An Infrastructure for Customizable and Divisible Card Payments for Online Purchases

Using JSP on Apache Tomcat Server

Project Report for fall 2004 submitted to the
Department of Computer Science, College of Computing Sciences
New Jersey Institute of Technology

In Partial Fulfillment of the requirements for the
Degree of Master of Science in Computer Science

Submitted
By
Shreeshah Vedagiri
ID: xxx-xx-xxxx

Project Advisor: Dr. James Geller

Proposal Number: xxxxxxxxxx

New Jersey Institute of Technology

1. Approval by Project Advisor

Project Advisor: _____

Signature: _____

Date: _____

2. Approval by Graduate Advisor(s) / Committee

Proposal Number: _____ Submission Date: _____

Proposal Evaluation: _____

(by Graduate Advisor/Committee)

Date: _____

Signatures: _____
(Sign and write name if more than one sign

I hereby affirm that I have followed the directions as published in the program Web page at

http://www.cis.njit.edu/mscs

and I confirm that this report is my own personal work and that all material other than my own is properly referenced.

Student's Name: _____

Student's Signature: _____

Date: _____

Acknowledgement

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to Prof. James Geller. I can’t say thank you enough for his tremendous support and help. I feel motivated and encouraged every time I attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

Abstract

Customers are often better off if they can use a combination of credit cards for a single purchase. To support this functionality, we need two things. First, we need an infrastructure that allows the customer to divide a single purchase transaction into multiple cards. Second, we need a tool that assists the customer in making the complex decision of which combination of cards to use. This project provides the design a new infrastructure that supports the *divisible card payment* where a combination of multiple cards can be used for a single purchase. The main strength of this virtual card payment infrastructure is that it requires only two minor modifications to the existing infrastructure. First, the V-Card Manager (VCM) is added to the merchant side to handle the divisible card approval process from respective credit-card issuers. Second, the customer is equipped with the V-card Agent (VA) that generates a customized divisible card based on her preferences. "The Divisible Credit Card Payment Project" led by Dr. James Geller of the Computer Science Department of the New Jersey Institute of Technology aims to explore the possibility of applying divisible card payment to the existing infrastructure.

The "Divisible Credit Card Payment" project is divided into four modules. Output of one module is used as input to another module. Thus, modular design helps in further development and maintenance of the system.

The Virtual Agent is one of the four modules of the "Divisible Credit Card Payment" project. As a developer of the project, my responsibility was to develop the Virtual Agent and a bank simulation using Java Server Pages. This is a Web based application that uses Tomcat Web server to run the Java Server Pages. The database used to store the application details is Oracle.

TABLE OF CONTENTS

Part 1: Introduction

Credit cards are the payment choice in e-commerce. Despite the on-going development efforts on various kinds of new payment systems for e-commerce, online shoppers use credit cards for a majority of their purchases. Research shows that 85% of all Internet transactions are done with online credit card payments and that customers are more comfortable with and feel more secure about using credit cards over the Internet (Bohle 2002, Jewson 2001, Lawrence 2002).

When people use credit cards, they expect functionalities different from, say, cash transactions. Credit cards, although not providing anonymity, offer the balance carryover functionality such that the purchase amounts on a credit card can be carried over to the future and be paid in installments with interest. Many credit cards offer additional features, such as cash-back on a percentage of total purchases made, travel protection, additional warranty, or airline frequent flier miles. In such a myriad of choices and features, a customer may be better off using a particular card, depending on his/her preferences and spending habits. For example, a customer who carries a large balance may prefer a card with a lower interest rate, while another customer who does not carry a balance, but likes traveling, may prefer to use a card affiliated with an airline company to receive airline miles. Furthermore, customers are sometimes better off if they can use a combination of credit cards for a single purchase.

This project describes an infrastructure that supports the divisible payments of a single purchase (Soon Ae Chun 2004). In the new infrastructure, a Virtual card (V-card) is created and used each time the customer wants to use a combination of cards. This new infrastructure modifies the existing systems in two ways. First, to support the divisible card payments, the Virtual Card Manager (VCM) is added to the merchant side. The VCM handles the divisible card approval process between the merchant and the respective credit card issuers. Second, to support the customer's card-usage decisions, the new infrastructure provides the customer with a V-card Agent (VA). As which card to use is a complex decision, an optimization model is built into the VA. Based on the customer's preferences, the VA generates

the best option that may suggest using multiple cards for a single purchase (Soon Ae Chun 2004).

It is believed that the proposed infrastructure is well suited for online purchases. The creation of the V-card does not create a physical card but only a valid card number, and thus this is well suited for Web purchases where no physical card needs to be handled. The VA's optimization decision needs computing power, and therefore online purchases that use computers in the first place are a good fit for the divisible card payment infrastructure. It is also expected that this infrastructure will be effective in the emerging mobile commerce domain. The VA residing at the customer's mobile device, for example, may assist the customer's decisions at runtime.

The increased use of credit cards on the Internet has brought increased credit card fraud. Thus, the majority of research on credit card payments for e-commerce focuses on the security issues (Shankar *et al.* 2001). One study relevant to this work is the payment with single-use credit card numbers (Rubin and Wright 2001). In order to reduce the fraud with the permanent card numbers, the card issuing banks, such as American Express, Discover, and MBNA, may issue a one-time use credit card number instead. For example, American Express' Private Payments program allows consumers to obtain single-use numbers from American Express directly to be used for purchases. A card number expires after a purchase is made or after approximately 30 days from the date of issue. Although the one-time use credit card number is primarily designed for protecting against card fraud, it is applicable to this divisible card payment. When generating a virtual card, the Virtual Agent may use this method to create the one-time use virtual-card number.

There have been studies on divisible e-cash payment protocols (Chan et al. 1998, Nakanish *et al.* 2000). These studies focus on payment solutions that ensure anonymity and unlinkability while allowing electronic coins to be divisible. That is, each coin can be spent incrementally as long as the total purchases are limited to the monetary value of the coin. These works look at multiple purchases and multiple merchants, while our work is about one purchase with one merchant but with multiple

credit card issuers. The solutions devised for the divisible e-cash therefore are not directly transferable.

Most of studies on credit card payment security do not focus on the credit card user's practical decision-making problem. Users may face a complex utility optimization problem on each purchase, namely, which card would be the best one to use among multiple cards for this particular purchase. The user's perspective of credit card uses and payments based on her preferences or goals, however, has not been addressed in the literature. The security and protection against fraud are of paramount importance, but as technologies advance, capturing the user's preferences and goals and customizing the use of credit cards should also be an important issue in the electronic payment system.

Part 2: Project Overview

The mission of the project is to provide the customers with a better way of managing their credit cards while minimally modifying the existing infrastructure. As the customers are better off if they can use a combination of cards for a single purchase, the new infrastructure allows the customers to use a combination of different credit cards for a purchase, i.e. divisible card payment.

To support the divisible card payments, two modifications are made to the existing infrastructure. First, a software agent called *Virtual Card Agent* (VA) is added to the client side. The VA recommends to the customer an optimal combination of credit cards to use. If the customer accepts its suggestion, the VA generates the *Virtual card* (*V-card* in short). As the V-card is used online, no physical card needs to be generated. Instead, the VA generates a unique card number, the amount in the card, and the divisible card billing information.

The V-card number is unique to prevent double spending. Rubin and Wright (Rubin and Wright 2001) discuss a method to generate a unique token off-line for limited-use credit cards, such as gift cards or calling cards, using cryptographic methods. This method can be adopted for generating a unique online V-card number. At present, however, a simpler method is used to generate a unique V-card number, where the V-number is based on the combination of credit card numbers used in the V-card. The V-card number is generated using the first two card numbers with the current timestamp. As each card number is unique to each individual, this simple method is sufficient to guarantee the unique V-card number.

When determining the optimal combination of cards to use, the VA may consider the customer's preferences over various factors such as interest rates, annual fees, mileage bonus, cash-back bonus, ongoing promotions, etc. The VA provides the GUI to the customer so that she can easily update her preference profile. Second, special software called a *Virtual Card Manager* (VCM) is added at the merchant side to

handle the V-card payment. When the V-card is up for approval, the VCM decrypts the divisible payment information and forwards the billing information to each card issuer involved in the V-card. Unlike the current protocol that contacts one credit card issuer for approval, the VCM needs to communicate with all the issuing banks involved in a V-card. Each card-issuing bank sends the approval code. When all the approval codes are sent back to VCM, VCM sends back the approval of the V-card to the payment gateway.

Part 3: System Architecture

During the standard transactions that do not use the V-card, the existing infrastructure and protocol can be used without any modification. When the V-card is used, the payment process will be as the shown in Figure 1.

- The online customer finds the desired product from the merchant's Web site. The VA makes a suggestion of which combination of cards to use. If the customer accepts the suggestion, the VA issues the V-card number and enters the V-card information on the secure Web page on the merchant's Web site. If there is no secure Web page on the merchant site, the customer is directed to the merchant's secure payment gateway where the V-card billing information is to be entered.
- The V-card information is passed to the payment gateway.
- The V-card billing information is transferred to the VCM of the merchant's account.
- The VCM transfers the billing information to each credit card issuing bank that is contained in the V-card for approval. Each issuing bank checks if the credit card information is valid and sees if the credit card has sufficient funds. If so, it sets aside the amount of purchase for the merchant.
- Each issuing bank of the V-card sends back the approval (or denial) code to the merchant's VCM. The VCM waits until all pertinent card issuers have sent back their approval (or denial) codes.
- When all card issuers in V-card have sent back the approval codes, the VCM generates an approval code for the V-card, and forwards the code to the payment gateway.
- The approval code is passed to the customer. The payment gateway emails the customer a payment receipt. The VA adjusts the credit card balances resulting from the current purchase with the V-card.

- At the end of the day, the merchant requests to settle all the transactions of the day. The merchant account sends the request to capture funds to the acquiring bank.
- The acquiring bank forwards the request to the issuing banks.
- The card issuing banks pay funds to the acquiring bank and the funds are deposited to the merchant's bank account. The actual funds reach the merchant's checking account in approximately two business days.
- If any one of the issuing banks does not approve the billing request, the V-card transaction should be considered denied, and any approved requests should be nullified.

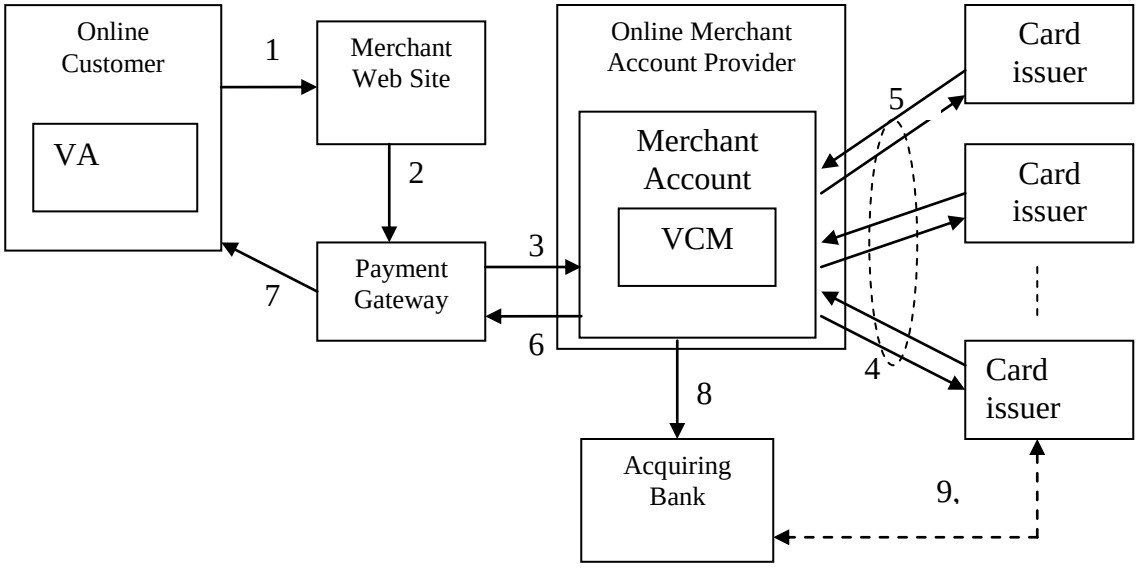

Figure 1: System Architecture of Virtual Card Payment Infrastructure (AMCIS 2004).

Part 4: Virtual Agent and Bank Simulation using JSPs

4.1 Why Java Server Pages (JSPs)?

The Virtual Agent is the most important module of the project. It is a software agent that is added to the client side. The VA recommends to the customer an optimal combination of credit cards to use. If the customer accepts its suggestion, the VA generates the *Virtual card* (*V-card* in short). As the V-card is used online, no physical card needs to be generated. Instead, the VA generates a unique card number, the amount in the card, and the divisible card billing information.

When determining the optimal combination of cards to use, the VA may consider the customer's preferences concerning various factors such as interest rates, annual fees, mileage bonus, cash-back bonus, ongoing promotions, etc. For instance, if a department store may have special discounts on the desired product for using its department store credit card, and if the customer prefers to get discounts, the VA may generate a V-card that includes the department store credit card. The VA provides the GUI to the customer so that he/she can easily update his/her preference profile.

JSP can maintain state on the server between requests. JSP spawns a new thread for each request. JSP does not have to be loaded each time, once it has been initiated. Also JavaScript which is embedded in JSP is used for the client side validations and to pass the control from one JSP to another JSP. JSP is the dynamic Web page which takes the dynamic requests and behaves accordingly. It also contains objects of the Java Classes that behave exactly in the same way they are defined in the Java Classes leading to code re-usability. Using the JSPs, Web based applications can be built which are easy to maintain.

4.2 What are Java Server Pages (JSP)?

Java Server Pages are text files (.jsp extension) that combine standard HTML and new scripting tags. JSPs look like HTML, but they get compiled into Java Servlets the first time they are invoked. The resulting Servlet is a combination of the HTML from the JSP file and embedded dynamic content specified by the new tags. That is not to say that JSPs must contain HTML. Some of them will contain only Java Code.

A JSP page is executed by a JSP engine or container, which is installed on a Web server, or on an application server. When the client asks for a JSP resource, the engine wraps that request and delivers it to the JSP engine along with a response object. The JSP processes the request and modifies the response object to incorporate the communication with the client. The JSP container then wraps up the responses from the JSP page and delivers it to the client. It is imperative to keep in mind that the responses are same as the Servlet Response objects. The first time the engine intercepts a request for a JSP, it compiles this translation unit into a class file that implements the Servlet Protocol. In simple words, Java Server Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically generated HTML.

4.3 What are the Advantages of JSP?

- **JSP vs. Active Server Pages (ASP).** ASP is a similar technology from Microsoft. The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other Microsoft specific languages, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.
- **JSP vs. Pure Servlets.** JSP doesn't give you anything that you couldn't in principle do with a Servlet. But it is more convenient to write (and to modify!) regular HTML than to have a zillion "println" statements that generate the HTML. Plus, by separating the look from the content you can put different people

on different tasks: Web page design experts can build the HTML, leaving places for Servlet programmers to insert the dynamic content.

- **JSP vs. Server-Side Includes (SSI).** SSI is a widely supported technology for including externally defined pieces into a static Web page. JSP is better because it lets you use Servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.
- **JSP vs. JavaScript.** JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server side resources like databases, catalogs, pricing information, and the like.
- **JSP vs. Static HTML.** Regular HTML, of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

4.4 What is Tomcat?

The Tomcat server is a Java-based Web Application container that was created to run Servlet and Java Server Pages (JSP) Web applications. It has become the reference implementation for both the Servlet and JSP specifications.

The Java Servlet and JSP specifications are being developed by Sun under the "Java Community Process." Tomcat 5.0.25 implements Servlets 2.3 and JSP 1.2 API Specifications.

4.5 Environment setup

Download a binary distribution of Tomcat from:

http://jakarta.apache.org/site/binindex.cgi. Save this zip file in a directory named tomcat under your afs home directory. Unzip the .zip file downloaded, using the unzip command on UNIX. You can also use winzip on windows if your network drive is mapped. After running the unzip, a directory "jakarta-tomcat-5.0.25" will have been created in the tomcat directory.

Set the following environment variables in the .login file of your afs home directory.

setenv **TOMCAT_HOME ${HOME}**/tomcat/jakarta-tomcat-5.0.25

setenv **CATALINA_HOME ${HOME}**/tomcat/jakarta-tomcat-5.0.25

setenv **CLASSPATH ${CLASSPATH}: {ORACLE_HOME}**/jdbc/lib/ojdbc14.zip

: ${TOMCAT_HOME}/classes **: ${ TOMCAT_HOME}**/lib

setenv **LD_LIRARY_PATH ${LD_LIBRARY_PATH}: ${ORACLE_HOME)**/lib

Here **HOME** is your afs home directory and **ORACLE_HOME** is already defined in the .login file.

To start the Server:

>cd **$TOMCAT_HOME**/bin

>startup.sh

To stop the Server:

>cd **$TOMCAT_HOME**/bin

>shutdown.sh

"**$TOMCAT_HOME**" is the root of the Tomcat installation directory.

Open a browser window and access the link http://afs1.njit.edu:8080

(You have to telnet to afs1 and execute startup.sh there.)

Part 5: Optimization problem of Virtual Agent

The customer's decision problem is to determine the best combination of cards to use for a purchase. This decision problem can be formulated as the following, which shall be denoted as problem P in this paper.

Model parameters

S Set of credit cards;

Y Purchase amount;

C_i Amount of available credit limit on card i;

B_i Amount of balance currently carried on card i;

I_i Interest rate of card i;

P_i Monthly minimum payment of card i;

f_i Bonus feature of card i;

Decision variables

y_i Amount to be charged on card i;

w_i Weight based on preference of card I;

v_i Present value of payment of card i;

PV Present value

U Utility function

$$U = \sum_{i \in S} (w_1 \cdot v_1(PV(B_i + y_i, I_i/12, P_i)) + w_2 \cdot v_2(f_i)) \quad (1)$$

subject to

$$B_i + y_i \le C_i \quad \forall i \in S \quad (2)$$

$$\sum_{i \in S} y_i = Y \quad (3)$$

$$y_i \ge 0 \quad (4)$$

The optimization in equation (1) maximizes the sum of the utilities of the customer. The utility function takes the form of multi-attribute linear utility function, where the weights are assigned by asking users preference questions (see the following section). The utility function in P has two attributes: total money spent (on interest and principal) and bonus feature, such as miles. The balance on card i after the optimization is the sum of the existing balance and any purchase made on card i. So, the first term in P represents the weighted utility of the present value of all the payments made until the balance is paid out, assuming the customer makes the minimum payment of Pi each month. Note that PV(x, rate, pmt) returns the present value of the total monthly payments made until the

principal of x amount is paid, where rate is the interest rate per month and pmt is the monthly payment. The second term represents the weighted utility of the card feature.

The constraint (2) ensures that the money charged on each card should be less than its credit limit. The constraint (3) ensures that all money spent for a purchase adds up to the purchase amount.

As the customer may use credit cards without the VA (e.g., use them in physical stores), the information regarding the balance and credit limit of each card (Bi and Ci) needs to be updated before performing the optimization. At present, the VA is assumed to know the most up-to-date information, but will have the capability of receiving such up-to-date information from credit card companies. Note that commercial personal financial softwares, such as Quicken and Money, have already employed this capability.

A simple optimization model is used in the prototype system, and we are currently extending its functionality to include more complex decisions, including balance transfers between cards. The VA incorporates the decision-making capability using our optimization model, and thus removes the burden of choosing and combining cards from the customers.

Part 6: More robust system

The Virtual Agent software developed using JSP is fast and robust.

The first login screen appears as shown in Figure 1.

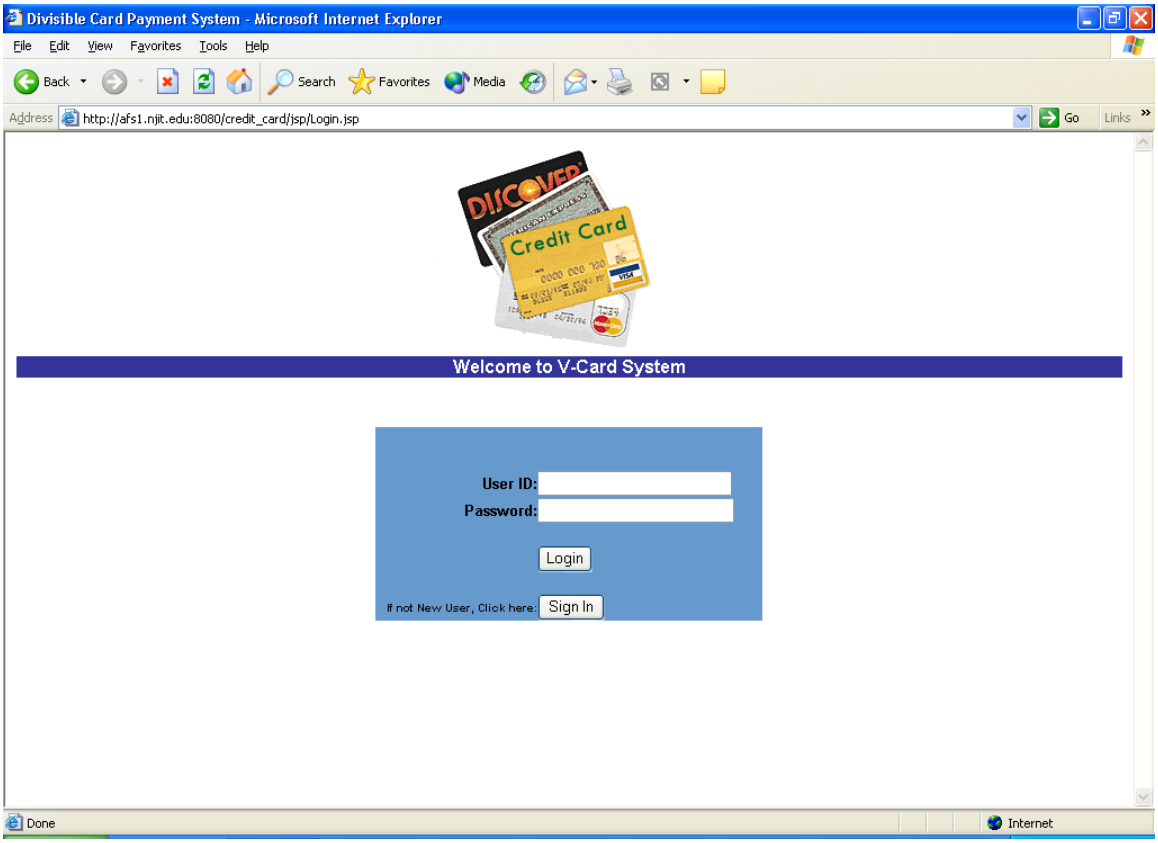

Figure 1: Login to VA

The first page shows the Login to VA. If the user is not identified by the system, then the user can register by clicking on the 'New User' button.

The registration screen appears as shown in the Figure 2.

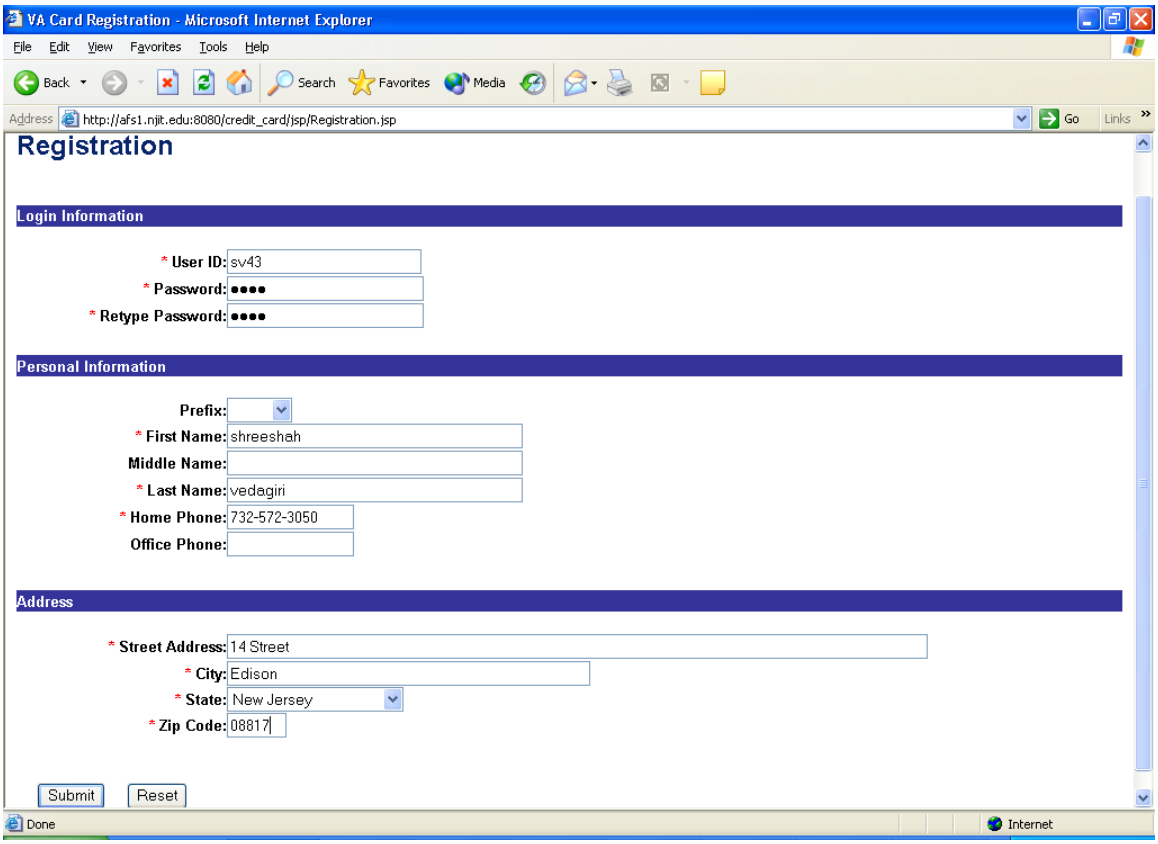

Figure 2: Registration to VA

When the ‘New User’ is clicked, the screen appears as shown in Figure 2 where the user can enter all the details and submit the form.

The VA menu screen appears as shown in the Figure 3.

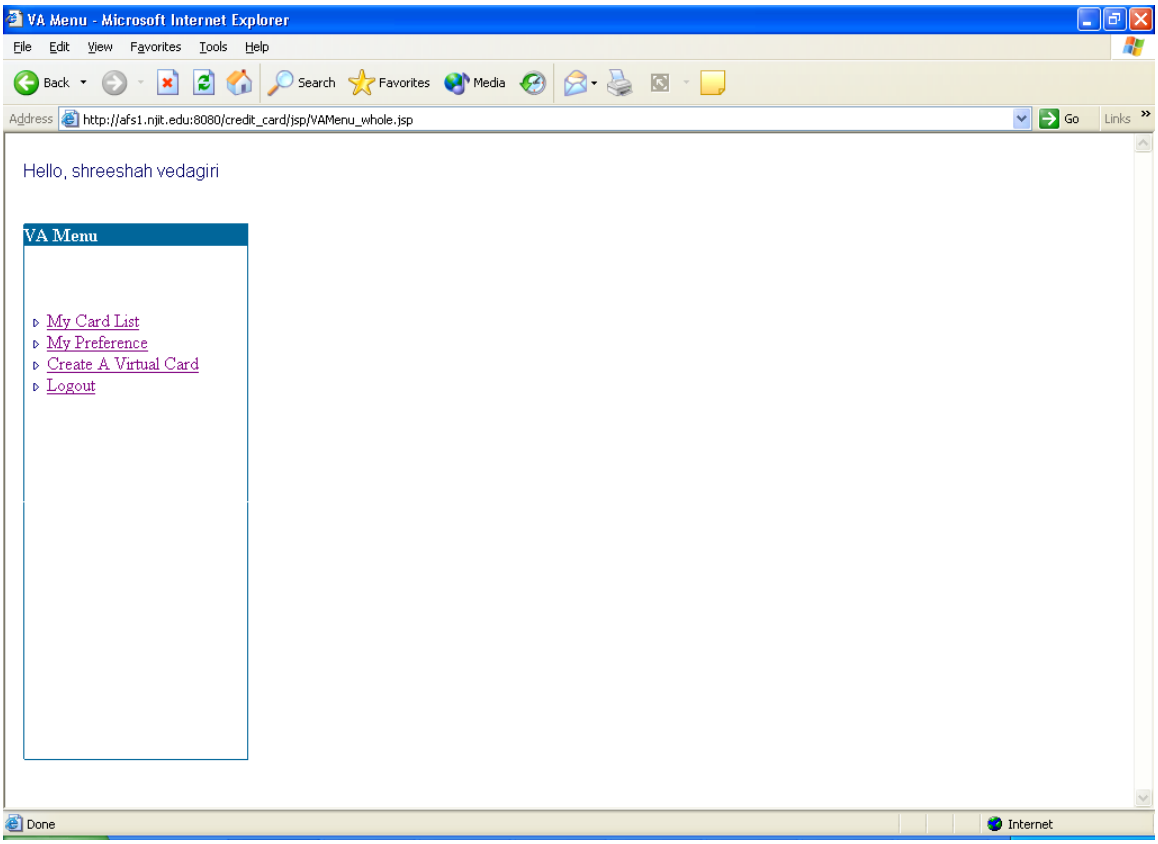

Figure 3: VA Menu

Figure 3 shows the main menu of the V-card system when the customer logs into the VA. The initial window consists of three sub-menus: ‘my card list,’ ‘my preference,’ and ‘create a V-card.’

The My Card List screen appears as shown in Figure 4.

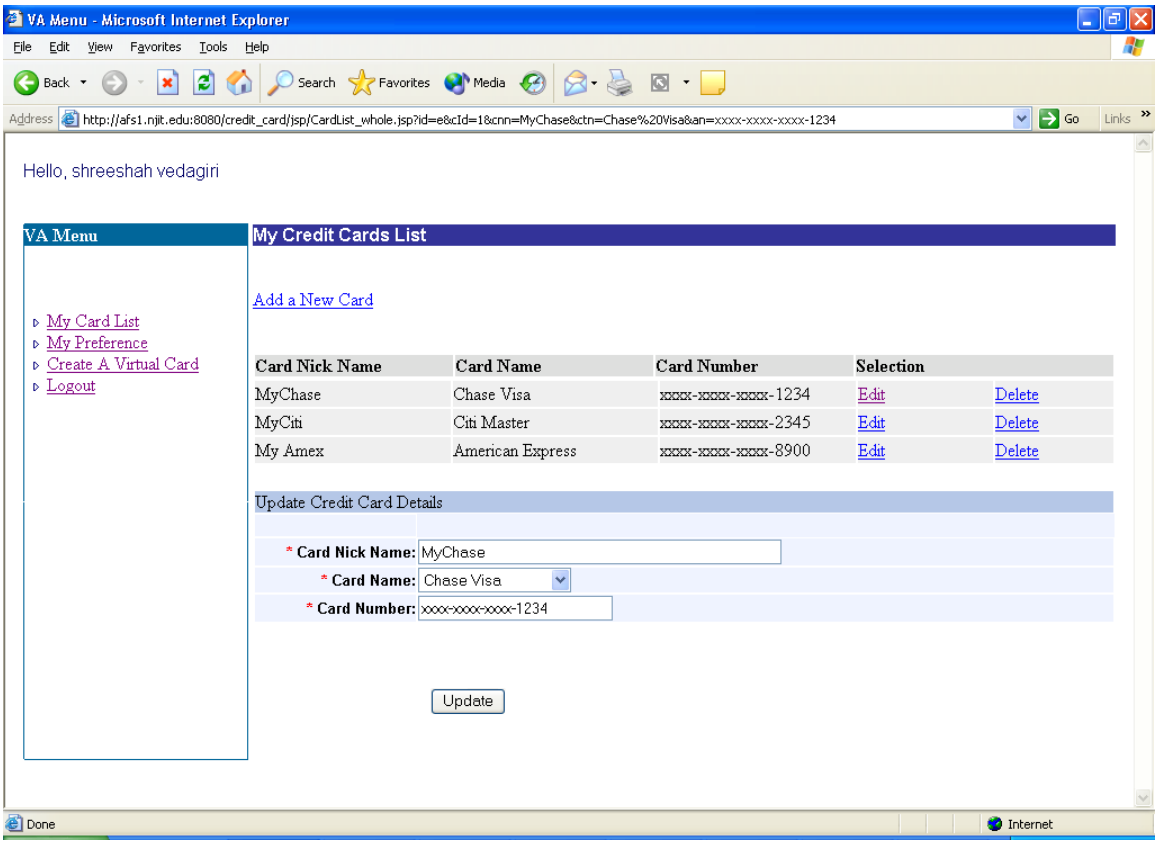

Figure 4: My Card List of VA

When clicking 'My Card List' the customer sees the list of his cards, as shown in Figure 4. The form displays the card nickname and its card number. Clicking the card nickname reveals more detailed information about that card. The interface also provides two functions to edit and delete the existing card information in the list.

The screen to “Add a New Card” appears as shown in the Figure 5.

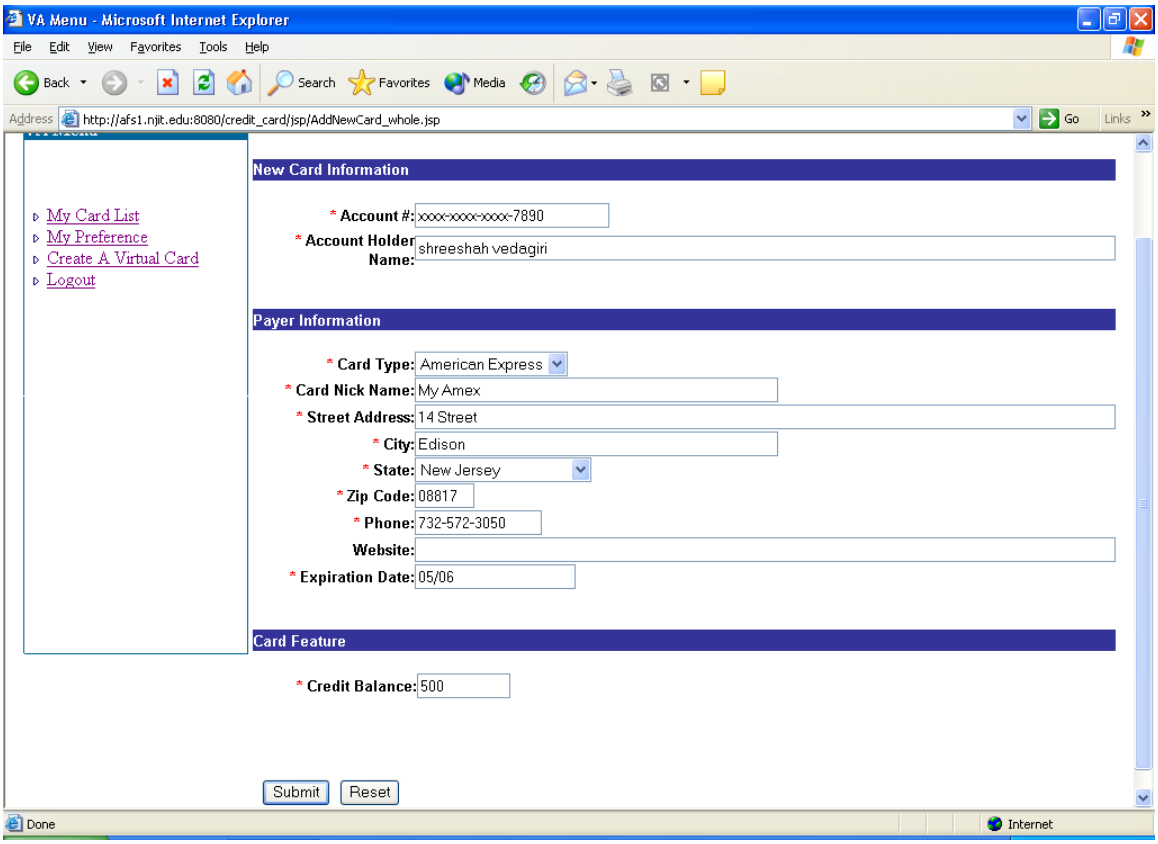

Figure 5: Add a New Card of VA

The customer may add an additional card by pressing the ‘Add a New Card’ link. Figure 5 shows the screen for capturing the new card information. In the prototype, the VA receives the information regarding the detailed features of the card from the customer, but we are exploring a way to automate the acquisition of such information. Once a new card is added, the VA can access the Website of that card to automatically receive the existing balance information (and therefore the available credit limit). This feature of automatically receiving the existing transactions has already been used in commercial personal finance software, such as Quicken and Money.

The My Preferences screen appears as shown in the Figure 6.

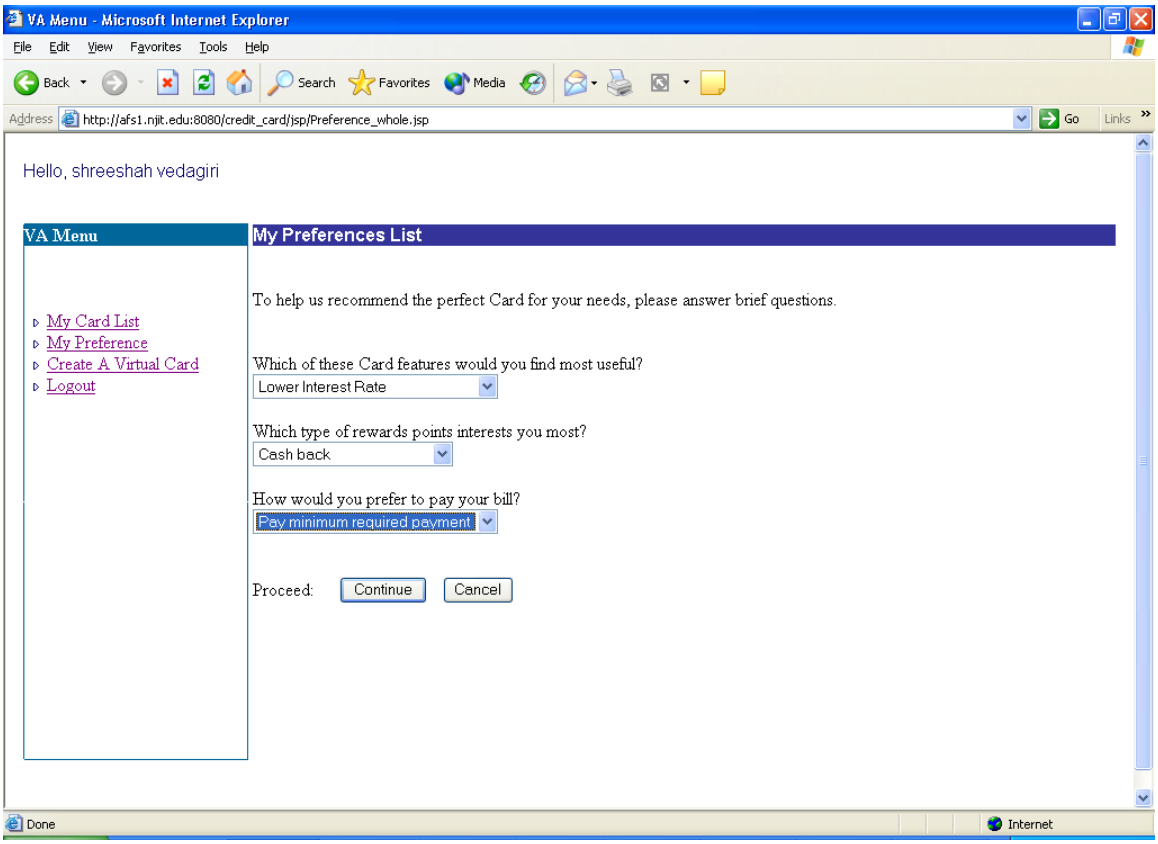

Figure 6: My Preference of VA

Figure 6 shows the window for capturing the user's preferences. At present, the utility function is computed by approximation based on a series of simple questions.

The Create a Virtual Card screen appears as shown in Figure 7.

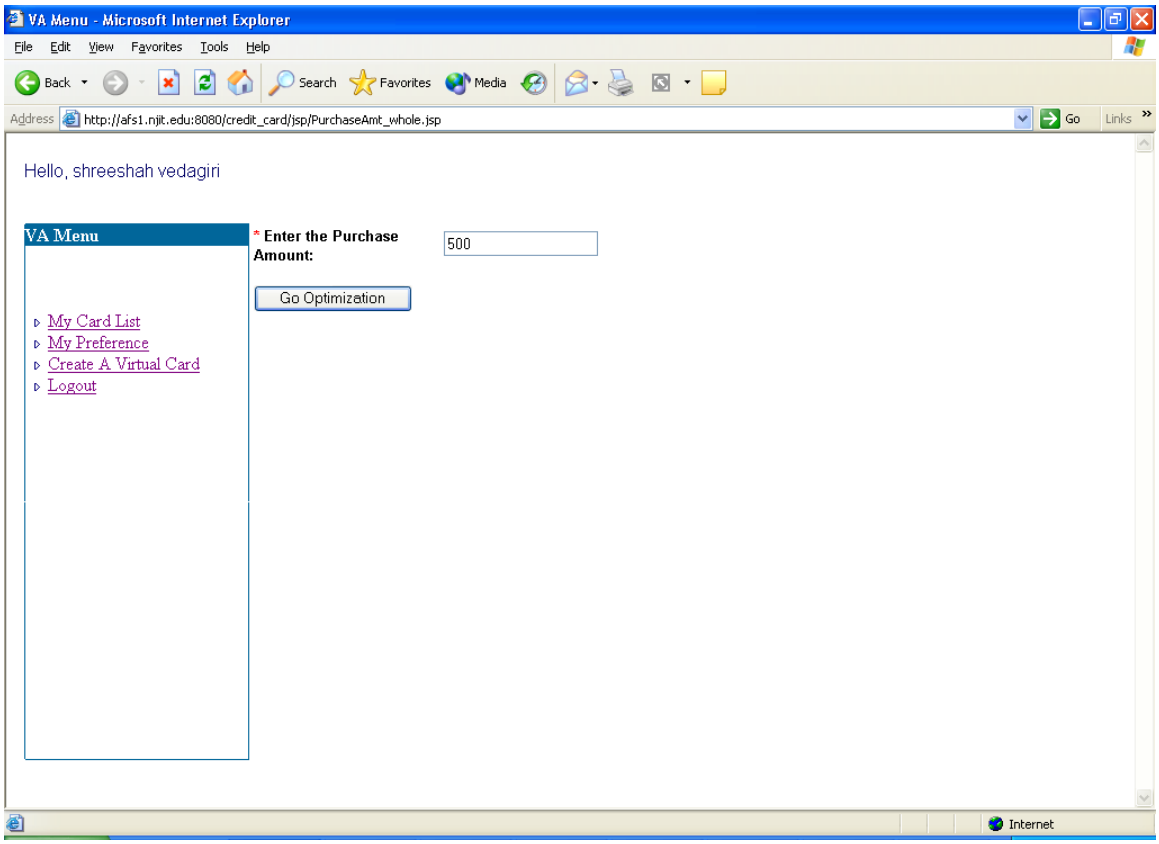

Figure 7: Screen shot for entering Purchase Amount

Figure 7 shows the process of creating a V-card. When the customer wants to purchase a product, she enters the purchase amount and pushes the 'Go Optimization' button. At present, the customer enters the purchase amount, but the VA should be able to receive this information automatically from the merchant site in the future. Then, the VA performs the optimization. In Figure 7, for example, the customer enters the purchase amount of $500. This information, in conjunction with the previously entered information about credit cards and preferences, is used to calculate the optimization result.

The result of Optimization appears as shown in the Figure 8.

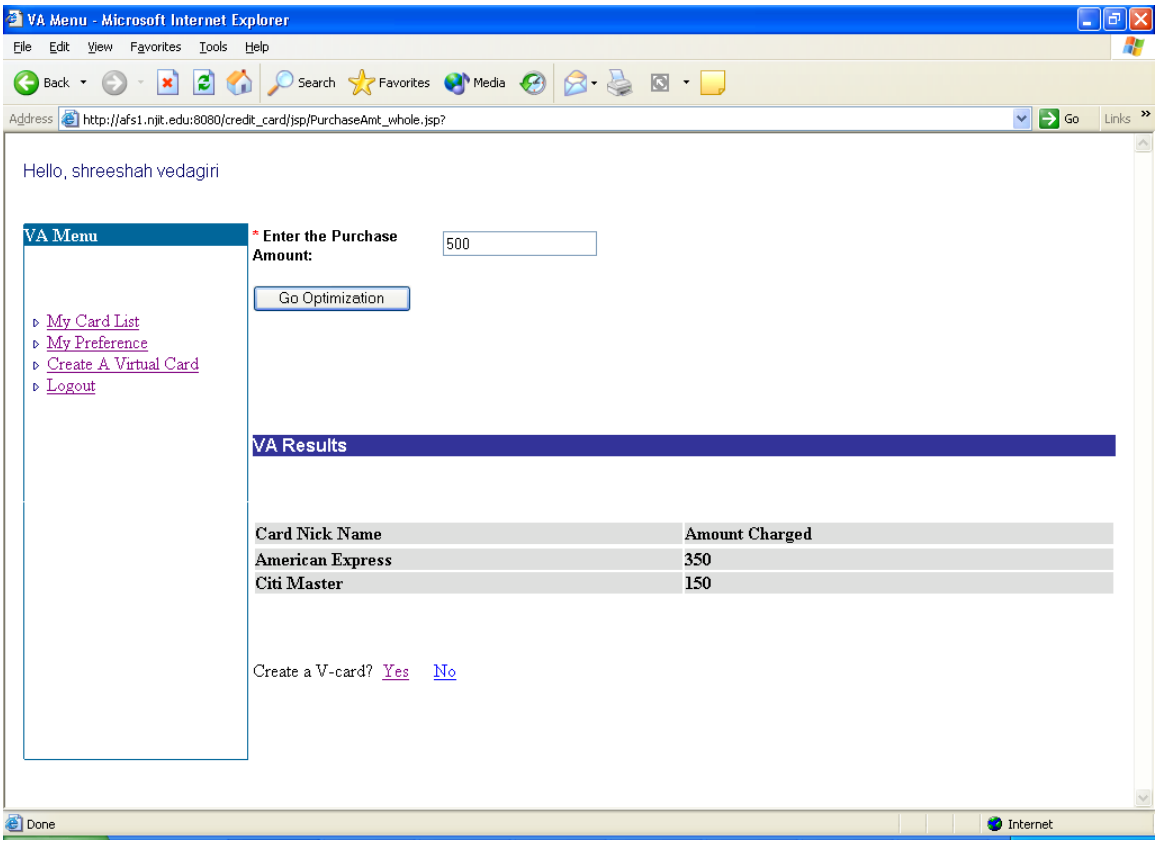

Figure 8: Optimization Results of VA

Figure 8 shows the result of the optimization. The optimized solution shows a list of cards to be used and the charge against each card. The example in Figure 8 shows a list of two cards (out of three cards in the VA database) with their nicknames and the amount charged on each card.

The V-Card screen appears as shown in Figure 9.

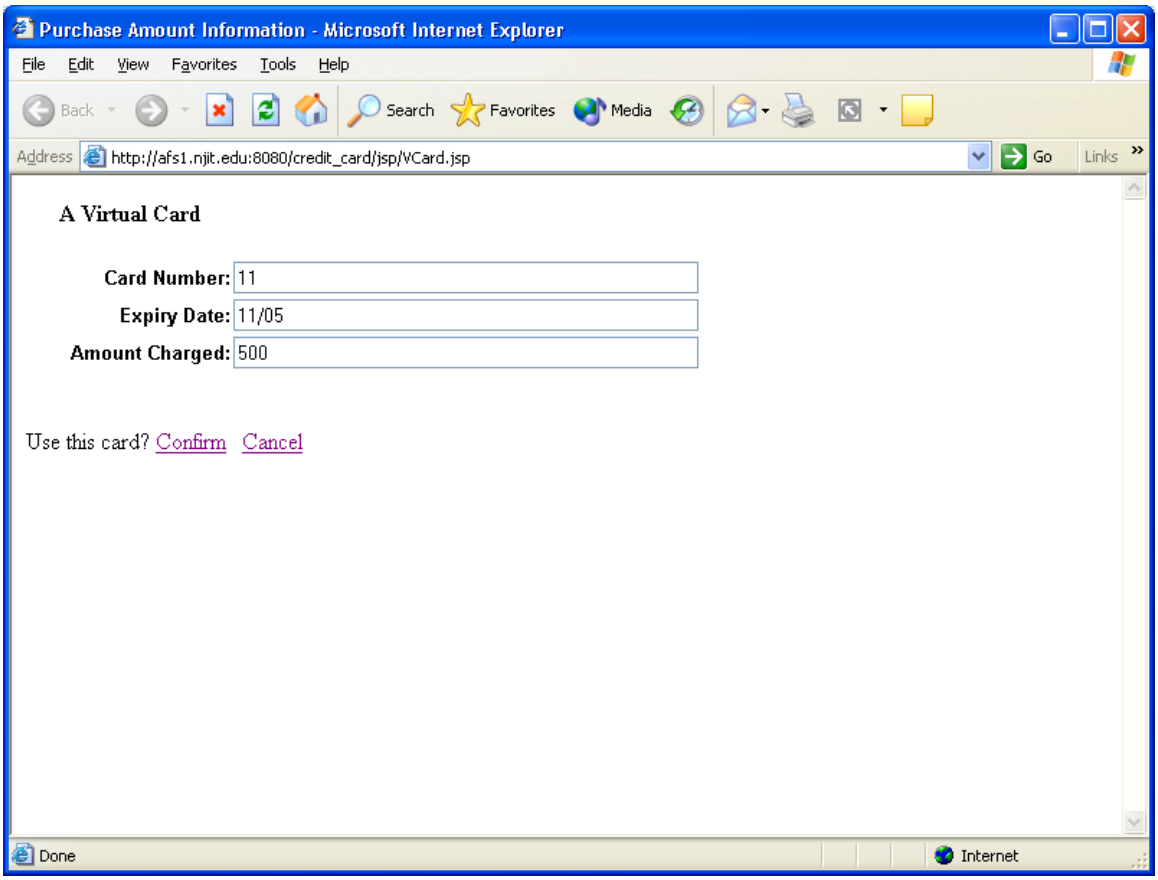

Figure 9: Pop-up window for V-Card

The final step is to create a V-card. When the customer follows the suggestion (by selecting "Yes" to the question of "Create a V-Card" in Figure 8), the VA creates a one-time use V-card number, as shown in Figure 9. The expiration date is set to be the next year from today at present.

The Bank Simulation appears as shown in the Figure 10.

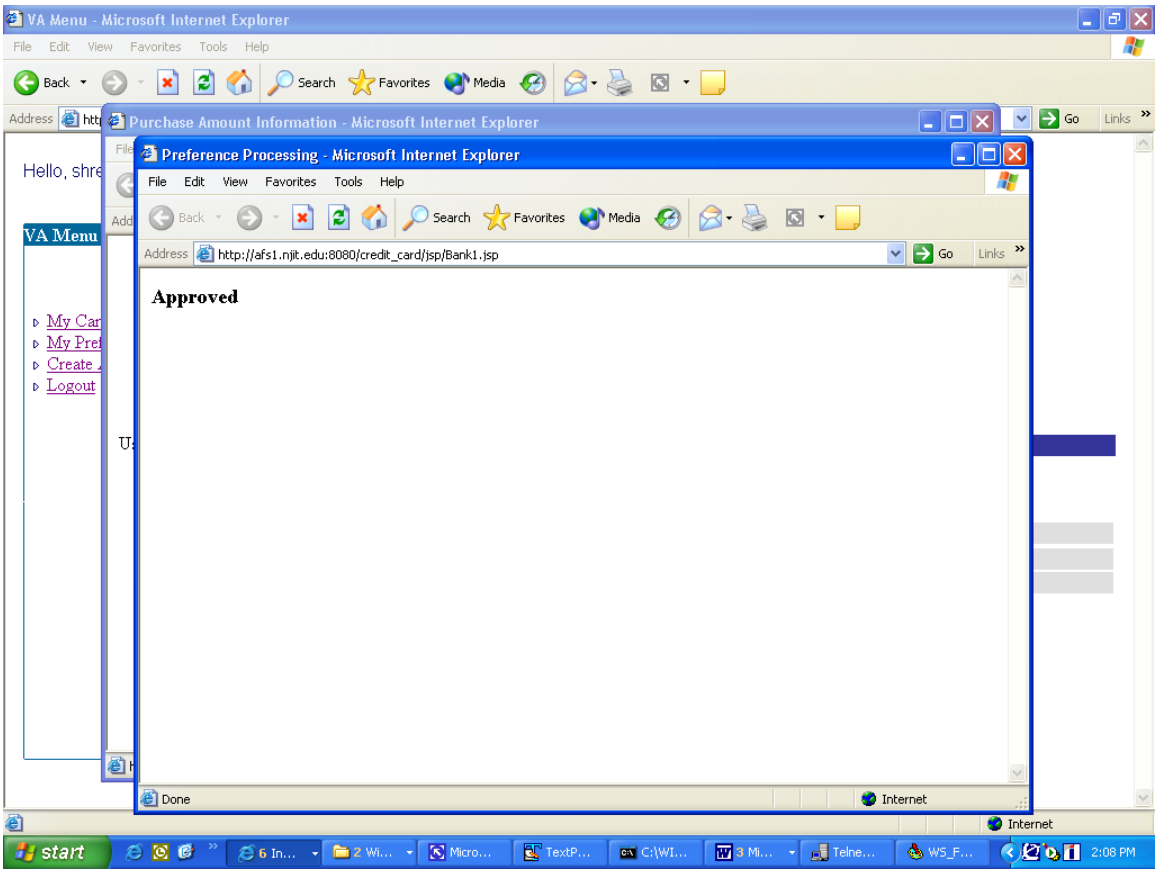

Figure 10: pop-up window for Bank Simulation

When the user clicks on ‘Confirm’ of Figure 9, the V- Card information is sent to the Bank. The Bank Simulation JSP checks with the database and returns the Approval Code. The Approval Code can be ‘Approved’ or ‘Denied’ based on the data and preferences.

Part 7: Future Perspectives

Building a User-Friendly functionality is an evolution process. There are always features and functionalities that can be improved, to provide users with more ease of use and better output. Here are a few suggestions to improve the technology use and user interaction.

1. When using the JDBC, getting a connection and logging in takes the most time. With the use of JDBC 2.0, there is a facility to cache connections. Cached connections are kept in a runtime pool and can be used and reused as needed by the application. The functionality is called Connection Pooling and implementing Connection Pooling will help efficient database connectivity. The reference site is http://java.sun.com/developer/onlineTraining/Database/JDBC20Intro/JDBC20.html
2. Currently Java Code is implemented in JSPs. If the Struts technology is used, it offers many benefits to the Web application developer, including the Model-View-Controller (MVC) design patterns in Web applications, also called Model 2. The Model 2 paradigm applied to Web applications lets us separate display code (for example, HTML and tag libraries) from flow control logic (action classes) and from the data model to be displayed and updated by the application. It also improves the efficiency and speed of the system. The reference site is http://struts.apache.org/
3. The system is currently using three credit cards and the optimization algorithm that is used to generate V-Card Results is not generalized. If the algorithm is extended to take into account multiple parameters that would increase the efficiency of the system.
4. At present, the utility function is computed by approximation based on a series of simple questions, but in future, it can be implemented by the process of adopting the method of "incremental" utility elicitation. In particular, we are using the imprecisely specified multi-attribute utility theory **(**ISMAUT), where the incompletely specified utility function is captured through a series of "comparison" questions (White *et al.* 1984). For example, the customer may be

asked to answer which one she prefers between "paying x_1 dollars while receiving y_1 bonus miles" and "paying x_2 dollars while receiving y_2 bonus miles." Depending on the customer's answer, the weight of attribute 'money' and the weight of attribute 'bonus miles' will be refined, and a new question that may further refine the weights would be asked. A series of such comparison questions will elicit preferences for optimization.

Part 8: Conclusions

I have implemented the Virtual Agent Software and Bank Simulation using the Java Server Pages technology. I have also implemented debugging and error logs for ease of maintenance. For each JSP, the call flow will be logged in the file whose file name is same as the JSP file name. Also if an error gets generated, it will be logged into error file of that day's date for ease. The database used is Oracle to store all the tables pertaining to this application. I also implemented Client Side Validations using JavaScript which checks simple validations and mandatory fields.

I have also implemented the algorithm to generate optimal combinations of credit cards based on some of the user' preferences in any order as follows:

- Lower interest rates
- Airline miles
- Cash-back

Since we don't have access to the real bank servers during the implementation of the project, I have implemented the bank simulation server. For simplicity, a back-end database is created for each bank. For the card issuing banks, when the banks receive the transaction requests, their simulated servers will check in their databases to validate the transactions. After validations, each of the issuing banks will return either a denial message or an approval code back to VCM. For the merchant's Acquiring Bank, it behaves similar to the card issuing banks, except for sending the request to the issuing banks and then collecting the funds from each of them and updating its own database.

Part 9: References

1. Bohle, K. “Integration of Electronic Payment Systems into B2C Internet-Commerce”. IPTS, April, 2002.
2. Chan A., Frankel, Y. and Tsiounis, Y. “Easy Come Easy Go Divisible Cash”, Proceedings of Eurocrypt '98 (Lecture Notes in Computer Science). Springer-Verlag, 1998. Also available at http://www.ccs.neu.edu/home/yiannis/pubs.html.
3. Chan, H., Dillion, T., Lee, R., and Chang, E. E-Commerce: Fundamentals and Applications. Wiley, 2002.
4. Jakobsson, M., Mraihi, D., Tsiounis, Y. and Yung, M. “Electronic Payments: Where Do We Go From Here?” Invited talk, CQRE Secure Congress & Exhibition , CQRE '99, Duesseldorf, Germany, 1999.
5. Jewson, R., “E-Payments: Credit Cards on the Internet”. Aconite white paper, Aconite, October 2001.
6. Lawrence, E., Newton, S., Corbitt, B., Braithwaite, R., and Parker, C. Technology of Internet Business. Wiley, 2002.
7. Lynch, N., Merritt, M., Weihl, W., and Fekete. A. Atomic Transactions. Morgan Kaufmann, San Mateo, 1994.
8. Nakanishi, T., and Sugiyama, Y. “Unlinkable Divisible Electronic Cash,” Proceedings of Third International Information Security Workshop, ISW 2000, Australia, 2000, Lecture Notes in Computer Science 1975, pp. 121-134, Springer, 2000.
9. O'Mahony, D., Peirce, M., Tewari, H., and O'Mahony, D. Electronic Payment Systems for E-Commerce. Artech House, 2001.
10. Rubin, A.D., and Wright, R.N. “Off-Line Generation of Limited-Use Credit Card Numbers.” Financial Cryptography, pp. 196-209, 2001.
11. Smith, S., “Internet Payments: Momentum or Muddle”. *Journal of Internet Banking and Commerce*, 1998.

12. Shankar, U., Walker, M. "A Survey of Security in Online Credit Card Payments," *UC Berkeley Class Notes,* May, 2001.
13. Tygar, J. D. "Atomicity versus Anonymity: Distributed Transactions for Electronic Commerce," *Proceedings of the 24th VLDB Conference,* New York, USA 1998.
14. Walsh, B. "Understanding Internet Payment Protocols". *Network Computing,* May 1999.
15. White, Chelsea C., Andrew P. Sage, and Shigeru Dozono. "A model of multi-attribute decision-making and tradeoff weight determination under uncertainty". *IEEE Transactions on Systems, Man and Cybernetics,* 14(2):223--229, 1984.
16. Soon Ae Chun "An Infrastructure for Customizable and Divisible Card Payments for Online Purchases, AMCIS 2004 Submissions to Business Models for The Digital Economy"

Appendix A: User Manual

The list of the files included in the source code

1. MyScript.sql has the script of all the tables and the data inserted in the master tables.
2. CreditForm.java is the Java class that defines getters and setters of credit card details.
3. DBConnection.java is the Java class that defines the Database connection
4. DebugLog.java is the Java class that defines the information of the debugging and logging information.
5. ErrorFound.java is the Java class that defines the information to log the error.
6. FeatureForm.java is the Java class that defines getters and setters of the credit card feature details
7. VAResultForm.java is the Java class that defines getters and setters of the VA Result Details.
8. AddCard.jsp is the JSP file that collects new credit card details from the user.
9. AddCardAction.jsp is the JSP file that gets information from the AddCard.jsp and stores that data in the database.
10. AddNewCard_whole.jsp is the JSP file that defines include files of AddCardAction.jsp and right_links.jsp
11. Bank1.jsp is the JSP file that is the simulation of the bank server
12. CardList.jsp is the JSP file that defines information of the existing credit cards of a particular user where the user can edit or delete the existing credit card details.
13. CardList_whole.jsp is the JSP file that defines include files of CardList.jsp and right_links.jsp
14. CardListAction.jsp is the JSP file that gets the information from CardList.jsp and updates or deletes that information to / from the database.
15. Login.jsp is the JSP file that collects login information from the user.
16. LoginAction.jsp is the JSP file that validates the login information based on the information stored in the database.

17. Logout.jsp is the JSP file that logs out the user from the system and reinitializes to Login screen.
18. Preference.jsp is the JSP file that collects user preferences from the user.
19. Preference_whole.jsp is the JSP file that defines include files of Preference.jsp and right_links.jsp
20. PreferenceAction.jsp is the JSP file that gets the information from the Preference.jsp and stores that information in the database.
21. PurchaseAmt.jsp collects purchase amount details from the user and also has the Optimization button so that the user can get the optimized results by clicking that button.
22. PurchaseAction.jsp is the JSP file that gets the information from the PurchaseAmt.jsp and stores that information in the database. It also has the optimization algorithm, stores the v-card results in the database and sends information to the bank database.
23. PurchaseAmt_whole.jsp is the JSP file that defines include files of PurchaseAmt.jsp and right_links.jsp
24. Registration.jsp is the JSP file that collects registration information from the user.
25. RegistrationAction.jsp is the JSP file that gets the information from the Registration.jsp file and stores that information in the database.
26. right_links.jsp is the JSP file that defines VA Menu
27. VAMenu_whole.jsp is the JSP file that defines include file of right_links.jsp
28. VCard.jsp is the JSP file that pops up showing the V-card information to the user.

The path of the Source Code is /afs/cad/research/p/2/Creditcard/Shreeshah /credit_card.zip

Set up information

1. Install the Tomcat Web server into the AFS home directory as described in the Section 4.5
2. Unzip the source code and place it under the webapps of the Tomcat Web server.

3. The directory structure for the credit_card application under the webapps folder is as follows:
    - Home directory of application is credit_card. It's path is webapps/credit_card
    - Under the credit_card directory, the following directories exists:
    - The documents directory that has conference paper, proposal, project report
    - errorlogs directory that has errorlogs and debug logs created by the application when the application is running
    - script directory that has Myscript.sql script file
    - jsp directory that has all JSPs and images. The list of JSPs is
        - AddCard.jsp
        - AddCardAction.jsp
        - AddNewCard_whole.jsp
        - Bank1.jsp
        - CardList.jsp
        - CardList_whole.jsp
        - CardListAction.jsp
        - Login.jsp
        - LoginAction.jsp
        - Logout.jsp
        - Preference.jsp
        - Preference_whole.jsp
        - PreferenceAction.jsp
        - PurchaseAction.jsp
        - PurchaseAmt.jsp
        - PurchaseAmt_whole.jsp
        - Registration.jsp
        - RegistrationAction.jsp
        - right_links.jsp
        - VAMenu.jsp
        - VAMenu_whole.jsp
    - src directory that has Java source files. The list of Java source files is:

    - CreditForm.java
    - DBConnection.java
    - DebugLog.java
    - ErrorFound.java
    - FeatureForm.java
    - VAResultForm.java
- WEB-INF directory that has classes and lib directories. It also has web.xml file which is like index file
- In the classes directory, there is common directory which contains compiled classes of the Java classes that are there in src directory.
- Lib directory has the ojdbc14.jar file that supports the JDBC connection.

Compilation Instructions:

- To compile Java classes, use the command javac classname.java and put the executable java class code under WEB-INF/classes/common directory.
- JSP's need not be compiled. They should be placed under/jsp directory and the tomcat server should be started

Appendix B: Source Code

DBConnection.java

```
package common;

import java.sql.*;
import javax.sql.*;

//code for establishing database connection


public class DBConnection {

  Connection conn = null;

  public DBConnection() {

                  DebugLog debug = new DebugLog("DBConnection.txt");

        if (conn == null )
          {
            try {
              debug.append("\n In DBConnection");

              //driver
              DriverManager.registerDriver(new
oracle.jdbc.driver.OracleDriver());


                    //url
                    String url =
"jdbc:oracle:thin:@seer.njit.edu:1521:research";

                    try {
                                String url1 =
System.getProperty("JDBC_URL");

                             if (url1 != null)
                               url = url1;
                           } catch (Exception e) {
                                // If there is any security exception,
ignore it and use the default
                                ErrorFound ef = new ErrorFound();
                                ef.appendError("DBConnection",
"DBConnection constructor", e.getMessage());
                                ef.close();
                    }

              debug.append("url is:"+url);

              //(url, userid, password)
```

```
              conn = DriverManager.getConnection (url, "sv43",
"srisai19");

              debug.append("In DBConnection, connection established");
              debug.close();

            }
            catch (Exception e) {
                        ErrorFound ef = new ErrorFound();
              ef.appendError("DBConnection", "DBConnection constructor
", e.getMessage());
              ef.close();
              conn = null;
            }
          }
     }

                //method to get the connection
      public Connection getConnection() {
          return conn;
      }

                //method to close the connection
      public void closeConnection() {
        try{
          conn.close();
        }catch (Exception e) {
                    ErrorFound ef = new ErrorFound();
          ef.appendError("DBConnection", "closeConnection Method",
e.getMessage());
          ef.close();
          conn = null;
        }
      }

}
```

DebugLog.java

```
package common;
import java.io.*;
import java.util.*;
import java.lang.*;
import java.text.*;


//code for logging the debug details
//creates new file if mentioned as a parameter

public class DebugLog {


  String ErrorFoundFile =
//"/afs/cad.njit.edu/u/s/v/sv43/public_html/jakarta-tomcat-
5.0.25/webapps/credit_card/errorlogs/";
                                                    "C:/Program
Files/Apache Software Foundation/Tomcat
5.0/webapps/credit_card/errorlogs/";

  PrintStream pout = null;

  public DebugLog(){
    try{
      File errorFile = new File( ErrorFoundFile+"err_Debug.txt" );
      if( errorFile.exists() ){
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }else {
        BufferedWriter bw = new BufferedWriter ( new FileWriter(
errorFile ) );
        bw.write("");
        bw.flush();
        bw.close();
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }

      pout.println("---New Entry-------------------------");

    }catch( FileNotFoundException e ){

    }catch( IOException e ){

    }
  }


  /**
   * Constructor for a Debugging Log
   * @param overwrite   true : to erase previous contents
   *                    false: same as if no param specified
   */
  public DebugLog( boolean overwrite ){

    try{
```

```
      File errorFile = new File(ErrorFoundFile + "err_Debug.txt");
      if (overwrite == true) {
        BufferedWriter bw = new BufferedWriter(new
FileWriter(errorFile));
        bw.write("");
        bw.flush();
        bw.close();
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }else {
        if (errorFile.exists()) {
          pout = new PrintStream(new FileOutputStream(errorFile,
true));
        }
        else {
          BufferedWriter bw = new BufferedWriter(new
FileWriter(errorFile));
          bw.write("");
          bw.flush();
          bw.close();
          pout = new PrintStream(new FileOutputStream(errorFile,
true));
        }


      }

      pout.println("---New Entry-------------------------");

    }
    catch (FileNotFoundException e) {

    }catch( IOException e ){

    }
  }


  public DebugLog(String fileName){
    try{
      File errorFile = new File( ErrorFoundFile+fileName );
      if( errorFile.exists() ){
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }else {
        BufferedWriter bw = new BufferedWriter ( new FileWriter(
errorFile ) );
        bw.write("");
        bw.flush();
        bw.close();
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }

      pout.println("---New Entry-------------------------");

    }catch( FileNotFoundException e ){

    }catch( IOException e ){
```

```
    }
  }

  public DebugLog(String fileName, boolean overwrite) {
    try {
      File errorFile = new File(ErrorFoundFile + fileName);
      if (overwrite == true) {
        BufferedWriter bw = new BufferedWriter(new
FileWriter(errorFile));
        bw.write("");
        bw.flush();
        bw.close();
        pout = new PrintStream(new FileOutputStream(errorFile, true));

      }
      else {
        if (errorFile.exists()) {
          pout = new PrintStream(new FileOutputStream(errorFile,
true));
        }
        else {
          BufferedWriter bw = new BufferedWriter(new
FileWriter(errorFile));
          bw.write("");
          bw.flush();
          bw.close();
          pout = new PrintStream(new FileOutputStream(errorFile,
true));
        }
      }
      pout.println("---New Entry-------------------------");

    }
    catch (FileNotFoundException e) {

    }
    catch (IOException e) {

    }
  }



  public void append( String debugMessage){
    pout.print( this.getDateTime());
    if( debugMessage != null )
      pout.println(debugMessage);
    else
      pout.print("NULL");
  }

  private String getDateTime(){

    Date today = new Date();

    DateFormat formatter
=DateFormat.getDateInstance(DateFormat.MEDIUM);
```

```
    String date = formatter.format(today);

    DateFormat formatter1 =
DateFormat.getTimeInstance(DateFormat.MEDIUM);

    String time = formatter1.format(today);

    return " ("+date+" : "+time+") ";

  }

  public void close(){
    pout.println("---End Entry----------------------------");

    pout.close();
  }

}
```

ErrorFound.java

```
package common;
import java.io.*;
import java.util.*;
import java.lang.*;
import java.text.*;

//code to log the error when ever application generates error

public class ErrorFound {


  String ErrorFoundFile =
//"/afs/cad.njit.edu/u/s/v/sv43/public_html/jakarta-tomcat-
5.0.25/webapps/credit_card/errorlogs/";
                                                    "C:/Program
Files/Apache Software Foundation/Tomcat
5.0/webapps/credit_card/errorlogs/";

  PrintStream pout = null;


  public ErrorFound(){
    try{
                String fileName = "err_"+this.getDate();
      for(int i=0;i<fileName.length();i++){
                  if (fileName.indexOf(" ")>-1){
                        fileName = fileName.substring(0,
fileName.indexOf(" "))
                                        +
fileName.substring(fileName.indexOf(" ")+1,fileName.length());
                  }
                }
                File errorFile = new File( ErrorFoundFile+ fileName
+".txt" );
      if( errorFile.exists() ){
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }else {
        BufferedWriter bw = new BufferedWriter ( new FileWriter(
errorFile ) );
        bw.write("");
        bw.flush();
        bw.close();
        pout = new PrintStream(new FileOutputStream(errorFile, true));
      }

      pout.println("New Error-------------------------");

    }catch( FileNotFoundException e ){

    }catch( IOException e ){

    }
  }
```

```
  public void appendError( String errorLocation, String errorTitle,
String errorMessage){
    pout.print( "("+this.getDate()+" : " + this.getTime()+")");
    pout.print( errorLocation);
    pout.print( ": " + errorTitle );
    pout.print( ": " );
    pout.println(errorMessage);
  }

  private String getDate(){

    Date today = new Date();

    DateFormat formatter
=DateFormat.getDateInstance(DateFormat.MEDIUM);

    String date = formatter.format(today);

    return date;

  }



  private String getTime(){

    Date today = new Date();

    DateFormat formatter1 =
DateFormat.getTimeInstance(DateFormat.MEDIUM);

    String time = formatter1.format(today);

    return time;

  }

  public void close(){
    pout.println("End Error----------------------------");
    pout.close();
  }

}
```

CreditForm.java

```
package common;

//setters and getters for the credit card details

public class CreditForm {

   private int creditLimit;
   private int creditBalance;
   private float interestRate;
   private int minPayment;

   public int getCreditLimit(){
                return creditLimit;
   }

   public void setCreditLimit(int i){
                creditLimit = i;
   }

   public int getCreditBalance(){
                return creditBalance;
   }

   public void setCreditBalance(int i){
                creditBalance = i;
   }

   public float getInterestRate(){
                return interestRate;
   }

   public void setInterestRate(float f){
                interestRate = f;
   }

   public int getMinPayment(){
                return minPayment;
   }

   public void setMinPayment(int i){
                minPayment = i;
   }

}
```

FeatureForm.java

```
package common;

//setters and getter for credit card features

public class FeatureForm {

   private int cardTypeId;
   private String cardTypeDesc;
   private int cardFeatureId;
   private String cardFeatureDesc;

   public int getCardTypeId(){
                return cardTypeId;
   }

   public void setCardTypeId(int i){
                cardTypeId = i;
   }

   public String getCardTypeDesc(){
                return cardTypeDesc;
   }

   public void setCardTypeDesc(String s){
                cardTypeDesc = s;
   }

   public int getCardFeatureId(){
                return cardFeatureId;
   }

   public void setCardFeatureId(int i){
                cardFeatureId = i;
   }

   public String getCardFeatureDesc(){
                return cardFeatureDesc;
   }

   public void setCardFeatureDesc(String s){
                cardFeatureDesc = s;
   }

}
```

VAResultForm.java

```
package common;

//setters and getters for VAResults

public class VAResultForm {

   private int cardTypeId;
   private String cardTypeDesc;
   private int amountCharged;


   public int getCardTypeId(){
                return cardTypeId;
   }

   public void setCardTypeId(int i){
                cardTypeId = i;
   }

   public String getCardTypeDesc(){
                return cardTypeDesc;
   }

   public void setCardTypeDesc(String s){
                cardTypeDesc = s;
   }

   public int getAmountCharged(){
                return amountCharged;
   }

   public void setAmountCharged(int i){
                amountCharged = i;
   }
}
```

AddCard.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--java script for evaluating the fields-->
<script language="JavaScript">
       function evalAll() {
                      test = true;
                     if((document.all) || (document.getElementById)){
                         var userID = document.regForm.txtUserID.value;
                         var passwd =
document.regForm.txtPassword.value;
                         var passwd_retype =
document.regForm.txtPasswordRetype.value;
                         var firstName =
document.regForm.txtFirstName.value;
                         var lastName =
document.regForm.txtLastName.value;
                         var homePhone =
document.regForm.txtHomePhone.value;
                         var city =document.regForm.txtCity.value;
                         var streetAddress =
document.regForm.txtStreetAddress.value;
                         var myIndex =
document.regForm.cboState.selectedIndex;
                         var selStateValue =
document.regForm.cboState.options[myIndex].value;
                         var zipCode =document.regForm.txtZipCode.value;

                         var cityb =document.regForm.txtbCity.value;
                         var streetAddressb =
document.regForm.txtbStreetAddress.value;
                         var myIndexb =
document.regForm.cbobState.selectedIndex;
                         var selStateValueb =
document.regForm.cbobState.options[myIndexb].value;
                         var zipCodeb
=document.regForm.txtbZipCode.value;
                     }

          if (passwd!=null && passwd_retype!=null){
                         if(passwd!=passwd_retype){
                               alert('Password and Retype Password
should be same');
                               document.regForm.txtPassword.value="";

               document.regForm.txtPasswordRetype.value="";
                               document.regForm.txtPassword.focus();
                               test= false;
                         }
                     }
                     if(userID == ""){
                           alert("User ID cannot be blank");
                document.regForm.txtUserID.focus();
                test = false;
                return test;
                     }else if(passwd == ""){
```

```
                  alert("Password cannot be blank");
            document.regForm.txtPassword.focus();
            test = false;
            return test;
                  }else if(passwd_retype == ""){
                        alert("Password Retype cannot be blank");
            document.regForm.txtPasswordRetype.focus();
            test = false;
            return test;
                  }else if(firstName == ""){
                        alert("First Name cannot be blank");
            document.regForm.txtFirstName.focus();
            test = false;
            return test;
                  }else if(lastName == ""){
                        alert("Last Name cannot be blank");
            document.regForm.txtLastName.focus();
            test = false;
            return test;
                  }else if(homePhone == ""){
                        alert("Home Phone cannot be blank");
            document.regForm.txtHomePhone.focus();
            test = false;
            return test;
                  }else if(streetAddress == ""){
                        alert("Street Address cannot be blank");
            document.regForm.txtStreetAddress.focus();
            test = false;
            return test;
                  }else if(city == ""){
                        alert("City cannot be blank");
            document.regForm.txtCity.focus();
            test = false;
            return test;
                  }else if(selStateValue == ""){
                        alert("State cannot be blank");
            document.regForm.cboState.focus();
            test = false;
            return test;
                  }else if(zipCode == ""){
                        alert("Zip Code cannot be blank");
            document.regForm.txtZipCode.focus();
            test = false;
            return test;
                  }

                  return test;
      }

               function evalAddress(){
                     var city =document.regForm.txtCity.value;
                     var streetAddress =
document.regForm.txtStreetAddress.value;
                     var myIndex =
document.regForm.cboState.selectedIndex;
                     var selStateValue =
document.regForm.cboState.options[myIndex].value;
```

```
                        var zipCode =document.regForm.txtZipCode.value;

                        //alert(document.regForm.chkAddrSame.checked);
                        if (document.regForm.chkAddrSame.checked) {
                              document.regForm.txtbCity.value = city;

                              document.regForm.txtbStreetAddress.value
= streetAddress;
                              document.regForm.cbobState.selectedIndex
=  myIndex;

               document.regForm.cbobState.options[myIndex].value =
selStateValue;
                              document.regForm.txtbZipCode.value =
zipCode;
                        } else {
                              document.regForm.txtbCity.value = "";

                              document.regForm.txtbStreetAddress.value
= "";
                              document.regForm.cbobState.selectedIndex
=  "";

               document.regForm.cbobState.options[myIndex].value = "";
                              document.regForm.txtbZipCode.value = "";
                        }
                  }
                  function validateInt() {
                        var iString = document.regForm.txtZipCode.value
                        alert(iString);
                   return (("" + parseInt(iString)) == iString);
                  }
                  function AllowNumericOnly(){
                Expression = '0123456789';
                var ch = String.fromCharCode(window.event.keyCode);
                ch = ch.toLowerCase();
                var a = Expression.indexOf(ch);
                if (a == -1)
                     window.event.keyCode = 0;
       }//end function


</script>
<html>
<head>
<title>Add a New Card</title>
</head>
<body>
<br>
<h1><font color="#08246B" size="5" face="arial,helvetica">
               Add a New Card</font>
</h1>
<form name="addForm" method="post" action="AddCardAction.jsp">
<br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
```

```
                <td width="19%" align="justify" bgcolor="#333399">
<font color="#FFFFFF" size="2" face="Arial, Helvetica"><b>
                New Card Information</b></font></td>
              <td width="19%" align="justify"
bgcolor="#333399"> </td>
</tr>
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>

<tr>
              <td width="19%" align="right"><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Account #:</b></font></td>
              <td width="81%"><input name="txtAccountNum" type="text"
maxlength="25" size="25"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Account Holder Name:</b></font></td>
              <td width="81%"><input name="txtAccountHolderName"
maxlength="100" size="100"></td>
</tr>
</table>
<br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
              <td width="19%" align="justify" bgcolor="#333399">
                 <font color="#FFFFFF" size="2" face="Arial,
Helvetica"><b>Payer Information</b></font>
              </td>
              <td width="81%" bgcolor="#333399"> </td>
</tr>
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Card Type:</b></font></td>
              <td width="81%"><font color="#000000" size="2"
face="Arial, Helvetica">
                 <select name="cboCardType" size="1">
                   <option selected> </option>
         <%
                      //getting the credit card details
                            DBConnection myConn1 = null;
                      ResultSet rs1 = null;
                            String sqlQuery1 = null;
                      DebugLog debug1 = new DebugLog("AddCard.txt");
```

```
                        try{
                              myConn1 = new DBConnection();
                              Connection conn1 =
myConn1.getConnection();
                              Statement stmt1 =
conn1.createStatement();
                                    sqlQuery1 = "SELECT CARD_TYPE_ID,
CARD_TYPE_NAME FROM CARD_TYPE";
                                    debug1.append("SQL Query: " +
sqlQuery1);
                                    rs1 =
stmt1.executeQuery(sqlQuery1);
                                    while (rs1.next()) {
                                          String cardTypeID =
rs1.getString("CARD_TYPE_ID");
                                          String cardTypeName =
rs1.getString("CARD_TYPE_NAME");
                                          debug1.append("Card Type ID:
" + cardTypeID + "Card Type Name: "  + cardTypeName);
                        %>
                        <option
value="<%=cardTypeID%>"><%=cardTypeName%></option>
                        <%          } //end of while
                                    rs1.close();
                                    stmt1.close();
                                    debug1.close();
                                    myConn1.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("AddNewCard", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
        </select></font>
                  </td>
</tr>
<tr>
                  <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
                  <font color="#000000" size="2" face="Arial,
Helvetica"><b>Card Nick Name:</b></font></td>
                  <td width="81%"><input name="txtCardNickName" type="text"
maxlength="50" size="50"></td>
</tr>
<tr>
                  <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
                  <font color="#000000" size="2" face="Arial,
Helvetica"><b>Street Address:</b></font></td>
                  <td width="81%"><input name="txtStreetAddress"
type="text" maxlength="100" size="100"></td>
</tr>
<tr>
                  <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
```

```
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>City:</b></font></td>
              <td width="81%"><input name="txtCity" type="text"
maxlength="50" size="50"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>State:</b></font></td>
              <td width="81%"><font color="#000000" size="2"
face="Arial, Helvetica">
                  <select name="cboState" size="1">
                    <option selected> </option>
          <%
                              DBConnection myConn = null;
                        ResultSet rs = null;
                              String sqlQuery = null;
                        DebugLog debug = new DebugLog("Reg.txt");
                        try{
                              myConn = new DBConnection();
                              Connection conn = myConn.getConnection();
                              Statement stmt = conn.createStatement();

                                    sqlQuery = "SELECT STATE_ID,
STATE_DESC FROM STATE";
                                    debug.append("SQL Query: " +
sqlQuery);
                                    rs = stmt.executeQuery(sqlQuery);
                                    while (rs.next()) {
                                          String stateID =
rs.getString("STATE_ID");
                                          String stateDesc =
rs.getString("STATE_DESC");
                                          debug.append("State ID: " +
stateID + "State Desc: "  + stateDesc);
                        %>
                        <option
value="<%=stateID%>"><%=stateDesc%></option>
                        <%          } //end of while
                                    rs.close();
                                    stmt.close();
                                    myConn.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("Registration", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
        </select></font>
              </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
```

```
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Zip Code:</b></font></td>
              <td width="81%"><input name="txtZipCode" type="text"
maxlength="5" size="5"  onKeyPress="AllowNumericOnly();"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Phone:</b></font></td>
              <td width="81%"><input name="txtPhone" type="text"
maxlength="15" size="15"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#000000"
size="2" face="Arial, Helvetica"><b>Website:</b></font></td>
              <td width="81%"><input name="txtWebsite" type="text"
maxlength="100" size="100"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Expiration Date:</b></font></td>
              <td width="81%"><input name="txtExpDate" type="text"
maxlength="20" size="20"></td>
</tr>
</table>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
              <td width="19%" align="justify" bgcolor="#333399">
                 <font color="#FFFFFF" size="2" face="Arial,
Helvetica"><b>Card Feature</b></font>
              </td>
              <td width="81%" bgcolor="#333399"> </td>
</tr>
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Credit Balance:</b></font></td>
              <td width="81%"><input name="txtCreditBalance"
type="text" maxlength="10" size="10"></td>
</tr>
</table>
<br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
               <td width="10%" align="center">
                 <input name="pagemode" type="hidden" value="submit">
```

```
                  <input type="submit" onKeyDown='return evalAll();'
value="Submit" onClick='return evalAll();' >
                 </td>
                 <td width="90%"><input type="reset" value="Reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

AddCardAction.jsp

```
<%@ page language="java" import="java.sql.*,common.*"%>
<html>
<head>
<title>Add Card Processing</title>
</head>
<body>
<form name="addCardAction" action="CardList_whole.jsp">
<%
               //action file that gets information from AddNewCard.jsp
               //and stores that information in database
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    DebugLog debug = new DebugLog("AddCardAction.txt");
               int rowsAffected = 0, rowsAffectedBA = 0;
               Statement stmt = null;
               debug.append("Obtained values:");
               String accountNum =
request.getParameter("txtAccountNum");
               String accountHolderName =
request.getParameter("txtAccountHolderName");
               String cardTypeId = request.getParameter("cboCardType");
               String cardNickName =
request.getParameter("txtCardNickName");
               String streetAddress =
request.getParameter("txtStreetAddress");
               String city = request.getParameter("txtCity");
               String state = request.getParameter("cboState");
               String zipCode = request.getParameter("txtZipCode");
               String phone = request.getParameter("txtPhone");
               String website = request.getParameter("txtWebsite");
               String expDate = request.getParameter("txtExpDate");


               debug.append("Account Number: " + accountNum);
               debug.append("Account Holder Name: " +
accountHolderName);
               debug.append("Card Type Id: " + cardTypeId);
               debug.append("Card Nick Name: " + cardNickName);
               debug.append("Street Address: " + streetAddress);
               debug.append("City: " + city);
               debug.append("State: " + state);
               debug.append("Zip Code: " + zipCode);
               debug.append("Phone: " + phone);
               debug.append("Website: " + website);
               debug.append("Expiration Date: " + expDate);
               String userId = null;

    try{
               myConn = new DBConnection();
               conn = myConn.getConnection();
               stmt = conn.createStatement();
```

```
              int creditBalance =
Integer.parseInt(request.getParameter("txtCreditBalance"));
                  debug.append("Credit Balance: " + creditBalance);


        if (session.getAttribute("userID") != null) {
                        userId = (String)
session.getAttribute("userID");
                  }


                  sqlQuery = "INSERT INTO  CREDIT_CARD_DTLS(USER_ID,
ACCOUNT_NUM, ACCOUNT_HOLDER_NAME, CARD_TYPE_ID, CARD_NICK_NAME, "
                        + "PHONE_NUM, WEBSITE, EXPIRATION_DATE,
CREDIT_BALANCE) "
                        + "VALUES('"+ userId + "','" + accountNum +
"','" + accountHolderName + "',"+ cardTypeId +",'" +
                         cardNickName + "','" +  phone + "','" +
website  + "','" +
                         expDate + "'," + creditBalance + ")";

                  debug.append("SQL Query: " + sqlQuery);
                  rowsAffected = stmt.executeUpdate(sqlQuery);
                  debug.append("Rows effected for credit card details:
" + rowsAffected);

                  sqlQuery = "INSERT INTO BILLING_ADDRESS(USER_ID,
STREET_ADDRESS, CITY, STATE_ID, " +
                                    "ZIP_CODE) VALUES ( '"  + userId +
"','" + streetAddress + "','" +
                                     city + "'," + state + "," +
zipCode + ")";
                        debug.append("SQL Query: " + sqlQuery);
                        rowsAffectedBA = stmt.executeUpdate(sqlQuery);
                        debug.append("Rows effected for billing
address: " + rowsAffectedBA);
                        debug.close();
                  if (rowsAffected == 1 && rowsAffectedBA==1) {

%>
                  <script language="JavaScript">

              document.addCardAction.action="CardList_whole.jsp";
                     document.addCardAction.submit();
                  </script>
<%  }else  { %>
                  <h3> There is an error in the application </h3>
<%                }//end of if rowsAffected
                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("AddCardAction", "try block",
"Exception e =" + ex.getMessage());
                        ef.close();
                  }
    stmt.close();
              myConn.closeConnection();
%>
```

```
</form>
</body>
</html>
```

AddNewCard_whole.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>VA Menu</title>
</head>
<body>
<table width=100%>
   <tr>
      <td valign="top"  >

      </td>
   </tr>
   <tr>
      <td>
        <TABLE width="100%" align="center">
          <TR>
                      <TD width="20%" valign="top" align="center">
              <!-- right part of the jsp shud come here -->
              <%@ include file="right_links.jsp" %>
            </TD>
            <TD width="80%" valign="top" align="center">
                    <!-- The main middle page jsp name shud be included
here -->
                       <%@ include file="AddCard.jsp" %>
            </TD>
          </TR>
        </TABLE>

      </td>
     </tr>
  </table>
</body>
</html>
```

Bank1.jsp

```
<%@ page language="java" import="java.sql.*,common.*,java.util.*"%>
<html>
<head>
<title>Preference Processing</title>
</head>
<body>
<form name="bank1">
<%
               //gets the approval code from the database
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    DebugLog debug = new DebugLog("Bank1.txt");
               int rowsAffected = 0, rowsAffectedBA = 0;
               Statement stmt = null;
               debug.append("Obtained values:");

               String userId = null;

    try{
               myConn = new DBConnection();
               conn = myConn.getConnection();
               stmt = conn.createStatement();

        if (session.getAttribute("userID") != null) {
                        userId = (String)
session.getAttribute("userID");
                  }

                  sqlQuery = "SELECT APPROVALCODE FROM BANK1";
                  debug.append("SQL Query: " + sqlQuery);
                  rs = stmt.executeQuery(sqlQuery);
                  while(rs.next()){
                        if(rs.getInt("APPROVALCODE")==1) {
%>
                              <h3> Approved </h3>
<%                }     else  {%>
                              <h3> Denied </h3>

<%                }
                  }


                  rs.close();
                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("PreferenceAction", "try block",
"Exception e =" + ex.getMessage());
                        ef.close();
                  }
               debug.close();
    stmt.close();
               myConn.closeConnection();
```

```
%>
</form>
</body>
</html>
```

CardList.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--java script for client validations-->
<script language="JavaScript">
function evalAll() {
                    test = true;
                    if((document.all) || (document.getElementById)){
                        var cardNickName =
document.cardListForm.txtcardNickName.value;
                        var cardName =
document.cardListForm.txtcardName.value;
                        var cardNum =
document.cardListForm.txtcardNumber.value;
                    }
                    if(cardNickName == ""){
                          alert("Card Nick Name cannot be blank");
              document.cardListForm.txtcardNickName.focus();
              test = false;
              return test;
                    }else if(cardName == ""){
                          alert("Card Name cannot be blank");
              document.cardListForm.txtcardName.focus();
              test = false;
              return test;
                    }else if(cardNum == ""){
                          alert("Card Number cannot be blank");
              document.cardListForm.txtcardNumber.focus();
              test = false;
              return test;
                    }
                    return test;
       }
</script>
<html>
<head>
<title>My Card List</title>
</head>
<body>
<form name="cardListForm" method="post"
action="CardListAction.jsp?id=u">
<table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr>
              <td width="100%" align="left" bgcolor="#333399">
              <font color="#FFFFFF" size="3" face="Arial,
Helvetica"><b>My Credit Cards List</b></font>
    </td>
</tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td><a href="AddNewCard_whole.jsp">Add a New Card</a></td></tr>
<tr><td> </td></tr>
```

```
<tr><td> </td></tr>

<tr><td>
<table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#EFF3FF">
              <tr bgcolor="DEDFDE">
                 <td width="21%" ><strong>Card Nick Name</strong></td>
                 <td width="21%" ><strong>Card Name</strong></td>
                 <td width="21%" ><strong>Card Number</strong></td>
                 <td width="27%" ><strong>Selection</strong></td>
    </tr>
</table>
 <tr bgcolor="B5C7E7"><td colspan="4">
 <table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#EFF3FF">
 <%
    //getting the card details of a particualar user
    DBConnection myConn = null;
              Connection conn = null;
              Statement stmt = null;
    ResultSet rs = null;
              String sqlQuery = null;
              String userId = null;

    try{
              myConn = new DBConnection();
              conn = myConn.getConnection();
              stmt = conn.createStatement();

              DebugLog debug = new DebugLog("CardListJSP.txt");

        if (session.getAttribute("userID") != null){
                        userId = (String)
session.getAttribute("userID");
                  }

                  sqlQuery = "SELECT CARD_ID, CARD_NICK_NAME,
CARD_TYPE_NAME, ACCOUNT_NUM FROM CREDIT_CARD_DTLS, CARD_TYPE " +
                         "WHERE USER_ID = '" + userId + "' " +
                         "AND
CREDIT_CARD_DTLS.CARD_TYPE_ID=CARD_TYPE.CARD_TYPE_ID ";
                  debug.append(sqlQuery);
                  rs = stmt.executeQuery(sqlQuery);
                  while(rs.next()){
                    int cardId = rs.getInt("CARD_ID");
                    String cardNickName =
rs.getString("CARD_NICK_NAME");
                    String cardTypeName =
rs.getString("CARD_TYPE_NAME");
                    String accountNum = rs.getString("ACCOUNT_NUM");

 %>
        <tr bgcolor="EFEFEF" bordercolor="EFEFEF">
          <td bordercolor="FFFFFF" width = "23%" height="25"
align="left"><%=cardNickName%></td>
          <td bordercolor="FFFFFF" width = "24%" height="25"
align="left"><%=cardTypeName%></td>
```

```
                    <td bordercolor="FFFFFF" width = "23%" height="25"
align="left"><%=accountNum%></td>
          <td bordercolor="FFFFFF" width = "16%" height="25"
align="left">
          <a
href="CardList_whole.jsp?id=e&cId=<%=cardId%>&cnn=<%=cardNickName%>&ctn
=<%=cardTypeName%>&an=<%=accountNum%>">Edit </a></td>
          <td bordercolor="FFFFFF" width = "31%" height="25"
align="left">
          <a
href="CardListAction.jsp?id=d&cId=<%=cardId%>">Delete</a></td>
        </tr>
 <% } //end of while loop
     debug.close();
               }catch(Exception ex){
                   ErrorFound ef = new ErrorFound();
                   ef.appendError("CardListJSP", "try block", "Exception
e =" + ex.getMessage());
                   ef.close();
               }
%>
 </table>
 </td></tr>
<% if(request.getParameter("id")!=null &&
request.getParameter("id").equals("e")){ %>
<tr><td>
<table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#EFF3FF">
<tr bgcolor="FFFFFF">
      <td colspan="4"> </td>
</tr>
<tr bgcolor="B5C7E7">
      <td colspan="4">Update Credit Card Details
      </td>
<tr><td> </td></tr>
               <td width="19%" align="right" ><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
               <font color="#000000" size="2" face="Arial,
Helvetica"><b>Card Nick Name:</b></font></td>
               <td width="81%"><input name="txtcardNickName" type="text"
maxlength="50" size="50" value="<%=request.getParameter("cnn")%>"></td>
</tr>
<tr>
               <td width="19%" align="right" ><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
               <font color="#000000" size="2" face="Arial,
Helvetica"><b>Card Name:</b></font></td>
               <td width="81%">
               <select name="cboCardType" size="1">
                     <option selected> </option>
          <%
                              DBConnection myConn1 = null;
                         ResultSet rs1 = null;
                              String sqlQuery1 = null;
                         DebugLog debug1 = new DebugLog("AddCard.txt");
                         try{
                              myConn1 = new DBConnection();
```

```
                              Connection conn1 =
myConn1.getConnection();
                              Statement stmt1 =
conn1.createStatement();
                                    sqlQuery1 = "SELECT CARD_TYPE_ID,
CARD_TYPE_NAME FROM CARD_TYPE";
                                    debug1.append("SQL Query: " +
sqlQuery1);
                                    rs1 =
stmt1.executeQuery(sqlQuery1);
                                    while (rs1.next()) {
                                          String cardTypeID =
rs1.getString("CARD_TYPE_ID");
                                          String cardTypeName =
rs1.getString("CARD_TYPE_NAME");
                                          debug1.append("Card Type ID:
" + cardTypeID + "Card Type Name: "  + cardTypeName);
                        %>
                        <option value="<%=cardTypeID%>"

              <%if(cardTypeName.equals(request.getParameter("ctn"))){

              out.println("selected");}%>><%=cardTypeName%></option>
                        <%          } //end of while
                                    rs1.close();
                                    stmt1.close();
                                    debug1.close();
                                    myConn1.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("AddNewCard", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
        </select></font>
                        </td>
</tr>
<tr>
                        <td width="19%"align="right" ><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
                        <font color="#000000" size="2" face="Arial,
Helvetica"><b>Card Number:</b></font></select></td>
                        <td width="81%"><input name="txtcardNumber" type="text"
maxlength="25" size="25" value="<%=request.getParameter("an")%>"></td>
</tr>
<input type="hidden" name="cardId"
value="<%=request.getParameter("cId")%>">
</table>
<br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
                  <td width="50%" align="center">
                    <input name="pagemode" type="hidden" value="submit">
```

```
                  <input type="submit" value="Update" onKeyDown='return
evalAll();' value="Submit" onClick='return evalAll();'>
                </td>
                <td width="50%"> </td>

</tr>
</table>
</td></tr>
<%}%>
</table>
</form>
</body>
</html>
```

CardList_whole.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>VA Menu</title>
</head>
<body>
<table width=100%>
   <tr>
      <td valign="top"  >

      </td>
   </tr>
   <tr>
      <td>
        <TABLE width="100%" align="center">
          <TR>
                      <TD width="20%" valign="top" align="center">
              <!-- right part of the jsp shud come here -->
              <%@ include file="right_links.jsp" %>
            </TD>
            <TD width="80%" valign="top" align="center">
                    <!-- The main middle page jsp name shud be included
here -->
                     <% if(request.getParameter("id")!=null &&
request.getParameter("id").equals("e")){ %>
                           <input type="hidden" name="id"
value="<%=request.getParameter("id")%>">
                           <input type="hidden" name="cId"
value="<%=request.getParameter("cId")%>">
                           <input type="hidden" name="cnn"
value="<%=request.getParameter("cnn")%>">
                          <input type="hidden" name="ctn"
value="<%=request.getParameter("ctn")%>">
                           <input type="hidden" name="an"
value="<%=request.getParameter("an")%>">
                      <%}%>
            <%@ include file="CardList.jsp" %>
            </TD>
          </TR>
        </TABLE>

      </td>
     </tr>
  </table>
</body>
</html>
```

CardListAction.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<html>
<head>
<title>Card List</title>
</head>
<body>
<form name="cardListAction" method="post">
<%

               //updating the credit card details of a particular user
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    String userID = null;
               int rowsAffected = 0;
               Statement stmt = null;
               DebugLog debug = new DebugLog("CardListAction.txt");
               debug.append("Obtained values:");

               try{
                   myConn = new DBConnection();
               conn = myConn.getConnection();
               stmt = conn.createStatement();
                   if(session.getAttribute("userID")!=null){
                         userID = (String)
session.getAttribute("userID");
                         debug.append("CardListAction : User Id :" +
userID);
                   }
                   //if action is update
                   if(request.getParameter("id").equals("u")) {
                         sqlQuery =  "UPDATE CREDIT_CARD_DTLS SET
CARD_NICK_NAME='"+ request.getParameter("txtcardNickName") +"',"
                                     +   "CARD_TYPE_ID ="+
request.getParameter("cboCardType") +",ACCOUNT_NUM
='"+request.getParameter("txtcardNumber")+"' "
                                     +   "WHERE CARD_ID =
"+Integer.parseInt(request.getParameter("cardId"));


                         debug.append("SQL Query: " + sqlQuery);
                         rowsAffected = stmt.executeUpdate(sqlQuery);

                   }else {
                         rowsAffected =0; //Anonymous User.
                   }

                   debug.append("Rows effected for user address: " +
rowsAffected);
                   if (rowsAffected == 1) {
%>
                   <script language="JavaScript">
```

```
document.cardListAction.action="CardList_whole.jsp";
                    document.cardListAction.submit();
               </script>
<%  }else  { %>
                    <tr><td width="81%"> <font color="#003366" size="2"
face="Arial, Helvetica">
                    There is an error in the application
</font></td></tr>
<%                  }//end of if rowsAffected

                    //deleting the credit card details of a particular
user
                    if(request.getParameter("id").equals("d")){
                          sqlQuery =  "DELETE FROM CREDIT_CARD_DTLS "
                                      + "WHERE CARD_ID =
"+Integer.parseInt(request.getParameter("cId"));


                          debug.append("SQL Query: " + sqlQuery);
                          rowsAffected = stmt.executeUpdate(sqlQuery);

                    }else {
                          rowsAffected =0; //Anonymous User.
                    }

                    if (rowsAffected == 1) {
%>
                    <script language="JavaScript">

document.cardListAction.action="CardList_whole.jsp";
                       document.cardListAction.submit();
               </script>
<%  }else  { %>
                    <tr><td width="81%"> <font color="#003366" size="2"
face="Arial, Helvetica">
                    There is an error in the application
</font></td></tr>
<%                  }//end of if rowsAffected
                    debug.close();
                    }catch(Exception e){
                          ErrorFound ef = new ErrorFound();
                          ef.appendError("CardListAction.jsp", "try block
of transaction", "Exception e=" + e.getMessage());
                          ef.close();
                    }
                    stmt.close();
                    myConn.closeConnection();
    %>
<tr><td> 
</td></tr>
</form>
</body>
</html>
```

Login.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>

<!-- jsp that collects the login details-->
<script language="JavaScript">
                  function callReg() {
                     document.loginForm.action="Registration.jsp";
                     document.loginForm.submit();
        }
</script>
<html>
<head>
<title>Divisible Card Payment System</title>
</head>
<body>
<form name="loginForm" method="post" action="LoginAction.jsp">
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
   <td width="100%" align="center" bgcolor="#FFFFFF">
       <IMG border=0 height=180 src="title.gif" width="244">
   </td>
</tr>
<tr>
    <td width="100%" align="center" bgcolor="#333399">
               <font color="#FFFFFF" size="3" face="Arial,
Helvetica"><b>Welcome to V-Card System</b></font>
    </td>
</tr>
</table>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<table width="35%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#6699CC">
<% if (session.getAttribute("login")!=null &&
session.getAttribute("msg")!= null) {
                  if (session.getAttribute("login").equals("false")) {
%>
  <tr>
    <td width="100%" align="center"><font color="#000000" size="2"
face="Arial, Helvetica"><b>
                  <%=session.getAttribute("msg")%></b></font></td>
  </tr>
<% } else { %>
  <tr>
    <td width="100%"> </td>
  </tr>
<% } //end if inner if
   }//end if outer if
%>
</table>
<table width="35%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#6699CC">
  <tr>
    <td width="42%"> </td>
    <td width="58%"> </td>
```

```
  </tr>
  <tr>
    <td width="42%"> </td>
    <td width="58%"> </td>
  </tr>
  <tr>
    <td width="42%" align="right"><font color="#000000" size="2"
face="Arial, Helvetica"><b>User
      ID:</b></font></td>
    <td width="58%"><input name="txtUserID" type="text" maxlength="25"
size="25"></td>
  </tr>
  <tr>
    <td width="42%" align="right"><font color="#000000" size="2"
face="Arial, Helvetica"><b>Password:</b></font></td>
    <td width="58%"><input name="txtPassword" type="password"
maxlength="25" size="25"></td>
  </tr>
  <tr>
    <td width="42%"> </td>
    <td width="58%"> </td>
  </tr>
  <tr>
    <td width="42%"> </td>
             <td width="42%" align="left"><input value="Login"
type="submit" size="30"></td>
  </tr>
  <tr>
    <td width="42%"> </td>
    <td width="58%"> </td>
  </tr>
  <tr>
    <td width="42%" align="right"><font color="#000000" size="1"
face="Arial, Helvetica"><b>If not New User, Click here:</b></font></td>
             <td width="42%" align="left"><input value="Sign In"
onClick="callReg()" type="submit" size="30"></td>
  </tr>
</table>
</form>
</body>
</html>
```

LoginAction.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<html>
<head>
<title>Login to VAAgent</title>
</head>
<body>
<form name="loginAction" method="post">
<%
               //action class that takes login details and validates the
login information
               DBConnection myConn = null;
               ResultSet rs = null;
               String sqlQuery = null;
               String loginMsg =  null;
               DebugLog debug = new DebugLog("Login.txt");
               try{
                   myConn = new DBConnection();
                   Connection conn = myConn.getConnection();
                   Statement stmt = conn.createStatement();
                   debug.append("userID: " +
request.getParameter("txtUserID"));
                   debug.append("Password: " +
request.getParameter("txtPassword"));
                   if(request.getParameter("txtUserID")!=null &&
request.getParameter("txtPassword")!=null){
                         sqlQuery = "SELECT COUNT(*) FROM USERS WHERE
USER_ID = '"  +

               request.getParameter("txtUserID") + "' AND PASSWD = '" +

               request.getParameter("txtPassword") + "'";
                         debug.append("SQL Query: " + sqlQuery);
                         rs = stmt.executeQuery(sqlQuery);
                         while(rs.next()){
                               if( rs.getString(1).equals("1")){
                                     loginMsg ="Logged in successfully";
                                     session.setAttribute("login",
"true");
                                     sqlQuery = "SELECT FIRST_NAME,
LAST_NAME FROM USERS WHERE USER_ID = '"  +

               request.getParameter("txtUserID") + "' AND PASSWD = '" +

               request.getParameter("txtPassword") + "'";

                                     debug.append("SQL Query: " +
sqlQuery);
                                     rs = stmt.executeQuery(sqlQuery);
                                     while (rs.next()) {
                                           debug.append("First Name: "
+rs.getString("FIRST_NAME")   + " " + "Last Name: " +
rs.getString("LAST_NAME"));
```

```
                session.setAttribute("userName",
rs.getString("FIRST_NAME") + " " + rs.getString("LAST_NAME"));

                session.setAttribute("userID",
request.getParameter("txtUserID"));
                                        }

                                  }else{
                                        loginMsg ="Login failed. Please try
Again";
                                        session.setAttribute("login",
"false");
                                  }
                            }
                     }
                     session.setAttribute("msg", loginMsg);
                     debug.append("Login Msg: " + loginMsg);
                     debug.close();
                     rs.close();
                     stmt.close();
                     myConn.closeConnection();
                   }catch(Exception ex){
                     ErrorFound ef = new ErrorFound();
                     ef.appendError("LoginAction", "try block", "Exception
e =" + ex.getMessage());
                     ef.close();
                }
                if(session.getAttribute("login").equals("true")) {
%>
                <script language="JavaScript">
                       document.loginAction.action="VAMenu_whole.jsp";
                       document.loginAction.submit();
    </script>
<% }else { %>
                <script language="JavaScript">
                       document.loginAction.action="Login.jsp";
                       document.loginAction.submit();
    </script>
<% } %>
</form>
</body>
</html>
```

Logout.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<% session.invalidate();%>
<html>
<head>
<title>Logout</title>
</head>
<body>
<form name="logoutForm" method="post" action="Login.jsp">

<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
              <td width="100%" align="center" bgcolor="#333399">
              <font color="#FFFFFF" size="3" face="Arial,
Helvetica"><b>You are signed off</b></font>
    </td>
</tr>
</table>
<br><br>
<table width="35%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
  <tr>
              <td width="100%" align="center"><input value="Login"
type="submit" size="30"></td>
  </tr>
</table>
</form>
</body>
</html>
```

Preference.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--java script for client side validations-->
<script language="JavaScript">
              function evalAll() {
                    test = true;
                    if((document.all) || (document.getElementById)){
                      var myIndex1 =
document.prefForm.cboCardFeatures.selectedIndex;
                        var selValue1 =
document.prefForm.cboCardFeatures.options[myIndex1].value;
                        var myIndex2 =
document.prefForm.cboRewardPoints.selectedIndex;
                        var selValue2 =
document.prefForm.cboRewardPoints.options[myIndex2].value;
                        var myIndex3 =
document.prefForm.cboPayType.selectedIndex;
                        var selValue3 =
document.prefForm.cboPayType.options[myIndex3].value;
                    }
          if(myIndex1 == "0"){
                        alert("Please select Card Feature");
              document.prefForm.cboCardFeatures.focus();
              test = false;
              return test;
                    }else if(myIndex2 == "0"){
                          alert("Please select Reward Point");
              document.prefForm.cboRewardPoints.focus();
              test = false;
              return test;
                    }else if(myIndex3 == "0"){
                          alert("Please select Pay Type");
              document.prefForm.cboPayType.focus();
              test = false;
              return test;
                    }

                    return test;
       }
</script>
<html>
<head>
<title>My Preference List</title>
</head>
<body>
<form name="prefForm" method="post" action="PreferenceAction.jsp">
<table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr>
              <td width="100%" align="left" bgcolor="#333399">
              <font color="#FFFFFF" size="3" face="Arial,
Helvetica"><b>My Preferences List</b></font>
```

```
    </td>
</tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td>To help us recommend the perfect Card for your needs, please
answer brief questions.</td></tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td>Which of these Card features would you find most
useful?</td></tr>
<tr><td>
<select name="cboCardFeatures" size="1">
                    <option selected value="0">Select one...</option>
          <%
                        //getting preference details from preference
master table
                              DBConnection myConn1 = null;
                        ResultSet rs1 = null;
                              String sqlQuery1 = null;
                        DebugLog debug1 = new
DebugLog("Preferences.txt");
                        try{
                              myConn1 = new DBConnection();
                              Connection conn1 =
myConn1.getConnection();
                              Statement stmt1 =
conn1.createStatement();
                                    sqlQuery1 = "SELECT
CARD_FEATURE_ID, CARD_FEATURE_DESC FROM CARD_FEATURES WHERE
CATEGORY_ID=1";
                                    debug1.append("SQL Query: " +
sqlQuery1);
                                    rs1 =
stmt1.executeQuery(sqlQuery1);
                                    while (rs1.next()) {
                                          String cardFeatureID =
rs1.getString("CARD_FEATURE_ID");
                                          String cardFeatureDesc =
rs1.getString("CARD_FEATURE_DESC");
                                          debug1.append("Card Feature
ID: " + cardFeatureID + "Card Feature Desc: "  + cardFeatureDesc);
                        %>
                        <option
value="<%=cardFeatureID%>"><%=cardFeatureDesc%></option>
                        <%          } //end of while
                                    rs1.close();
                                    stmt1.close();

                                    myConn1.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("Preferences", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
```

```
        </select></font>
</td></tr>
<tr><td> </td></tr>
<tr><td>Which type of rewards points interests you most?</td></tr>
<tr><td>
<select name="cboRewardPoints" size="1">
                    <option selected value="0">Select one...</option>
          <%
                        try{
                              myConn1 = new DBConnection();
                              Connection conn1 =
myConn1.getConnection();
                              Statement stmt1 =
conn1.createStatement();
                                    sqlQuery1 = "SELECT
CARD_FEATURE_ID, CARD_FEATURE_DESC FROM CARD_FEATURES WHERE
CATEGORY_ID=2";
                                    debug1.append("SQL Query: " +
sqlQuery1);
                                    rs1 =
stmt1.executeQuery(sqlQuery1);
                                    while (rs1.next()) {
                                          String cardFeatureID =
rs1.getString("CARD_FEATURE_ID");
                                          String cardFeatureDesc =
rs1.getString("CARD_FEATURE_DESC");
                                          debug1.append("Card Feature
ID: " + cardFeatureID + "Card Feature Desc: "  + cardFeatureDesc);
                        %>
                        <option
value="<%=cardFeatureID%>"><%=cardFeatureDesc%></option>
                        <%          } //end of while
                                    rs1.close();
                                    stmt1.close();

                                    myConn1.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("Preferences", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
        </select></font>
</td></tr>
<tr><td> </td></tr>
<tr><td>How would you prefer to pay your bill?</td></tr>
<tr><td>
<select name="cboPayType" size="1">
                    <option selected value="0">Select one...</option>
          <%
                        try{
                              myConn1 = new DBConnection();
                              Connection conn1 =
myConn1.getConnection();
```

```
                              Statement stmt1 =
conn1.createStatement();
                                    sqlQuery1 = "SELECT
CARD_FEATURE_ID, CARD_FEATURE_DESC FROM CARD_FEATURES WHERE
CATEGORY_ID=3";
                                    debug1.append("SQL Query: " +
sqlQuery1);
                                    rs1 =
stmt1.executeQuery(sqlQuery1);
                                    while (rs1.next()) {
                                          String cardFeatureID =
rs1.getString("CARD_FEATURE_ID");
                                          String cardFeatureDesc =
rs1.getString("CARD_FEATURE_DESC");
                                          debug1.append("Card Feature
ID: " + cardFeatureID + "Card Feature Desc: "  + cardFeatureDesc);
                        %>
                        <option
value="<%=cardFeatureID%>"><%=cardFeatureDesc%></option>
                        <%          } //end of while
                                    rs1.close();
                                    stmt1.close();

                                    myConn1.closeConnection();
                              }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("Preferences", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                              }
                        %>
                  </select></font>
</td></tr>
</table>
<br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
      <td width="10%" align="left">
                     Proceed:
                </td>
                <td width="12%" align="left">
                  <input type="submit" value="Continue"
onKeyDown='return evalAll();' onClick='return evalAll();'>
                </td>
                <td width="78%" align="left"><input type="reset"
value="Cancel"</td>

</tr>
</table>
</td></tr>
</table>
</form>
</body>
</html>
```

Preference_whole.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>VA Menu</title>
</head>
<body>
<table width=100%>
   <tr>
      <td valign="top"  >

      </td>
   </tr>
   <tr>
      <td>
        <TABLE width="100%" align="center">
          <TR>
                      <TD width="20%" valign="top" align="center">
              <!-- right part of the jsp shud come here -->
              <%@ include file="right_links.jsp" %>
            </TD>
            <TD width="80%" valign="top" align="center">
                    <!-- The main middle page jsp name shud be included
here -->
                       <%@ include file="Preference.jsp" %>
            </TD>
          </TR>
        </TABLE>

      </td>
     </tr>
  </table>
</body>
</html>
```

PreferenceAction.jsp

```
<%@ page language="java" import="java.sql.*,common.*"%>
<html>
<head>
<title>Preference Processing</title>
</head>
<body>
<form name="prefAction" action="Preference_whole.jsp">
<%
               //gets the preference information selected by user and
stores in the database
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    DebugLog debug = new DebugLog("PreferenceAction.txt");
               int rowsAffected = 0, rowsAffectedBA = 0;
               Statement stmt = null;
               debug.append("Obtained values:");

               String userId = null;
               int cardFeaturesId =0;

    try{
               myConn = new DBConnection();
               conn = myConn.getConnection();
               stmt = conn.createStatement();

                   if (session.getAttribute("userID") != null) {
                         userId = (String)
session.getAttribute("userID");
                   }

               if(request.getParameter("cboCardFeatures")!=null){
                   cardFeaturesId =
Integer.parseInt(request.getParameter("cboCardFeatures"));

                   sqlQuery = "INSERT INTO  USER_PREFERENCES(USER_ID,
CARD_FEATURE_ID) "
                         + "VALUES('"+ userId + "'," + cardFeaturesId  +
")";

                         debug.append("SQL Query: " + sqlQuery);
                         rowsAffected = stmt.executeUpdate(sqlQuery);
                         debug.append("Rows effected for user
preferences details: " + rowsAffected);
               }
               if(request.getParameter("cboRewardPoints")!=null){
                         cardFeaturesId =
Integer.parseInt(request.getParameter("cboRewardPoints"));

                         sqlQuery = "INSERT INTO
USER_PREFERENCES(USER_ID, CARD_FEATURE_ID) "
                         + "VALUES('"+ userId + "'," + cardFeaturesId  +
")";
```

```
                        debug.append("SQL Query: " + sqlQuery);
                        rowsAffected = stmt.executeUpdate(sqlQuery);
                        debug.append("Rows effected for user
preferences details: " + rowsAffected);
                  }
                  if(request.getParameter("cboPayType")!=null){
                        cardFeaturesId =
Integer.parseInt(request.getParameter("cboPayType"));

                        sqlQuery = "INSERT INTO
USER_PREFERENCES(USER_ID, CARD_FEATURE_ID) "
                        + "VALUES('"+ userId + "'," + cardFeaturesId  +
")";

                        debug.append("SQL Query: " + sqlQuery);
                        rowsAffected = stmt.executeUpdate(sqlQuery);
                        debug.append("Rows effected for user
preferences details: " + rowsAffected);
                  }

                  debug.close();
                  if (rowsAffected == 1) {
%>
                  <script language="JavaScript">

               document.prefAction.action="PurchaseAmt_whole.jsp";
                      document.prefAction.submit();
                  </script>
<%  }else  { %>
                  <h3> There is an error in the application </h3>
<%                }//end of if rowsAffected
                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("PreferenceAction", "try block",
"Exception e =" + ex.getMessage());
                        ef.close();
                  }
    stmt.close();
               myConn.closeConnection();
%>
</form>
</body>
</html>
```

PurchaseAction.jsp

```
<%@ page language="java" import="java.sql.*,common.*,java.util.*"%>
<html>
<head>
<title>Preference Processing</title>
</head>
<body>
<form name="prefAction" action="Preference_whole.jsp">
<%
               //inserts the purchase amount in the database
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    DebugLog debug = new DebugLog("PurchaseAction.txt");
               int rowsAffected = 0, rowsAffectedBA = 0;
               Statement stmt = null;
               debug.append("Obtained values:");

               String userId = null;
               float purchaseAmt = 0;
               int purchaseId = 0;
               int vcardId = 0;
               String expiryDate = null;

    try{
               myConn = new DBConnection();
               conn = myConn.getConnection();
               stmt = conn.createStatement();

               if(request.getParameter("txtPurchaseAmount")!=null){
                   purchaseAmt =
Float.parseFloat(request.getParameter("txtPurchaseAmount"));
               }

               debug.append("Purchase Amount: " + purchaseAmt);

        if (session.getAttribute("userID") != null) {
                        userId = (String)
session.getAttribute("userID");
                  }

                  sqlQuery = "SELECT PURCHASEID_SEQ.NEXTVAL PURCHASE_ID
FROM DUAL";
                  debug.append("SQL Query: " + sqlQuery);
                  rs = stmt.executeQuery(sqlQuery);
                  while(rs.next()){
                        purchaseId = rs.getInt("PURCHASE_ID");
                  }

                  rs.close();

                  sqlQuery = "INSERT INTO PURCHASE_DTLS(USER_ID,
PURCHASE_ID, PURCHASE_AMT) "
```

```
                        + "VALUES('"+ userId + "'," + purchaseId + ","
+ purchaseAmt + ")";

                    debug.append("SQL Query: " + sqlQuery);
                    rowsAffected = stmt.executeUpdate(sqlQuery);
                    debug.append("Rows effected for purchase amount
details: " + rowsAffected);

                        //if purchase amount is inersted into the
database then get all the details of
                        //all the cards from the database and generate
the optimized solution of credit cards
                   if (rowsAffected == 1) {
                       int S = 0;
                       int Y = 0;
                       int P=0;
                       int card1Amt =0;
                       int card2Amt=0;
                       int card3Amt=0;
                       int card1Bal=0;
                       int card2Bal=0;
                       int card3Bal=0;
                       Vector featureVect = new Vector();
                       int returnCode =0;
                       StringBuffer sb = new StringBuffer();
                       String arr [][] = new String [3][2];

                        sqlQuery = "SELECT COUNT(*) COUNT FROM
CREDIT_CARD_DTLS WHERE USER_ID = '" + userId + "'";
                        debug.append("SQL Query: " + sqlQuery);
                        rs=stmt.executeQuery(sqlQuery);
                        while(rs.next()){
                              S = rs.getInt("COUNT");
                        }

                        rs.close();

                        if(purchaseAmt!=0){
                              Y = (int)purchaseAmt;
                        }

                        sqlQuery = "SELECT B.CARD_TYPE_ID CARD_TYPE_ID,
CT.CARD_TYPE_NAME CARD_TYPE_NAME, " +
                                    "C.CARD_FEATURE_ID CARD_FEATURE_ID,
C.CARD_FEATURE_DESC CARD_FEATURE_DESC " +
                                    "FROM CARD_FEATURES C, USER_PREFERENCES
U, BONUS_FEATURES B, CARD_TYPE CT " +
                                    "WHERE C.CARD_FEATURE_ID =
U.CARD_FEATURE_ID " +
                                    "AND B.CARD_FEATURE_ID =
C.CARD_FEATURE_ID " +
                                    "AND B.CARD_TYPE_ID = CT.CARD_TYPE_ID " +
                                    "AND U.USER_ID='" + userId + "'";
                        debug.append("SQL Query: " + sqlQuery);

                        rs=stmt.executeQuery(sqlQuery);
                        while(rs.next()){
```

```
                              FeatureForm featureForm = new
FeatureForm();

               featureForm.setCardTypeId(rs.getInt("CARD_TYPE_ID"));

               featureForm.setCardTypeDesc(rs.getString("CARD_TYPE_NAME"
));

               featureForm.setCardFeatureId(rs.getInt("CARD_FEATURE_ID")
);

               featureForm.setCardFeatureDesc(rs.getString("CARD_FEATURE
_DESC"));
                              featureVect.add(featureForm);
                        }

                        rs.close();

                        int w1=0, w2=0, w3=0;
                        String cardName1=null, cardName2=null,
cardName3=null;

                        for(int i=0;i<featureVect.size();i++){
                              FeatureForm fForm = (FeatureForm)
featureVect.elementAt(i);

                              debug.append("Card Id selected are: " +
fForm.getCardTypeId());

                              if(fForm.getCardTypeId()==1){
                                    w1 = w1+1;
                                    debug.append("w1 is" + w1);
                                    arr[0][0] =
fForm.getCardTypeDesc();
                              }else if(fForm.getCardTypeId()==2){
                                    w2 = w2+1;
                                    debug.append("w2 is: " + w2);
                                    arr[1][0] =
fForm.getCardTypeDesc();
                              }else if(fForm.getCardTypeId()==3){
                                    w3 = w3+1;
                                    debug.append("w3 is : "+ w3);
                                    arr[2][0] =
fForm.getCardTypeDesc();
                              }
                        }//end of for loop

                        if(w1!=0){
                              sqlQuery = "SELECT C.CREDIT_LIMIT
CREDIT_LIMIT, CC.CREDIT_BALANCE CREDIT_BALANCE, " +
                                    "C.INTEREST_RATE INTEREST_RATE,
C.MIN_PAYMENT MIN_PAYMENT " +
                                    "FROM CARD_TYPE C, CREDIT_CARD_DTLS
CC " +
                                    "WHERE USER_ID='" + userId + "' " +
                                    "AND C.CARD_TYPE_ID =
CC.CARD_TYPE_ID " +
```

```
                                    "AND C.CARD_TYPE_ID = 1 ";
                              debug.append("SQL Query: " + sqlQuery);

                              rs=stmt.executeQuery(sqlQuery);

                              int x=0;

                              while(rs.next()){
                                    x =
(rs.getInt("INTEREST_RATE")*rs.getInt("MIN_PAYMENT"));
                                    card1Bal =
rs.getInt("CREDIT_BALANCE");
                                    if(rs.getInt("CREDIT_BALANCE")-
x>=50){
                                          card1Amt = x;
                                    }else{
                                          card1Amt = x-50;
                                    }
                              }//end of while loop
                              arr[0][1] = String.valueOf(card1Amt);
                              debug.append("Card 1 Amount: " +
card1Amt);
                              rs.close();
                        }//end of if loop of w1!=0

                        if(w2!=0){
                              sqlQuery = "SELECT C.CREDIT_LIMIT
CREDIT_LIMIT, CC.CREDIT_BALANCE CREDIT_BALANCE, " +
                                    "C.INTEREST_RATE INTEREST_RATE,
C.MIN_PAYMENT MIN_PAYMENT " +
                                    "FROM CARD_TYPE C, CREDIT_CARD_DTLS
CC " +
                                    "WHERE USER_ID='" + userId + "' " +
                                    "AND C.CARD_TYPE_ID =
CC.CARD_TYPE_ID " +
                                    "AND C.CARD_TYPE_ID = 2 ";

                              rs=stmt.executeQuery(sqlQuery);

                              int x=0;

                              while(rs.next()){
                                    x =
(rs.getInt("INTEREST_RATE")*rs.getInt("MIN_PAYMENT"));
                                    card2Bal =
rs.getInt("CREDIT_BALANCE");
                                    if(rs.getInt("CREDIT_BALANCE")-
x>=50 && card1Amt!=0){
                                          card2Amt = Y-card1Amt;
                                    }else
if(rs.getInt("CREDIT_BALANCE")-x>=50){
                                          card2Amt = x;
                                    }else{
                                          card2Amt = x-50;
                                    }
                              }//end of while loop
                              arr[1][1] = String.valueOf(card2Amt);
```

```
                              debug.append("Card 2 Amount: " +
card2Amt);
                              rs.close();
                        }//end of if loop of w2!=0

                        if(w3!=0){
                              sqlQuery = "SELECT C.CREDIT_LIMIT
CREDIT_LIMIT, CC.CREDIT_BALANCE CREDIT_BALANCE, " +
                                    "C.INTEREST_RATE INTEREST_RATE,
C.MIN_PAYMENT MIN_PAYMENT " +
                                    "FROM CARD_TYPE C, CREDIT_CARD_DTLS
CC " +
                                    "WHERE USER_ID='" + userId + "' " +
                                    "AND C.CARD_TYPE_ID =
CC.CARD_TYPE_ID " +
                                    "AND C.CARD_TYPE_ID = 3 ";
                              debug.append("SQL Query: " + sqlQuery);

                              rs=stmt.executeQuery(sqlQuery);

                              int x=0;

                              while(rs.next()){
                                    x =
(rs.getInt("INTEREST_RATE")*rs.getInt("MIN_PAYMENT"));
                                    card3Bal =
rs.getInt("CREDIT_BALANCE");
                                    if(rs.getInt("CREDIT_BALANCE")-
x>=50 && card1Amt!=0){
                                          card3Amt = Y-card1Amt;
                                    }if(rs.getInt("CREDIT_BALANCE")-
x>=50 && card2Amt!=0){
                                          card3Amt = Y-card2Amt;
                                    }else
if(rs.getInt("CREDIT_BALANCE")-x>=50){
                                          card3Amt = x;
                                    }else{
                                          card3Amt = x-50;
                                    }
                              }//end of while loop
                              arr[2][1] = String.valueOf(card3Amt);
                              debug.append("Card 3 Amount: " +
card3Amt);
                              rs.close();
                        }//end of if loop of w3!=0

                        if((card1Amt!=0 && card2Amt!=0)
&&(card1Amt+card2Amt<Y)){
                              if((card3Bal - card1Amt+card2Amt)>=50){
                                    card3Amt = Y - card1Amt+card2Amt;
                                    arr[2][1] =
String.valueOf(card3Amt);
                              }
                        } else if((card2Amt!=0 && card3Amt!=0)
&&(card2Amt+card3Amt<Y)){
                              if((card1Bal - card2Amt+card3Amt)>=50){
                                    card1Amt = Y - card2Amt+card3Amt;
```

```
                                    arr[0][1] =
String.valueOf(card1Amt);
                              }
                        } else if((card1Amt!=0 && card3Amt!=0)
&&(card1Amt+card3Amt<Y)){
                              if((card2Bal - card1Amt+card3Amt)>=50){
                                    card2Amt = Y - card1Amt+card3Amt;
                                    arr[1][1] =
String.valueOf(card2Amt);
                              }
                        }


                        debug.append("array size is: " + arr.length);

                        debug.append("array results are:");


                        debug.append("arr[0][0]: " + arr[0][0]);
                        debug.append("arr[0][1]: " + arr[0][1]);
                        debug.append("arr[1][0]: " + arr[1][0]);
                        debug.append("arr[1][1]: " + arr[1][1]);
                        debug.append("arr[2][0]: " + arr[2][0]);
                        debug.append("arr[2][1]: " + arr[2][1]);

                        if(arr[0][0]!=null){
                              returnCode = 1;
                              sb.append("<tr bgcolor=\"DEDFDE\">" +
                                          "\n<td
width=\"50%\"><strong>"+arr[0][0]+"</strong></td>" +
                                          "\n<td
width=\"50%\"><strong>"+arr[0][1]+"</strong></td></tr>");
                        }
                        if(arr[1][0]!=null){
                              returnCode = 1;
                              sb.append("\n<tr bgcolor=\"DEDFDE\">" +
                                          "\n<td
width=\"50%\"><strong>"+arr[1][0]+"</strong></td>" +
                                          "\n<td
width=\"50%\"><strong>"+arr[1][1]+"</strong></td></tr>");
                        }
                        if(arr[2][0]!=null){
                              returnCode =1;
                              sb.append("\n<tr bgcolor=\"DEDFDE\">" +
                                          "\n<td
width=\"50%\"><strong>"+arr[2][0]+"</strong></td>" +
                                          "\n<td
width=\"50%\"><strong>"+arr[2][1]+"</strong></td></tr>");
                        }

                        sqlQuery = "DELETE FROM BANK1";

                        debug.append("SQL Query: "+ sqlQuery);

                        stmt.executeUpdate(sqlQuery);
```

```
                        sqlQuery = "INSERT INTO BANK1 (APPROVALCODE)
VALUES (" + returnCode + ")";

                        debug.append("SQL Query: "+ sqlQuery);

                        stmt.executeUpdate(sqlQuery);

                        sqlQuery = "SELECT VCARDID_SEQ.NEXTVAL VCARD_ID
FROM DUAL";
                        debug.append("SQL Query: " + sqlQuery);
                        rs = stmt.executeQuery(sqlQuery);
                        while(rs.next()){
                              vcardId = rs.getInt("VCARD_ID");
                        }

                        rs.close();

                        sqlQuery = "SELECT TO_CHAR(SYSDATE+365,
'MM/YY') EXPIRY_DATE FROM DUAL";
                        debug.append("SQL Query: " + sqlQuery);
                        rs = stmt.executeQuery(sqlQuery);
                        while(rs.next()){
                              expiryDate = rs.getString("EXPIRY_DATE");
                        }

                        rs.close();


                        sqlQuery = "INSERT INTO VCARD(USER_ID,
VCARDNUM, EXPIRY_DATE, AMOUNT_CHARGED) "
                                    + "VALUES('"+ userId + "'," + vcardId +
",'" + expiryDate +"', " +  Y + ")";

                        debug.append("SQL Query: " + sqlQuery);
                        stmt.executeUpdate(sqlQuery);

                        session.setAttribute("VAResults",
sb.toString());
%>
                  <script language="JavaScript">

               document.prefAction.action="PurchaseAmt_whole.jsp";
                       document.prefAction.submit();
                  </script>
<%  }else  { %>
                  <h3> There is an error in the application </h3>
<%                }//end of if rowsAffected
                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("PreferenceAction", "try block",
"Exception e =" + ex.getMessage());
                        ef.close();
                  }
               debug.close();
    stmt.close();
               myConn.closeConnection();
%>
```

```
</form>
</body>
</html>
```

PurchaseAmt.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--collects the purchase amount-->
<script language="JavaScript">
</script>
<html>
<head>
<title>Purchase Amount Information</title>
</head>
<body>
<form name="purAmountForm" method="post" action="PurchaseAction.jsp">
<table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
<tr><td > </td></tr>
<tr><td > </td></tr>
<tr><td> </td></tr>
<tr>
               <td width="22%" align="left" ><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
               <font color="#000000" size="2" face="Arial,
Helvetica"><b>Enter the Purchase Amount:</b></font></td>
               <td width="78%" align="left"><input
name="txtPurchaseAmount" type="text" maxlength="10" size="19"></td>
</tr>
</table>
<br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
  <td width="13%" align="right"><input type="submit" name="opt"
value="Go Optimization" ></td>
  <td width="87%"> </td>
</tr>
<tr><td > </td></tr>
</table>
<% if(session.getAttribute("VAResults")!=null) {
                  Object sb = session.getAttribute("VAResults");
%>
<br><br><br><br>
<table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr><td> </td></tr>
<tr>
               <td width="100%" align="left" bgcolor="#333399">
               <font color="#FFFFFF" size="3" face="Arial,
Helvetica"><b>VA Results</b></font>
    </td>
</tr>
</table>
<br><br><br>
<table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#EFF3FF">
               <tr bgcolor="DEDFDE">
```

```
                <td width="50%"><strong>Card Nick Name</strong></td>
                <td width="50%"><strong>Amount Charged</strong></td>
    </tr>
</table>
<table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#EFF3FF">
<% out.print(sb.toString());%>
</table>
<br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
      <td width="13%" align="left">
                  Create a V-card?
               </td>
               <td width="5%" align="left">
                 <a href="VCard.jsp" target="_blank">Yes</a>
               </td>
               <td width="69%" align="left"><a
href="purAmountForm.jsp">No</a></td>
</tr>
</table>
<%  } //end of session variable
session.removeAttribute("VAResults");
%>
</form>
</body>
</html>
```

PurchaseAmt_whole

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>VA Menu</title>
</head>
<body>
<table width=100%>
   <tr>
      <td valign="top"  >

      </td>
   </tr>
   <tr>
      <td>
        <TABLE width="100%" align="center">
          <TR>
                      <TD width="20%" valign="top" align="center">
              <!-- right part of the jsp shud come here -->
              <%@ include file="right_links.jsp" %>
            </TD>
            <TD width="80%" valign="top" align="center">
                    <!-- The main middle page jsp name shud be included
here -->
                       <%@ include file="PurchaseAmt.jsp" %>
            </TD>
          </TR>
        </TABLE>

      </td>
     </tr>
  </table>
</body>
</html>
```

Registration.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--java script for client side validations-->
<!--jsp that collects the registration information-->
<script language="JavaScript">
       function evalAll() {
                      test = true;
                     if((document.all) || (document.getElementById)){
                         var userID = document.regForm.txtUserID.value;
                         var passwd =
document.regForm.txtPassword.value;
                         var passwd_retype =
document.regForm.txtPasswordRetype.value;
                         var firstName =
document.regForm.txtFirstName.value;
                         var lastName =
document.regForm.txtLastName.value;
                         var homePhone =
document.regForm.txtHomePhone.value;
                         var city =document.regForm.txtCity.value;
                         var streetAddress =
document.regForm.txtStreetAddress.value;
                         var myIndex =
document.regForm.cboState.selectedIndex;
                         var selStateValue =
document.regForm.cboState.options[myIndex].value;
                         var zipCode =document.regForm.txtZipCode.value;

                         var cityb =document.regForm.txtbCity.value;
                         var streetAddressb =
document.regForm.txtbStreetAddress.value;
                         var myIndexb =
document.regForm.cbobState.selectedIndex;
                         var selStateValueb =
document.regForm.cbobState.options[myIndexb].value;
                         var zipCodeb
=document.regForm.txtbZipCode.value;
                     }

          if (passwd!=null && passwd_retype!=null){
                         if(passwd!=passwd_retype){
                               alert('Password and Retype Password
should be same');
                               document.regForm.txtPassword.value="";

               document.regForm.txtPasswordRetype.value="";
                               document.regForm.txtPassword.focus();
                               test= false;
                         }
                     }
                     if(userID == ""){
                           alert("User ID cannot be blank");
               document.regForm.txtUserID.focus();
               test = false;
               return test;
```

```
                  }else if(passwd == ""){
                        alert("Password cannot be blank");
            document.regForm.txtPassword.focus();
            test = false;
            return test;
                  }else if(passwd_retype == ""){
                        alert("Password Retype cannot be blank");
            document.regForm.txtPasswordRetype.focus();
            test = false;
            return test;
                  }else if(firstName == ""){
                        alert("First Name cannot be blank");
            document.regForm.txtFirstName.focus();
            test = false;
            return test;
                  }else if(lastName == ""){
                        alert("Last Name cannot be blank");
            document.regForm.txtLastName.focus();
            test = false;
            return test;
                  }else if(homePhone == ""){
                        alert("Home Phone cannot be blank");
            document.regForm.txtHomePhone.focus();
            test = false;
            return test;
                  }else if(streetAddress == ""){
                        alert("Street Address cannot be blank");
            document.regForm.txtStreetAddress.focus();
            test = false;
            return test;
                  }else if(city == ""){
                        alert("City cannot be blank");
            document.regForm.txtCity.focus();
            test = false;
            return test;
                  }else if(selStateValue == ""){
                        alert("State cannot be blank");
            document.regForm.cboState.focus();
            test = false;
            return test;
                  }else if(zipCode == ""){
                        alert("Zip Code cannot be blank");
            document.regForm.txtZipCode.focus();
            test = false;
            return test;
                  }

                  return test;
      }

               function evalAddress(){
                     var city =document.regForm.txtCity.value;
                     var streetAddress =
document.regForm.txtStreetAddress.value;
                     var myIndex =
document.regForm.cboState.selectedIndex;
```

```
                        var selStateValue =
document.regForm.cboState.options[myIndex].value;
                        var zipCode =document.regForm.txtZipCode.value;

                        //alert(document.regForm.chkAddrSame.checked);
                        if (document.regForm.chkAddrSame.checked) {
                              document.regForm.txtbCity.value = city;

                              document.regForm.txtbStreetAddress.value
= streetAddress;
                              document.regForm.cbobState.selectedIndex
=  myIndex;

               document.regForm.cbobState.options[myIndex].value =
selStateValue;
                              document.regForm.txtbZipCode.value =
zipCode;
                        } else {
                              document.regForm.txtbCity.value = "";

                              document.regForm.txtbStreetAddress.value
= "";
                              document.regForm.cbobState.selectedIndex
=  "";

               document.regForm.cbobState.options[myIndex].value = "";
                              document.regForm.txtbZipCode.value = "";
                        }
                  }
                  function validateInt() {
                        var iString = document.regForm.txtZipCode.value
                        alert(iString);
                   return (("" + parseInt(iString)) == iString);
                  }
                  function AllowNumericOnly(){
                Expression = '0123456789';
                var ch = String.fromCharCode(window.event.keyCode);
                ch = ch.toLowerCase();
                var a = Expression.indexOf(ch);
                if (a == -1)
                     window.event.keyCode = 0;
       }//end function


</script>
<html>
<head>
<title>VA Card Registration</title>
</head>
<body>
<br>
<h1><font color="#08246B" size="5" face="arial,helvetica">
               Registration</font>
</h1>
<form name="regForm" method="post" action="RegistrationAction.jsp">
<br>
```

```
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
                <td width="19%" align="justify" bgcolor="#333399">
<font color="#FFFFFF" size="2" face="Arial, Helvetica"><b>Login
        Information</b></font></td>
              <td width="19%" align="justify"
bgcolor="#333399"> </td>
</tr>
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>

<tr>
              <td width="19%" align="right"><font  color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>User ID:</b></font></td>
              <td width="81%"><input name="txtUserID" type="text"
maxlength="25" size="25"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Password:</b></font></td>
              <td width="81%"><input name="txtPassword" type="password"
maxlength="25" size="25"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Retype Password:</b></font></td>
              <td width="81%"><input name="txtPasswordRetype"
type="password" maxlength="25" size="25"></td>
</tr>
</table>
<br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
              <td width="19%" align="justify" bgcolor="#333399">
                 <font color="#FFFFFF" size="2" face="Arial,
Helvetica"><b>Personal Information</b></font>
              </td>
              <td width="81%" bgcolor="#333399"> </td>
</tr>
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#000000"
size="2" face="Arial, Helvetica"><b>Prefix:</b></font></td>
```

```
              <td width="81%"><font color="#000000" size="2"
face="Arial, Helvetica">
                  <select name="cboPrefix" size="1">
                    <option selected> </option>
          <option>Mr.</option>
          <option>Mrs.</option>
          <option>Miss.</option>
        </select></font>
              </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>First Name:</b></font></td>
              <td width="81%"><input name="txtFirstName" type="text"
maxlength="40" size="40"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#000000"
size="2" face="Arial, Helvetica"><b>Middle Name:</b></font></td>
              <td width="81%"><input name="txtMiddleName" type="text"
maxlength="40" size="40"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Last Name:</b></font></td>
              <td width="81%"><input name="txtLastName" type="text"
maxlength="40" size="40"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Home Phone:</b></font></td>
              <td width="81%"><input name="txtHomePhone" type="text"
maxlength="15" size="15"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#000000"
size="2" face="Arial, Helvetica"><b>Office Phone:</b></font></td>
              <td width="81%"><input name="txtOfficePhone" type="text"
maxlength="15" size="15"></td>
</tr>
</table>
<br><br><br><br><br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
              <td width="19%" align="justify" bgcolor="#333399">
                 <font color="#FFFFFF" size="2" face="Arial,
Helvetica"><b>Address</b></font>
              </td>
              <td width="19%" align="justify"
bgcolor="#333399"> </td>
```

```
<tr>
              <td width="19%"> </td>
              <td width="81%"> </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Street Address:</b></font></td>
              <td width="81%"><input name="txtStreetAddress"
type="text" maxlength="100" size="100"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>City:</b></font></td>
              <td width="81%"><input name="txtCity" type="text"
maxlength="50" size="50"></td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>State:</b></font></td>
              <td width="81%"><font color="#000000" size="2"
face="Arial, Helvetica">
                  <select name="cboState" size="1">
                    <option selected> </option>
          <%
                        //gets the state information
                        DBConnection myConn = null;
                        ResultSet rs = null;
                              String sqlQuery = null;
                        DebugLog debug = new DebugLog("Reg.txt");
                        try{
                              myConn = new DBConnection();
                              Connection conn = myConn.getConnection();
                              Statement stmt = conn.createStatement();

                                    sqlQuery = "SELECT STATE_ID,
STATE_DESC FROM STATE";
                                    debug.append("SQL Query: " +
sqlQuery);
                                    rs = stmt.executeQuery(sqlQuery);
                                    while (rs.next()) {
                                          String stateID =
rs.getString("STATE_ID");
                                          String stateDesc =
rs.getString("STATE_DESC");
                                          debug.append("State ID: " +
stateID + "State Desc: "  + stateDesc);
                        %>
                        <option
value="<%=stateID%>"><%=stateDesc%></option>
                        <%          } //end of while
                                    rs.close();
```

```
                                    stmt.close();
                                    myConn.closeConnection();
                                }catch(Exception ex){

                                    ErrorFound ef = new ErrorFound();
                                    ef.appendError("Registration", "try
block of Address", "Exception e =" + ex.getMessage());
                                    ef.close();
                                }
                            %>
        </select></font>
              </td>
</tr>
<tr>
              <td width="19%" align="right"><font color="#FF0000"
size="2" face="Arial, Helvetica"><b>*</b></font>
              <font color="#000000" size="2" face="Arial,
Helvetica"><b>Zip Code:</b></font></td>
              <td width="81%"><input name="txtZipCode" type="text"
maxlength="5" size="5"  onKeyPress="AllowNumericOnly();"></td>
</tr>
</table>
<br><br><br><br><br><br><br><br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
                <td width="10%" align="center">
                  <input name="pagemode" type="hidden" value="submit">
                  <input type="submit" onKeyDown='return evalAll();'
value="Submit" onClick='return evalAll();' >
                </td>
                <td width="90%"><input type="reset" value="Reset"></td>
</tr>
</table>
</form>
</body>
</html>
```

RegistrationAction.jsp

```
<%@ page language="java" import="java.sql.*,common.*"%>
<html>
<head>
<title>Registration Processing</title>
</head>
<body>
<form name="regAction" action="Login.jsp">
<%
               //stores the registration information in the database
               DBConnection myConn = null;
               Connection conn = null;
    ResultSet rs = null;
               String sqlQuery = null;
    DebugLog debug = new DebugLog("RegAction.txt");
               int rowsAffected = 0, rowsAffectedUA = 0, rowsAffectedBA
= 0;
               Statement stmt = null;
               debug.append("Obtained values:");
               String userID = request.getParameter("txtUserID");
               String passwd = request.getParameter("txtPassword");
               String passwd_retype =
request.getParameter("txtPasswordRetype");
               String userPrefix = request.getParameter("cboPrefix");
               String firstName = request.getParameter("txtFirstName");
               String middleName =
request.getParameter("txtMiddleName");
               String lastName = request.getParameter("txtLastName");
               String homePhone =request.getParameter("txtHomePhone");
               String officePhone =
request.getParameter("txtOfficePhone");
               String streetAddress =
request.getParameter("txtStreetAddress");
               String city = request.getParameter("txtCity");
               String state = request.getParameter("cboState");
               String zipCode = request.getParameter("txtZipCode");
               String addrSame = request.getParameter("chkAddrSame");
               String streetbAddress =
request.getParameter("txtbStreetAddress");
               String cityb = request.getParameter("txtbCity");
               String stateb = request.getParameter("cbobState");
               String zipbCode = request.getParameter("txtbZipCode");
               boolean userFound = false;

               debug.append("User ID: " + userID);
               debug.append("Password: " + passwd);
               debug.append("Password Retype: " + passwd_retype);
               debug.append("Prefix: " + userPrefix);
               debug.append("First Name: " + firstName);
               debug.append("Middle Name: " + middleName);

               debug.append("Last Name: " + lastName);
               debug.append("Home Phone: " + homePhone);
               debug.append("Office Phone: " + officePhone);
               debug.append("Street Address: " + streetAddress);
```

```
                debug.append("City: " + city);
                debug.append("State: " + state);
                debug.append("Zip Code: " + zipCode);
                debug.append("Address Checked: " +  addrSame);

                debug.append("Billing Street Address: " +
streetbAddress);
                debug.append("Billing Address City: " + cityb);
                debug.append("Billing Address State: " + stateb);
                debug.append("Billing Address Zip Code: " + zipbCode);

    try{
                myConn = new DBConnection();
                conn = myConn.getConnection();
                stmt = conn.createStatement();

                    sqlQuery = "SELECT COUNT(*) FROM USERS WHERE USER_ID
= '" + userID + "'";
                    debug.append(sqlQuery);
                    rs = stmt.executeQuery(sqlQuery);
                    while(rs.next()){
          if( !rs.getString(1).equals("0")){
                          userFound = true;
                          session.setAttribute("USER_ID",(String)
userID);
                          debug.append("RegistrationAction : User Id :" +
userID);
                %>
                     <script language="JavaScript">
                          alert("This User ID ALREADY EXISTS! Please
enter different User ID");
                          document.regAction.action="Registration.jsp";
                        document.regAction.submit();
                     </script>

    <%     }//end of if
         } //end of while

        rs.close();

                    if (userFound == false) {

                          sqlQuery = "INSERT INTO USERS (USER_ID, PASSWD,
PASSWD_RETYPE, PREFIX, FIRST_NAME, "

+ "LAST_NAME, MIDDLE_NAME, PHONE_NUM_RES, PHONE_NUM_OFF) "
                                                    + "VALUES('"+ userID +
"','" + passwd + "','" + passwd_retype + "','" +
                                                                  userPrefix
+ "','" +  firstName + "','" + lastName  + "','" +
                                                                  middleName
+ "','" + homePhone + "','" +  officePhone + "')";
                          debug.append("SQL Query: " + sqlQuery);
                          rowsAffected = stmt.executeUpdate(sqlQuery);
                          debug.append("Rows effected for user details: "
+ rowsAffected);
```

```
                        sqlQuery = "INSERT INTO USER_ADDRESS(USER_ID,
STREET_ADDRESS, CITY, STATE_ID, " +
                                                                "ZIP_CODE)
VALUES ( '"  + userID + "','" + streetAddress + "','" +
                                                                 city +
"'," + state + "," + zipCode + ")";
                        debug.append("SQL Query: " + sqlQuery);
                        rowsAffectedUA = stmt.executeUpdate(sqlQuery);
                        debug.append("Rows effected for user address: "
+ rowsAffectedUA);
                        debug.close();
                  if (rowsAffected == 1 && rowsAffectedUA==1) {

%>
                  <h3> Hello <%=firstName%><%=lastName%></h3>
                  <input type="submit">
                  <script language="JavaScript">
                        document.regAction.action="Login.jsp";
                      document.regAction.submit();
                  </script>
<%  }else  { %>
                  <h3> There is an error in the application </h3>
<%                }//end of if rowsAffected
                   }//end of if userFound
                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("RegistrationAction", "try
block", "Exception e =" + ex.getMessage());
                        ef.close();
                  }
    userFound = false;
              stmt.close();
              myConn.closeConnection();
%>
</form>
</body>
</html>
```

right_links.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<!--VAMenu included in all the files-->
<script language="JavaScript">
</script>
<html>
<head>
<title>Virtual Agent Menu</title>
</head>
<body>
<form name="rightLinkForm" target="_parent">
<TABLE border=0 cellpadding=0 cellspacing=0 width="200">
<% if (session.getAttribute("userName") != null){ %>
              <tr>
              <td width="100%"><font color="#000066" size="3"
face="Arial, Helvetica">
                          Hello,
<%=session.getAttribute("userName")%></font></td>
              </tr>
<% } %>
</table>
<TABLE border=0 cellpadding=0 cellspacing=0 width="200">
  <tbody>
   <TR><TD colspan="4"> </TD></TR>
  <TR><TD colspan="4"> </TD></TR>
  <TR>
    <TD bgcolor="#01669A" colspan="4">
         <IMG height="1" src="bit.gif" width="1" alt="."></TD></TR>
  <TR >
    <TD bgcolor="#01669A" colspan="4"><font color="white"><strong>VA
Menu</strong></font></TD>
  </TR>
  <TR>
    <TD bgcolor="#01669A" width="1"><IMG height=1 src="bit.gif"
width="1" alt="."></TD>
    <TD bgcolor="#FFFFFF"><IMG height=1 src="bit.gif" width="4"
alt="."></TD>
    <TD bgcolor="#FFFFFF" >
                 <br>
                 <br>
                 <br>
       <IMG border=0 height=10 hspace=3 src="arrow.gif" width="5"
alt=".">
       <a href="CardList_whole.jsp">My Card List</a><br>
       <IMG border=0 height=10 hspace=3 src="arrow.gif" width="5"
alt=".">
       <a href="Preference_whole.jsp">My Preference</a><BR>
       <IMG border=0 height=10 hspace=3 src="arrow.gif" width="5"
alt=".">
       <a href="PurchaseAmt_whole.jsp">Create A Virtual Card</a><BR>
       <IMG border=0 height=10 hspace=3 src="arrow.gif" width="5"
alt=".">
                <a href="Logout.jsp">Logout</a>
                <br>
                <br>
```

```
                 <br>
                 <br>
                 <br>
                 <br>
                 <br>
                 <br>
                 <br>
                 <br>
         <br>
                 <br>
                 <br>
                 <br>
                 <br>
         <br>
                 <br>
                 <br>


    </TD>
    <TD bgcolor="#01669A" width="1"><IMG height="1" src="bit.gif"
width="1" alt="."></TD>
  </TR>
  <TR>
    <TD bgcolor="#01669A" colspan="4"><IMG height="1" src="bit.gif"
width="1" alt="."></TD>
  </tr>
  </tbody>
</TABLE>
</form>
</body>
</html>
```

VAMenu_whole.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>VA Menu</title>
</head>
<body>
<table width=100%>
   <tr>
      <td valign="top"  >

      </td>
   </tr>
   <tr>
      <td>
        <TABLE width="100%" align="center">
          <TR>
                      <TD width="20%" valign="top" align="center">
              <!-- right part of the jsp shud come here -->
              <%@ include file="right_links.jsp" %>
            </TD>
            <TD width="80%" valign="top" align="center">
                    <!-- The main middle page jsp name shud be included
here -->
                        
            </TD>
          </TR>
        </TABLE>

      </td>
     </tr>
  </table>
</body>
</html>
```

VCard.jsp

```
<%@ page language = "java" import = "java.sql.*,common.*"%>
<script language="JavaScript">
</script>
<html>
<head>
<title>Purchase Amount Information</title>
</head>
<body>
<form name="purAmountForm" method="post" action="Bank1.jsp">
<%
              //generates pop-up virtual agent screen with the details
from the database
              DBConnection myConn = null;
              Connection conn = null;
    ResultSet rs = null;
              String sqlQuery = null;
    DebugLog debug = new DebugLog("VCard.txt");
              int rowsAffected = 0, rowsAffectedBA = 0;
              Statement stmt = null;
              debug.append("Obtained values:");

              String userId = null;
              int vcardNum =0;
              String expiryDate = null;
              int amountCharged = 0;


    try{
              myConn = new DBConnection();
              conn = myConn.getConnection();
              stmt = conn.createStatement();

        if (session.getAttribute("userID") != null) {
                        userId = (String)
session.getAttribute("userID");
                  }

                  sqlQuery = "SELECT VCARDNUM, EXPIRY_DATE,
AMOUNT_CHARGED " +
                        "FROM VCARD WHERE USER_ID= '" + userId + "'";
                  debug.append("SQL Query: " + sqlQuery);
                  rs = stmt.executeQuery(sqlQuery);
                  while(rs.next()){
                        vcardNum = rs.getInt("VCARDNUM");
                        expiryDate = rs.getString("EXPIRY_DATE");
                        amountCharged = rs.getInt("AMOUNT_CHARGED");
                  }
                        rs.close();

                  }catch(Exception ex){
                        ErrorFound ef = new ErrorFound();
                        ef.appendError("PreferenceAction", "try block",
"Exception e =" + ex.getMessage());
                        ef.close();
```

```
                }
                debug.close();
    stmt.close();
                myConn.closeConnection();
%>
<table WIDTH="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#FFFFFF" bgcolor="#FFFFFF">
    <tr>
                <td align="center"><strong>A Virtual Card</strong></td>
                </tr>
                <tr>
      <td colspan="4"> </td>
                </tr>
                <tr>
                    <td width="19%" align="right" ><font color="#000000"
size="2" face="Arial, Helvetica"><b>Card Number:</b></font></td>
                    <td width="81%"><input name="txtcardNumber"
type="text" maxlength="50" size="50" value="<%=vcardNum%>"></td>
                </tr>
                <tr>
                    <td width="19%" align="right" ><font color="#000000"
size="2" face="Arial, Helvetica"><b>Expiry Date:</b></font></td>
                    <td width="81%"><input name="txtexpiryDate"
type="text" maxlength="50" size="50" value="<%=expiryDate%>"></td>
                </tr>
                <tr>
                    <td width="19%" align="right" ><font color="#000000"
size="2" face="Arial, Helvetica"><b>Amount Charged:</b></font></td>
                    <td width="81%"><input name="txtamtCharged"
type="text" maxlength="50" size="50" value="<%=amountCharged%>"></td>
                </tr>
</table>
<br><br>
<table width="100%" border="0" align="left" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
<tr>
      <td width="11%" align="left">
                  Use this card?
                </td>
                <td width="7%" align="left">
                  <a href="Bank1.jsp" target="_blank">Confirm</a>
                </td>
                <td width="72%" align="left"><a
href="VCard.jsp">Cancel</a></td>
</tr>
</table>
</form>
</body>
</html>
```