JAVA-MySQL based Asset Management System

Rishabh Gulati

Abstract— Dealing with enormous amount of information has become an integral part of complex systems across all sectors.This concept of efficient management of data when extended to the personal level can go a very long way in saving an individual's time. In the current era, an individual owns a variety of assets ranging from monetary assets like Fixed Deposits, Bank Accounts to Immovable Assets like Land, Houses, etc. This paper delineates the concept of Asset Management using a JAVA-MySQL based system and also describes the various operations this system incorporates for managing the different assets efficiently.

Index Terms— Asset Management, Data Flow, Data Management, Java, JavaServer Page, JDBC Driver, MySQL

1 INTRODUCTION

The Asset Management System deals with the basic problem of managing enormous data. It caters to the organization of data in a well structured form facilitating its maintainance and expansion.

It makes use of the JAVA programming language to process the data and MySQL database management system for storing the data. This system uses a web-browser for user interaction and provides the user the facility to add, modify and view asset information.

The JavaServer Pages are used for collecting, modifying and displaying information to the user. The system also incorporates user authentication leading to better data security.

2 SYSTEM REQUIREMENTS

2.1 Software

2.1.1 Eclipse IDE (Juno)

Eclipse is an open source Integrated Development Environment (IDE) used extensively for web-application development. It has various versions like Callisto, Europa, Ganymede, Galileo, Helios, Indigo and Juno. This system uses Eclipse for processing the information received via JavaServer Pages. It is also used for interacting with the database, in this case MySQL, for storing, modifying and retrieving information.

2.1.2 Apache Tomcat (7.0.29)

Apache Tomcat is an open source web server and servlet container. Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle Corporation, and provides a pure Java HTTP web server environment for Java code to run. This would be used to host the JavaServer Pages required for user interaction.

2.1.3 MySQL

MySQL is an open source database management system. In this asset management system, MySQL is used for organizing the user information. The MySQL Workbench is used for creating and executing SQL queries. It also helps in designing and managing the database more efficiently.

It provides a console to gain better visibility into a database. Any further expansion or manipulation to the existing data structure can be achieved expeditiously using MYSQL workbench.

2.1.4 Web Browser

A web browser is a software application for retrieving and presenting information on the World Wide Web. This system is hosted using the Apache Tomcat web server and can be accessed via a web browser by entering the URL of the hosted JavaServer Page. All the user-system interactions are done through the web browser.

3 DATA FLOW AND MANAGEMENT

When the user accesses the system using a web browser, he is presented with a login page. The user information is maintained in the MySQL database. On entering the credentials, username and password, the user is authenticated with the values present in the MySQL database. After successful authentication, the user gets access to his personal asset management system. There are various operations which the user is presented with, namely,

1) Adding an asset.
2) Modifying an existing asset entry.
3) Viewing asset information.

The data flow and management with regards to the above three operations is detailed in succeeding subsections.

3.1 Adding an asset

By selecting this option, the user is presented with a list of asset types. This list contains all the major asset types that can be possessed by an individual. A few of the asset types presented to the user are listed below –

1) Immovable asset (Property)
2) Cash At Bank (Savings)
3) Fixed Deposits
4) Mutual Funds
5) Insurance Policies
6) Appliances
7) Provident Fund
8) Miscellaneous

All the asset types provide certain input options to the user corresponding to that asset. On selection of an asset, a corresponding form with several input fields like Asset Name, Asset Value, Asset Purchase date, Asset Physical file reference are presented to the user.

Upon form submission, several client side validations are addressed for each input field for having consistency in data processing and storage. The user is intimated with appropriate error messages in case if a validation fails. After successful client-side validation the data received by the Java Beans, which are bound to the corresponding form input fields, is validated. In case the server-side validation is not successful, an appropriate message is displayed back to the user giving information about the cause of the error.

After successful server-side validation, the connection with MySQL database is established using MySQL JDBC (Java Database Connectivity) driver. For setting up this connection, there are three essential parameters required, namely, the MySQL-JDBC URL, the database Username and the database Password. Using these three parameters a connection is established with the MySQL database. If the connection fails to establish, an appropriate error message is displayed to the user.

Once the connection is established, the existing data structure information of the user is extracted. The data structure information maps the users with their associated data tables. This information portrays the existing tables for different types of assets for a particular user. If a table already exists for a particular asset for a particular user, the information is added to the table. In case it is the first entry of a particular asset type a new table is created and the asset information is updated. This table information is then appended to the user's data structure information for future use. After successful addition of the asset, the user is intimated about the same.

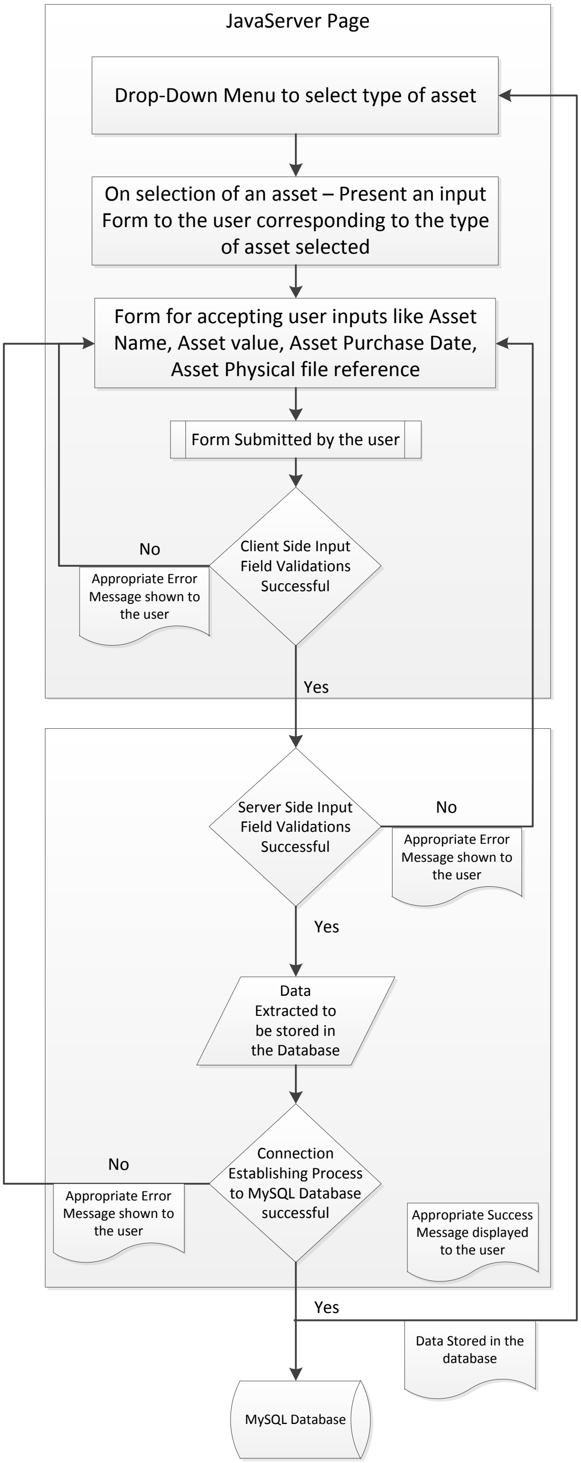

JavaServer Page
Drop-Down Menu to select type of asset
On selection of an asset – Present an input Form to the user corresponding to the type of asset selected
Form for accepting user inputs like Asset Name, Asset value, Asset Purchase Date, Asset Physical file reference
Form Submitted by the user
Client Side Input Field Validations Successful
No
Appropriate Error Message shown to the user
Yes
Server Side Input Field Validations Successful
No
Appropriate Error Message shown to the user
Yes
Data Extracted to be stored in the Database
Connection Establishing Process to MySQL Database successful
No
Appropriate Error Message shown to the user
Appropriate Success Message displayed to the user
Yes
Data Stored in the database
MySQL Database

Figure1: adding an asset – Data Flow

3.2 Modifying an existing asset

Modifying an existing asset entry gives a user the flexibility to update his asset information. When the user selects this option, he is provided with a list of asset types in which there are existing entries. This list is extracted from the users existing data structure information, by establishing a connection with MySQL database. Due to this, the user is presented with a drop-down menu with only those asset types which have a minimum of one asset entry.

On choosing a particular asset type, all the entries for that asset are staged to the user in the form of a list. The user can then select an entry from the list for updation. Upon selection of a particular entry, the input fields in form for updation are embedded with the previous values which allow the user to update/modify only the desired field(s). When the user submits this form, it undergoes the same cycle of events which take place while adding an asset.

In case the user wants to delete an existing entry, he can do so by selecting a particular entry and using the delete entry option. The user is prompted before the deletion transaction is initiated. Once the user's consent is acknowledged by the system, a connection with MYSQL database is established and the particular entry is removed. After successful deletion of an entry the user is notified about same. The JavaServer Page is refreshed with an updated asset list.

3.3 Viewing Asset Information

This feature provides the facility of viewing asset information for all or selected asset types. When the user selects this option, the user is presented with a list of asset types, extracted by using his data structure information. The user is provided with check-boxes alongside this list of asset types. Also, an option of Select All asset types is provided to the user in case there is more than one asset type available for viewing.

Upon submission of this form, validation with regards to the checkbox selection is performed. If the user submits the form without selecting any checkbox, he is intimated with regards to the same. In case one or more than one checkbox selection is made by the user, this information is processed to extract detailed information for the different entries of the selected asset types.

The connection to MySQL Database is established and the database is queried for detailed information of the selected asset types. The information with regards to the various entries in a particular asset type is collected in a list. The number of lists created equal the number of asset types selected by the user. These lists are then fed back to the JavaServer Page for viewing.

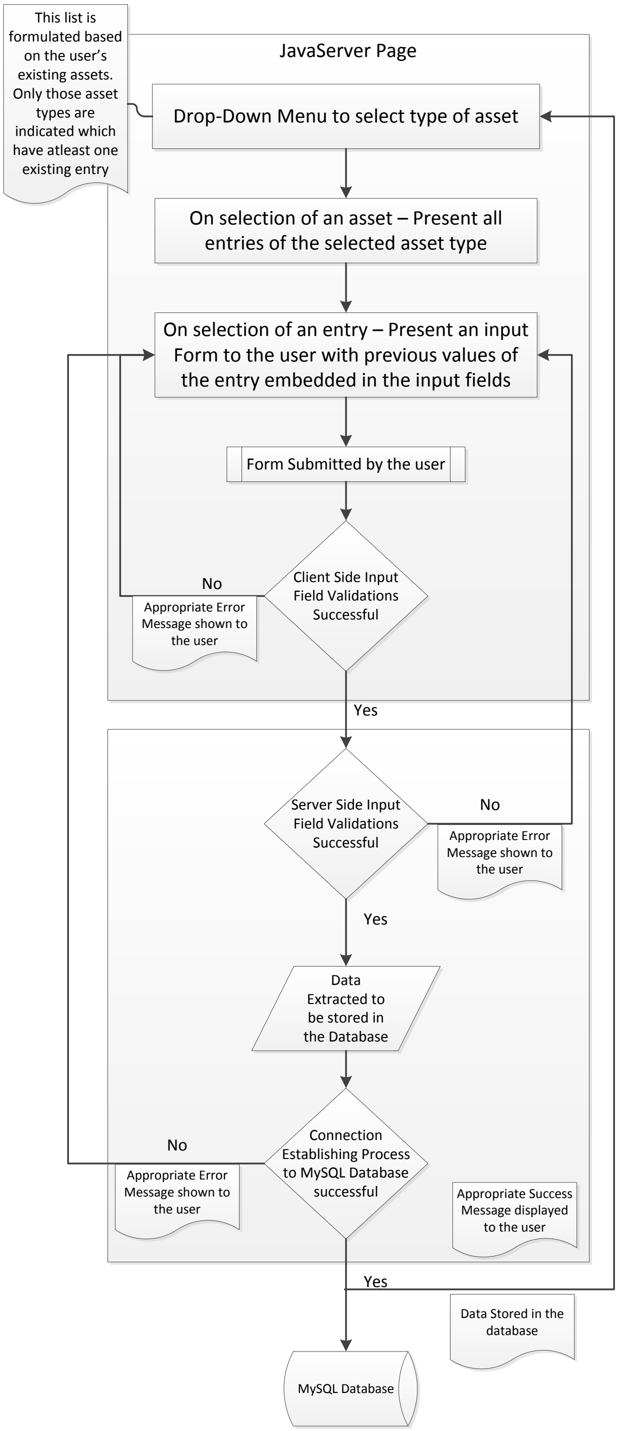

This list is formulated based on the user's existing assets. Only those asset types are indicated which have atleast one existing entry
JavaServer Page
Drop-Down Menu to select type of asset
On selection of an asset – Present all entries of the selected asset type
On selection of an entry – Present an input Form to the user with previous values of the entry embedded in the input fields
Form Submitted by the user
Client Side Input Field Validations Successful
No
Appropriate Error Message shown to the user
Yes
Server Side Input Field Validations Successful
No
Appropriate Error Message shown to the user
Yes
Data Extracted to be stored in the Database
Connection Establishing Process to MySQL Database successful
No
Appropriate Error Message shown to the user
Appropriate Success Message displayed to the user
Yes
Data Stored in the database
MySQL Database

Figure2: modifying an asset – Data Flow

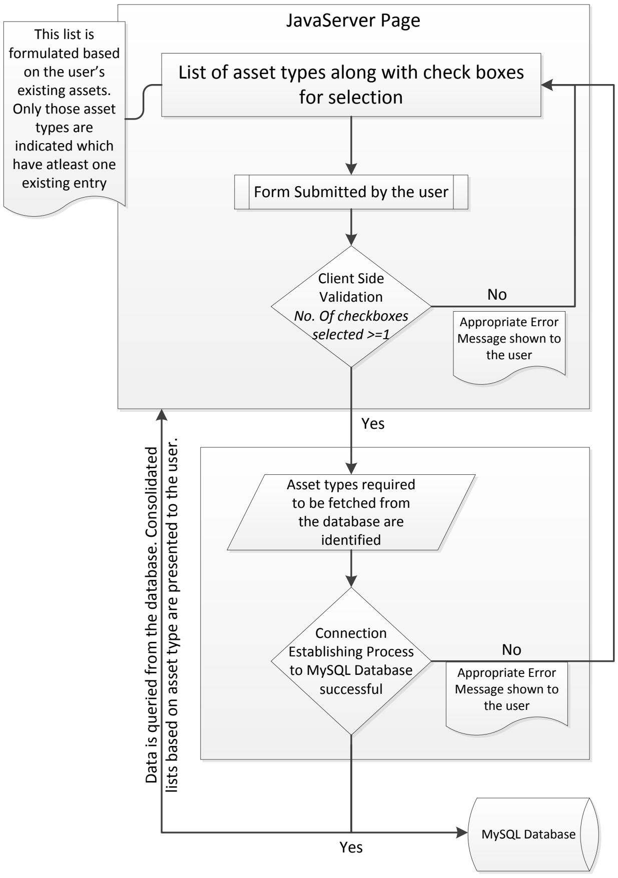

JavaServer Page
This list is formulated based on the user's existing assets. Only those asset types are indicated which have atleast one existing entry
List of asset types along with check boxes for selection
Form Submitted by the user
Client Side Validation
No. Of checkboxes selected >=1
No
Appropriate Error Message shown to the user
Yes
Asset types required to be fetched from the database are identified
Connection Establishing Process to MySQL Database successful
No
Appropriate Error Message shown to the user
Yes
MySQL Database
Data is queried from the database. Consolidated lists based on asset type are presented to the user.

Figure3: viewing asset information – Data Flow

4 Conclusion

The asset management system is designed for efficient management of several assets possessed by an individual. It allows the user to add information about different assets in a quicker and systematic way. The data is organised efficiently to aid future expansion of the system.

The user is facilitated with the feature of updating the existing information of an asset entry. This feature is especially helpful while maintaining assets like Bank Accounts and Mutual funds, since their values are constantly changing.

As observed in figure4, the asset management system provides the facility to view the different asset entries pertaining to different asset types. This feature helps to keep a track of all assets in a much quicker and efficient way as compared to the traditional method of poring over innumerable files.

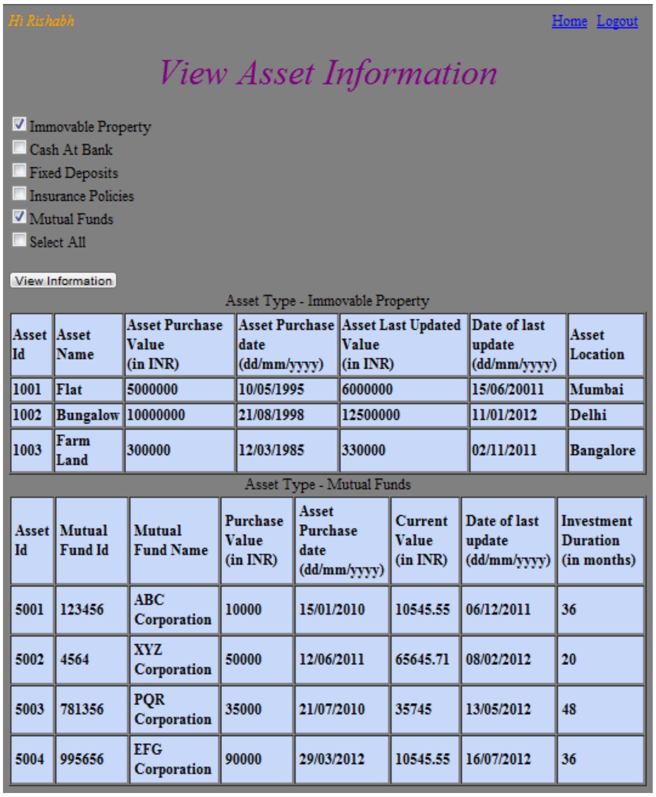
5 Future Scope

Since, the system is organized in a very structured manner using MySQL database, it is possible to expand the system to encompass even more information. More number of attributes regarding an asset can be stored, modified and viewed depending upon the requirement.

The number of possible operations on the data can be increased. Features like sorting asset entries based on their value, last modified date, etc can be incorporated.

Computational features such as calculating the current value of assets, calculating the increase or decrease in the value of an asset can also be integrated with the existing system.

A feature to create soft-copy in different file formats like .xls, .pdf, .doc, etc of different assets possessed by a user can also be unified in the system. This feature would help the user to instantly process an electronic copy of the list of assets currently maintained in the system.



View Asset Information

☑ Immovable Property
☐ Cash At Bank
☐ Fixed Deposits
☐ Insurance Policies
☑ Mutual Funds
☐ Select All

Asset Type - Immovable Property

Asset Id	Asset Name	Asset Purchase Value (in INR)	Asset Purchase date (dd/mm/yyyy)	Asset Last Updated Value (in INR)	Date of last update (dd/mm/yyyy)	Asset Location
1001	Flat	5000000	10/05/1995	6000000	15/06/20011	Mumbai
1002	Bungalow	10000000	21/08/1998	12500000	11/01/2012	Delhi
1003	Farm Land	300000	12/03/1985	330000	02/11/2011	Bangalore

Asset Type - Mutual Funds

Asset Id	Mutual Fund Id	Mutual Fund Name	Purchase Value (in INR)	Asset Purchase date (dd/mm/yyyy)	Current Value (in INR)	Date of last update (dd/mm/yyyy)	Investment Duration (in months)
5001	123456	ABC Corporation	10000	15/01/2010	10545.55	06/12/2011	36
5002	4564	XYZ Corporation	50000	12/06/2011	65645.71	08/02/2012	20
5003	781356	PQR Corporation	35000	21/07/2010	35745	13/05/2012	48
5004	995656	EFG Corporation	90000	29/03/2012	10545.55	16/07/2012	36

Figure4: view asset information – JavaServer Page View

References

[1] Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy Zawodny, Arjen Lentz and Derek J. Balling, *High Performance MySQL*

[2] Mark Matthews, Jim Cole, Joseph D. Gradecki, *MySQL and Java Developer's Guide*

[3] Budi Kurniawan, *Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions*

[4] Deepak Alur, John Crupi, Dan Malks, *Core J2EE Patterns: Best Practices and Design Strategies (Second Edition)*

[5] http://tomcat.apache.org/tomcat-7.0-doc/index.html

[6] http://dev.mysql.com/doc/workbench/en/index.html