

# DATAVISION

## ***Abstract***

This project Implements a Database Independent DB Report Generator, which can generate reports from databases or text data files. Any database with an available JDBC driver should work: Oracle, PostgreSQL, MySQL, Informix, hsqldb, Microsoft Access, Progress, and more. Columns read from text files can be separated by any character. Report descriptions are stored as XML files. This means not only the DB Report GUI but also edit reports using favorite text editor. When different companies provide different drivers to connect to their databases, we seldom found a server side application to interact with any database like oracle, my -sql offered by a database providers and also there is a need to have a user friendly interface to query the database and obtain the results from the server.

## ***Existing System***

The existing system has different tools or consoles for each and differently. The user feels uneasy to switch over to a different database by learning the how to use console for that database. User has to make connection each time for each database and he/she is limited to some specified tables, user feel very uneasy to generate report. Whatever report is generated by user is not look attractive. This is the main drawback of the system. The system is not user friendly

## ***Proposed System***

This project Implements a Database Independent DB Report Generator, which can generate reports from databases or text data files. Any database with an available JDBC driver should work: Oracle, PostgreSQL, MySQL, Informix, hsqldb, Microsoft Access, Progress, and more. Columns read from text files can be separated by any character. Report descriptions are stored as XML files. This means not only the DB Report GUI but also edit reports using favorite text editor.

## ***Scope of the System***

The scope of the project has no boundaries. The system can work on any platform and connect to any database like ORACLE, MYSQL, BD2, MSACCESS and many more. By this system you can modify your report according to you like font Report header, Report footer, Page Header, Page footer. You can also export your report into HTML, PDF, .xls etc. By this system you need not to worry about query that how you fetch column form the database you just provide the link to the database it automatically create a link with the database and fetch all table to you from the database after you just drag and drop field from the table for your choice and prepared your report.

## ***Module Description***

*The proposed system consists of 4 modules.*

**1. User Interface Module:** *This module is responsible for developing user interface. This is developed using java.swing package.*

**2. Script Generating Module:** *This module is responsible for generating scripts when formulas are used. For generating scripts JRuby is used.*

**3 Query Processing Module:** *This module contains functionality to process queries, sort records, process formulas etc.*

**4. Exporting Module:** *In this module the functionality to export generated report to different formats is developed. The exported file can be html, xml, pdf(portable document format), csv(comma separated values), tab separated values, xls .*

## ***Features to be implemented***

- *Drag-and-drop report builder.*
- *Runs anywhere Java runs: Windows, Linux, Mac OS X, BSD, Solaris, and more.*
- *Works with any database that has a JDBC driver: Oracle, PostgreSQL, MySQL, Informix, hsqldb, Microsoft Access, DB/2, and more.*
- *Report headers and footers, page headers and footers, group headers and footers.*
- *Report field formatting: color, numeric and date format, italic, bold, borderlines, etc.*
- *Subtotals by group and grand total.*
- *Record sorting.*
- *Record selection (SQL WHERE clause for database data sources, Ruby script that accepts/rejects records for character-separated file data sources).*
- *Formulas. The formula language is Ruby (via J Ruby).*
- *Run-time parameters; asks the user for values when the report runs; if running from the command line, reads values in from an XML file.*
- *Run-time variables (again thanks to Ruby). A Ruby script run at the start of each report run is the perfect place to set all your initial values.*
- *User-defined SELECT clause columns; useful for calling stored procedures or SQL functions.*
- *Hide columns and entire sections.*
- *Reports can read data from different data sources. Currently, data sources are defined for databases and text data files (comma-separated, tab-separated, etc.)*

## ***Technologies to be used***

- *Operating System: WINDOWS XP/2003 or Linux/Solaris*
- *Programming Language: Java*
- *Data Structures: JAVA Collection API*
- *Printing Process: JAVA Printing API*
- *Graphical User interface: JAVA Swing*
- *Network Programming: JAVA Socket API*
- *IDE/Workbench: Eclipse with MyEclipse Plug-in*

## ***Hardware requirements***

- |                       |       |                                |
|-----------------------|-------|--------------------------------|
| • <b>Processor</b>    | ----- | <b>Intel P-IV based system</b> |
| • <b>RAM Capacity</b> | ----- | <b>512MB</b>                   |
| • <b>Hard Disk</b>    | ----- | <b>20GB</b>                    |
| • <b>CD-ROM Drive</b> | ----- | <b>32 HZ</b>                   |
| • <b>KEYBOARD</b>     | ----- | <b>108 Standard</b>            |