EAST WEST UNIVERSITY

Mini Project: Part-2

Topic: Health Emergency

CSE435
Section-1

Submitted by:
Md. Sadat Ahmed
ID:2017-2-60-062

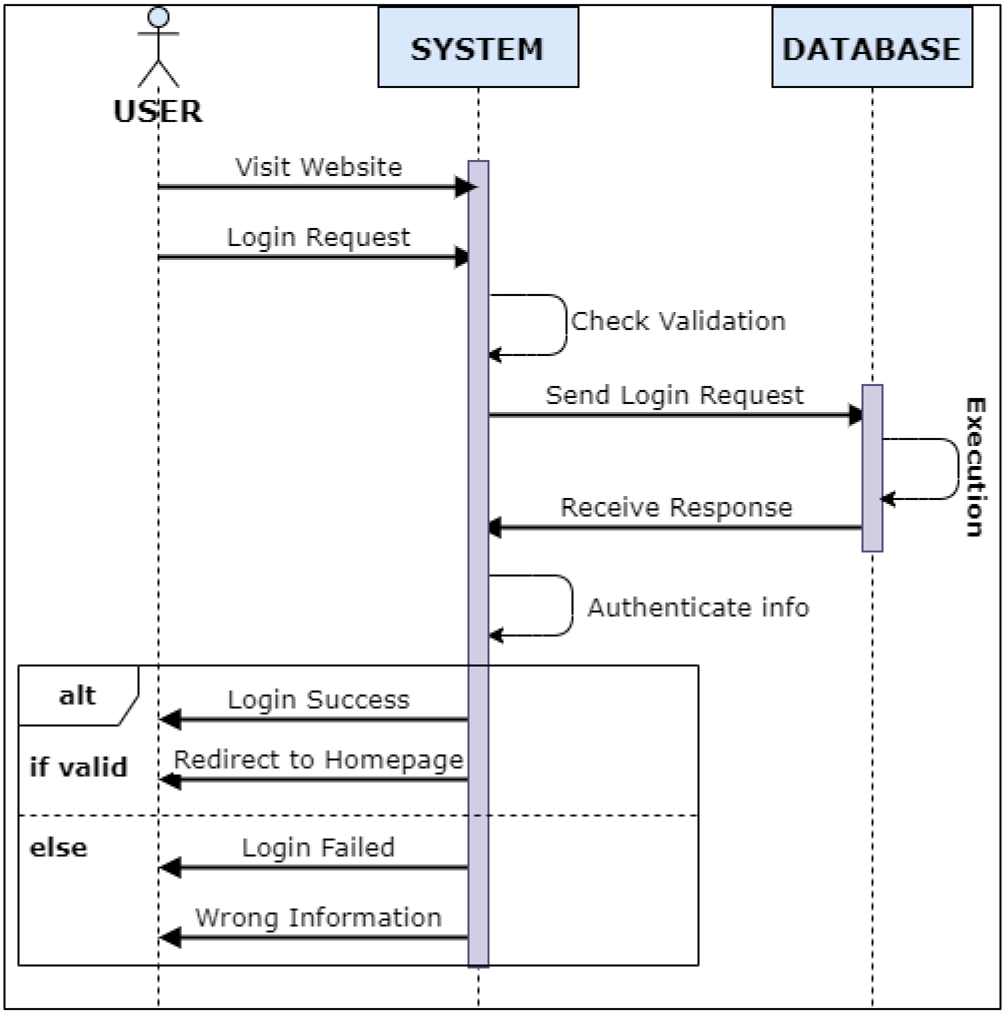
Nahida Sultana
ID: 2017-2-60-139

SUBMITTED TO
DR. SHAMIM H RIPON
Professor
Department of Computer Science & Engineering
East West University

Description: This is a web based software management system which called “Health Emergency”. The purpose of the project is to provide best quality health service to the user. There is a potential possibility for further development in the health sector and for that it needs to improve the health-care online system where there can have a simple and easy communication way between one people to another people. People can share their necessary health product & user can easily find their necessary equipment. Then user can collect the product & get health services very easily through the system. Here, sequence diagrams of some features are shown.

1st sequence diagram (User login):

This diagram shows that user must need to login before registration. If the information is valid then login will be successful or information is invalid then login will be failed. Then user can request their registration, if the information is valid, registration will be successful & it gives message. If there is any failure to registration user will get return message of failed registration. After registration user's information will be stored in the database. Also user can request their update information; if the information is valid then the update will be successful.

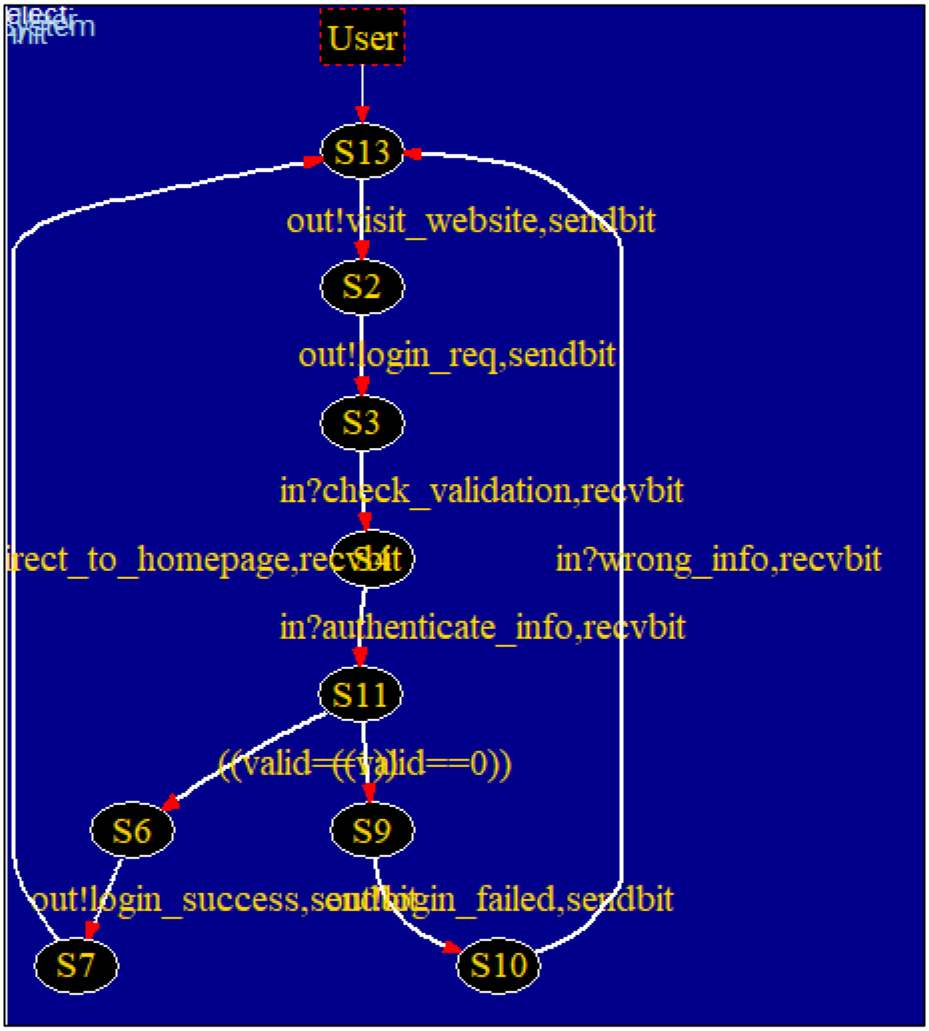


Promela Code for 1st Sequence Diagram:

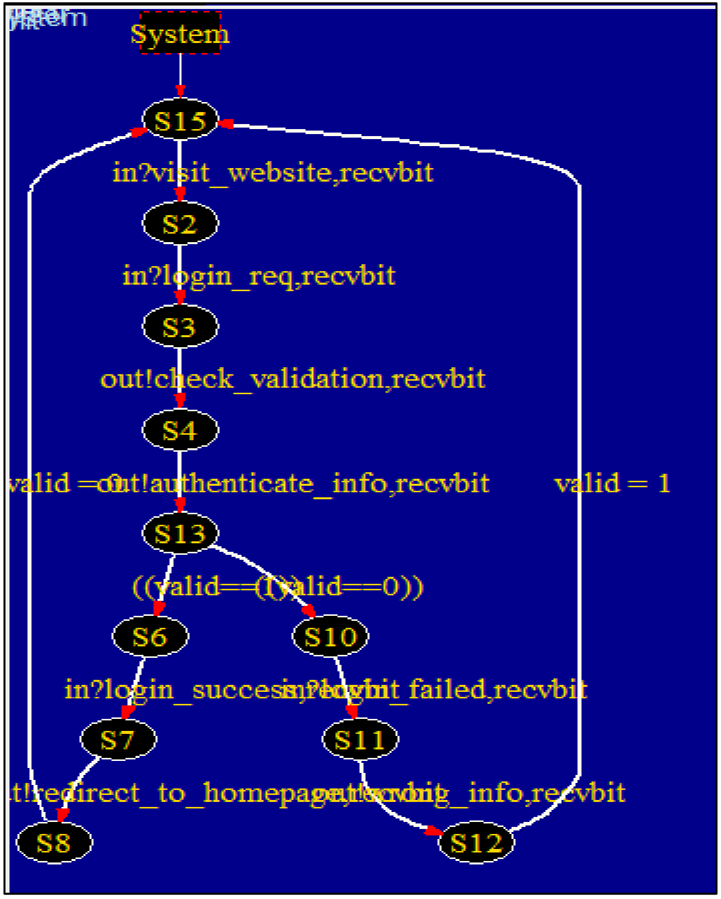
```
mtype = {visit_website, login_req, check_validation, send_login_req,
execution, receive_response, authenticate_info, login_success,
redirect_to_homepage, valid_authentication_info,
invalid_authentication_info, login_failed, wrong_info};
chan toUser = [2] of {mtype,bit};
chan toSystem = [2] of {mtype,bit};
bool valid = 1;
proctype User(chan in, out)
{
    bit sendbit, recvbit;
    do
    ::
        out ! visit_website, sendbit;
        out ! login_req, sendbit;
        in ? check_validation, recvbit;
        in ? authenticate_info, recvbit;
    if
    :: valid == 1 ->
        out ! login_success, sendbit ->
        in ? redirect_to_homepage, recvbit;

    ::  valid == 0 ->
        out ! login_failed, sendbit ->
        in ? wrong_info, recvbit;
    fi
    od
}
proctype System(chan in, out)
{
    bit recvbit;

    do::
    in ? visit_website(recvbit);
    in ? login_req(recvbit);
    out ! check_validation(recvbit);
    out ! authenticate_info(recvbit);
    if
    :: valid == 1 -> in ? login_success(recvbit) -> out !
redirect_to_homepage(recvbit); valid = 0;
    :: valid == 0 -> in ? login_failed(recvbit) -> out !
wrong_info(recvbit); valid = 1;
    fi
    od
}
```

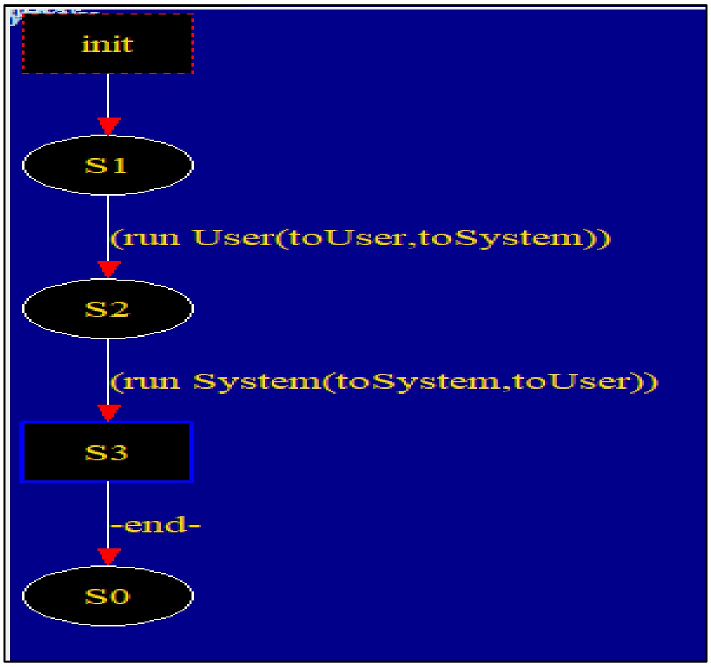
```
init
{
    run User(toUser, toSystem);
    run System(toSystem, toUser);
}
```

Automata for user:

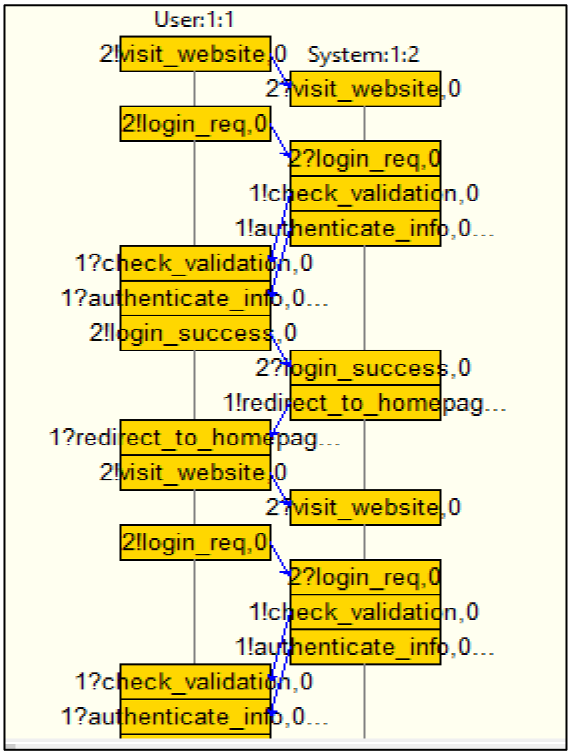

system
init
User
S13
out!visit_website,sendbit
S2
out!login_req,sendbit
S3
in?check_validation,recvbit
S4
irect_to_homepage,recvbit
in?wrong_info,recvbit
in?authenticate_info,recvbit
S11
((valid=
((valid==0))
S6
S9
out!login_success,sendbit
out!login_failed,sendbit
S7
S10

Automata for System:

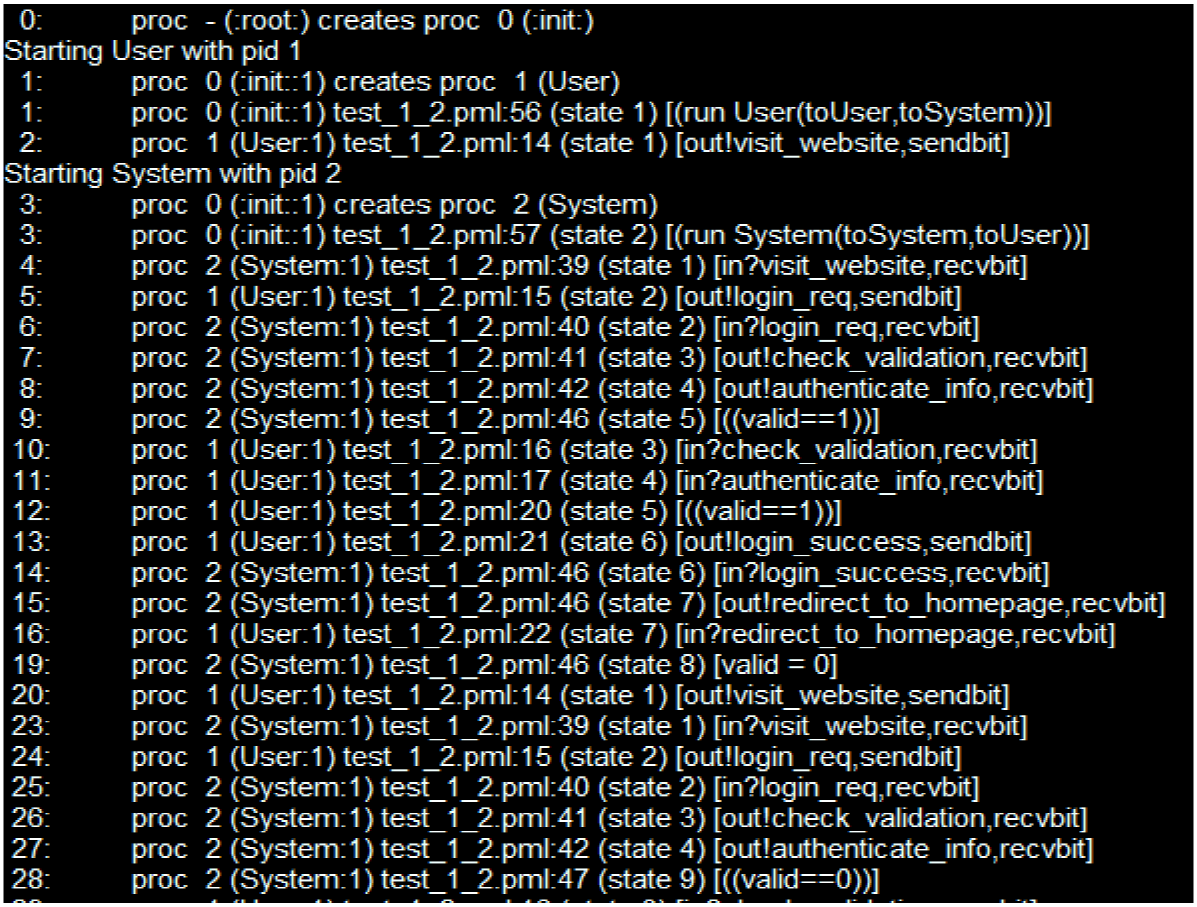




Process Simulation:



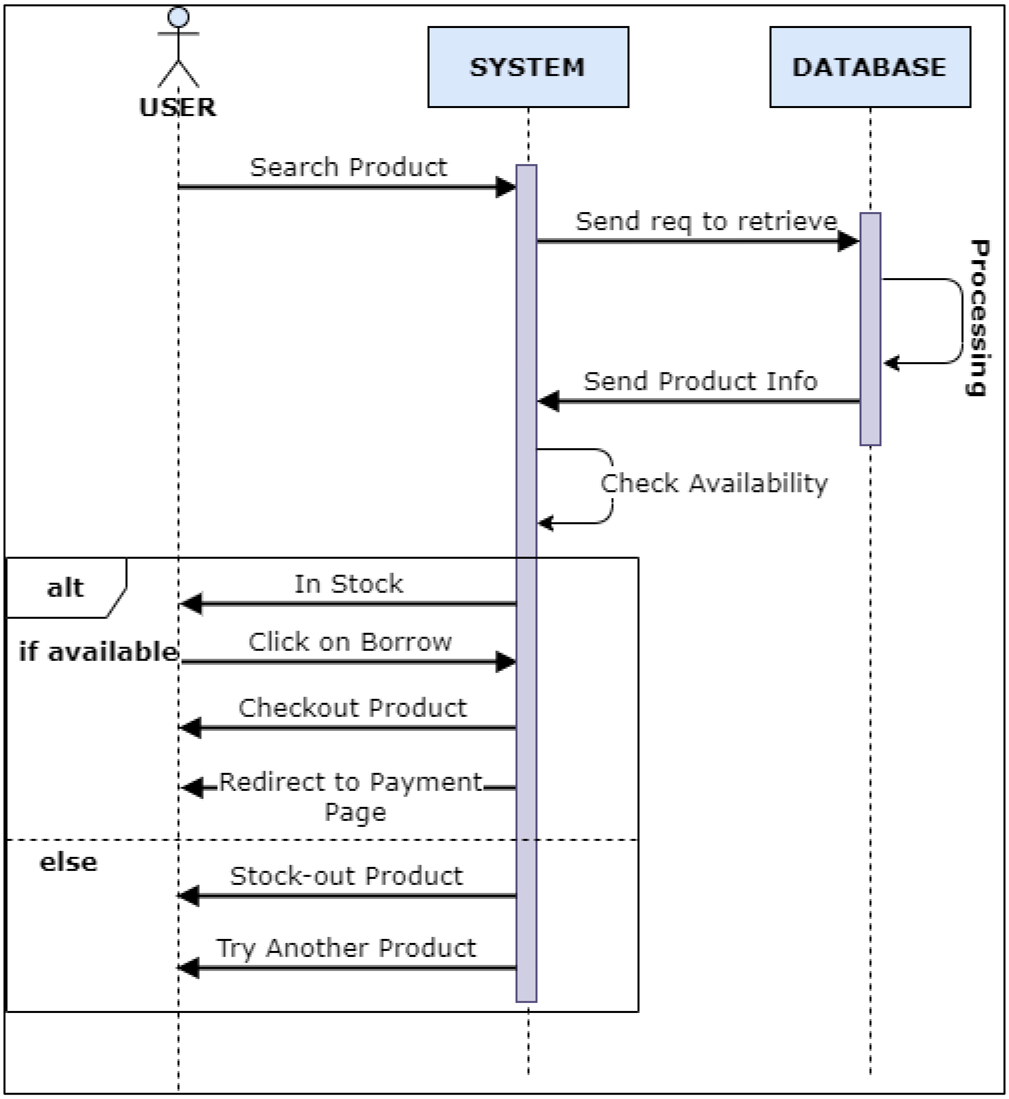
Process console:



```
  0:        proc  - (:root:) creates proc  0 (:init:)
Starting User with pid 1
  1:        proc  0 (:init::1) creates proc  1 (User)
  1:        proc  0 (:init::1) test_1_2.pml:56 (state 1) [(run User(toUser,toSystem))]
  2:        proc  1 (User:1) test_1_2.pml:14 (state 1) [out!visit_website,sendbit]
Starting System with pid 2
  3:        proc  0 (:init::1) creates proc  2 (System)
  3:        proc  0 (:init::1) test_1_2.pml:57 (state 2) [(run System(toSystem,toUser))]
  4:        proc  2 (System:1) test_1_2.pml:39 (state 1) [in?visit_website,recvbit]
  5:        proc  1 (User:1) test_1_2.pml:15 (state 2) [out!login_req,sendbit]
  6:        proc  2 (System:1) test_1_2.pml:40 (state 2) [in?login_req,recvbit]
  7:        proc  2 (System:1) test_1_2.pml:41 (state 3) [out!check_validation,recvbit]
  8:        proc  2 (System:1) test_1_2.pml:42 (state 4) [out!authenticate_info,recvbit]
  9:        proc  2 (System:1) test_1_2.pml:46 (state 5) [((valid==1))]
 10:        proc  1 (User:1) test_1_2.pml:16 (state 3) [in?check_validation,recvbit]
 11:        proc  1 (User:1) test_1_2.pml:17 (state 4) [in?authenticate_info,recvbit]
 12:        proc  1 (User:1) test_1_2.pml:20 (state 5) [((valid==1))]
 13:        proc  1 (User:1) test_1_2.pml:21 (state 6) [out!login_success,sendbit]
 14:        proc  2 (System:1) test_1_2.pml:46 (state 6) [in?login_success,recvbit]
 15:        proc  2 (System:1) test_1_2.pml:46 (state 7) [out!redirect_to_homepage,recvbit]
 16:        proc  1 (User:1) test_1_2.pml:22 (state 7) [in?redirect_to_homepage,recvbit]
 19:        proc  2 (System:1) test_1_2.pml:46 (state 8) [valid = 0]
 20:        proc  1 (User:1) test_1_2.pml:14 (state 1) [out!visit_website,sendbit]
 23:        proc  2 (System:1) test_1_2.pml:39 (state 1) [in?visit_website,recvbit]
 24:        proc  1 (User:1) test_1_2.pml:15 (state 2) [out!login_req,sendbit]
 25:        proc  2 (System:1) test_1_2.pml:40 (state 2) [in?login_req,recvbit]
 26:        proc  2 (System:1) test_1_2.pml:41 (state 3) [out!check_validation,recvbit]
 27:        proc  2 (System:1) test_1_2.pml:42 (state 4) [out!authenticate_info,recvbit]
 28:        proc  2 (System:1) test_1_2.pml:47 (state 9) [((valid==0))]
```

2nd sequence diagram (Borrow request):

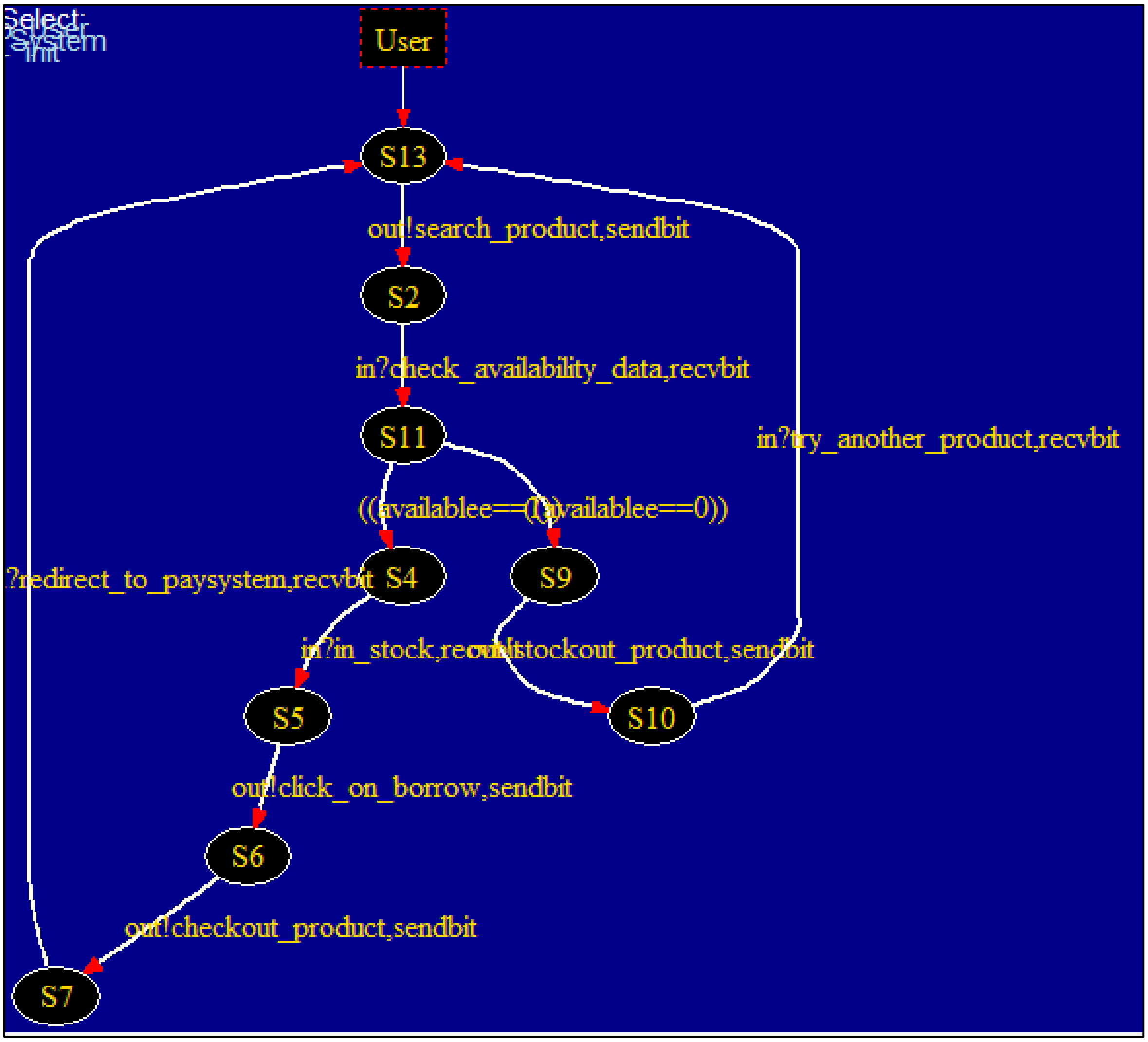
This diagram shows that users can borrow their necessary equipment. If the product is available then it gives a message, the user checkout that product, and the user will be connected to the homepage. If the product is not available, it gives a message the product is stock out then the user tries for another product. This information also stores in database.



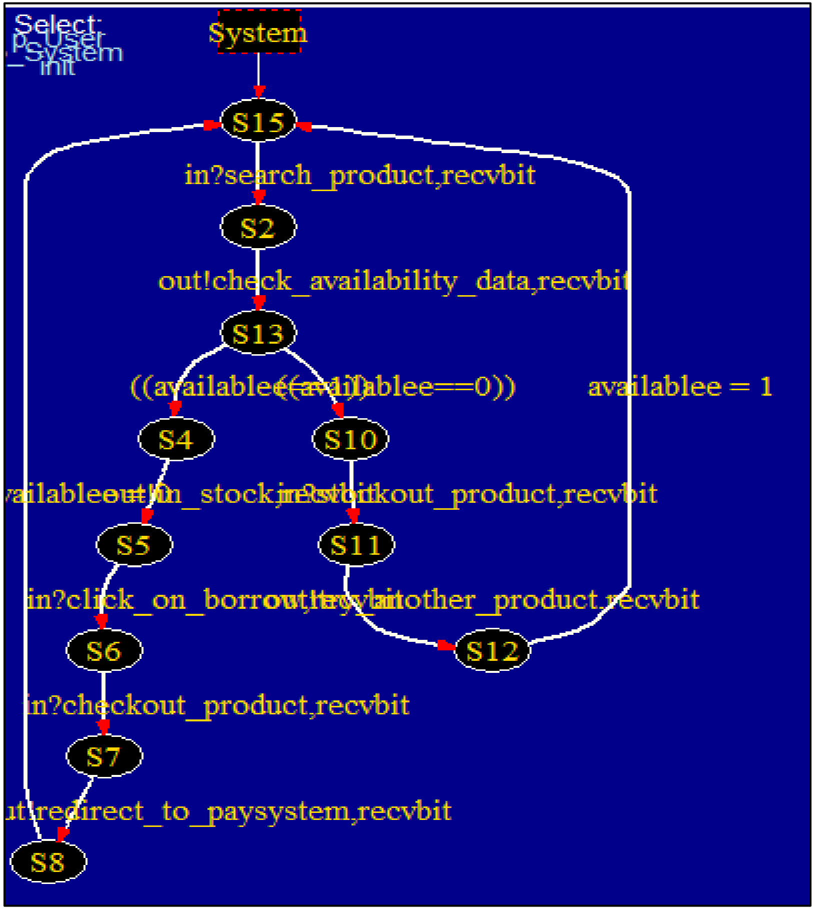
Promela Code for 2nd Sequence Diagram:

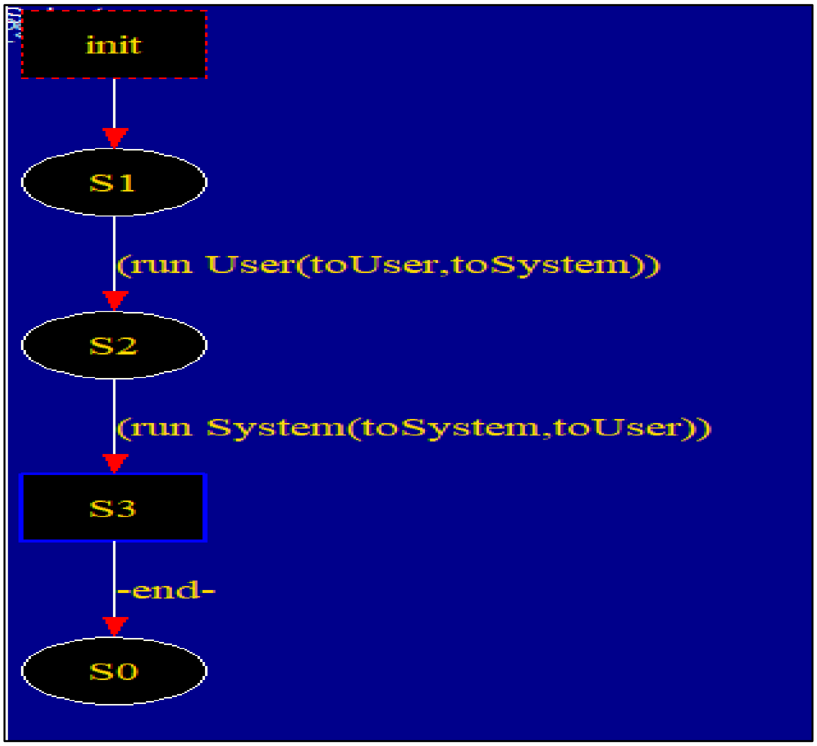
```
mtype = {search_product, send_request_to_retrive, processing_req,
receive_product_info, check_availability_data, available,
checkout_product, redirect_to_paysystem, not_available,
stockout_product, try_another_product, click_on_borrow, in_stock};

chan toUser = [2] of {mtype,bit};
chan toSystem = [2] of {mtype,bit};
bool availablee = 1;
proctype User(chan in, out)
{
    bit sendbit, recvbit;
    do
    ::out ! search_product, sendbit;
       in ? check_availability_data, recvbit;
    if
    :: availablee == 1 ->
        in ? in_stock, recvbit;
        out ! click_on_borrow, sendbit ->

        out ! checkout_product, sendbit ->
        in ? redirect_to_paysystem, recvbit;

    ::  availablee == 0 ->
        out ! stockout_product, sendbit ->
        in ? try_another_product, recvbit;
    fi
    od
}
proctype System(chan in, out)
{
    bit recvbit;

    do::

    in ? search_product(recvbit);
    out ! check_availability_data(recvbit);
    if
    :: availablee == 1 ->
    out ! in_stock(recvbit);
    in ? click_on_borrow(recvbit);
    in ? checkout_product(recvbit) -> out !
redirect_to_paysystem(recvbit);
      availablee = 0;
    :: availablee == 0 -> in ? stockout_product(recvbit) -> out !
try_another_product(recvbit);
     availablee = 1;
    fi
    od
}
init
{
    run User(toUser, toSystem);
    run System(toSystem, toUser);
}
```

Automata for User:

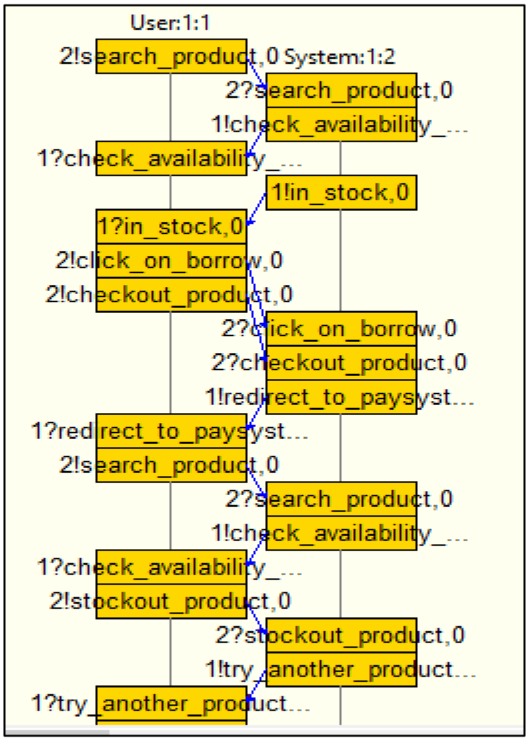

Select:
User
System
init
User
S13
out!search_product,sendbit
S2
in?check_availability_data,recvbit
S11
((availablee==1))
((availablee==0))
S4
S9
in?try_another_product,recvbit
?redirect_to_paysystem,recvbit
in?in_stock,recvbit
out!stockout_product,sendbit
S5
S10
out!click_on_borrow,sendbit
S6
out!checkout_product,sendbit
S7

Automata for System:

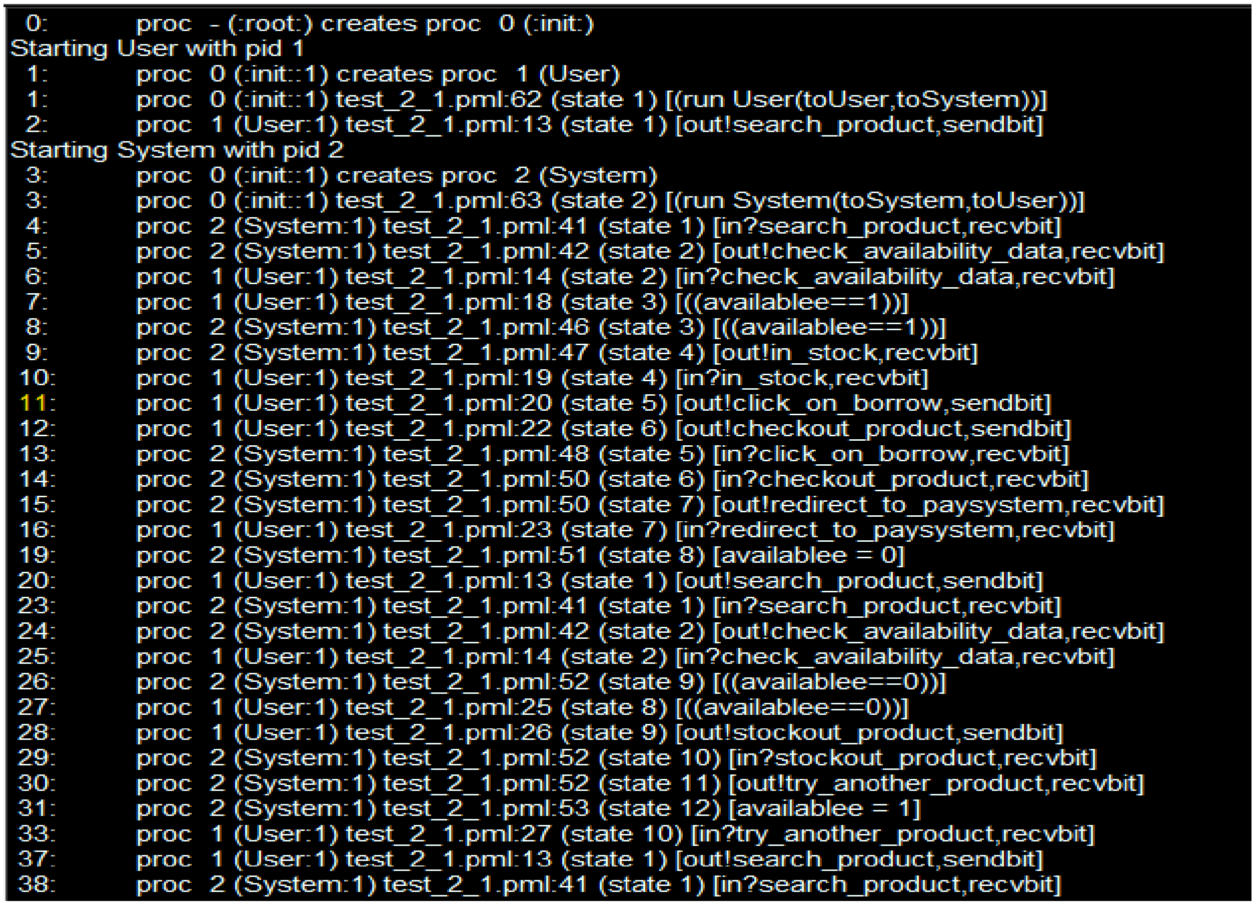

Select:
p_User
p_System
_init
System
S15
in?search_product,recvbit
S2
out!check_availability_data,recvbit
S13
((availablee==1))
((availablee==0))
availablee = 1
S4
S10
ailablee = 1
out!in_stock
in?checkout_product,recvbit
S5
S11
in?click_on_borrow,recvbit
out!try_another_product,recvbit
S12
S6
in?checkout_product,recvbit
S7
ut!redirect_to_paysystem,recvbit
S8



Process Simulation:



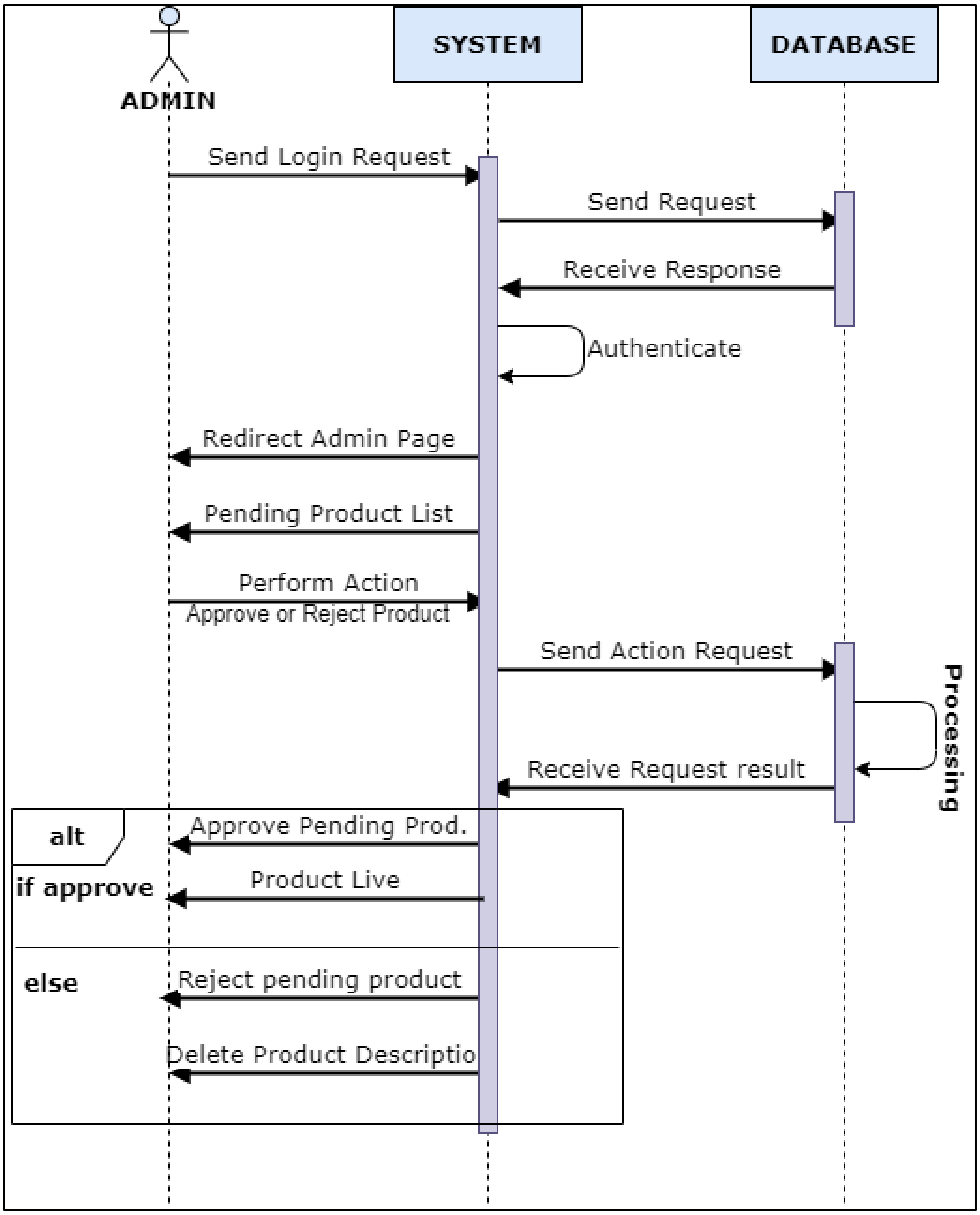
Process console:



```
  0:        proc  - (:root:) creates proc  0 (:init:)
Starting User with pid 1
  1:        proc  0 (:init::1) creates proc  1 (User)
  1:        proc  0 (:init::1) test_2_1.pml:62 (state 1) [(run User(toUser,toSystem))]
  2:        proc  1 (User:1) test_2_1.pml:13 (state 1) [out!search_product,sendbit]
Starting System with pid 2
  3:        proc  0 (:init::1) creates proc  2 (System)
  3:        proc  0 (:init::1) test_2_1.pml:63 (state 2) [(run System(toSystem,toUser))]
  4:        proc  2 (System:1) test_2_1.pml:41 (state 1) [in?search_product,recvbit]
  5:        proc  2 (System:1) test_2_1.pml:42 (state 2) [out!check_availability_data,recvbit]
  6:        proc  1 (User:1) test_2_1.pml:14 (state 2) [in?check_availability_data,recvbit]
  7:        proc  1 (User:1) test_2_1.pml:18 (state 3) [((availablee==1))]
  8:        proc  2 (System:1) test_2_1.pml:46 (state 3) [((availablee==1))]
  9:        proc  2 (System:1) test_2_1.pml:47 (state 4) [out!in_stock,recvbit]
 10:        proc  1 (User:1) test_2_1.pml:19 (state 4) [in?in_stock,recvbit]
 11:        proc  1 (User:1) test_2_1.pml:20 (state 5) [out!click_on_borrow,sendbit]
 12:        proc  1 (User:1) test_2_1.pml:22 (state 6) [out!checkout_product,sendbit]
 13:        proc  2 (System:1) test_2_1.pml:48 (state 5) [in?click_on_borrow,recvbit]
 14:        proc  2 (System:1) test_2_1.pml:50 (state 6) [in?checkout_product,recvbit]
 15:        proc  2 (System:1) test_2_1.pml:50 (state 7) [out!redirect_to_paysystem,recvbit]
 16:        proc  1 (User:1) test_2_1.pml:23 (state 7) [in?redirect_to_paysystem,recvbit]
 19:        proc  2 (System:1) test_2_1.pml:51 (state 8) [availablee = 0]
 20:        proc  1 (User:1) test_2_1.pml:13 (state 1) [out!search_product,sendbit]
 23:        proc  2 (System:1) test_2_1.pml:41 (state 1) [in?search_product,recvbit]
 24:        proc  2 (System:1) test_2_1.pml:42 (state 2) [out!check_availability_data,recvbit]
 25:        proc  1 (User:1) test_2_1.pml:14 (state 2) [in?check_availability_data,recvbit]
 26:        proc  2 (System:1) test_2_1.pml:52 (state 9) [((availablee==0))]
 27:        proc  1 (User:1) test_2_1.pml:25 (state 8) [((availablee==0))]
 28:        proc  1 (User:1) test_2_1.pml:26 (state 9) [out!stockout_product,sendbit]
 29:        proc  2 (System:1) test_2_1.pml:52 (state 10) [in?stockout_product,recvbit]
 30:        proc  2 (System:1) test_2_1.pml:52 (state 11) [out!try_another_product,recvbit]
 31:        proc  2 (System:1) test_2_1.pml:53 (state 12) [availablee = 1]
 33:        proc  1 (User:1) test_2_1.pml:27 (state 10) [in?try_another_product,recvbit]
 37:        proc  1 (User:1) test_2_1.pml:13 (state 1) [out!search_product,sendbit]
 38:        proc  2 (System:1) test_2_1.pml:41 (state 1) [in?search_product,recvbit]
```

3rd sequence diagram (Admin login):

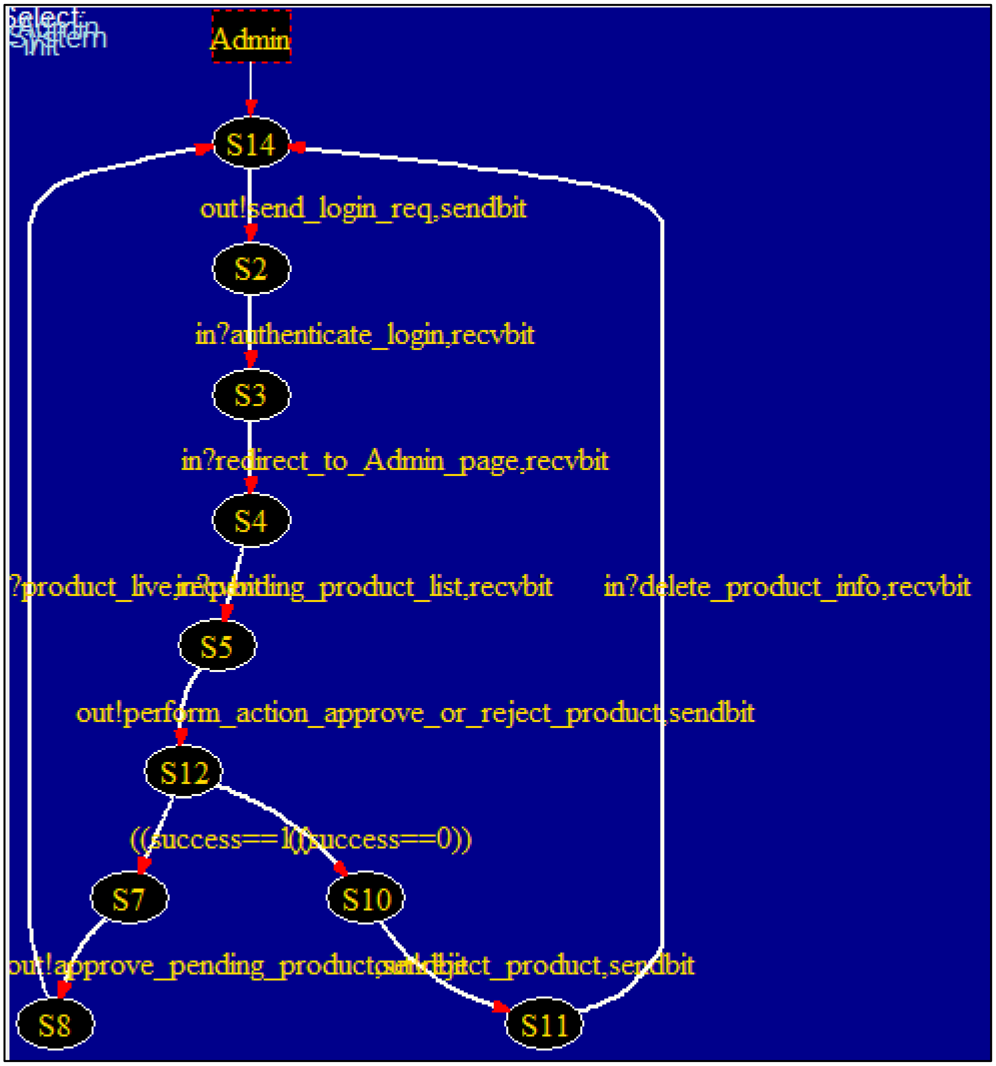
In this diagram admin also can login, reject pending product, and delete product description. This information also stores in database.



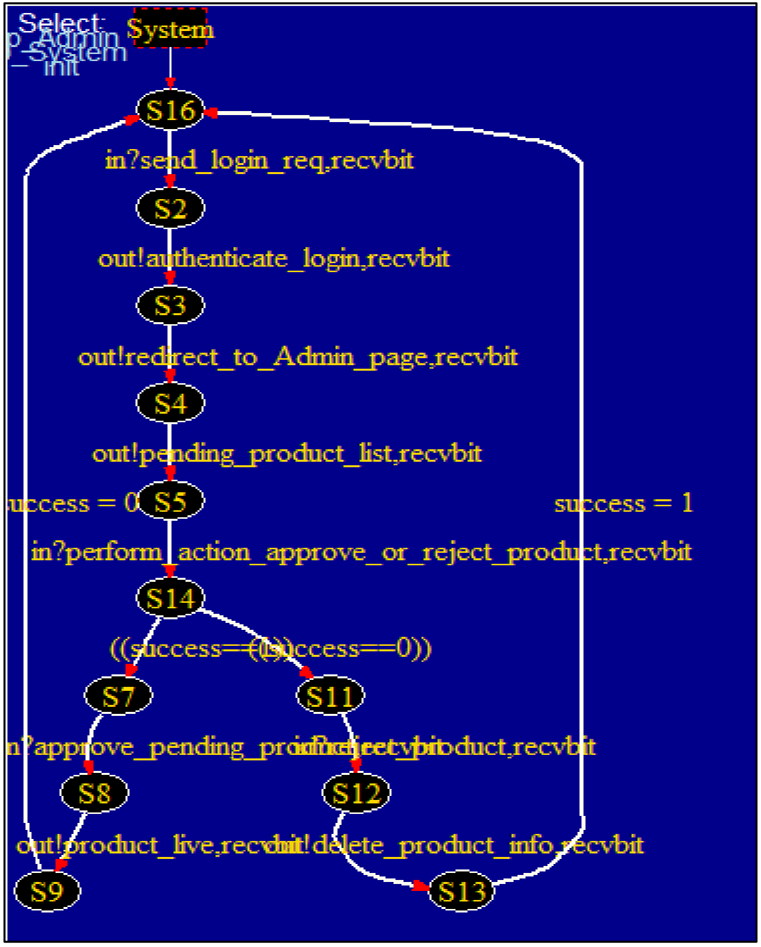
Promela Code for 3rd Sequence Diagram:

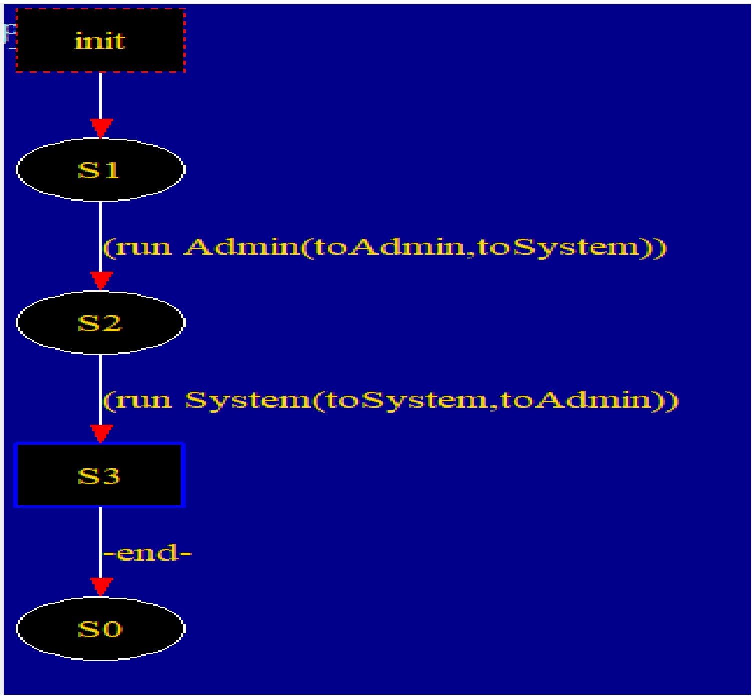
```
mtype = {send_login_req, send_request, receive_response,
authenticate_login, redirect_to_Admin_page, pending_product_list,
perform_action_approve_or_reject_product, send_action_request,
processing, receive_request_result, approve_pending_product,
product_live, reject_product, delete_product_info};

chan toAdmin = [2] of {mtype,bit};
chan toSystem = [2] of {mtype,bit};
bool success = 1;
proctype Admin(chan in, out)
{
    bit sendbit, recvbit;
    do
    :: out ! send_login_req, sendbit;
    in ?authenticate_login, recvbit;
    in ? redirect_to_Admin_page, recvbit;
    in ? pending_product_list, recvbit;
    out ! perform_action_approve_or_reject_product, sendbit;
    if
    :: success == 1 ->
        out ! approve_pending_product, sendbit ->
        in ? product_live, recvbit;
    ::  success == 0 ->
        out ! reject_product, sendbit ->
        in ? delete_product_info, recvbit;
    fi

    od
}
proctype System(chan in, out)
{
    bit recvbit;
    do::
    in ? send_login_req(recvbit);
    out ! authenticate_login(recvbit);
    out ! redirect_to_Admin_page(recvbit);
    out ! pending_product_list(recvbit);
    in ? perform_action_approve_or_reject_product(recvbit);
    if
    :: success == 1 -> in ? approve_pending_product(recvbit) -> out !
product_live(recvbit); success = 0;
    :: success == 0 -> in ? reject_product(recvbit) -> out !
delete_product_info(recvbit);
    success = 1;
    fi
    od
}
init
{
    run Admin(toAdmin, toSystem);
    run System(toSystem, toAdmin);
}
```

Automata for Admin:

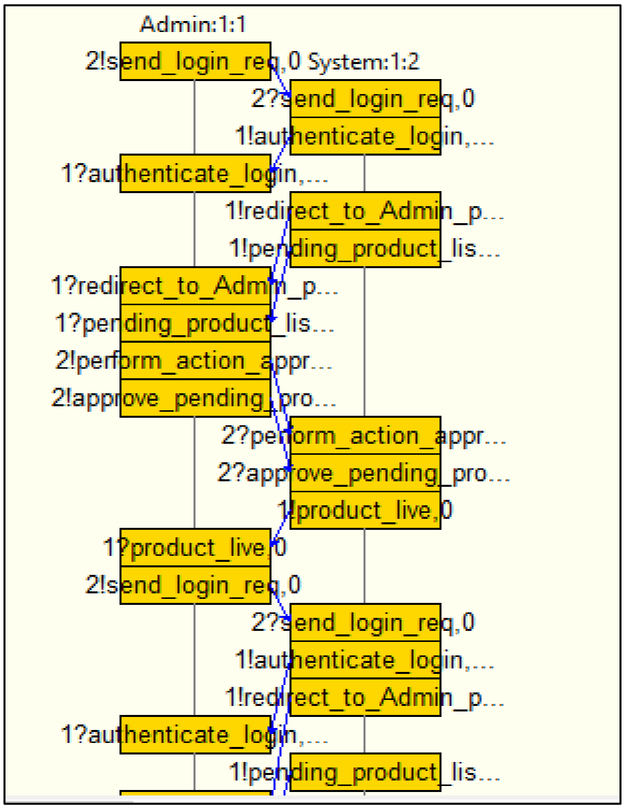

Select:
Admin
System
init
Admin
S14
out!send_login_req,sendbit
S2
in?authenticate_login,recvbit
S3
in?redirect_to_Admin_page,recvbit
S4
?product_live,recvbit
in?pending_product_list,recvbit
in?delete_product_info,recvbit
S5
out!perform_action_approve_or_reject_product,sendbit
S12
((success==1))
((success==0))
S7
S10
out!approve_pending_product,sendbit
out!reject_product,sendbit
S8
S11

Automata for System:

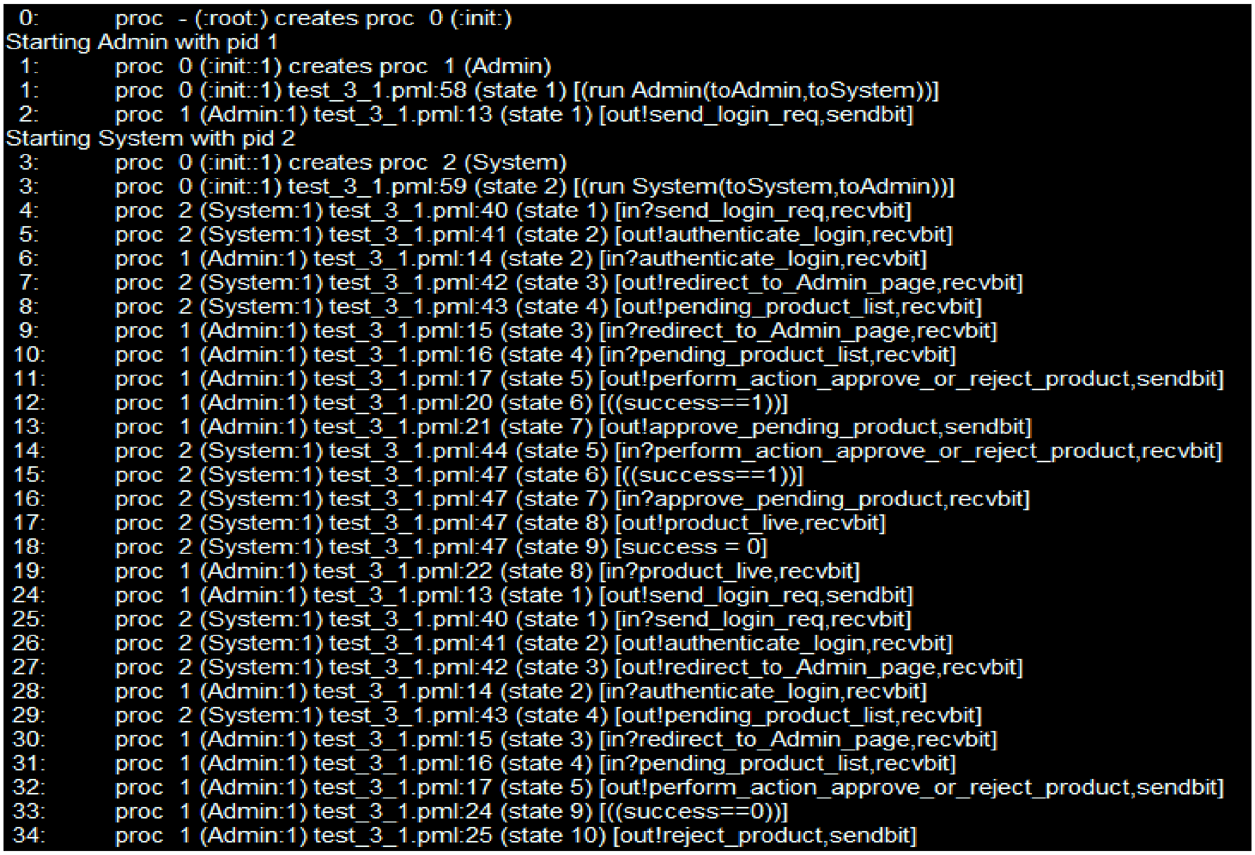

Select:
p_Admin
p_System
_init
System
S16
in?send_login_req,recvbit
S2
out!authenticate_login,recvbit
S3
out!redirect_to_Admin_page,recvbit
S4
out!pending_product_list,recvbit
uccess = 0
S5
success = 1
in?perform_action_approve_or_reject_product,recvbit
S14
((success==1))
((success==0))
S7
S11
n?approve_pending_product,recvbit
in?reject_product,recvbit
S8
S12
out!product_live,recvbit
out!delete_product_info,recvbit
S9
S13



Process Simulation:



Process console:



```
 0:        proc  - (:root:) creates proc  0 (:init:)
Starting Admin with pid 1
 1:        proc  0 (:init::1) creates proc  1 (Admin)
 1:        proc  0 (:init::1) test_3_1.pml:58 (state 1) [(run Admin(toAdmin,toSystem))]
 2:        proc  1 (Admin:1) test_3_1.pml:13 (state 1) [out!send_login_req,sendbit]
Starting System with pid 2
 3:        proc  0 (:init::1) creates proc  2 (System)
 3:        proc  0 (:init::1) test_3_1.pml:59 (state 2) [(run System(toSystem,toAdmin))]
 4:        proc  2 (System:1) test_3_1.pml:40 (state 1) [in?send_login_req,recvbit]
 5:        proc  2 (System:1) test_3_1.pml:41 (state 2) [out!authenticate_login,recvbit]
 6:        proc  1 (Admin:1) test_3_1.pml:14 (state 2) [in?authenticate_login,recvbit]
 7:        proc  2 (System:1) test_3_1.pml:42 (state 3) [out!redirect_to_Admin_page,recvbit]
 8:        proc  2 (System:1) test_3_1.pml:43 (state 4) [out!pending_product_list,recvbit]
 9:        proc  1 (Admin:1) test_3_1.pml:15 (state 3) [in?redirect_to_Admin_page,recvbit]
10:        proc  1 (Admin:1) test_3_1.pml:16 (state 4) [in?pending_product_list,recvbit]
11:        proc  1 (Admin:1) test_3_1.pml:17 (state 5) [out!perform_action_approve_or_reject_product,sendbit]
12:        proc  1 (Admin:1) test_3_1.pml:20 (state 6) [((success==1))]
13:        proc  1 (Admin:1) test_3_1.pml:21 (state 7) [out!approve_pending_product,sendbit]
14:        proc  2 (System:1) test_3_1.pml:44 (state 5) [in?perform_action_approve_or_reject_product,recvbit]
15:        proc  2 (System:1) test_3_1.pml:47 (state 6) [((success==1))]
16:        proc  2 (System:1) test_3_1.pml:47 (state 7) [in?approve_pending_product,recvbit]
17:        proc  2 (System:1) test_3_1.pml:47 (state 8) [out!product_live,recvbit]
18:        proc  2 (System:1) test_3_1.pml:47 (state 9) [success = 0]
19:        proc  1 (Admin:1) test_3_1.pml:22 (state 8) [in?product_live,recvbit]
24:        proc  1 (Admin:1) test_3_1.pml:13 (state 1) [out!send_login_req,sendbit]
25:        proc  2 (System:1) test_3_1.pml:40 (state 1) [in?send_login_req,recvbit]
26:        proc  2 (System:1) test_3_1.pml:41 (state 2) [out!authenticate_login,recvbit]
27:        proc  2 (System:1) test_3_1.pml:42 (state 3) [out!redirect_to_Admin_page,recvbit]
28:        proc  1 (Admin:1) test_3_1.pml:14 (state 2) [in?authenticate_login,recvbit]
29:        proc  2 (System:1) test_3_1.pml:43 (state 4) [out!pending_product_list,recvbit]
30:        proc  1 (Admin:1) test_3_1.pml:15 (state 3) [in?redirect_to_Admin_page,recvbit]
31:        proc  1 (Admin:1) test_3_1.pml:16 (state 4) [in?pending_product_list,recvbit]
32:        proc  1 (Admin:1) test_3_1.pml:17 (state 5) [out!perform_action_approve_or_reject_product,sendbit]
33:        proc  1 (Admin:1) test_3_1.pml:24 (state 9) [((success==0))]
34:        proc  1 (Admin:1) test_3_1.pml:25 (state 10) [out!reject_product,sendbit]
```