## Problem Statement: Large scale configurable notification system (Problem #1)

## Classes involved in the solution:

## MainLoadCSV : Class is responsible for creating multiple threads for loading the CSV data into DB.

## LoadCSV: Class is responsible for loading the data from CSV to DB and updating the stale records from updated one.

## GetQuery: Class is basically dealing with all the fetch operations form the DB.

## NotofySubscriber: Class is providing the functionality of notifying the customers based on the conditions provided in the configuration file.

## Main: Entire flow of the process starts with this class. Along with, this class is getting the user inputs form console to provide the entire attributes details for a particular item provided by user.

## Inputs to the system:

## .conf file: this is the global file cantains all the configurations details for every subscriber.

## .csv files: csv file with data at fixed location.

## Dummy contol file: To build the control file for SQl loader script at run time.

## 3 jar file were added to provide the database operations and email delivery.

## Ojdbc6.jar, activation.jar and mail.jar. All three jars are standard and are being used globally.

## Process flow:

## System starts with the main class(Main) that creates the single thread of MainLoadCSV class.

## Thread of the MainLoadCSv class works as parent and creates multiple threads of LoadCSv class.

## Each thread of LoadCSV class process single file(.csv) from the shared location and loading it into DB. Post successful loading this class is updating the stale records if any update for any of the item attribute is available in any of the CSV file.

## System is capable to update the latest records in DB even after multiple csv file are being processed at the same time. System is using timestamp of the csv file to identify the latest change.

## Also, this class is calling NotofySubscriber class.

## NotofySubscriber class is reading the configuration file and deriving the corresponding sql query. Query checks the for the available updated for the configuration.

## Also, notifying the subscriber with desired notification through asynchronous messaging system(email).

## Simultaneously, system is reading the console to respond back the GET <item> command.

## DB Schema:

## CREATE TABLE ALL\_CSV\_RECORDS(ITEM NUMBER, ATTRIBUTE\_NAME VARCHAR(2000), ATTRIBUTE\_VALE VARCHAR(3000), INSERT\_AT NUMBER);

Assumption:

1. Shared path needs to be provided at each place in the code.
2. Email ID/Password needs to be provided for SMTP authentication.