DESIGN AND IMPLEMENTATION OF MASSIVE EXCEL DATA INTELLIGENT IMPORT SYSTEM TO DATABASE (MYSQL) BY USING LIBRARIES

[1]KAMALAKANT L BAWANKULE, [2]N B.RAUT

[1]M.Tech 4th (CSE) Student, N.U.V.A. Nagpur
[2]Assistant Prof., Gurunanak College of Engineering, Nagpur

Abstract—To import nearly 1,0000 to attribute to the tables of 10-20 column information of students and papers database of Excel format into the target tables of MySQL database, we developed the intelligent import system which distinguished the simple import system. The system uses Netbeans/Eclipse as the development tool, object-oriented language java as the programming language, MySQL as the background database, POI-HSSF as the apache Library for reading data from excel, used AWT and SWING for design of an application ,and uses the mysql-connector as connection technology to database. The system realizes to import the massive data of Excel format into the MySQL database intellectually that convenient for integrating and managing the multiple data required for regular examination process in the institutes and universities.HSSF is the POI Project's pure Java implementation of the Excel '97(-2007) file format. XSSF is the POI Project's pure Java implementation of the Excel 2007 OOXML (.xlsx) file format.HSSF and XSSF provide ways to read spreadsheets create, modify, read and write XLS spreadsheets.

Keywords—Database, JAVA, POI-HSSF, HSSF, MySQL, XSSF.

I. INTRODUCTION

Today in many industries employee data, salary data, purchase data and etc is present in excel format. To create reports, manipulate and update data we need to search data in the excel. Data analysis in excel format is very tedious, while generating report and formats. Many industries also has their old data in excel so they need an application to import data from excel to database (MySQL).Industry should be able to reuse their old data. POIFSFileSystem is the complete file system which will be used to handle excel files.FileInputStream is used to load the excel file to the application .Data importing application has facility to read data from excel and import it into database (MySQL).Excel data is in table format same as database, apache library functions are used to read each column in the spreadsheet and store data into vector or an array. The data in vector or an array is compiled in a single vector. Direct import of vector to the database is done through connecting to mysql.Thier is no limit of reading data massive data can be read through excel and can be stored into database. Importing data from excel will help to generate report and make it easy way to analyze data. Data imported can be read, write and can be updated if needed. It becomes very much easy way to for importing data with apache library HSSF.

II. APACHE POI-HSSF

HSSF is the POI Project's pure Java implementation of the Excel '97(-2007) file format. XSSF is the POI Project's pure Java implementation of the Excel 2007 OOXML (.xlsx) file format.

1) HSSF and XSSF provide ways to read spreadsheets create, modify, read and write XLS spreadsheets. They provide:

- low level structures for those with special needs
- an event model api for efficient read-only access
- a full usermodel api for creating, reading and modifying XLS files

2) An alternate way of generating a spreadsheet is via the Cocoon serializer (yet you'll still be using HSSF indirectly). With Cocoon you can serialize any XML datasource (which might be a ESQL page outputting in SQL for instance) by simply applying the stylesheet and designating the serializer.

3) If you're merely reading spreadsheet data, then use the eventmodel api in either the org.apache.poi.hssf.eventusermodel package, or the org.apache.poi.xssf.eventusermodel package, depending on your file format.

	HSSF (.xls)		XSSF (.xlsx)		
	eventmodel	usermodel	eventmodel	usermodel	SXSSF
API Type	Streaming (ala SAX)	In memory tree (ala DOM)	Streaming (ala SAX)	In memory tree (ala DOM)	Buffered Streaming
CPU and Memory Efficiency	Good	Varies	Good	Varies	Good
Forward Only	Yes	No	Yes	No	Yes
Read Files	Yes	Yes	Yes	Yes	No
Write Files	No	Yes	No	Yes	Yes
Feature:					
create sheets / rows / cells	No	Yes	No	Yes	Yes
styling cells	No	Yes	No	Yes	Yes
delete sheets / rows/cells	No	Yes	No	Yes	No
shift rows	No	Yes	No	Yes	No
cloning sheets	No	Yes	No	Yes	No
formula evaluation	No	Yes	No	Yes	No
cell comments	No	Yes	No	Yes	No
pictures	No	Yes	No	Yes	Yes

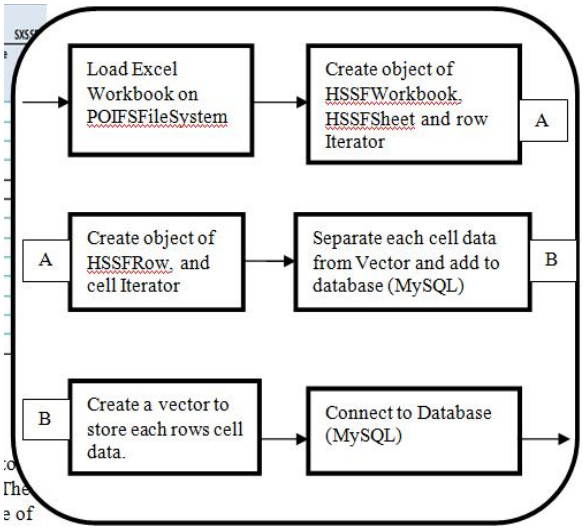
III. DATABASE (MYSQL)

Database has to be maintained by the system to store information about students, invigilators, class rooms etc. The various updates must be saved and stored in the database of the system. Therefore MySQL, relational database that can handle large amount of data on relatively cheap hardware has been used. All the reports, formats will be made available on a single button click. The automated system helps to save the time and the laborious work. MySQL is the world's most used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. Universities, internet service providers and nonprot organizations are the main users of MySQL, mainly because of its price. Free software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available and over additional functionality.

IV. DESIGN AND IMPLEMENTATION

I] Design

Load Excel workbook which contains data in rows and columns i.e. in the form of table. Create object of HSSFWorkbook, HSSFSheet and row Iterator which will help to read the excel sheet present in the workbook. Create object of HSSFRow, and cell Iterator to read excel sheet row by row and each cell in each row. Declare a vector of initialize size as 0; add each read cell data to vector. Data of the workbook will be collected completely in the vector. Find the size of vector, parse the vector and separate each cell data in vector. Connect to database (MySQL) for adding data of each cell to database. Add data into database from vector in the table for each row and for each cell.



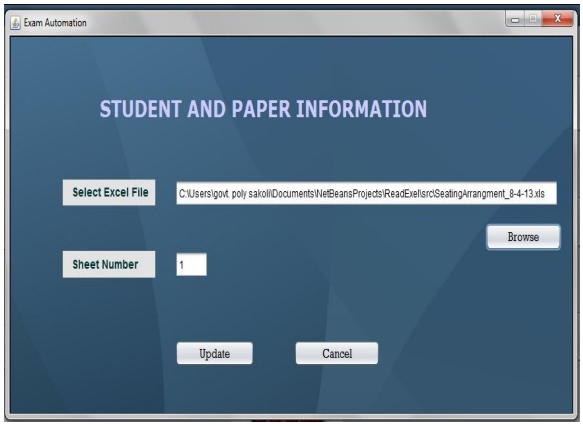
II] Implementation/Code

```
import com.mysql.jdbc.PreparedStatement;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Iterator;
import java.util.Vector;
import java.util.regex.Pattern;
import javax.swing.JOptionPane;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import
org.apache.poi.hssf.usermodel.HSSFWorkbook;
import
org.apache.poi.poifs.filesystem.POIFSFileSystem;

        Vector cellVectorHolder = new Vector();
        Vector cellVectorHolder1 = new Vector();

  // Code to read the Excel worksheet given by user
try {
FileInputStream myInput = new
FileInputStream(fileName);

POIFSFileSystem myFileSystem = new
 POIFSFileSystem(myInput);

HSSFWorkbook myWorkBook = new
HSSFWorkbook(myFileSystem);

HSSFSheet mySheet =
myWorkBook.getSheetAt(sheetno);

Iterator rowIter = mySheet.rowIterator();

while (rowIter.hasNext()) {

HSSFRow myRow = (HSSFRow) rowIter.next();
Iterator cellIter = myRow.cellIterator();
Vector cellStoreVector = new Vector();

while (cellIter.hasNext()) {

HSSFCell myCell = (HSSFCell) cellIter.next();
cellStoreVector.addElement(myCell);
String stringCellValue1 = myCell.toString();
                }

cellVectorHolder.addElement(cellStoreVector);
              }
          }
             catch (Exception ef) {

JOptionPane.showMessageDialog(null, "Sheet index
is out of range (0..etc)");
```

```
JOptionPane.showMessageDialog(null, "All Sheet
Updated Successfully");
                this.hide();

            }
       int size = cellVectorHolder.size();
 int k = 1, t = 2, r = 3, s = 4, p = 5, m = 0, n = 0,
       q = 0, b = 0, a = 0, c = 0;
 String srno[] = new String[size];
 String seatno[] = new String[size];
 String instcode[] = new String[size];
 String corsecode[] = new String[size];
 String yearcode[] = new String[size];
 String mascode[] = new String[size];

 for (int i = 0; i < cellVectorHolder.size(); i++) {
 Vector cellStoreVector = (Vector)
 cellVectorHolder.elementAt(i);
 for (int j = 0; j < cellStoreVector.size(); j = k + 6) {
  HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 srno[m] = stringCellValue;
 //System.out.print(srno[m]);
 m = m + 1;

 }
 for (int j = k; j < cellStoreVector.size(); j = k + 5) {

 HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 seatno[n] = stringCellValue;
 //System.out.print(seatno[n]);
 n = n + 1;

 }
 for (int j = t; j < cellStoreVector.size(); j = t + 4) {

  HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 instcode[q] = stringCellValue;
 //System.out.print(instcode[q]);
 q = q + 1;
              }
 for (int j = r; j < cellStoreVector.size(); j = r + 3) {
 HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 corsecode[c] = stringCellValue;
 //System.out.print(corsecode[c]);
                  c = c + 1;
              }
 for (int j = s; j < cellStoreVector.size(); j = s + 2) {
 HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 yearcode[b] = stringCellValue;
 //System.out.print(yearcode[b]);
 b = b + 1;
              }
 for (int j = p; j < cellStoreVector.size(); j = p + 1) {
 HSSFCell myCell = (HSSFCell)
 cellStoreVector.elementAt(j);
 String stringCellValue = myCell.toString();
 mascode[a] = stringCellValue;
 //System.out.print(mascode[a]);
 a = a + 1;
              }

          }
```

V. SYSTEM DESIGN

The below image shows the application system developed for the massive excel data intelligent import system to database (MySQL) by using apache POI libraries.The excel row, column table format data is read with the help of use of Apache –POI library. After reading the data it is being imported to MySQL database.Application is very much useful during the massive import of data from excel to database, so that user can perform different types of operation on the data present in database.



The above developed system needs input as excel file and user need to give the sheet number he wants to read. Implementation of the complete system is done with the help of JAVA .To read data from excel here used the apache poi library. Spreadsheet data read is then imported in database (My SQL) with the help of J-Connector of MySQL and JAVA.

CONCLUSION

Massive Data read from excel in table format and imported same directly to database with the help of APACHE POI-HSSF. Less time required for planning and no need of typing the complete details of particular seat number of candidate. Less man power required for planning and arranging the table data. No

tiredness and No frustration. No chance for mistake as all the reports are system generated. Proposed System will be helping to save the time, man power and laborious work. Java application will occupy very less memory in the system in the bytes.The system will be user friendly which will help user to make the examination a grand success.

REFERENCES

[1] http://howtodoinjava.com/2013/06/19/readingwriting-excel-files-in-java-poi-tutorial/, "Reading/writing excel files in java : POI tutorial".

[2] http://poi.apache.org/spreadsheet/quick-guide.html "Busy Developers' Guide to HSSF and XSSF Features".

[3] http://www.vogella.com/articles/JavaExcel/article.html,"Excel and Java - Read and Write Excel with Java"

[4] http://mrbool.com/reading-excel-file-with-java/24562,"Reading Excel file with Java".

[5] http://www.javacoderanch.com/how-to-read-excel-file.html,"How to read Excel file ?".

[6] HTTP:// JAVA-READ-WRITE-EXCEL-FILE-APACHE,"READ / WRITE EXCEL FILE IN JAVA USING APACHE POI"

[7] http://javabeginnerstutorial.com/code-base/write-excel-file/," Read and Write Excel with Java using PoI".

[8] Andrew C. Oliver, Nicola Ken Barozzi "POI-HSSF and POI-XSSF - Java API To Access Microsoft Excel Format Files."

[9] Zhang Ning, Jia Zi-Yan, Shi Zhong-Zhi, Research on Technology of ETL in Data Warehouse Computer Engineering and Applications, vol.24, no.12, 2002, pp.212-216.

[10] http://dev.mysql.com/doc/connector-j-usagenotes-connect-drivermanager.html,"Connecting to MySQL Using the JDBC DriverManager Interface".

[11] http://java67.blogspot.in/2013/02/how-to-connect-mysql-database-from-java.html," Java program to connect MySQL database to execute query".

[12] http://www.javaworkspace.com/connectdatabase/connectMysql.do, "CONNECT TO MYSQL 5.1".

[13] http://poi.apache.org/apidocs/org/apache/poi/hssf/usermodel/HSSFDataFormat.html, "Class HSSFDataFormat".

[14] http://poi.apache.org/apidocs/org/apache/poi/hssf/usermodel/HSSFRichTextString.html " Class HSSFRichTextString".

[15] http://poi.apache.org/spreadsheet/ ,"POI-HSSF and POI-XSSF – Java API To Access Microsoft Excel Format Files".

[16] http://docs.oracle.com/cd/B10501_01/server.920/a96652/ch02.htm, "What Is the Import Utility?".

★★★