

**Project: Donate Blood Save Life**

Course Code: CSE435

Section: 02

**Submitted To:**

**Dr. Shamim H Ripon**

Professor  
Department of Computer Science & Engineering

**Submitted by:**

**Name: ID:**

Muhammad Farabi Hossain 2016-3-60-051

Istiak Ahmed 2016-3-60-042

Md.Tauhidul islam Bhuiyan 2016-2-60-036

**Introduction:**

Donor seeker is an online platform for all kind of people who need emergency blood. This is not

like traditional blood banks, it is a place of regular blood donor. The blood seeker will be able to

search blood donor by his/her(blood seeker) location, the blood seeker will be able to search by

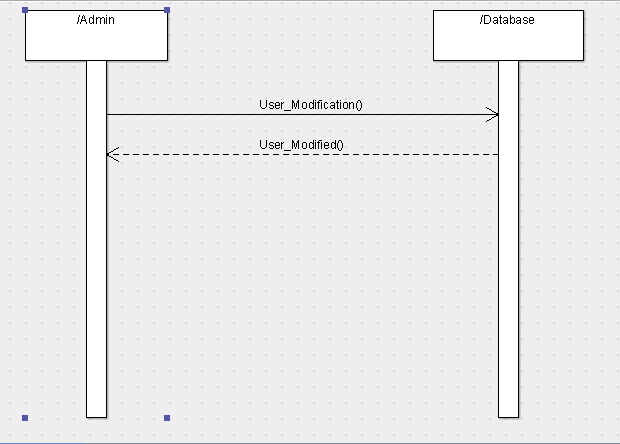
two available options such as he/she can post to forum by his/her certain location, donor will get

notification according to certain location. Another option is blood seeker will see the list of all

donors by blood groups and locations, blood seeker will be able to call or notify one by one from

the list of donors...

**Sequence Diagram(1) for Admin to Database :**

****

**Promela Code(1) :**

mtype {MSG, ACK};

chan toAdmin = [1] of {mtype, bit};

chan toDatabase = [1] of {mtype, bit};

proctype Admin(chan in, out)

{

bit sendbit, recvbit;

do

:: out ! MSG, sendbit ->

in ? ACK, recvbit;

if

:: recvbit == sendbit ->

sendbit = 1-sendbit;

:: else

fi

od

}

proctype Database(chan in, out)

{

bit recvbit;

do

:: in ? MSG(recvbit) ->

out ! ACK(recvbit);

od

}

init

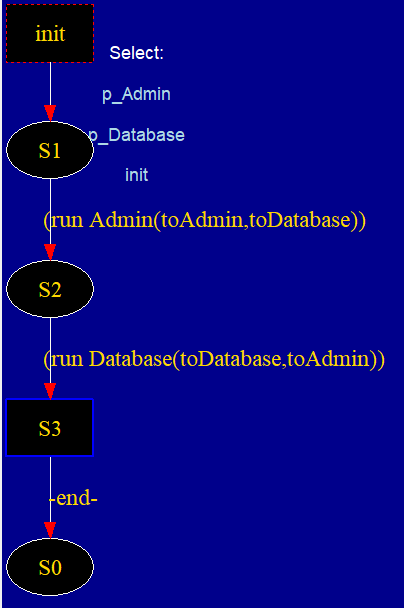
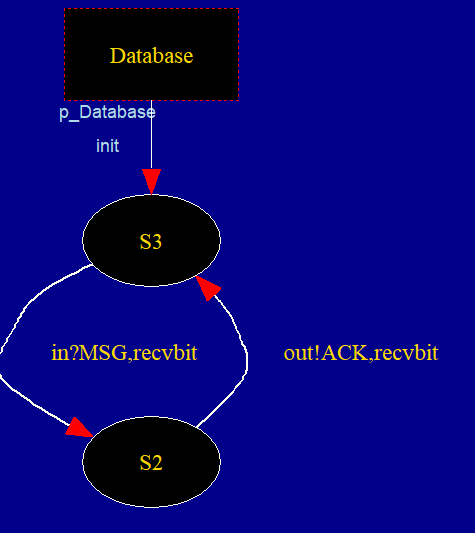
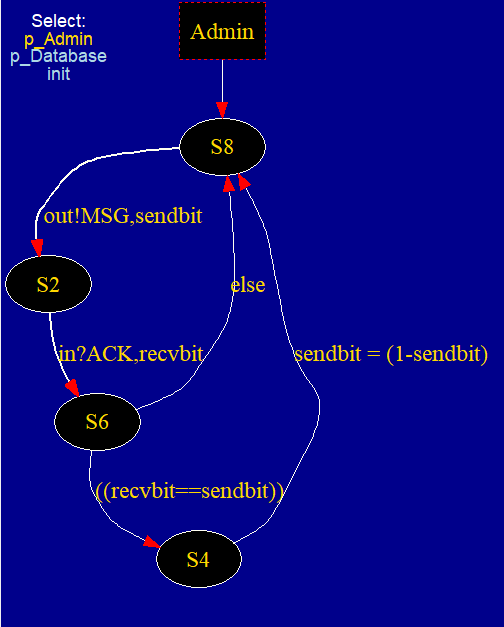
{

run Admin(toAdmin, toDatabase);

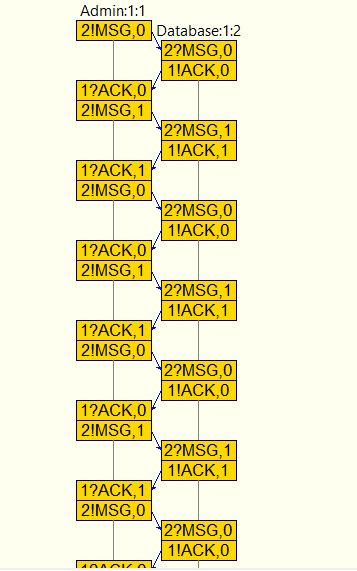
run Database(toDatabase, toAdmin);

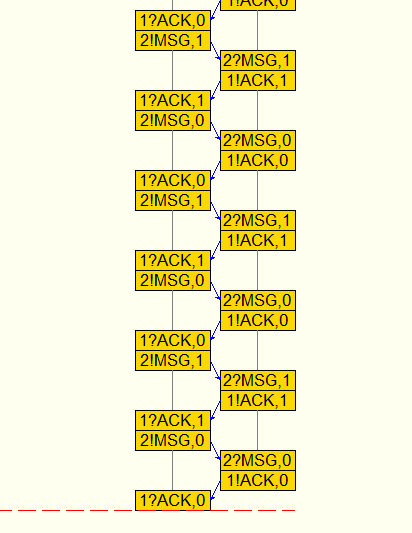
}

**Automata View :**

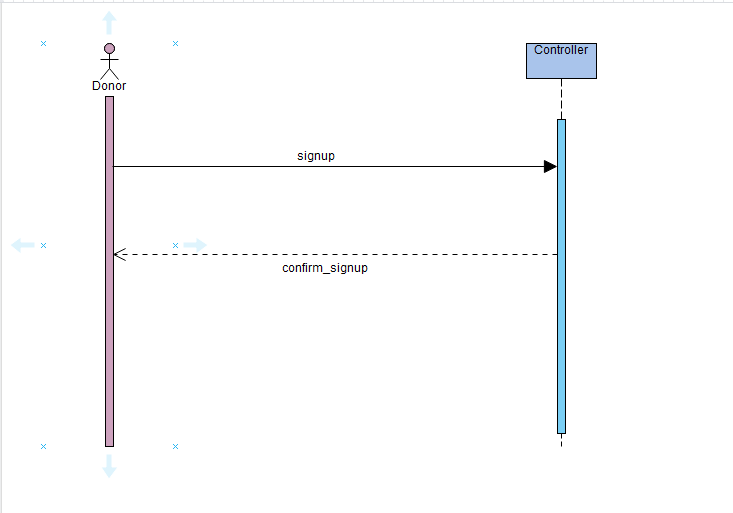
****

**Simulation :**

****

****

**Sequence Diagram(2) for Donor sign up:**

****

**Promela Code(2)**

mtype {MSG, ACK};

chan toDonor = [1] of {mtype, bit};

chan toController = [1] of {mtype, bit};

proctype Donor(chan in, out)

{

bit sendbit, recvbit;

do

:: out ! MSG, sendbit ->

in ? ACK, recvbit;

if

:: recvbit == sendbit ->

sendbit = 1-sendbit;

:: else

fi

od

}

proctype Controller(chan in, out)

{

bit recvbit;

do

:: in ? MSG(recvbit) ->

out ! ACK(recvbit);

od

}

init

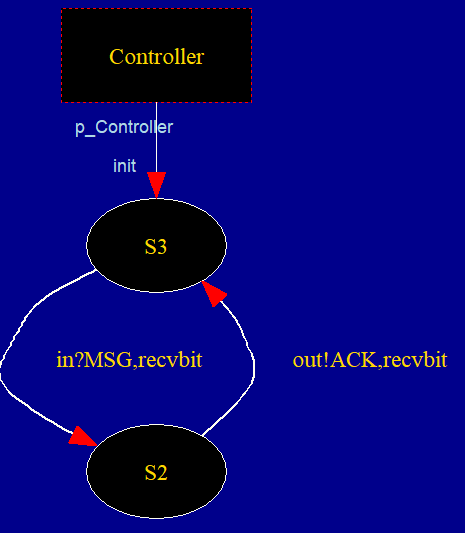
{

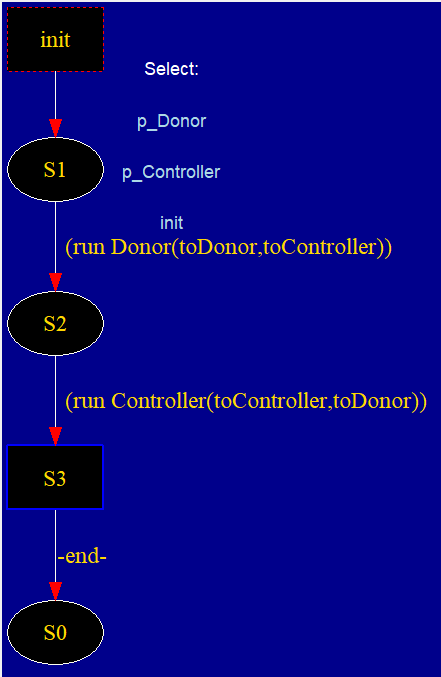
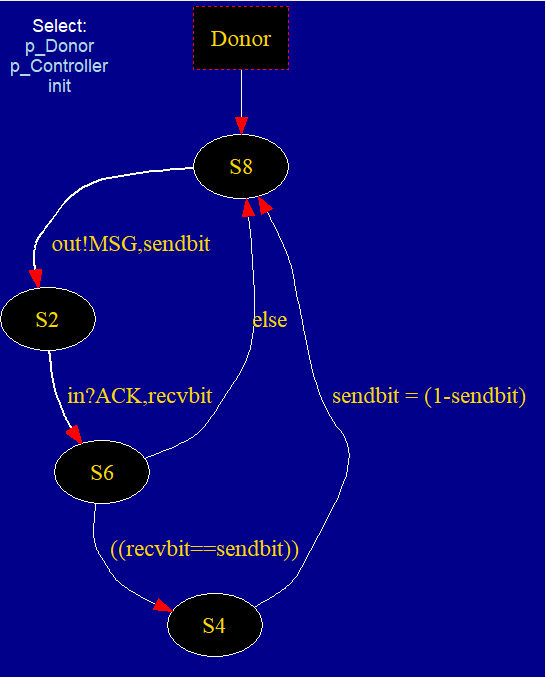
run Donor(toDonor, toController);

run Controller(toController, toDonor);

}

**Automata View :**





**Simulation:**

