TRANSACTION FRAUD DETECTION BASED ON SPENDING PROFILES USING HIDDEN MARKOV MODEL

A PROJECT REPORT

Submitted by

ANUSUYA.E(810015104010)

JAYASUDHA.R(810015104029)

in partial fulfillment for the award of the degree
of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING

BIT CAMPUS,TIRUCHIRAPPALLI

ANNA UNIVERSITY::CHENNAI 600 025

APRIL 2019

UNIVERSITY COLLEGE OF ENGINEERING,
BIT CAMPUS,
TIRUCHIRAPPALLI-620 024

BONAFIDE CERTIFICATE

Certified that this project report ”**Transaction Fraud detection Based on Spending profiles Using Hidden Markov Model**” is the bonafide work of “ **Ms. E.ANUSUYA (810015104010)** and **Ms R.JAYASUDHA (810015104029)** “ who carried out the project work under my supervision.

SIGNATURE
Dr. D. Venkatesan
HEAD OF THE DEPARTMENT
Assistant Professor
Computer Science & Engineering
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

SIGNATURE
Ms.N.Vivekapriya
SUPERVISOR
Assistant Professor
Computer Science & Engineering
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

Submitted for the project Viva voice examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare the work entitled **"TRANSACTION FRAUD DETECTION BASED ON SPENDING PROFILES USING HIDDEN MARKOV MODEL"** is submitted in partial fulfillment of the requirement for the award of the degree in B.E., Computer Science and Engineering, University College of Engineering(BIT Campus), Tiruchirappalli, is a record of our own work carried out by us during the academic year 2018-2019 under the supervision and guidance of Ms.N.Vivekapriya, Assistant professor, Department of Computer Science and Engineering, University College of Engineering(BIT Campus), Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any degree, either in this or any other University.

SIGNATURE OF THE CANDIDATES

E.ANUSUYA (810015104010)

R.JAYASUDHA (810015104029)

I certify that the declaration made above by the candidate is true.

SIGNATURE OF THE GUIDE
Ms.N.VIVEKAPRIYA
Assistant professor,
Department of CSE,
University College of Engineering,
BIT Campus, Anna University,
Tiruchirappalli-620 024.

ACKNOWLEDGEMENT

I would like to convey my heartfelt thanks to our honorable Dean **Dr. T. SENTHILKUMAR,** Associate Professor for having provided me with all required facilities to complete my project without hurdles.

I would like to express my sincere thanks and deep sense of gratitude to guide **Dr. D. VENKATESAN,** Assistant Professor and Head, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

I would like to thank my project guide **Ms.N.VIVEKAPRIYA,** Assistant Professor, Department of Computer Science and Engineering, for his valuable guidance throughout the phase of the project. It is our responsibility to thank our project coordinator **Mr. C. SANKAR RAM,** Assistant Professor, Department of Computer science and Engineering for his constant inspiration that he has all through the project period.

I would like to thank **Mr. C. SURESH KUMAR,** Teaching Fellow, Department of Computer Science and Engineering, for his encouragement for this work.

I extend my thanks to all other teaching and non-teaching staffs for their encouragement and support.

I thank my beloved parents and friends, for their full support in my career development of this project.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF FIGURES	iv
	LIST OF ABBREVIATIONS	v
1	**INTRODUCTION**	1
2	**LITERATURE SURVEY**	2
3	**SYSTEM ANALYSIS**	5
	3.1 EXISTING SYSTEM	5
	3.2PROPOSED SYSTEM	5
	3.3ADVANTAGES	6
4	**SYSTEM SPECIFICATION**	7
	4.1 HARDWARE REQUIREMENTS	7
	4.2 SOFTWARE REQUIREMENTS	7
	4.3 ABOUT THE SOFTWARE	8
5	**SYSTEM DESIGN**	9
	5.1 SYSTEM ARCHITECTURE	9
	5.2 MODULES	9
	5.3 UML DIAGRAMS	11
	5.3.1 USE CASE DIAGRAM	11
	5.3.2 CLASS DIAGRAM	12
	5.3.3 SEQUENCE DIAGRAM	12
	5.3.4 COMPONENT DIAGRAM	13

	5.1 SYSTEM ARCHITECTURE	9
	5.2 MODULES	9
	5.3 UML DIAGRAMS	11
	5.3.1 USE CASE DIAGRAM	11
	5.3.2 CLASS DIAGRAM	12
	5.3.3 SEQUENCE DIAGRAM	12
	5.3.4 COMPONENT DIAGRAM	13
6	**TECHNOLOGY SPECIFICATIONS**	14
	6.1 HMM MODEL	15
	6.2 DATABASE	15
	6.3 SQL	16
	6.4 JAVA	16
7	**SOURCE CODE**	17
8	**SCREENSHOTS**	26
9	**CONCLUSION AND FUTUREWORKS**	35
	9.1 Conclusion	35
	9.2 Future Works	35
10	**REFERENCES**	36

ABSTRACT

Due to a rapid advancement in the electronic commerce technology, the use of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In this paper, we model the sequence of operations in credit card transaction processing using a Hidden Markov Model(HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected. We present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
5.1	SYSTEM ARCHITECTURE	9
5.2	UML DIAGRAMS	11
5.2.1	USE CASE DIAGRAM	11
5.2.2	CLASS DIAGRAM	12
5.2.3	SEQUENCE DIAGRAM	12
5.2.4	COMPONENT DIAGRAM	13

LIST OF ABBREVIATIONS

HMM	Hidden Markov Model
FDS	Fraud Detection System
SP	Spending Profiles
BP	Behavior Profiles
TPR	True Positive Rate
FPR	False Positive Rate
OM	Observation Medium
SQL	Structured Query Language
WAMP	Windows, Apache, MYSQL and PHP

CHAPTER 1

INTRODUCTION

Credit-card-based purchases can be categorized into two types are physical card and virtual card. In a physical-card based purchase, the cardholder presents his card physically to a merchant for making a payment. To carry out fraudulent transactions in this kind of purchase, an attacker has to steal the credit card. If the cardholder does not realize the loss of card, it can lead to a substantial financial loss to the credit card company. In the second kind of purchase, only some important information about a card (card number, expiration date, secure code) is required to make the payment. Such purchases are normally done on the Internet or over the telephone. To commit fraud in these types of purchases, a fraudster simply needs to know the card details. Most of the time, the genuine cardholder is not aware that someone else has seen or stolen his card information. The only way to detect this kind of fraud is to analyze the spending patterns on every card and to figure out any inconsistency with respect to the "usual" spending patterns. Fraud detection based on the analysis of existing purchase data of cardholder is a promising way to reduce the rate of successful credit card frauds. Since humans tend to exhibit specific behaviorist profiles, every cardholder can be represented by a set of patterns containing information about the typical purchase category, the time since the last purchase, the amount of money spent, etc. Deviation from such patterns is a potential threat to the system.

CHAPTER 2

LITERATURE SURVEY

Ghosh and Reilly have proposed credit card fraud detection with a neural network:

They have built a detection system, which is trained on a large sample of labeled credit card account transactions. These transactions contain example fraud cases due to lost cards, stolen cards, application fraud, counterfeit fraud, mail-order fraud, and nonreceived issue (NRI) fraud**.**

Syeda et al. have used parallel granular neural networks (PGNNs) for improving the speed of data mining and knowledge discovery process in credit card fraud detection:

A complete system has been implemented for this purpose. Stolfo et al. suggest a credit card fraud detection system (FDS) using metalearning techniques to learn models of fraudulent credit card transactions. Metalearning is a general strategy that provides a means for combining and integrating a number of separately built classifiers or models. A metaclassifier is thus trained on the correlation of the predictions of the base classifiers. The same group has also worked on a cost-based model for fraud and intrusion detection . They use Java agents for Metalearning (JAM), which is a distributed data mining system for credit card fraud detection. A number of important performance metrics like True Positive—False Positive (TP-FP) spread and accuracy have been defined by them. Aleskerov et al. present CARDWATCH, a database mining system used for credit card fraud detection. The system, based on a neural learning module, provides an interface to a variety of commercial databases**.**

Kim and Kim have identified skewed distribution of data and mix of legitimate and fraudulent transactions as the two main reasons for the cmplexity of credit card fraud detection:

Based on this observation, they use fraud density of real transaction data as a confidence value and generate the weighted fraud score to reduce the number of misdetections. Fan et al. suggest the application of distributed data mining in credit card fraud detection. Brause et al. have developed an approach that involves advanced data mining techniques and neural network algorithms to obtain high fraud coverage.

Chiu and Tsai have proposed Web services and data mining techniques to establish a collaborative scheme for fraud detection in the banking industry:

With this scheme, participating banks share knowledge about the fraud patterns in a heterogeneous and distributed environment. To establish a smooth channel of data exchange, Web services techniques such as XML, SOAP, and WSDL are used. Phua et al. have done an extensive survey of existing data-mining-based FDSs and published a comprehensive report. Prodromidis and Stolfo use an agent-based approach with distributed learning for detecting frauds in credit card transactions. It is based on artificial intelligence and combines inductive learning algorithms and metalearning methods for achieving higher accuracy. Phua .suggest the use of meta classifier similar to in fraud detection problems. They consider naive Bayesian, C4.5, and Back Propagation neural networks as the base classifiers. A metaclassifier is used to determine which classifier should be considered based on skewness of data. Although they do not directly use credit card fraud detection as the target application, their approach is quite generic.

Vatsa et al. have recently proposed a game-theoretic approach to credit card fraud detection:

They model the interaction between an attacker is tage game between two players, each trying to maximize his payoff. The problem with most of the abovementioned approaches is that they require labeled data for both genuine,as well as fraudulent transactions, to train the classifiers.

Getting real-world fraud data is one of the biggest problems associated with credit card fraud detection:

Also, these approaches cannot detect new kinds of frauds for which labeled data is not available. In contrast, we present a Hidden Markov Model (HMM)-based credit card FDS, which doesnot require fraud signatures and yet is able to detect frauds by considering a cardholder's spending habit. We model a credit card transaction processing sequence by the stochastic process of an HMM. The details of items purchased in individual transactions are usually not known to an FDS running at the bank that issues credit cards to the cardholders. This can be represented as the underlying finite Markov chain, which is not observable. The transactions can only be observed through the other stochastic process that produces the sequence of the amount of money spent in each transaction. Hence, we feel that HMM is an ideal choice for addressing this problem. Another important advantage of the HMM-based approach is a drastic reduction in the number of False Positives (FPs)—transactions identified as malicious by an FDS although they are actually genuine. Since the number of genuine transactions is a few orders of magnitude higher than the number of malicious transactions, an FDS should be designed in such a way that the number of FPs is as low as possible.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

In case of the existing system the fraud is detected after the fraud is done that is, the fraud is detected after the complaint of the card holder. And so the card holder faced a lot of trouble before the investigation finish. And also as all the transaction is maintained in a log, we need to maintain a huge data. And also now a days lot of online purchase are made so we don't know the person how is using the card online, we just capture the IP address for verification purpose. So there need a help from the cyber crime to investigate the fraud. To avoid the entire above disadvantage we propose the system to detect the fraud in a best and easy way.

3.2 PROPOSED SYSTEM:

In proposed system, we present a Hidden Markov Model (HMM).Which does not require fraud signatures and yet is able to detect frauds by considering a cardholder's spending habit. Card transaction processing sequence by the stochastic process of an HMM. The details of items purchased in Individual transactions are usually not known to an FDS running at the bank that issues credit cards to the cardholders. Hence, we feel that HMM is an ideal choice for addressing this problem. Another important advantage of the HMM-based approach is a drastic reduction in the number of False Positives transactions identified as malicious by an FDS although they are actually genuine. An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification. FDS receives the card details and the value of purchase to verify, whether the transaction is genuine or not. The types of goods that are bought in that transaction are not known to the FDS. It tries to find any anomaly in the transaction based on the spending profile of the cardholder, shipping address, and billing address, etc.

3.3 ADVANTAGES:

Highly Security from unauthorized use of credit card

1. Avoids fraud usage of card through online transactions.

2. Detect if card used by others if card lost.

CHAPTER 4

SYSTEM SPECIFCATION

4.1 Hardware Requirements:

- Processor : Pentium III.
- Hard Disk : 20 GB.
- Floppy Drive : 1.44 MB.
- Monitor : SVGA.
- Mouse : Two or Three Button Mouse.
- RAM : 256 MB(min).
- Keyboard : Standard Windows Keyboard.

4.2 Software Requirements:

- Operating system : Windows 10.
- Coding Language : Java.
- Sql server (wamp server).
- Netbeans

4.3 About The Software

Netbeans:

- Netbeans is an integrated development environment (IDE) for java. Netbeans allows application to be developed from a set of modular software components called modules.
- Netbeans runs on Windows, macOS, Linux and Solaris. In addition to java development, it has extensions for other languages like PHP, C, C++, HTML5 and JavaScript.
- Applications based on Netbeans, including the Netbeans IDE, can be extended by third party developers.

- Netbeans IDE provides different views of your data, from multiple project windows to helpful tools for setting up your applications and managing them efficiently.
- It is free and open source and has a large community of users and developers around the world.
- The Netbeans editor indents lines, matches words and brackets and highlight source code syntactically and semantically.

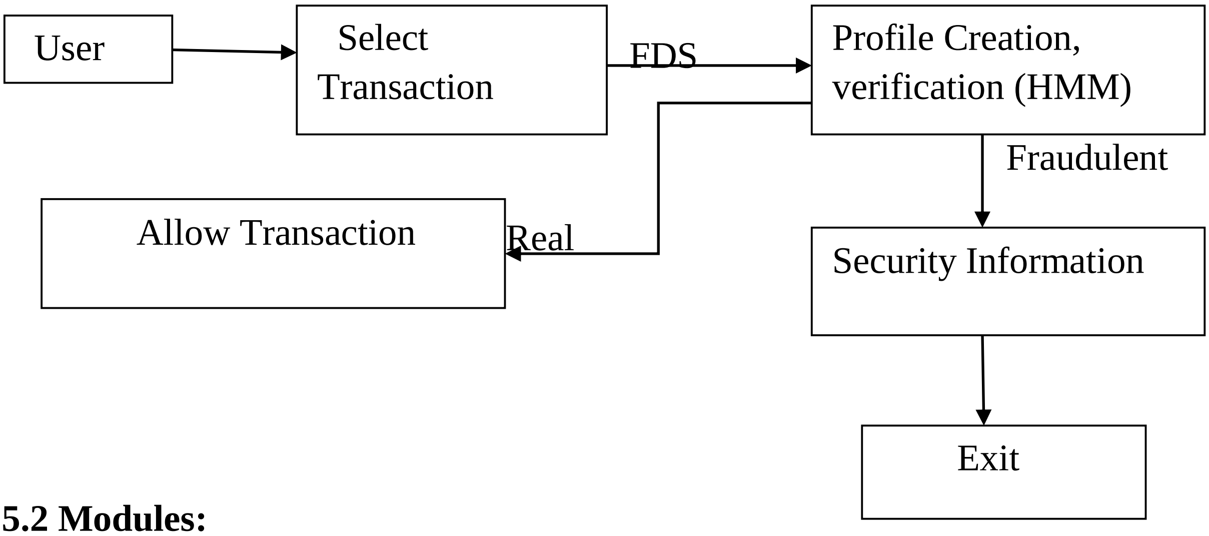
WAMPSERVER:

- WampServer refers to a software stack for the Microsoft windows operating system created by Romain Bourdon and consisting of the apache web server, open SSL for SSL support, MYSQL database and PHP programming language.
- It is often used for web development and internal testing, but may also be used to serve live websites.
- WAMP also includes MYSQL and PHP, which are two of the most common technologies used for creating dynamic websites.
- MYSQL is a high speed database while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.
- By running a local apache web server on a windows machine, a web developer can test web pages in a web browser without publishing them live on the internet.

CHAPTER 5

SYSTEM DESIGN

5.1 System Architecture:



5.2 Modules:

- **New card**
- **Login**
- **Security information**
- **Transaction**
- **Verification**

Module Description

New card

In this module, the customer gives there information to enroll a new card. The information is all about their contact details. They can create their own login and password for their future use of the card.

Login

In Login Form module presents site visitors with a form with username and password fields. If the user enters a valid username/password combination they will be granted access to additional resources on website. Which additional resources they will have access to can be configured separately.

Security information

In Security information module it will get the information detail and its store’s in database. If the card lost then the Security information module form arise. It has a set of question where the user has to answer the correctly to move to the transaction section. It contain informational privacy and informational self-determination are addressed squarely by the invention affording persons and entities a trusted means to user, secure, search, process, and exchange personal and/or confidential information.

Transaction

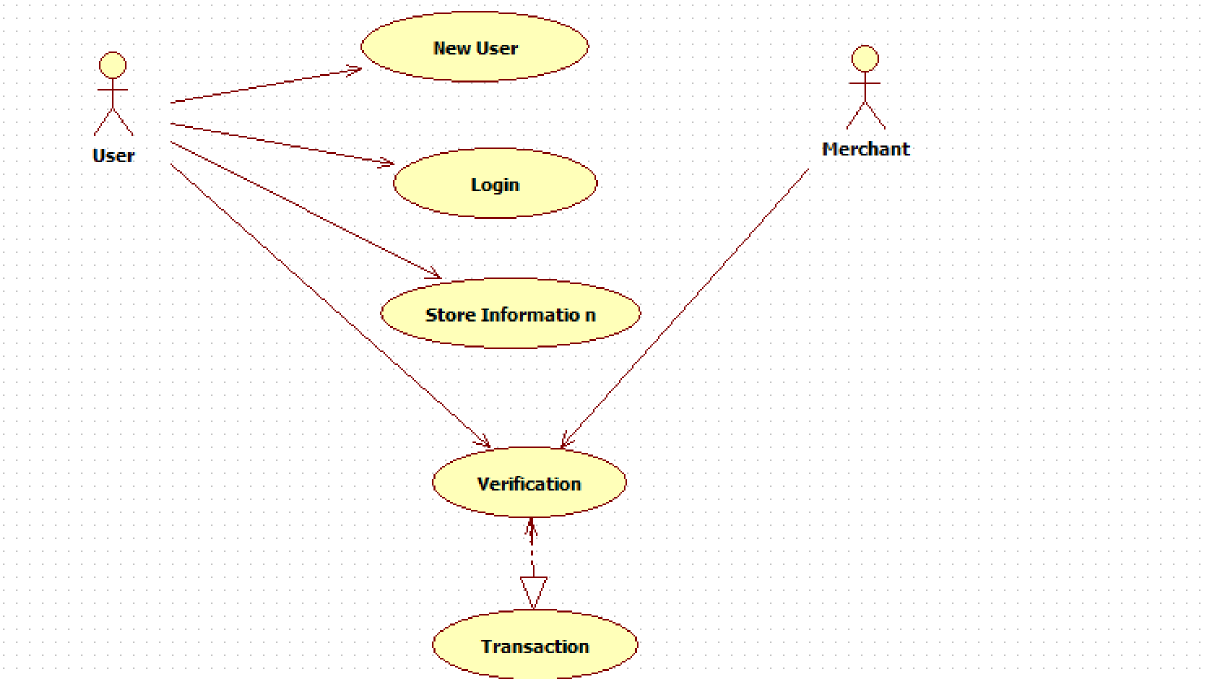
The method and apparatus for pre-authorizing transactions includes providing a communications device to a vendor and a credit card owner. The credit card owner initiates a credit card transaction by communicating to a credit card number, and storing therein, a distinguishing piece of information that characterizes a specific transaction to be made by an authorized user of the credit card at a later time. The information is accepted as "network data" in the data base only if a correct personal identification code (PIC) is used with the communication. The "network data" will serve to later authorize that specific transaction. The credit card owner or other authorized user can then only make that specific transaction with the credit card. Because the transaction is pre-authorized, the vendor does not need to see or transmit a PIC.

Verification

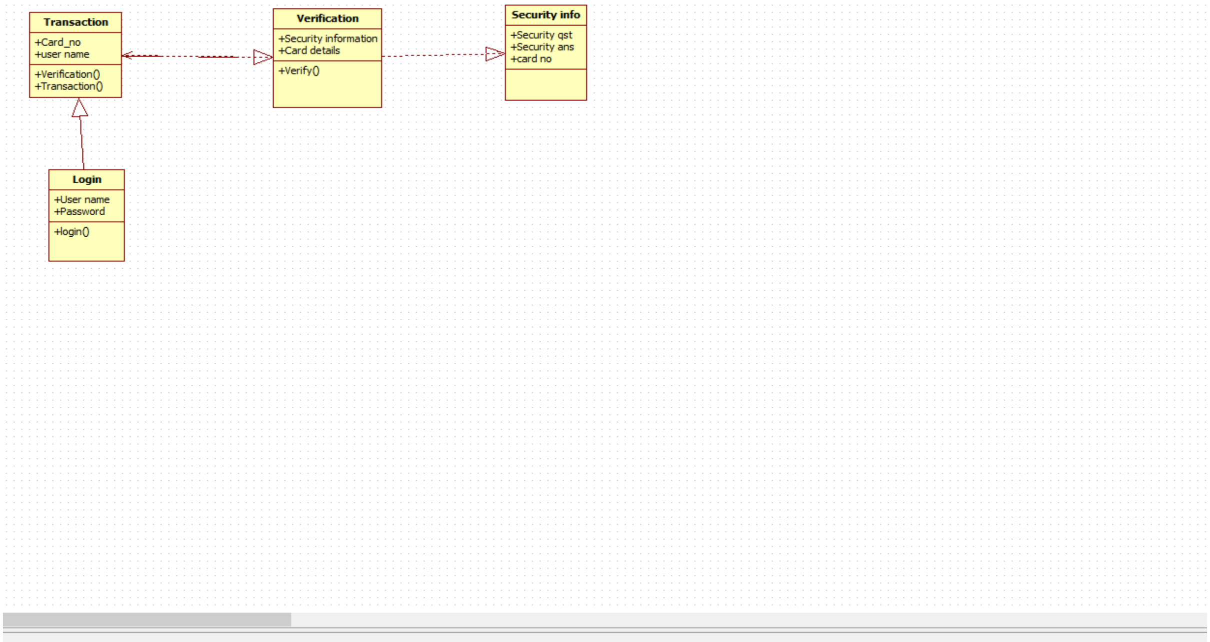
Verification information is provided with respect to a transaction between an initiating party and a verification-seeking party, the verification information being given by a third, verifying party, based on confidential information in the possession of the initiating party. In verification the process will seeks card number and if the card number is correct the relevant process will be executed. If the number is wrong, mail will be sent to the user saying the card no has been block and he can't do the further transaction.

5.3 UML Diagrams:

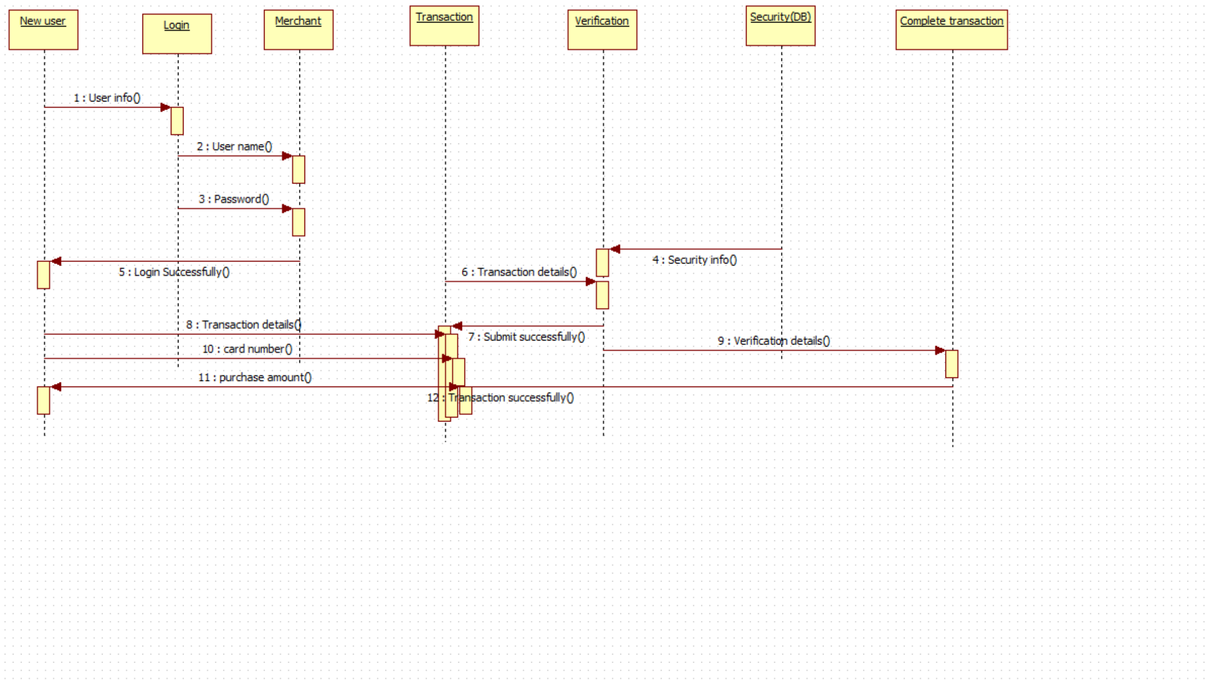
5.3.1 Use case Diagram:



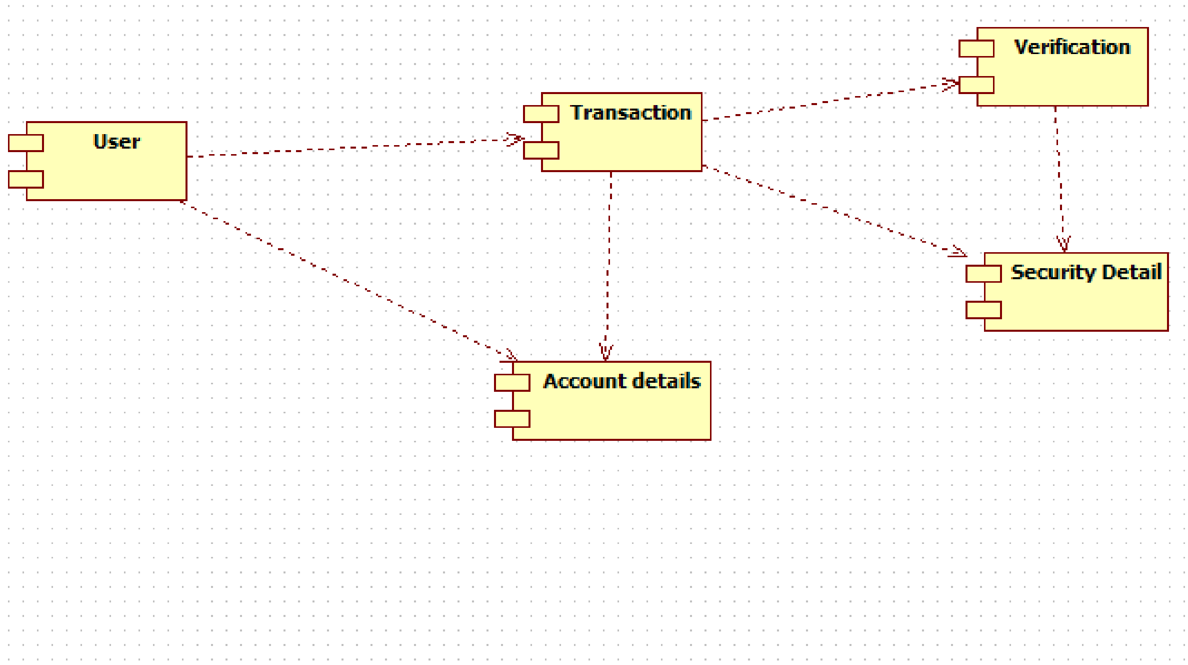
5.3.2 Class Diagram:



5.3.3 Sequence Diagram:



5.3.4 Component Diagram:



CHAPTER 6

TECHNOLOGY SPECIFICATIONS

6.1 HMM Model:

- To map the credit card transaction processing operation in terms of an HMM, we start by first deciding the observation symbols in our model. We quantize the purchase values x into M price ranges V1; V2; . . . VM, forming the observation symbols at the issuing bank. The actual price range for each symbol is configurable based on the spending habit of individual cardholders. These price ranges can be determined dynamically by applying a clustering algorithm on the values of each cardholder’s transactions, as shown in Section 5.2. We use Vk, k ¼ 1; 2; . . . M, to represent both the observation symbol, as well as the corresponding price range.
- In this work, we consider only three price ranges, namely, low (l), medium (m), and high (h). Our set of observation symbols is, therefore, V ¼ fl; m; hg making M ¼ 3. For example, let l= (0, $100], m = ($100, $500], and h= ($500, credit card limit]. If a cardholder performs a transaction of $190, then the corresponding observation symbol is m.
- A credit cardholder makes different kinds of purchases of different amounts over a period of time. One possibility is to consider the sequence of transaction amounts and look for deviations in them. However, the sequence of types of purchase is more stable compared to the sequence of transaction amounts. The reason is that, a cardholder makes purchases depending on his need for procuring different types of items over a period of time. This, in turn, generates a sequence of transaction amounts. Each individual transaction amount usually depends on the corresponding type of purchase. Hence, we consider the transition in the type of purchase as state transition in our model. The type of each purchase is linked to the line of business of the corresponding merchant. This information about

the merchant’s line of business is not known to the issuing bank running the FDS. Thus, the type of purchase of the cardholder is hidden from the FDS. The set of all possible types of purchase and, equivalently, the set of all possible lines of business of merchants forms the set of hidden states of the HMM. It should be noted at this stage that the line of business of the merchant is known to the acquiring bank, since this information is furnished at the time of registration of a merchant. Also, some merchants may be dealing in various types of commodities (For example, Wal-Mart, K-Mart, or Target sells tens of thousands of different items). Such types of line of business are considered as Miscellaneous, and we do not attempt to determine the actual types of items purchased in these transactions. Any assumption about availability of this information with the issuing bank and, hence, with the FDS, is not practical and, therefore, would not have been valid.

6.2 Database:

The MYSQL database has became the world’s most popular open source database because of its consistent fast performance, high reliability and ease of use. It’s used on every continent – Yes, even Antarctica!—by individual web developers as well as many of the world’s largest and fastest-growing organisation to save time and money powering their high-volume websites, business-critical systems and packaged software—including industry leaders such as Yahoo!.

In the medium range, MYSQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with giga bytes of memory.

6.3 Structured Query Language:

To work with data in a database, you have to use a set of commands and statements (language) defined by the DBMS software. Several different languages can be used with relational database; the most common is SQL.The American National Standards Institute and the International

Standards Organisation(ISO) define software standards, including standards for the SQL language.SQL server2012 supports the entry level of SQL-92,the SQL standard published by ANSI and ISO in 1992.The dialect of SQL supported by microsoftSQL server is called Transact-SQL(T-SQL).T-SQL is the primary language used by Microsoft SQL server applications.

6.4 Java:

- Java is a general purpose computer programming language that is concurrent, class based, object oriented, and specifically designed to have as few implementation dependencies as possible.
- It is intended to let application developers “write once, run anywhere” (WORA), meaning that compiled java code can run on all platforms that support java without the need for recompilation.
- Java applications are typically compiled to “Byte code “that can run on any java virtual machine (JVM) regardless of the underlying computer architecture.
- The original and reference implementation java compilers, virtual machines and class libraries where originally released by Sun under proprietary licences.

APPENDIX

SOURCE CODE

Home.java

```
package creditcard;

public class home extends javax. swing. Frame {

    public home() {

initComponents ();

private void initComponents ()

 jPanel1 = new javax. swing. JPanel ();

  jPanel2 = new javax. swing.JPanel();

  jLabel1 = new javax. swing. JLabel();

   jPanel3 = new javax. swing. JPanel ();

   jLabel2 = new javax.swing. JLabel();

   jButton1 = new javax.swing.JButton();

   jButton2 = new javax. Swing.JButton();

   jLabel4 = new javax.swing.JLabel();

 setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

       setMinimumSize(new java.awt.Dimension(617, 489));
jPanel1.setLayout(null);

  jPanel2.setBackground(new java.awt.Color(204, 204, 204));
jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18));
jLabel1.setForeground(new java.awt.Color(0, 51, 204));
jLabel1.setText("Credit Card Fraud Detection Using Hidden Markov Model");
```

```
javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);

 jPanel2.setLayout(jPanel2Layout);

      jPanel2Layout.setHorizontalGroup(


jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

           .addGroup(jPanel2Layout.createSequentialGroup()

               .addGap(73, 73, 73)

               .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 497,
javax.swing.GroupLayout.PREFERRED_SIZE)

               .addContainerGap(18, Short.MAX_VALUE))

       );

       jPanel2Layout.setVerticalGroup(


jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

           .addGroup(jPanel2Layout.createSequentialGroup()

               .addContainerGap()

               .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE, 38, Short.MAX_VALUE)

               .addContainerGap())

       );



       jPanel1.add(jPanel2);
```

```
        jPanel2.setBounds(10, 11, 592, 64);


        jPanel3.setBorder(new javax.swing.border.SoftBevelBorder(0));


        jLabel2.setIcon(new
javax.swing.ImageIcon("C:\\Users\\ETPL21\\Documents\\NetBeansProjects\\cr
ditcard\\src\\crditcard\\index.jpg"));


        jButton1.setFont(new java.awt.Font("Times New Roman", 1, 18));
        jButton1.setForeground(new java.awt.Color(153, 0, 0));
        jButton1.setText("SIGN IN");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });


        jButton2.setFont(new java.awt.Font("Times New Roman", 1, 18));
        jButton2.setForeground(new java.awt.Color(153, 0, 0));
        jButton2.setText("SIGN UP");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
```

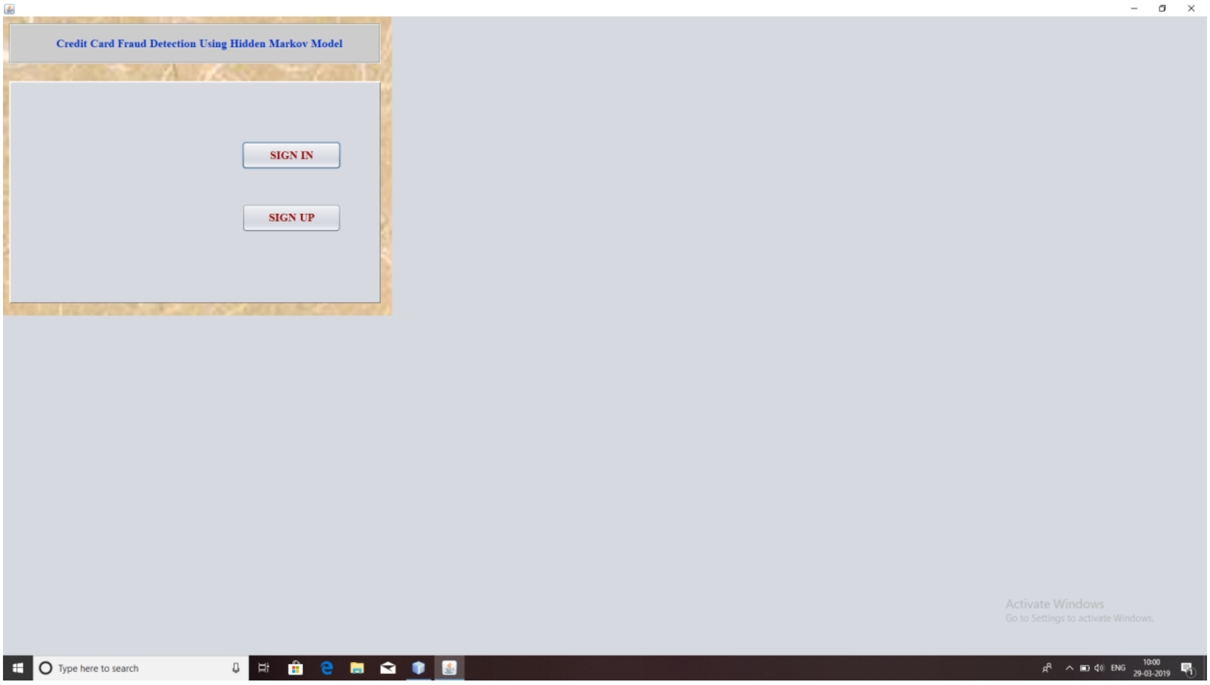
```
          }

        });



        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);

        jPanel3.setLayout(jPanel3Layout);

        jPanel3Layout.setHorizontalGroup(


jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

            .addGroup(jPanel3Layout.createSequentialGroup()

                .addGap(54, 54, 54)

                .addComponent(jLabel2)


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
71, Short.MAX_VALUE)


.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)

                    .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                    .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE, 158, Short.MAX_VALUE))

                .addGap(60, 60, 60))

        );

        jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)

            .addGroup(jPanel3Layout.createSequentialGroup()

                .addContainerGap(72, Short.MAX_VALUE)


.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

                        .addComponent(jLabel2)

                        .addGap(67, 67, 67))

                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

                        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE,                              44,
javax.swing.GroupLayout.PREFERRED_SIZE)

                        .addGap(55, 55, 55)

                        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE,                              45,
javax.swing.GroupLayout.PREFERRED_SIZE)

                        .addGap(110, 110, 110))))

        );



        jPanel1.add(jPanel3);

        jPanel3.setBounds(10, 103, 592, 353);
        jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/crditcard/bk.jpg")));

        jPanel1.add(jLabel4);

        jLabel4.setBounds(0, 0, 620, 480);


        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
617, Short.MAX_VALUE)

        );

        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 476,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(0, 2, Short.MAX_VALUE))

        );


        pack();
```

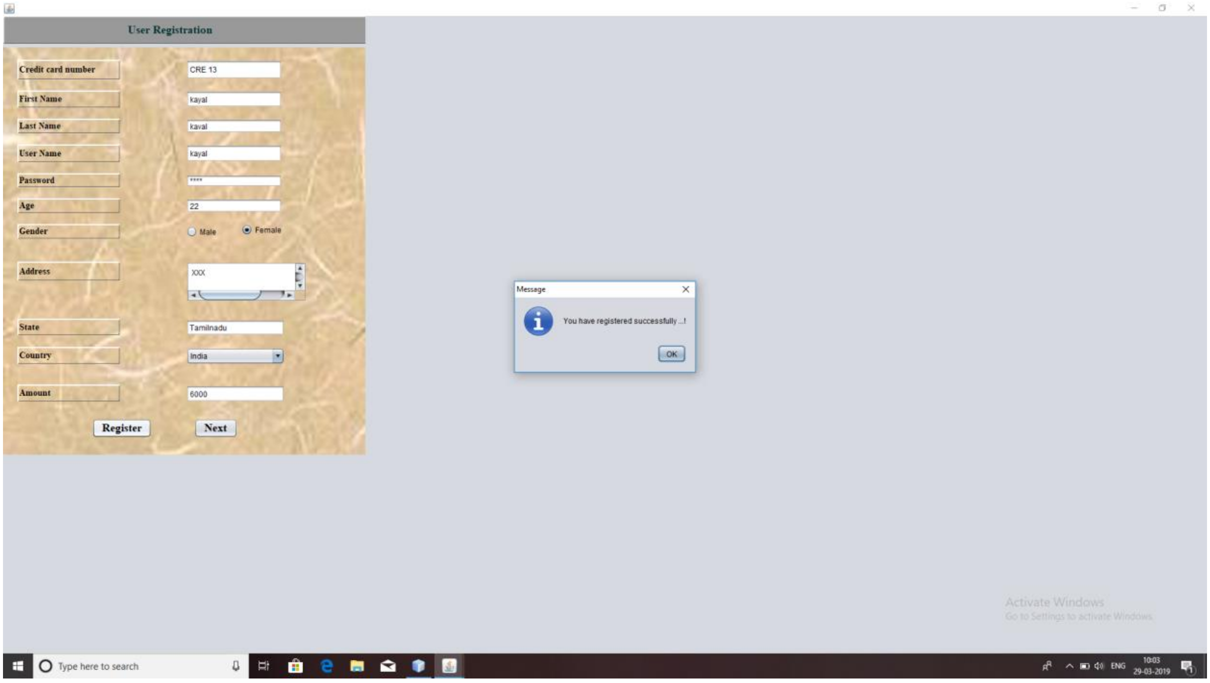
```
}


private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {


     second st=new second();

    st.setVisible(true);

}


private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

    first fst=new first();

    fst.setVisible(true);


}

Public static void main(String args[]) {

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);

        }



        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new home().setVisible(true);

            }

        });

    }



    private javax.swing.JButton jButton1;

    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;

    private javax.swing.JLabel jLabel2;

    private javax.swing.JLabel jLabel4;

    private javax.swing.JPanel jPanel1;

    private javax.swing.JPanel jPanel2;

    private javax.swing.JPanel jPanel3;


}
```

CHAPTER 8

SCREENSHOTS


Credit Card Fraud Detection Using Hidden Markov Model
SIGN IN
SIGN UP
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
10:00
29-03-2019

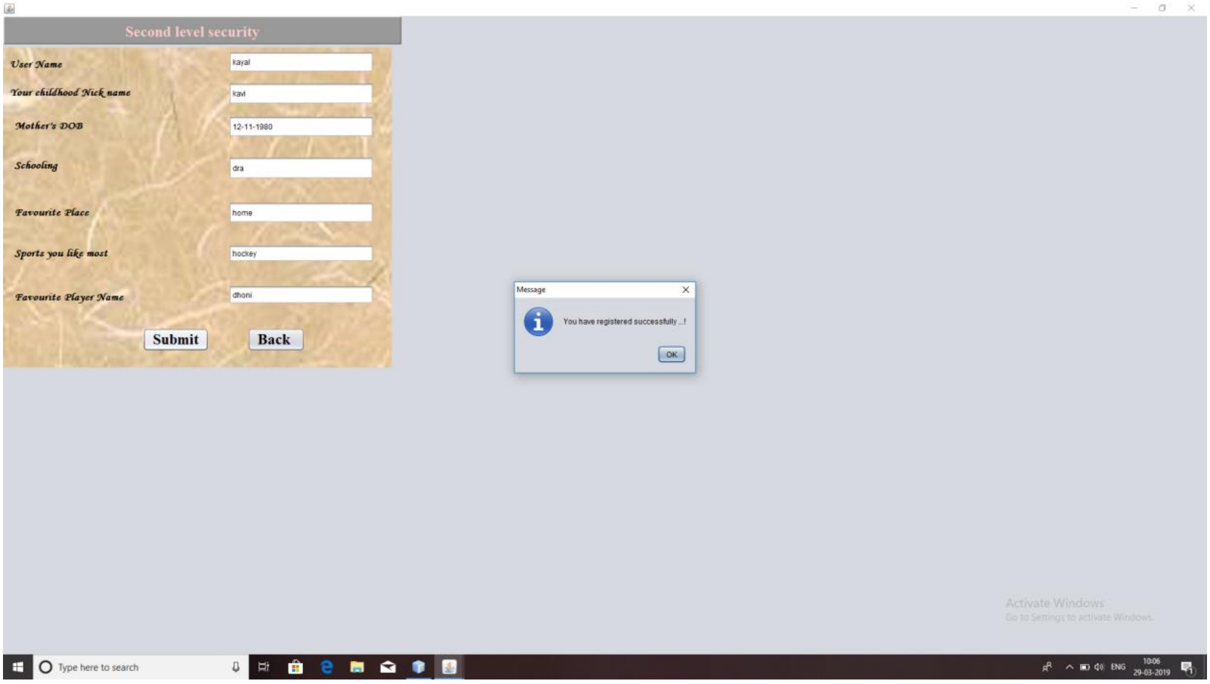


User Registration

Credit card number	CRE 13
First Name	kayal
Last Name	kaval
User Name	kayal
Password	****
Age	22
Gender	○ Male ● Female
Address	XXX
State	Tamilnadu
Country	India
Amount	6000

Message

You have registered successfully ...!

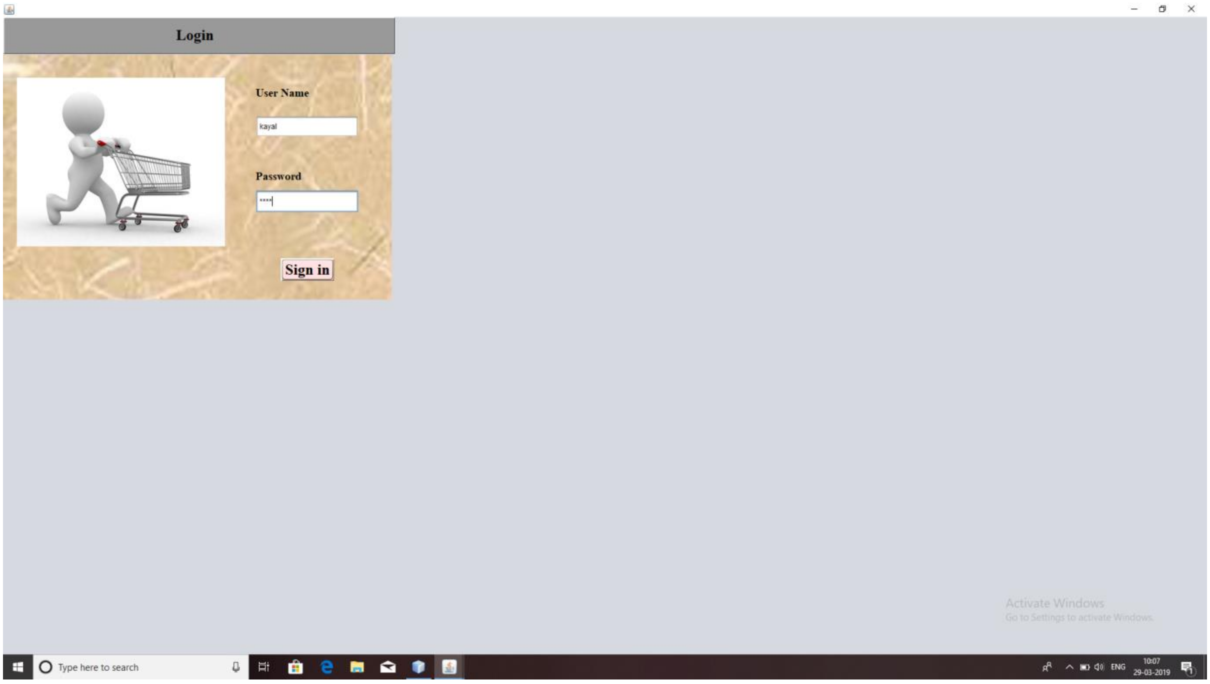


Second level security

User Name	kayal
Your childhood Nick name	kavi
Mother's DOB	12-11-1980
Schooling	dra
Favourite Place	home
Sports you like most	hockey
Favourite Player Name	dhoni

Message

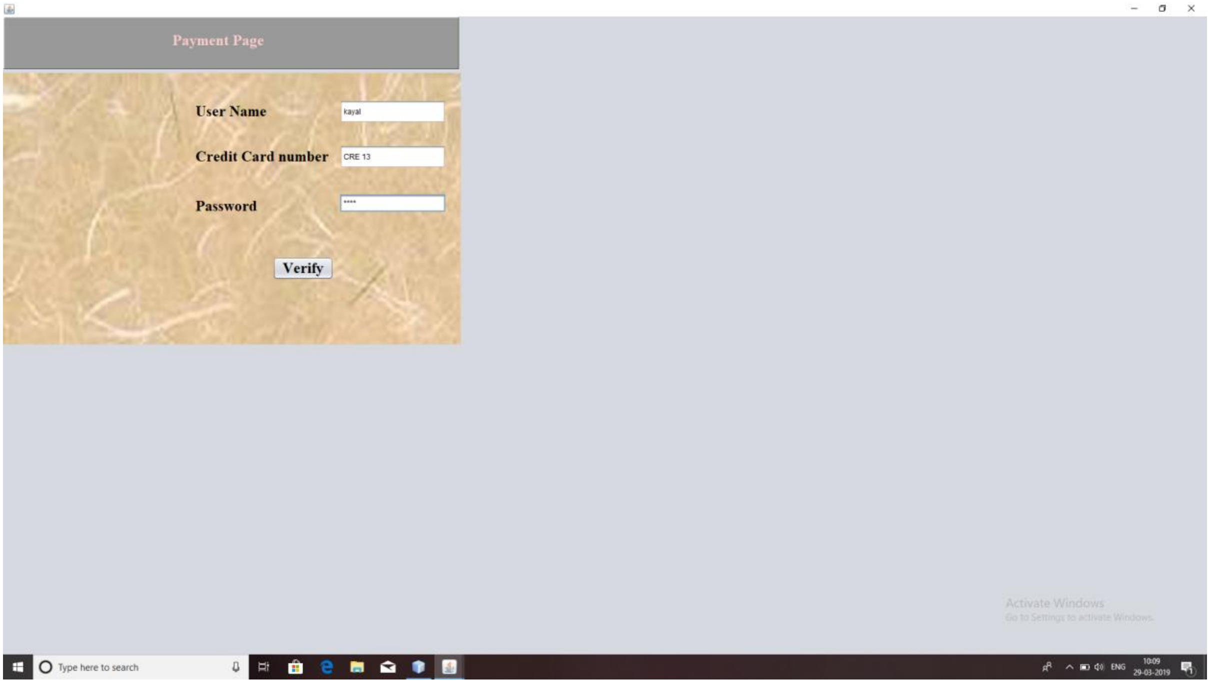
You have registered successfully ...!



Login

User Name kayal

Password ****

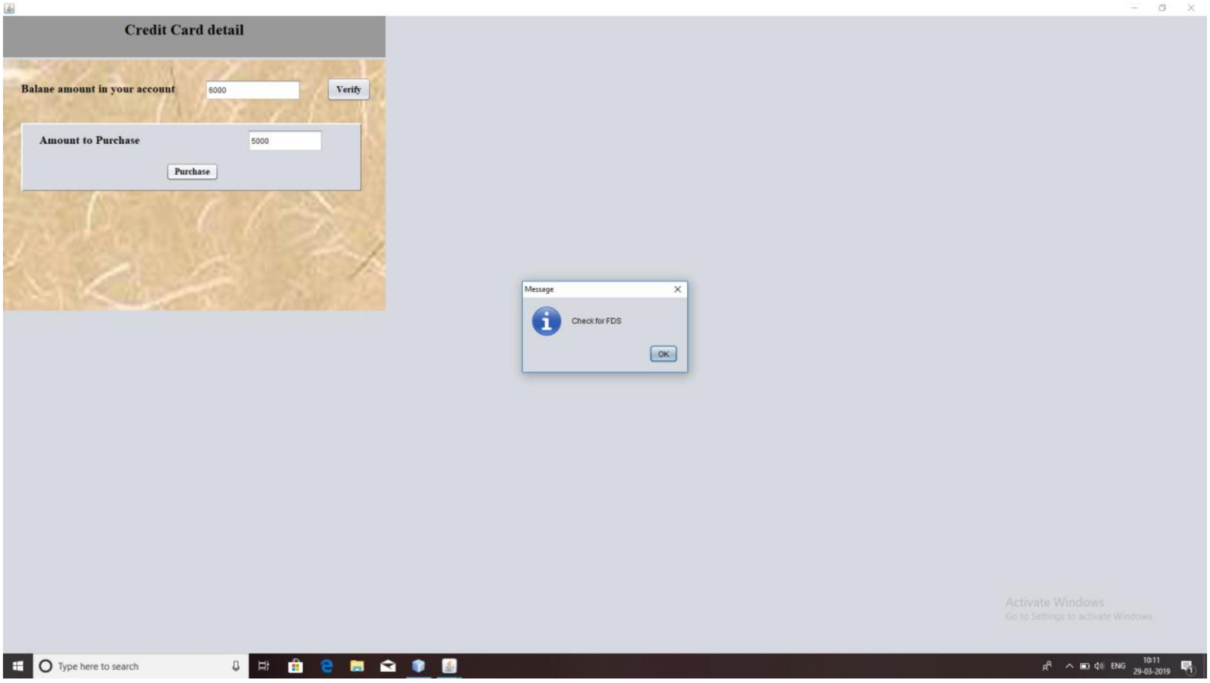


Payment Page

User Name kayal

Credit Card number CRE 13

Password ****



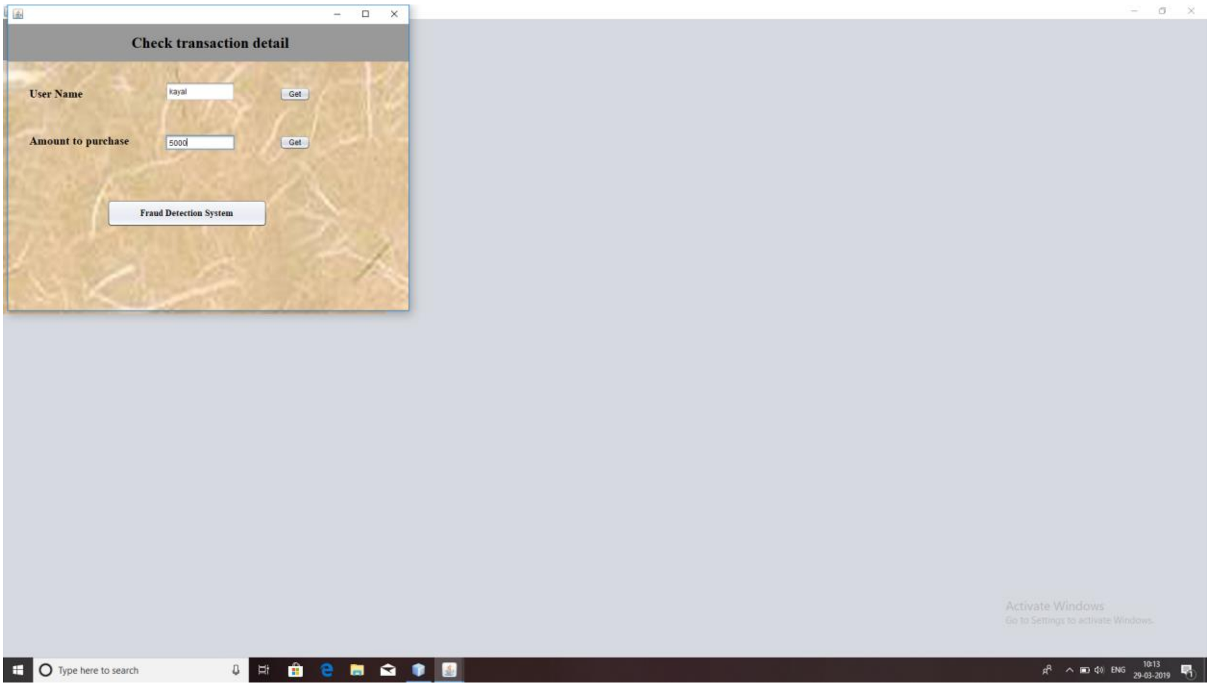
Credit Card detail

Balane amount in your account: 6000

Amount to Purchase: 5000

Message

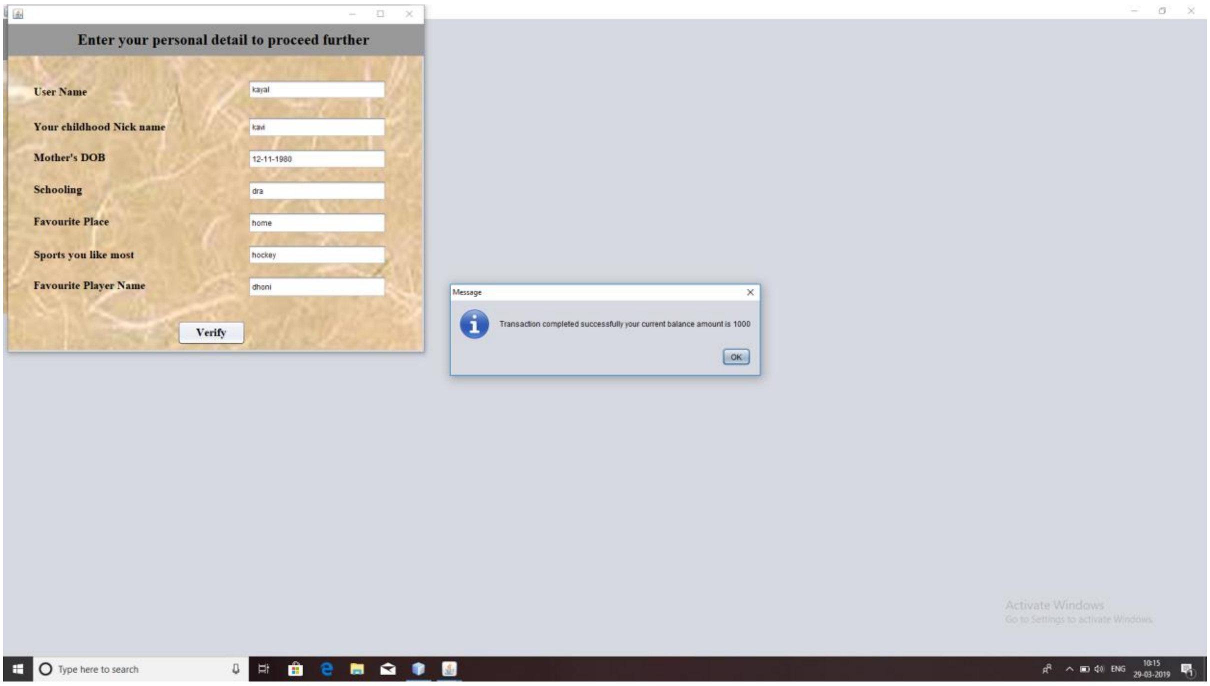
Check for FDS



Check transaction detail

User Name: kayal

Amount to purchase: 5000



Enter your personal detail to proceed further

User Name: kayal

Your childhood Nick name: kavi

Mother's DOB: 12-11-1980

Schooling: dra

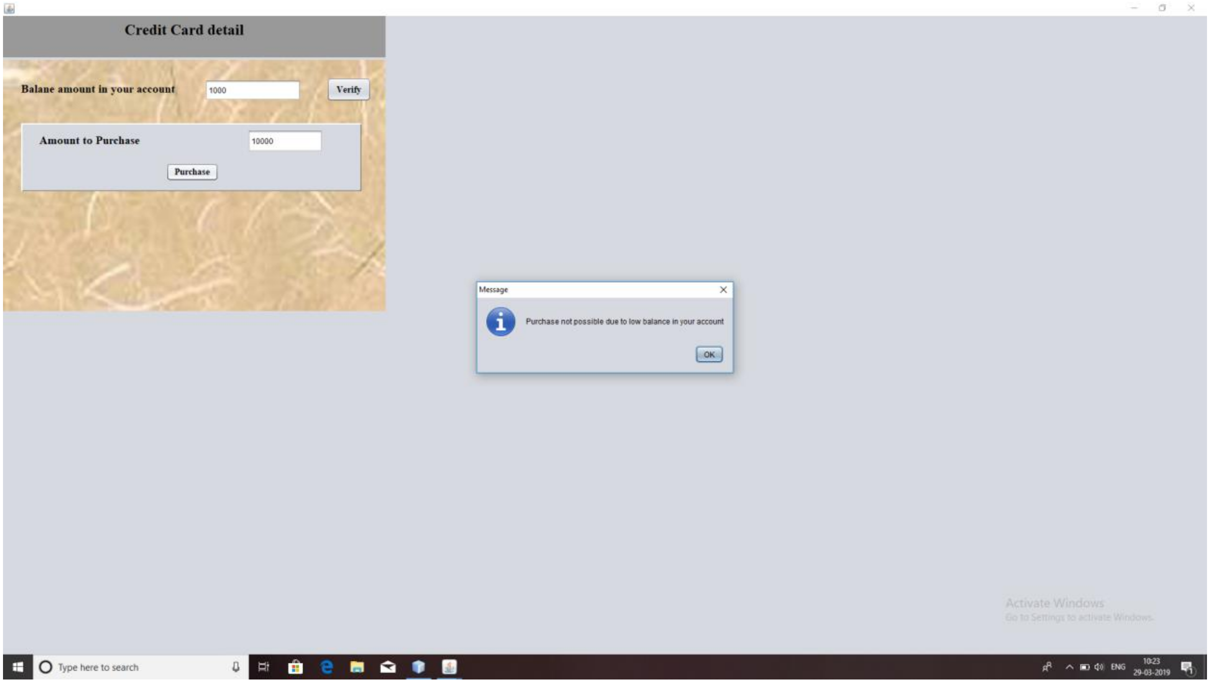
Favourite Place: home

Sports you like most: hockey

Favourite Player Name: dhoni

Message

Transaction completed successfully your current balance amount is 1000



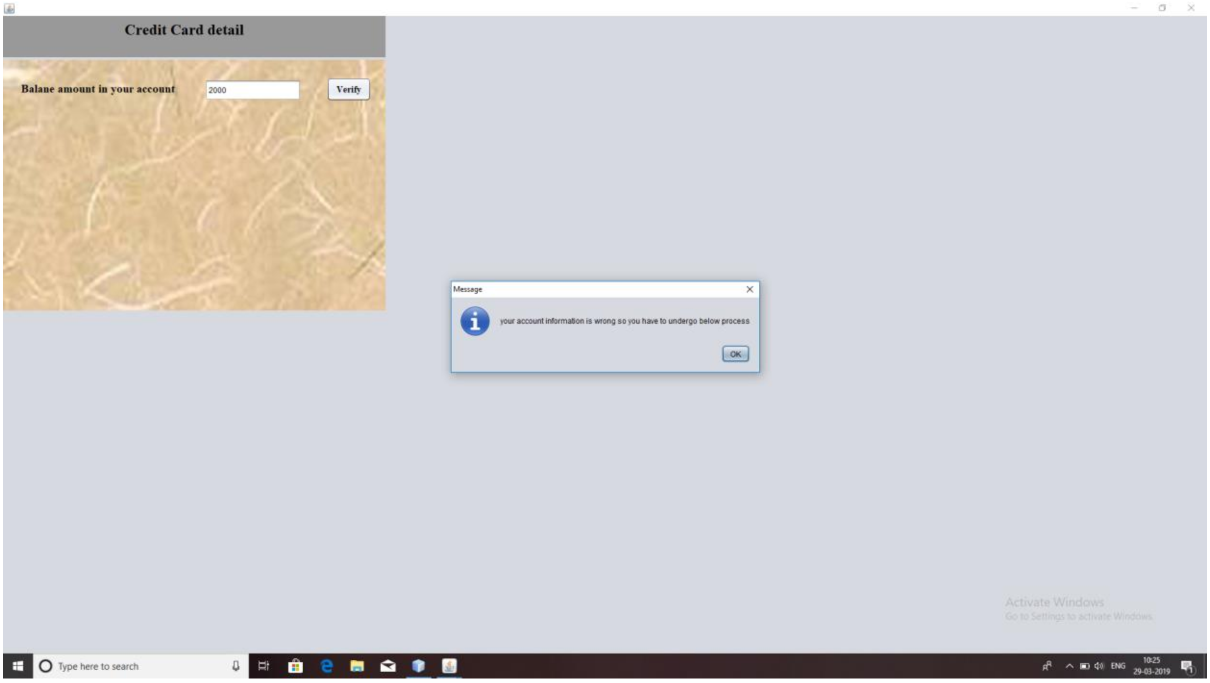
Credit Card detail

Balane amount in your account: 1000

Amount to Purchase: 10000

Message

Purchase not possible due to low balance in your account

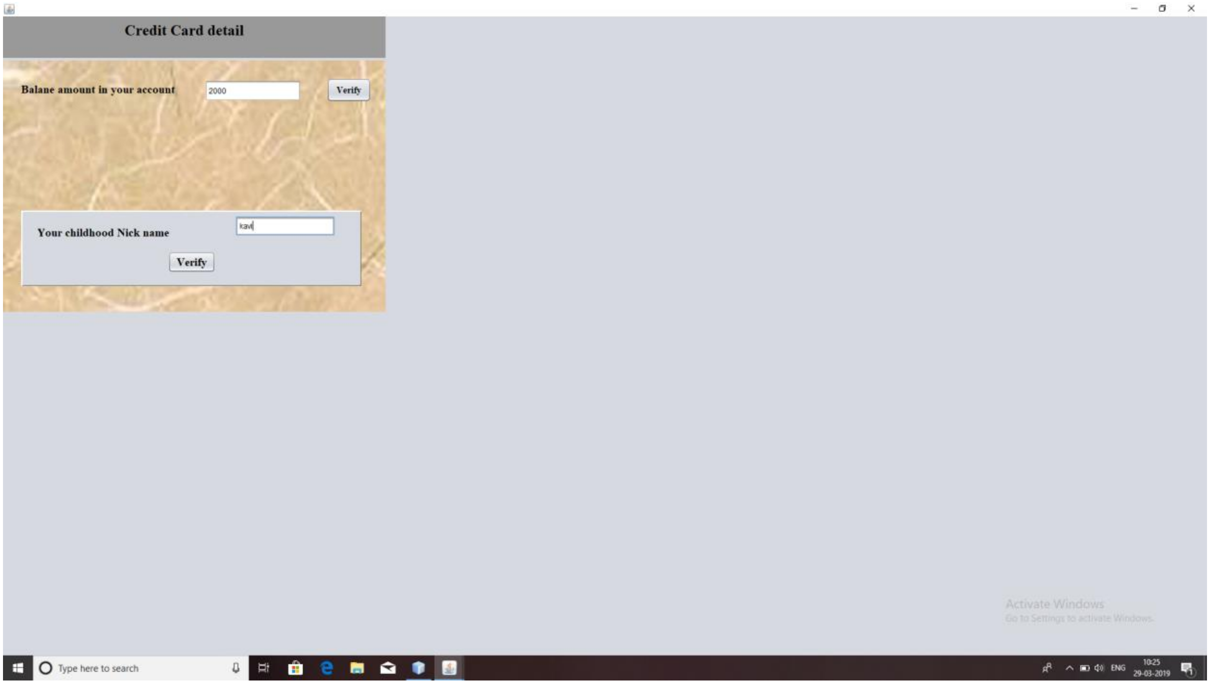


Credit Card detail

Balane amount in your account: 2000

Message

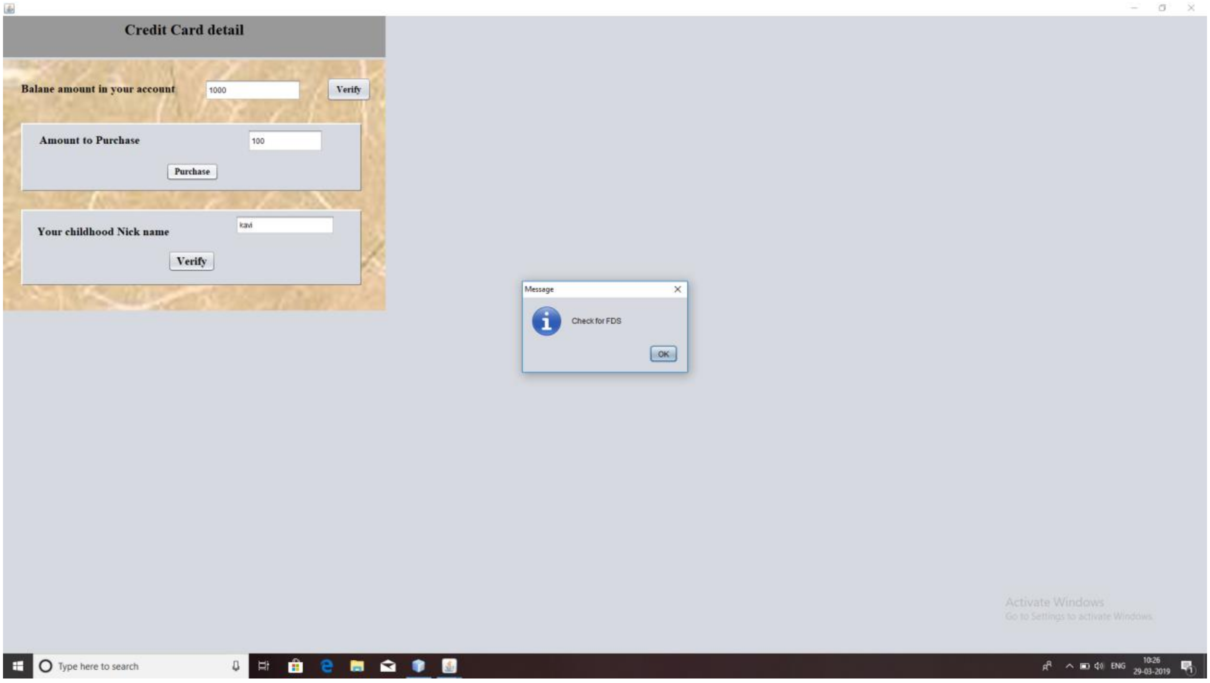
your account information is wrong so you have to undergo below process



Credit Card detail

Balane amount in your account: 2000

Your childhood Nick name: kavi



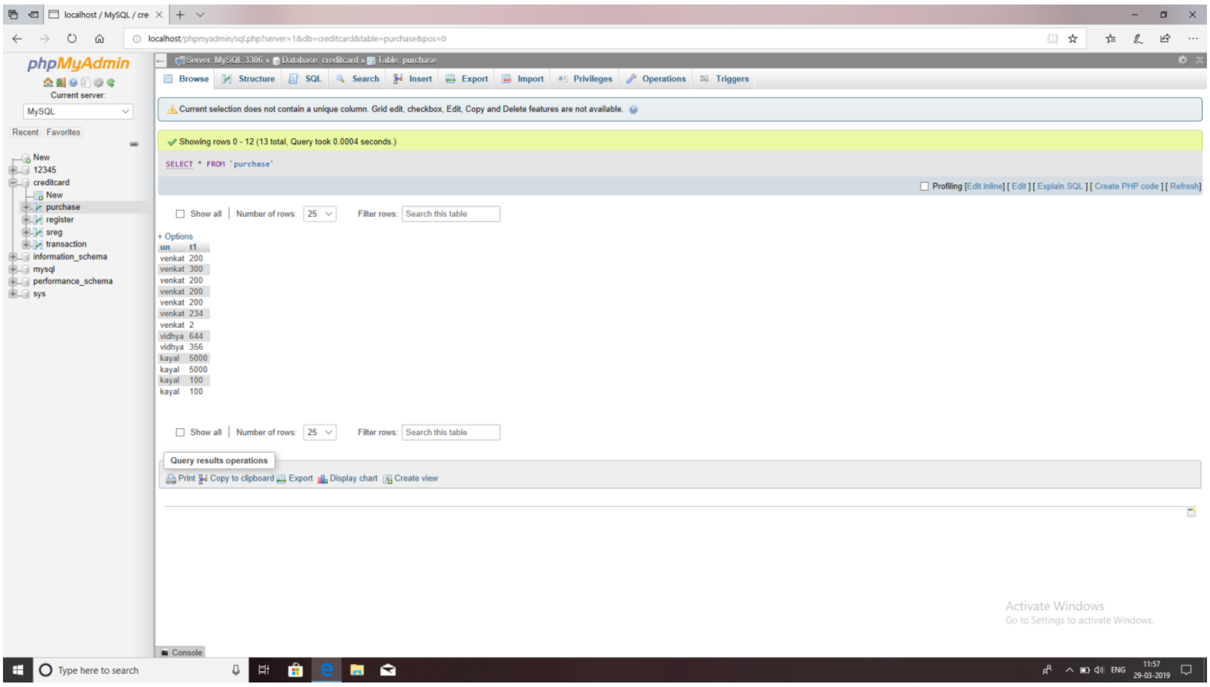
Credit Card detail

Balane amount in your account: 1000

Amount to Purchase: 100

Your childhood Nick name: kavi

Message: Check for FDS



localhost/phpmyadmin/sql.php?server=1&db=creditcard&table=purchase&pos=0

Server: MySQL:3306 » Database: creditcard » Table: purchase

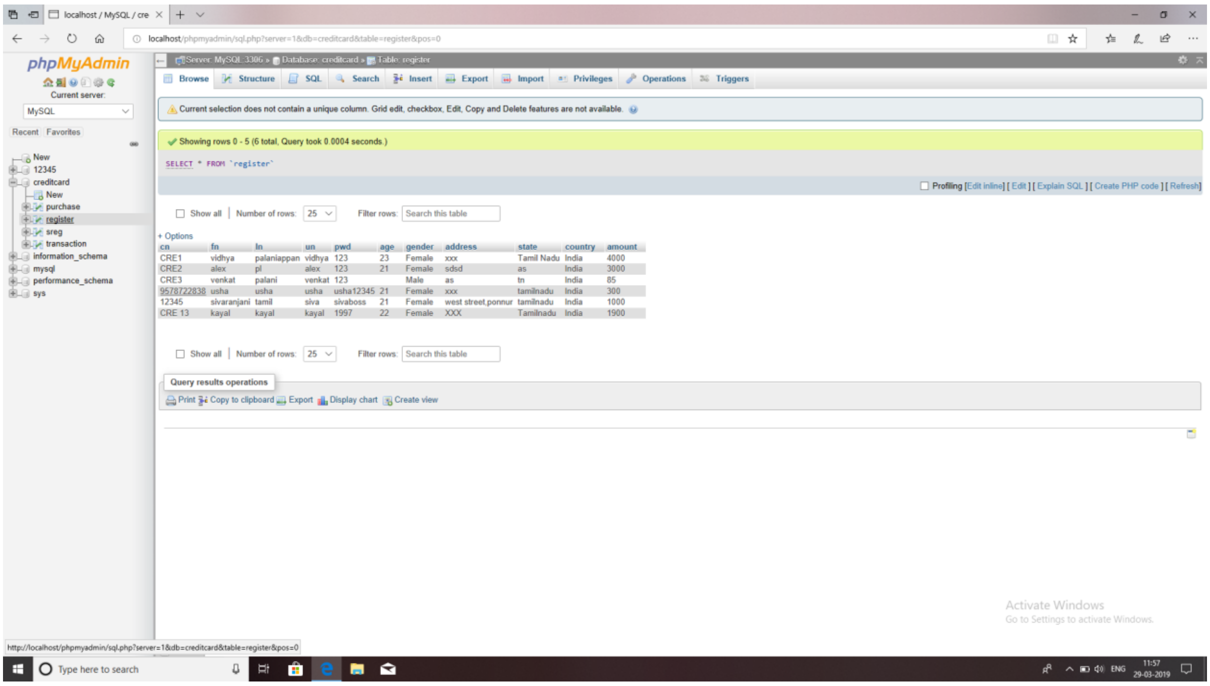
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 12 (13 total, Query took 0.0004 seconds.)

```
SELECT * FROM `purchase`
```

□ Show all | Number of rows: 25 Filter rows: Search this table

un	t1
venkat	200
venkat	300
venkat	200
venkat	200
venkat	200
venkat	234
venkat	2
vidhya	644
vidhya	356
kayal	5000
kayal	5000
kayal	100
kayal	100



localhost/phpmyadmin/sql.php?server=1&db=creditcard&table=register&pos=0

Server: MySQL:3306 » Database: creditcard » Table: register

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

```
SELECT * FROM `register`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

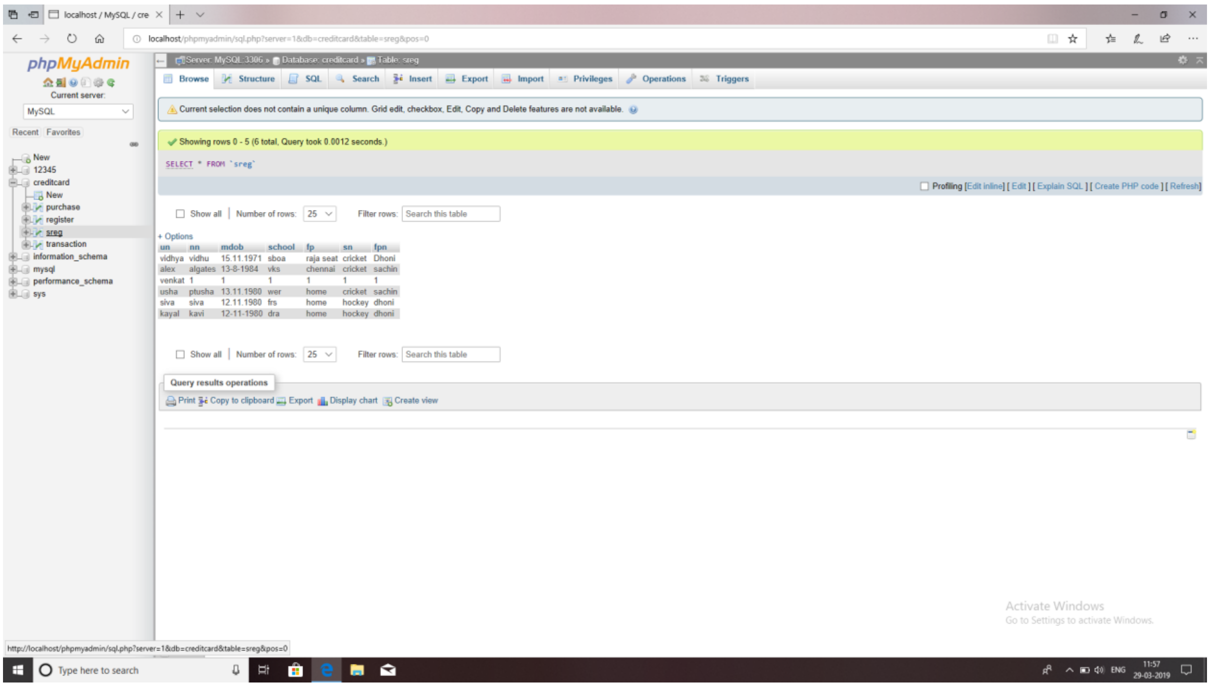
+ Options

cn	fn	ln	un	pwd	age	gender	address	state	country	amount
CRE1	vidhya	palaniappan	vidhya	123	23	Female	xxx	Tamil Nadu	India	4000
CRE2	alex	pl	alex	123	21	Female	sdsd	as	India	3000
CRE3	venkat	palani	venkat	123		Male	as	tn	India	85
9578722838	usha	usha	usha	usha12345	21	Female	xxx	tamilnadu	India	300
12345	sivaranjani	tamil	siva	sivaboss	21	Female	west street,ponnur	tamilnadu	India	1000
CRE 13	kayal	kayal	kayal	1997	22	Female	XXX	Tamilnadu	India	1900

□ Show all | Number of rows: 25 Filter rows: Search this table

Query results operations

http://localhost/phpmyadmin/sql.php?server=1&db=creditcard&table=register&pos=0



localhost/phpmyadmin/sql.php?server=1&db=creditcard&table=sreg&pos=0

Server: MySQL:3306 » Database: creditcard » Table: sreg

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 5 (6 total, Query took 0.0012 seconds.)

```
SELECT * FROM `sreg`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

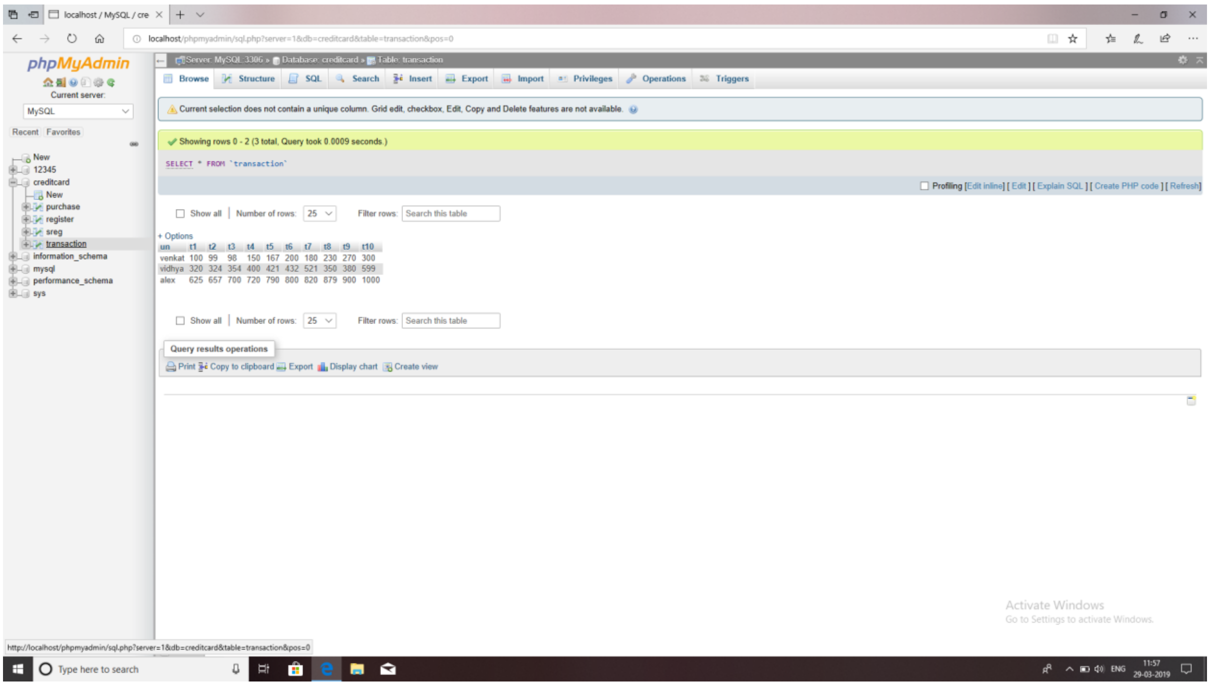
+ Options

un	nn	mdob	school	fp	sn	fpn
vidhya	vidhu	15.11.1971	sboa	raja seat	cricket	Dhoni
alex	algates	13-8-1984	vks	chennai	cricket	sachin
venkat	1	1	1	1	1	1
usha	ptusha	13.11.1980	wer	home	cricket	sachin
siva	siva	12.11.1980	frs	home	hockey	dhoni
kayal	kavi	12-11-1980	dra	home	hockey	dhoni

□ Show all | Number of rows: 25 Filter rows: Search this table

Query results operations

http://localhost/phpmyadmin/sql.php?server=1&db=creditcard&table=sreg&pos=0



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

```
SELECT * FROM `transaction`
```

un	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
venkat	100	99	98	150	167	200	180	230	270	300
vidhya	320	324	354	400	421	432	521	350	380	599
alex	625	657	700	720	790	800	820	879	900	1000

CHAPTER 9

9.1 CONCLUSION

This paper proposed an application of HMM in credit card fraud detection.The different step in credit transaction processing are represented as the underlying stochastic process of an HMM.Using the ranges of transaction amount as the observation symbols,whereas type of item have been considered to the states of the HMM.This paper suggested a method finding the spending profiles of cardholders,as well application of these knowledge in deciding the value of observation symbols,and initial estimate of the model parameters.It has also been explained how the HMM can detect whether on incoming transaction is fraudulent or not.The system also scallable for handling large volume of transaction.

9.2 FUTUREWORKS

The future works focuses on some machine learning methods to automatically classify the values of transaction attributes so that our model can characterize the user's personalised behaviour most precisely.In addition ,planned to extend behaviour profiles by considering other data such as user's command.

REFERENCES

- J.T. Quah and M.Sriganesh, “Real-time credit card fraud detection using computational intelligence,” Expert Syst. Appl., vol. 35, no. 4,pp. 1721-1732, 2008.
- Z.Zojaji, R. E. Atani, and A.H. Monadjemi. (2016). “A survey of credit card fraud detection techniques: Data and technique oriented perspective[online] Available: http://arxiv.org/abs/1611.06439.
- C.Jiang, J.Song, G. Liu, l. Zheng, and W. Luan, “credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism.” IEEE Internet Things J. to be published. doi:10.1109/JIOT.2018.2816007.
- W.-H. Ju and Y. Vardi, “A hybrid high-order Markov chain model for computer intrusion detection,” *J. Comput. Graph. Stat.,* vol. 10, no. 2, pp. 277–295, 2004.
- J. Lopes, O. Belo, and C. Vieira, “Applying user signatures on fraud detection in telecommunications networks,” in *Proc. Ind. Conf. Data Mining,* 2011, pp. 286–299.
- J. Lobo, “Internet banking fraud detection using HMM,” in *Proc. 3rd Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT),* Jul. 2012, pp. 1–4.
- M. Schonlau, W. DuMouchel, W. H. Ju, A. F. Karr, M. Theus, and Y. Vardi, “Computer intrusion: Detecting masquerades,” *Stat. Sci.,* vol. 16, no. 1, pp. 58–74, 2001.
- L. Seyedhossein and M. R. Hashemi, “Mining information from credit card time series for timelier fraud detection,” in *Proc. Int. Symp.Telecommun.,* Dec. 2010, pp. 619–624.
- Shen, R. Tong, and Y. Deng, “Application of classification models on credit card fraud detection,” in *Proc. Int. Conf. Service Syst. Service Manage.,* Jun. 2007, pp. 1–4.
- E. Shi, Y. Niu, M. Jakobsson, and R. Chow, “Implicit authentication through learning user behavior,” in *Proc. Int. Conf. Inf. Secur.,* 2011, pp. 99–113.