

SOCKET PROGRAMMING

SERVER SIDE

```
import java.io.*;
import java.net.*;
public class MyServer {
public static void main(String[] args){
try{
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes connection
DataInputStream dis=new DataInputStream(s.getInputStream());
String str=(String)dis.readUTF();
System.out.println("message= "+str);
ss.close();
}catch(Exception e){System.out.println(e);}
}
}
```

CLIENT SIDE

```
import java.util.*;
import java.io.*;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
String str;
Scanner sc=new Scanner(System.in);
System.out.println("Enter your message");
str=sc.next();
dout.writeUTF(str);
dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}
```

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\visitor>cd desktop
C:\Users\visitor\Desktop>set path=C:\Program Files\Java\jdk1.6.0\bin
C:\Users\visitor\Desktop>javac MyServer.java
C:\Users\visitor\Desktop>java MyServer
message= hello
C:\Users\visitor\Desktop>
```

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\visitor>cd desktop
C:\Users\visitor\Desktop>set path=C:\Program Files\Java\jdk1.6.0\bin
C:\Users\visitor\Desktop>javac MyClient.java
C:\Users\visitor\Desktop>java MyClient
Enter your message
hello
C:\Users\visitor\Desktop>
```

CLIENT-SERVER AUTHENTICATION

CLIENT PROGRAM

```
import java.io.*;
import java.net.*;
import java.util.*;
public class client {
    public static void main(String[] args){
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept();
            DataInputStream dis=new DataInputStream(s.getInputStream());
            int otp=dis.readInt();
            System.out.println("Put OTP and press enter!!!");
            int validation;
            System.out.println("OTP = ");
            Scanner sc = new Scanner(System.in);
            validation= sc.nextInt();
            if(otp==validation)
            {
                System.out.println("Authenticated");
            }
            else{
                System.out.println("Unauthorised");
            }
            ss.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

SERVER PROGRAM

```
import java.io.*;
import java.net.*;
import java.util.Random;
public class server{
    public static void main(String[] args) {
        try{
            Socket s=new Socket("localhost",6666);
            Random rand = new Random();
            int n = rand.nextInt(50) + 1;
            int otp=n;
            System.out.print(otp);
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            dout.writeInt(otp);
            dout.flush();
        }
    }
}
```

```

dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}

```

OUTPUT

Authorized:

The image shows two separate Command Prompt windows running on Microsoft Windows 6.3.9600. Both windows have identical command histories:

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\benny.sebin>set path="C:\Program Files\Java\jdk1.6.0\bin"
C:\Users\benny.sebin>cd Downloads
C:\Users\benny.sebin\Downloads>javac client.java
C:\Users\benny.sebin\Downloads>java client
Put OIP and press enter!!
OTP =
40
Authenticated
C:\Users\benny.sebin\Downloads>

```

Unauthorized:

The image shows two separate Command Prompt windows running on Microsoft Windows 6.3.9600. Both windows have identical command histories, with one additional line of output in the second window:

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\benny.sebin>set path="C:\Program Files\Java\jdk1.6.0\bin"
C:\Users\benny.sebin>cd Downloads
C:\Users\benny.sebin\Downloads>javac client.java
C:\Users\benny.sebin\Downloads>java client
Put OIP and press enter!!
OTP =
40
Authenticated
C:\Users\benny.sebin\Downloads>java client
Put OIP and press enter!!
OTP =
12
Unauthorised
C:\Users\benny.sebin\Downloads>

```

IMPLEMENTATION OF ENCRYPTION

Server:

```
import java.io.*;
import java.net.*;
public class Server {
    public static void main(String[] args) {
        try {
            System.out.println("Message :");
            String str="";
            int ascii;
            ServerSocket ss = new ServerSocket(1997);
            StringBuilder sb = new StringBuilder();
            Socket s= ss.accept();
            DataInputStream dis = new
            DataInputStream(s.getInputStream());
                while(str!="exit") {
                    str =
                    (String)dis.readUTF();
                    System.out.println("Message
Received="+str);
                    String
encrpt=dis.readUTF();

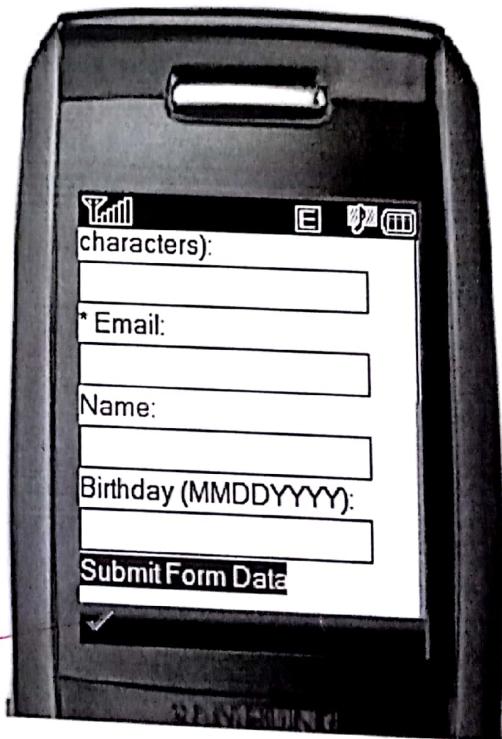
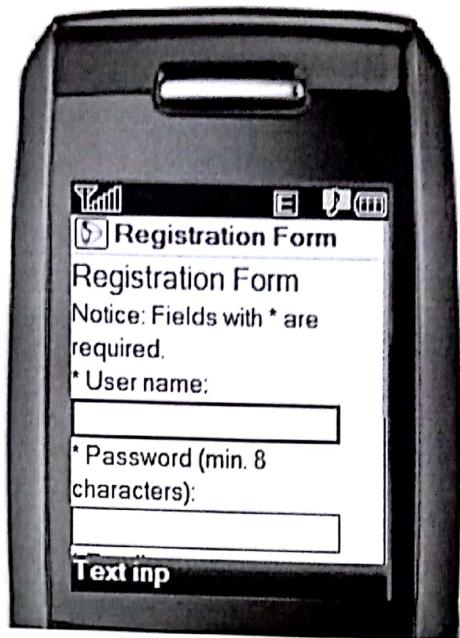
                    System.out.println("Encrypted
Received="+encrpt);
                    for(int
i=0;i<encrpt.length();i++){
                        char characters =
encrpt.charAt(i);
                        if(characters=='*')
                            {sb.append('a')};
                        else if
                            (characters=='#')
                            {sb.append('e')};
                        else if
                            (characters=='(')
                            {sb.append('i')};
                        else if
                            (characters==')')
                            {sb.append('o')};
                        else if
                            (characters=='@')
                            {sb.append('u')};
                        else
                            {ascii =(int)characters -
1;
                            sb.append((char)ascii)};
                    }
                    System.out.println("Decrypted
message:"+sb.toString());
                    ss.close();
                    System.out.println("received exit command.
Exiting....");
                }catch(Exception e) {}
        }
    }
}
```

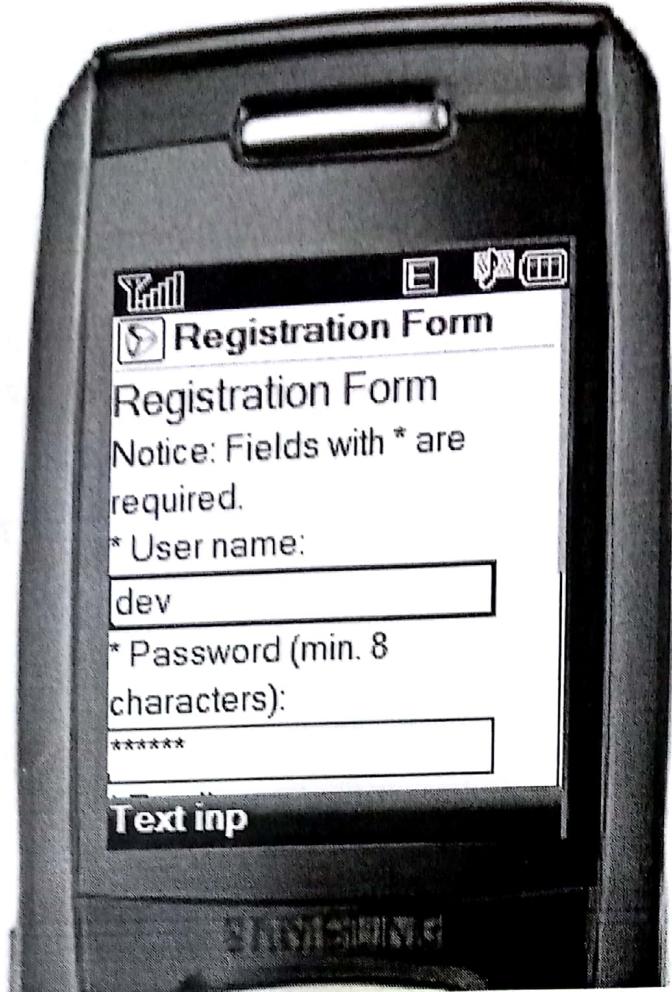
Client:

```
import java.net.*;
import java.util.*;
import java.io.*;
public class Client {
    public static void main(String args[]) {
        try {
            int opt =0;
            String str= "";
            int ascii;
            StringBuilder sb = new
            StringBuilder();
            Socket s = new
            Socket("localhost",1997);
            Scanner sc = new
            Scanner(System.in);
            DataOutputStream dout = new
            DataOutputStream(s.getOutputStream());
            while(opt!=2) {
                System.out.print("Enter
your option.\n1.Send message\n2.Exit\nYour
option:");
                opt= sc.nextInt();
            }
        }
    }
}
```

CREATION OF STUDENT REGISTRATION FORM USING WML

```
<?xml version="1.0"?>
<wml>
  <card id="card1" title="Registration
Form">
    <p>
      <big>Registration Form</big><br/>
      Notice: Fields with * are
      required.<br/><br/>
      <b>${errorMsg}</b><br/>
      * User name:<br/>
      <input name="username"/><br/>
      * Password (min. 8 characters):<br/>
      <input type="password"
name="password"/><br/>
      * Email:<br/>
      <input name="email"/><br/>
      Name:<br/>
      <input name="name"/><br/>
      Birthday (MMDDYYYY):<br/>
      <input name="birthday"
format="NNNNNNNN"
emptyok="true"/><br/><br/>  </p>
    </card>
  </wml>
```





Deck and Card

Source Code:

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>
<card title="Formatting" id="card01">
<p align="left">
Enter text<br/>
<input name="text" emptyok="false"
type="text" id="text01"/>
<anchor>
<go href="#card02"/>
Change the font
</anchor>
<br/>
<anchor>
<go href="#card03"/>
Change the alignment
</anchor>
<br/>
<anchor>
<go href="#card04"/>
Change the size
</anchor>
</p>
</card>

<card title="Font" id="card02">
<p align="center">
Select a font style:
<select title="fontstyles"
name="selection_list">
<option onpick="#Bold">Bold</option>
<option onpick="#Italic">Italic</option>
<option
onpick="#Underline">Underline</option>
</select>
</p>
</card>
<card id="Bold" title="Bold">
<p>
<b>${text}</b>
</p>

```

↙

```

</card>
<card id="Italic">
<p>
<i>${text}</i>
</p>
</card>
<card id="Underline">
<p>
<u>${text}</u>
</p>
</card>

<card title="Align" id="card03">
<p align="center">
Select an alignment
<select title="alignstyles"
name="selection_list">
<option onpick="#Center">Center</option>
<option onpick="#Left">Left</option>
<option onpick="#Right">Right</option>
</select>
</p>
</card>
<card id="Center">
<p align="center">
${text}
</p>
</card>
<card id="Left">
<p align="left">
${text}
</p>
</card>
<card id="Right">
<p align="right">
${text}
</p>
</card>

<card title="Size" id="card04">
<p align="center">
Select a size
<select title="fontsize" name="selection_list">
<option onpick="#Big">Big</option>

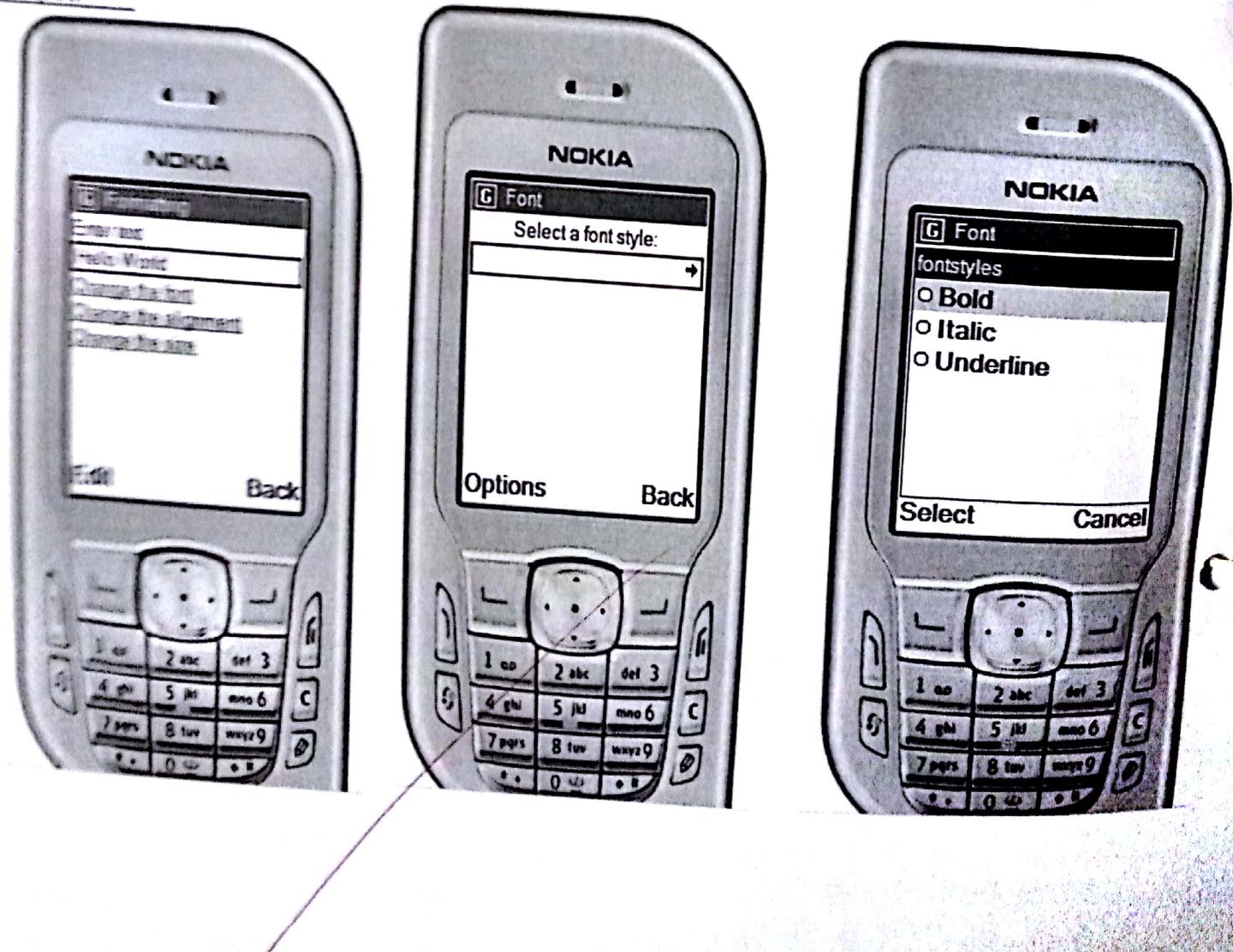
```

```

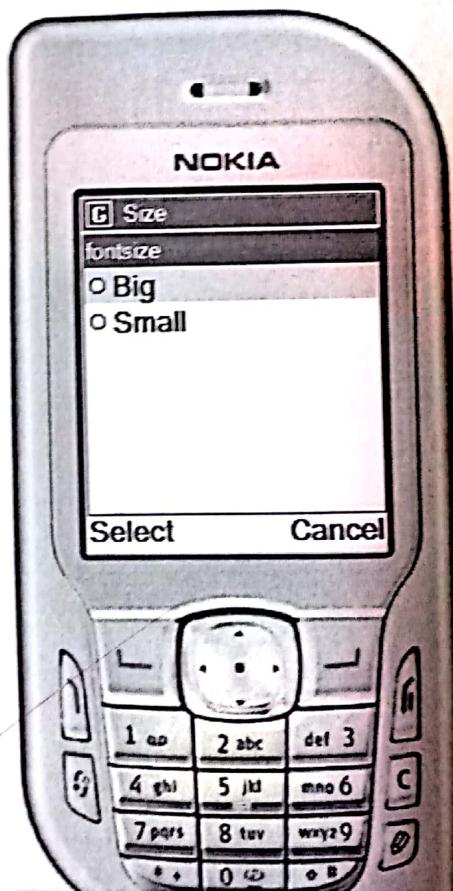
<option onpick="#Small">Small</option>
</select>
</p>
</card>
<card id="Big">
<p><big>
$(text)
</big>
</p>
</card>
<card id="Small">
<p><small>
$(text)
</small>
</p>
</card>
</wml>

```

Output:







Experiment:Implementation of anchor,images,tables in wml

Experiment :

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
    <card id ="card1" title="Image in wml">
        <p>
            <br/>
            marksheet<br/><br/>
            <em>      FCRIT,vashi    </em>
        <anchor>
            <go href="#card2"/>
            images
        </anchor>

        <table columns="4">
            <tr>
                <td><b>SUB</b></td>
                <td><b>Term/Pracs</b></td>
                <td><b>End sem</b></td>
                <td><b>Total</b></td>
            </tr>
            <tr>
                <td>PHY</td>
                <td>19</td>
                <td>79</td>
                <td>98</td>
            </tr>
            <tr>
                <td>CHEM</td>
                <td>18</td>
                <td>80</td>
                <td>98</td>
            </tr>
            <tr>
                <td>MATH</td>
                <td>10</td>
                <td>60</td>
                <td>70</td>
            </tr>
            <tr>

```

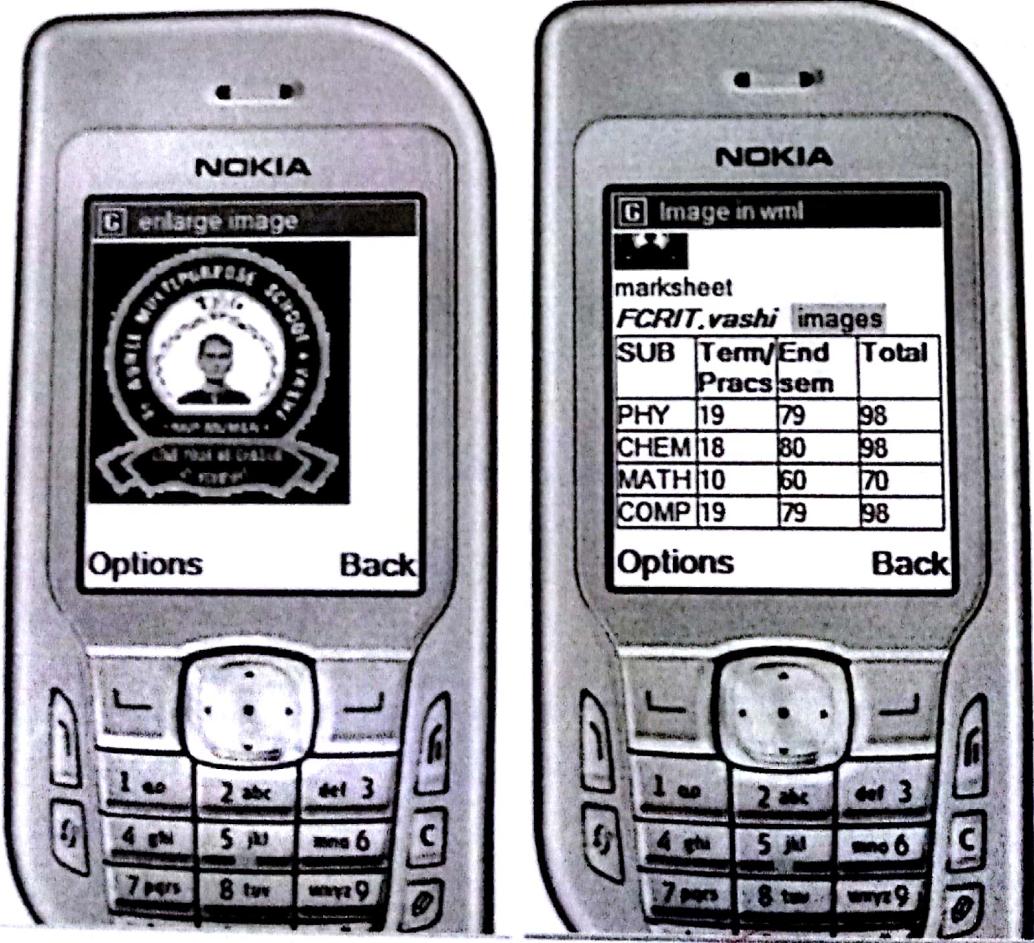
```

        <td>COMP</td>
        <td>19</td>
        <td>79</td>
        <td>98</td>
    </tr>

</table>
</p>
</card>
<card id ="card2" title="enlarge image">
<p>
    <br/>
</p>
</card>
</wml>

```

OUTPUT:



Mobile Menu

Source Code:

```

?xml version="1.0"?
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

MAIN MENU :
<card id="menu" title="Phone Menu">
<p>
<a href="#contact"> </a>
<a href="#google"> </a>
<a href="#msg"></a>
</p>
<p>
<p>
<a href="#cal"> </a>
<a href="#radio"> </a>
<a href="#music"></a>
</p></p>
</card>

```

CONTACT:

```

<card id="contact" title="Contact">
<p>
<br/><br/>
<a href="#add">Add Contact</a>
<br/><br/>Contacts:
<br/><br/><br/>
<a href="#1">Aaron</a><br/>
<a href="#2">Abhiyah</a><br/>
<a href="#3">Benetta</a><br/>
<a href="#4">Feba</a><br/>
<a href="#5">Hanna</a><br/>
<br/><br/>

```

```

<a href="#menu">Back To Menu</a>
</p>
</card>

<card id="1" title="Contacts">
<p align="center">
<br/>
Name:- <b>Aaron</b> <br/>
Phone NO:- <b>9876543210</b><br/>
<a href="#call">Call</a>
</p>
<p align="right">
<a href="#menu">Back</a>
</p>
</card>

<card id="2" title="Contacts">
<p align="center">
<br/>
Name:- <b>Abhiyah</b> <br/>
Phone NO:- <b>9876543210</b><br/>
<a href="#call">Call</a></p>
<p align="right">
<a href="#menu">Back</a>
</p>
</card>

<card id="3" title="Contacts">
<p align="center">
<br/>
Name:- <b>Benetta</b> <br/>
Phone NO:- <b>9876543210</b><br/>
<a href="#call">Call</a>
</p>
<p align="right">
<a href="#menu">Back</a>
</p>
</card>

<card id="4" title="Contacts">
<p align="center">

```

```

<br/>
Name:- <b>Feba</b> <br/>
Phone NO:- <b>9876543210</b><br/>
<a href="#call">Call</a>
</p>
<p align= "right">
<a href="#menu">Back</a>
</p>
</card>

<card id="5" title="Contacts">
<p align= "center">
<br/>
Name:- <b>Hanna</b> <br/>
Phone NO:- <b>9876543210</b><br/>
<a href="#call">Call</a>
</p>
<p align= "right">
<a href="#menu">Back</a>
</p>
</card>

<card id="add" title="Add Contact">
<p align= "center">
<big>ADD Contact</big>
<br/><br/>
<b>Name:</b><br/>
<input name="name"/><br/>
<b>Phone no:</b>
<br/>
<input name="number"
format="NNNNNNNNNN"/><br/><br/>
<a href="#save">Save Contact</a>
</p>
<p align= "right">
<a href="#menu">Back</a>
</p>
</card>

<card id="save" title="Contact Added">
<p align="center">
<br/><br/>
<big><b>Contact Saved Successfully</b></big>
</p>
<br/>

```

```

<p align= "right">
<a href="#contact">Back</a>
</p>
</card>

<card id="call" title="Calling">
<p>

</p>
<p align= "center">
<a href="#menu">End Call</a>
</p>
</card>

BROWSER :
<card id="google" title="Google Chrome">
<p>

</p>
<p align= "center">
<a href="#menu">Back</a>
</p>
</card>

MESSAGING :
<card id="msg" title="Messaging">
<p>
<big><b>New Message</b></big>
<br/>
<b>To:</b><br/>
<input name="number"
format="NNNNNNNNNN"/><br/><br/>
<b>Message:</b>
<input name="name"/><br/>
</p>
<p align= "left">
<a href="#send">Send</a>
</p>
<br/>
<p align= "center">
<a href="#menu">Back</a>
</p>
</card>

```

```

<card id="send" title="Sending">
<p>

</p>
<p align="center">
<a href="#menu">Back</a>
</p>
</card>

```

CALENDAR:

```

<card id="cal" title="Calendar">
<p>

</p>
<p align="center">
<a href="#menu">Back</a>
</p>
</card>

```

RADIO:

Output:

CONTACT:

```

<card id="radio" title="Radio">
<p>

</p>
<p align="center">
<a href="#menu">Back</a>
</p>
</card>

```

MUSIC:

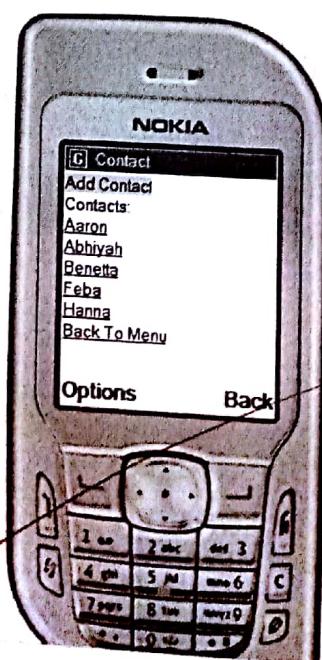
```

<card id="music" title="Music">
<p>

</p>
<p align="center">
<a href="#menu">Back</a>
</p>
</card>

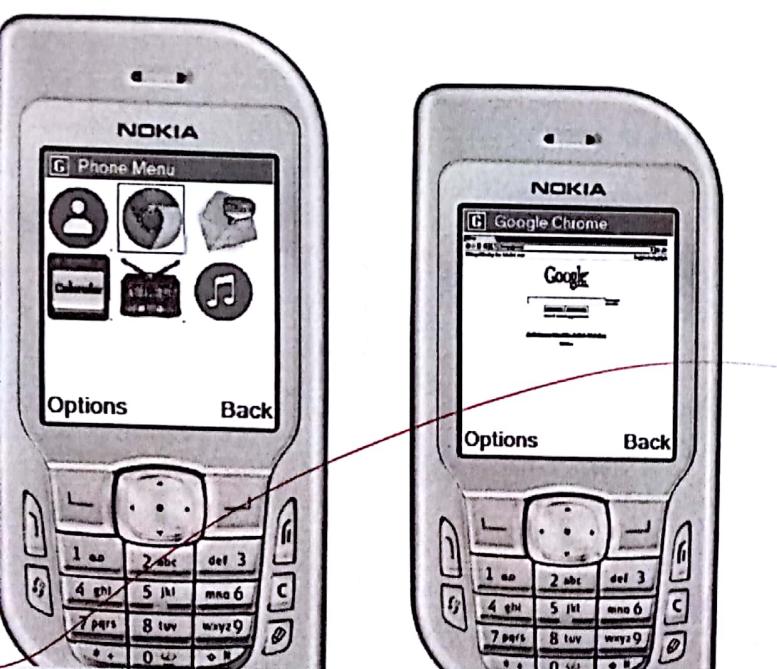
```

</wml>



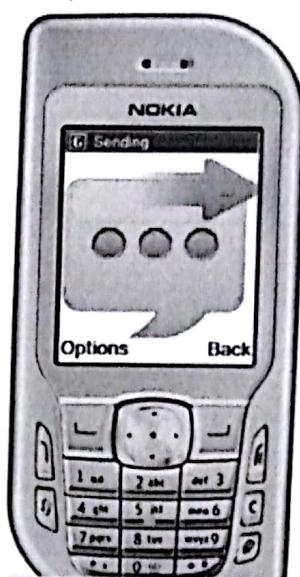
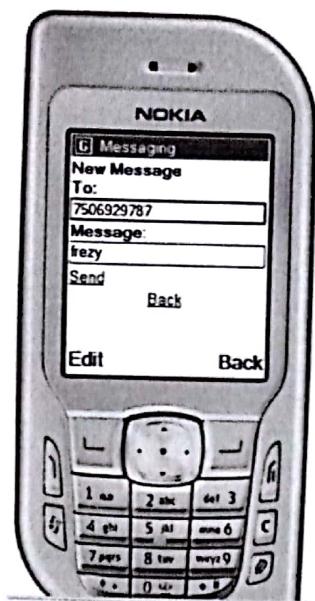


BROWSER:

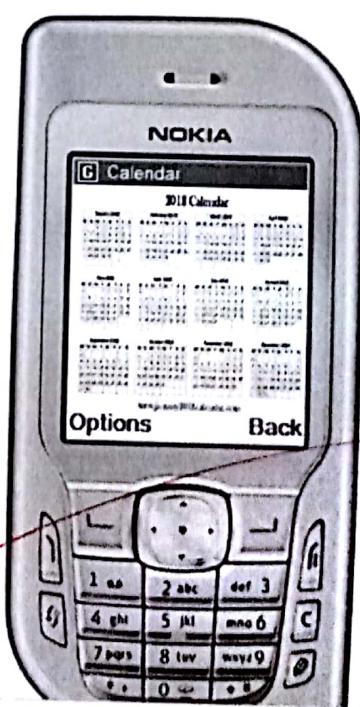


61

MESSAGING:



CALENDAR:



RADIO:



MUSIC:



PROGRAM FOR MOBILE NETWORKS IN NS2

Program : Infrastructure

```

set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#n1,n3 AP

$ns duplex-link Sn0 Sn1 1Mb 10ms DropTail
$ns duplex-link Sn1 Sn2 1Mb 10ms DropTail
$ns duplex-link Sn1 Sn3 1Mb 10ms DropTail

$ns duplex-link-op Sn0 Sn1 orient right
$ns duplex-link-op Sn1 Sn2 orient right-up
$ns duplex-link-op Sn1 Sn3 orient right-down

set udp0 [new Agent/UDP]
$ns attach-agent Sn0 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbro set packetSize_ 500
$cbro set interval_ 0.005
$cbro attach-agent $udp0

set udp1 [new Agent/UDP]
$ns attach-agent $n3 $udp1

set cbr1 [new Application/Traffic/CBR]
$cbri set packetSize_ 500
$cbri set interval_ 0.005
$cbri attach-agent $udp1

set null0 [new Agent/Null]
$ns attach-agent Sn2 $null0
set null1 [new Agent/Null]
$ns attach-agent $n0 $null1

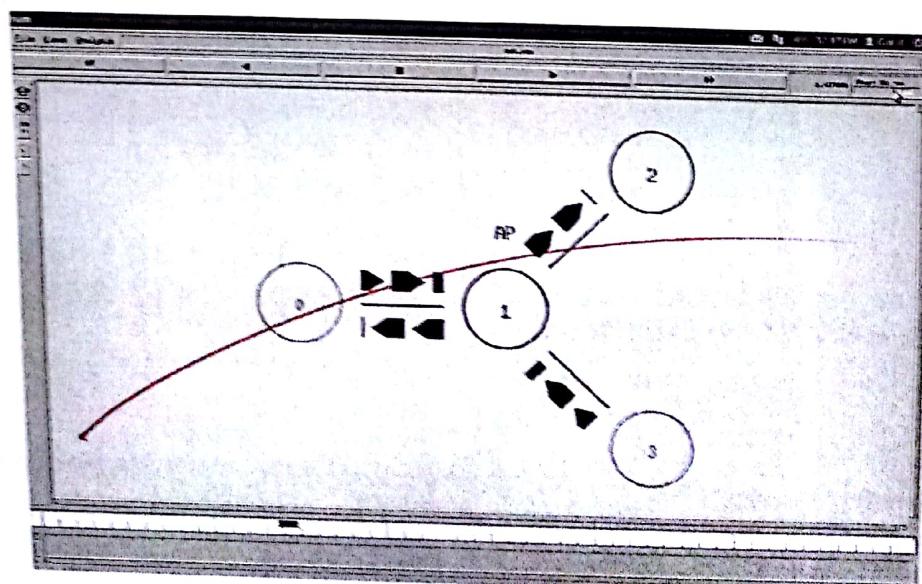
$udp0 set class_ 1
$udp1 set class_ 2

$ns color 1 Blue
$ns color 2 Red

$ns connect $udp0 $null0
$ns connect $udp1 $null1
$ns at 0.0 "$n1 label AP"
$ns at 0.0 "$n0 color green"
$ns at 0.0 "$n2 color magenta"
$ns at 0.0 "$n3 color green"
$ns at 0.5 "$cbro start"
$ns at 0.9 "$cbri start"
$ns at 4.9 "$cbri stop"
$ns at 4.5 "$cbro stop"
$ns at 5.0 "finish"
$ns run

```

Output:



Program: Ad-hoc

```
set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail/PriQueue ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 50 ;
set val(nn) 5 ;
set val(rp) AODV ;
set val(x) 350 ;
set val(y) 350 ;
set val(stop) 10 ;

set ns [new Simulator]
set tracefd [open thesis.tr w]
set namtrace [open thesis.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x)
$val(y)

set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phType $val(netif) \
-channelType $val(chan) \Output:
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON

$ns node-config -energyModel EnergyModel \
-initialEnergy 20 \
-txPower 0.744 \
-rxPower 0.0648 \
-idlePower 0.05 \
-sensePower 0.0175

for {set i 0} {$i < $val(nn)} {incr i} {
    set n($i) [$ns node]
}

$n(0) set initialEnergy 5
$n(1) set initialEnergy 25
$n(2) set initialEnergy 15

$n(0) set X_ 100.0
$n(0) set Y_ 200.0
$n(0) set Z_ 0.0

$n(1) set X_ 200.0
$n(1) set Y_ 150.0
$n(1) set Z_ 0.0

$n(2) set X_ 100.0
$n(2) set Y_ 50.0
$n(2) set Z_ 0.0

$n(3) set X_ 300.0
$n(3) set Y_ 200.0
$n(3) set Z_ 0.0

$n(4) set X_ 300.0
$n(4) set Y_ 50.0
$n(4) set Z_ 0.0

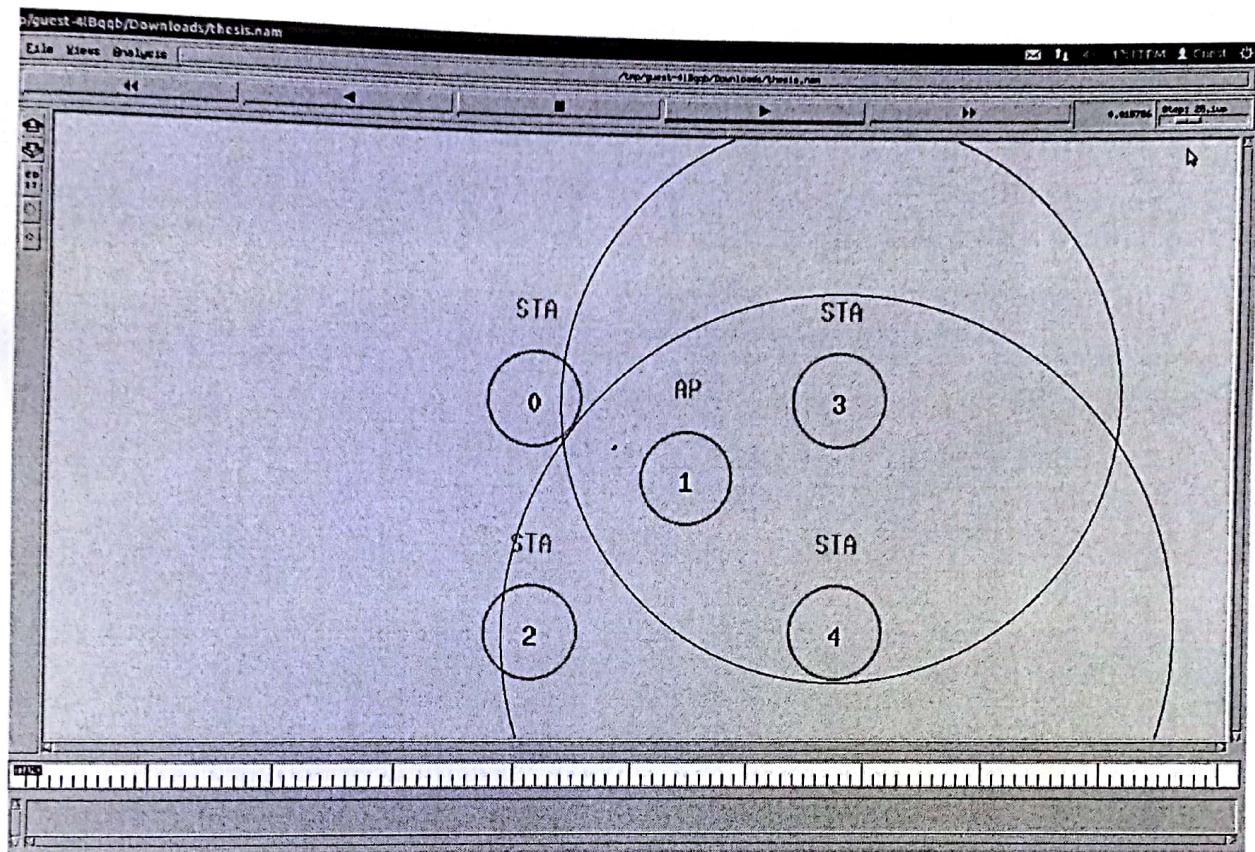
for {set i 0} {$i < 5} {incr i} {
    set udp($i) [new Agent/TCP/Newreno]
    set sink($i) [new Agent/TCPSink]
    $ns attach-agent $n($i) $udp($i)
    $ns attach-agent $n($i) $sink($i)
    $ns connect $udp($i) $sink($i)
    set cbr($i) [new Application/Traffic/CBR]
    $cbr($i) set packetSize_ 500
    $cbr($i) set interval_ 0.005
    $cbr($i) attach-agent $udp($i)
    $ns at $i "$cbr($i) start"
}

for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $n($i) 60
    $ns at 0.0 "$n($i) label STA"
    $ns at $val(stop) "$n($i) reset";
}
$ns at 0.0 "$n(1) label AP"
$ns at $val(stop) "$ns nam-end-wireless
$val(stop)"
```

6
\$ns at \$val(stop) "stop"
\$ns at 10.01 "puts \"end simulation\" ; \$ns halt"
proc stop {} {
global ns tracefd namtrace
\$ns flush-trace
close \$tracefd

close \$namtrace
exec nam thesis.nam &
exit 0
}
\$ns run

Output:



PROGRAM TO CALCULATE TIME REQUIRED FOR RECIEVER TO REPLY IN NS2

Program: Round-Trip-Time

```

set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
Output:
Agent/Ping instproc recv {from rtt} {
    $self instvar node_
    puts "node [$node_id] received ping

```

answer from \
\$from with round-trip-time \$rtt ms."
}

```

set p0 [new Agent/Ping]
$ns attach-agent $n0 $p0
set p1 [new Agent/Ping]
$ns attach-agent $n2 $p1
$ns connect $p0 $p1
$ns at 0.2 "$p0 send"
$ns at 0.4 "$p1 send"
$ns at 0.6 "$p0 send"
$ns at 0.6 "$p1 send"
$ns at 1.0 "finish"

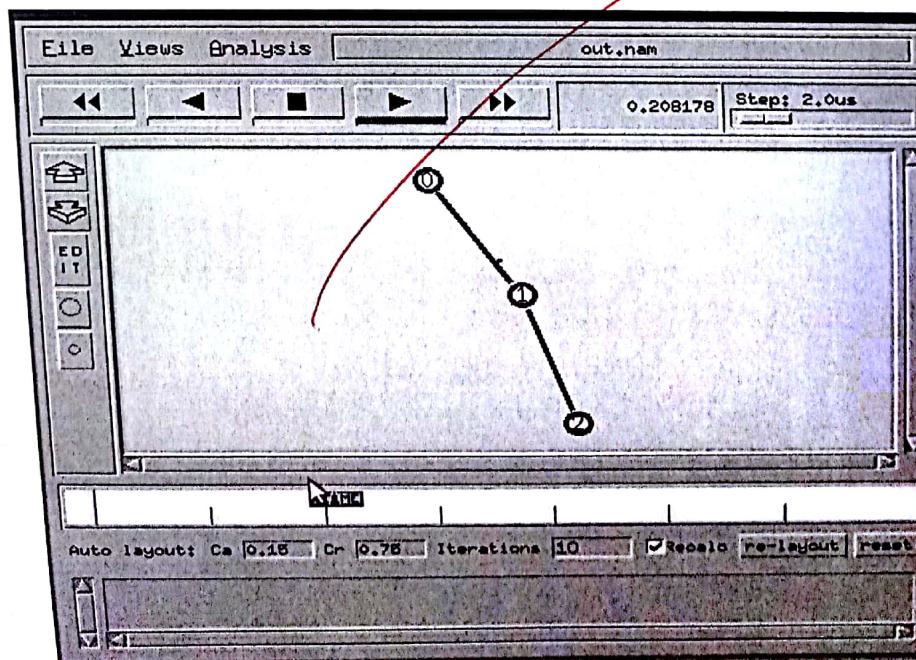
```

\$ns run

Output:

In Terminal:

node 0 received ping answer from 2 with round-trip-time 42.0 ms.
 node 2 received ping answer from 0 with round-trip-time 42.0 ms.
 node 0 received ping answer from 2 with round-trip-time 42.0 ms.
 node 2 received ping answer from 0 with round-trip-time 42.0 ms.



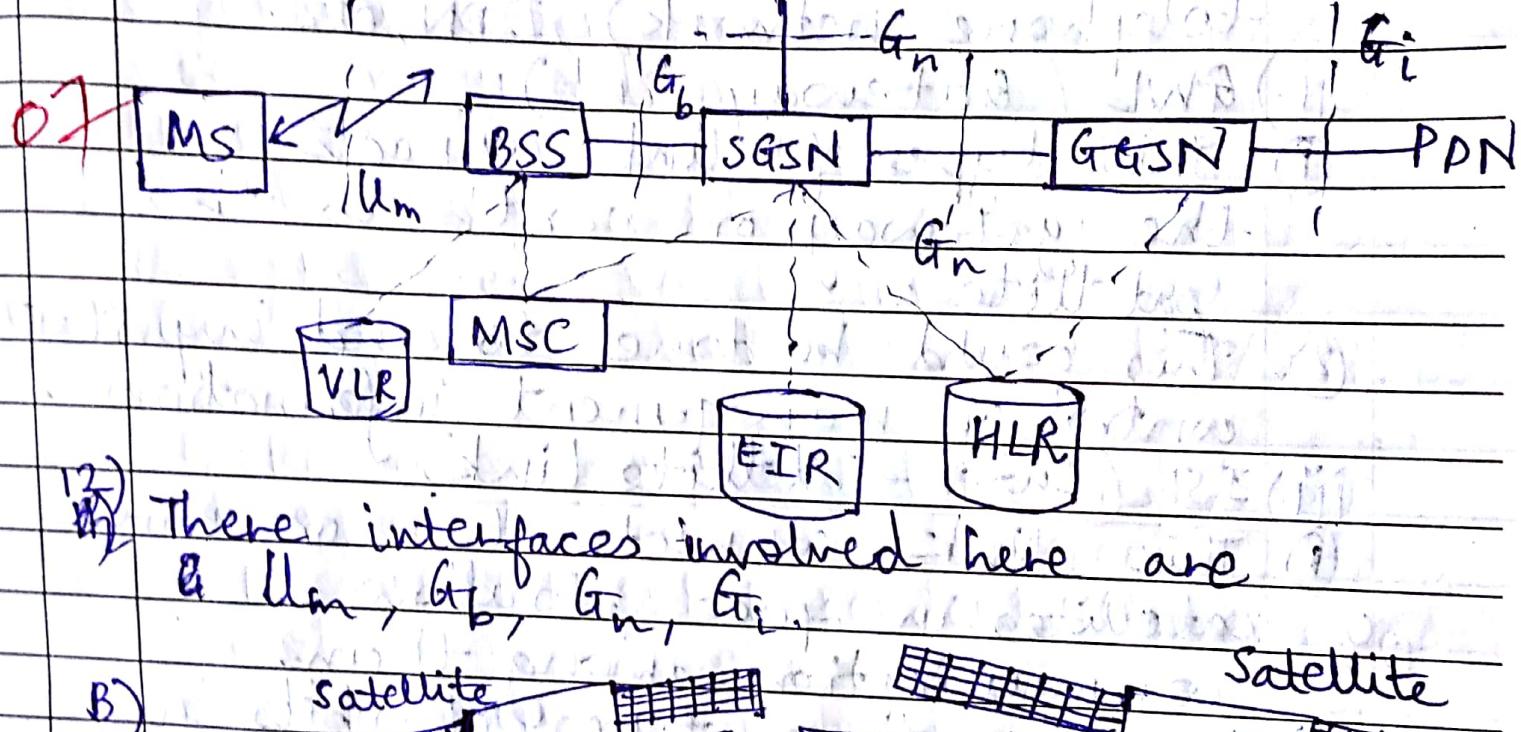
Q2. $Q.2 = 07 + 08 = \boxed{15}$

A)

- 1) GSM consists of various components -
 - BTS (Base Transceiver Station)
 - BSS (Base Station Subsystem)
 - MSC (Mobile Service Switching Centre)
 - BSC (Base Station Controller)
 - MS (Mobile Station)
- 2) Various registers, such as
 - EIR (Equipment Identity Register)
 - HLR (Home Location Register)
 - VLR (Visitor Location Register)
- 3) GPRS (General Packet Radio Service) enables packet mode communication over air interface.
- 4) In addition to above components, GPRS contains the following -
 - i.) An additional register, GPRS register (GR).
 - ii.) Two additional GPRS support nodes (GSNs) :
 - Gateway GSN (GGSN)
 - Serving GSN (SGSN)
- 5) Gateway GSN acts as a gateway between the GPRS network and the public data network (PDN).
- 6) GGSN provides routing information for GPRS users, performs address

conversion and the tunneling of data through radio channels.

- i) Serving GSN keeps track of current MS location.
- ii) Perform SGSN collects billing information and provides security functions such as access controls.
- iii) SGSN takes user addresses from GR.
- iv) SGSN connects MS to the GPRS network, through Uu interface.
- v) EIR stores IMEI (International Mobile Equipment [SGSN] Identity) of devices



- Q7) There are interfaces involved here are
a) Uu, Gb, Gn, Gi.

