

# Manifold Entrust Scheme

M. Dhanalakshmi

N. Parasuram

A. Sunil

M. Shirishakumar

Department of Computer Engineering Fr.G.P.C.E.T,Venkaypalle,Nandhikotkur,

**Abstract**—The Manifold Entrust Scheme Interface is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing banks and provides business solutions for both retail and corporate.

This system acts as a standard interface between the clients and all the banks, By using this portal any client who maintain accounts in various banks can directly log on to Multi Banking System Interface and make any kind of transactions. In the backend, system will take care of the entire obligation required in order to carry on transaction smoothly

**Keywords**— Include at least 5 keywords or phrases

## 1.Introduction

The Manifold Entrust Scheme Interface is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing banks and provides business solutions for both retail and corporate.

This system acts as a standard interface between the clients and all the banks, By using this portal any client who maintain accounts in various banks can directly log on to Multi Banking System Interface and make any kind of transactions. In the backend, system will take care of the entire obligation required in order to carry on transaction smoothly.

## SYSTEM ARCHITECTURE

### Architecture flow

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called

presentation layer, business logic layer and data link layer. This project was developed using 3-tire architecture.

### 1.Presentation Layer

Also called as client layer, comprises of components that are dedicated to presenting the data to the user. For example Windows/Web Forms and buttons, edit boxes, Text boxes, labels grids, etc.

### 2.Business Logic Layer

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process.

### 3.Data Link Layer

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

### 5.Database Layer

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using

SQL Server, Oracle, Flat files, etc. In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected

## II. Proposed System

The Manifold Entrust Scheme System Interface is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing banks and provides business solutions for both retail and corporate.

this system acts as a standard interface between the clients and all the banks, by using this portal any client who maintain accounts in various banks can directly log on to multi banking system interface and make any kind of transactions. in the backend, system will take care of the entire obligation required in order to carry on transaction smoothly

## III. Modular Design

Number of Modules

After careful analysis the system has been identified to have the following modules:

1. Admin Module
2. Customer Module
3. Bank Admin Module
4. Reports Module

### 1. Admin Module

The admin module will be used by the administrator of this portal, admin can accept or reject the requests from the bankers, and also admin can accept or reject the requests from the users. The requests are in the form of bank registration, customer registration. This module is having following functionalities.

**Pending Bankers Requests:** By using this functionality Administrator can give access permeations to all bankers who are registered in this portal.

**Pending User Requests:** By using this functionality Administrator can give access permeations to all users who are registered in this portal.

### 2. Customer Module

This module describes all about customers, by using this module any customer can do some operations like create a new account, view the account information, Transfer amount from one

account to other account and customer can also see the Transaction Reports. This module consists following functionalities.

**Create New Account:** By using this functionality user can create a new account in any bank by selecting bank name option.

**View Account Information:** By using this functionality user view all his account details, this can be viewed by users who are having account in any bank.

**Transfer Amount:** By using this functionality user can transfer money from his account to other accounts of same bank or other banks.

**Transaction Reports:** By using this functionality user can get all his transaction reports like accepted transactions, rejected transactions and pending transactions.

### 3. Bank Admin Module

This module deals with all transactions of bank management. By using this module bank staff can view all details of customers, they can go for any transactions of their customers and also they can give access permeations to all customers of that bank. This module consists following functionalities.

**List of Customers:** By using this functionality Bank admin can get their entire customers list and their details.

**List of Accounts:** By using this functionality Bank admin can get their entire customers list based on selected account type like saving account, current account etc.

**Transfer Pending:** By using this functionality Bank admin can maintain money transfer details of customers.

**Transfer Declines:** By using this functionality Bank admin maintain money transfer rejected customer details.

**New Accounts Pending:** By using this functionality Bank admin can maintain entire user details who are requesting for new account in that bank.

### 4. Reports Module

In this module administrator will get different types of reports regarding customers like Number of customers of this portal and no. of banks registered in this portal. This module is controlled by administrator only.

## Software Engineering Methodology

Object Oriented Analysis and Design (OOAD Standards)

### STUDY OF THE SYSTEM

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The 'administrative user interface' concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The 'operational or generic user interface' helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

### **INPUT & OUTPUT REPRESENTATION**

Input design is a part of overall system design. The main objective during the input design is as given below:

To produce a cost-effective method of input.

To achieve the highest possible level of accuracy.

To ensure that the input is acceptable and understood by the user.

### **INPUT STAGES**

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

### **INPUT TYPES**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

External inputs, which are prime inputs for the system.

Internal inputs, which are user communications with the system.

Operational, which are computer department's communications to the system?

Interactive, which are inputs entered during a dialogue.

### **INPUT MEDIA:**

At this stage choice has to be made about the input media.

To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device

### **OUTPUT DESIGN**

In general are:

External Outputs whose destination is outside the organization.

Internal Outputs whose destination is with in organization and they are the User's main interface with the computer. Outputs from

computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs Operational outputs whose use is purely with in the computer department.

Interface outputs, which involve the user in communicating directly with the system.

### **OUTPUT DEFINITION**

**The outputs should be defined in terms of the following points:**

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

Will decimal points need to be inserted

Should leading zeros be suppressed.

### OUTPUT MEDIA

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

The suitability for the device to the particular application.

The need for a hard copy.

The response time required.

The location of the users

The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

### PROCESS MODEL USED WITH JUSTIFICATION

#### SDLC Methodology:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

A preliminary design is created for the new system.

A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning and designing the second prototype.
4. Constructing and testing the second prototype.

At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.

The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

The final system is constructed, based on the refined prototype.

The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time

### TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?

Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

## OPERATIONAL FEASIBILITY

### OPERATIONAL FEASIBILITY

#### User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

#### Reliability

The package will pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

#### Security

The web server and database server should be protected from hacking, virus etc

#### Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

#### Availability

This software will be available always.

#### Maintainability

The system called the ewheelz uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients).

The database will be running at the server. Users access these forms by using the user-ids and the passwords.

## ECONOMIC FEASILITY

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread through out the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open source software like TOMCAT, JAVA, Mysql and Linux is used to minimize the cost for the

CustomerS.

## IV. TECHNICAL REQUIREMENTS

1. Operating System : Windows
2. Technology : Java/j2ee (JDBC, Servlets, JSP)
3. Web Technologies : Html, JavaScript, CSS
4. Web Server : Tomcat
5. Database : Oracle
6. Software's : J2SDK1.5, Tomcat 5.5, Oracle 9i

### Hardware requirements:

Hardware : Pentium based systems  
with a minimum of p4

RAM : 512MB (minimum)

### Additional Tools

HTML Designing : Dream weaver Tool  
Development Tool kit : My Eclipse

## V. FUTURE SCOPE

The Manifold Entrust Scheme Interface is targeted to the future banking solution for the users who is having multiple bank accounts in multiple banks. This interface integrates all existing banks and provides business solutions for both retail and corporate.

## VI. CONCLUSION

We finally conclude that using this project we can provide a great interface between the user and the banking environment, thus satisfying the requirements of multiple users. It provides an efficient ways for people to involve in on-line transactions. We are providing a monitoring mechanism for admin which is having the ultimate power. And finally the users will be satisfied with our service.

## VII. REFERENCES

- [1] JAVA Technologies
- [2] JAVA Complete Reference
- [3] Java Script Programming by Yehuda Shiran
- [4] Mastering JAVA Security
- [5] JAVA2 Networking by Pistoria
- [6] JAVA Security by Scott oaks
- [7] Head First EJB Sierra Bates
- [8] J2EE Professional by Shadab siddiqui
- [9] JAVA server pages by Larne Pekowsley
- [10] JAVA Server pages by Nick Todd
- [11] HTML
- [12] HTML Black Book by Holzner
- [13] JDBC
- [14 ] Java Database Programming with JDBC by Patel moss.  
Software Engineering by Roger Pressman