

HEALTH EMERGENCY

East West University

Mini Project : Part-1

CSE435
Section-1

Submitted by:

Md. Sadat Ahmed

ID:2017-2-60-062

Nahida Sultana

ID:2017-60-139

SUBMITTED TO
DR. SHAMIM H RIPON

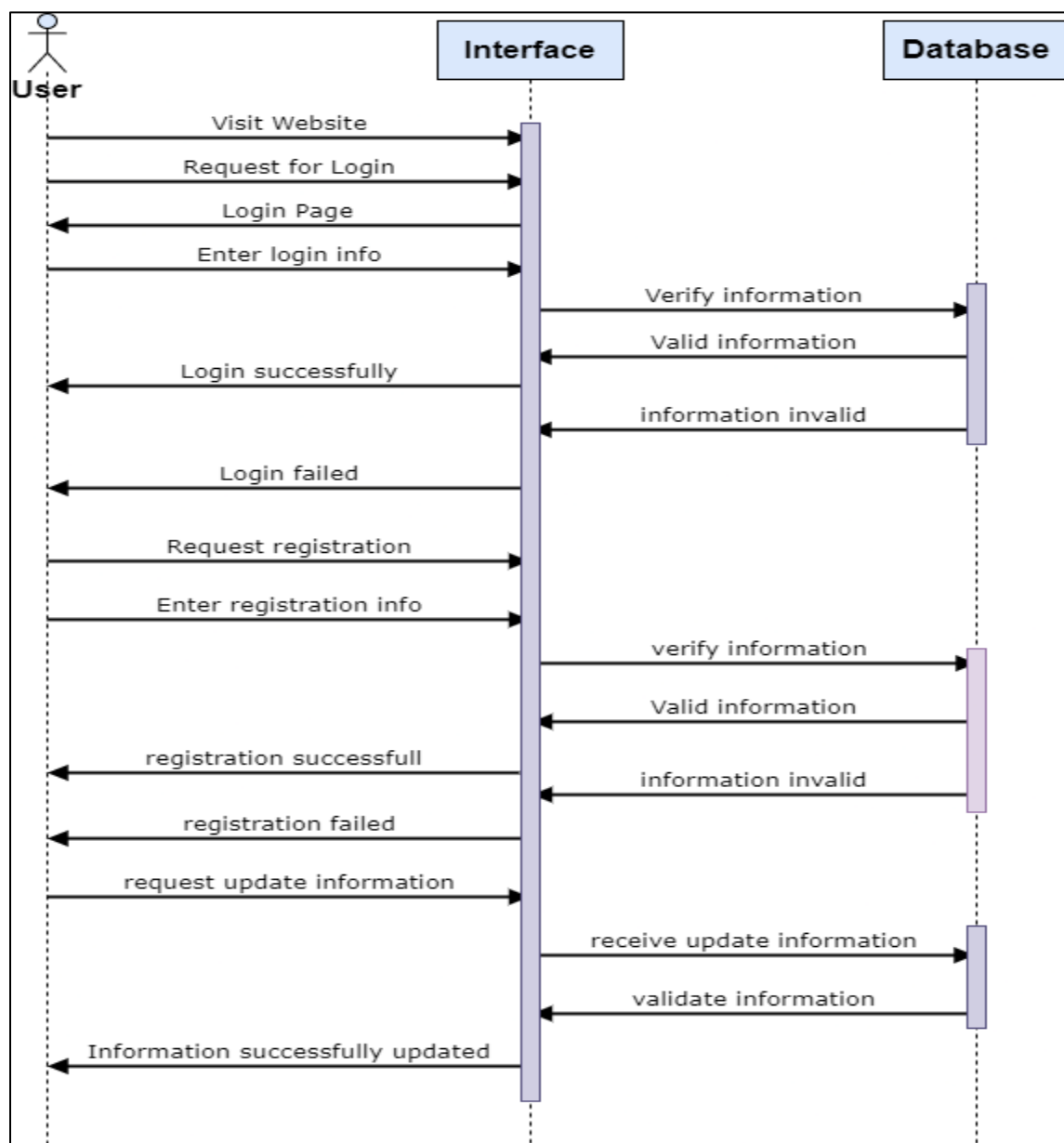
Professor Department of Computer Science &
Engineering
East West University

Health Emergency

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, logout, request registration & update):

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.



FSP Code:

USER = (visit_website -> req_login -> enter_logininfo -> req_registration -> enter_reg_info -> req_upd_info -> USER).

INTERFACE = (login_page -> verify_info -> login_succ -> login_fail -> verify_reg_info -> reg_succ -> reg_fail -> verify_upd_info -> info_updated -> INTERFACE).

DATABASE = (valid_info -> invalid_info -> valid_reg_info -> invalid_reg_info -> valid_upd_info -> DATABASE).

||USER_LOG = (USER || INTERFACE || DATABASE)
/{login_succ/valid_info, login_fail/invalid_info, reg_succ/valid_reg_info,
reg_fail/invalid_reg_info, info_updated/valid_upd_info,
enter_logininfo/verify_info, enter_reg_info/verify_reg_info,
req_upd_info/verify_upd_info, login_page/req_login}.

Transition Diagram:

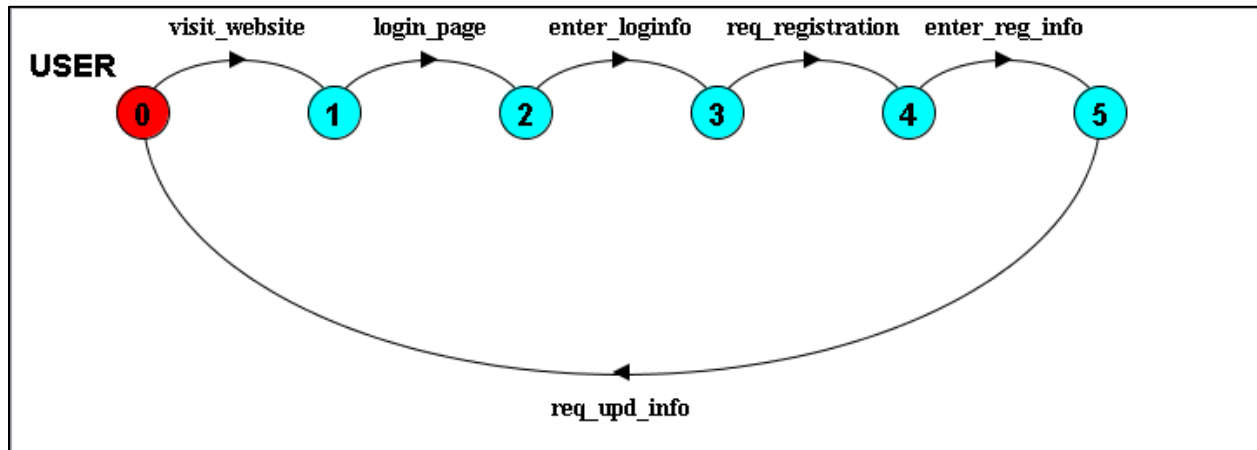


Figure: USER FSP Diagram.

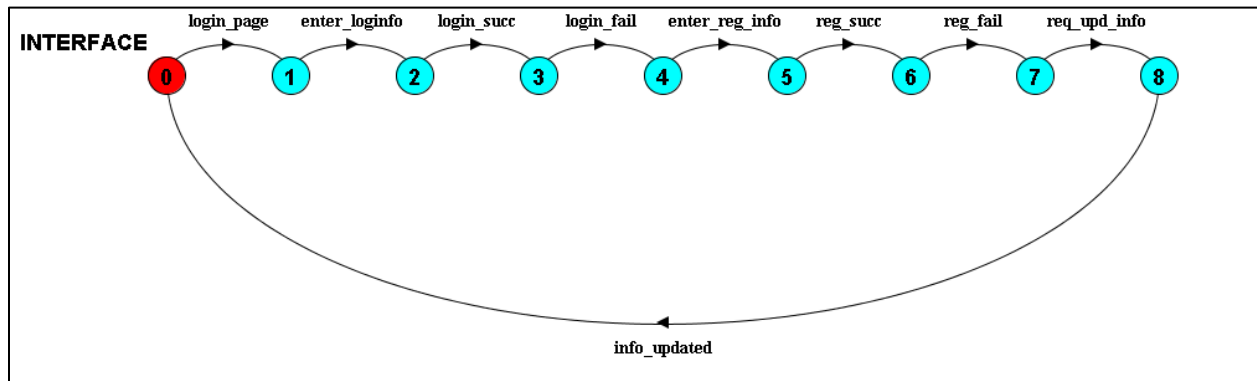


Figure: INTERFACE FSP Diagram.

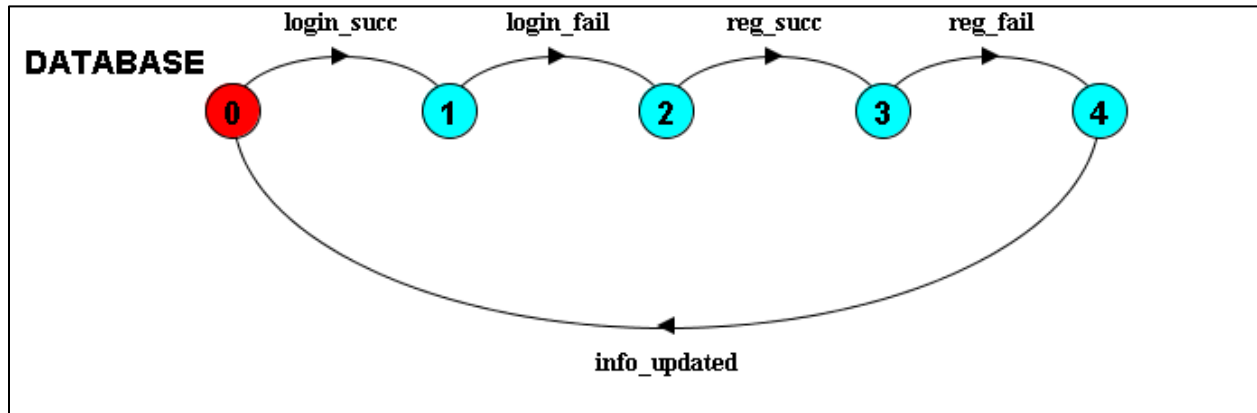


Figure: DATABASE FSP Diagram.

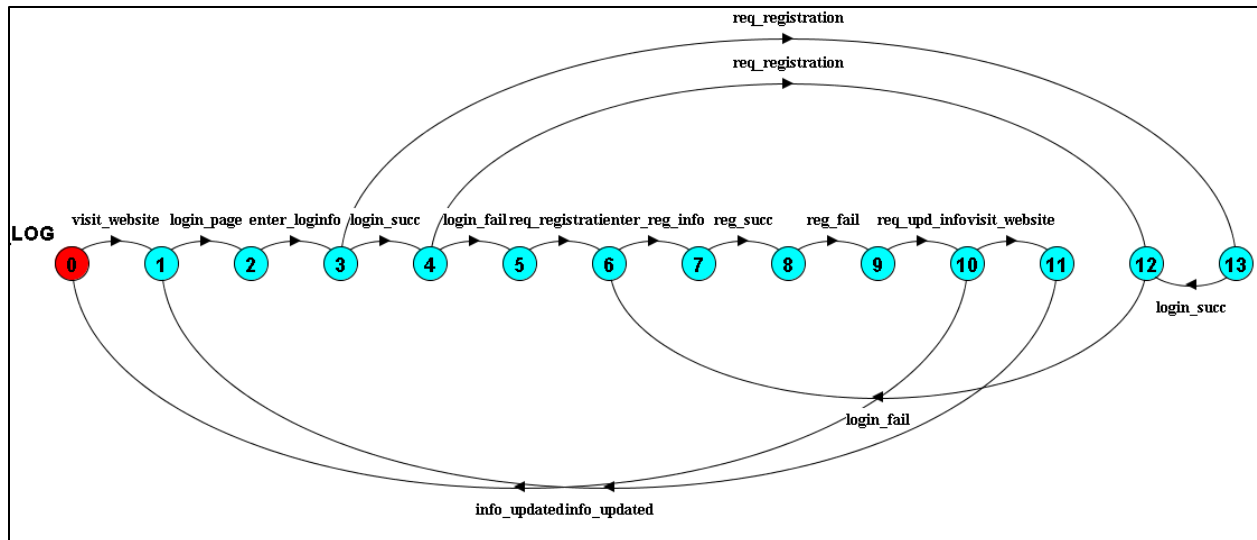
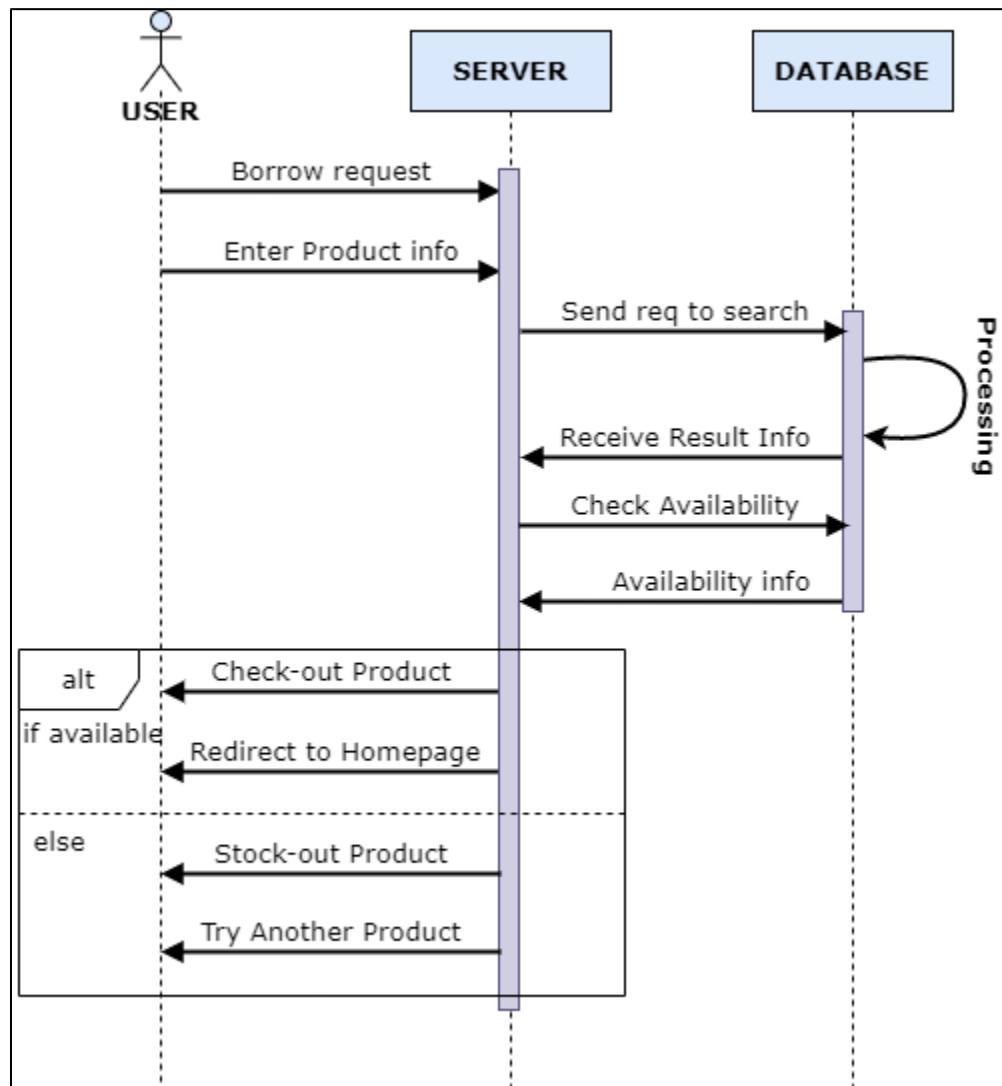


Figure: USER_LOG FSP Diagram.

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.



FSP Code:

USER =(send_borrow_req -> send_product_info -> (checkout_product -> return_home -> USER
| stockout_product -> try_other_product -> USER)).

SERVER = (receive_borrow_req -> receive_product_info -> send_req_to_search ->
receive_result_info -> check_availability -> receive_available_info -> (available -> return_home
-> SERVER | not_available -> try_other_product -> SERVER)).

DATABASE = (receive_req_to_search -> processing_req -> send_result_info ->
check_availability -> send_available_info -> DATABASE).

||BORROW_REQ = (USER || SERVER || DATABASE)

/ {send_borrow_req/receive_borrow_req, send_product_info/receive_product_info,
checkout_product/available, stockout_product/not_available,
send_req_to_search/receive_req_to_search, receive_search_info/send_search_info,
send_req_to_search/receive_req_to_search, receive_result_info/send_result_info,
receive_available_info/send_available_info}.

Transition Diagram:

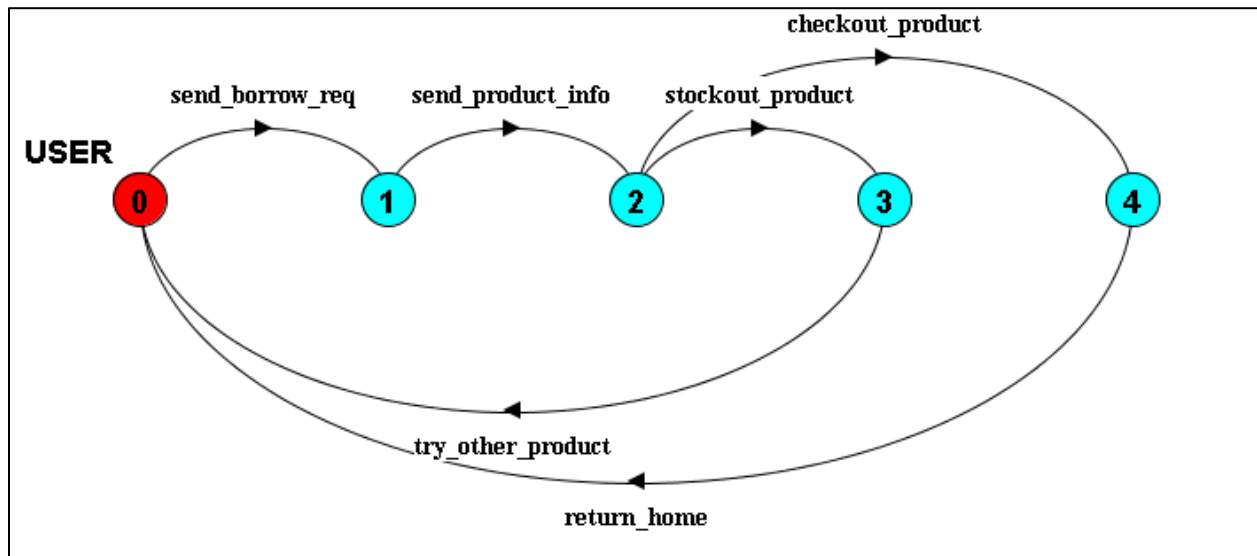


Figure: USER FSP Diagram.

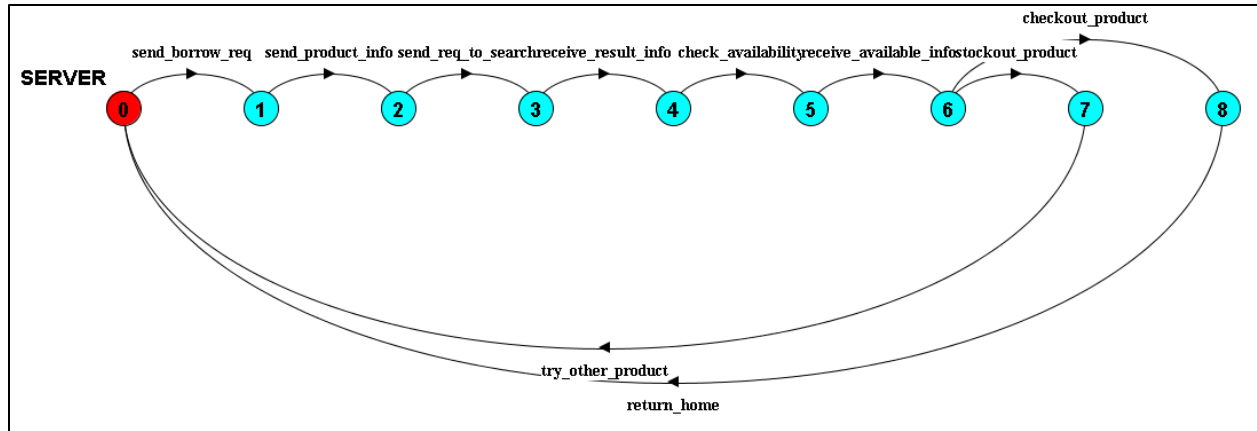


Figure: SERVER FSP Diagram.

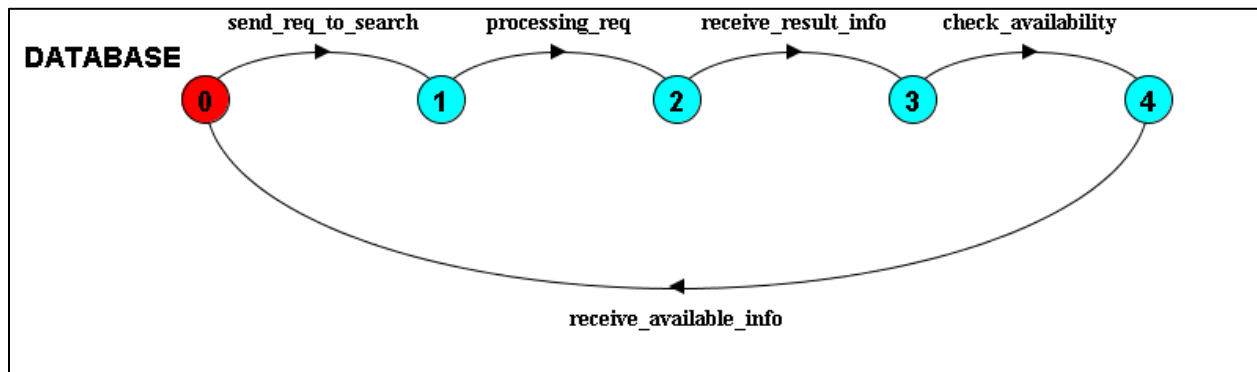


Figure: DATABASE FSP Diagram.

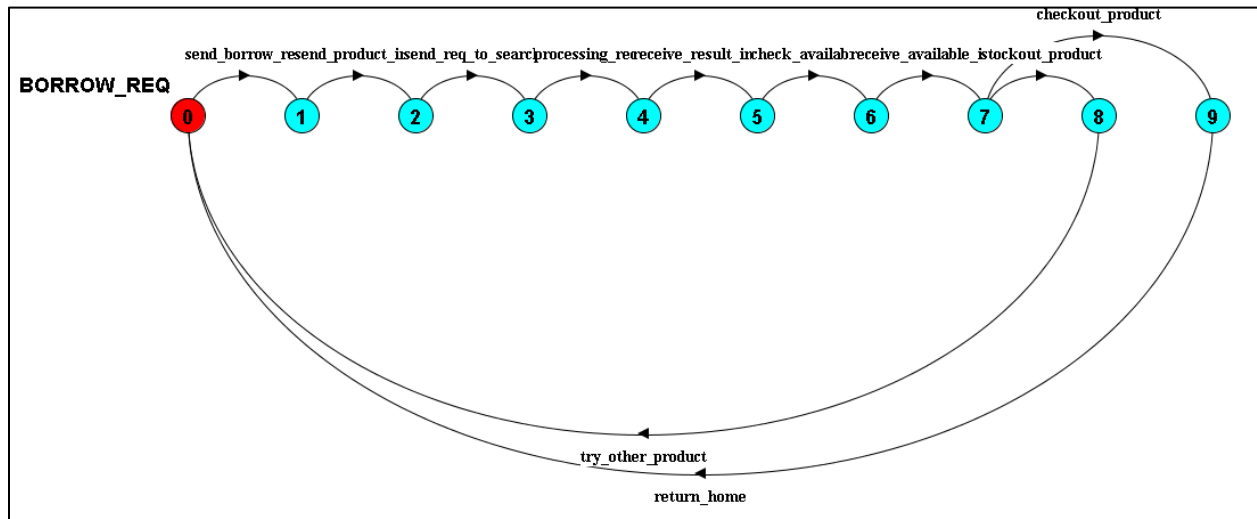
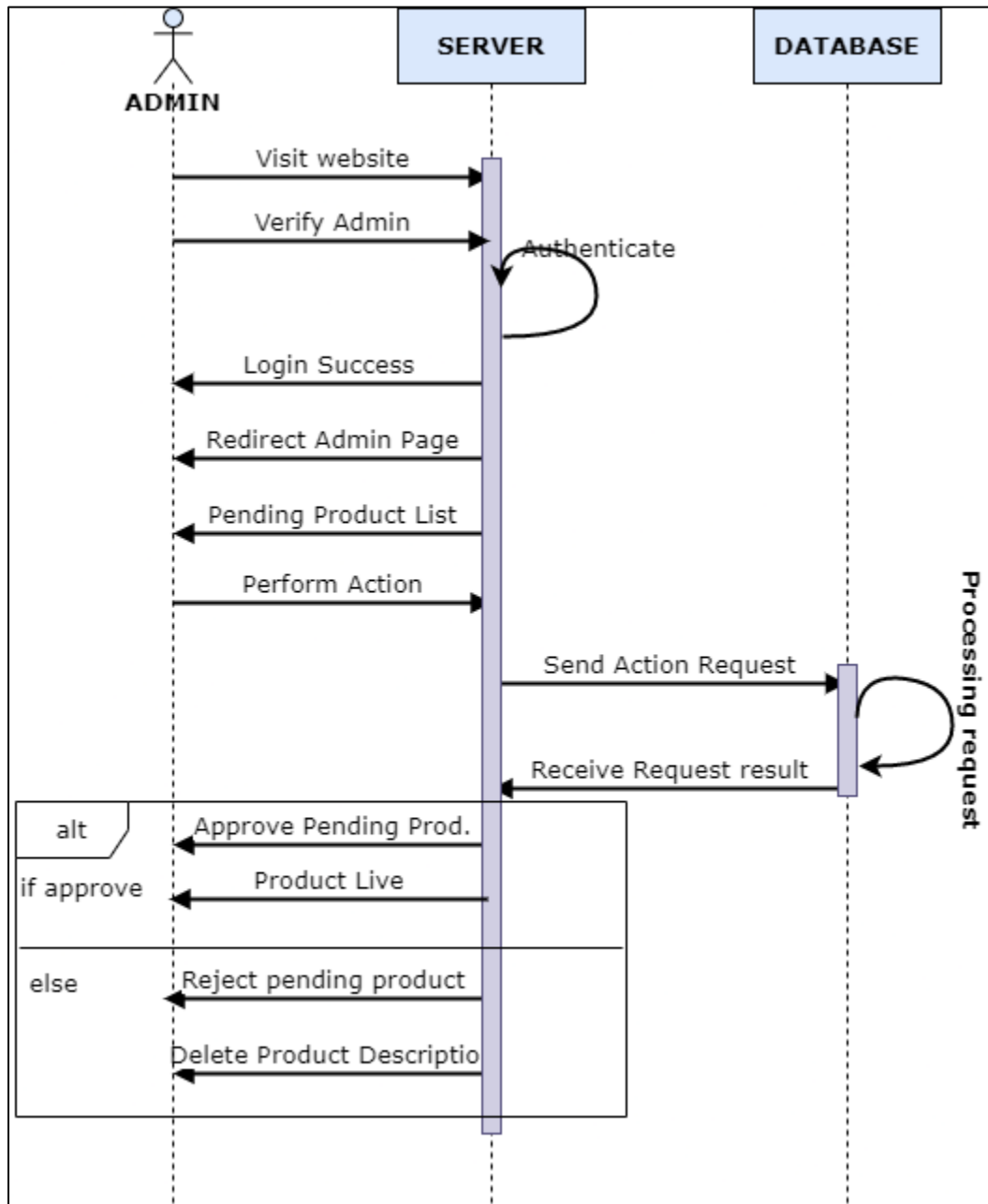


Figure: BORROW_REQ FSP Diagram.

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.



FSP Code:

ADMIN =(visit_web -> verify_admin -> receive_login_succ -> redirect_adminpage -> receive_pending_list -> perform_act -> (approve_pending_pro -> product_live -> ADMIN | reject_pending_pro -> reject_succ -> ADMIN)).

SERVER = (visit_web -> verify_admin -> authenticate -> send_login_succ -> redirect_adminpage -> send_pending_list -> perform_act -> send_act_req -> receive_result -> (allow_access -> product_live -> SERVER | product_info_remove_req -> delete_pro_description -> SERVER)).

DATABASE = (receive_act_req -> processing_req -> send_result -> DATABASE).

||ADMIN_SER = (ADMIN || SERVER || DATABASE)
/{loggedin/send_login_succ, loggedin/receive_login_succ,
pending_list/send_pending_list, pending_list/receive_pending_list,
approve/approve_pending_pro, approve/allow_access,
reject_pending_pro/product_info_remove_req,
reject_succ/delete_pro_description, action/send_act_req, action/receive_act_req,
receive_result/send_result}.

Transition Diagram:

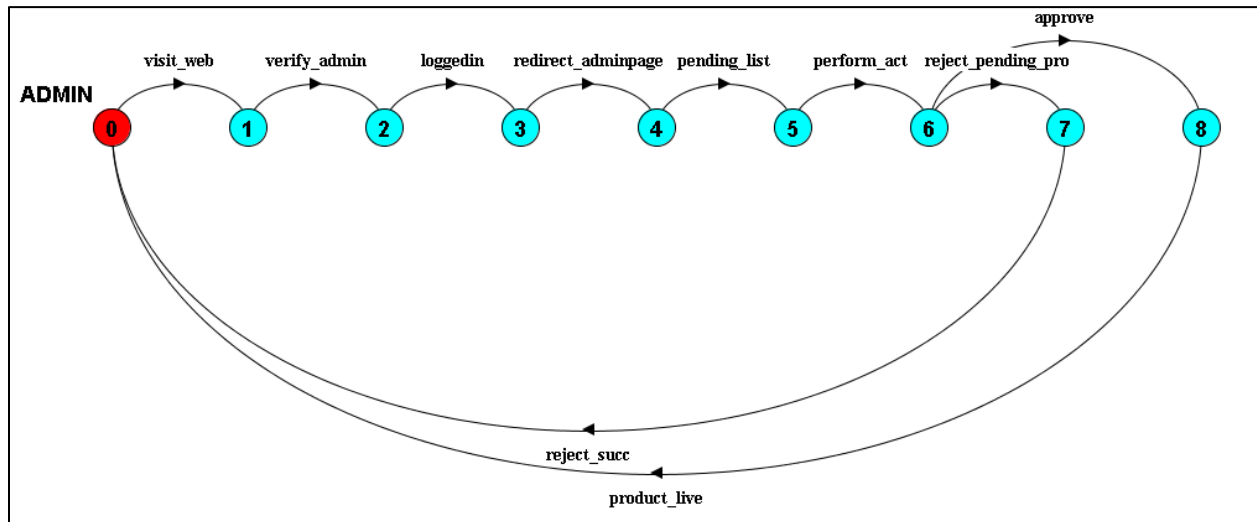


Figure: ADMIN FSP Diagram.

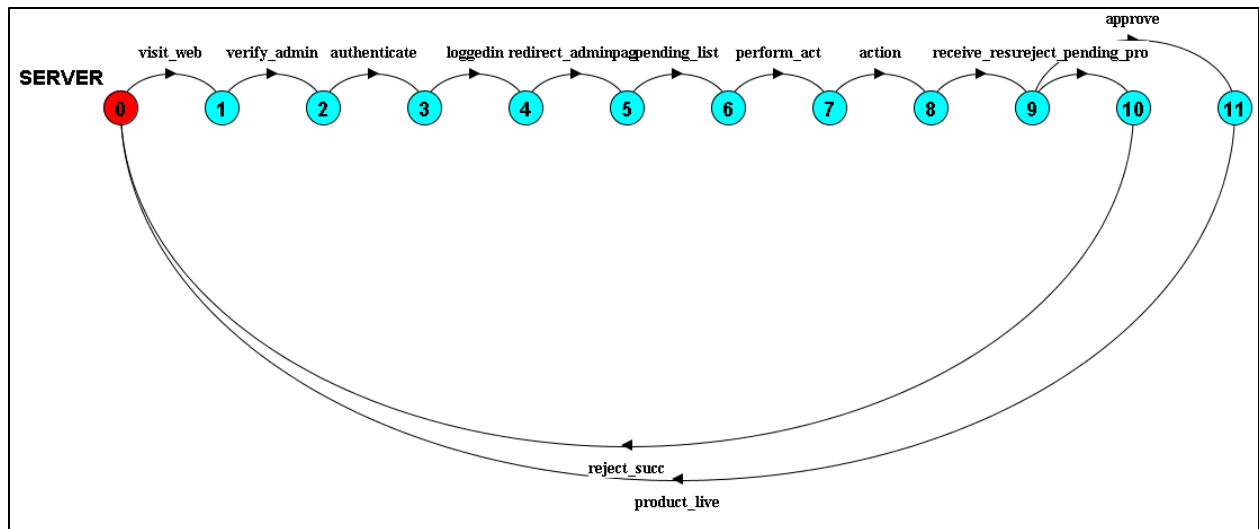


Figure: SERVER FSP Diagram.

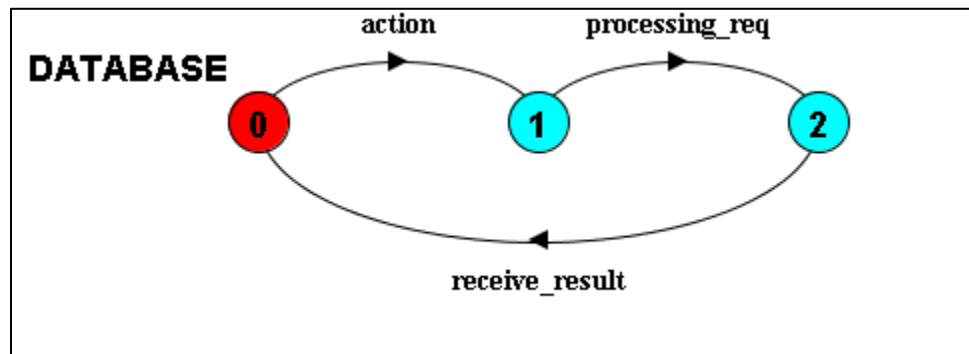


Figure: DATABASE FSP Diagram.

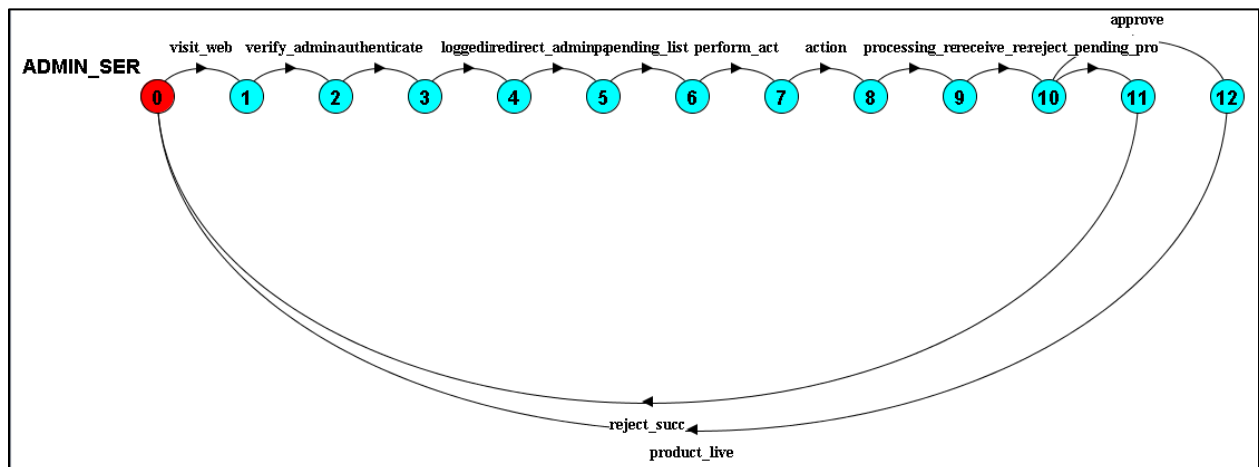


Figure: ADMIN_SERVER FSP Diagram.