



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering (SITE)

MTech (Software Engineering)

**INFORMATION AND SECURITY SYSTEM
(SWE3002)**

REVIEW -III

SECURE CARD PAYMENT

DONE BY,

18MIS0292 - J THARAKA

**Under the guidance of
Prof. MANGAYARKARASI R**

ABSTRACT:

Due to increasing of the online shopping and online food delivering, e-commerce sites and bank transaction become easier there is a need for some encryption technique to ensure security and a way to ensure that the user's data are securely stored in the database.

There is increasing the fraud business in some of the cities and normal places. Most of the users are hackers and using other people credit card and taking money away from them. And most of credit card fraud is happening in numerous ways facilitated by poor internet security. The criticality, danger, and higher priority importance of any e-commerce money transfer makes it a hot area of research interest in modern computer science and informatics. Thus, the system introduces RSA for this purpose. The RSA algorithm is a kind of asymmetric encryption algorithm which appeared in 1978. RSA is more secure because it uses Prime numbers. The algorithm is public key encryption algorithm which is a widely accepted and implemented by public. The use of RSA in this the system makes the process more secure.

Now the bank transactions can be done securely without worrying about attacker getting access to the database as the data will be in encrypted form.

Recently we have seen some frauds for example someone delivers something and they say only card payments are accepted. The customer gives the card and he clicks more amount to be debited. So to avoid this type of fraud the customer receives the otp message with amount to have been debited.

INTRODUCTION:

As we know that the more frauds are happening by the credit card. Usually, transactions involved in online payments are „card not present“ type transactions. This means the cardholder is absent physically at the time transaction is processed. This makes online transactions vulnerable to fraud and cyber-attacks. In today's era of digital technology, online businesses, e, and m-commerce customers and merchants both expect a swift and secure digital experience. In the customer's case, it will be surfing, shopping, and paying online. In the merchant's case showcasing, marketing, and processing payments digitally. The innovations, modern technologies have already simplified the b2b and c2b experience. But as the online marketing and selling experience is reaching to more and more customers, user demands are also increasing with it. However, in parallel, the attacks procedures and strategies are as advanced as

the security solutions. In this way, we chose the most dominant electronic payment systems and the most successful attack strategies against them.

- The name of Card-present-transaction is derived from presentation of the card to the merchant by the card holder, at the time of transaction. This means that any online payment via internet.
- In a Card Payment transaction, the card information is sent to the payment gateway and after verification, the gateway will send the transaction authorization to the merchant along with a receipt to the card holder. In such this transaction, in case of fraud by customer, the merchant is not liable of the losses because of the payment gateway authorization. The important thing in Card-present-transaction is PIN and CVV number

In this project we have implemented the most securing payment during the online shopping. This may reduce the fraud business using credit card. We are implementing this to reduce fraud by providing e-sign of the card owner and at the end of the transaction the owner receives the OTP to their mobile number and then only the payment will successfully completed. So in this way the fraud business may reduce.

RELATIVE WORK:

1. TITLE: A Secure Operational Model for Mobile Payments

PUBLISHED YEAR: 2015

AUTHORS: Jung-Fa Tsai.

DESCRIPTION: Mobile devices are almost ubiquitous, their computational power is rapidly increasing, and they are as connected as personal computers or laptops. Gartner estimated that during 2013 mobile phones would replace personal computers as the most common web access device, while Forrester predicted that in the same year 48 percent of all US mobile-phone subscribers would be using smartphones, a striking increase from just 7 percent in 2008. A growing number of customers can use their mobile phones as keys, cameras, and TVs, and their use as a payment tool would further add to this convenience. A mobile payment has been defined as “any payment where a mobile device is used in order to initiate, activate, and/or confirm this payment”. Although large-scale mobile payment systems are still under development, several mobile financial and mobile commerce applications (e.g., the Starbucks app, iTunes, and Google Wallet) are helping to increase user experiences and encourage the adoption of mobile payments among customers. Customers usually pay for their commodities with a

prepaid card or credit card in supermarkets. Many prepaid cards issued by specific stores cannot be identified when they are lost, and anyone who picks up a lost prepaid card can use it without being caught. Since shop cashiers seldom check the signature, people are usually not aware if small amounts of money are withdrawn from the balance of a lost credit card if they do not check their accounts regularly

2. **TITLE:** A study and implementation of SMS security for business transactions

PUBLISHED YEAR: 2015

AUTHORS: Shadab Haider, Dr. RK Singh

DESCRIPTION: The security of many cryptographic schemes and protocols depends on the hardness of finding the factors of large integers drawn from an appropriate distribution. To determine what key sizes are appropriate for a given application, one needs concrete estimates for the cost of factoring integers of various sizes. Predicting these costs is difficult, for two reasons. First, the performance of modern factoring algorithms is not understood very well. Second, even when the exact algorithmic complexity is known, it is hard to estimate the concrete cost of a large-scale computational effort using current technology. Due to these difficulties, common practice is to rely on extrapolations from past factorization experiments. Many such experiments have been performed and published.

3. **TITLE:** Basic Security of the e-cash Payment System

PUBLISHED YEAR: 2015

AUTHORS: Berry Schoenmakers

DESCRIPTION: A basic requirement of a payment protocol is that it allows a payee to receive payments from any payer. A payment can be seen as some sort of authentication of the payer towards the payee (to show that the payment is authentic). Authentication can be based on

secret key cryptography or on public key cryptography. In the latter case, the payee only needs to have a public key available in order to verify incoming payments. Although the costs of equipping smart cards with crypto co-processors are expected to become marginal, it is important to note that the property of public verifiability can be obtained using simple smart cards only, provided one applies a method of what we call signature transport. In such a system, signatures are created by the issuer only, and later endorsed by the payer during the payment protocol.

depending on a challenge from the payee. The trick is to achieve that sufficiently many payments can be made between successive reloads, which requires optimal use of the limited amount of EEPROM available on simple smart cards. The added advantage is that the secret key for creating signatures is only used by the issuer.

4. **TITLE:** Security of Mobile Payments and Digital Wallets

AUTHORS:Romana Sachová,

PUBLISHED YEAR:2011

DESCRIPTION:The primary objective of this paper is the production of guidelines to assist mobile payment developers and mobile payment providers towards recommended security controls which if implemented would help ensure that consumers, retailers and financial institutions are all safeguarded from cyber threats. A secondary objective is to define minimum measures that should be followed by mobile payment providers in the EU, and to provide security recommendations for organisations wishing to provide mobile payment services within the EU.

5. **TITLE:** card payment using RSA

AUTHORSS: Priya Deshmukh

PUBLISHED YEAR:2016

DESCRIPTION: Due to increasing e-commerce activity nowadays, there is a need for some encryption technique to ensure security and a way to ensure that the user's data are securely stored in the database. Thus the system introduces RSA for this purpose. The RSA algorithm is a kind of asymmetric encryption algorithm which appeared in 1978. The algorithm is public key encryption algorithm which is a widely accepted and implemented by public. The use of RSA in this the system makes the process more secure. Now the bank transactions can be done securely without worrying about attacker getting access to the database as the data will be in encrypted form.

6. **TITLE:** Technical Review On Secure Banking Using RSA And AES Encryptor Methodologies

AUTHORSS: Abhilesh S. Jadhao, Shital B. Kumbhalkar

PUBLISHED YEAR:2013

DESCRIPTION: In order to visualize the effect and evaluate the performance of the encryption and decryption of each technique used in communication systems, Visual Basic simulation program that encrypt and decrypt data were developed, written and tested. Different data block sizes were captured and plotted against total time response taken during data encryption using Microsoft Excel. Multiple security attacks can be attempted during the ATM transactions to gain unauthorized access to the precious information and use the same information against clients. In order to prevent such cybercrimes, various data security levels and encryption standards are used for secure transactions

7. **TITLE:** Simple and Secure Credit Card-Based Payment System

AUTHORSS: Chi Po Cheong

PUBLISHED YEAR:2013

DESCRIPTION: the success of an Internet payment system, a set of factors should be evaluated. Technically, the four sets of criteria that must be weighed [1]: security, cost, convenience and universality Security features consist of identification, authentication, confidentiality, integrity, and non-repudiation They should all be covered by a reliable payment system. Secure Socket Layer (SSL) is the most commonly used protocol in ecommerce, but it is not a payment protocol. It only protects the confidential data in the communication channel between the merchant and the consumer. The credit card number or information could be stolen on the merchant's side during the payment process. Two types of payment protocols are used in the credit card-based payment system. The first type uses a virtual number instead of the actual credit card number during the payment cycle [2]. The second one uses the actual credit card number with encryption

8. **TITLE:** Money, Credit, and Digital Payment

AUTHORS:Sebastian Gießmann

PUBLISHED YEAR:2015

DESCRIPTION: The next adoption approach is Convenience, but that's not an isolated approach. While the original ease-of-use path was adopted, a few other areas were impacted – Security and Feature Expansion. The fewer steps needed to send a payment, the greater the

initial adoption... Until you send money to the wrong person and cannot get it back. Security of the service generally and of each transaction specifically adds back the complexity that application developers worked so hard to remove. Users are growing more aware of this gap, especially

as reporting of these issues increases, so apps like Venmo, Cash App (by Square), and Zelle's standalone version are expanding security as well as contact confirmation prior to remittance. Additionally, supporting a Convenience mode as they reduce your interface with other applications, P2P apps are adding messaging, emojis, activity feeds, and more as features they feel will add fun while reducing overall clicks.

9. **TITLE :** E-Banking Security using Cryptography, Steganography and Data Mining)

AUTHORS : Namrata Devadiga Harshad Kothari

PUBLISHED YEAR:2014

DESCRIPTION: The system has been successfully implemented and

provides a secure EBanking experience for every user. The system provides a strong mechanism to prevent online frauds by using cryptography and steganography on the client side. On the server side of the system Data mining ensures fraud detection. Based on all the tests conducted on the image generated by steganography one can deduce that the image satisfies all the necessary criteria in terms of quality and efficiency. It fails to prevent the hacking of the user card details in the very first place. Steganography has a very significant plus over cryptography which is that the intended secret message to be transmitted over a network does not garner any attention to itself as an item of examination.

10. **TITLE:** Security of Mobile Payments and Digital Wallets

AUTHORS : Romana Sachová,

PUBLISHED YEAR:2011

DESCRIPTION:The primary objective of this paper is the production of guidelines to assist mobile payment developers and mobile payment providers towards recommended security controls which if implemented would help ensure that consumers, retailers and financial institutions are all safeguarded from cyber threats. A secondary objective is to define minimum measures that should be followed by mobile payment providers in the EU, and to provide security recommendations for organisations wishing to provide mobile payment services within the EU.

MODULES IN IMPLEMENTATION:

Step 1: Home page

- =>login
- =>user login
- =>admin login

Step 2; In admin login

- =>Add product
- =>check product availability

Step 3: In user login

- =>buy product
- =>Return product

Step 4: Payment

- =>credit card
- =>debit card

Step 5: In credit and debit

- =>enter card number
- =>card details
- =>e-sign
- =>verify sign
- =>encrypt
- =>decrypt

MODULES:

i. Collect Details:

In this module we actually collect the card detail from the user.

And these details will be later used in the encryption process.

ii. Symmetric Encryption :

In this module , we take the CVV number of the card from collected details.

Then the using the CVV and the common key which is agreed on both side , the encryption takes place .

Then the resultant cipher text is provided to the next module.

iii. RSA Encryption :

In this module, we take the resultant cipher text of CVV from the symmetric encrypt module.

Then using that cipher text and the generated public key RSA start to encrypt .

Then the result of the RSA is rsa cipher text is then shared to the merchant.

iv. RSA Decryption :

In this module , the merchant collect the rsa ciphered text from the user or customer.

Then using this rsa cipher text and his private key he start to decrypt the cipher text.

After decrypt the cipher text , he gets the symmetric ciphered text.

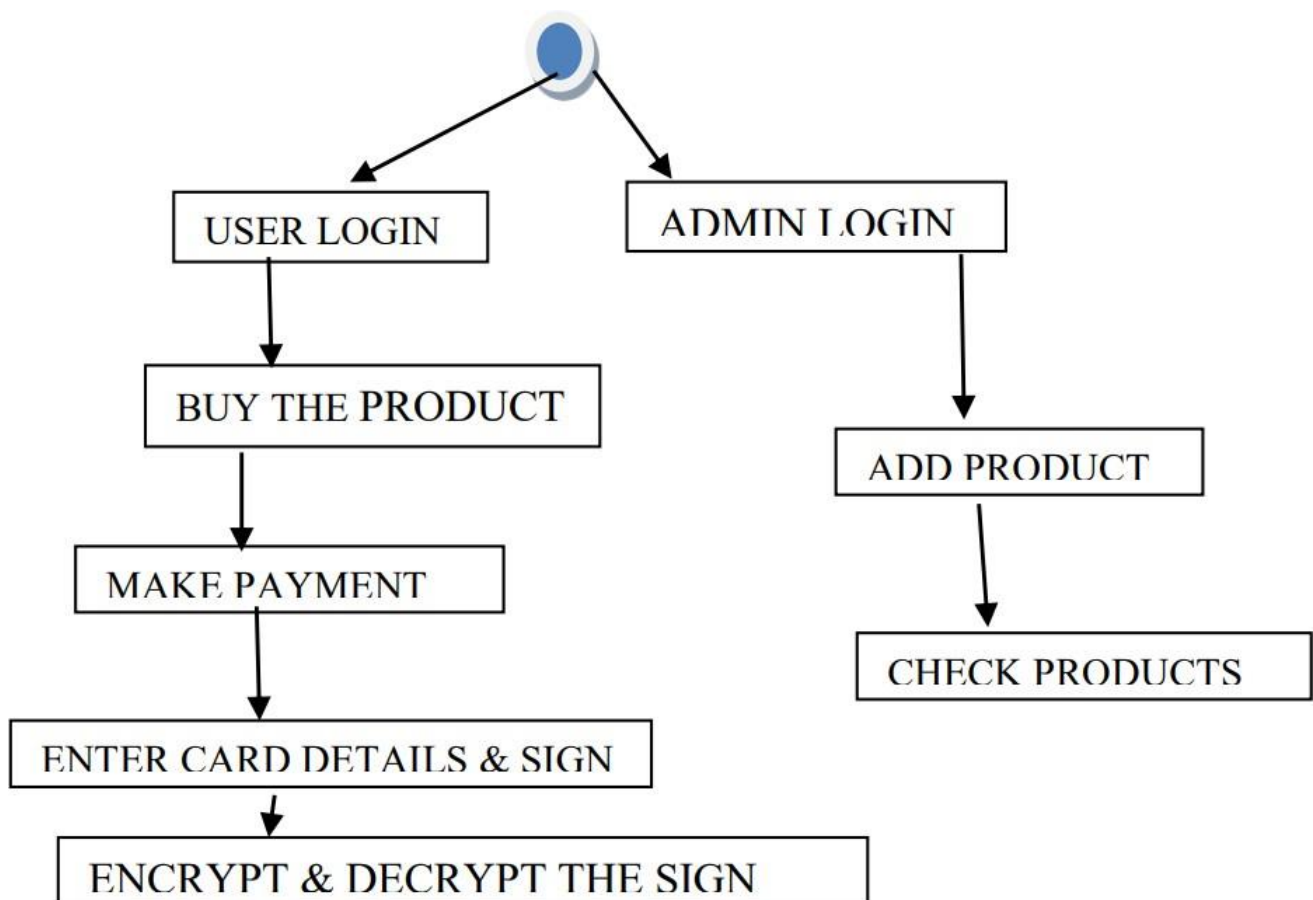
v. Symmetric Decryption:

In this module the merchant take the obtained symmetric ciphertext from the decryption.

Then he use this and the common key they agreed to decrypt the symmetric encryption.

Now he got the original CVV and proceeds the payment process

SYSTEM FLOW ARCHITECTURE:



CAR RENTAL SYSTEM WITH SECURE TRANSACTION:

- Step 1: The user enters his/her credit card details on the merchant site. On submission the data is sent to the merchant server.
- Step 2: Symmetric encryption are generated.
- Step 3: In Symmetric algorithm, we take a key to encrypt the plain text which length should be equal or maybe unequal to the length of the plain text.
- Step 4: After the encryption, merchant server sends a request to the bank server for an RSA public key which is required to encrypt message with RSA 1024.
- Step 5: On the reception of RSA public key at merchant server symmetric cipher texts are encrypted using RSA.
- Step 6: RSA encrypted card details are sent to the bank server .
- Step 7: After the data is received at the bank server, using RSA private key which is generated along with the RSA public key in Step 4 will decrypt to get the symmetric ciphertext.
- Step 8: The obtained symmetric cipher text and the common key are used to decrypt the card details encrypted at merchant server using symmetric algorithm.

Implementation :

Home page :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    login l=new login();  
    l.setVisible(true);  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    adminlog s= new adminlog();  
    s.setVisible(true);  
}
```

Admin page :

```
private void jMenuItem1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    sell s=new sell();
    s.setVisible(true);
}

private void jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    sold s=new sold();
    s.setVisible(true);
}

private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    contact c=new contact();
    c.setVisible(true);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    home h=new home();
    h.setVisible(true);
}
```

Signup page :

```
import java.awt.event.ActionEvent;
import java.math.BigInteger;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Random;
import javax.swing.JOptionPane;

/**
 *
 * @author USER
 */
public class signin extends javax.swing.JFrame {
    PreparedStatement pst=null;
    private BigInteger p;
    private BigInteger q;
    private BigInteger N;
    private BigInteger phi;
    private BigInteger e;
    private BigInteger d;

    private int bitlength = 1024;
    private Random r;
    byte[] encrypted;
```

```

/**
 * Creates new form signIn
 */
public signIn() {
    initComponents();

    r = new Random();

    p = BigInteger.probablePrime(bitlength, r);
    q = BigInteger.probablePrime(bitlength, r);
    N = p.multiply(q);
    phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
    e = BigInteger.probablePrime(bitlength / 2, r);
    while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi)
    < 0)

    {

    e.add(BigInteger.ONE);
    }
    d = e.modInverse(phi);
    }

    public signIn(BigInteger e, BigInteger d, BigInteger N)
    {

    this.e = e; this.d = d; this.N = N;
    }

    private static String bytesToString(byte[] encrypted)

    {

    String test = "";

    for (byte b : encrypted)
    {

    test += Byte.toString(b);

    }

    return test;

    }

```

```

public byte[] encrypt(byte[] message)

{

return (new BigInteger(message)).modPow(e, N).toByteArray();

}

public byte[] decrypt(byte[] message)
{

return (new BigInteger(message)).modPow(d, N).toByteArray();

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String Users = jTextField1.getText();
    String Cpasses=jPasswordField1.getText();
    String Pass=jTextField2.getText();
    String Pno=jTextField3.getText();
    String Add=jTextField4.getText();
    try
    {
        signin rsa = new signin();
        String en= jTextField5.getText();
        byte[] numbe=bytesToString(en.getBytes()).getBytes();
        String enc=new String(numbe);
        //JOptionPane.showMessageDialog(null, "Encrypted sucessfully
        "+bytesToString(en.getBytes()));
        encrypted= rsa.encrypt(en.getBytes());
        //  jTextField1.setText(bytesToString(en.getBytes()));
        //JOptionPane.showMessageDialog(null, "Encrypted sucessfully
        "+encrypted);
        byte[] decrypted = rsa.decrypt(encrypted);
        bytesToString(decrypted);
        String S=new String(decrypted);

        Connection con
        =DriverManager.getConnection("jdbc:derby://localhost:1527/carproject", "
        root", "Nasusa9866@");
    }
}

```

```

String sql="insert into
SIGNIN(USERNAME,PASSWORD,CONFIRMPASSWORD,PHONEN
UMB,ADDRESS,DSE,DSD)
values('"+Users+"','"+Cpasses+"','"+Pass+"','"+Pno+"','"+Add+"','"+enc+"
','"+S+"')";
        Statement st= (Statement) con.createStatement();

        if(Cpasses.equals(Pass)){
            st.executeUpdate(sql);
            JOptionPane.showMessageDialog(null,"Registered
SUCESSFULLY...");

            login l=new login();
            l.setVisible(true);

        }else{

            JOptionPane.showMessageDialog(null,"both password and
confirm password should be same");
        }

    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(null,e);
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    login l=new login();
    l.setVisible(true);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    jTextField1.setText("");
    jPasswordField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
}

```

```
jTextField4.setText("");
    jTextField5.setText("");

}
```

Customer Welcome page :

```
private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try{

        String n=jLabel3.getText();
        Connection con
        =DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
        root,"Nasusa9866@");
        String sql="select*from DP IMG where USERNAME='"+n+"' ";
        Statement st= (Statement) con.createStatement();
        ResultSet rs=st.executeQuery(sql);

        while(rs.next()){
            String s=rs.getString("IMG");
            ImageIcon icon= new ImageIcon(s);

            Image image=icon.getImage().getScaledInstance(jLabel6.getWidth(),
            jLabel6.getHeight(),Image.SCALE_SMOOTH);
```



```

        jLabel6.setIcon(icon);
    }

    }catch(SQLException e)
    {
JOptionPane.showMessageDialog(null,e);
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    JFileChooser chooser=new JFileChooser();
    chooser.showOpenDialog(null);
    File f=chooser.getSelectedFile();
    filename=f.getAbsolutePath();
    jTextField1.setText(filename);
    Image getAbsolutePath=null;
    ImageIcon icon= new ImageIcon(filename);
    Image image=icon.getImage().getScaledInstance(jLabel6.getWidth(),
jLabel6.getHeight(),Image.SCALE_SMOOTH);
    jLabel6.setIcon(icon);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    try{
        String n=jLabel3.getText();
        String imageIcon=jTextField1.getText();
        Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");

String sql="insert into DP(IMG,USERNAME)
values('"+imageIcon+"','"+n+"')";
//String sql1="update DP set IMG='"+imageIcon+"' where
USERNAME='"+n+"'";
Statement st= con.createStatement();
        st.executeUpdate(sql);
        // st.executeUpdate(sql1);
        JOptionPane.showMessageDialog(null,"DP updated");
    }catch(SQLException e)
    {
JOptionPane.showMessageDialog(null,"something went wrong");
    }
}

```

```

    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
    {

        // TODO add your handling code here:
        login l=new login();
        l.setVisible(true);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:

        account a= new account();
        a.setVisible(true);

        String n=jLabel3.getText();
        try{
            Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject","
root","Nasusa9866@");
            String sql="select* from SIGNIN where USERNAME='"+n+"'";
            Statement st= con.createStatement();
            ResultSet rs= st.executeQuery(sql);
            while(rs.next()){
                String u=rs.getString("USERNAME");
                String p=rs.getString("PASSWORD");
                String ph=rs.getString("PHONENUMB");
                String ad=rs.getString("ADDRESS");

                a.jLabel11.setText(u);
                a.jLabel12.setText(p);
                a.jLabel13.setText(ph);
                a.jLabel14.setText(ad);
            }

        }catch(SQLException e)
        {
            JOptionPane.showMessageDialog(null,"something went wrong");
        }
    }

```

```

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:

    try{
        sell s=new sell();

        s.setVisible(true);
        String n=jLabel3.getText();
        Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
        String sql="select* from SIGNIN where USERNAME='"+n+"'";
        Statement st= con.createStatement();
        ResultSet rs= st.executeQuery(sql);
        while(rs.next()){
            String ph=rs.getString("PHONENUMB");
            s.jLabel4.setText(n);
            s.jLabel6.setText(ph);
        }
    }catch(SQLException e)
    {
        JOptionPane.showMessageDialog(null,"something went wrong");
    }

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    String n=jLabel3.getText();
    buy b=new buy();
    b.setVisible(true);
    b.jLabel3.setText(n);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    String n=jLabel3.getText();
    retur r =new retur();
    r.setVisible(true);
    r.jLabel1.setText(n);
    r.Show();
}

```

}

Buy page :

```
public ArrayList<user> userList(){
    ArrayList<user> userList=new ArrayList<>();
    try{
        String n=jTextField1.getText();
        Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
        String sql="select* from SELL where MODEL='"+n+"'";
        Statement st= con.createStatement();
        ResultSet rs= st.executeQuery(sql);
```

```

        user User;
        while(rs.next()){
            User=new
user(rs.getString("MODEL"),rs.getString("NEOL"),rs.getString("PRICE")
,rs.getString("NAME"),rs.getString("CC"));
            userList.add(User);

        }

    }catch(SQLException e)
{
OptionPane.showMessageDialog(null,"something went wrong");
}

    return userList;
}
public void Show(){
    ArrayList<user>list=userList();
    DefaultTableModel model=(DefaultTableModel)table.getModel();
    Object[] row=new Object[50];
    for(int i=0;i<list.size();i++){
        row[0]=list.get(i).getModel();
        row[1]=list.get(i).getNEOL();
        row[2]=list.get(i).getPRICE();
        row[3]=list.get(i).getNAME();
        row[4]=list.get(i).getCC();
        model.addRow(row);
    }
}

private void tableMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    buyd bd=new buyd();
    bd.setVisible(true);
    DefaultTableModel model=(DefaultTableModel)table.getModel();
    int i=table.getSelectedRow();
    String m=model.getValueAt(i,0).toString();
    String p=model.getValueAt(i,1).toString();
    String fs=model.getValueAt(i,2).toString();
    String o=model.getValueAt(i,3).toString();
    String no=model.getValueAt(i,4).toString();
    String s=jLabel3.getText();
    bd.jLabel1.setText(m);
    bd.jLabel5.setText(p);
    bd.jLabel4.setText(fs);
    bd.jLabel7.setText(o);
    bd.jLabel16.setText(s);
}

```

```

try{

Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");

String sql="select*from SELL where CC='"+no+"' ";
Statement st= con.createStatement();

ResultSet rs=st.executeQuery(sql);
while(rs.next()){
String km=rs.getString("KM");
String img=rs.getString("IMAGE");
String name=rs.getString("NAME");
String num=rs.getString("NUMB");
String sp=rs.getString("SPECIFICATION");
ImageIcon icon= new ImageIcon(img);

Image
image=icon.getImage().getScaledInstance(bd.jLabel10.getWidth(),
bd.jLabel10.getHeight(),Image.SCALE_SMOOTH);
bd.jLabel8.setText(km);
bd.jLabel10.setIcon(icon);
bd.jLabel12.setText(no);
bd.jLabel14.setText(num);
bd.jTextArea1.setText(sp);

}
}catch(SQLException e)
{
JOptionPane.showMessageDialog(null,e);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
// TODO add your handling code here:
try
{
welcome w=new welcome();

```

```

w.setVisible(true);
    String n=jLabel3.getText();
    w.jLabel3.setText(n);
    Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
    String sql1="select*from DP where USERNAME='"+n+"' ";

Statement st= con.createStatement();
    ResultSet rs=st.executeQuery(sql1);
    while(rs.next()){

String img=rs.getString("IMG");
        ImageIcon icon= new ImageIcon(img);

        Image
image=icon.getImage().getScaledInstance(w.jLabel6.getWidth(),
w.jLabel6.getHeight(),Image.SCALE_SMOOTH);

        w.jLabel6.setIcon(icon);
    }
} catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);
}

```

Credit page :

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.math.BigInteger;
import java.net.HttpURLConnection;
import java.net.URL;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;
import javax.swing.JOptionPane;

/**
 *
 * @author USER
 */
public class credit extends javax.swing.JFrame {
    int OTP;
    PreparedStatement pst=null;
    private BigInteger p;
    private BigInteger q;
    private BigInteger N;
    private BigInteger phi;
```



```

private BigInteger e;
    private BigInteger d;

private int bitlength = 1024;
private Random r;
byte[] encrypted;

/**
 * Creates new form signin
 */
public credit() {
    initComponents();

    r = new Random();

    p = BigInteger.probablePrime(bitlength, r);
    q = BigInteger.probablePrime(bitlength, r);
    N = p.multiply(q);
    phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
    e = BigInteger.probablePrime(bitlength / 2, r);
    while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi)
    < 0)

    {

    e.add(BigInteger.ONE);
    }
    d = e.modInverse(phi);
    }

public credit(BigInteger e, BigInteger d, BigInteger N)
{

this.e = e; this.d = d; this.N = N;
}

private static String bytesToString(byte[] encrypted)

{

String test = "";

for (byte b : encrypted)
{

```

```

test += Byte.toString(b);

}

return test;

}

public byte[] encrypt(byte[] message)

{

return (new BigInteger(message)).modPow(e, N).toByteArray();

}


public byte[] decrypt(byte[] message)
{

return (new BigInteger(message)).modPow(d, N).toByteArray();


}


private void jTextField2MouseEntered(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
    jTextField2.setText("");
}


private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    try {
        signin rsa = new signin();

        //JOptionPane.showMessageDialog(null, "Encrypted sucessfully
message is"+" "+bytesToString(en.getBytes()));


        //JOptionPane.showMessageDialog(null, "Encrypted form is      "+"
"+ encrypted);

```

```

        String apiKey = "apikey=" +
"YWZiMWUxYTBjMGQ1Y2RiN2Y2Nzg4YTc5NGVmMGE0N2M=";
        Random rand=new Random();
        OTP=rand.nextInt(999999);
        String name=jTextField4.getText();

```

```

String mge = ""+OTP;
        String sender = "&sender=" + "TXTLCL";
        String numbers = "&numbers=" + jTextField5.getText();
        jLabel11.setText(mge);
        String en= jLabel11.getText();
        encrypted= rsa.encrypt(en.getBytes());

```

```

byte[] numbe=bytesToString(en.getBytes()).getBytes();
        String enc=new String(numbe);
        byte[] decrypted = rsa.decrypt(encrypted);
        bytesToString(decrypted);
        String S=new String(decrypted);
        String message = "&message=" + "Hey "+name+ " your Encrypted OTP
is "+enc+" And your decrypted OTP is " +S+" give the encrypted OTP to
proceed to buy. ";

```

```

        HttpURLConnection conn = (HttpURLConnection) new
URL("https://api.textlocal.in/send/?"+apiKey).openConnection();
        String data = apiKey + numbers + message + sender;
        conn.setDoOutput(true);
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Content-Length",
Integer.toString(data.length()));
        conn.getOutputStream().write(data.getBytes("UTF-8"));
        final BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        final StringBuffer stringBuffer = new StringBuffer();
        String line;
        while ((line = rd.readLine()) != null) {
            stringBuffer.append(line);
        }
        rd.close();
//return stringBuffer.toString();

```

```

        JOptionPane.showMessageDialog(null,"OTP has sended
successfully...");
        jTextField6.setText(enc);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"Error SMS "+e);
    }
}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here
    String n=jLabel8.getText();
    String no=jLabel9.getText();
    try{
        Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
        String sql="select*from SIGNIN where USERNAME='"+n+"'";

        Statement st= con.createStatement();
        ResultSet rs=st.executeQuery(sql);

        while(rs.next()){
            String p=rs.getString("PHONENUMB");
            String a=rs.getString("ADDRESS");
            String ds=rs.getString("DSD");

String sql1="select*from SELL where CC='"+no+"'";
            st= con.createStatement();
            rs=st.executeQuery(sql1);
            while(rs.next()){
                String na=rs.getString("MODEL");
                String k=rs.getString("KM");
                String pr=rs.getString("PRICE");
                String s=rs.getString("SPECIFICATION");
                String fs=rs.getString("NEOL");
                String y=rs.getString("Y");
                String num=rs.getString("NUMB");
                String o=rs.getString("NAME");

```

```

        if(Integer.parseInt(jTextField6.getText())==OTP){
JOptionPane.showMessageDialog(null,"OTP is successfully verified click
ok to get receipt");
    receipt r=new receipt();
    r.setVisible(true);

```

```

r.area.setText(r.area.getText()+"*****
*****\n");
r.area.setText(r.area.getText()+"          ONLINE CAR
PUCHASING RECEIPT          \n");

```

```

r.area.setText(r.area.getText()+"*****
*****\n");

```

```

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy ");

```

```

    Date date = new Date();

```

```

    r.area.setText(r.area.getText()+"
" +date+ "\n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+" your details :\n");

```

```

    r.area.setText(r.area.getText()+" ----- \n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+"          Name :"+n+ "\n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+"Phone number :"+p+ "\n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+"Address :"+a+ "\n");

```

```

    r.area.setText(r.area.getText()+"\n");

```

```

    r.area.setText(r.area.getText()+"          "
+ds+"\n");

```

```

    r.area.setText(r.area.getText()+"
*****\n");

```

```

    r.area.setText(r.area.getText()+"          Digital
Signature\n");

```

```

r.area.setText(r.area.getText()+"*****
*****\n");

```

```

    r.area.setText(r.area.getText()+"          " Car details :\n");

```

```

r.area.setText(r.area.getText()+" -----\n");
r.area.setText(r.area.getText()+"                               "+fs+"\n");
r.area.setText(r.area.getText()+"Car name :          " +na+ "\n");

```

```

r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"Car number :        " +no+ "\n");
r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"Year of manufacture :      " +y+ "\n");
r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"kilometer travelled : " +k+ "\n");
r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"car problem :      " +s+ "\n");
r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"Car price :        " +pr+ "\n");

```

```

r.area.setText(r.area.getText()+"*****\n");
r.area.setText(r.area.getText()+"                To contact owner "+num+"\n");

```

```

r.area.setText(r.area.getText()+"*****\n");
r.area.setText(r.area.getText()+"\n");
r.area.setText(r.area.getText()+"                *THANK YOU FOR\nSHOPPING*\n");

```

```

String sql2="insert into
SOLD(NAME,CAR,PRICE,FRS,CARNO,OWNER,ONO)
values('"+n+"','"+na+"','"+pr+"','"+fs+"','"+no+"','"+o+"','"+num+"')";
st= con.createStatement();
st.executeUpdate(sql2);

```

```

String sql3="delete from SELL where CC='"+no+"'";
st= con.createStatement();
st.executeUpdate(sql3);

```

```

}
else
{
JOptionPane.showMessageDialog(null,"wrong OTP");
}
}

```

```

    }
    }catch(SQLException e)
{
JOptionPane.showMessageDialog(null,e);
}

}

private void jTextField3MouseEntered(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
    jTextField3.setText("");
}

private void jTextField1MouseEntered(java.awt.event.MouseEvent evt)
{

// TODO add your handling code here:
    jTextField1.setText("");
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    try{
        signin rsa = new signin();
        String n=jLabel8.getText();
        String en= jTextField7.getText();
        JOptionPane.showMessageDialog(null, "Encrypted sucessfully message
is "+" "+bytesToString(en.getBytes()));
        encrypted= rsa.encrypt(en.getBytes());
        byte[] numbe=bytesToString(en.getBytes()).getBytes();
        String enc=new String(numbe);
        //JOptionPane.showMessageDialog(null, "Encrypted form is "+" "
"+ encrypted);
        byte[] decrypted = rsa.decrypt(encrypted);
        bytesToString(decrypted);
        String S=new String(decrypted);
        //JOptionPane.showMessageDialog(null, "Decrypted form is "+" " + S);
        JOptionPane.showConfirmDialog(this, "click yes to verify the signature",
"confirm", JOptionPane.YES_NO_OPTION);

```

```
Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
```

```
String sql="select*from SIGNIN where USERNAME='"+n+"' ";
Statement st= con.createStatement();
ResultSet rs=st.executeQuery(sql);
while(rs.next()){
    String de=rs.getString("DSE");

    if(enc.equals(de)){

        JOptionPane.showMessageDialog(null, "successfully verified
"+" " + S);
    }else{
        JOptionPane.showMessageDialog(null, "wrong signature you
are trying to duplicate");
        jTextField7.setText("");
    }
}
```

```
}catch(SQLException e)
{
JOptionPane.showMessageDialog(null,e);
}
}
```

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    signin rsa = new signin();
    String en= jLabel11.getText();
    encrypted= rsa.encrypt(en.getBytes());
    byte[] decrypted = rsa.decrypt(encrypted);

    bytesToString(decrypted);
    String S=new String(decrypted);
    jTextField6.setText(S)
```


Return page :

```
import java.awt.Image;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
/**
```

```
 *
```

```

* @author USER
*/
public class retur extends javax.swing.JFrame {

    /**
     * Creates new form retur
     */
    public retur() {
        initComponents();
        Show();
    }

    public ArrayList<returnn> userList(){
        ArrayList<returnn> userList=new ArrayList<>();
        try{
            String n= jLabel1.getText();
            Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject","
root","Nasusa9866@");
            String sql="select* from SOLD where NAME='"+n+"' ";
            Statement st= con.createStatement();
            ResultSet rs= st.executeQuery(sql);
            returnn Users;
            while(rs.next()){
                Users=new
returnn(rs.getString("NAME"),rs.getString("CAR"),rs.getString("PRICE")
,rs.getString("FRS"),rs.getString("CARNO"),rs.getString("OWNER"),rs.g
etString("ONO"));
                userList.add(Users);

            }

        }catch(SQLException e)
{
JOptionPane.showMessageDialog(null,"something went wrong");
}

        return userList;
    }

    public void Show(){
        ArrayList<returnn>list=userList();
        DefaultTableModel model=(DefaultTableModel)jTable1.getModel();
        Object[] row=new Object[50];
        for(int i=0;i<list.size();i++){
            row[0]=list.get(i).getName();
            row[1]=list.get(i).getCAR();

```

```

row[2]=list.get(i).getPRICE();
    row[3]=list.get(i).getFRS();
    row[4]=list.get(i).getCARNO();
    row[5]=list.get(i).getOWNER();
    row[6]=list.get(i).getONO();

    model.addRow(row);
}
}

/**
 * This method is called from within the constructor to initialize the
form.
 * WARNING: Do NOT modify this code. The content of this method is
always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CL
OSE);

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Name", "car name", "price", "F or S", "car number", "owner",
"owner number"
        }
    ));
    jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jTable1MouseClicked(evt);
        }
    });
    jScrollPane1.setViewportViewView(jTable1);

```

```

jButton1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jButton1.setText("Return");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jButton2.setText("Back");

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 10)); // NOI18N

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

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.DEFAULT_SIZE, 585, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addGap()
                .addComponent(jButton2,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 92,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(124, 124, 124)
                .addComponent(jLabel1,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 123,
                    javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButton1,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 94,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap())
        );
    layout.setVerticalGroup(

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

```

```

        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 261,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
                ATED)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
                nt.LEADING)
                .addComponent(jLabel1,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 24, Short.MAX_VALUE)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
                    nt.BASELINE)
                        .addComponent(jButton1)
                        .addComponent(jButton2)))
                .addGap(9, 9, 9))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:

    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        return bd=new returnd();
        bd.setVisible(true);
        DefaultTableModel model=(DefaultTableModel)jTable1.getModel();
        int i=jTable1.getSelectedRow();
        String n=model.getValueAt(i,0).toString();
        String cn=model.getValueAt(i,1).toString();
        String pr=model.getValueAt(i,2).toString();
        String fs=model.getValueAt(i,3).toString();
        String cno=model.getValueAt(i,4).toString();
        String o=model.getValueAt(i,5).toString();
        String ono=model.getValueAt(i,6).toString();

        bd.jLabel2.setText(n);
        bd.jLabel4.setText(cn);
    }

```

```
bd.jLabel6.setText(pr);
    bd.jLabel8.setText(fs);
    bd.jLabel10.setText(cno);
    bd.jLabel12.setText(o);
    bd.jLabel14.setText(ono);
    try{
```

```
Connection con
=DriverManager.getConnection("jdbc:derby://localhost:1527/carproject",
root,"Nasusa9866@");
```

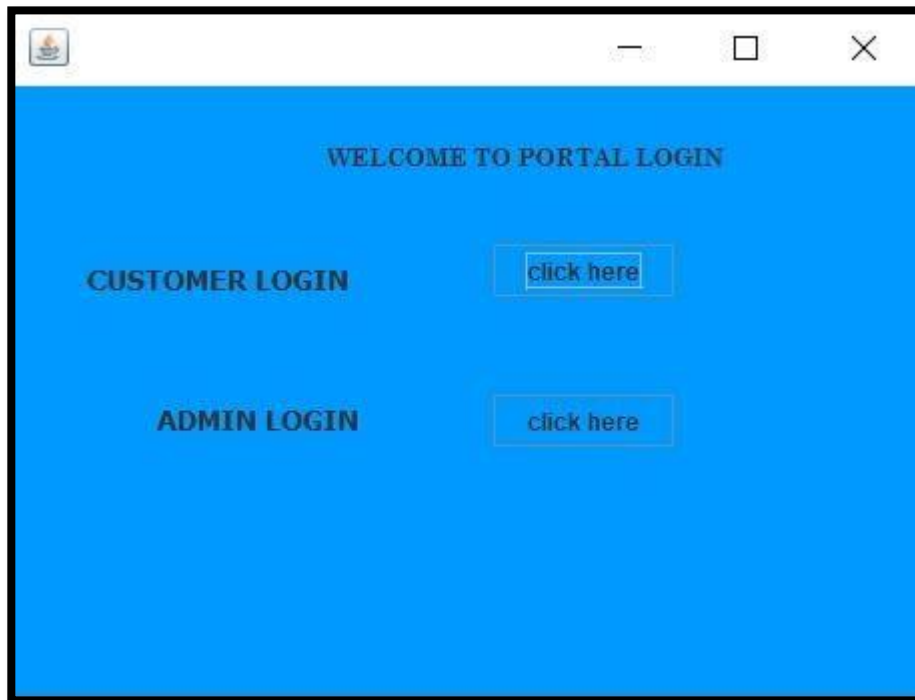
```
String sql="select*from SELL where CC='"+cno+"' ";
Statement st= con.createStatement();
ResultSet rs=st.executeQuery(sql);
while(rs.next()){
    // String km=rs.getString("KM");
    String img=rs.getString("IMAGE");
```

```
        ImageIcon icon= new ImageIcon(img);
```

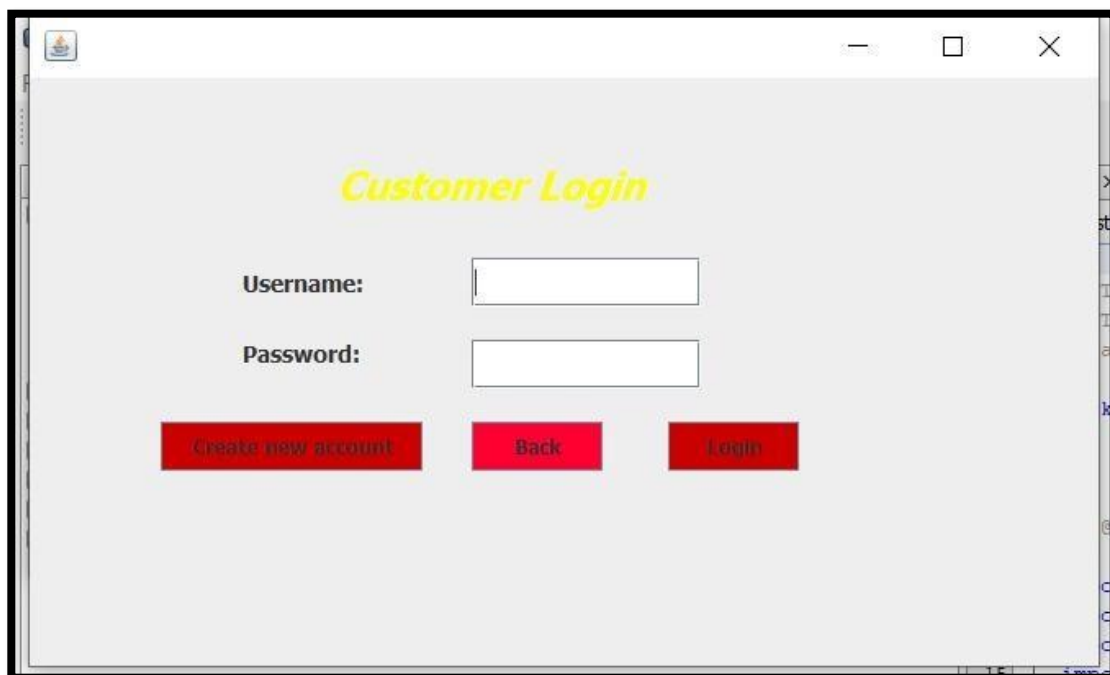
```
    }
    }catch(SQLException e)
{
JOptionPane.showMessageDialog(null,e);
}
}
```

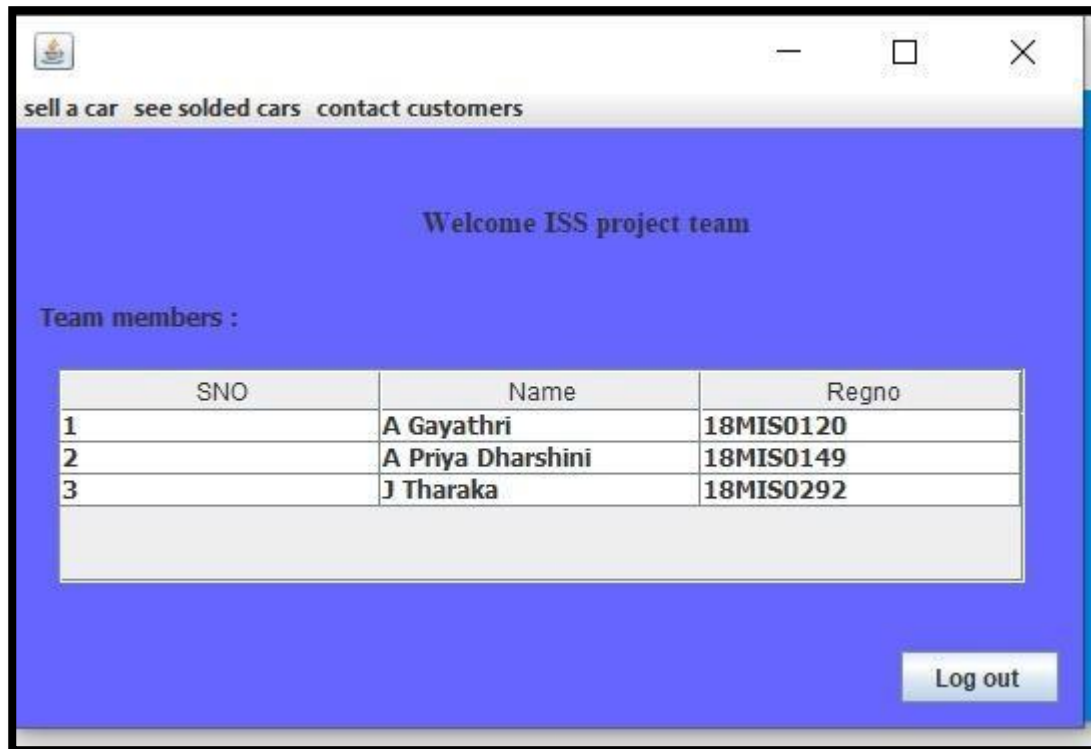
RESULT:

Welcome page:

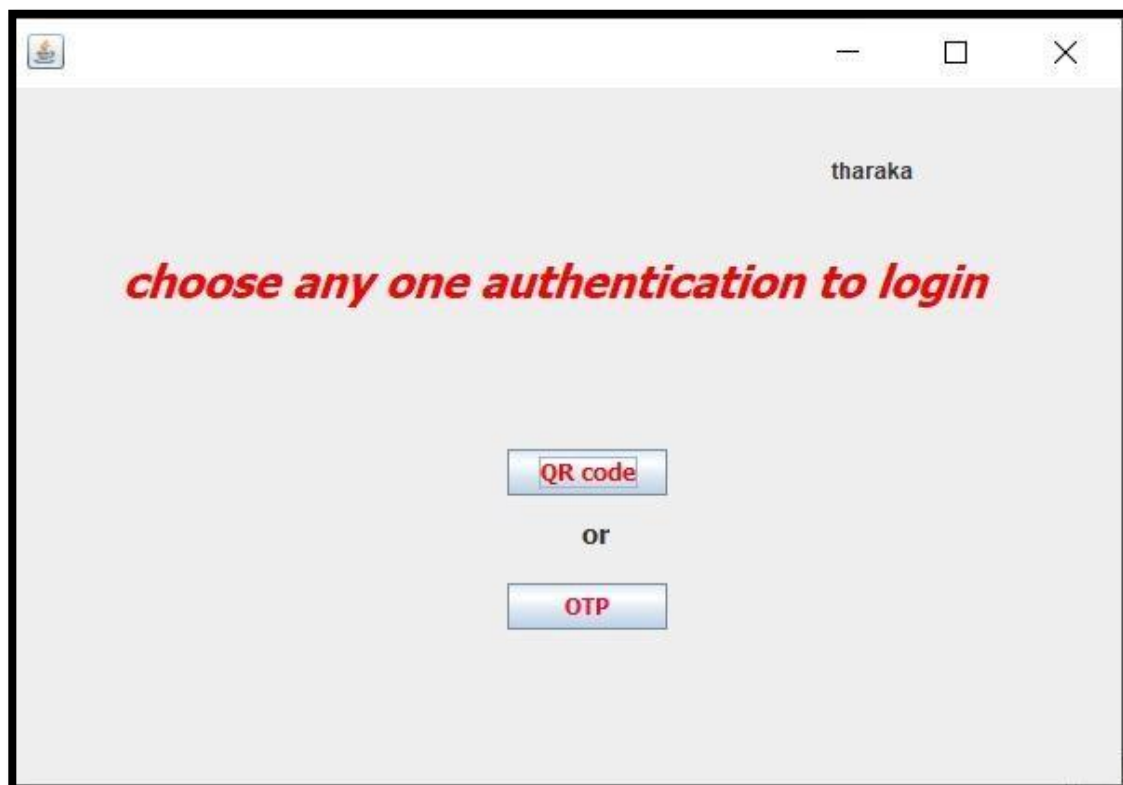


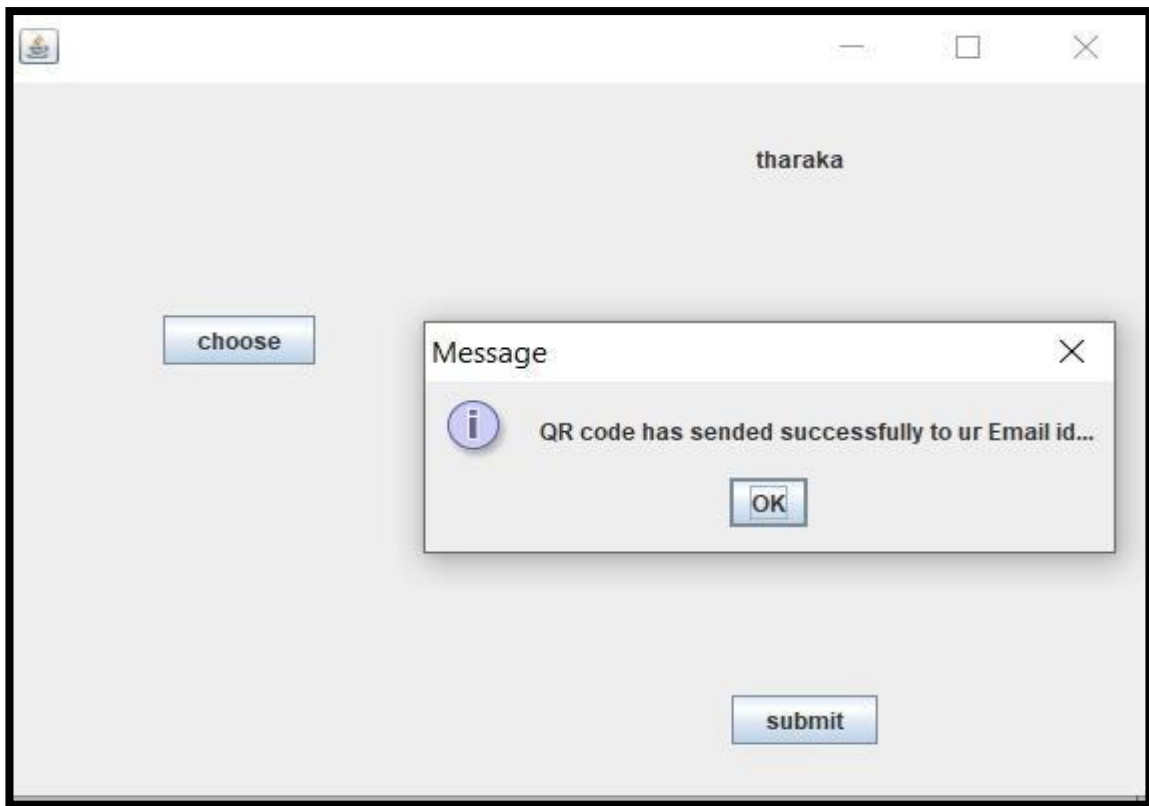
Login page:



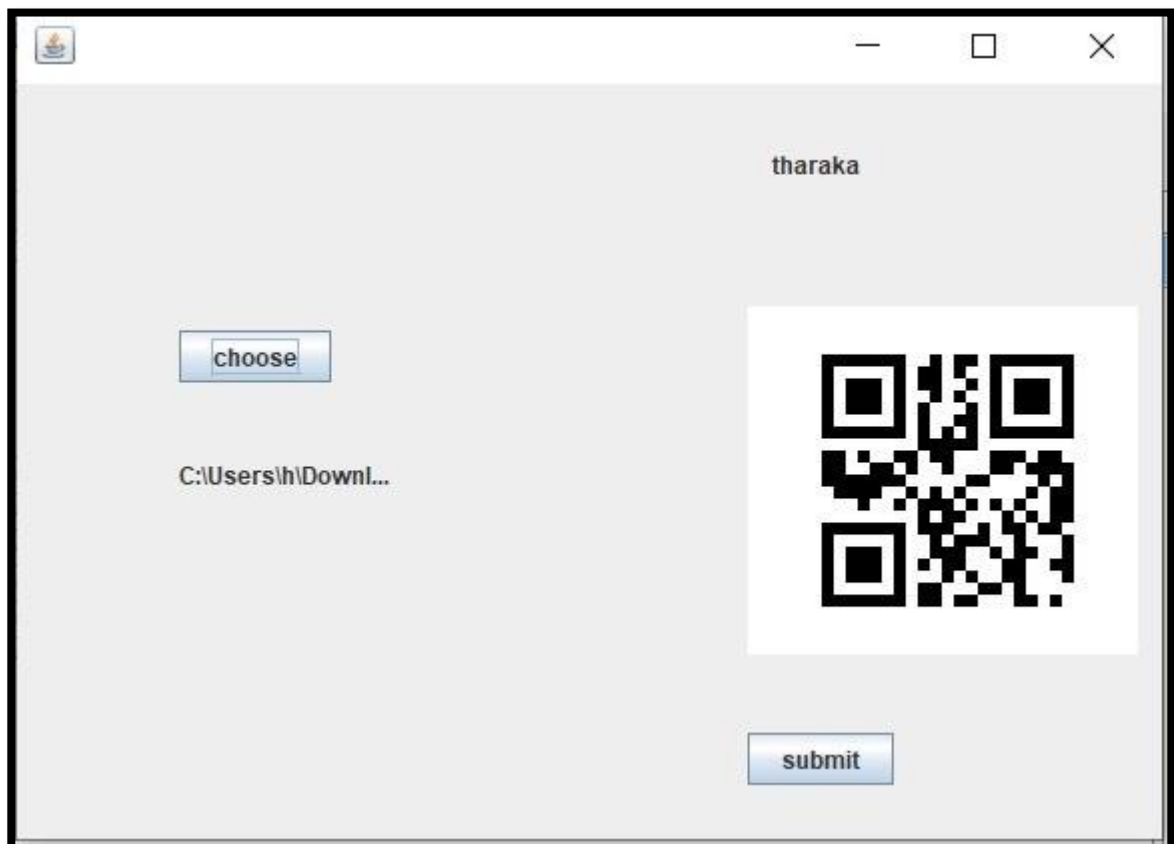


Authentication:





Qr Code:



Buy a car:

ONLINE CAR PURCHASE AND SALES SYSTEM

welcome tharaka

Attach Insert

jLabel8

To buy a car ...

To buy a car ...

CLICK HERE..

CLICK HERE..

CLICK HERE..

Modify account details Logout

Data set:

Search a car to buy.....

tata

Search

tharaka

(Search cars only in small letters)

car model	F or S	Price	Owner of car	car number
tata	First hand	60000	gayathri	ap07jh2636

Ba...

Buying car:

tharaka

First hand

Extra Specification :

Owner : gayathri

KM travelled : 4000

Price : 60000

Model name : tata

Car number : ap07jh2636

To contact owner 4562356423

Back

Proceed to Buy

NAME : tharaka

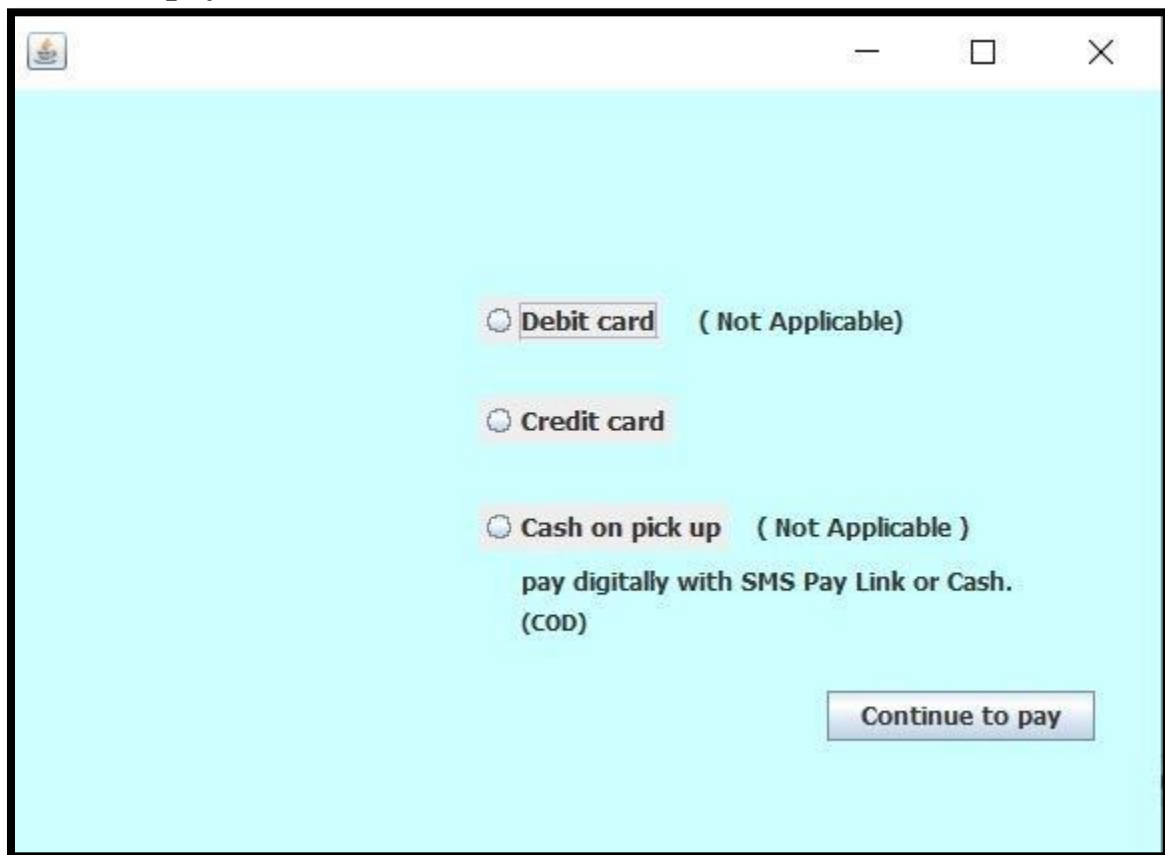
Address 1 :

chittoor,ap

Add new address :

Continue

Card payment:



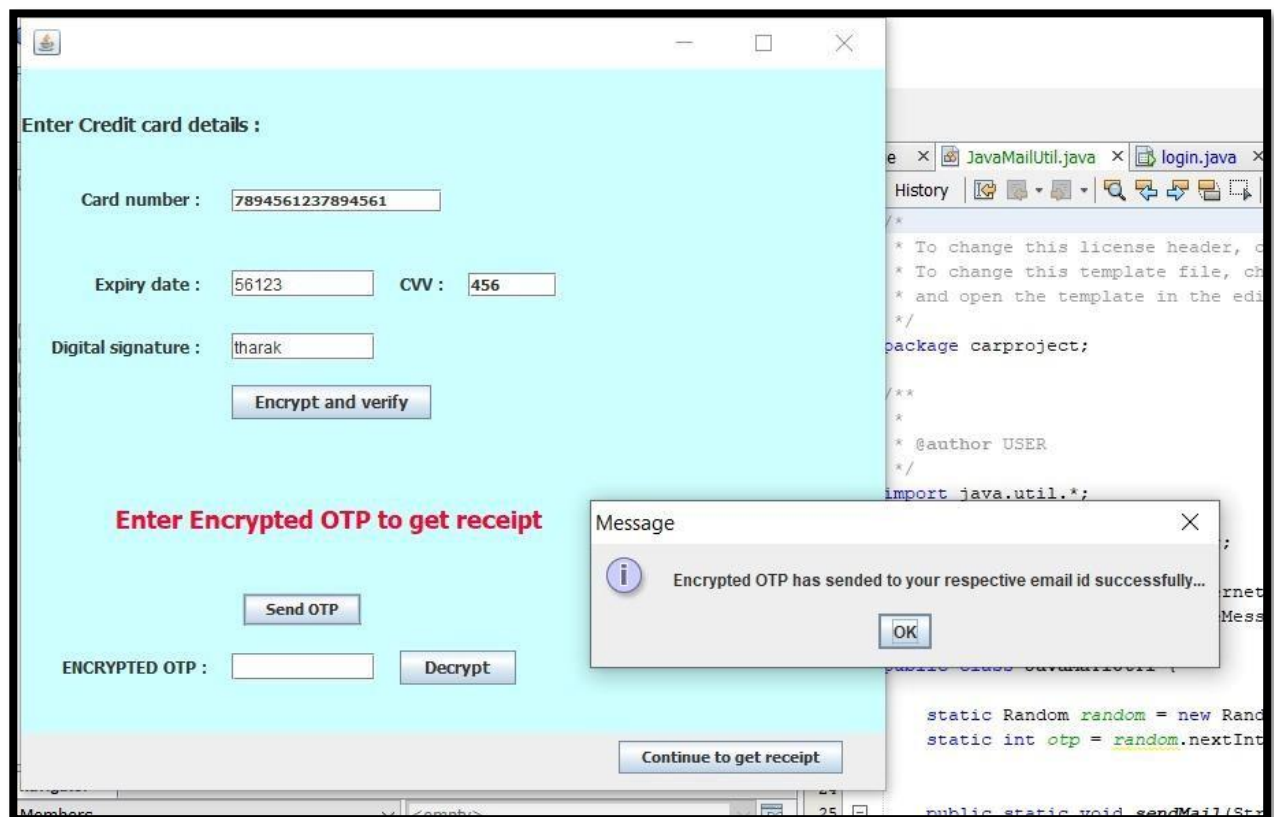
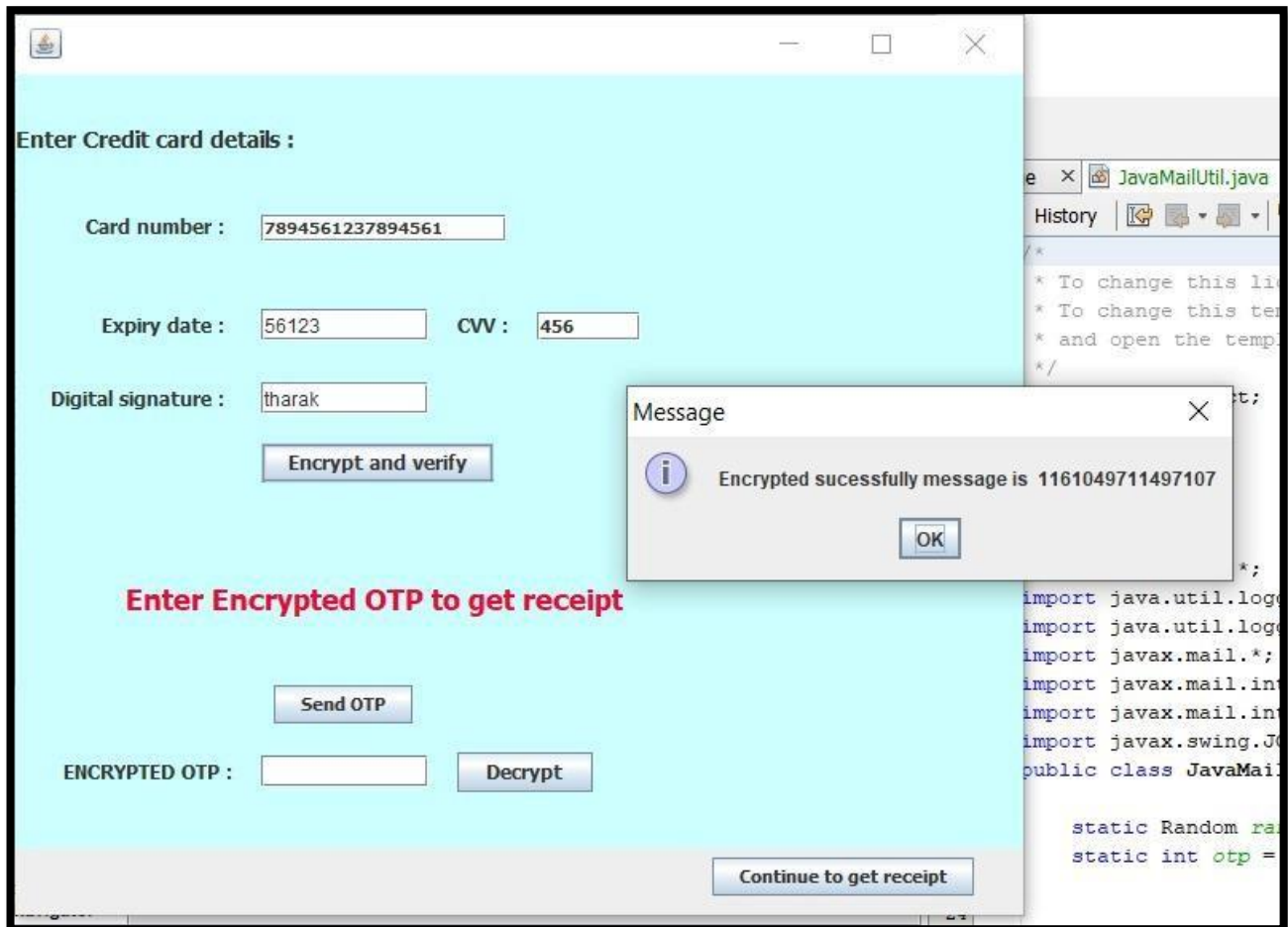
☐ Debit card (Not Applicable)

☐ Credit card

☐ Cash on pick up (Not Applicable)

pay digitally with SMS Pay Link or Cash.
(COD)

[Continue to pay](#)



OTP encryption and decryption:

The screenshot shows a JavaMailUtil.java application window with a light blue background. It contains a form for entering credit card details and an OTP verification section. A message dialog box is open, indicating successful OTP verification.

Enter Credit card details :

Card number :

Expiry date : CVV :

Digital signature :

Enter Encrypted OTP to get receipt

ENCRYPTED OTP :

Message

OTP is successfully verified click ok to get receipt

```
package carproject;

import java.util.*;
import java.util.logging.Level;

static Random random = new Random();
static int otp = random.nextInt(1000000);
```

Receipt:

Thu Apr 21 22:49:17 IST 2022

your details :

Name :tharaka
Phone number :8008162514
Address :chittoor,ap
tharak

Digital Signature

Car details :

First hand
Car name : tata
Car number : ap07jh2636
Year of manufacture : 2015
kilometer travelled :4000
car problem :
Car price : 60000

To contact owner 4562356423

THANK YOU FOR SHOPPING

Print Receipt

Discussion:

In the current payment systems, a customer's payment info is distributed to a payment gateway via a merchant. This makes the payment system at risk of intrusions and data leaks, inflicting client knowledge felony, fraud and deceitful transactions. to guard a customer's money info from being compromised, we tend to developed an approach for on-line payment systems during which a customer's payment info is directly provided to a payment gateway instead of sent through a merchant.

CONCLUSION:

We showed that the proposed system works practically in the credit card payment environment. We can tremendously decrease the number of credit card fraud. To acquire this goal in the process described about the transactions are encrypted with RSA algorithm. RSA is widely and popularity cryptography model. So this process provides two layer of security and it can provide security in the case of stolen or lost. With the proposed algorithm, transactions can be done securely without worrying about the attacker.

In this research, a detailed implementation of 1024-bit RSA encryption/decryption algorithm is presented for use in securing ecommerce payment information. The RSA algorithm has remained a secure scheme for sending encrypted messages for almost 40 years, earning Rivest, Shamir, and Adleman the Association for Computing Machinery's 2002 Alan Turing Award, among one of the highest honours in computer science. RSA keys are typically 1024 to 2048 bits long, though some experts believe that 1024-bit keys could be broken in the near future. It is generally believed that 4096-bit keys are unlikely to be broken in the foreseeable future, meaning that RSA should remain secure as long as n is chosen to be sufficiently large. It is currently recommended that n be at least 2048 bits long.

REFERENCES:

1. <https://ieeexplore.ieee.org/document/4244863>
2. <https://ieeexplore.ieee.org/document/6567226>
3. <https://ieeexplore.ieee.org/document/1656672>
4. <https://www.researchgate.net/publication/312119638> The RSA Algorithm Explored
5. https://www.schneier.com/blog/archives/2008/12/credit_card_wit.html