References

[1] Juneja, Ashok, (Juneja, 2007) . Source: *Washington Post*

[2] (Report, 2006) Micro Payment Ref Sect 2.2 and 2.3

[3] (Report, 2006) MicroPayement Ref Sect. 2

[4] http://www.gsmworld.com/mmt/index.shtml

[5] http://www.fundamo.com/solutions.aspx

[6]http://globaltechforum.eiu.com/index.asp?layout=rich_story&channelid=4&categoryid=31&title=Africa%3A+M-commerce+finds+a+home&doc_id=11053

[7] http://www.nextbillion.net/activitycapsule/topic/general-banking

[8] (Presentation, 2005) G-Cash Presentation.

[9] **(**Workshop 19 December 2006) m-banking, m-remittances Case studies from the Philippines Michael Trucano, *info*DevICT and Social Sector Innovation Specialist Nigeria FSS 2020

[10] http://www.onjava.com/pub/a/onjava/2006/03/08/what-is-java.html

[11]http://www.justobjects.org/cowcatcher/browse/stdjava/slides/java-concept/core/whatis/slide.0.5.html

[12]http://profile.iiita.ac.in/rverma2_01/ranjana/po/grpmembers/grace/16-08-2004/jSMSEngine.htm

[13] http://smslib.sourceforge.net/

[14] http://java.sun.com/products/jsp/overview.html

[15] Chris Durachka (Presentation, 2004) JSP - What is it Good For? ,pp 1-20

[16] http://en.wikipedia.org/wiki/AJAX

[17]http://www.w3schools.com/Ajax/ajax_intro.asp

[18] http://en.wikipedia.org/wiki/AJAX

[19] http://www.xul.fr/en-xml-ajax.html

[20] http://en.wikipedia.org/wiki/MySQL

[21] http://en.wikipedia.org/wiki/Log4j

[22] http://www.laliluna.de/download/log4j-tutorial-en.pdf

[23] http://searchciomidmarket.techtarget.com/sDefinition/0sid183_gci516025,00.html

[24] http://en.wikipedia.org/wiki/Unified_Modeling_Language

[25] http://en.wikipedia.org/wiki/Image:Uml_diagram.svg

[26] http://en.wikipedia.org/wiki/Waterfall_model

[27]http://www.thriftyscot.co.uk/Banking-Savings/022007/pilot-program-for-mobile-money-transfer.html

[28] http://www.remittances.info/

[29] http://en.wikipedia.org/wiki/Electronic_money

[30] Dr. karunananda, (2008), Guidelines for preparation of dissertations MSc in IT

Appendix A

Design Diagrams

Use case diagrams

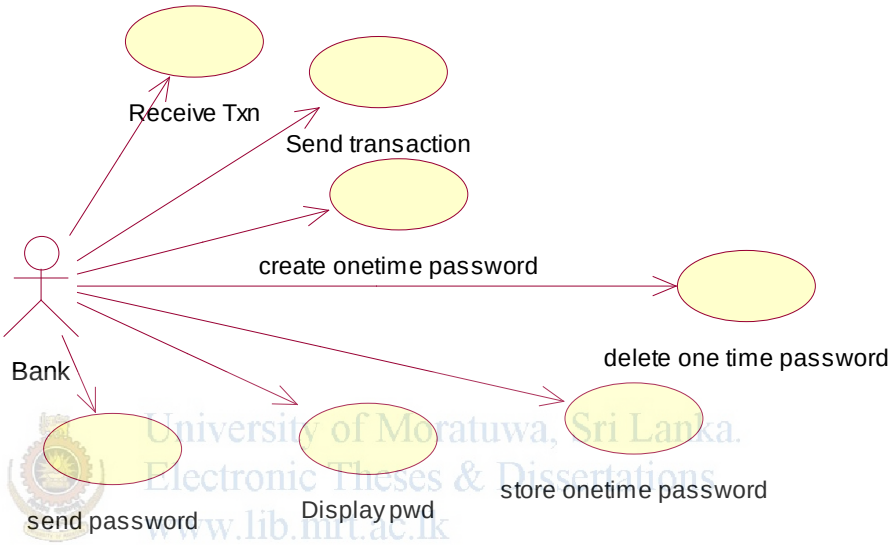


Figure A.1 – Bank use case diagram

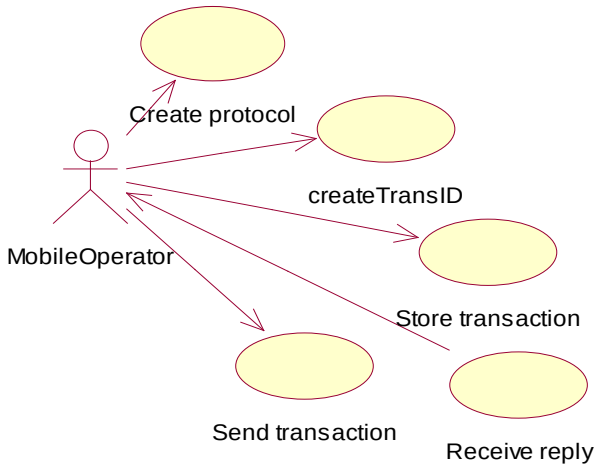


Figure A.2 – Mobile operator use case diagram

Activity Diagrams

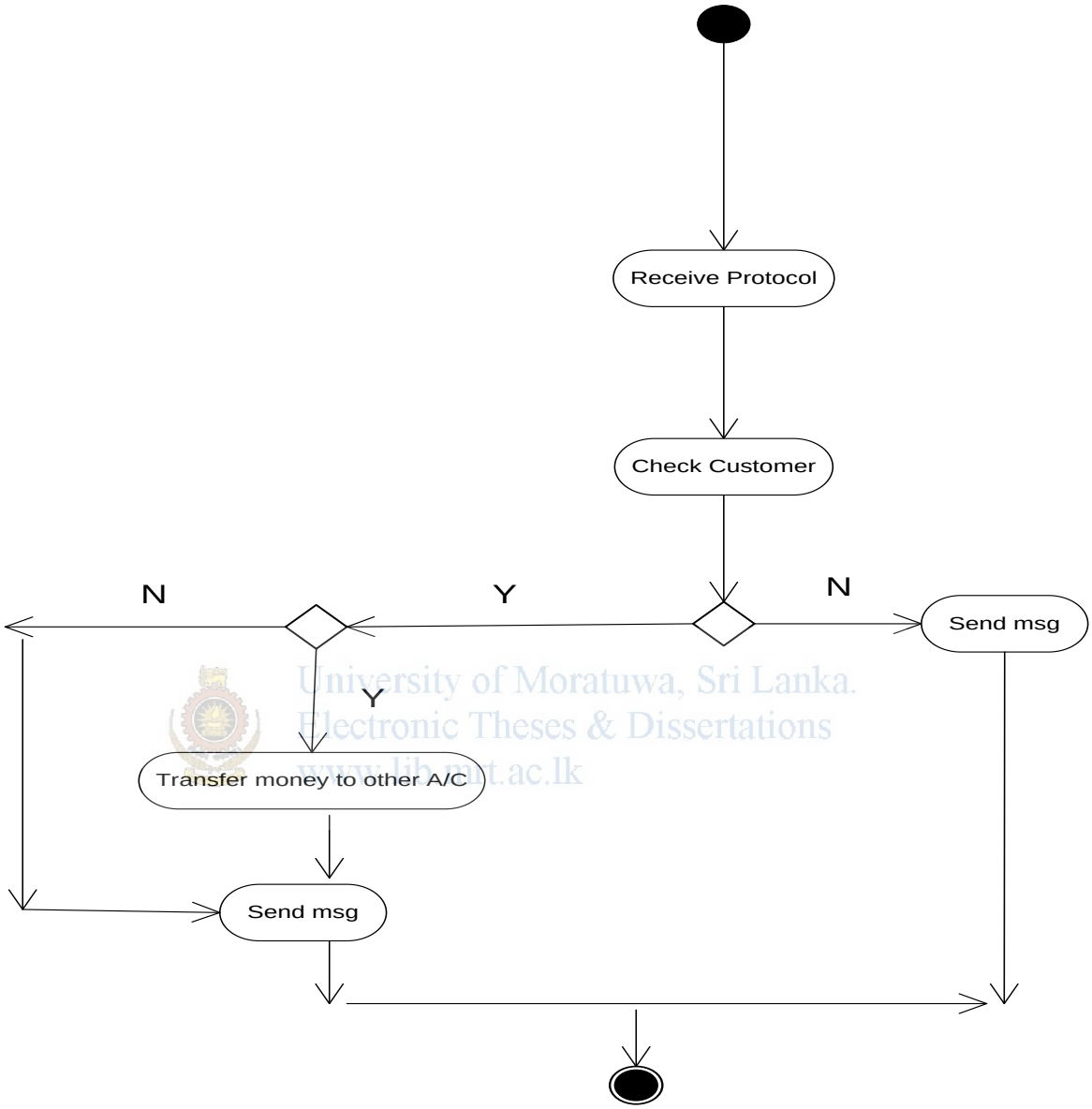


Figure A.3 – Activity Diagram, Bank – Receive reply- peer to peer money transfer.

Retailer – check customer validity

Figure A.4– Cashless purchasing system- Retailer verify customer.

Sequence Diagrams

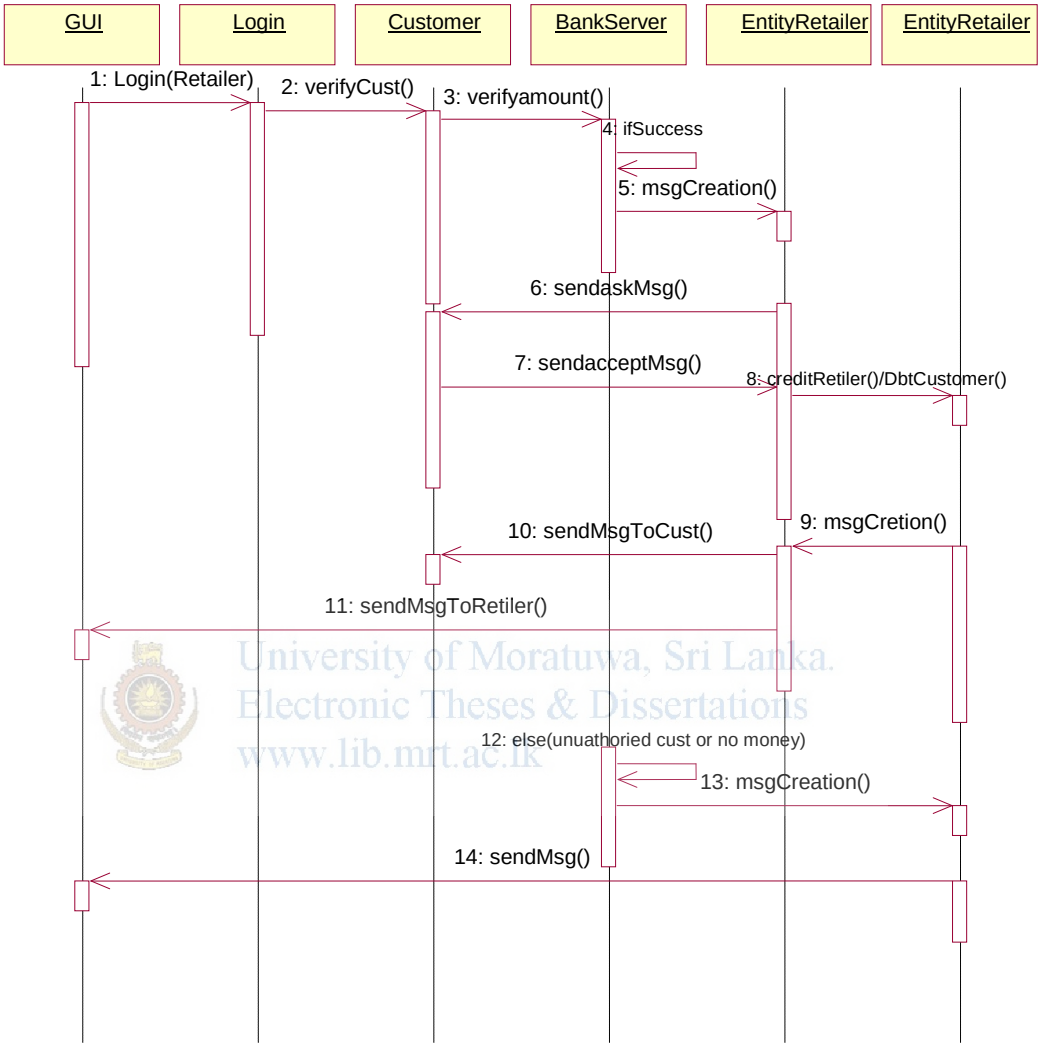


Figure A.5– Sequence diagram, Cashless purchasing at super market.

Project schedule

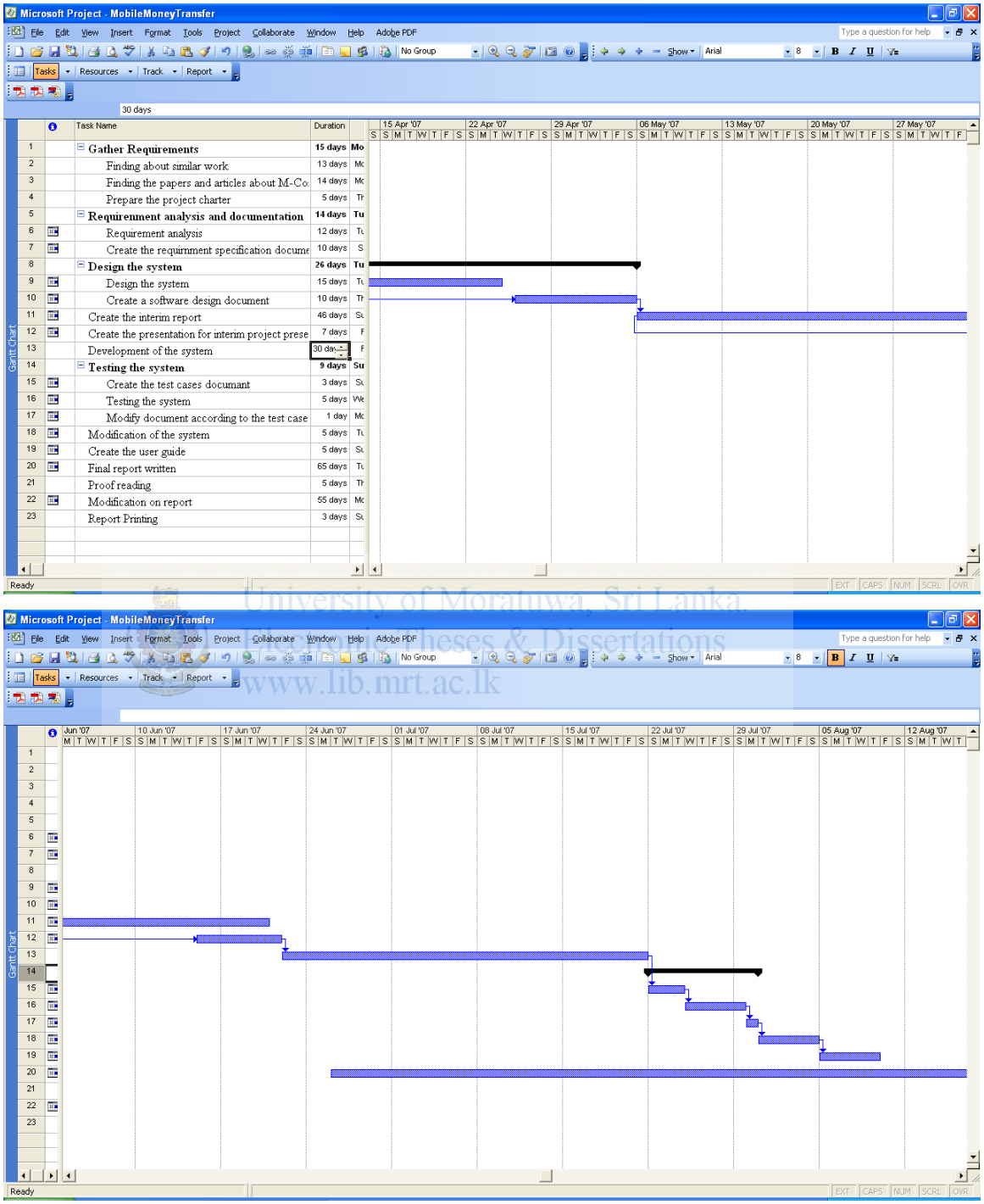


Figure A.6- Gantt chart for project schedule.

Appendix B

Implementation Details and User Interfaces Design

Chapter includes pseudo code of the applications, sample code segments and user interfaces

1. SMS Module

Pseudo Code - Read Message

```
Always server is connected
Messages collect to the list
If (message == new message){
 Read message
  If (message.length > 7)
        Store message.
  Else
        Reject message}

Delete read message
Clear message list
Server disconnected
```

Send message code – Send message

```
Server connect
Get number list (if flag==3)
Traverse through the list
Create COutgoingMessage class
Send message
Increment++
```

Code Segments

```
public static CService connectToServer(){
  try{
   srv = new CService("COM7", 38400);
   //    Initialize service.
    srv.initialize();
    //   Set the cache directory.
    srv.setCacheDir(".\\");
    System.out.println("SERVER Connection Success");
  }catch(Exception e){
    System.out.println("]SendMessageNew()][connectToServer] "+e.toString());

  }
  return srv;
 }
```


ReadMessage class

```
 status = srv.connect();
        System.out.println("Connectio ok To read message " + status);
        if (status == CService.ERR_OK) {
           //Set the operation mode to PDU
           srv.setOperationMode(CService.MODE_PDU);

           if   (srv.readMessages(msgList,   CIncomingMessage.CLASS_REC_UNREAD)
== CService.ERR_OK) {
              for (int i = 0; i < msgList.size(); i++) {
                 CIncomingMessage msg = (CIncomingMessage) msgList.get(i);
                 String Msisdn = msg.getOriginator();
                 String newMsisdn = Msisdn.substring(3, Msisdn.length());
          }}
```

SendMessage class

```
status = srv.connect();
if (status == CService.ERR_OK) {
vecsms            = entity.sendPeerToPeerReplySMS();
int ireplyVec      = vecsms.size();
          try {
              while (j < ireplyVec) {
                 System.out.println("Inside  While peer send");
                 record = vecsms.elementAt(j).toString();
                 strArr = record.split("\\|");
                 COutgoingMessage msg = new COutgoingMessage(strArr[0], strArr[1]);
                 if (srv.sendMessage(msg) == CService.ERR_OK) {
                    System.out.println("Reply Message Sent!" + strArr[0]);
                    entity.updateReplyMsgSendSender(strArr[0]);
                    entity.updateReplyMsgSendReceiver(strArr[0]);
                 } else {
                    System.out.println("Reply Message Failed!");
                 }
                 j++;
              }
```

2. Database module

Database module is used to store and retrieve data.

Software: - Java 5

Hardware: - Computer.

Classes:- DBConnection

EntityPeerToPeer

EntityBE

EntityBankPeer

Pseudo Code

EntityBE class

If(userid && password){

Login to the system.

}

Else{

Display error message

}

Code segments

DBConnection

```
public Connection getDBConnection(){
   try{
   String userName = "";
   String password = "";
   String url = "jdbc:mysql://localhost/moneytransfer";
   Class.forName ("com.mysql.jdbc.Driver").newInstance ();
   conn = DriverManager.getConnection (url, userName, password);
   }catch(SQLException e){
   System.out.println("[DBConnection][getDBConnection() "+e.toString());
```

```
    }
    catch(Exception e){
     System.out.println("[DBConnection][getDBConnection() "+e.toString());

    }
    return conn;
    }
```

EntityBE

```
public boolean isLoginCorrect(String pstrMobile,String password){
    PreparedStatement chkCust = null;
    boolean flag=false;
try{
    chkCust = getDBConnection().prepareStatement("SELECT * from BELogin
WHERE username=? and password=?");
    chkCust.setString(1, pstrMobile);
    chkCust.setString(2, password);
    rs = chkCust.executeQuery();
    if(rs.next()){
    flag = true;
    }
 } catch (SQLException ex) {
          System.out.println("[EntityBankPeer][isLoginCorrect()] "+ex.toString());
       }
       catch(Exception e){
         System.out.println("[EntityBankPeer][isLoginCorrect()] "+e.toString());

       }
       finally{
       dbConnClose();
       } return flag;     }
```

3. Bank application module

Pseudo Code - Socket cilent

```
If(flag==1){
        Select data to send bank
If(message.length>12){
        Send data to bank server
        Update flag
        Connection flush.
        Connection close()
}
}
while ((message=is.readLine()) != null) {
get the message
if(message.length>22)
if(message.startwith(“BN001”)
store acknowledgement message
if(message.startwith(“BN002”)
store reply message
}
Outputstream.close()
Inputstream.close()
Socket.close()
```


MobileOperatorServer

```
while ((inputLine = in.readLine()) != null) {
if(inputLine.length > 50){
split protocol add message}}
```

4. Reporting Modules

```
public String getCustomerStatus(String retID,String dateFrom,String dateTo){
```

```
PreparedStatement getCust = null;
boolean flag=false;
String strFull="";
String header="<table border=1><tr><td>TRANS ID</td><td>CUSTOMER MOBILE</td><td>AMOUNT</td><td>TRANS_DATE</td><td>STATUS</td></tr>";
try{
 java.sql.Date d1 = java.sql.Date.valueOf(dateFrom);
  java.sql.Date d2 = java.sql.Date.valueOf(dateTo);
   getCust = getDBConnection().prepareStatement("SELECT * FROM retalier_cust_txn WHERE retailerID=? and txndate>=? AND txndate<=?");
 getCust.setString(1, retID);
 getCust.setDate(2, d1);
 getCust.setDate(3, d2);
  rs = getCust.executeQuery();

 while(rs.next()){
 String status = rs.getString("flag");
 if(status.equals("1")){
 status = "PENDING - authorized request message does not send yet";
 }
 if(status.equals("2")){
 status = "SEND SMS - Authorized request SMS has been sent";
 }
 if(status.equals("3")){
 status = "Authorized - Customer has given the permission";
 }
 if(status.equals("4")){
 status = "REJECT - Customer has rejected the transaction"; }
if(status.equals("5")){
 status = "SEND Last SMS - Both customer and retailer has been sent the SMS";
```

```
    }
strFull=strFull+"<tr><td>"+rs.getString("transid")+"</td><td>"+rs.getString("custMobil
e")+"</td><td>"+rs.getString("amount")+"</td><td>"+rs.getString("txndate")+"</td><td
>"+status+"</td></tr>";
    }
    } catch (SQLException ex) {
        System.out.println("[EntityBankPeer][getCustomerStatus()] "+ex.toString());
    }
   catch(Exception e){
     System.out.println("[EntityBankPeer][getCustomerStatus()] "+e.toString());
   }
   finally{
   dbConnClose();
   }
    if(strFull.length()==0){
   strFull="<tr><td colspan='5'><font color='red'>No Record Found</font></td></tr>";
   }
      return header+strFull+"</table>";
   }
```


txnRpt.jrxml

```
<?xml version="1.0" encoding="UTF-8"  ?>
<!-- Created with iReport - A designer for JasperReports -->
<!DOCTYPE jasperReport PUBLIC "//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">
<jasperReport
                name="Untitled_report_1"
                columnCount="1"
                printOrder="Vertical"
                orientation="Portrait"
                pageWidth="595"
                pageHeight="842"
                columnWidth="535"
                columnSpacing="0"
                leftMargin="30"
                rightMargin="30"
```

```
                topMargin="20"
                bottomMargin="20"
                whenNoDataType="NoPages"
                isTitleNewPage="false"
                isSummaryNewPage="false">
        <property name="ireport.scriptlethandling" value="2" />
        <property name="ireport.encoding" value="UTF-8" />
        <import value="java.util.*" />
        <import value="net.sf.jasperreports.engine.*" />
        <import value="net.sf.jasperreports.engine.data.*" />
        <parameter name="txndate1" isForPrompting="true" class="java.util.Date"/>
        <parameter name="txndate2" isForPrompting="true" class="java.util.Date"/>
        <queryString><![CDATA[SELECT * FROM retalier_cust_txn WHERE
retailerID="SPU001" and txndate>$P{txndate1} AND
txndate<=$P{txndate2}]]></queryString>
        <field name="transid" class="java.lang.String"/>
        <field name="retailerID" class="java.lang.String"/>
        <field name="custNIC" class="java.lang.String"/>
        <field name="custMobile" class="java.lang.String"/>
        <field name="amount" class="java.lang.Long"/>
        <field name="txndate" class="java.sql.Date"/>
        <field name="flag" class="java.lang.String"/>
```


Pseudo code – Retailer check customer validity

```
If(customer mobile NIC==true){
If(amount<balance+2000){
Add record to the system.
}
}
Else{
Give message – not registered user. Or not enough money to do transaction.
}
while(flag==1){
        send askConfirmationmessage to do transaction.
        Update flag
}
Read confirmation message.
```

If(confmsg ==“YES”){

Deduct amount from the customer and credit to the retailer account.

Send message to the customer.

Send message to the retailer.

Display status in web interface}

else{

Display status in web interface

}

2. User interfaces

Describe the logical characteristics of each interface between the software product and the users. This include sample screen images, any GUI standards that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

2.1 User interface for the peer to peer money transfer. (In mobile device).

This will be a menu based interface. User has to go through the menu interface.

Main Screen

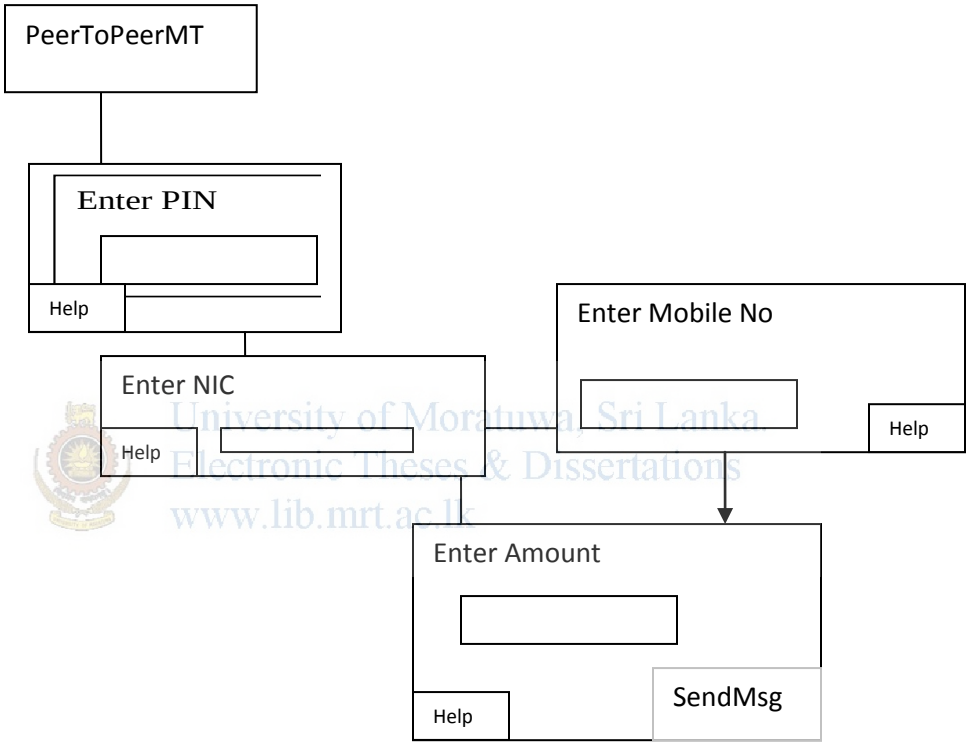


Figure B.1 – Main interface. Peer to peer money transfer.

PIN Number Field:-

This should be enter 6 alpha numeric characters. When user type the pin number it should be appears actual character at once and then it should be converted to the stars.

NIC Field:-

Filed should be alpha numeric characters. Length should be 10 digits.

Amount Field:-

Amount should be numeric only. It doesn't allow entering text and it should be allow entering decimal places.

2.2 Web interface for Peer to Peer Money transfer –

Bank executive is checking the customer transaction. When customer comes to the counter and collects the money.

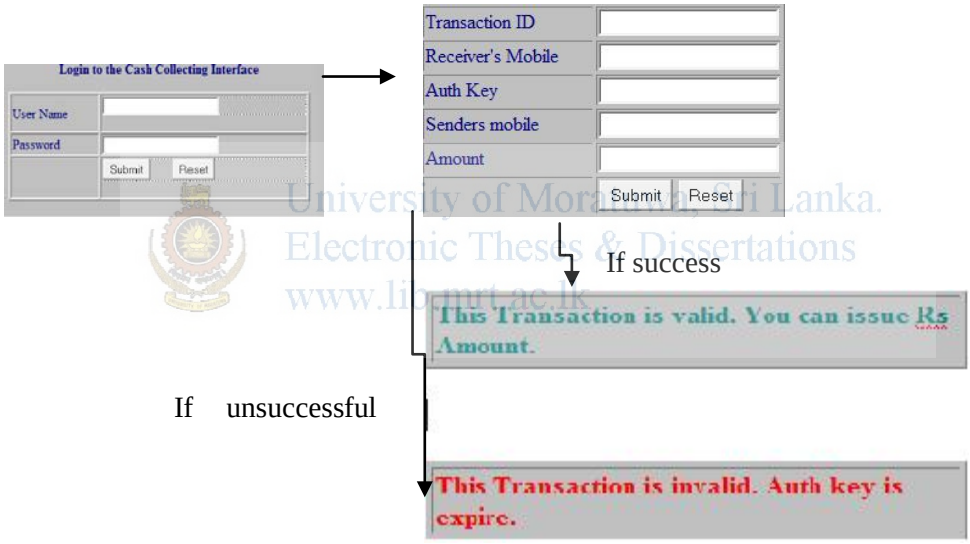


Figure B.2 – Web interface for bank executive.

2.3 Web interface for the Cashless purchasing system.

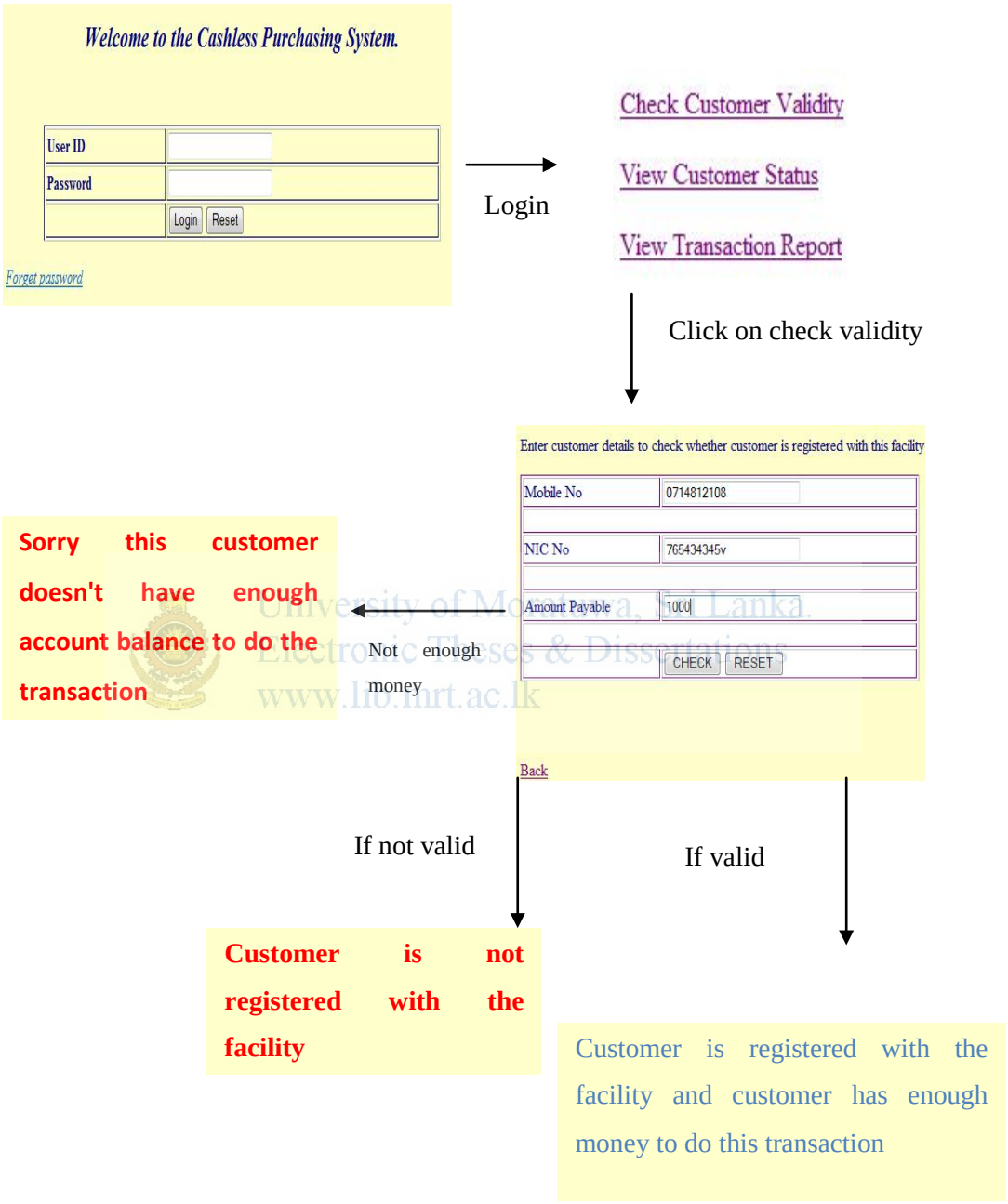
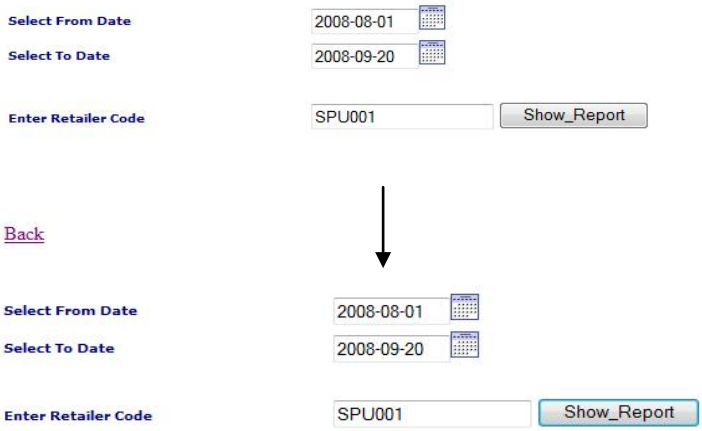


Figure B.3 – Web interface for cashless purchasing

3. Reports Interfaces

3.1 Transaction report for cashless purchasing.



TRANS ID	CUSTOMER MOBILE	AMOUNT	TRANS_DATE	STATUS
000002	0714812108	500	2008-09-07	SEND Last SMS - Both customer and retailer has been sent the SMS
000003	0714812108	1000	2008-09-07	SEND Last SMS - Both customer and retailer has been sent the SMS
000004	0714812108	500	2008-09-07	SEND Last SMS - Both customer and retailer has been sent the SMS
000005	0714812108	500	2008-09-07	SEND Last SMS - Both customer and retailer has been sent the SMS
000006	0714812108	400	2008-09-07	REJECT - Customer has rejected the transaction
000007	0714812108	1000	2008-09-20	PENDING - authorized request message does not send yet
000008	0714812108	100	2008-09-20	PENDING - authorized request message does not send yet

Back

3.2 Transaction report for peer to peer money transfer.

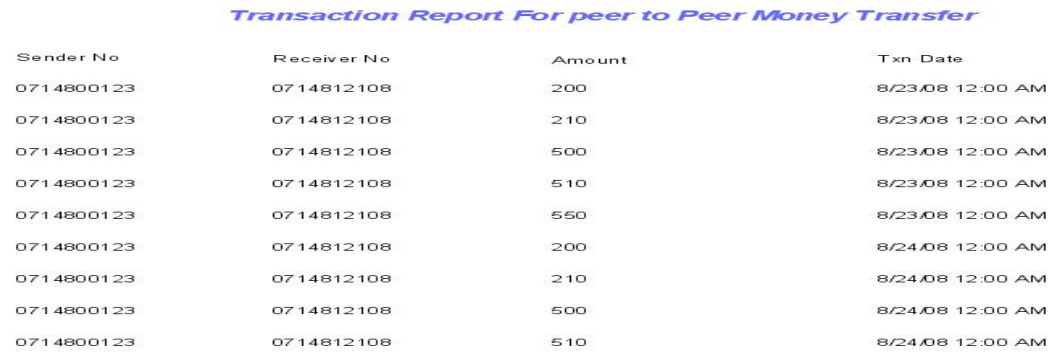


Figure B.4 – Transaction report for cashless purchasing system

Appendix C

Background materials

1. Knowledge of the traditional and existing cash change methods.

".Traditional methods of moving money in small amounts can be slow and costly

• Western Union, the world's largest in money transfer charge $22 in fee on a $26 transfer from London to Manilla.

• Banks also demand substantial fees apart from that there are hardly any branches in rural areas. Most receivers of money do not have a bank account".

Traditional money transferring methods are slow and costly. This is identified by Ashok[1] Juneja, Ashok, (Juneja, 2007) . Source: Washington Post

2. Modern e-commerce and m-commerce methods.

2.1 Without doubt, the largest and arguably most successful m-Commerce applications are to be found in the Philippines with over 3.5 million m-Commerce users on the two major networks. In discussion with the two networks, it was identified that the key success factors for that market included the ability to load prepaid airtime credits as well as the ability to transfer both cash and airtime credits between customers.

Coupled with these were the low values set by the operator for such prepaid top-ups or credit transfers. Typical top-ups of US$ 47 to 57 cents were allowed by the networks (equivalent to around four to five minutes of calls) while transfers between customers of both cash and airtime credits were permitted as low as US$ 4 cents.

Most of the developed countries used mobile based applications and people are highly using those applications. [2].

2.2 In Filipino terminology, the target market was adjusted to 'sachet purchasing' or the practice of purchasing goods in very small quantities. This phenomenon is known to be

common in other developing markets where the populace rely on cash for all trading and can afford to buy provisions for just a few days consumption. This market does not exhibit bulk purchase tendencies and an m-Commerce offering that involves a significant cash deposit or payment will be unlikely to find any significant uptake from the target market.[3]

2.3 Over $275 million is remitted every year from developed economies to emerging markets through recorded channels and the World Bank estimates that half as much again travels through unrecorded channels. The use of the mobile phone as a sending and receiving mechanism for remittances has the potential to enable low denomination remittances (sub $100) to be made much more affordable. This will create a new market, driving the accessibility of remittances globally to reach estimated 1.5-2bn recipients. The impact of this will be to increase the overall penetration and usage of m-transfers, m-payments, and m-banking globally.

"Our vision of money transfer is that it should no longer be restricted to the 'traditional' physical methods of moving money. Money should be available and able to be moved 24x7, 365 days of the year wherever you are.

MMT services will offer the opportunity to send money to in excess of 3 billion mobile phone users across all networks and geographic boundaries. "

Money (Business) should be moved without any time or geographical barriers. [4].

2.4 "It is estimated that some five billion people around the world lack access to traditional financial services. In some cases this is because of lack of ATMs and bank branches, poor regulation, lack of financial literacy or other weaknesses in infrastructure. Meanwhile, international remittances from immigrant and expatriate employees are a significant market, estimated at $257 billion in 2005. And it is estimated that informal remittances make up a similar amount."

2.5 "MasterCard is to provide payment card products and international transaction switching, clearing and settlement. The MasterCard global processing platform can

process transactions in 210 countries and 160 different currencies. It can also switch its part of the transaction in as little as 120 milliseconds which will speed up payments for recipients. GSMA represents more than 700 mobile phone operators worldwide and will work with local or regional banks to deliver the service. MasterCard's president of Global Technology and Applications, Roy Dunbar, comments: ' This pilot provides a unique opportunity to test the use of our global payments products and platform to help create access to the global economy for people facing barriers to participation. We look forward to working with the GSM Association and its member operators in local markets, along with financial institutions, to assess how we can bring much needed payment and money transfer alternatives to the vast community of under banked--as well as all consumers wishing to transfer money internationally." [27]

3 "New opportunities are arising due to communication technologies"

3.1 "Remittances

International remittances are huge, second only to foreign direct investment. The global average transaction cost of sending money home is around 12%. Current fee structures make transfers of small amount of costly. Money transfer within developing countries is also significant to the economy and to house holds.

Mobile Phones with a few exceptions, access to mobile telephone is high in the south even in Africa research shows the majority of households now have access to a phone. 45% of households who have made a call in the last 3 months about remittance. Most phone companies are working on some form of money transfer using a phone. Phones are ideal for handling (and logging may small transactions.)." [28]

1.2 "Most money in today's world is electronic, and tangible cash is becoming less frequent. With the introduction of internet / online banking, debit cards, online bill payments and internet business, paper money is becoming a thing of the past. Banks now offer many services whereby a customer can transfer funds, purchase stocks, contribute to their retirement plans (such as Canadian RRSP) and offer a variety of other services without having to handle physical cash or cheques. Customers do not have to

wait in lines; this provides a lower-hassle environment. Debit cards and online bill payments allow immediate transfer of funds from an individual's personal account to a business's account without any actual paper transfer of money. This offers a great convenience to many people and businesses alike." [29]

Appendix D

System User Guide

User guide contains main 3 parts.

1. Mobile interface
2. SMS capture and sending part
3. Retailer specific sub system.

According to the Figure D-1 user enter details to send money to the receiver.

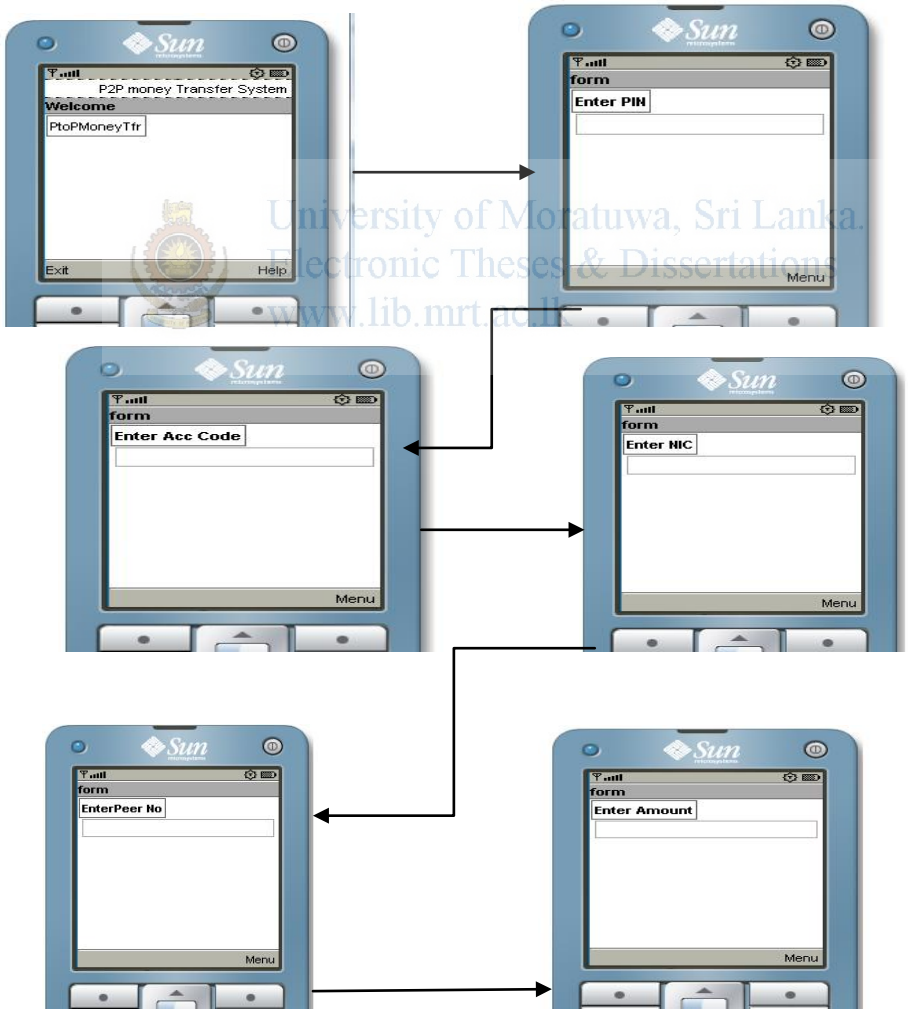


Figure D-1 mobile device user interface.

Modem will connect to the serial port of the computer. This will assign to the virtual port and it should mention in the application. Figure D-2 show how to identify the virtual port which assign to the modem.

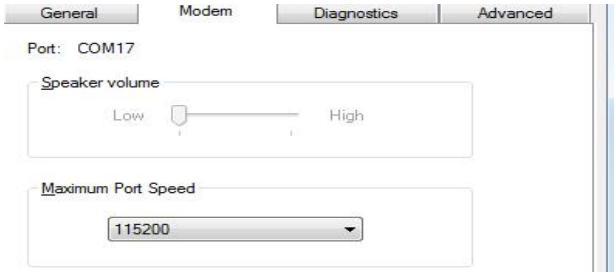


Figure D-2 – Virtual port define window.

JDK 1.5 environment should be configured before using the application. JSMEngine.jar and MYSQLConnector.jar should configure to the class path.

1. Windows environment setting class path.

Right click mycomputer → get system settings → System properties

Set the variable name and value ← System variables ← advanced tab

Then modem application should start using following command

Go to the root directory (d:/AutomatedMoneyTransfer)

java com.sms.RunModem

```
Connectio ok To read message3 0
** GSM MESSAGE **
  Type: Incoming.
  Id: null
  Memory Index: 0
  Date: Mon Sep 22 22:39:07 IST 2008
  Originator: +94714800123
  Recipient: null
  Text: 2222 10 785678678v 0714812108 1000
  Hex Text: 3232323220313020373835363738363738762030373134383132313038203130303030
  Encoding: 1
```

Figure D-3 Modem receiving message

After receiving the message it should send to the bank. It will send using socket client application and will run using,

java com.mo.conectivity. MobileOperatorClient

```
Client /127.0.0.1 is connected...***
inside get output streamtrue
Client messageNB001|000029|0714800123|2008-09-22|2222|10|785678678v|0714812108|1000|
OK
Read Finished
```

Figure D-4 Socket client send message

Receiving message from the bank using server application

java com.mo.conectivity. MobileOperatorServer

```
Connected to Server localhost:4444
Inside if sender - NB001|000029|0714800123|2008-09-22|2222|10|785678678v|0714812108|1000|
Reply from the bank BN002|000029|0714800123|0714812108|79872008-09-22|1000|Amount successfully transfer to the receiver. Auth key is 79872008-09-22|
```


Figure D-5 Socket server receive message.

RunModem application used to send reply message to customer.

(ReadMessageNew.java)

```
Inside send Message.
connection ok.0
Inside  While peer send
Reply Message Failed!
Inside  While peer send
Reply Message Sent!0714800123
Inside  While peer send
Reply Message Sent!0714812108
```

Figure D-6 – Modem send message

2. Web interface for bank executive

Bank executive login to the system and validate customer. Web address is http://localhost:8080/AutomatedMoneyTransfer/BELogin.jsp

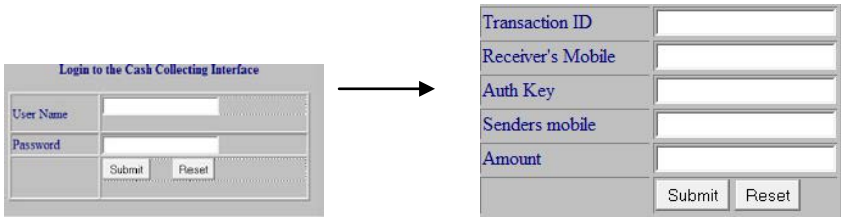


Figure D-7 Web interface for bank executive

3. Retailer user guide

Retailer is login to the system to check customer validity. Web address is 'http://localhost:8080/ AutomatedMoneyTransfer/ Retailer/RetailerLogin.jsp'
See Figure D-8

Figure D-8 – Login interface for retailer

After login to the system it will redirect to the menu screen See Figure D-9

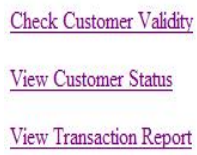


Figure D-9 – Main menu – retailer.

Retailer should click on ‘check customer validity’ link. It will come to the below mentioned interface.

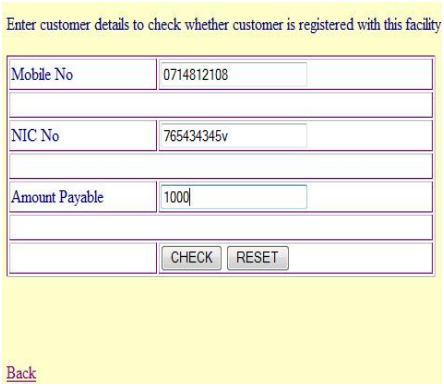


Figure D-10 – interface for check customer validity.

After submitting data, result will display as whether the customer is registered or not. Same time customer will receive the message and ask confirmation to do the transaction. Customer’s reply SMS will receive by the modem application.

All customer transactions and their status can be seen in the retailer web interface.