|  |
| --- |
|  |
| Open Reports System Technical Specification and Architecture |
|  |

Table of Contents

[Table of Contents 3](#_Toc524517073)

[Architectural Design Diagram 4](#_Toc524517074)

[Overview 4](#_Toc524517075)

[LIS 5](#_Toc524517076)

[EBEIS 6](#_Toc524517077)

[Server Specifications 7](#_Toc524517078)

# Architectural Design Diagram

## Overview

Open Reports is a generic reporting system that enables user to select existing datasets, select the desired elements from the selected dataset/s, apply filters then generate report. The system uses a queue to regulate the flow of report request in order for the server to not drown itself from requests.

## Diagram

****

## Components

* ORS UI
  + Contains all the UI related scripts (i.e js, html, jsp)
  + Once a report is requested to be processed, it queues the report via EJB to wildfly server
* QueueEJB
  + Is an interface enabling outside application (i.e ORS UI) to queue report.
  + This encapsulates queuing in the ActiveMQ
* Queue Listener
  + Is the one manages report generation
  + It listens to the ActiveMQ. Once an entity is queued to the ActiveMQ, it immediately executes its listener orchestrating the following
    - Consumes the entity
    - Pass the entity to the SqlBuilder and returns an object
    - Pass the object to ReportGenerator to generate the sql string and generate the physical file. The generator then returns the location of the physical file
    - The physical file URL is then registered to the file manager
    - Emails the user for the download link / error
* Apache ActiveMQ
  + Enabled in wildfly server. A data structure.
  + See documentation for more details [here](http://activemq.apache.org/)
* File Manager
  + Responsible for keeping track of the generated files
  + Deletes the file if the Time to Live of the file expires
  + Has also an EJB acting as an interface for registering a file
  + Uses Redis for trigger to when it should delete a file
* Redis
  + An in memory database storing and keeping track of generated files and when it will be deleted via its time to live feature
  + Informs file manager when a file is to be deleted
* Mail Manager
  + Responsible for sending mails

Servers

* Tomcat
  + App server for the UI
* Wildfly
  + App server for backend processing of reports (asynchronous)

# Server Specifications

