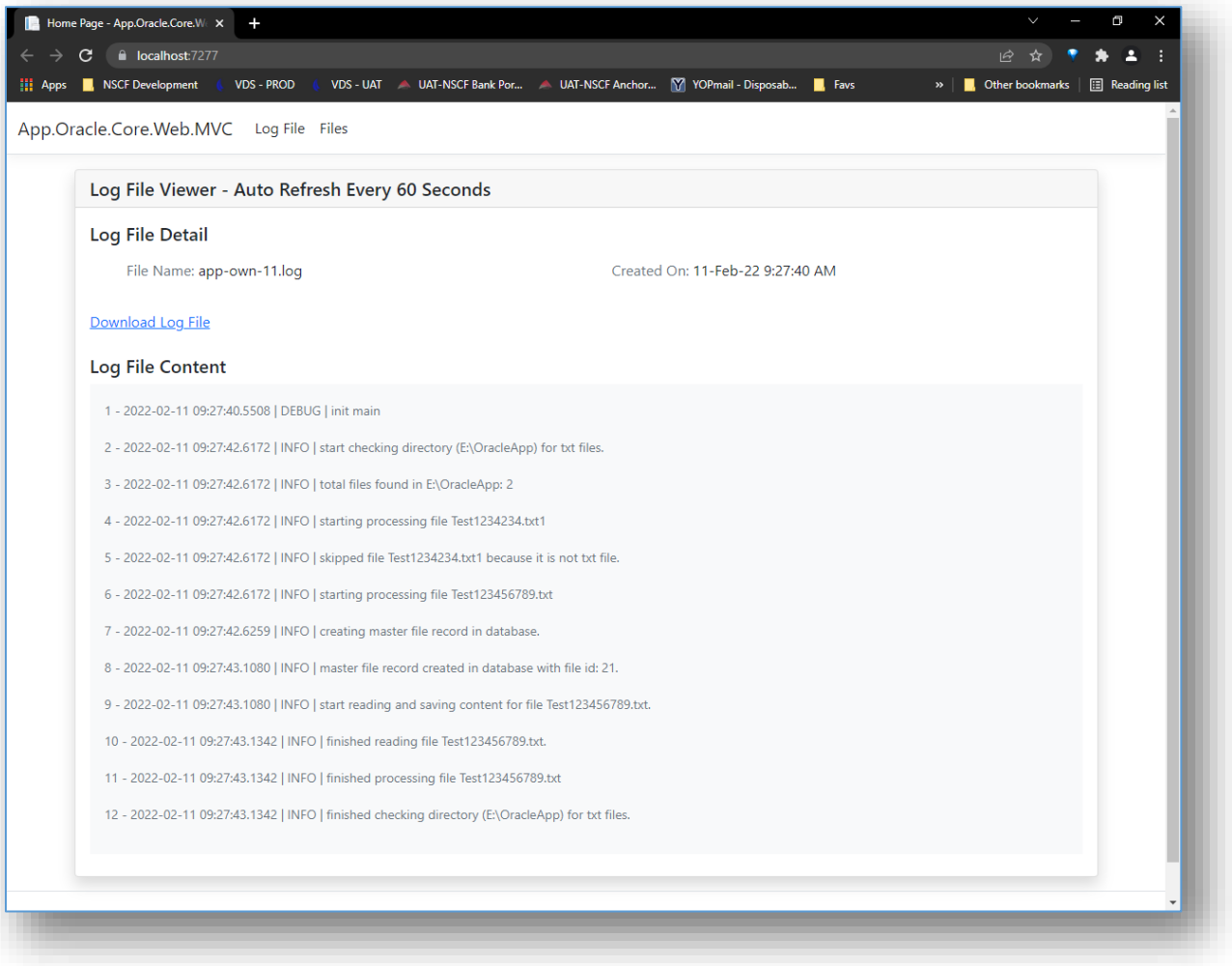
2 – API basic Auth Credentials -> credentials are stored under “Auth” section. Credentials are encrypted via AES cryptography which is part of Shared library. (App.Oracle.Core.Shared/Util/Cryptography). These credentials are used by Web Pages and Worker service to access the Web API.

3 – Directory path to look for .txt files -> Folder path to process .txt files are defined under “AppSettings” section. There are two path defined in configuration one for read the files and other to back up the files once processed.

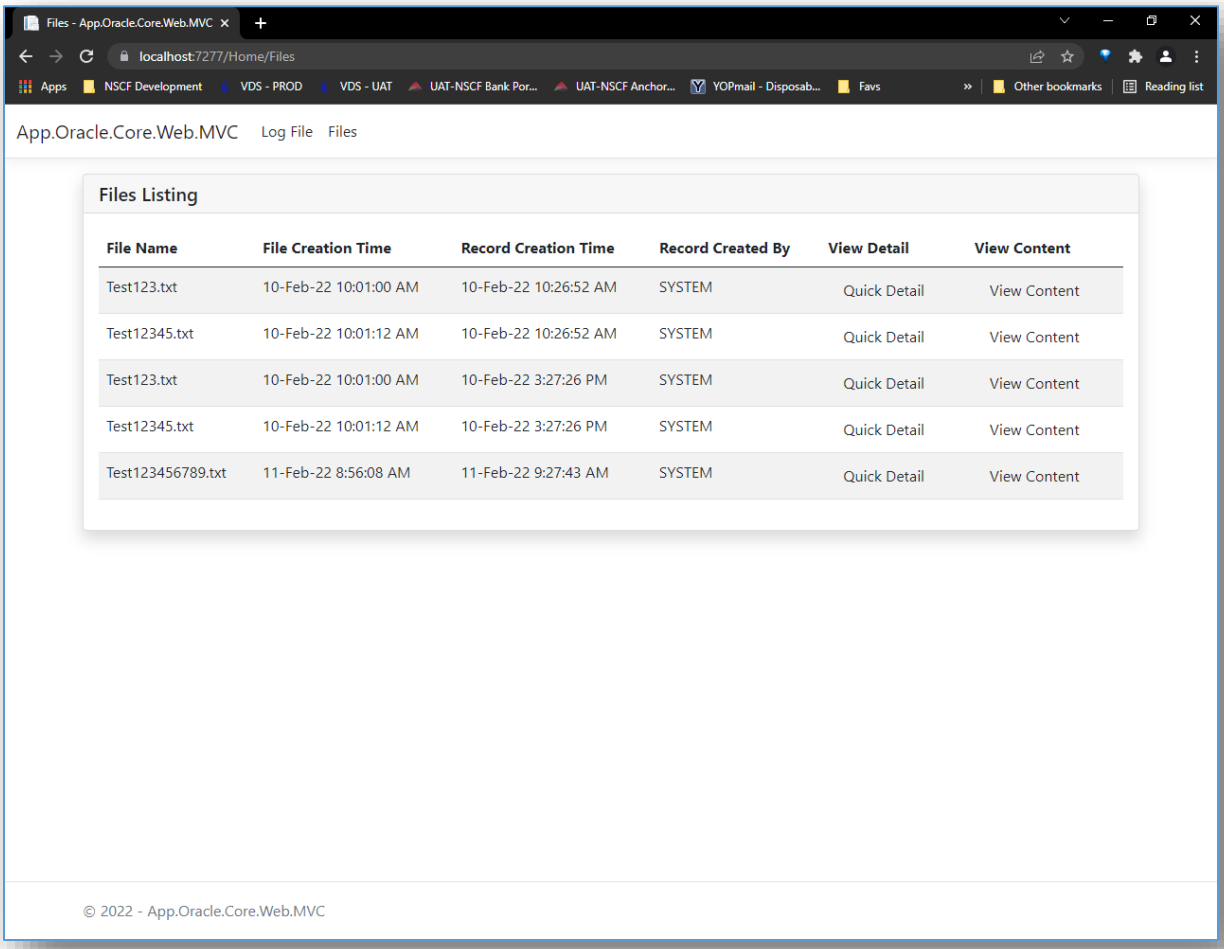
3.3 Web Pages / Frontend / Presentation Layer

A frontend developed in ASP.NET CORE MVC using bootstrap 5 most popular front-end open source toolkit to design responsive web pages. It supports the dependency injection (DI) software design pattern. Application contains three web pages.

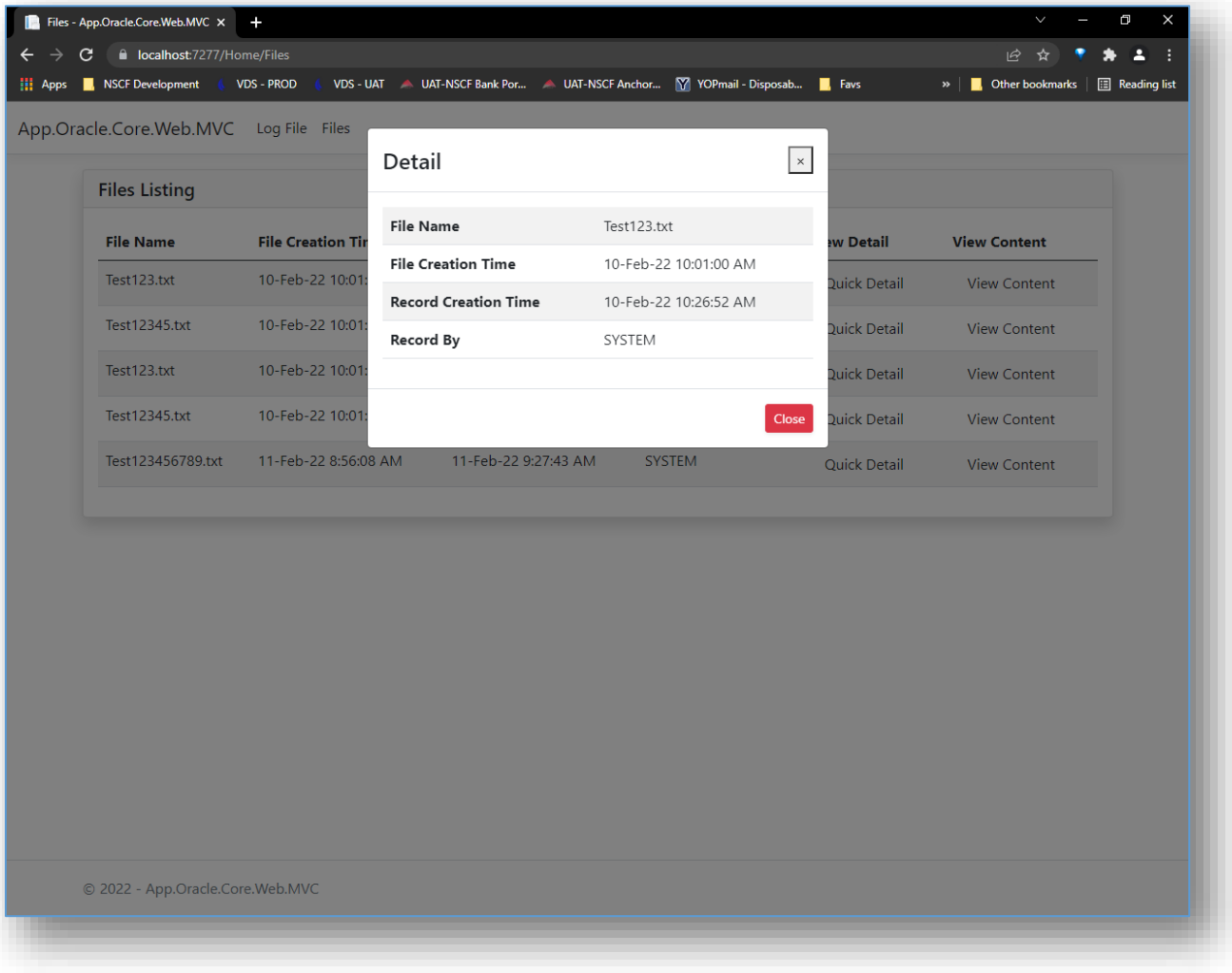
1 – Display the log file details along with file content (Auto refresh every 1 minute)



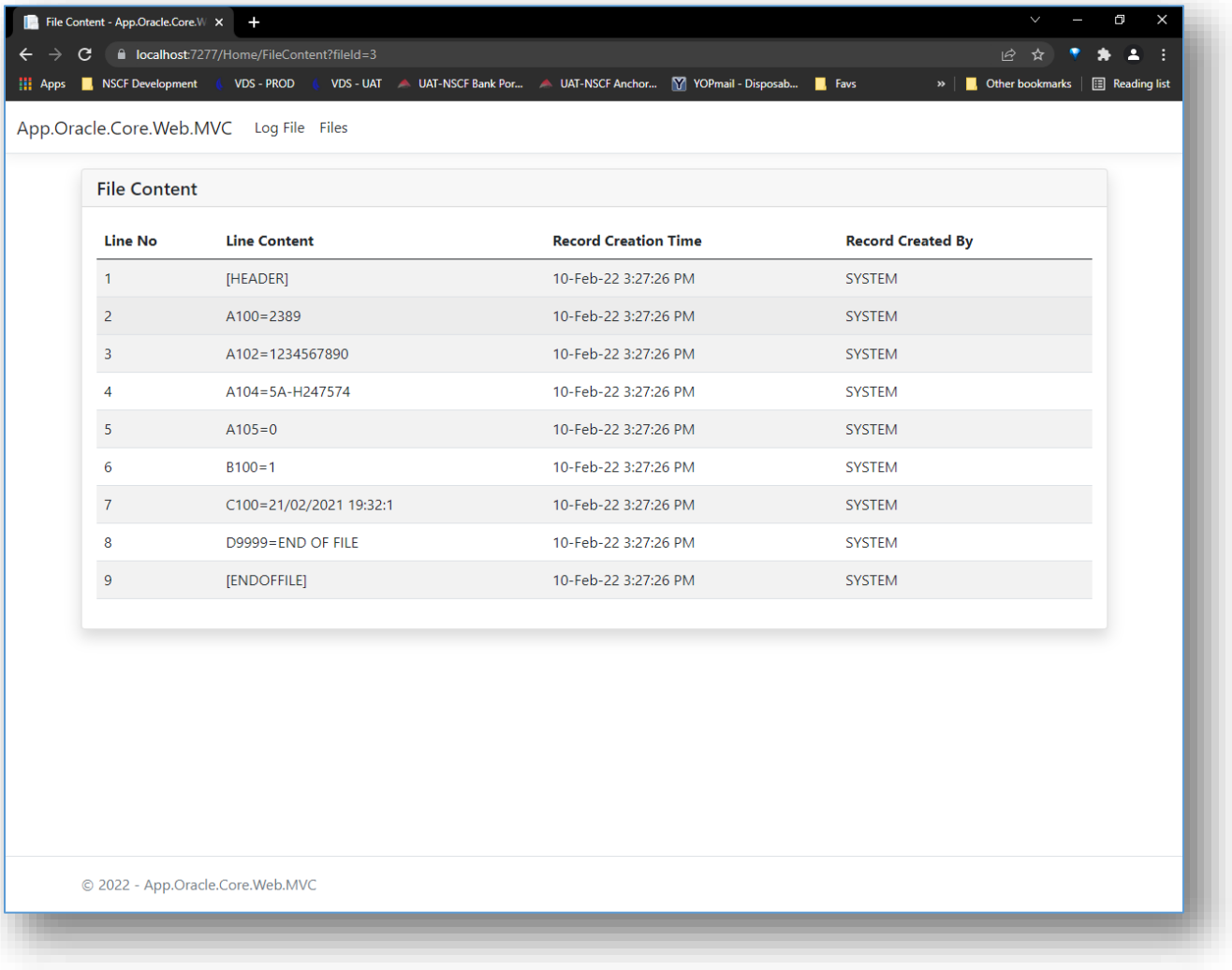
2 – Listing the files master details from Oracle database



3 - Bootstrap modal to display the quick details of record



4 – Display the content of requested file from Oracle database



Configurations:

All major configurations are stored in appsettings.json file.

1 – Web API Connectivity -> Base URL and Credential to access the Web API are stored under “WebService” section.

2 – Auto Refresh Time -> Settings to change the time for auto refresh log file display page can be find under “AppSettings” section named as “AutoRefreshPage”. Time is defined in seconds and currently it is set as 60 seconds.

3.4 Background Worker / Windows Service / .NET Worker

A .NET Worker service to perform certain action automatically after every 1 minute. Service will call the WEB API endpoint to check for files in pre-defined directory. It supports the dependency injection (DI) software design pattern.

Configurations:

All major configurations are stored in appsettings.json file.

1 – Web API Connectivity -> Base URL and Credential to access the Web API are stored under “WebService” section.

2 – Interval Time-> Settings to change the time for internal of service can be find under “Worker” section named as “DelayTimeInSeconds”. Time is defined in seconds and currently it is set as 60 seconds.