**PROJECT REPORT ON**

**EMPLOYEE MANAGEMENT SYSTEM USING SPRING FRAMEWORK AND RESTFUL API**

Submitted in partial fulfillment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY IN**

**COMPUTER SCIENCE AND ENGINEERING**

**OF SASTRA UNIVERSITY**

**Submitted by**

**Kuna Praneetha(117003102)**



**Under the Guidance of**

**B.Rama Subramanian**

**i|Nautix Technologies-A BNY Mellon Company CHENNAI**

**SCHOOL OF COMPUTING**

**SHANMUGHA**

**ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY**

**(SASTRA UNIVERSITY)**

**(A University Established under section 3 of the UGC Act, 1956)**

**TIRUMALAISAMUDRAM**

**THANJAVUR – 613 401**

**April 2017**

**SCHOOL OF COMPUTING**

**SHANMUGHA**

**ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY**

**(SASTRA UNIVERSITY)**

**(A University Established under section 3 of the UGC Act, 1956)**

**TIRUMALAISAMUDRAM, THANJAVUR – 613401**



**BONAFIDE CERTIFICATE**

Certified that this project work entitled “Quality Checking automation using rules engine” submitted to the Shanmugha Arts, Science, Technology & Research Academy (SASTRA University), Tirumalaisamudram- 613401 by

**Kuna.Praneetha with 117003102**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER** **SCIENCE AND ENGINEERING** is the original and independent work carried out under my guidance, during the period January2017 - April 2017**.**

|  |  |
| --- | --- |
| **INTERNAL GUIDE** | **ASSOCIATE DEAN** |
| **B.RAMA SUBRAMANIAN, LNT department** | **Dr. A. UMAMAKESWARI** |
| **i|Nautix Technologies-A BNY Mellon Compnay,Chennai** | **SCHOOL OF COMPUTING** |

Submitted for University Examination held on\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EXAMINER - I** **EXAMINER - II**

**SCHOOL OF COMPUTING**

**SHANMUGHA**

**ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY**

**(SASTRA UNIVERSITY)**

**(A University Established under section 3 of the UGC Act, 1956)**

**TIRUMALAISAMUDRAM, THANJAVUR – 613401**



**DECLARATION**

We submit this project work entitled **“EMPLOYEE MANAGEMENT SYSTEM USING SPRING FRAMEWORK AND RESTFUL API”** to the Shanmugha Arts, Science, Technology & Research Academy (SASTRA) University, Tirumalaisamudram–613 401, in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** and declare that it is our original and independent work carried out under the guidance of Mr. **B.Rama Subramanian,i|Nautix Technologies-A BNY Mellon Company.**

**Date: 03/4/2017 Name: Kuna Praneetha Signature:**

**Place: Chennai Reg. No: 117003102**

**ACKNOWLEDGEMENT**

It is a great pleasure for us to present this project to all of you. We would like to acknowledge each and every one who had a role to play in making our humble efforts an out-to-out success.

We would like to thank our Honorable **Vice Chancellor (Col.) R. Sethuraman** for providing us with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

We find very heartening the encouragement and strategic support offered at every step of our college life by **Dr. S. Vaidhyasubramaniam**, **Dean, Planning & Development** and **Dr. V. Badrinath, Dean, Training & Placement and School of Management** and for that we would like to thank them time and again.

We wish to express our gratitude to **Dr. A Umamakeswari**, **Associate Dean**, **and School of computing** for her foresight and guidance, to complete our project in time.

We extend our heartfelt thanks to the **Registrar**, **Dr. G. Bhalachandran**, for providing the opportunity to pursuit this project.

We are very thankful to our internal guide **B.Rama Subramanian**, **LNT Department**,i|Nautix Technologies, for encouragement, guidance and support throughout the project.

We express our sincere thanks to all the staff of School of computing, for the help throughout the project.

We express our wholehearted gratitude to our parents and friends who kept my spirits high without whose cooperation the completion of the project would not be possible.

**TABLE OF CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **S. no.** | **CHAPTER** | **Page Number** |
| **1** | **Introduction** | **1** |
| **2** | **Time Frame Chart** | **2** |
| **3** | **Statement of the Problem** | **3** |
| **4** | **Objectives of the study** | **3** |
| **5** | **Literature Review** | **4** |
| **6** | **Software, Hardware Requirement Specification** | **5** |
| **7** | **Conceptual Modeling** | **8** |
| **8** | **Interaction Scenario** | **15** |
| **9** | **Methodology and Approach** | **21** |
| **10** | **Sample Code** | **27** |
| **11** | **Output** | **36** |
| **12** | **Conclusion** | **45** |
| **13** | **References** | **46** |

**1. INTRODUCTION**

**1.1 About the Company**

i|Nautix Technologies India Private Limited is a group company of Bank of New York Mellon - a leading financial services provider. The Bank of New York is the oldest banking corporation in the United States, and the 20th-oldest bank in the world, having been established on June 9, 1784, by American Founding Father Alexander Hamilton.

iNautix provide technology development, business & technology operations and remote infrastructure management services for BNY Mellon and its subsidiaries. iNautix also develops and delivers comprehensive technology solutions and software development products for customers of BNY Mellon.iNautix is a vital arm of Client Technology Solutions at BNY Mellon.

i|Nautix provides the following:

* Deliver service excellence and innovative solutions
* Foster a collaborative work environment that values inclusion and offers associates, opportunity for growth and recognition
* Deliver insights in shaping the future of our industry
* Support local and global communities

i|Nautix provides Cutting-Edge Technology Services. Application performance, availability and response times are critical factors for success in the constantly evolving and highly competitive capital markets. i|Nautix have a expertise in building a wide array of financial systems which includes

1. broker customer management platform,
2. asset movement systems,
3. back office processing systems,
4. advisor platforms,
5. retail client platforms and
6. institutional platforms

iNautix combines technical expertise with agility, experience, value, and flexibility to deliver technological solutions.

To ensure a prompt service response in today's rapidly changing market situations, the entire iNautix team works as a unified whole from start to finish by pooling their extensive domain experience & technology expertise to provide a competent range of service offerings including

- Full Life-cycle Application Development

- Infrastructure Hosting Services

- Human Factors Engineering

- Product Validation Services

- Financial Systems Reengineering

- Technology Consulting

- Automating Processes

- Datawarehousing and Business Analytics

**Company Technologies**

***Nexen Gateway:***

BNY Mellon has always been at the forefront of innovation, leading the way with new thinking as well as products and services designed to help our clients succeed.

From cloud computing to mobile platforms to big data technologies and more, BNY Mellon are creating a new way of developing and delivering solutions to clients and focusing on

on increasing access and collaboration in the markets, improving flexibility and efficiency, strengthening risk management and facilitating growth through access to value-added information-based services.

BNY Mellon is launching the biggest technological transformation which is NEXENSM, a new open-source, cloud-based technology platform. NEXEN is designed to empower clients to change the way they operate to be more flexible, client-responsive, efficient and fast. Its cutting-edge capabilities will help clients at every stage of the investment lifecycle.

The main objective of iNautix is to create new and original solutions that will position our clients for success which include:

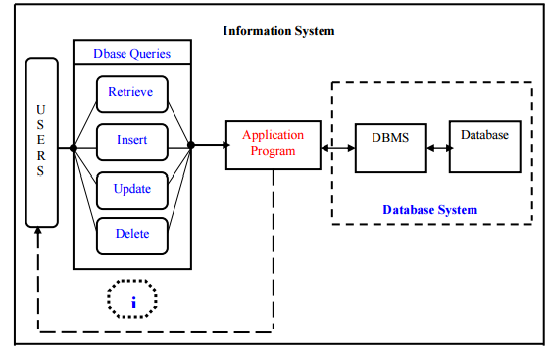
* **Cloud-based platform** – NEXEN, our new open-source cloud-based technology platform, is on the cutting-edge of financial services.
* **Big Data Insights** – Digital Pulse and Digital Workplace provide data-driven insights to make evidence-based decisions faster.
* **Acquisition Strategy** – Leveraging advanced investment technology from BNY Mellon acquisitions such as AlbridgeTM, Eagle and Hedgemark.
* **Innovation Centers** – Our new Innovation Centers in Silicon Valley, Pune, Chennai, and Jersey City will maintain close contact with leading-edge thinkers, new talent and open-source culture.

**1.2 BACKGROUND OF THE PROJECT**

Employee management is the management of subordinates in an organization. Often, large organizations have many of these functions performed by a specialist department, such as Personnel or Human Resources, but all line managers are still required to supervise and administer the activities, and ensure the well-being, of the employee that report to them.

Most of the contemporary Information systems are based on the Database technology, as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain and control access to the database. The process of constructing such kind of systems is not so simple. It involves mutual development of application program and database. The application program is, actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and database are very important for the reliability, flexibility and functionality of the system. The so defined systems differentiate to each other and their development comprises a great variety of tasks to be resolved and implemented.

The basic idea can be depicted on Figure below:



Employee management system suggests a computer technology to be used in order to provide information to users in an organization (for instance), as for the purposes of data transformation into useful information; computer hardware and software are designed and used .

A particular case is the Human Resources Information System development. This kind of systems are responsible for storing data of the employee within an organization and generating reports upon request. Such kind of system could be integrated with other Information systems or modules:

* Accounting Information System(AIS)-designed to transform financial data information or

* Management Information System (MIS) that provides decision-oriented information to managers, and so on…

“Organizations depend on Information Systems in order to stay competitive. Productivity, which is crucial to staying competitive, can be increased through better

Information Systems.”

The End users can insert, delete, update and retrieve the data from the database. The Application program acts as a interface between the database and the user.

**2. PROBLEM STATEMENT**

Management of employee information is very crucial to any organization and it is very important to every organization to have their employee detail record stored. This makes the company very easy to browse about the current employee regarding the ID, place, age and address etc.

Previously the data of the employees was maintained in a spreadsheet format but maintaining the data in a spreadsheet format will make the date vulnerable to human errors. And such applications would be not fast and efficient in retrieving the data. Hence so after the databases were developed the work became much simpler for retrieving, inserting any information about the employee of the organization

This system makes the process of scheduling much easier and computerized. By this system the manager or top level designated employee can view the details of any employee working under him, he even can check whether the employee is free in the particular time or in other times. Thus the top level management can easily fix the process of scheduling, and even can change the appointment which is reflected immediately to the related employee avoiding direct contact of the employee resulting in saving lot of time and work overhead.

**3. OBJECTIVE**

The objectives of the system are:

* This system will reduce the complexity of employee management.
* By using this system we can easily maintain all the records about” ON

EMPLOYEES” or “OFF EMPLOYEES”.

* It will reduce searching time.
* It can be easily handled by the person who have elementary knowledge of computer because it provides an user friendly environment.

**3.1 The Domain**

Employee Management System is one in which it is easy and simpler to maintain the information regarding an employee like the name, year of joining, projects etc. The traditional way of storing the details of an employee

The problem is very fiddly for traditional programming

* The problem cannot be solved by any of the existing algorithms
* The business logic changes very often that capturing the change is a challenge
* Domain experts are available but are not well versed in programming

The quality checking problem matches all the constraints mentioned above. The problem is fiddly in a way it is not very complex but it needs a detailed implementation to be done. It cannot be solved by any algorithmic solution and the business logic changes rapidly thereby, leading to a change in the corresponding quality check to be performed. There are people who are experts in doing quality check without programming knowledge.

**3.2 Existing System**

Existing system requires manual integration of data and management of different-different processes. Company requires different systems for Human Resource Information management, Project management and client & Product management. Searching and tracking of information becomes complex and time consuming. The Existing System was a tightly coupled java application. In case of any enhancements the entire code needs to be modified which is a time consuming process.

**3.3 Proposed System**

The tool has been replaced with an interface much better to use and also matching with the current trends. User experience part has been increased with additional options to import his/her file. The time to code the quality checklist has been greatly reduced as the user has to just write the rules in simple English terms and need not have knowledge in programming or the details of the system.

The user can easily modify the rules when the business logic changes without changing the complete system.

**5. LITERATURE REVIEW**

This platform for automating the quality checking process uses a rules engine called Drools for writing the rules, Java for the server side code and the back end, an existing tool developed by the company called MuFlow for the user interface and integrating drools with the types of input and output databases.

**5.1 DROOLS**

Drools is a business rule management system (BRMS) with a forward and backward chaining inference based rules engine, more correctly known as a production rule system, using an enhanced implementation of the Rete algorithm. Drools is an Object-Oriented Rule Engine for Java. Drools provides for declarative logic Programming and is flexible enough to match the semantics of problem domain. It can be integrated with debuggers and can be used as a plug-in in eclipse.

* 1. **Eclipse**

Eclipse is a Java-based open source platform that allows a software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members. Eclipse provides a common user interface (UI) model for working with tools.  It is designed to run on multiple operating systems while providing robust integration with each underlying OS.  Plug-ins can program to the Eclipse portable APIs and run unchanged on any of the supported operating systems

* 1. **Java Swings**

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element. Swing is currently in the process of being replaced by JavaFX.

* 1. **JExcel API**

Java Excel API is a mature, open source java API enabling developers to read, write, and modify Excel spreadsheets dynamically. Now java developers can read Excel spreadsheets, modify them with a convenient and simple API, and write the changes to any output stream (e.g. disk, HTTP, database, or any socket). Any operating system which can run a Java virtual machine (i.e., not just Windows) can both process and deliver Excel spreadsheets. Because it is Java, the API can be invoked from within a servlet, thus giving access to Excel spreadsheets over internet and intranet web applications.

* 1. **MuFlow**

MuFlow is an analytics tool built by Mu Sigma is a rapid analytics automation workbench that enhances ROI from analytical initiatives. In order to extract the value from data, analytics workflows need to be implemented in a quick and automated manner - agile analytics, a trend towards using technology to rapidly integrate and deploy solutions is a framework that exactly encapsulates this philosophy. Mu Sigma’s analytics automation workbench, enables one to build integrated human and machine workflows and deploy them seamlessly within the enterprise in an agile manner. It enables one to spend more time on creating and consuming tactical and strategic analytics rather than their design and deployment, thereby enhancing ROI on analytical initiatives.

**6. SOFTWARE REQUIREMENT SPECIFICATION**

**6.1 Software requirements**

The following are the minimum requirements that the target system (Automated Rules Engine) is expected to satisfy for smooth running of this application.

Operating System: Windows 2000 or above, any Linux system

Eclipse: 3.4 or greater

Eclipse GEF: 3.4 or greater

Drools plug-in: 5.0 or above

**6.2 Hardware Requirements**

The following are the minimum hardware requirements that the target system (android device) is expected to satisfy for running this program.

Processor: 1 GHz or Higher

RAM: 1GB or higher

Storage Space: 50MB free space

**6.3 Development Specifications**

Technologies used: Drools, Java (Eclipse), Muflow

IDE used: Eclipse

**6.4 Functional requirements**

The following functional requirements were specified

* The tool has to check if the user has browsed any input file. If not, it has to prompt the user to attach an input file
* When the user uploads an input file, the tool has to check the format of the file and notify the user if any unacceptable file has been attached
* If the file is of correct format, it has to display all the quality checks and the header row in the file so as to enable the user to select the type of quality check he/she needs to perform and on which column it has to be performed
* When a column and the type of quality check is selected, it has to perform the required processing on the column selected by the user in order to complete the selected quality check
* Support for exporting the results displayed after performing the quality check into required formats for higher understandability and further usage

**6.4 NON FUNCTIONAL REQUREMENTS**

**6.4.1 Performance requirements**

The input browsing, processing and the output exporting should be carried out without any delay. The delay if existed, should be less than when the process is done manually. The user should be able know the next step in the process with certain prompts. This makes the tool user friendly. Care should be taken as to observe that no column is filled with erroneous or ghost data. So the tool should make the user feel like the delay is less and the tool can be used for easy and faster usage. The tool should read data from the input file without any errors.

**6.4.2 Operational Requirements**

The tool to optimize quality check should be ready to be deployed in other applications with limited and easy customizations such as the supported formats for file and data and can be integrated as a component or plug-in. This makes the tool that is to be developed behave as a component in the real world of analytic tools.

**6.4.3 Browsing the input**

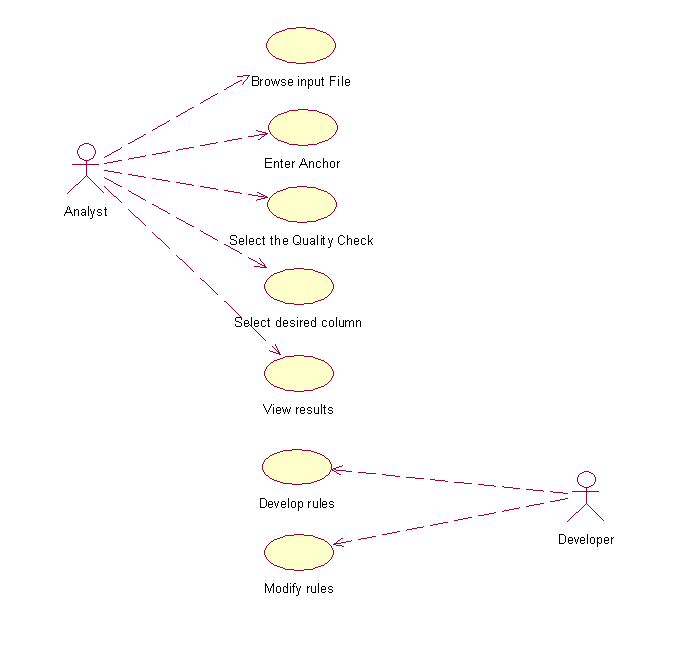
There must be a section where users will be able to browse the input file for processing. This file may be of any format. It can be simple excel sheets, R datasets, TeraData, SAS datasets which are pulled from a database, R server, SAS server etc. The tool should identify the file format correctly and make sure that the imported file format can be processed by it. The anchor, the cell where the data begins should be taken as another input from the user. The tool should be able to pull the header columns starting from the anchor entered by the user. No error should happen in displaying the header columns.

**6.4.4 Alert Messages**

The user should know the process and must be alerted with accurate error messages and the error codes must be decoded or translated in such a way that the user should know what exactly the error is and what is supposed to be done by the user for the process to be completed successfully.

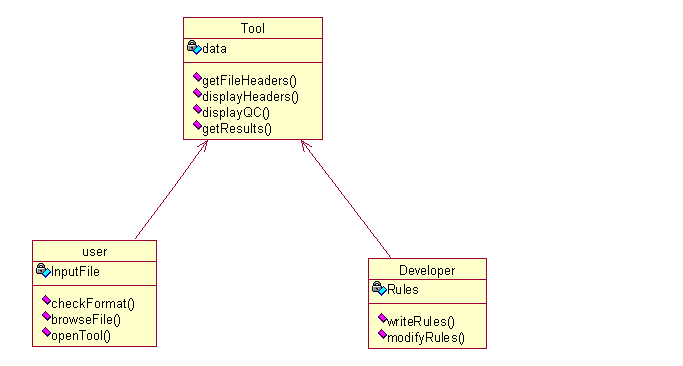
**7. CONCEPTUAL MODELLING**

**7.1 Use-Case Diagram:**

**

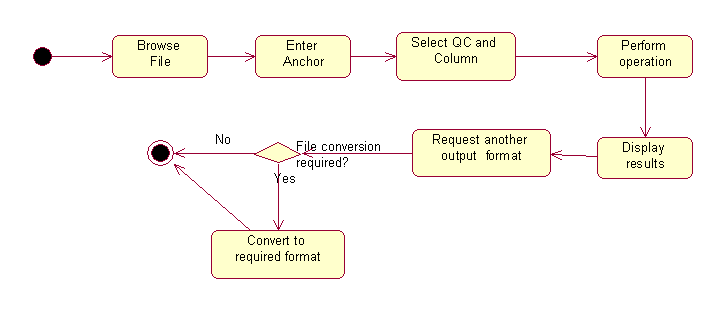
*Figure 7.1.1. Use case diagram for various users*

**7.2 Class Diagram**

****

*Figure 7.2.1 Class diagram*

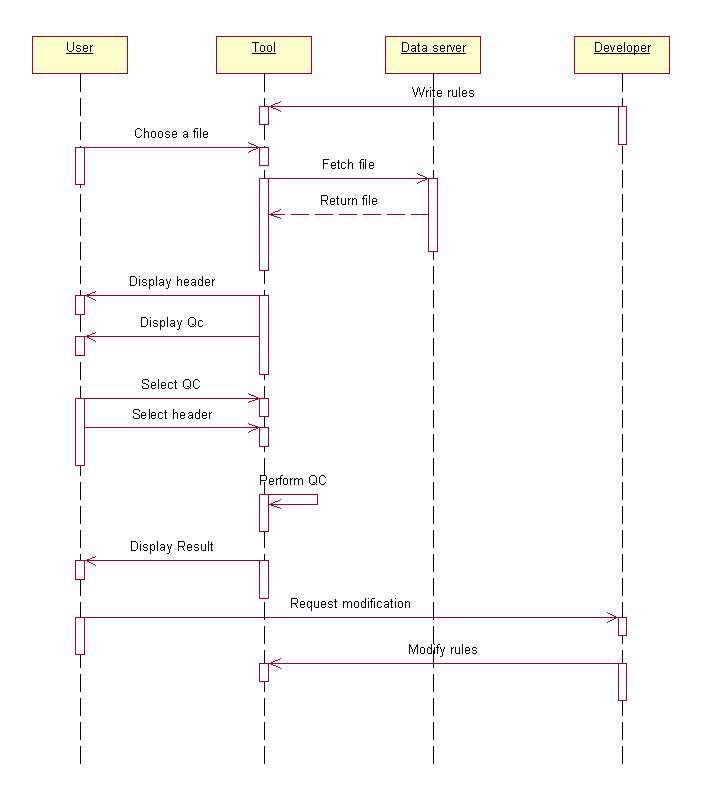
**7.3 Activity Diagram**

**

*Figure* 7.3.1 *Activity Diagram*

**8. INTERACTION SCENARIO**

**8.1 Sequence Diagram**

**

*Figure 8.1. Sequence Diagram*

**9. METHODOLOGIES AND APPROACH**

**9.1 Design**

**9.1.1 Client-Side Design**

The user interface is kept simple and understandable. The user need not take any additional effort to understand the functionality and navigation in the application. The layouts are chosen in such a way that user can easily understand where the input has to be given. Non mandatory fields are mentioned with required descriptions to help the user in giving the correct input.

The following are the main screens and features in this application.

* Home screen
* Quality checks screen
* Result screen

**Home screen:**  
 This is the first screen that any user is greeted with. It is used to do two things:

* **“Browse” Button:** To browse the input file and select it
* **“Anchor” Textbox:** To enter the anchor of the entered excel file
* **“Next” Button:** To transit to the quality checks screen
* **“Error” Dialog:** Pops up when any mandatory field left empty

**Quality checks screen:**  
  This is the second screen that allows user to choose the type of quality check to be performed.

* **“Missing Values” Button:** This button is used to evaluate the missing value present in the selected column of the given input
* **“Duplicates Values” Button:** This button is used to evaluate the number of duplicates present in the selected column of the given input
* **“Range Check” Button:** This button is used to display the number of values lying within and outside the range of the values in the selected column of the input file. Range is mentioned by the user.

**Quality check (Missing values) panel:**  
  This panel displays the available columns in the given excel file once the anchor is selected in the home screen along with file selected and helps the user to evaluate number of missing values present in the selected column once the result button is pressed. This panel consists of:

* “**Column Name” Combo Box:** To select on which column the quality check is to be done
* **“Result” Button:** This button is used to evaluate the given quality check and transit to the result screen

**Quality check (Duplicate values) panel:**  
  This panel displays the available columns in the given excel file once the anchor is selected in the home screen along with file selected and the user can choose on which column he wants to perform the duplicates value check and click on the result button. This panel consists of:

* “**Column Name” Combo Box:** To select on which column the quality check is to be done
* **“Result” Button:** This button is used to evaluate the given quality check and transit to the result screen

**Quality check (Range check) panel:**  
  This panel displays the available columns in the given excel file once the anchor is selected in the home screen along with file selected and the “from” “to” range values as empty boxes where user can specify the range of values that the column needs to contain and the result is displayed once the result button is pressed. This panel consists of:

* “**Column Name” Combo Box:** To select on which column the quality check is to be done
* **“From” Textbox:**  To enter the minimum value of the required range
* **“To” Textbox:** To enter the maximum value of the required range
* **“Result” Button:** This button is used to evaluate the given quality check and transit to the result screen

**Result Screen:**

  This screen helps the user to see the required output based on the quality check selected in the previous screen. There are three items displayed in this screen:

* **“Quality check name” Label**: The quality check which is performed
* **“Column name” Label**: The column name in the input file on which the quality check is done.

**“Result” Label**: Corresponding result of the quality check done

# **9.1.2 Server-Side Design**

**Activities:**

* Storing the file path and anchor of the input file
* Generating the header names present in the input file
* Storing the selected column and required fields for evaluation
* Performing the quality check
* Viewing the result.

**Storing the file path and the anchor of the input:**

Once the next button is pressed in the home screen the mandatory fields are checked, if not empty the field’s values are retrieved and stored in their corresponding variables on the server side and if empty and error dialog is shown to check the mandatory fields.

**Generating the header names present in the input file:**

Based on the anchor and file path input once the next button is pressed the header names of the given input file are generated based on the anchor value. If the anchor value is not mentioned the default value “A1” is taken by the tool. Once the header names are generated they are populated into the pull down item of the column name once the quality check button is pressed.

**Storing the selected column and required fields for evaluation:**

Once the result button is pressed the required fields in each panel are acquired and stored in their respective variables and the corresponding details are stored in the QCDetails class object and sent to the rules engine for the evaluation process.

**Performing the Quality check:**

The result button leads to the evaluation process. Once the required fields of the QC are stored in the respective variables and objects are created accordingly. These objects are put into the rules engine and then based on the rules which are written in the drl file, corresponding rules get fired based on the objects put into the rules engine. The core of the rules engine here is the Inference Engine which matches the facts (objects) with the rules.

**Viewing the result:**

After the evaluation process the UI is set with the output generated by the rules engine. The overall report is created and then those values with respect to the QC selected are displayed immediately along with QC name and column name on which this QC is done.

**9.2 CODE**

“UiDesign.java”

package com.sample;

import java.awt.EventQueue;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import java.awt.CardLayout;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JButton;

import javax.swing.JTextField;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JComboBox;

import org.apache.commons.lang.ArrayUtils;

import com.sample.DroolsTest.QCResult;

import java.awt.Component;

import javax.swing.SwingConstants;

public class UiDesign {

private JFrame frame;

private JTextField filetext;

private JTextField anchortxt;

private JTextField textField\_2;

private JTextField textField\_3;

private JPanel welcome;

private JPanel select;

private JPanel compopan;

private JPanel rangepan;

private JPanel misspan;

private JPanel result;

JFileChooser fileChooser;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

UiDesign window = new UiDesign();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public UiDesign() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 500, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(new CardLayout(0, 0));

fileChooser=new JFileChooser();

/\* Creating Panels For Each Process(Input, QC List and Output)\*/

welcome = new JPanel();

frame.getContentPane().add(welcome, "name\_29245372342829");

welcome.setLayout(null);

welcome.setVisible(true);

select = new JPanel();

frame.getContentPane().add(select, "name\_29247628575820");

select.setLayout(null);

welcome.setVisible(false);

result = new JPanel();

frame.getContentPane().add(result, "name\_29249848410123");

result.setLayout(null);

welcome.setVisible(false);

filetext = new JTextField();

filetext.setBounds(97, 71, 228, 23);

/\* Browsing the file\*/

final JButton browsebtn = new JButton("Browse");

browsebtn.setFont(new Font("Tahoma", Font.PLAIN, 13));

browsebtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent a) {

if(a.getSource()==browsebtn)

{

if (fileChooser.showOpenDialog(null)== JFileChooser.APPROVE\_OPTION) {

filetext.setText(fileChooser.getSelectedFile().getAbsolutePath());

}

}

}

});

browsebtn.setBounds(335, 71, 89, 23);

welcome.add(browsebtn);

welcome.add(filetext);

filetext.setColumns(10);

anchortxt = new JTextField();

anchortxt.setBounds(97, 113, 86, 20);

welcome.add(anchortxt);

anchortxt.setColumns(10);

JButton nextbtn = new JButton("Next");

nextbtn.setFont(new Font("Tahoma", Font.PLAIN, 13));

nextbtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(filetext.getText().equals("")){

JOptionPane.showMessageDialog(welcome, "File Path should not be left empty","Error",JOptionPane.ERROR\_MESSAGE);

}

else{

DroolsTest.filepath=filetext.getText();

if(!anchortxt.getText().equalsIgnoreCase(""))

DroolsTest.AnchorValue=anchortxt.getText();

welcome.setVisible(false);

select.setVisible(true);

}

}

});

nextbtn.setBounds(122, 176, 89, 23);

welcome.add(nextbtn);

/\* Anchor text Label\*/

JLabel lblAnchor = new JLabel("Anchor");

lblAnchor.setFont(new Font("Tahoma", Font.PLAIN, 15));

lblAnchor.setBounds(10, 114, 46, 14);

welcome.add(lblAnchor);

/\* File Path text label\*/

JLabel lblFilePath = new JLabel("File Path");

lblFilePath.setFont(new Font("Tahoma", Font.PLAIN, 15));

lblFilePath.setBounds(10, 75, 64, 20);

welcome.add(lblFilePath);

JLabel lblIfAnchor = new JLabel("\*\* If Anchor is not entered A1 is taken by default\*\*");

lblIfAnchor.setFont(new Font("Tahoma", Font.BOLD, 10));

lblIfAnchor.setBounds(10, 139, 281, 14);

welcome.add(lblIfAnchor);

compopan = new JPanel();

compopan.setBounds(20, 85, 314, 123);

select.add(compopan);

compopan.setLayout(new CardLayout(0, 0));

compopan.setVisible(false);

misspan = new JPanel();

compopan.add(misspan, "name\_30048791662056");

misspan.setLayout(null);

misspan.setVisible(false);

rangepan = new JPanel();

compopan.add(rangepan, "name\_30051614600488");

rangepan.setLayout(null);

rangepan.setVisible(false);

final JButton btnResult = new JButton("Result");

final JComboBox comboBoxmis = new JComboBox();

comboBoxmis.setBounds(110, 33, 106, 20);

/\*Evaluation of the Missing values in the given input\*/

JButton btnQc = new JButton("Missing Values");

btnQc.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DroolsTest.FillList();

comboBoxmis.removeAllItems();

//System.out.println(DroolsTest.HeaderList.length);

for(int i=0;i<DroolsTest.HeaderList.length;i++)

{

comboBoxmis.addItem(DroolsTest.HeaderList[i]);

//System.out.println(DroolsTest.HeaderList[i]);

}

DroolsTest.QCSelected="Missing Values";

compopan.setVisible(true);

misspan.setVisible(true);

btnResult.setVisible(true);

rangepan.setVisible(false);

}

});

btnQc.setBounds(10, 51, 141, 23);

select.add(btnQc);

final JComboBox comboBox\_1 = new JComboBox();

comboBox\_1.setBounds(158, 11, 106, 20);

/\* Evaluation of the Range Check process \*/

JButton btnRan = new JButton("Range Check");

btnRan.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DroolsTest.FillList();

comboBox\_1.removeAllItems();

//System.out.println(DroolsTest.HeaderList.length);

for(int i=0;i<DroolsTest.HeaderList.length;i++)

{

comboBox\_1.addItem(DroolsTest.HeaderList[i]);

//System.out.println(DroolsTest.HeaderList[i]);

}

DroolsTest.QCSelected="Range Check";

compopan.setVisible(true);

rangepan.setVisible(true);

btnResult.setVisible(true);

misspan.setVisible(false);

}

});

btnRan.setBounds(161, 51, 152, 23);

select.add(btnRan);

/\*Evaluation of the duplicates from the given input\*/

JButton btnDuplicates = new JButton("Duplicates Check");

btnDuplicates.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DroolsTest.FillList();

comboBoxmis.removeAllItems();

//System.out.println(DroolsTest.HeaderList.length);

for(int i=0;i<DroolsTest.HeaderList.length;i++)

{

comboBoxmis.addItem(DroolsTest.HeaderList[i]);

//System.out.println(DroolsTest.HeaderList[i]);

}

DroolsTest.QCSelected="Duplicates Check";

compopan.setVisible(true);

rangepan.setVisible(false);

btnResult.setVisible(true);

misspan.setVisible(true);

}

});

btnDuplicates.setBounds(323, 51, 151, 23);

select.add(btnDuplicates);

JLabel lblColumnName = new JLabel("Column Name");

lblColumnName.setFont(new Font("Tahoma", Font.PLAIN, 13));

lblColumnName.setBounds(10, 34, 96, 16);

misspan.add(lblColumnName);

misspan.add(comboBoxmis);

JLabel label = new JLabel("Column Name");

label.setFont(new Font("Tahoma", Font.PLAIN, 13));

label.setBounds(56, 12, 96, 16);

rangepan.add(label);

rangepan.add(comboBox\_1);

JLabel lblFrom = new JLabel("From");

lblFrom.setFont(new Font("Tahoma", Font.PLAIN, 13));

lblFrom.setBounds(24, 55, 38, 18);

rangepan.add(lblFrom);

textField\_2 = new JTextField();

textField\_2.setBounds(56, 55, 80, 20);

rangepan.add(textField\_2);

textField\_2.setColumns(10);

JLabel lblTo = new JLabel("To");

lblTo.setFont(new Font("Tahoma", Font.PLAIN, 13));

lblTo.setBounds(168, 56, 21, 16);

rangepan.add(lblTo);

textField\_3 = new JTextField();

textField\_3.setBounds(189, 55, 86, 20);

rangepan.add(textField\_3);

textField\_3.setColumns(10);

JLabel lblDefaultValues = new JLabel("\*\* Default Values of From and To are set to 0 \*\*");

lblDefaultValues.setFont(new Font("Tahoma", Font.BOLD, 10));

lblDefaultValues.setBounds(34, 86, 252, 14);

rangepan.add(lblDefaultValues);

final JLabel resultlab = new JLabel("");

resultlab.setHorizontalAlignment(SwingConstants.CENTER);

resultlab.setFont(new Font("Tahoma", Font.PLAIN, 13));

resultlab.setBounds(29, 161, 425, 41);

final JLabel QCname = new JLabel("");

QCname.setFont(new Font("Tahoma", Font.PLAIN, 13));

QCname.setBounds(143, 68, 311, 16);

final JLabel columnlab = new JLabel("");

columnlab.setFont(new Font("Tahoma", Font.PLAIN, 13));

columnlab.setBounds(143, 111, 311, 17);

btnResult.setVisible(false);

btnResult.addActionListener(new ActionListener() {

private String[] args;

/\* Server side calling process(Functionality invoking)\*/

public void actionPerformed(ActionEvent e) {

if(DroolsTest.QCSelected.equalsIgnoreCase("Missing Values")){

DroolsTest.colVal=ArrayUtils.indexOf(DroolsTest.HeaderList,comboBoxmis.getSelectedItem().toString());

DroolsTest.ColSelected=comboBoxmis.getSelectedItem().toString();

DroolsTest.main(args);

QCname.setText(DroolsTest.QCSelected);

columnlab.setText(DroolsTest.ColSelected);

resultlab.setText("No. of missing values are: "+DroolsTest.QCResult.getCount());

QCResult.resetCount();

}

else if(DroolsTest.QCSelected.equalsIgnoreCase("Duplicates Check")){

DroolsTest.colVal=ArrayUtils.indexOf(DroolsTest.HeaderList,comboBoxmis.getSelectedItem().toString());

DroolsTest.ColSelected=comboBoxmis.getSelectedItem().toString();

DroolsTest.main(args);

QCname.setText(DroolsTest.QCSelected);

columnlab.setText(DroolsTest.ColSelected);

resultlab.setText("Duplicate values found are: "+DroolsTest.Hashvar.dataset);

}

else if(DroolsTest.QCSelected.equalsIgnoreCase("Range Check")){

DroolsTest.colVal=ArrayUtils.indexOf(DroolsTest.HeaderList,comboBox\_1.getSelectedItem().toString());

DroolsTest.ColSelected=comboBox\_1.getSelectedItem().toString();

DroolsTest.QCDetails.From=Integer.parseInt(textField\_2.getText());

DroolsTest.QCDetails.To=Integer.parseInt(textField\_3.getText());

DroolsTest.main(args);

QCname.setText(DroolsTest.QCSelected);

columnlab.setText(DroolsTest.ColSelected);

resultlab.setText("Values within range are: "+DroolsTest.QCResult.getCount()+"\n "

+ " Values out of range are: "+(DroolsTest.sheet.getRows()-1-DroolsTest.QCResult.getCount()));

}

select.setVisible(false);

result.setVisible(true);

}

});

btnResult.setBounds(100, 219, 89, 23);

select.add(btnResult);

JLabel lblQcName = new JLabel("QC Name:");

lblQcName.setHorizontalAlignment(SwingConstants.RIGHT);

lblQcName.setAlignmentX(Component.RIGHT\_ALIGNMENT);

lblQcName.setFont(new Font("Tahoma", Font.PLAIN, 15));

lblQcName.setBounds(10, 67, 112, 17);

result.add(lblQcName);

JLabel lblColumnName\_1 = new JLabel("Column Name:");

lblColumnName\_1.setHorizontalAlignment(SwingConstants.RIGHT);

lblColumnName\_1.setFont(new Font("Tahoma", Font.PLAIN, 15));

lblColumnName\_1.setBounds(0, 111, 127, 17);

result.add(lblColumnName\_1);

result.add(resultlab);

result.add(QCname);

result.add(columnlab);

}

}

**“DroolsTest.java”**

package com.sample;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.util.ArrayList;

import java.util.HashSet;

import jxl.Cell;

import jxl.CellType;

import jxl.Sheet;

import jxl.Workbook;

import jxl.format.UnderlineStyle;

import jxl.write.Formula;

import jxl.write.Label;

import jxl.write.Number;

import jxl.write.WritableCell;

import jxl.write.WritableCellFormat;

import jxl.write.WritableFont;

import jxl.write.WritableSheet;

import jxl.write.WritableWorkbook;

import jxl.write.WriteException;

import jxl.write.biff.RowsExceededException;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

/\*\*

\* This is a sample class to launch a rule.

\*/

public class DroolsTest {

private static File inputfilename;

private static Workbook w;

protected static Sheet sheet;

protected static int colLen=0;

protected static int colVal=0;

protected static int HeaderSelect=0;

private static int rowBeg=0;

private static int colBeg=0;

protected static String filepath="";

protected static String[] HeaderList;

protected static String AnchorValue="A1";

protected static String QCSelected="Missing Values";

protected static String ColSelected="ClgName";

protected static int MissValueCnt=0;

protected static int DuplicateValueCnt=0;

private static int i;

public static final void main(String[] args) {

try {

// load up the knowledge base

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

/\* Importing the excel file from the given path\*/

// System.out.println(rowBeg+" "+colBeg);

/\*Putting the acquired data into the Rules Engine For Evaluation \*/

QCDetails detail=new QCDetails();

detail.setQCname(QCSelected);

detail.setColName(ColSelected);

for(i=rowBeg+1;i<sheet.getRows();i++)

{

KieSession kSession = kContainer.newKieSession("ksession-rules");

kSession.insert(detail);

kSession.insert(sheet.getCell(colVal,i));

kSession.fireAllRules();

kSession.dispose();

}

//System.out.println("Duplicates found in date elements:"+Hashvar.dataset);

Hashvar.set.clear();

} catch (Throwable t) {

t.printStackTrace();

}

}

/\* Header Names generation process and Storage \*/

public static void FillList()

{

try{

inputfilename= new File(filepath);

w=Workbook.getWorkbook(inputfilename);

sheet= w.getSheet(0);

/\*Anchor conversion process\*/

Anchor anc=new Anchor(AnchorValue);

rowBeg=anc.getRowIndex()-1;

colBeg=anc.getColIndex()-1;

colLen=sheet.getColumns();

System.out.println(colLen);

}

catch(Exception e){}

HeaderList=new String[colLen];

for(i=colBeg;i<colLen;i++)

{

HeaderList[i]=sheet.getCell(i, rowBeg).getContents();

//System.out.println(HeaderList[i]);

}

}

/\*class for duplicates value storage and calculation\*/

public static class Hashvar {

public static final HashSet<String>set=new HashSet<String>();

public static final HashSet<String>dataset=new HashSet<String>();

}

/\* QC Details are included in the below class\*/

public static class QCDetails

{

private static String QCname="";

private static String ColName="";

private static String Result="";

public static int To=0;

public static int From=0;

public static String getQCname() {

return QCname;

}

public static void setQCname(String qCname) {

QCname = qCname;

}

public static String getColName() {

return ColName;

}

public static void setColName(String colName) {

ColName = colName;

}

public static String getResult() {

return Result;

}

public static void setResult(String result) {

Result = result;

}

public static int convert(String s){

return Integer.parseInt(s);

}

}

/\*QC Results are stored in the below class\*/

public static class QCResult

{

private static int count =0;

public static int getCount() {

return count;

}

public static void setCount() {

QCResult.count++;

}

public static void resetCount(){

count=0;

}

}

}

**“Anchor.java”**

package com.sample;

public class Anchor implements Cloneable {

private String label = null;

private int colIndex = 0;

private int rowIndex = 0;

public Anchor(String label) {

this.label = label;

int digit\_count = 0;

int alpha\_count = 0;

for (int i = label.length() - 1; i >= 0; i--) {

char c = label.charAt(i);

if (Character.isDigit(c)) {

int n = Integer.parseInt(new String(new char[] { c }));

rowIndex += (n \* Math.pow(10, digit\_count));

digit\_count++;

} else {

char col\_char = Character.toUpperCase(c);

colIndex += ((col\_char - 'A' + 1) \* Math.pow(26, alpha\_count));

alpha\_count++;

}

}

checkColumnBound(colIndex);

checkRowBound(rowIndex);

}

private void updateLabel() {

StringBuffer \_label = new StringBuffer("");

int c;

checkColumnBound(colIndex);

checkRowBound(rowIndex);

int factor = colIndex / 26;

int remain = colIndex % 26;

if (remain == 0) {

if (factor == 1) {

\_label.append("Z");

} else {

c = ('A' + factor) - 2;

\_label.append((char) c);

\_label.append("Z");

}

} else {

if (factor == 0) {

c = ('A' + remain) - 1;

\_label.append((char) c);

} else {

c = ('A' + factor) - 1;

\_label.append((char) c);

c = ('A' + remain) - 1;

\_label.append((char) c);

}

}

\_label.append(rowIndex + "");

label = \_label.toString();

}

public String getLabel() {

return label;

}

public int getColIndex() {

return colIndex;

}

public int getRowIndex() {

return rowIndex;

}

/\*\*

\* Returns the index starting from 0

\*

\* @return

\*/

public int getJavaRowIndex() {

return rowIndex - 1;

}

public int getJavaColIndex() {

return colIndex - 1;

}

public void setLabel(String label) {

this.label = label;

}

public void setColIndex(int index) {

colIndex = index;

updateLabel();

}

public void setRowIndex(int index) {

rowIndex = index;

updateLabel();

}

public Object clone() {

try {

return­­­­­­­­­­­­ super.clone();

} catch (CloneNotSupportedException e) {

throw new Error(e);

}

}

private void checkColumnBound(int index) {

if (index > 702) {

throw new IndexOutOfBoundsException(

"Excel spreadsheet column index " + index

+ " exceeds the limit of ZZ(702 columns)");

}

}

private void checkRowBound(int index) {

if (index > 1048575) {

throw new IndexOutOfBoundsException(

"Excel spreadsheet column index " + index

+ " exceeds the limit of ZZ(702 columns)");

}

}

}

**“Sample.drl”**

package com.sample

import com.sample.DroolsTest.QCDetails;

import com.sample.DroolsTest.QCResult;

import jxl.Cell;

import java.util.HashSet;

import com.sample.DroolsTest.Hashvar;

import java.lang.Integer;

rule "Missing values"

when

q: QCDetails(QCname.equalsIgnoreCase("Missing Values")) and

c: Cell( getContents().length()==0)

then

QCResult.setCount();

end

rule "Duplicates detection"

when

q: QCDetails(QCname.equalsIgnoreCase("Duplicates Check")) and

c: Cell(!(Hashvar.set).add(getContents()))

then

(Hashvar.dataset).add(c.getContents());

end

rule "Range Check"

when

q: QCDetails(QCname.equalsIgnoreCase("Range Check"))

c: Cell((q.convert(getContents()))>=QCDetails.From && (q.convert(getContents()))<=QCDetails.To)

//eval(q.convert(c.getContents())>=q.From)

//Boolean(booleanValue == true) from q.convert(c.getContents()) >= q.From

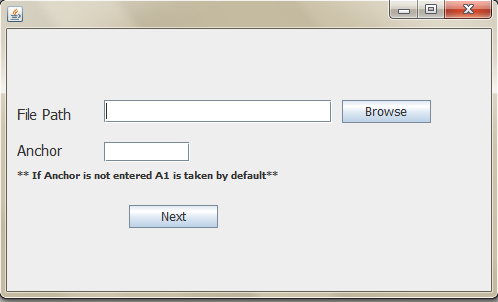
then

//System.out.println("...");

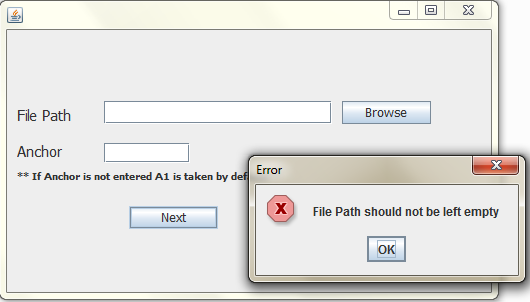
QCResult.setCount();

end

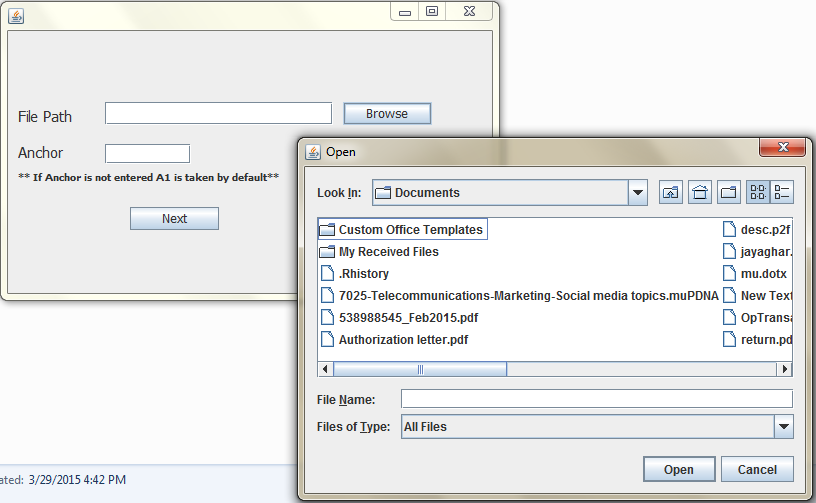
11. OUTPUT SCREENSHOTS



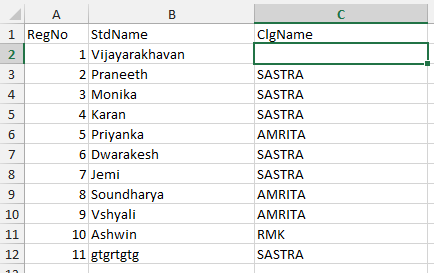
*Figure 11.1. Home Screen*



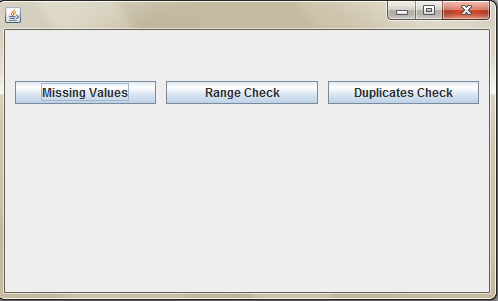
*Figure 11.2. Error Dialog (When file path left empty)*



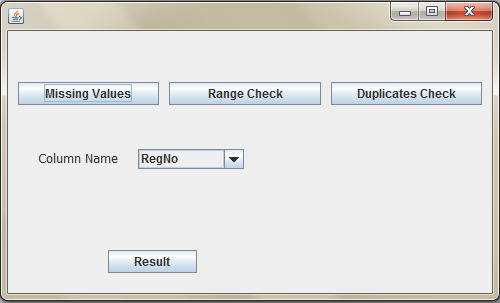
*Figure 11.3. File Browsing Window*



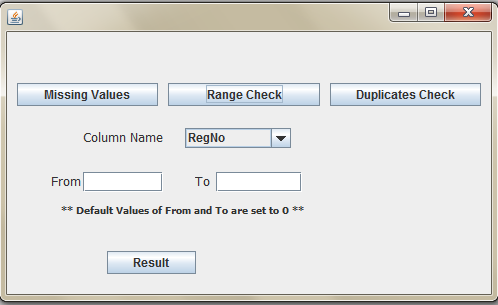
*Figure 11.4. Sample Dataset*



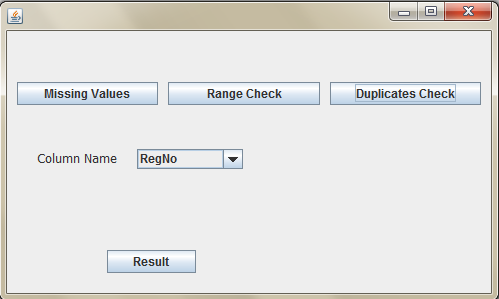
*Figure 11.5. Quality Check Screen*



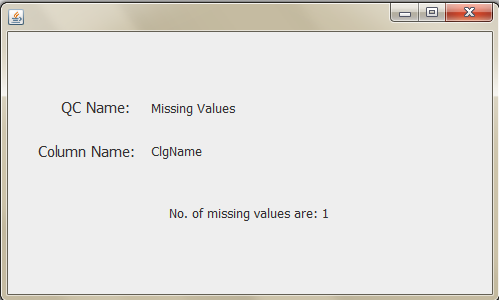
*Figure 11.6. Missing Values Panel Display*



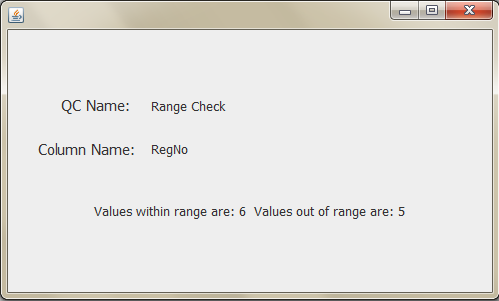
*Figure 11.7. Range Check Panel Display*



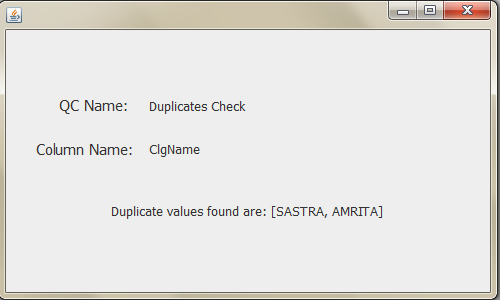
*Figure 11.8. Duplicates Check Panel Display*



*Figure 11.9. Result Screen of Missing Values*



*Figure 11.9. Result Screen of Range Check*



*Figure 11.10. Result Screen of Duplicates Check*

**12. CONCLUSION**

The tool developed has been subjected to various tests and measures that include factors to check starting from the least to-be-considered to that of the most important factor. This tool needs to be further integrated with the Analysis Automation tool as a quality checking component. Integration is being done by adding a new layer as Rules Engine layer in the Analysis Automation tool architecture. A special thanks to the all the faculty members for their cooperation in successfully producing the project report.

**13. REFERENCES**

* <http://en.wikipedia.org/wiki/Rete_algorithm>
* <https://docs.jboss.org/drools/release/6.0.0.Final/drools-expert-docs/pdf/drools-expert-docs.pdf>
* <http://swt-www.informatik.uni-hamburg.de/uploads/media/Diplomarbeit_Eryk_Lagun.pdf>