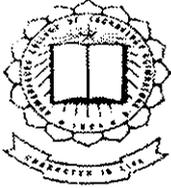


P-715

CONSUMER BANKING



PROJECT REPORT

SUBMITTED BY

R. BALAJI

GUIDED BY

Mrs. RAMATHILAGAM, MSc
Lecturer.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF SCIENCE IN

2001-2002

APPLIED SCIENCE-COMPUTER TECHNOLOGY
OF THE BHARATHIAR UNIVERSITY, COIMBATORE

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006



KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

Department of computer Science & Engineering

CERTIFICATE

This to certify that the project report entitled

CONSUMER BANKING

has been submitted by

Mr. R. BALAJI

in partial fulfillment of the requirements for the Award of degree of Master Of Science in Applied Sciences (Computer Technology) of the Bharathiar University, Coimbatore-46 during the year 2001-2002.

(Paravathipatti 15/4/02)

(Guide)

S. J. ...

(Head Of Department)

Certified that the candidate was examined by us in the project viva-voce examination held on 25/4/02 & the University Register number is CO37Q0629

R. ...
(Internal Examiner) *25/4/02*

M. ...
(External Examiner)

8th April 2002

To Whom It May Concern

This is to certify that Balaji R currently a final year student of the course M.Sc (Computer Technology) from Kumaraguru College of Technology has successfully completed his project titled "**Consumer Banking**". His project duration was from January 2002 to April 2002.

We wish him the best in all his future endeavors.

Thanking You.

For **POLARIS SOFTWARE LAB LTD**



Hema Mohandas
Asst. Manager – Human Resources.

Corporate Headquarters

POLARIS SOFTWARE
LAB LIMITED
Foundation,
34 IT Highway
Chennai - 600 103, INDIA
91 4114 35001
Fax : 91 4114 45166
chennai@polaris.com

Registered Office

POLARIS SOFTWARE
LAB LIMITED
Polaris House
244 (Old No.713), Anna Salai
Chennai - 600 006, INDIA
91 44 852 4154
Fax : 91 44 852 3280
chennai@polaris.com

Development Center

POLARIS SOFTWARE
LAB LIMITED
Polaris House
B-19, Sector 2
Noida - 201 301, INDIA
91 118 444 3456
Fax : 91 118 4551440
delhi@polaris.com

North American Headquarters

POLARIS SOFTWARE
LAB LIMITED
Woodbridge Place
517, Route 1 South Suite 2103
Iselin, NJ 08830
1 32404 1199
Fax : 1 32404 1188
np@polaris.com

Asia Pacific Headquarters

POLARIS SOFTWARE
LAB PTE LIMITED
12-01 High Street Centre
1, North Bridge Road
Singapore - 179094
95 333 1344
Fax : 95 333 1441
singapore@polaris.com

India Regional Headquarters

POLARIS SOFTWARE
LAB LIMITED
19A, Jawahar Avenue
Jawahar Building
602 003 K
91 114 65 3828134
91 114 65 3828134
kr@polaris.com

*Dedicated to My Beloved
Parents & Friends*

ACKNOWLEDGEMENT

I wish to express my sincere and heartfelt gratitude to our esteemed Dr. K.K. Padhmanabhan, B.Sc.,(Engg), Mtech., Ph.D., principal of Kumaraguru College of Technology for giving us the needed encouragement in starting this project and carrying it out successfully.

My hearty Thanks and a deep sense of gratitude to our professor Dr.S.Thangaswamy,B.E.(Hons), Ph.D., Professor and Head of Department of Computer Science and engineering for his benevolent attitude to push up this project to its position of success.

I would like to thank our guide Mrs. Ramathilagam M.Sc(Comp.Sci.), Lecture of Department of Computer Science and Engineering without whose motivation and guidance I would not have been able to embark on a project of this magnitude.

I am deeply indebted to my father Er.Mr.N.Ramachandran LTM ., ATA., F.I.E., Rtd Technical Director of NTC and Mr.Ayyappan Asst.Vice-President, Ms. Srividhya Project Manager and Mr. Shankar Project Leader for enabling me to do this project in POLARIS SOTWARE LAB LTD.

I Reciprocate the Kindness shown to us by the staff members namely Saraswathy, Rajesh, Jesudoss ,Gopal ,GuhaAnand and Varadarajan and to all the members who have supported in getting this project done.

CONTENTS

1. INTRODUCTION	2
1.1 PROJECT OVERVIEW	3
1.2 ORGANIZATION PROFILE	5
2 SYSTEM STUDY & ANALYSIS	7
2.1 EXISTING SYSTEM-LIMITATIONS	7
2.2 USER CHARACTERESTICS	9
3 SOFTWARE PROGRAMMING ENVIRONMENT	10
3.1 HARDWARE CONFIGURATION	10
3.2 SOFTWARE COFIGURATION	10
4 SOFTWARE SYSTEM DESIGN & DEVELOPMENT	11
4.1 INPUT DESIGN	12
4.2 PROCESS DESIGN	14
5 CONCLUSION & FUTURE SCOPE	39
6 BIBLIOGRAPHY	40
7 APPENDIX	41

CONSUMER BANKING

1 INTRODUCTION

This project entitled "Consumer Banking" is done on the basis of customer service by enabling them to withdraw amount anytime through ATM. The main purpose of this project is "TO ISSUE A CITICARD FOR ACCESSING ATM AND MAINTAINING THE STATUS OF A CARD".

POST CARD -ISSUANCE

Cards can be issued to the client / customer at a later stage after opening an account. Acting upon to the request of the customer, this option will enable the user to issue a card depending on the eligibility criteria of the customer. This option will enable the user to generate a new card or link the account to the existing card.

STATUS CHANGE

The status of the card decides the access and validity of the card. Status of the card can be of Active Status, Host Status and Closed Status. Only Active status allows the card to access the accounts linked to that card. Access of the card is temporarily denied for "HOT " status. Access is denied completely for "CLOSED" status of the card.

COMMON SEARCH ENGINE – USER INPUTS.

DESCRIPTION

This is a common screen for user input. Based on the user input, it will search for the given input and based on the module, it will display a set of results thru which the user is able to fulfill his request without knowing the entire details of the customer.

1.1 PROJECT OVERVIEW

This project is based on the card type as the user selects . There are three types

- a) Embossed Card
- b) Pre Embossed Card
- c) Pre Embossed Card Input

According to the card selected we have operations called **ISSUE , VIEW & DELETE.**

ATM cards provide a convenient way of accessing the accounts, round the clock 24 X 7 days, across geographical locations. With the help of ATM cards, a customer can operate any type of transaction like eCash Withdrawal, Cash Deposits, Funds Transfer from one account to another account, verify balances, Deposit booking, viewing Mutual Funds Details and many services which were offered thru Teller front office operations.

Debit Card Administration Module plays a vital role is administering and controlling the behaviour of the card. Each card is liable to certain limits and restrictions. All such limits are enabled thru this administrative Module. The components of the Debit card administrative module are

- Post Account – Card Issuance
- Account Linkage / delink

- Primary Account Change
- Facilities Modification
- Limit Maintenance
- Status Change
- Card type Facility Modification
- Replacement of card
- Pin Change
- Account Grouping Maintenance
- Product Mapping Maintenance.

1.2 ORGANISATION PROFILE:

Polaris Software Lab Limited , one of the "SIZZLERS '2000" Software companies as acknowledged by "Data Quat, a leading IT magazine, registered a growth rate of 180 percent over the period of last three year (CAGR), with the last years growth rate of over 140% the previous year. It was instituted in 1983 to provide quality information technology solution to clients on a global basis. It is one among the few Indian companies which has both **ISO - 9001** and **SEI level - 4** certification.

Today the company is globally competitive with operations in USA, Middle East, Singapore, Asia Pacific and Australia.

The core competence area of the company is the banking and enterprise related solutions while relationship management is also a strong capability of the company. The company has come out with a concept called ENTITY - ExteNded Technology faciliTY to provide software solution to the client. Existing is the relationship model which works as a virtual extension of the customer's office. Polaris Work Center each client on long - term relationship basis rather than fast a one-off client. In 1986, PSL entered to such a relationship with CITIBANK India, for their "Real Banking Automaton" ever the years PSL has provided customer delight and contributed to the operational growth of the client Citi Bank, India, through successful delivery of software development and project services in build up of their credit card system, Loan installment system, time deposits system, posnetwork, global finance system, security management system and branch banking.

The organisation works in a Total Quality Management (TQM) environment and quality assurance standards by conducting quality orientation programmes for the employee, providing comprehensive and practical solution with the highest level of quality of the customers.

PRODUCTS

Polaris has also developed the following products:-

INSPIRE

A software process improvement tool. It is an acronym to integrated software process implementation and regulation. It is a quality focused process management tool that lays emphasis on proactive planning, synergized team work and Organisational learning.

SUPERSTORE 2000

It is the most comprehensive user - friendly WINDOWS based retail solutions with bar coding facility. This integrated sales, inventory, purchase and financial accounting system has merged retail management with information technology.

ENTERPRISE

Enterprise consists of an existing base solution, along with customized add-ons as per the customers exact requirements. NTenterprise has been designed and developed by considering an application as a collection of business objects. Stored procedures and triggers have been written in re-usable fashion and are thus object oriented in concept. This concept of optimized re-usage of code has been branded as RUSA technology (Reusable Software Assets).

PARSLEY

Parsley, the customer service helps desk companies to increase their service, quality, reliability, efficiency and ultimately their probability. This concept first of its kind from an Indian Software company, has scaled new heights with WEB hotline, the internet based complete service check solutions.

*SYSTEM STUDY
&
ANALYSIS*

2 SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM - LIMITATIONS

DESIGN ALTERNATIVE

The basic design was to use Java Server Pages / HTML to publish the front-end screens. Enterprise Java Beans (EJB) basically a service provider will act as a mediator between the Front End User interactive screens and the actual storage of data. Weblogic is used as a server to entertain these tasks. Currently ORACLE RDBMS v 8.0 is used to store and retrieve the valid information in the form of Tables.

As the product is Multilingual, all multilingual text and labels are stored in flat file for easy access and adaptability. Database access queries and manipulation statements are kept in a property file which will be in weblogic startup class for ready usage.

RECOMMENDED ALTERNATIVES

DATABASE DESIGN ISSUES

The process of back-up of the database is done continuously and the files are backed up. The archiving process of Oracle is set and continuously the operations that are performed on the database are taken back up. If there is a database crash, then the database is recreated based on the back and applying the archive files on the back up.

PROCESS DESIGN ALTERNATIVES

The process of design alternative would be that existing system which has all functionalities without Multilingual capability is there already implemented and successfully functioning in Luxembourg, Monaco, Czech Republic and London. In case of any problems with the new system, then the old system can be used as the contingency measure.

The other alternative would be that the user could be given a spread sheet, which will contain the base and accounts with necessary details that would furnish the customer's needs. This will help user attend to customer's queries. The necessary details would be given based on the requirement of the user on a periodical basis

LIMITATIONS:

THE PRESENT MODULE IS DONE USING LAYERS WHICH WILL BE SUPPORTED BY NETSCAPE ONLY. SO THE INTERNET EXPLORER WON'T BE SUITABLE TO THIS. IF WE WANT TO RUN THIS MODULE THERE SHOULD BE A WEBLOGIC SERVER IN THE MACHINE.

DESIGN ISSUES

FUNCTIONAL DESIGN ISSUES

There were functional issues with the user requirement and the following assumptions were made:

- For tracking Limits, it is assumed that the limits for different set of periods are maintained one for a period .
- Limits are converted in the HOME CURRENCY according to the conversion rate currently available.
- Conversion of the limit to the HOME CURRENCY will be rounded off to the nearest integer.

- The existing version of Embossing programs will be utilized for embossing the cards. Facilities will be provided to opt online / Batch embossing of cards by modifying a maintenance. The existing embossing program will cater to the needs of one machine "DC 9000 ".

2.2 USER CHARACTERISTICS:

The form or any part of this module is user friendly and customer can enter any form only if he is having an account in Citibank and if he knows Account Number or CIF Number or Base Number. Others can just view only the important pages.

GENERAL CONSTRAINTS :

The General Constraints are Base Number , CIF number, CIF Name and Encrypted Pin. The pin is the one which gives the higher security .

SOFTWARE PROGRAMMING ENVIRONMENT



3 SOFTWARE PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION :

PLATFORM : UNIX OPERATING SYSTEM
VERSION : UNIX 5.8 RELEASE
SERVER : SUN SOLARIS
HARD DISK : 100 GB
MEMORY : 5 GB
APPLICATION SERVER : WEBLOGIC 5.1

3.2 SOFTWARE CONFIGURATION :

WEB TOOLS : HTML , JSP

SCRIPT : JAVA SCRIPT

HTML : For Designing a Web Page.

**JSP : For Retriving values and modifying
and inserting values into BackEnd**

JAVASCRIPT : For validation.

BACK END : ORACLE

HTML

What is HTML?

HTML stands for the HyperText Markup Language. HTML is the major language of the Internet's World Wide Web. Web sites and web pages are written in HTML. With HTML and the world wide web, you have the ability to bring together text, pictures, sounds, and links... all in one place! HTML files are plain text files, so they can be composed and edited on any type of computer... Windows, Mac, UNIX, whatever.

This tutorial is for anyone who is serious about learning HTML. Perhaps you want your own web page, or an entire web site. Perhaps you're setting up a web site for your business or organization. You've come to the right place. You do not need any prior experience in making web pages to take this course (although I will assume you know how to do some basic things with your computer, like use a word processor.)

This tutorial is a "hands on" tutorial... you get to test your new HTML skills immediately after you learn them, chapter by chapter ... all within your web browser! There is no need to go fiddle with other applications while you are learning HTML with this tutorial

Rules and Regulations of Use

These rules and regulations cover all pages contained within the following site(s):

Accessing the Tutorial and Linking the Tutorial

The tutorial may be accessed free of charge from the davesite.com server, and may be linked to the davesite.com location from anywhere on the internet for free. Screen shots may be taken for up to two (2) pages, but may only be used in media, such as article write-ups.

B. Information in the tutorial is provided as-is, the author will not be held responsible for any event caused by use of the tutorial, such as, but not exclusively, loss of profit or personal injury.

. One (1) copy of any or all of the pages of the tutorial may be printed into hardcopy form FREE for personal use only by the person that accessed the site. This hardcopy may not be reproduced electronically or by any other means, altered or unaltered. Doing so acknowledges that you have purchased a license for the number of copies you have made, at the following price in United States Dollars (via check or money order

JAVA SERVER PAGES

Code Factoring and Role Separation

One of the main reasons why the Java Server Pages technology has evolved into what it is today (and it's still evolving) is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. The foundation for JSP was laid down with the initial development of the Java Web Server from Sun, which utilized page compilation and focused on embedding HTML inside Java code. As applications came to be based more on business objects and n-tier architectures, the focus changed to separating HTML from Java code, while still maintaining the integrity and flexibility the technology provided.

Java beans and objects can be bound to different contexts just by defining a certain scope. Good application design builds on this idea and tries to separate the objects, the presentation and the manipulation of the objects into distinct, distinguishable layers.

Another benefit of utilizing JSP is that it allows us to more cleanly separate the roles of a web production/HTML designer individual from a software developer. Remember that a common development scenario with Servlets was to embed the HTML presentation markup within the Java code of the Servlet itself, which can be troublesome. In our discussion, we will consider the Servlet solely as a container for Java code, while our entire HTML presentation template is encapsulated within a JSP source page. The question then arises as to how much Java code should remain embedded within our JSP source pages, and if it is taken out of the JSP source page, where should it reside?

Let's investigate this further. On any web-based project, multiple roles and responsibilities will exist. For example, an individual who designs HTML pages fulfills a web production role while someone who writes software in the Java programming language fulfills a software development role.

On small-scale projects these roles might be filled by the same individual, or two individuals working closely together. On a larger project, they will likely be filled by multiple individuals, who might not have overlapping skill sets, and are less productive if made too dependent on the workflow of the other.

If code that could be factored out to a mediating Servlet is included instead within HTML markup, then the potential exists for individuals in the software development role and those in the web production role to become more dependent than necessary on the progress and workflow of the other. Such dependencies may create a more error-prone environment, where inadvertent changes to code by other team members become more common.

This gives us some insight into one reason why we continue to develop basic Servlets: they are an appropriate container for our common Java code that has been factored out of our JSP pages, giving our software development team an area of focus that is as loosely coupled to our JSP pages as possible. Certainly, there will be a need for these same individuals to work with the JSP source pages, but the dependency is reduced, and these pages become the focus of the web-production team instead. Of course, if the same individual fulfills both roles, as is typical on a smaller project, such dependencies are not a major concern.

So, we should try to minimize the Java code that we include within our JSP page, in order to uphold this cleaner separation of developer roles. As we have discussed, some of this Java code is appropriately factored to a mediating Servlet.

become more sophisticated, as well. Thus, our example will evolve and we will gain an understanding of how the various architectures that we discuss will help us build a system that satisfies these requirements in an elegant and effective manner.

Let's look at some examples of architectures that utilize these approaches and discuss the tradeoffs and usage.

The "Page-Centric" Approach

We start by looking at architectures based on the page-centric or client-server approach. We will look at the page-view and page-view with bean architectures. Applications built using a client-server approach have been around for some time; they consist of one or more application programs running on client machines, and connecting to a server-based application to work. (A good example would be a PowerBuilder or Oracle Forms based system.) CGIs and pre-Servlet applications were generally based on this simple 2-tier model, and with the introduction of Servlets, 2-tier applications could also be created in Java.

This model allows JSPs or Servlets direct access to some resource like a database or legacy application to service a client's request: the early JSP specifications termed this a "Model 1" programming approach. The JSP page is where the incoming request is intercepted processed, and the response sent back to the client; JSPs only differed from Servlets in this scenario by providing cleaner code, and separating code from the content by placing data access in beans

Code that is common to multiple requests, such as authentication, is a good candidate for a mediating Servlet. Such code is included in one place, the Servlet, instead of potentially being cut and pasted into multiple JSPs.

We will also want to remove much of our business logic and data access code from our JSP page and encapsulate it within JavaBeans, called worker or helper beans. We start to see a pattern of code movement from our JSP into two areas: a Servlet (or JSP) that sits in front of the main JSP, and JavaBeans that sit in back. We refer to this common pattern as 'Factor Forward-Factor Back', as shown in the figure below:

Redirecting and Forwarding

Redirecting and forwarding requests in JSPs and Servlets takes place often, and it is important to understand the subtle difference between these two mechanisms even though they achieve the same goal (that is, a client asks for a resource on the server and a different resource is served to it):

When a Servlet or JSP resource chooses to redirect the client (using a `response.sendRedirect(url)`) the request object does not reach the second resource directly since the underlying implementation is an HTTP redirect. The server sends an HTTP 302 message back to the client telling it that the resource has moved to another URL, and that the client should access it there.

The bottom line is that the lifecycle of the initial request object that was accessed in the first JSP terminates with the end of the service method in the first JSP, or with the reply from the server.

In a forward mechanism the request object is forwarded to the second resource, thus maintaining any object bindings to the request and its state, without a round trip to the client on the network.

This allows the first JSP to do some work internally and then send information to the second JSP asking it to do its bit. (Servlets used a chaining mechanism to do this). See Chapter 5 to get a clearer picture of scope.

JSPs and Servlets can utilize the forwarding mechanism to delegate tasks amongst themselves, in the process of separating dynamic and static content.

Now, let's investigate how we build these systems.

Architectures

Before discussing specific architectures that we can use to build systems with Servlets and JSP, it is worth mentioning two basic ways of using the JSP technology. Each of the architectures discussed in this chapter will be based on one of these approaches:

The first method is referred to here as the page-centric (or Client-Server) approach. This approach involves request invocations being made directly to JSP page.

In the second method, the dispatcher (or N-tier) approach, a basic Servlet or JSP acts as a mediator or controller, delegating requests to JSP pages and JavaBeans.

We will examine these approaches in light of a simple example, which will evolve to satisfy the requirements of various scenarios. The initial scenario involves providing a web interface for guessing statistics about a soon-to-be-born baby. The guesses are stored, and can be reviewed later by the parents, to see who has guessed the closest. As the requirement scenarios become more sophisticated, such as adding the desire for a persistence mechanism, the solution scenarios will

4 SOFTWARE SYSTEM DESIGN AND DEVELOPMENT

CARD ISSUANCE SCREEN – MAKER LEVEL

DESCRIPTION

The Card Issuance screen will have three options such as Issue, delete and View.

On clicking the Issue button,

DATA ELEMENTS

ISSUE

- Acts as a sub menu of Card Issuance.
- The Issue button is for initiating a new request for Issuance of a card for a customer.
- The value of the button is "Issue".
- The name of the element would be *****
- If the button is pressed, it will invoke the search engine as to which customer the card to be issued.

DELETE

- Acts as a sub menu of Card Issuance.
- The Delete button is for cancelling a card issuance request already made at Maker level.
- The value of the button is "Delete".
- The name of the element would be *****
- If the button is pressed, it will cancel the card issue request.

VIEW

- Acts as a sub menu of Card Issuance.
- The View button is to view the details of the card issue requested at Maker level.
- The value of the button is "View".
- The name of the element would be *****
- If the button is pressed, it will just display the details of the customer and the card type issued, card number, expiry dates of the card, facilities of the card, Limits offered at various level etc.

4.1 INPUT DESIGN :

LIST OF INPUTS

DATA ELEMENTS BASE NUMBER

- User can input Base number and related details will be displayed.
- The base number is a text field.
- The name of the element will be *****
- The element length would be 6 digits.

CIF-NUMBER

- The Customer Identification number can be inputted and related customer information details will be displayed.
- The CIF number is a text field.
- The name of the element is *****
- The element length would be 10 digit

CUSTOMER NAME

- The user can input customer name and get the details.
- The Customer Name is a text field. It contains of First name, Middle Name and Last Name.
- The name of the element is *****
- The element length would be 35digit for a part of name.

OTHERS

- If the user doesn't know any of the above customer related information, he can take the help of the "Others" option, where in he can give Card number, Account Number, Calling Code etc.

ACCOUNT -NUMBER

- The user can input account number and get the details of the account.
- The Account number is a text field.
- The name of the element is *****
- The element length would be 10 digit

CARD-NUMBER

- The user can input card number and get the details of the customer.
- The card number is a text field.
- The name of the element is *****
- The element length would be 16 digit

CALLING-CODE

- The user can input the calling code of the customer to get the details of the customer.
- The calling code input is a text field.
- The name of the element is *****
- The element length would be 10 digit

4.2 PROCESS DESIGN

CARD ISSUANCE -ISSUE

DESCRIPTION

The Card Issuance screen is to issue a new ATM card for a customer. No of cards that can be issued is restricted based on the parameter " NOOFCARDSPERCIF or NOOFCARDSPERBASE". Prior check is validated before issuing a card to a customer, whether the customer is eligible to issue a card for that account or a base.

The card number can be generated and embossed is based on the Implementation parameter. The system has the functionality of accepting the card number and the PIN as User Input. It

can also facilitate to pick from a list of Card pool, which is stored in a storage area. Based on the parameter maintained, the card issuance user interface screen will dynamically framed.

DATA ELEMENTS

Cif Number

The cif number denotes the customer identification number to which the card is issued.

The name of the element will be *****

The length of the element would be 10 digits.

This is a mandatory field, which will be passed by the search engine.

The customer identification number is display field and the user is not allowed to change the value.

Card Number

The card number is the number assigned with the ATM card and it is for identifying the customer's card uniquely.

The name of the element will be 

The length of the element would be 16 digits.

The field is mandatory.

Based on the parameter maintenance, the system will request the user to give inputs.

If the Emboss Flag required parameter maintained as "Y", as mentioned in the design document Vol 1, the card number will be generated with a predefined logic provided by the bank. If it is maintained as "A", it will pick from a pool of cards, which is already pre-embossed and stored in a separate storage area for ready usage. If the parameter is maintained as "N", then the card number is a user-inputted field.

CARD STATUS

The card status denotes the current status of the customer card.

The name of the element will be *****

The length of the element would be 10 digits.

This is a mandatory field, which displays the status of the card such as ACTIVE, HOT, CLOSED.

The card status field is a display field and the user is not allowed to change the value. The current status of the card is displayed as default value.

Primary Account Number:

The Primary Account Number would denote the primary account for the card. If a card provided an International Access, then only the primary account can be accessed by the card.

The element length would be 15 digits.

The name of the element will be *****

The element is of type List Box.

Account related to that Customer would be populated in the list box.

This is a mandatory field.

CIF-NAME

This CIF Name would denote the customer name.

The element length would be 35 characters.

The name of the element would be *****

The element is of type display text.

The field is mandatory, displayed automatically with the value from search engine.

CARD TYPE

The card type would denote the type of the plastic issued.

The element length would be dynamic depending upon the maintenance.

The name of the element would be *****

The element is of type list box.

This field is mandatory.

Different types of plastic issued to the customer will be populated in the list box, which is picked from common parameter maintenance.

ENCRYPTED -PIN

The encrypted Pin field denotes the PIN number of the customer.

The element length would be 16 digits.

The name of the element would be *****

The element is of type user input / display. It depends on the common parameter maintenance.

This field is mandatory.

ISSUE-DATE

The Issue date field denotes the date of the card issued to the customer.

The element length would be 10

The name of the element would be *****

The element is of type text box.

The field is mandatory.

The date format is as maintained in the common parameter maintenance table.

EXPIRY-TIME

The Expiry date field denotes the expiry date of the card, until which the card is valid.

The element length would be 10

The name of the element would be *****

The element is of type text box.

The field is mandatory.

The date format is as maintained in the common parameter maintenance table.

CARD FACILITES

Every card issued is associated with a set of facilities offered by the bank. Each facility will identify a type of transaction. Certain facilities will be allowed for a type of card. In addition

to that, a facility can be enabled or disabled for an individual card, by making the value of the field as "1".

DIRECT-DEBIT

Direct debit Flag denotes the ATM to allow direct debits for a card.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Direct debits transaction will be enabled when it is checked and it is disabled when it is unchecked.

RESIDENT-STATUS

Resident Status flag is to denote the customer's residential status whether Resident or Non-Resident.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the card holder is resident when the checkbox is enabled and indicates non-resident when the check box is disabled.

EXTERNAL TRANSFER

External Transfer is a type of facility offered to a card, which allows transfer of funds to a different customer base.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the external funds transfer is allowed when the checkbox is enabled and not allowed when the check box is disabled.

HOME BANKING

Home Banking is a type of facility offered to a card, which allows access to Home banking.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the home banking access is allowed, when the checkbox is enabled and is not allowed when the check box is disabled.

SELL_MF

Sell MF is a type of facility offered to a card, which can be utilized to sell Mutual funds from the ATM itself.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the access is allowed to sell mutual funds, when the checkbox is enabled and is not allowed when the check box is disabled.

BUY MF

Buy MF is a type of facility offered to a card, which can be utilized to buy Mutual funds from the ATM itself.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the access is allowed to buy mutual funds, when the checkbox is enabled and is not allowed when the check box is disabled.

DEPOSITS

Deposits is a type of facility offered to a card which can be utilized to book time deposits from the ATM itself.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the access is allowed to book deposits, when the checkbox is enabled and is not allowed when the check box is disabled.

STANDING ORDER

Standing Order is a type of facility offered to a card thru which a customer can initiate a standing instruction from the ATM machine itself.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates the access is allowed to initiate standing instruction, when the checkbox is enabled and is not allowed when the check box is disabled.

SEE INFO

See Info is a type of facility offered to a card through which a customer can see the account details and verify the balances.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates that the access is allowed to print the balances when the checkbox is enabled and is not allowed when the check box is disabled.

USE REFERENCE CURRENCY

Use Reference Currency is a type of facility offered to a card through which a customer can see the .

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates value 1 when checked.

0 - No reference Currency Specified.

1 - Reference Currency Specified.

REF Ccy EURO

Ref Ccy EURO is a type of facility offered to a card through which a customer can check all his balance and account details which are in IN currencies can be verified in EURO currency and EURO currency balance information can be viewed in IN currency.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

This facility is allowed when the check box is enabled.

Indicates value 1 when checked.

0 - Converts Euro Balance into NCD.

1 - Converts NCD balance into EURO.

RELATED ACCOUNT INDICATOR

Relational Account Indicator is a facility to denote a particular account as a Primary Account Revolving Credit Account etc.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

This facility is allowed when the check box is enabled and not allowed when disabled.

Use Reference Currency

Use Reference Currency is a type of facility offered to a card through which a customer can see the

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates value 1 when checked.

0 - No reference Currency Specified.

1 - Reference Currency Specified.

INTERCITI

Interciti is a type of facility offered to a card through which a customer can transfer funds from an account to another account of other branch of the citibank.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates value 1 when checked.

0 - Interciti transfers not allowed.

1 - Interciti transfers allowed.

Limit Profile Identifier

Limit Profile Identifier is a type of facility, which denotes the availability of the Limit Profile of a card.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates value 1 when checked.

0 - Limit Profile Disabled.

1 - Limit Profile Enabled.

CATEGORY TOTALS

Category Totals is a type of facility, which denotes the eligibility of printing the Statement of Accounts through ATM.

The name of the element would be *****

The element of type Check Box

The length of the element is 1.

The default value of the element is as maintained for that card type.

Indicates value 1 when checked.

0 - Category Totals

1 - Limit Profile Enabled.

CASH AMOUNT

Cash Amount, which denotes the maximum amount that a customer can withdraw through his card.

The name of the element would be *****

The element of type Text Box

The length of the element is 25.

The default value of the element is as maintained for that card type.

CASH CURRENCY

Cash Currency denotes in which currency the maximum withdrawal amount is specified for a customer.

The name of the element would be *****

The element of type List Box

The length of the element is 3.

The default value of the element is as maintained for that card type.

List of Values is picked from the currency master.

TRANSCATION COUNT

Transaction Count denotes the maximum no of transaction that a customer can use the card to withdraw the amount.

The name of the element would be *****

The element of type Text Box

The length of the element is 2.

The default value of the element is as maintained for that card type

TRANSCATION PEROID

Transaction Period denotes the period of the transaction count specified.

The name of the element would be *****

The element of type List Box

The length of the element is 25.

The default value of the element is as maintained for that card type.

List of values is picked form the common parameter maintenance table.

PURCHASE AMOUNT

Purchase Amount, which denotes the maximum amount that a customer can purchase goods through his card.

The name of the element would be *****

The element of type Text Box

The length of the element is 25.

The default value of the element is as maintained for that card type.

Purchase Currency

Purchase Currency denotes in which currency the maximum purchase amount is specified for a customer.

The name of the element would be *****

The element of type List Box

The length of the element is 3.

The default value of the element is as maintained for that card type.

List of Values is picked from the currency master.

Transaction Count

Transaction Count denotes the maximum no of transaction that a customer can use the card to make purchase of goods.

The name of the element would be *****

The element of type Text Box

The length of the element is 2.

The default value of the element is as maintained for that card type.

Transaction Period

Transaction Period denotes the period of the transaction count specified.

The name of the element would be *****

The element of type List Box

The length of the element is 25.

The default value of the element is as maintained for that card type.

List of values is picked form the common parameter maintenance table.

SUBMIT/CANCEL

The submit & cancel buttons are for putting forward the request for issuance of a card or to cancel the request initiated.

The values of the button are submit and cancel.

If the submit button is pressed, then the card issuance request is initiated and the request is locked in the database.

The cancel button will cancel the card issuance request

CARD ISSUANCE – MAKEL LEVEL DELETE

DESCRIPTION

The user can revoke or cancel the maker level request for card issuance for a customer. The system will automatically delete the maker level request from the temporary storage area. Only the user who has initiated the request has the authorisation to revoke or cancel the request.

DATA ELEMENTS

Card Number

Card Number is a Hyper Link display field, fro which the user has initiated a request for card issuance at Maker Level.

On clicking the Hyperlink, a pop up window will display the details of the card parameters.

The name of the element would be *****

The Length of the element is 20

The value of the card number is picked from the temporary storage area and displayed as default value.

CIF Number

Customer Identification Number is a display field, denotes to which customer the card issuance request has been initiated.

The name of the element would be *****

The Length of the element is 10

The element is of type Display Box.

The value of the customer deification number is picked from the temporary storage area and displayed as default value.

Primary Account Number

Primary Account Number is a Hyperlink display field, denotes the primary account for the card, to which a card issuance request has been initiated.

The name of the element would be *****

The Length of the element is 14

The element is of type Display Box.

The value of the Account number is picked from the temporary storage area and displayed as default value.

On Clicking the Hyper Link, a small popup window will display the details of the account.

Card Type

Card Type denotes the type of the card selected for issuance.

The name of the element would be *****

The Length of the element is 20

The element is of type Display Box.

The value of the Card Type is picked from the temporary storage area and displayed as default value.

Delete All

The Delete All button will cancel the card issuance request and delete from the temporary storage area.

The value of the button is Delete All.

If the Delete All button is pressed, then all the card issuance request initiated at the maker level will be permanently canceled and the records in the temporary storage will be deleted.

Delete Selected

The Delete Selected button will cancel the card issuance request, which is marked selected by using the check box, and delete those selected card numbers from the temporary storage area.

The value of the button is Delete Selected.

If the Delete Selected button is pressed, then the card issuance request initiated at the maker level, for those selected will be permanently canceled and the records in the temporary storage will be deleted.

Home

The Home button are for cancelling the request initiated and transferring the control to the main menu.

The value of the button is HOME.

If the HOME button is pressed, then the card issuance – delete option request is cancelled.

CARD ISSUANCE – MAKER LEVEL VIEW

DESCRIPTION

The user can view all the un authorised pending card issuance request, or verification of values. Only the maker, who has initiated the request, can view the details of the card issuance request.

DATA ELEMENTS

All Data elements displayed in Card Issuance Add option will be displayed in this screen.

All fields are of type Display field. No value is allowed to be modified in this option. User can just verify values he inputted for a card.

ISSUANCE - AUTHORISATION

DESCRIPTION

Card Issuance Authorization is used to authorize the card issuance change request initiated at the maker level. Any card issued should accompany with facilities offered to that particular card with limit specification. The authorizer will have the right to authorize or reject the request initiated by the maker. In case the authorizer, wishes to modify any of the values, the user should reject and ask the maker to initiate a fresh request with the required values / reason text.

STATUS CHANGE – Add

Description

This option allows the user to change the card status of a customer according to his wish. There are three types of status allowed. They are 1. ACTIVE 2 .HOT 3 .CLOSED. The card can be useful to anyone only if the card status is in ACTIVE State. If the status is HOT then the card will be temporarily terminated of its usage. It is CLOSED then the card cannot be used further in the future. Initially the card status will be ACTIVE to the customer. The user is allowed to change the card status accordingly depending on the customer's request. The Display will show some of the card details like card number, card type, expiry date, existing status. It will also display all the account numbers along with its account type, currency and balance which are linked to the card.

Data Elements

Card Number

- The card number is the number assigned with the ATM card and it is for identifying the customer's card uniquely. This field displays the existing card number for the customer.
- The length of the element would be 16 digits.
- This is a display field.

Card Type

- The card type depicts the type of the Card which the customer have. Here the card type will be the existing card numbers card type.
- This is a display field.

Expiry Date

- The expiry date tells the date on which the card expires.
- This is a display field.

Existing Status

- The Existing Status shows the current status of the card displayed.
- This is a display field.

New Status

- This data element displays the other available status of the card. This is a parameter maintenance.
- This is an input field of selection type.
- Each and every card status is internally mapped with a unique status code which is of two digit length.
- This data element is mandatory.
- The name of the element would be *****.

Status Reason

- This data element displays the available status reason for the new status selected. This is parameter maintenance and their values depend on the card status maintenance.
- This is an input field of selection type.
- Each and every status reason is internally mapped with a unique reason code which is of two digit length.
- This data element is mandatory.
- The name of the element would be *****.

Account Number

- An account number is a unique number given to the customer to access his products which he is having with the bank.
- The length of the data element is 10.
- This is a display field.

Account Type

- An account type is the type of the account which the customer is having with the bank.

- This display is internally mapped with the product code which is a 3 digit code. e.g., 001 code denotes that it is a current account.
- This is a display field.

Currency

- The currency is the account currency of the customer.
- The swift code of the currency is displayed in this field.
- The length of the data element is 3.
- This is a display field.

Balance

- The balance shows the current withdrawal balance allowed from its respective account.
- This is a display field.

Submit

- This element is of type button. Its process is to initiate all the required transfers to the back end for permanent storage of the changes caused by the program.
- This element on clicking validates all the mandatory values in the form.
- The name of the element would be *****

Cancel

- This element is of type button. Its process is to initiate the cancellation of the request put by the user.
- This element on clicking will takes you back to the previous menu.
- The name of the element would be *****

Status Change – Modify Option

- The modify option is for modifying the maker level entries that has been posted.
- The modify option can be done only by the same maker who has initiated the maker level modification.
- The modify option will allow to change the Status of the Card.
- The same set of field level validations are applicable for the modify option as that of the add option.
- The existing values are displayed from the master table and the new values are taken from the temporary tables where the maker entry was put and the user will be allow to modify the same.
- On submitting the values, the same mandatory checks will be done on the modified entries.

Status Change – View Option

- The view option is for display of the maker level modified entries.
- Any person who is enabled for this application can do the view option.

- The view option will not allow any modification to the fields and all the fields are display only.
- The same set of fields will be displayed in view option as that of add or modify option.
- The changed primary account number will be taken from the temporary table where the maker entry was put.
- The existing values will display the existing values fetched from the master table.

Status Account Change – Delete Option

- The delete option is for deleting the maker level modification entry that the maker has initiated.
- Only the same person who has initiated the maker level entry can do the delete option.
- While displaying the details of the maker level entries that the user has done, the user will not be allowed to change any of the fields as they will all be display only fields.
- On submit, the maker level entry is deleted from the temporary tables.
- There is a cancel option, which on clicking will not delete the maker entry and the entry will be left unaffected.

Status Change – Authorisation

Description

Status Change Authorisation is used to authorise the card status change request initiated at the maker level. Any status change should have a valid reason. The authoriser will have the right to authorise or reject the request initiated by the maker. In case the authoriser , wishes to modify any of the values , the user should reject and ask the maker to initiate a fresh request with the required values / reason text.

CONSUMER BANKING

THE TABLE STRUCTURE AS FOLLOWS

TABLE NAME :- CARD DETAILS

CIF NO	VARCHAR2(16)
CIF NAME	VARCHAR2(16)
AC NO	VARCHAR2(10)
CARD TYPE	VARCHAR2(10)
CARD NAME	VARCHAR2(15)
CARD NO	VARCHAR2(16)
ENCRYPTED PIN	VARCHAR2(16)
ISSUE DATE	VARCHAR2(12)
EXPIRY DATE	VARCHAR2(12)

TABLE NAME :- ACCOUNT DETAILS

AC NO	VARCHAR2(10)
BASE NO	VARCHAR2(10)
CIF NO	VARCHAR2(16)
CIF NAME	VARCHAR2(16)
ACC CURRENCY	VARCHAR2(15)
REF CURRENCY	VARCHAR2(12)
LEGAL STATUS	VARCHAR2(12)
SIGNATURE RULE	VARCHAR2(9)

TABLE NAME :- EMBOSS CARD DETAILS

SNO	VARCHAR2(5)
CARD NAME	VARCHAR2(9)
CARD NO	VARCHAR2(16)
STATUS	VARCHAR2(10)

5 CONCLUSION AND FUTURE SCOPE

Thus the Project namely “Consumer Banking” mainly concentrates on the Customer Requirements and mainly on the security by enabling the Status to HOT if there is a problem to the customer card. They will temporarily withdraw the card.

Future Scope:

There is a tremendous scope for this project as each and every product in this world can be brought with this Citicard and the amount can be withdrawn anytime. The status of the card can be changed according to the customer and primary account can also be changed.

Future Enhancement:

The Citicard can be used as Credit card, which is presently used for Debit card and ATM. Additional Facilities can be included for Card Type namely CITIGOLD and CITIBLUE. Additional Card Type namely CITISILVER can be included.

----- X X -----

6 BIBLIOGRAPHY

The following books and websites were referred for developing the project

Books:

1. "Java Server Programming", J2EE Edition, Wrox Publication
2. "Java Server Pages" – whitehead

SOURCE CODE

CARD ISSUANCE- ADDING A DETAILS FOR ISSUANCE OF A CARD

```
<html>
<head>
<title> hi </title>
<script src = "Date.js">
</script>
<script>
var m1 = 'a1';
var m2 = 'b1';
var m3 = 'c1';
function s1()
{
  if(isNaN(document.frm1.t1.value))
  {
    alert('Please Type a Number');
    document.frm1.t1.focus();
  }
  if(document.frm1.t1.value.length<10)

  {
    alert('please type a 10 digit number');
    document.frm1.t1.focus()}
function s2()
{

  if(isNaN(document.frm1.t2.value))
  {

  }
  else

  {
    alert('Please Type a Character');
    document.frm1.t2.value="";
    document.frm1.t2.focus();
  } }

}
function s7()
```

SOURCE CODE

CARD ISSUANCE- ADDING A DETAILS FOR ISSUANCE OF A CARD

```
<html>
<head>
<title> hi </title>
<script src = "Date.js">
</script>
<script>
var m1 = 'a1';
var m2 = 'b1';
var m3 = 'c1';
function s1()
{
  if(isNaN(document.frm1.t1.value))
  {
    alert('Please Type a Number');
    document.frm1.t1.focus();
  }
  if(document.frm1.t1.value.length<10)
  {
    alert('please type a 10 digit number');
    document.frm1.t1.focus()
  }
function s2()
{
  if(isNaN(document.frm1.t2.value))
  {

  }
  else
  {
    alert('Please Type a Character');
    document.frm1.t2.value="";
    document.frm1.t2.focus();
  }
}
function s7()
```

```

document.frm1.sel1.focus();
}

else if(m2=='b1') {
alert('select any CardType');
document.frm1.sel.focus();
}

else if(m3=='c1')
{
alert('select any CardName');
document.frm1.sel10.focus();
}
else if(document.frm1.t6.value=="")
{
alert('please type pin of length 16');
document.frm1.t6.focus();
}
else if(document.frm1.t7.value=="")
{
ialert('select date by clicking button');
document.frm1.b1.focus();
}
else if(document.frm1.t8.value=="")
{
alert('select date by clicking button');

```

Command

```

{
function sub3()
{
var k=document.frm1.sel10[document.frm1.sel10.selectedIndex].value;

if(k=='PEI-CARD')
{
document.layers[0].visibility='show';
}
else
{
document.layers[0].visibility='hide';
}
}
function sub4()
{
m1=document.frm1.sel1[document.frm1.sel1.selectedIndex].value;

m2=document.frm1.sel[document.frm1.sel.selectedIndex].value;

```

```

m3=document.frm1.sel10[document.frm1.sel10.selectedIndex].value;

document.frm1.t1.value="";
document.frm1.t2.value="";
document.frm1.t6.value="";
document.frm1.t7.value="";
document.frm1.t8.value="";
document.lay0.document.fom.no.value="";
m1=='a1';
m2=='b1';
m3=='c1';
}
</script>
</head>
<body bgcolor = "#caeafa" text='blue'>
<form name='frm1' method='post' action='ten.jsp' >
<input type='hidden' name=no value="">
<center></center></img>
<a href="atm2.html">Go Back </a>
<table bgcolor="blue" width="100%">
<tr>
<td><h5><b><font color='white'> CITICARD ISSUANCE-
ADMINISTRATION</font></b></td>
</tr>
</table>

<table width="100%" >
<tr>
<td><h5>CIF Number</h5></td>
<td><input type="text" name='t1' size='10' maxlength='10'onChange='s1();' ></td>

<td><h5>CIF Name</h5></td>
<td><input type="text" name='t2' size='10' maxlength='16'onChange='s2();' ></td>
</tr>
<tr>
<td><h5>Primary AccountNo</h5></td>
<td><select name='sel1' >
<option value='a1'> </option>
<option name='a2'>A00001</option>
<option name='a3'>A00002</option>
<option name='a4'>A00003</option>
<option name='a4'>A00003</option>
<option name='a5'>A00004</option>
<option name='a6'>A00005</option>
<option name='a7'>A00006</option>

```


CITICARD PROCESSING DETAILS

```
<html>
<head>
<title> ATM </TITLE>
</head>
<body bgcolor ="#caeafa" text = "blue" >
<form>
<center><img src='mayank.jpg'width='90%'></center>
<center><img src='balaji.jpg'width='90%'></center>

<a href="#bottom" name='top'>Go Bottom</a>
<h4><u><I> CITICARD ISSUANCE :</I></u></h4>
<H4><p><FONT FACE='PALATINO LINOTYPE' COLOR='blue'>
Cards can be issued to the client / customer at a later stage after opening an aan account.
Acting upon to the request of the customer, this option will enable the user to issue a card
depending on the eligibility criteria of the customer. This option ble the user to generate a
new card or link the account to the existing card.
</FONT></h4></p>
<h4>
<dl>
<dt>User Inputs are:- </dt>
<br><br>
<dd>
<FONT COLOR = "purple"> Base Number:</FONT>
<BR> User can input Base number and related details will be displayed.
<BR><BR>

<BR><BR>
<dd> <FONT COLOR="purple">CIF Number :</FONT>
<BR> The Customer Identification number can be inputted and related customer
infinputted and related customer information details will be displayed.
<BR><BR>
<dd> <FONT COLOR="purple">Customer Name :</FONT>
<BR> The user can input customer name and get the details</H4>
</dl>
<br><br>
<h4>For Eg</h4>

<center>

<table bgcolor="#caeafa" width="100%" frame = 'vhs' border = 3 >
```


ATM - MAIN MENU

```
<html>
<head>
<title> ATM PROCESSING MENU </TITLE>
<script src = "Date.js">
</script>

<script>
function fun(n)
{
if(n==1)
{

if (document.layers[1].visibility == "hide")
{
document.layers[1].visibility="show";
}

else
{
document.layers[1].visibility="hide";
}
}
}
</script>
</head>

<body bgcolor = "#caeafa" text = "blue">
<form name='f1' method='post'>
<center> </center>
<center> </center>
<br>
<h4> <CENTER><u><b> ATM CARD ISSUANCE MENU
</b></u></CENTER></h4>

<layer top = "250" left = "270" height="40" width="50%" name = "a1" >

<table bgcolor='blue'width="63%" border=1>
<tr>
```

```
<td ALIGN = 'CENTER'> <a href='javascript:fun(1)'><font color='white'> CITICARD  
ISSUANCE</font> </a> </td>  
</tr>  
</table>
```

```
</layer>
```

```
<layer ALIGN = 'CENTER' width = "70%" top = "277" left = "270" height = "40" name  
= "a2" visibility='hide'><table width="45%" bgcolor = 'blue' frame='vhs' border = 1>  
<tr><td align='center'><a href = "second.html"><font color='white'> ADD</font> <  
<tr><td align='center'><a href = "view2.jsp"><font color='white'> VIEW</font> </  
<tr><td align='center'><a href = "deletel.jsp"><font color='white'> DELETE</font
```

```
</table>
```

```
</layer>
```

```
<layer name='l1' top='400' left=0 width='100%' height='2%' bgcolor="blue">
```

```
<table border=0 width="100%">
```

```
<tr>
```

```
<td>
```

```
<a href='atm1.html'></a>
```

```
</td>
```

```
<td>
```

```
<a href='atm0.html'></a>
```

```
</td>
```

```
<td >
```

```
<a href='atma.html'></a>
```

```
</td>
```

```
</tr>
```

```
</tr>
```

```
</table>
```

```
</layer>
```

```
</form>
```

```
</body>
```

```
</html>
```

CARD ISSUANCE – ADD

```
<%@ page import="java.sql.*,java.util.*,java.util.Properties ,
weblogic.common.T3Client,java.net.InetAddress" %>
<%
String CNO=null;
String CNA=null;
String ACN=null;
String CTP=null;
String CAN=null;
String TNO=null;
String EPN=null;
String ISD=null;
String EPD=null;

CNO = request.getParameter("t1");
CNA = request.getParameter("t2");
ACN = request.getParameter("sel1");
CTP = request.getParameter("sel");
CAN = request.getParameter("sel10");

EPN = request.getParameter("t6"); Connection con = null;
Statement st = null;
ResultSet rs = null;

try{
    Class.forName("weblogic.jdbc.pool.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);

    if(con==null)
    {
        out.println("Connection null");
    }
} catch(Exception e )
{
    out.println("Error getting Connection :" +e);
}

try{
    st = con.createStatement();
    CAN=CAN!=null?CAN:"";
    if(CAN.equals("PEI-CARD"))
    {
```

```

ISD = request.getParameter("t7"); TNO=request.getParameter("no");
    if(TNO.length() != 0)
    {
        rs = st.executeQuery("select '"+TNO+"' card_no from dual");
    }
}
else if(CAN.equals("E-CARD"))
{
    rs = st.executeQuery("select card_no from emboss where card_no in (se
}
else if(CAN.equals("PE-CARD"))
{
    rs = st.executeQuery("select card_no from preemboss where card_no in
}

while(rs.next())
{
    Statement st1=con.createStatement();
    if(CAN.equals("E-CARD"))
    {
        st1.executeUpdate("update emboss set status='inactive' where card_
    }
    else if(CAN.equals("PE-CARD"))
EPD = request.getParameter("t8");
String myarray[]={ " ", "CITIGOLD", "CITIBLUE"};

Connection con = null;
Statement st = null;
ResultSet rs = null;

try{
    Class.forName("weblogic.jdbc.pool.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);

    if(con==null)
    {
        out.println("Connection null");
    }
} catch(Exception e )
{
    out.println("Error getting Connection :"+e);
}
try{
    st = con.createStatement();
    CAN=CAN!=null?CAN:"";

```

```

if(CAN.equals("PEI-CARD"))
{
    TNO=request.getParameter("no");if(TNO.length() != 0)
    {
        rs = st.executeQuery("select '"+TNO+"' card_no from dual");
    }
}
else if(CAN.equals("E-CARD"))
{
    rs = st.executeQuery("select card_no from emboss where card_no in (select
min(card_no) from emboss where status='active') ");
}
else if(CAN.equals("PE-CARD"))
{
    rs = st.executeQuery("select card_no from premboss where card_no in (select
min(card_no) from premboss where status='active') ");
}

while(rs.next())
{
    Statement st1=con.createStatement();
    if(CAN.equals("E-CARD"))
    {
        st1.executeUpdate("update emboss set status='inactive' where card_
    }
    else if(CAN.equals("PE-CARD"))
{
    rs = st.executeQuery("select card_no from premboss where card_no in
}

while(rs.next())
{
    Statement st1=con.createStatement();
    if(CAN.equals("E-CARD"))
    {
        st1.executeUpdate("update emboss set status='inactive' where card_
    }
    else if(CAN.equals("PE-CARD"))
    {
        st1.executeUpdate("update premboss set status='inactive' where ca
    }
    String TNO1 = rs.getString("CARD_NO");
%>
<html>
<head>
<title> hi </title>

```

```

if(CAN.equals("PEI-CARD"))
{
    TNO=request.getParameter("no");if(TNO.length() != 0)
    {
        rs = st.executeQuery("select '"+TNO+"' card_no from dual");
    }
}
else if(CAN.equals("E-CARD"))
{
    rs = st.executeQuery("select card_no from emboss where card_no in (select
min(card_no) from emboss where status='active') ");
}
else if(CAN.equals("PE-CARD"))
{
    rs = st.executeQuery("select card_no from preemboss where card_no in (se lect
min(card_no) from preemboss where status='active') ");
}

while(rs.next())
{
    Statement st1=con.createStatement();
    if(CAN.equals("E-CARD"))
    {
        st1.executeUpdate("update emboss set status='inactive' where card_
    }
    else if(CAN.equals("PE-CARD"))
{
    rs = st.executeQuery("select card_no from preemboss where card_no in
}

while(rs.next())
{
    Statement st1=con.createStatement();
    if(CAN.equals("E-CARD"))
    {
        st1.executeUpdate("update emboss set status='inactive' where card_
    }
    else if(CAN.equals("PE-CARD"))
    {
        st1.executeUpdate("update preemboss set status='inactive' where ca
    }
    String TNO1 = rs.getString("CARD_NO");
}

%>
<html>
<head>
<title> hi </title>

```

```
<script src = "Date.js">
```

```
{
```

```
<script>
```

```
// var window1=open("ten.jsp","secondWindow", "scrollbars=yes,width=2
```

```
function yes()
```

```
{
```

```
var CTP = '<%=CTP %>';
```

```
if(CTP=='CITIGOLD')
```

```
{
```

```
document.layers[0].visibility = 'show';
```

```
}
```

```
else if(CTP=='CITIBLUE')
```

```
{
```

```
document.layers[1].visibility = 'show';
```

```
}
```

```
else if(CTP=='')
```

```
{
```

```
document.layers[2].visibility = 'show';
```

```
}
```

```
}
```

```
function change()
```

```
{
```

```
var CTP = '<%=CTP %>'; if(CTP=='CITIGOLD')
```

```
{
```

```
if(document.l6.document.fm4.d1.checked == true)
```

```
{
```

```
document.l4.visibility='show';
```

```
}
```

```
else
```

```
{
```

```
document.l4.visibility='hide';
```

```
}
```

```
}
```

```
if(CTP=='CITIBLUE')
```

```
{
```

```
if(document.l6.document.fm4.d1.checked == true)
```

```
{
```

```
document.l8.visibility='show';
```

```
}
```

```

        else
        {if(CTP=='CITIGOLD')
{
if(document.l7.document.fm5.d2.checked == true)
{
document.l5.visibility='show';
}
else
{
document.l5.visibility='hide';

}
}
}
if(CTP=='CITIBLUE')
{
if(document.l7.document.fm5.d2.checked == true)
{
document.l9.visibility='show';
}
else
{function sub()
{

document.frm.cno.value=document.frm1.t1.value;
document.frm.cna.value=document.frm1.t2.value;
document.frm.acn.value=document.frm1.t9.value;
document.frm.ctp.value=document.frm1.t6.value;
document.l9.visibility='hide'; </script>
</head>
<body bgcolor = "#caeafa" text='blue' onLoad=yes()>

<form name='frm' method= 'post' action='eleven.jsp'>
<input type='hidden' name='cno' value="">
<input type='hidden' name='cna' value="">
<center>
<table border=1 bgcolor="#caeafa" width = "50%">
<caption><b><u> LIMIT TRADING FOR THE CASH WITHDRAW</u></b></caption>
<tr>
<center><h4><td><b> Cash Amount</b></td></h4><h4><td><b>Cash Currency
</td></h4>
</tr>
<tr> <td><input type = "text" size = 10 value='16000' onFocus=this.blur()></td>
<td><Select name = "sel5">
<option name = 'a1' width =10> </option>
<option name = 'a2' width = 10 selected> U.S.DOLLAR </option>

```

```

<option name = 'a3' width = 10> RUSSIAN ROUBEL </option>
<option name = 'a4' width = 10> AUSTRALAIAN DOLLAR </option>
<option name = 'a5' width = 10> HONGKONG DOLLAR </option>
<option name = 'a6' width = 10> JAPANESE YEN </option>
<option name = 'a7' width = 10> INDIAN RUPEE </option>
<option name = 'a8' width =10> OTHERS </option>
</select></td>
<td><input type = "text" size = 10 value='12'onFocus=this.blur()> </td>
<td><SELECT NAME = "SEL6">
<option name = 'a1' width =10> </option><tr><td ><input type = "text" size = 10
value='12000'onFocus=this.blur()></td>
<td><Select name = "sel7">
<option name = 'a1' width =10> </option>
<option name = 'a2' width = 10> U.S.DOLLAR </option>
<option name = 'a3' width = 10> RUSSIAN ROUBEL </option>
<option name = 'a4' width = 10 selected> AUSTRALAIAN DOLLAR </option>
<option name = 'a5' width = 10> HONGKONG DOLLAR </option>
<option name = 'a6' width = 10> JAPANESE YEN </option>
<option name = 'a7' width = 10> INDIAN RUPEE </option>
<option name = 'a8' width =10> OTHERS </option>
</select></td>>table border=1 bgcolor="#caeafa" width = "50%">
<caption><b><u> LIMIT TRADING FOR THE CASH WITHDRAW</u></b></caption>
<tr>
<center><h4><td><b> Cash Amount</b></td></h4><h4><td><b>Cash Currency
</td></h4>
</tr>
<tr> <td><input type = "text" size = 10 value='13000'onFocus=this.blur() ></td>

    <td><Select name = "sel5a">
<option name = 'a1' width =10> </option>
<option name = 'a2' width = 10 selected> U.S.DOLLAR </option>
<option name = 'a3' width = 10> RUSSIAN ROUBEL </option>
<option name = 'a4' width = 10> AUSTRALAIAN DOLLAR </option>
<option name = 'a5' width = 10> HONGKONG DOLLAR </option>
<option name = 'a6' width = 10> JAPANESE YEN </option>
<option name = 'a7' width = 10> INDIAN RUPEE </option>
<option name = 'a8' width =10> OTHERS </option>
</%

}
rs.close();

con.close();
}
catch(Exception e)
{

```


CARD ISSUANCE – VIEWING OF CARD DETAILS

```
<% @ page import = "java.sql.*,java.util.*,java.util.Properties,weblogic.common.T3client:
%>
<html>
<head>
<title> VIEW </title>
</head>
<body bgcolor = "#caeafa" text = "blue">
<form name='fom'>
<center><img src='mayank.jpg'width='90%'></center>
<center><img src='balaji.jpg'width='90%'></center>
<a href='atm2.html' title='click'> Main Menu </a>
<center><layer name='a1' width='85%' top=150 left=0 bgcolor='blue'>
<b><font color='white'><center> CitiBank Administration Maintainence - Viewing
</layer>
</center>
<layer name = "lay1" left = 10 width = "100%" top='200' >
<form name = "frm">

<table bgcolor= '#caeafa' width = "75%" border = 1>
<tr>
<th bgcolor='blue'><b><font color='white'> CIF-Number</font></b></th><th bgcolor
<th bgcolor='blue'><b><font color='white'>CardNumber</font></b></th></tr>

<%
%>

Connection con=null;

Statement st = null;

ResultSet rs = null;

try{
    Class.forName("weblogic.jdbc.pool.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);
    con.setAutoCommit(false);

    if(con==null)
```

```

{
System.out.println("Connection null");
} try{
    st = con.createStatement();
    rs = st.executeQuery(" select cif_no,ac_no,card_type,card_name,card_no "
    System.out.println (rs);

    while ( rs.next() )
    {
String CifNumber = rs.getString("cif_no");
String AccountNumber= rs.getString("ac_no");
String CardType = rs.getString("card_type");
String CardName = rs.getString("card_name");
String CardNumber=rs.getString("card_no");

out.print("<tr>");
out.print("<td>" + CifNumber + "</td>");
out.print("<td>" + CardNumber + "</td>");
out.print("</tr>");

    }
rs.close();
con.close();
    }
catch(Exception e)
    {
    System.out.println("Error level3 :" + e);
    }
%> /table>

</form>
</layer>
</body>
</html>

```

CARD ISSUANCE – VIEWING OF A PERSONAL DETAILS

```
<%@ page import = "java.sql.*,java.util.*,java.util.Properties,weblogic.common.T
<html>
<head>
<title> VIEW 1</title>
</head>
<body bgcolor = "#caeafa" text = "blue" >
<form name='fom1'>
<center><img src='mayank.jpg'width='90%'></center>
<center><img src='balaji.jpg'width='90%'></center>
<a href='atm2.html' title='click'> Main Menu </a>

<layer name = "lay1" left = 10 width = "100%" top='150' >
<form name = "frm1">
<%

Connection con=null;

Statement st = null;

ResultSet rs = null;

String ac_no = null;

ac_no = request.getParameter("ac_no");

try{

Class.forName("weblogic.jdbc.pool.Driver").newInstance();
con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);
con.setAutoCommit(false);

if(con==null) <layer name = "lay1" left = 10 width = "100%" top='150' >
```

```

<form name = "frm1">
<%

Connection con=null;

Statement st = null;

ResultSet rs = null;

String ac_no = null;

ac_no = request.getParameter("ac_no");

try{

    Class.forName("weblogic.jdbc.pool.Driver").newInstance();
    con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);
    con.setAutoCommit(false);

    if(con==null) String BaseNumber= rs.getString("BASE_NO");
        String CifNumber = rs.getString("CIF_NO");
        String CifName = rs.getString("CIF_NAME");
        String AccountCurrency=rs.getString("ACC_CURRENCY");
        String RefCurrency=rs.getString("REF_CURRENCY");
        String LegalStatus=rs.getString("LEGAL_STATUS");
        String SignatureRule=rs.getString("SIGNATURE_RULE");

    %>

<center><table bgcolor='#cacafa' border=2>
<caption> <b><u> Personal Details</u></b></caption>
<tr> <td><b>AccountNumber </b></td>
<%

out.print(" <td>" + ac_no1 +" </td></tr>");
%>
<tr><td><b>BaseNumber </b></td>
<%

out.print(" <td> " + BaseNumber + "</td></tr>");
%>
<tr><td><b>CifNumber </b></td>
%>

```

```
<center><table bgcolor='#caeafa' border=2>
<caption> <b><u> Personal Details</u></b></caption>
<tr> <td><b>AccountNumber </b></td>
<%
```

```
out.print(" <td>" + ac_no1 + " </td></tr>");
<%>
<tr><td><b>BaseNumber </b></td>
<%
```

```
out.print(" <td> " + BaseNumber + " </td></tr>");
<%>
<tr><td><b>CifNumber </b></td>
<%
```

```
out.print("<td>" + CifNumber + " </td></tr>");
<%>
<tr><td><b>CifName </b></td>
<%
```

```
<%>
<tr><td><b>BaseNumber </b></td>
<%
```

```
out.print(" <td> " + BaseNumber + " </td></tr>");
<%>
<tr><td><b>CifNumber </b></td>
<%
```

```
out.print("<td>" + CifNumber + " </td></tr>");
<%>
<tr><td><b>CifName </b></td>
<%
```

```
out.print("<td>" + CifName + " </td></tr>");
<%>
<tr><td><b>AccountCurrency </b></td>
<%
```

```
out.print("<td>" + AccountCurrency + " </td></tr>");
<%>
<tr><td><b>RefferenceCurrency</b></td>
<%
```

```
out.print("<td>" + RefCurrency + " </td></tr>");
<%>
<tr><td><b>LegalStatus </b></td>
out.print("<td>" + RefCurrency + " </td></tr>");
<%>
```

```
<tr><td><b>LegalStatus </b></td>
<%>
out.print("<td>" + LegalStatus + " </td></tr>");
```

```
%>
<tr><td><b>SignatureRule </b></td>
<%
out.print("<td>" + SignatureRule + "</td></tr>");
%>
<%
}
rs.close();
con.close();
}
catch(Exception e)
{
System.out.println("Error level3 :" + e);
}
%>
</table>
</center> </form>
</layer>
<layer name='lr1' width='100%' laeft=0 top='400'>
<form name='fmr' >
<center><a href='view2.jsp'>View</a> </center>
</form>
</layer>
</body>
</html>
```

DELETION :- DELETION OF CARD

```
<%@ page import = "java.sql.*,java.util.*,java.util.Properties,weblogic.common.T
<html>
<head>
<title> Delete </title>
<script>
function del()
{
  var i = 0;
  wlen=document.lay1.document.frm.c.length;
  for(var j=0;j<wlen;j++)
  {
    var k = "document.lay1.document.frm.c[j].checked=true";
    eval(k);
  }
}

function del2()
{
  var l="";
  var i = 0;
  wlen=document.lay1.document.frm.c.length;
  alert(wlen);
  for(var j=0;j<wlen;j++)

for(var j=0;j<wlen;j++)
  {
    var k = "document.lay1.document.frm.c[j].checked";
    if(eval(k) == true )
    {
      w = "document.lay1.document.frm.c[j].value";
      varcard = eval(w);
      var l = var l+varcard+'|';
    }
  }
  len1 = var l.length;
  len1 = len1 - 1;
  var l=var l.substring(0,len1);
```

```
document.frm1.pack.value=var1;
alert(document.frm1.pack.value);
document.frm1.action='delete2.jsp';
document.frm1.submit();
}
```

```
function cut()
```

```
{
  wlen=document.lay1.document.frm.c.length;
  for (var j=0;j<wlen;j++)
  {
    document.lay1.document.frm.c[j].checked=false;
  }
}
```

```
</script>
```

```
</head>
```

```
<body bgcolor = "#cacafa" text = "blue">
```

```
<form name='fom'>
```

```
<center><img src='mayank.jpg'width='90%'></center>
```

```
<center><img src='balaji.jpg'width='90%'></center>
```

```
<a href='atm2.html' title='click'> Main Menu </a>
```

```
<center><layer name='a1' width='85%' top=150 left=0 bgcolor='blue'>
```

```
<b><font color='white'><center> CitiBank Administration Maintainence - Deleting
```

```
</layer>
```

```
</center>
```

```
<form name='frm1' method= 'post' >
```

```
<input type='hidden' name='pack' value="">
```

```
</form>
```

```
th bgcolor='blue'><b><font color='white'>CardNumber</font></b></th></tr>
```

```
<%
```

```
System.out.println("hi This is delete phase");
```

```
Connection con=null;
```

```
Statement st = null;
```

```
ResultSet rs = null;
```

```
try{
```

```
Class.forName("weblogic.jdbc.pool.Driver").newInstance();
```

```
con = DriverManager.getConnection("jdbc:weblogic:pool:ipboasis",null);
```

```
con.setAutoCommit(false);
```

```
if(con==null)
```

```
{
```

```
System.out.println("Connection null");
```



```
</body>
</html>
} catch(Exception e)
```

DELETE :- DELETION OF A CARD

```
..<%@ page import="java.sql.*,java.util.*,java.util.Properties,java.util.Enumeration %>
```

```
<html>
<head>
<title> delete1</title>
</head>
<body bgcolor="#cacafa" text="blue">
```

```
<%
    request.getParameterNames();
    String cif_no=null;
    cif_no = request.getParameter("pack");
%>
```

```
    <script>
        alert('<%=cif_no%>');
    </script>
```

```
<%
Connection con=null;
Statement st =null;
```

```
try{
    Class.forName("weblogic.jdbc.pool.Driver").newInstance();
if(con==null)
    {
        out.println("Connection null");
    }
} catch(Exception e)
    {
        out.println("Error getting Connection :"+e);
    }
}
```

```
try{
    st = con.createStatement();
    StringTokenizer st1 = new StringTokenizer(cif_no,"|");
    while(st1.hasMoreTokens())
    {
        String sqlstatement = "delete from at01tb where cif_no=" +st1.nextTo
        System.out.println("The cif_no "+cif_no);
    }
}
```

```
        st.executeUpdate(sqlstatement);
    }
    con.commit();
    con.close();
} catch (Exception e)
    {
        out.println("exc " + e );
    }
%>
<center>The Records have been Deleted.</center><br><br>
<center> <a href='atm2.html' title='Click'>Main Menu </a></center>
</body>
</html>
```

QUERY TOOL :- SEARCH ENGINE

```
<html>
<head>
<title>Searching </TITLE>
</head>

<script>

function fun8(n)
{
if(n=8)
{
if(document.layers[6].visibility == 'hide')
{
document.layers[6].visibility = 'show';
}
else
{
document.layers[6].visibility = 'hide';
}
}
}

function fun7(n)
{
if(document.layers[5].visibility == 'hide')
{
document.layers[5].visibility = 'show';
}
else
{
document.layers[5].visibility = 'hide';
}
}

function fun6(n)
{
if(n=6)
{
if(document.layers[5].visibility == 'hide')
{
```

```

document.layers[5].visibility = 'show';
}
else
{
document.layers[5].visibility = 'hide';
if(n=5)
{
if(document.layers[4].visibility == 'hide')
{
document.layers[4].visibility = 'show';
}
else
{
document.layers[4].visibility = 'hide';
}
}
}
}
}

```

function fun2(n)

```

{
if(n=1)
{
if(document.layer5.visibility == 'hide')
{
document.layer5.visibility = 'show';
}
else

```

function sub()

```

{
document.fam.bno.value=document.frm1.t1.value;
document.fam.cno.value=document.frm2.cif_no.value;
document.fam.acn.value=document.frm5.ac_no.value;
document.fam.fnm.value=document.frm3.t3.value;
document.fam.mnm.value=document.frm3.t4.value;
document.fam.lnm.value=document.frm3.t5.value;
document.fam.con.value=document.frm4.t6.value;
document.fam.cid.value=document.frm6.t8.value;
document.fam.submit();

}

```

```

</script> <body bgcolor = "#cacafa" text = "blue">
<center><img src='mayank.jpg' width='75%'></center>

```

```
<center><img src='balaji.jpg' width='75%'> <center>
```

```
<table bgcolor = "blue" width = "100%" border = 1 frames = "vhs">
<tr><td> <a href = 'javascript:fun()'><center><font color='white'> Base Number </fo
<td> <a href = 'javascript:fun4()'> <center><font color='white'> CIF Number</fo
<td> <a href = 'javascript:fun2()'><center> <center><font color='white'> Name</fo
<td> <a href = 'javascript:fun1()'><center><font color='white'> Others</font></c
</table>
```

```
<form name='fam' method= 'post'>
<input type='hidden' name='bno' value="">
<input type='hidden' name='cno' value="">
<input type='hidden' name='fnm' value="">
<input type='hidden' name='mnm' value="">
<input type='hidden' name='lnm' value="">
<input type='hidden' name='cnm' value=""> </form>
```

```
<layer name = "layer1" top =220 left = 170 width = "50%" height = "20%" visibi
<center> <form name = 'frm1' method = 'post' action=base.jsp > <input type =
</form>
</layer>
```

```
<layer name = "layer2" top = 220 left = 170 width = "50%" height = "20%" visibil
<center> <form name = 'frm2' method='post' action=atm5.jsp> <input type = "text"
</form>
</layer>
```

```
<layer name = "layer3" top = 130 left = 405 width = 60%" visibility = 'hide' >
<table bgcolor = "blue" border = 1 width = "25%">
<!--<tr><td> <a href = 'javascript:fun5()'><center><font color='white'>Individu
<!--<tr><td> <a href = 'javascript:fun6()'><center><font color='white'>CompanyN
</table>
</layer>
```

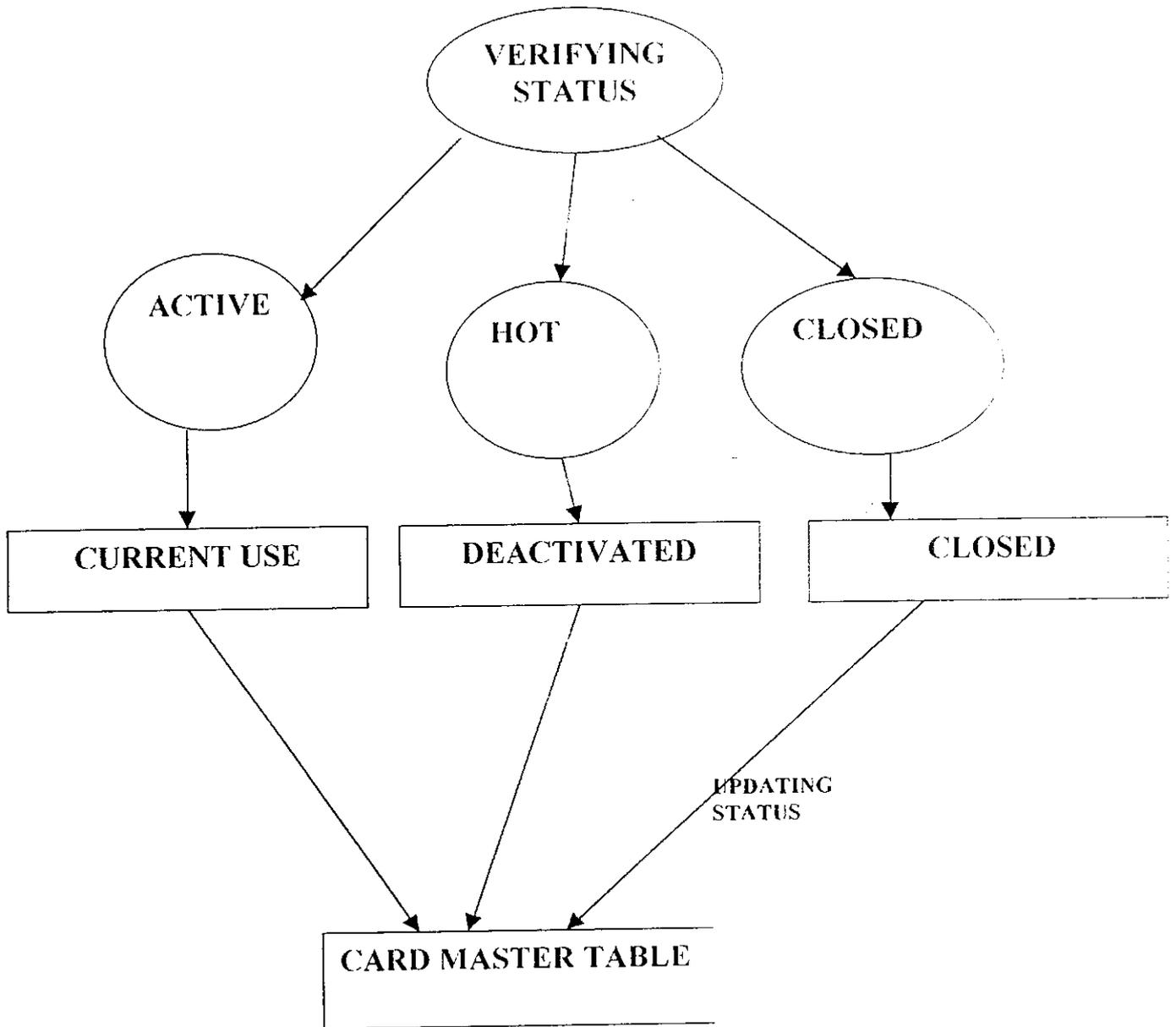
```
<layer name = "layer4" top = 150 left = 625 width = 55%" visibility = 'hide' >
<table bgcolor = "blue" border = 1 width = "25%">
<tr><td><a href = 'javascript:fun7()'><font color='white'> AccountNumber</font>
<input type='hidden' name='acn' value="">
<center> <form name = 'frm4'> <input type = "text" name = "t6" size = 16> <br>
</form>
</layer>-->
```

```
<layer name = "layer7" top = 220 left = 6 width = "100%" height = "40%" visibili
<center><form name = 'frm5' action=account.jsp><input type = "text" name = "ac_no
```

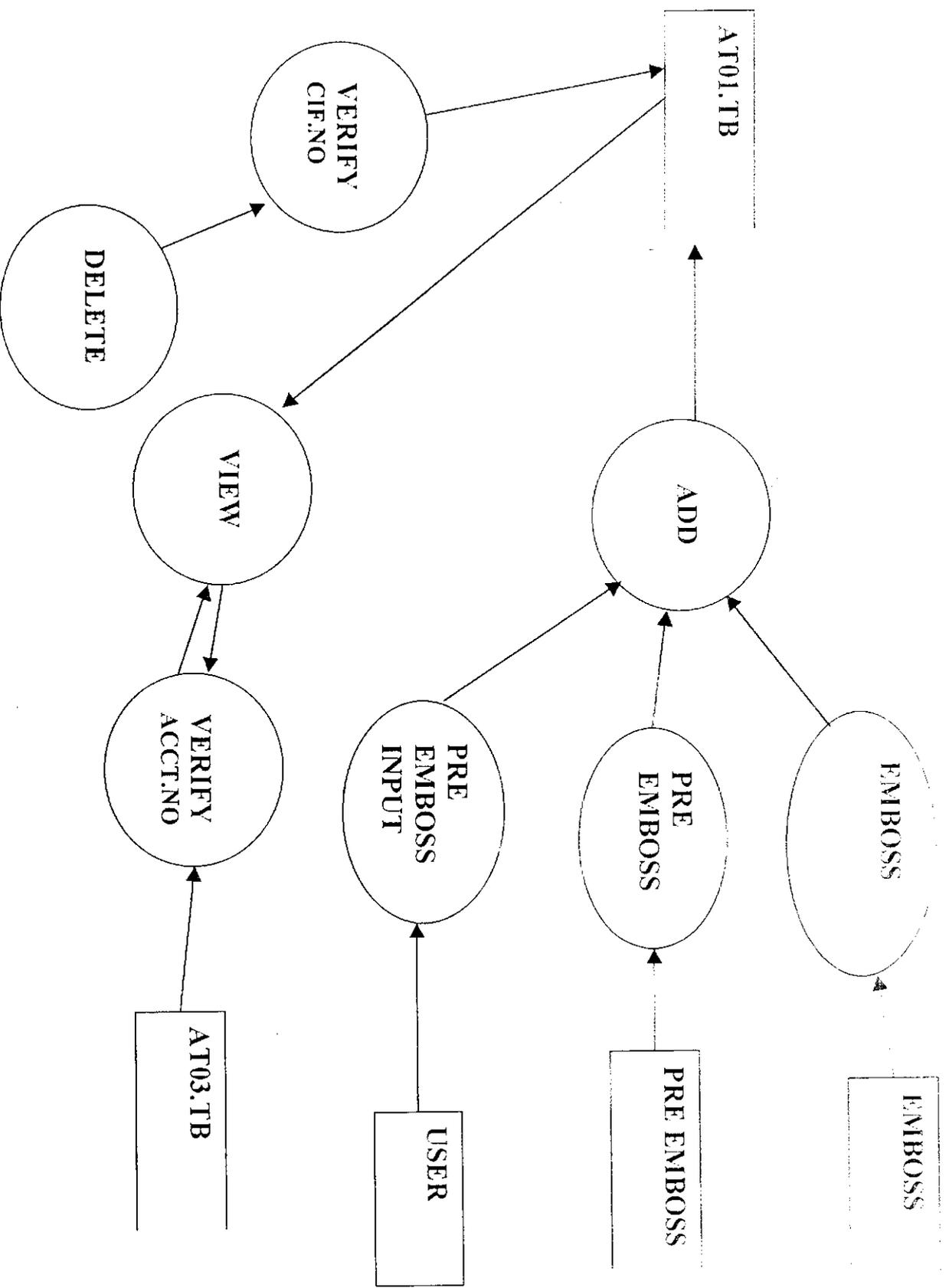
```
</form>
</layer>
```

```
<layer name = "layer8" top = 220 left = 6 width = "100%" height = "40%" visibili
<center><form name = 'frm6' method = 'post' action=cardsearch.jsp ><input type =
</form>
</layer>
</body>
</html>
```

STATUS CHANGE



DATA FLOW DIAGRAM - CARD ISSUANCE



Consumer Banking

ATM Processing Menu

[Go Back](#)

CITICARD ISSUANCE-ADMINISTRATION

CIF Number	<input type="text" value="1010101020"/>	CIF Name	<input type="text" value="basa"/>
Primary AccountNo	<input type="text" value="A00010"/>	Card Type	<input type="text" value="CITIGOLD"/>
Card No	<input type="text" value="E-CARD"/>	Encrypted Pin	<input type="text" value="abababababa"/>
Issue Date	<input type="text" value="14-JAN-2002"/>	Expiry Date	<input type="text" value="24-JAN-2012"/>

[Go Back](#)

CITICARD ISSUANCE-ADMINISTRATION

CIF Number	1010101020	CIF Name	pasa
Primary AccountNo	A00010	Card Type	CITIGOLD
Card No	E-CARD 30303030303	Encrypted Pin	bababababab
Issue Date	14-JAN-2002	Expiry Date	24-JAN-2012

CARD FACILITIES ADDITION

CITIGOLD

- Direct Debit
- Resident Status
- Sell MF
- Deposites
- Internal Transfer
- Home Banking
- Buy MF
- Standing Order

hi - Netscape

File Edit View Go Communicator Help

Limit Trading for the Cash Withdraw

Limit for the POS

Document Done

Limit Trading for the Cash Withdraw

Limit for the POS

LIMIT TRADING FOR THE CASH WITHDRAW

Cash Amount	Cash Currency	TxnCount	Limit Peroid
16000	U.S.DOLLAR	12	DAILY
12000	AUSTRALAIAN DOLLAR	7	WEEKLY
17000	JAPANESE YEN	8	MONTHLY
200000	INDIAN RUPEE	12	YEARLY

LIMIT TRADING FOR THE POS

Purchase Amount	Purchase Currency	TxnCount	Limit Peroid
12000	U.S.DOLLAR	9	DAILY
13000	RUSSIAN ROUBEL	6	WEEKLY
12000	AUSTRALAIAN DOLLAR	7	MONTHLY
15000	INDIAN RUPEE	9	YEARLY

Thank u

The Diabase has been Updated.

[Main Menu](#)

Go Back

CITICARD ISSUANCE-ADMINISTRATION

CIF Number	1010101021	CIF Name	sangeetha
Primary AccountNo	A00011	Card Type	CITIBLUE
Card No	PE-CARD 20202020202	Encrypted Pin	sansansansn
Issue Date	08-JAN-2002	Expiry Date	25-JAN-2012

CARD FACILITIES ADDITION

CITIBLUE

- See Information
- REF EcyEuro
- Related
- Unit Profit ID
- Intermitt
- Category

[Go Back](#)

CITICARD ISSUANCE-ADMINISTRATION

CIF Number	<input type="text"/>	CIF Name	<input type="text"/>
Primary AccountNo	<input type="text"/>	Card Type	<input type="text"/>
Card No	<input type="text" value="PEI-CARD"/>	Encrypted Pin	<input type="text"/>
Issue Date	<input type="text"/>	Expiry Date	<input type="text"/>



Online Realtime Banking Integrated Technology



Main Menu

CitiBank Administration Maintenance - Viewing Purpose:-

CIF-Number	AccountNumber	CardType	CardName	CardNumber
1010101020	A00010	CITIGOLD	E-CARD	3030303030303018
1010101013	A00004	CITIGOLD	E-CARD	3030303030303011
1010101014	A00005	CITIGOLD	PE-CARD	2020202020202011
1010101015	A00006	CITIBLUE	E-CARD	3030303030303012
1010101012	A00003	CITIGOLD	PEI-CARD	4040404040404010
1010101018	A00009	CITIGOLD	E-CARD	3030303030303014



Online Realtime Banking Integrated Technology



Main Menu

Personal Details

AccountNumber	A00010
BaseNumber	1020304054
CifNumber	1010101019
CifName	basa
AccountCurrency	usdollar
ReferenceCurrency	resident
LegalStatus	active
SignatureRule	single

View

CITIBANK



Online Realtime Banking Integrated Technology



Main Menu

CitiBank Administration Maintenance - Deleting Purpose:-

Delete	CIF-Number	AccountNumber	CardType	CardName	CardNumber
<input type="checkbox"/>	1010101020	A000010	CITIGOLD	E-CARD	3030303030303018
<input type="checkbox"/>	1010101013	A000004	CITIGOLD	E-CARD	3030303030303011
<input type="checkbox"/>	1010101014	A000005	CITIGOLD	PE-CARD	2020202020202011
<input type="checkbox"/>	1010101015	A000006	CITIBLUE	E-CARD	3030303030303012
<input type="checkbox"/>	1010101012	A000003	CITIGOLD	PEI-CARD	4040404040404010
<input type="checkbox"/>	1010101018	A000009	CITIGOLD	E-CARD	3030303030303014

CitiBank Administration Maintenance - Deleting Purpose: -

Delete	CIF-Number	AccountNumber	CardType	CardName	CardNumber
<input checked="" type="checkbox"/>	1010101020	A00010	CITIGOLD	E-CARD	3030303030303018
<input checked="" type="checkbox"/>	1010101013	A00004	CITIGOLD	E-CARD	3030303030303011
<input checked="" type="checkbox"/>	1010101014	A00005	CITIGOLD	PE-CARD	2020202020202011
<input checked="" type="checkbox"/>	1010101015	A00006	CITIBLUE	E-CARD	3030303030303012
<input checked="" type="checkbox"/>	1010101012	A00003	CITIGOLD	PE-CARD	4040404040404010
<input checked="" type="checkbox"/>	1010101018	A00009	CITIGOLD	E-CARD	3030303030303014

Track All cancel

CitiBank Administration Maintenance - Deleting Purpose:-

Delete	CIF-Number	AccountNumber	CardType	CardName	CardNumber
<input checked="" type="checkbox"/>	1010101020	A00010	CITIGOLD	E-CARD	3030303030303018
<input type="checkbox"/>	1010101013	A00004	CITIGOLD	E-CARD	3030303030303011
<input type="checkbox"/>	1010101014	A00005	CITIGOLD	PE-CARD	2020202020202011
<input type="checkbox"/>	1010101015	A00006	CITIBLUE	E-CARD	3030303030303012
<input type="checkbox"/>	1010101012	A00003	CITIGOLD	PEI-CARD	4040404040404010
<input checked="" type="checkbox"/>	1010101018	A00009	CITIGOLD	E-CARD	3030303030303014

Check All Delete cancel

The Records have been Deleted.

Main Menu



CITIBANK

Online Realtime Banking Integrated Technology



Base Number

CIF Number

Name

Others

AccountNumber

CitiCard

CITIBANKO

Online Realtime Banking Integrated Technology



Base Number

CIF Number

Name

Others

1020304040

Base Number

Online Realtime Banking Integrated Technology



Main Menu

Personal Details

BaseNumber	1020304040
AccountNumber	A00001
CifNumber	1010101010
CifName	nandhini
AccountCurrency	usdollar
ReferenceCurrency	resident
LegalStatus	active
SignatureRule	single

OK

CITIBANK

Online Realtime Banking Integrated Technology

Global

Base Number

CIF Number

Name

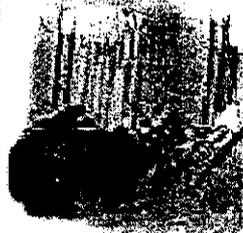
Others

1010101013

CIF Number



Online Realtime Banking Integrated Technology



Main Menu

Personal Details

CIFNumber	1010101013
AccountNumber	A00004
CardType	CITIGOLD
CardName	E-CARD
CardNumber	30303030303011

OK

CITIBANK

Online Realtime Banking Integrated Technology



Base Number

CIF Number

Name

Others

nandhini

Individual Name

СІТІБАНК

Online Realtime Banking Integrated Technology



Main Menu

Personnal Details

CIFName	nandhini
AccountNumber	A00001
BaseNumber	1020304040
CifNumber	1010101010
AccountCurrency	usdollar
ReferenceCurrency	resident
LegalStatus	active
SignatureRule	single

03

CITIBANKO

Online Realtime Banking Integrated Technology



Base Number

CIF Number

Name

Others

AccountNumber

CitiCard

A00001

Account Number

CITIBANK

Online Realtime Banking Integrated Technology



Main Menu

Personal Details

AccountNumber	A00001
BaseNumber	1020304040
CifNumber	1010101010
CifName	nandhini
AccountCurrency	usdollar
ReferenceCurrency	resident
LegalStatus	active
SignatureRule	single

OK



Online Realtime Banking Integrated Technology



<u>Base Number</u>	<u>CIF Number</u>	<u>Name</u>	<u>Others</u>
			<u>AccountNumber</u>
			<u>CitiCard</u>

[03030303011]

Citicard Number

CITIBANK

Online Realtime Banking Integrated Technology



Main Menu

Personal Details

CardNumber	3030303030303011
CIFNumber	1010101013
AccountNumber	A00004
CardType	CITIGOLD
CardName	E-CARD

UK

Card Administration

Maker Level		Status Change	Addition Mode
Card Number	5081400160010026	Card Type	CITIGOLD
Cif Number	000000002Z	Existing Status	HOT
Issue Date	01-MAR-1997	Expiry Date	31-DEC-2040
New Status	Active	Status Reason	ACTIVE

Accounts Linked to Card

Primary Account	Account Number	Account Type	Currency	Balance
Y	0000030007(CHE)	CURRENT ACCOUNT	USD	100.00000
Y	0000189006(CHE)	CURRENT ACCOUNT	USD	100.00000
N	0000030016(CHE)	CURRENT ACCOUNT	INR	1067790.2100

[+] Submit [F] Cancel



For Help, click Help Topics on the Help Menu.

Card Administration

Maker Level		Status Change		Modification Mode	
Card Number	5091400160010026	Card Type	CITI6OLD		
Cif Number	000000002Z	Existing Status	HOT		
Issue Date	01-MAR-1997	Expiry Date	31-DEC-2049		
New Status	Active	Status Reason	ACTIVE		

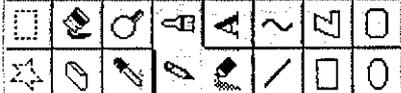
Accounts Linked to Card

Primary Account	Account Number	Account Type	Currency	Balance
Y	0000030007(CHE)	CURRENT ACCOUNT	USD	100.0000
Y	0000189006(CHE)	CURRENT ACCOUNT	USD	100.0000
H	0000030015(CHE)	CURRENT ACCOUNT	INR	1067700 2100



For Help, click Help Topics on the Help Menu.

95.90



Netscape

Card Administration

Master Level		Status Change		Deletion Mode
Card Number	5081400160010028	Card Type	CITIGOLD	
Cif Number	0000000027	Existing Status	HOT	
Issue Date	01-MAR-1997	Expiry Date	31-DEC-2049	
New Status	Active	Status Reason	ACTIVE	

Accounts Linked to Card

Primary Account	Account Number	Account Type	Currency	Balance
Y	0000030007[CHE]	CURRENT ACCOUNT	USD	100.0000
Y	0000189008[CHE]	CURRENT ACCOUNT	USD	100.0000
N	0000030015[CHE]	CURRENT ACCOUNT	INR	1087790.2100

[a] Submit [c] Cancel



For Help, click Help Topics on the Help Menu.

389,275



Press Enter to Continue

CIF_NO	CIF_NAME	AC_NO	CARD_TYPE	CARD_NAME
1818181813	savi	A88884	CITIGOLD	E-CARD
3838383838383811	5555555555555555	03-JAN-2002	02-JAN-2002	
1818181814	karthi	A88885	CITIGOLD	PE-CARD
2828282828282811	5555555555555555	01-JAN-2002	30-JAN-2002	
1818181815	vi.jay	A88886	CITIBLUE	E-CARD
3838383838383812	eeeeeeeeeeeeeeee	25-JAN-2012	25-JAN-2012	
1818181812	mayank	A88883	CITIGOLD	PEI-CARD
4848484848484818	mamamamamamama	15-JAN-2002	16-JAN-2002	
1818181818	baby	A88889	CITIGOLD	E-CARD
3838383838383814	4444444444444444	15-JAN-2002	23-JAN-2002	

Elapsed: 00:00:00.57

SQL>

Navigation buttons: Home, Back, Forward, Stop, Refresh, Print, etc.

File Edit Options Send Receive Window Help

Press Enter to Continue

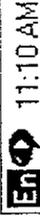
ACC_NO	BASE_NO	CIF_NO	CIF_NAME	ACC_CURRENCY
000002	203040450	1010101011	balaji	usdollar
resident	active	single		
000004	3020120320	1010101013	savi	usdollar
resident	active	single		
000005	4050607010	1010101014	karthi	usdollar
resident	active	single		
000001	1020304040	1010101010	nandhini	usdollar
resident	active	single		
000003	2030405010	1010101012	mayank	usdollar
resident	active	single		
000006	1020304050	1010101015	vijay	usdollar
resident	active	single		

REF_CURRENCY LEGAL STATUS SIGNATURE_

Press Enter to Continue

For

ANSI TCP/IP 00:08



project3 - Paint



Start Searching - Netscape

11:10 AM

169.165.109.38

My Computer | Internet Explorer | My Recent Places | Help



SQL> select * from emboss;
Press Enter to Continue

LSNO	CARD_NAME	CARD_NO	STATUS
1	emboss	30303030303010	inactive
2	emboss	30303030303011	inactive
3	emboss	30303030303012	inactive
4	emboss	30303030303013	inactive
5	emboss	30303030303014	inactive
6	emboss	30303030303016	inactive
7	emboss	30303030303017	inactive
8	emboss	30303030303018	inactive
9	emboss	30303030303019	inactive
10	emboss	30303030303020	inactive

10 rows selected.

Elapsed: 00:00:00.83

SQL>

Start

169.165.109.38

ANSI escape | 1947 | 0000

7:47 PM

169.165.109.38

File Edit Options Send Receive Window Help

File Edit Options Send Receive Window Help

ORA-00942: table or view does not exist

Elapsed: 00:00:00.10

SQL> select * from preemboss;

Press Enter to Continue

SNO	CARD_NAME	CARD_NO	STATUS
1	preemboss	2020202020202010	inactive
2	preemboss	2020202020202011	inactive
3	preemboss	2020202020202012	inactive
4	preemboss	2020202020202013	inactive
5	preemboss	2020202020202014	active
6	preemboss	2020202020202015	active
7	preemboss	2020202020202016	active
8	preemboss	2020202020202017	active
9	preemboss	2020202020202018	active
10	preemboss	2020202020202019	active

10 rows selected.

Elapsed: 00:00:00.70

3 (1)