Project Log Book

Group Members:

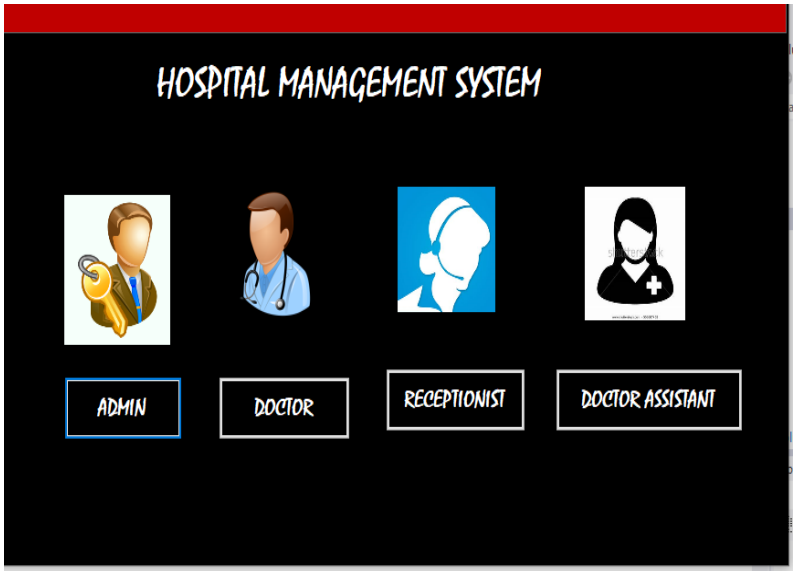
Entry Date	Work Done
March, 13,2022	Discussed the basic plan to build the prototype for CRM in class, noting down all constraints to be taken care of. Furthermore, we decided our next group meeting would be on March, 13, 2022(Saturday) at 3:30, meeting place: PAF KIET University.
March, 14,2022	Meeting at PAF KIET University: We discussed about the project objective. Using the Software Management Plan template printed from the web site, we stepped through each section and discussed what was required and what resources were available to us. We also discussed how this prototype should be flexible for other Management Systems. There was constant reference to the "Hospital management systems" and other related articles.
March, 15,2022	Finished a rough draft prototype and set it up on the online account.
March, 17,2022	All members checked the document of the Software Project Management Plan, and we made some correction marking the corrections in red.
September 29th, 2000	Meeting at PAF KIET University: We discussed parts 4 and 5 of the Software Project Management Plan in more detail and decided to update some information in the SPMP document. The different parts of the document were divided between the team for updates.
October 3th, 2000	Finished updating the rough draft prototype and set it up on the online account. Sent all team members email with link to latest copy of the document.
October 4th, 2000	All members checked the document of the Software Project Management Plan. The mistakes were corrected. The latest version of the document is available.

Last Updated on March 20th, 2022

Software Requirements Specification

for

HOSPITAL MANAGEMENT SYSTEM



Shahmeer khan (64160)
ZainUllah (63591)
Abul Raoof(63159)
Ezhar Karim(63604)

March 20, 2022

Version	Changes Made	Date
1.0	First Pass for Review	3/12/2022
1.2	Second Pass for Review	3/15/2022
1.3	Third Pass for Review	3/18/2022
1.4	CRM Review Version	3/20/2022

Table of Contents

1. Introduction

1.1 Purpose

a) The purpose of this document is to describe the project which is called **Hospital Management System(HSM)**.The main purpose of this project is to implement a computer based Healthcare Consulting System which allows users to get immediate supervision and response on their health issues. Purpose of this project is like Hospital Management. This system will help the users or patients to find out appropriate doctors for several diseases/symptoms by giving all the required details like availability, contact information about the doctors who are specialized in the diseases given by the user. Admin will maintain a database which is fed with different symptoms and associated diseases. Thus, the system will identify certain diseases by asking the users to answer certain questions. On the basis of the diagnosis received, the system will give some suggestions of medicines to the user but with advice to consult the doctor. This system is very effective and beneficial in providing medical assistance. This helps the patients to

b) maintain a healthy life by providing them with a smooth communication system between doctors and patients.
c) This document includes detailed discussion about requirements of the project. Moreover, the SRS document explains the existing system and improvement of the existing system considering the comfort of the doctors and patients (users).

1.2 Scope In

a) Traditional healthcare systems are unable to gratify everyone's needs due to the enormous increase in population. Despite having excellent infrastructure and innovative technologies, medical services are not affordable to everyone and out of reach of common people. One of the goals of Smart Health Consulting System is to educate the users about their medical status and to keep them health-aware. The required system entitles users to self-manage some emergency situations. It emphasizes on improving the quality and experience of the user. It helps in remote monitoring of patients and reducing the cost of the medications for the users. With an expanding trend towards smart cities, an effective Smart Health Consulting system guarantees a healthy living for its citizens.

The **objectives** of this development effort are:

1. To provide existing clerks with a new environment in which to make reservations and easily take appointment of doctor.
2. To provide an avenue for customers to get their appointment in a more convenient way.
3. To regain control of the appointment and manage whole hospital to avoid scalping and overselling of appointments.
4. To implement a prototype of a scaled down version of the final system to test the solution and further develop requirements.

5. To collect statistics in a more efficient manner for future railroad development and construction.
6. To increase efficiency of railroads.

1.3 Scope Out

The following features will not be the part of this Project:

1.

1.3 Definitions, Acronyms, and Abbreviations.

HMS – Hospital Management system
APPM – AsiaPac Marketing Manager
CASE – Computer Aided Software Engineering
PP - Project Plan
SDD - Software Design Description
SRS - Software Requirement Specification
SDS – Software Design Specification
SPMP - Software Project Management Plan
GUI – Graphical User Interface
QAM – Quality Assurance Manager
PDM – Project Development Manager
PMP – Project Management Professional
TBD – To be determined
UML – Unified Modeling Language

1.4 References

- The Aga Khan University Hospital, Pakistan
  https://hospitals.aku.edu/

- China 2000
  http://www.china2thou.com
- Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill Companies, Inc., 1997.

1.5 Overview

Chapter 2 of the SRS is a brief description of the characteristics of the software to be built, its functions, its users, its constraints and its dependencies.

Chapter 3 is about specific requirements, such as functional requirements, external interface requirements, performance requirements, and also design constraints and quality characteristics.

Finally, chapter 4 includes all the supporting information, such as the Table of Contents, the Appendices, and the Index.

2. The General Description

This **Hospital Management System(HMS)** is like an online Hospital Management System provider with easy to use customizable options and interface. The application is accessible from anywhere and it is designed to reduce the manual work and improve the quality of maintaining records and other information related to doctors or patients etc. It reduces the complexity by minimizing the time- frame in adding any information related to any government hospital or private doctor.

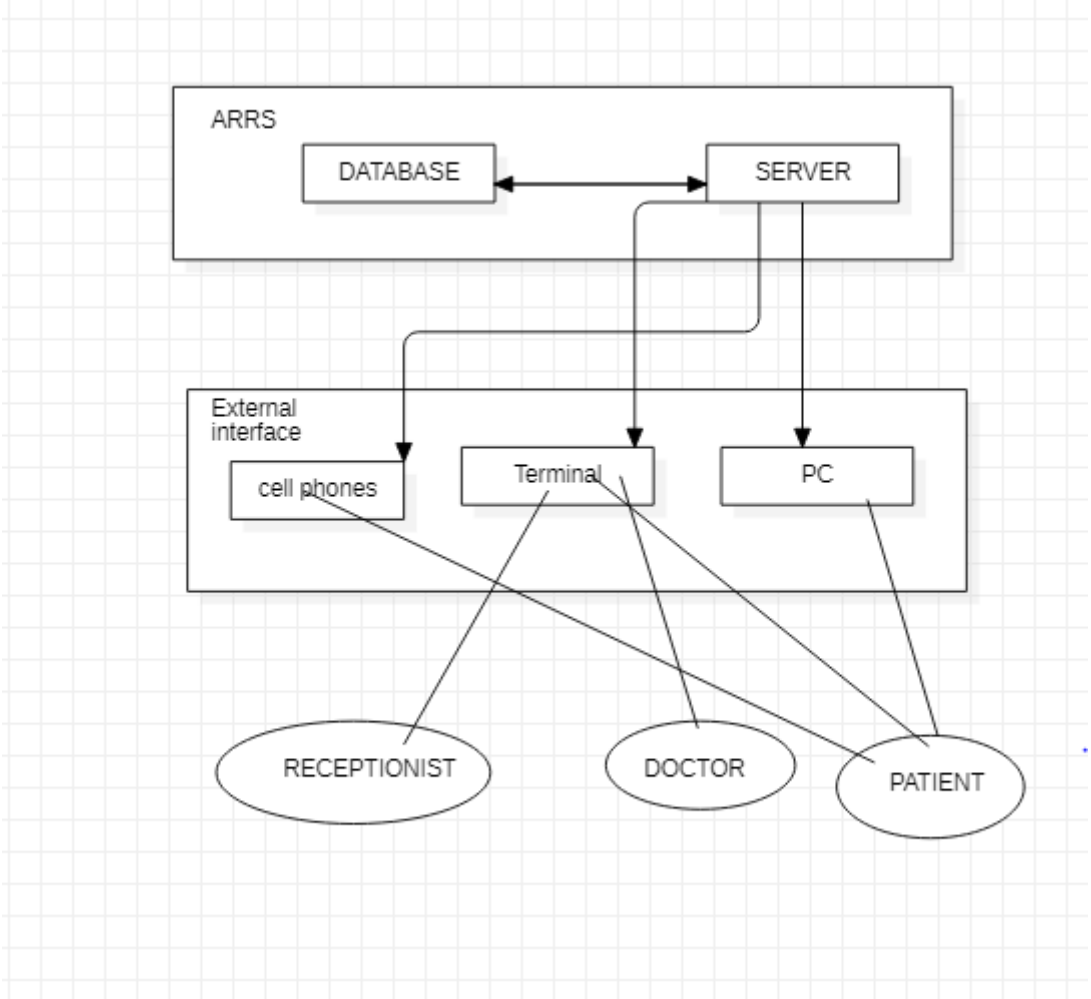
On the user's / patient's side, they will be able to find out the doctors available depending upon the city in which they dwell and according to their requirements i.e. the symptoms/disease they have. Different doctors for different specializations related to important parts of the body like heart, kidney, brain, liver, general physician, etc can be searched through this system with the doctor's availability timing and contact details. They can also check several symptoms by selecting their diseases and they can also get their diseases predicted by the system which will ask the users certain questions for accurate predictions. They can also select a disease to know the medicine specification with a recommendation to visit a doctor for further treatments. Moreover, they can also make online appointments after communicating with the respective doctor.

On the admin's side, he will be able to update the database of doctors and can also view the list of users of this application. The patient will have to enter his name before using this application and his name will be automatically added in the list of users which is accessible by admin. In addition to this, admin can also view complaints of patients and take necessary actions.

2.1 Product Perspective

The Hospital Management System diagram showing the overview of the system's modules and the relationship of the system to external interfaces is presented in Figure 2.1.

Figure 2.1 Overview/Architecture Diagram of the HMS



Functions of System Components:

Database:

- Stores data
- Provides access to data
- Updates information

Server:

- Provides access to the database
- Authenticates users
- Processes reservations
- Performs backups

External Interfaces:

Terminal

- This is the first form of the project in which admin, doctor and receptionist can go to login form after click the button
- Doctor and receptionist first register themselves after that they login with their registered name and password .after login they go to next form portal. Admin can only login by their name and pass

- This is the admin portal form and admin can add records
- After click on admin portal this form will open in this form we can add a records
- In last the data will b showed to doctors and receptionist forms

2.2 Product Functions

The system function in hospital management application will perform the following major function:-

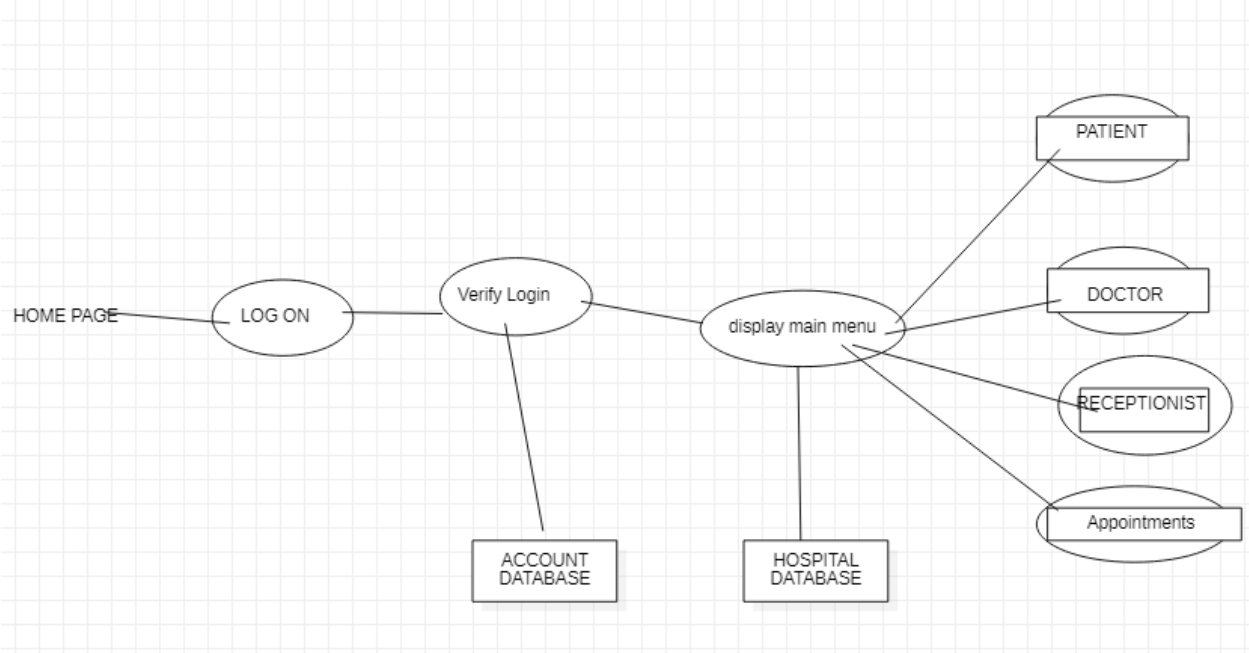
- Patient Details: – It includes inpatient and outpatient details.
- Doctors details
- Receptionist details
- Registration: doctor and receptionist first register them
- Login : then they login

- Add patient
- Appointments
- Add doctors
- Add receptionist
- Treatment details
- Add doctor assistant
- Doctor assistant details
- Show appointment
- Set medical report of each patient

2.2.1 Function Relationships

Figure 2.2 to 2.6 depict the relationships among the functions to be implemented by the system.

Figure 2.2 HMS General Function Relationship/**Higher Level Usecase** Diagram



2.2.2 Function Descriptions (Functional Requirement Listings)

2.2.2.1 Log In Function

Description: This class is made for every one which are admin, doctors, receptionist by which they can login for their diff portal

2.2.2 Module 1: Login

Description: This function allows the admin to [Make | Drop | View | Update] for updating the database of doctors. The admin can add any doctor from any city into the database along with his/her correct information such as contact number, Address, Specifications, gender etc. The admin will also provide the email address of doctors which will be used by patients to contact the doctor for the appointments.

2.2.3 Module 2: Doctor

Description: This admin module in this module admin manage doctor and patients patient details. Admin can access all modules of project. Admin can delete, update and add details of patients and doctors. Admin also add new doctors.

2.2.4 Module 3: Patient

Description: This function allows the Doctor to a show the patient ,Appointment , receptionist details and doctor treat them . If the patient does not already have a appointment, then a new appointment is created. If the patient appointment already has a previous reservation, a new reservation is added to the list of current reservations. And last doctor give the tests and medicine after treatment

2.2.5 Module 4: Receptionist (Appointment)

Description: Its is a receptionist in which we have method of add, delete, show, and update receptionist and all the date is store in SQL server database , receptionist also have duty to set the appointments by which doctor can treat the patient according to their given appointments list and also maintain the medical reports of each patient.

2.2.6 Module 5: Doctor Schedule

Description: This function

2.3 User Characteristics

The main users of the system will be the passengers buyin, the travel agents that process reservations for passengers, and the CRM administration that access the reports generated by the system. The users are not required to have knowledge in the computer field. The graphical interface provides an easy way of using the HMS system with minimum of training.

2.4 General Constraints

The constraints for the project are:

- The functional prototype should be available after 30 days upon the arrival of the management team to China. This may prove to be a serious time constraint on the development of a successful prototype.
- Communication with the Chinese team members may prove to be difficult since some Chinese developers do not speak English and the management team does not speak Chinese. Even with the presence of a translator, communication may be difficult. Absence of the translator may severely affect project development.
- Team members are restricted from bringing their own equipment, and insufficient equipment supply may hinder project development.
- Team members are restricted to bringing only the analysts of the team to China. This might affect the project development if more people are needed or the required skills are not available.
- The majority of the Chinese population does not have or have a limited access to the Internet.

2.5 Assumptions and Dependencies or Business Logic

The assumptions for the project are:

- Ten trains transport the passengers between three cities known as Guangzhou, Shanghai and Nanjing. These trains originate only in cities Guangzhou and Shanghai, and they make a stop at Nanjing before arriving to their destination.

- There are five classes of tickets as listed below
  - Sleeping (soft) - compartment style coaches - 4 passenger per compartment
  - Sleeping (hard) - compartment style coaches - 6 passenger per compartment

- Reservation can be made up to one month before a particular trip.
- Seats are assigned during reservation.
- Phone reservation involves tickets being purchased within 24 hours after making the reservation. Otherwise, the reservation will be cancelled.
- No reservations can be made 48 hours prior to the trip. Rather, it will be done on a first come first serve basis from that point on.
- Passenger lists will be provided for conductors at each stop.
- The expected reservations during test period may amount to approximately 25,000 per day. This volume varies by hour, day, and season.

- Chinese Ministry will provide us with information about identification process used in China, so that it can be applied to the reservation system and scalping of tickets is avoided.
- Network connection will always remain established.

<ADD OOAD REPORT DIAGRAMS HERE>

3. Specific Requirements

This section of the SRS contains design requirements for the Hospital Management System.

3.1 Functional Requirements

3.1.1 Log In Function

a) ***Description:*** This function ensures that only authorized users gain access to the Hospital databases. An authorized user is a user who has an account on the system. Users include Admin, Doctor officials, Receptionist officials and Doctor Assistant. The user must type a valid username and password to gain access.

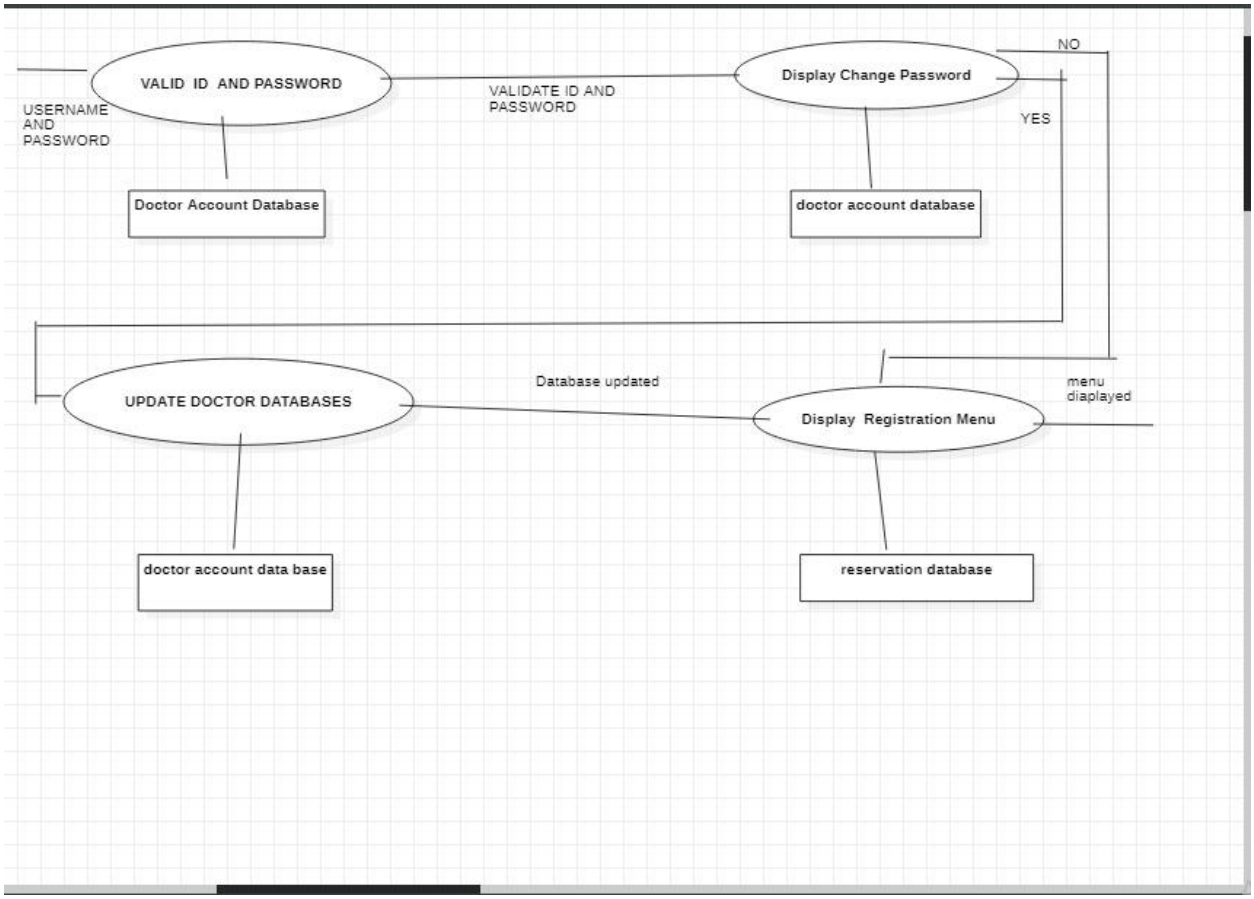
b) ***Usage Scenario/ Use case Description/ Specification:***

Description	Allows access to online HMS
Inputs	Username, password
Source	1. User inputs username and password 2. Press Login Button
Alternate case	
Outputs	Successful login; unsuccessful login
Destination	None
Precondition	Authorized User
Post Condition	No change to Passenger Accounts Database
Side Effects	Failures and successful logins are sent to Reservation Database

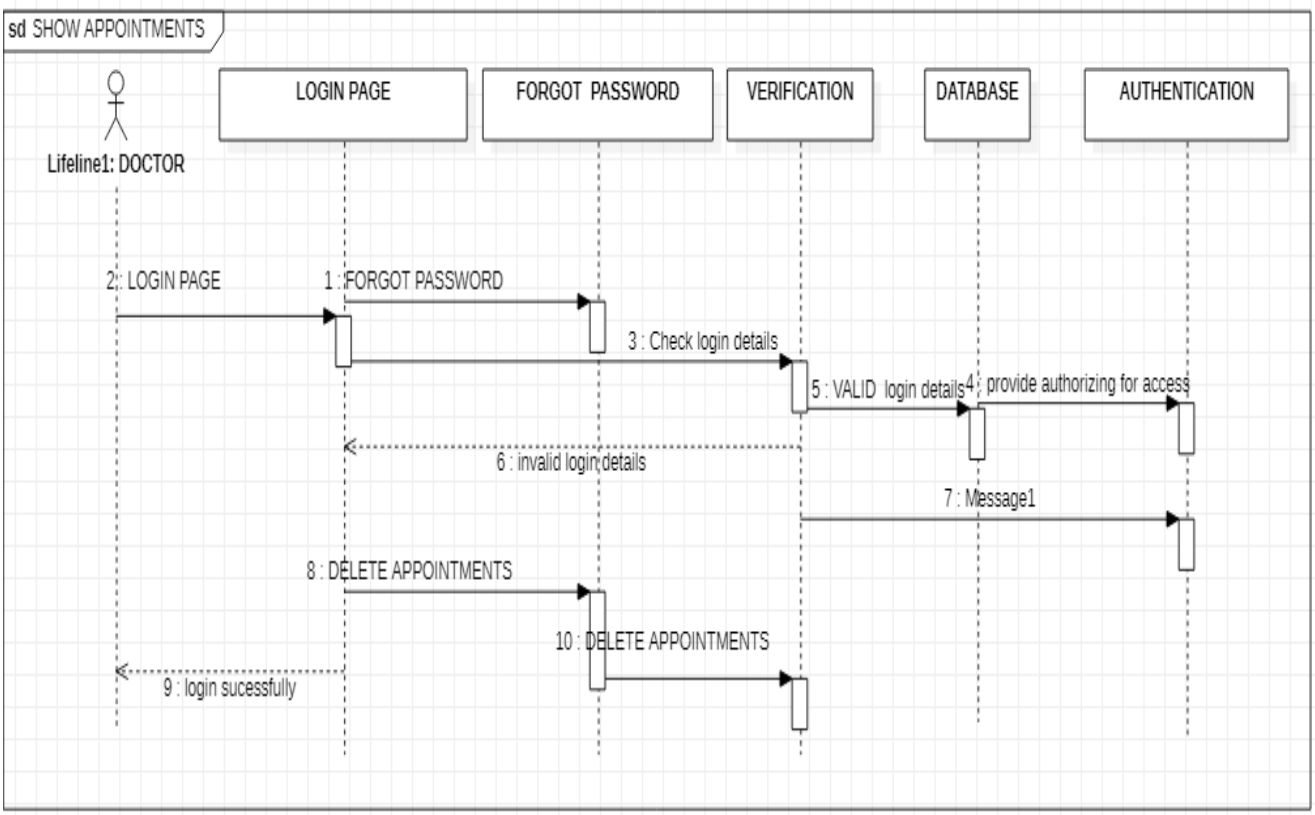
c) Detailed Use case Diagram for Login: optional

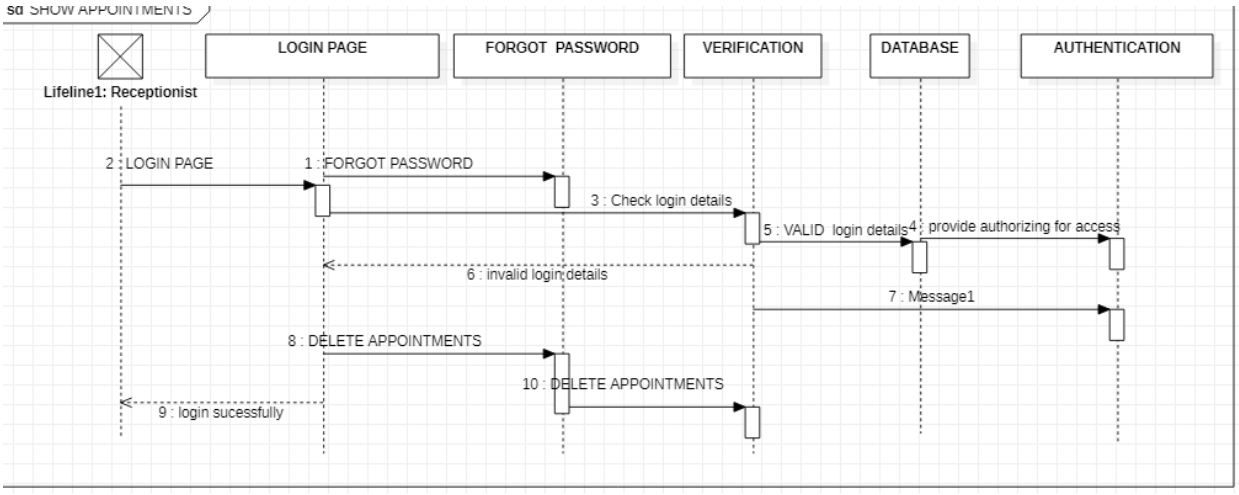
d) Use case Realization for Login: optional

e) Flow of Event or Data Flow Diagram for Login: optional



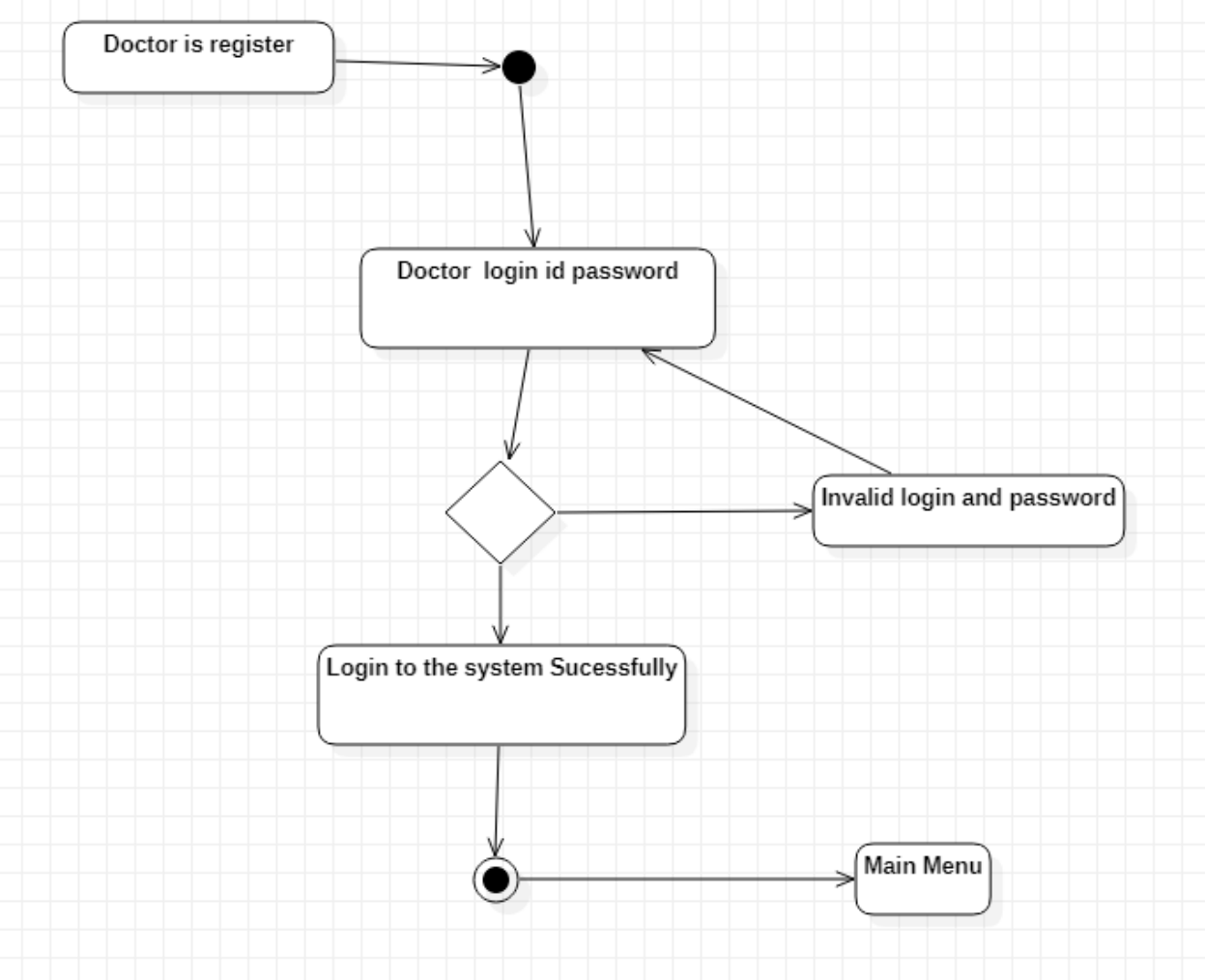
f) ***Sequence Diagram for Login: optional***

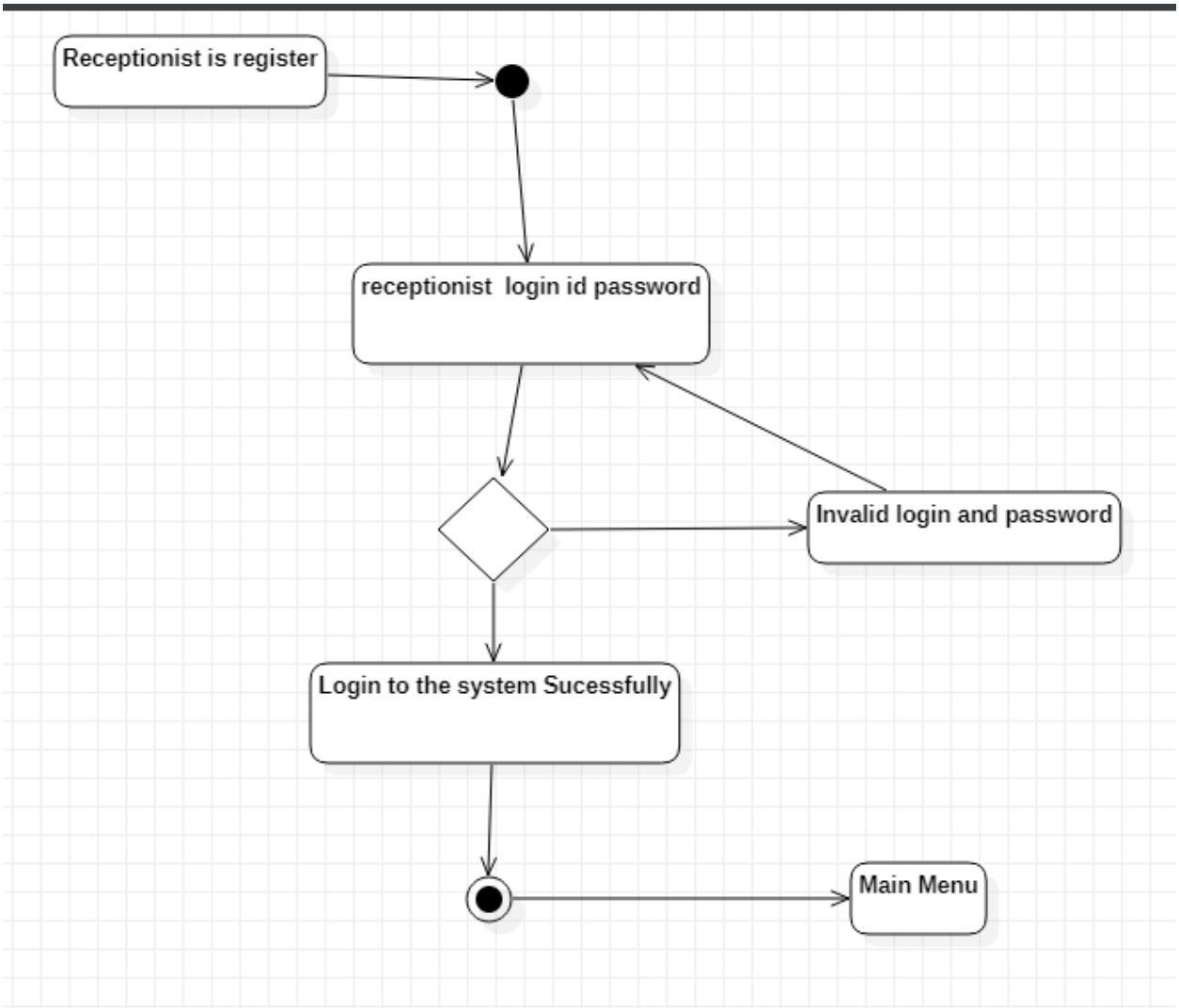




g) Collaboration Diagram for Login: optional

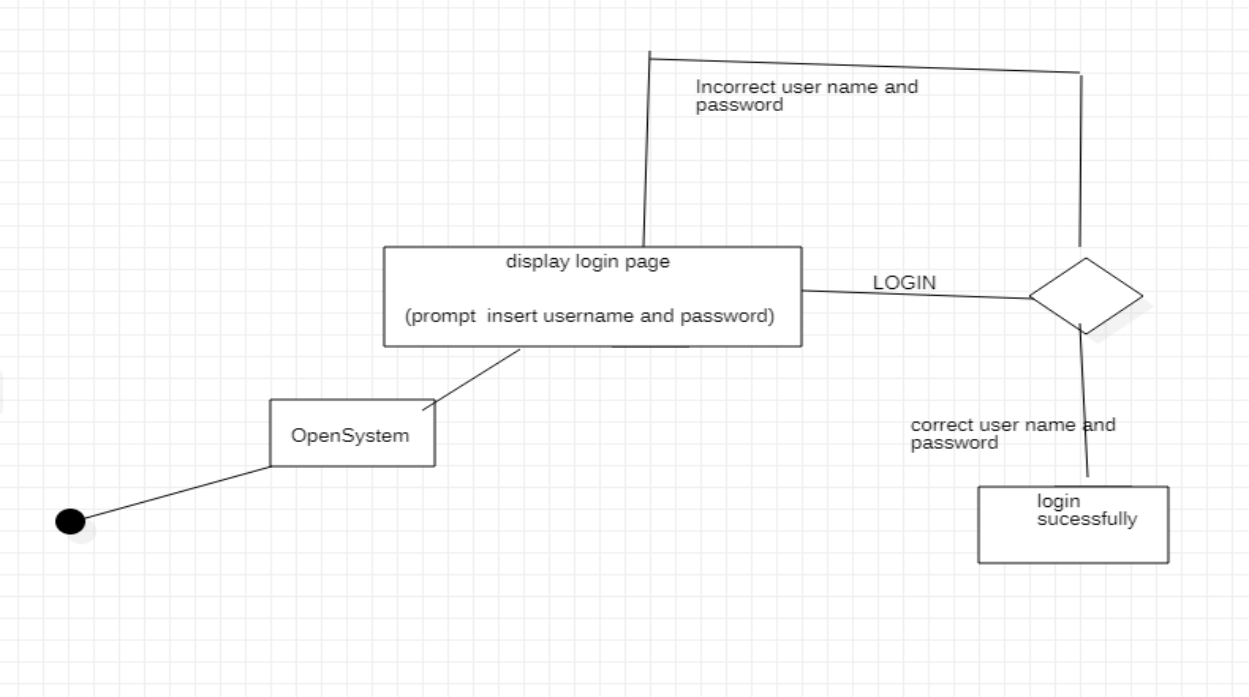
h) Activity Diagram for Login: optional


Doctor is register
Doctor login id password
Invalid login and password
Login to the system Sucessfully
Main Menu



i) Class Diagram for Login: optional

j) State Chart Diagram for Login: optional



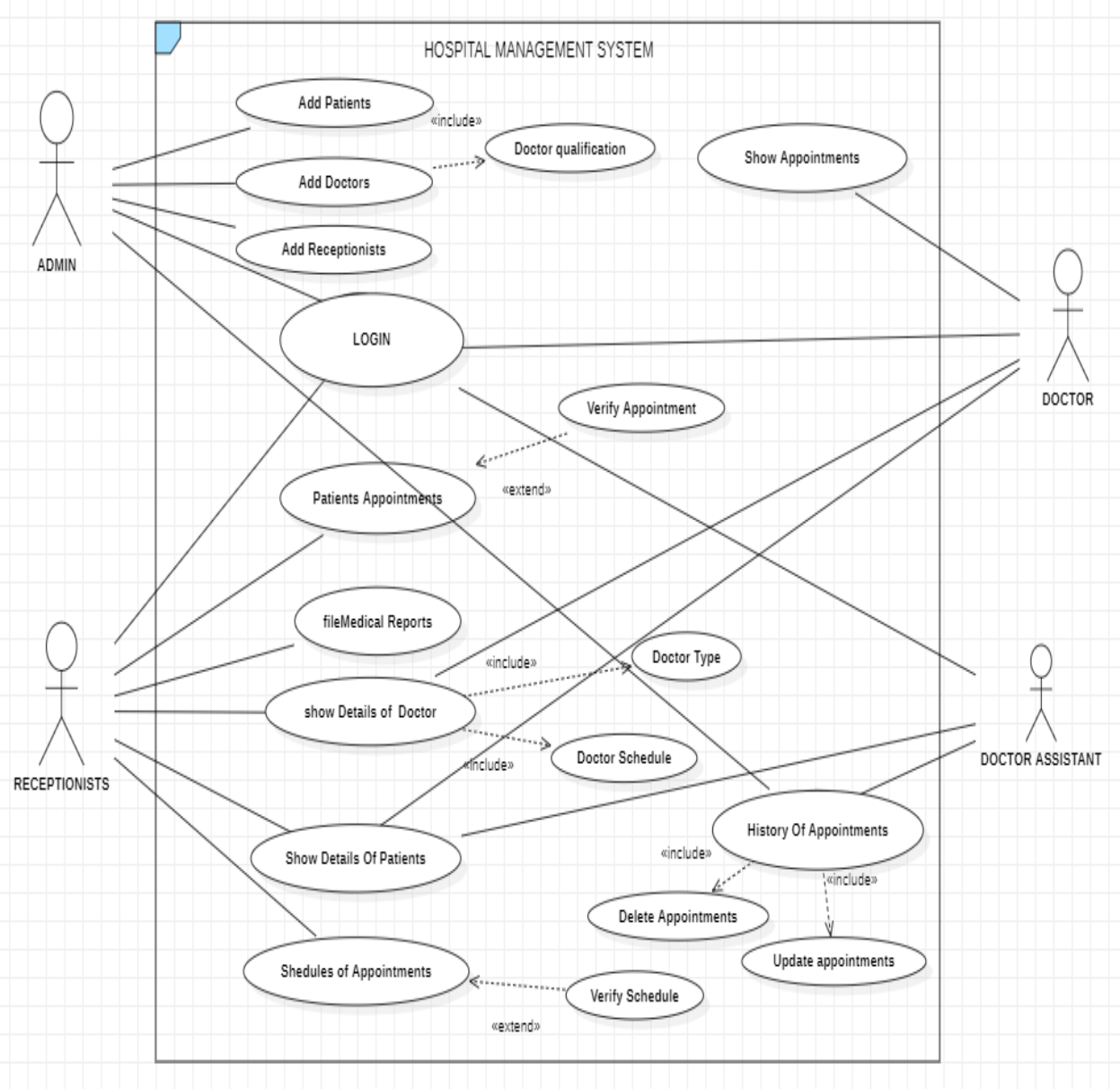
3.1.2 Module 2 complete CRUD Make a Reservation Function

a) ***Description:*** This function allows the user to [make | drop | view | update] a reservation for a particular patient , doctor, receptions, patient appointments on a particular date . If the user does not already have a reservation, then a new reservation is created. If the user already has a previous reservation, a new reservation is added to the list of current reservations, and the passenger account balance gets updated.

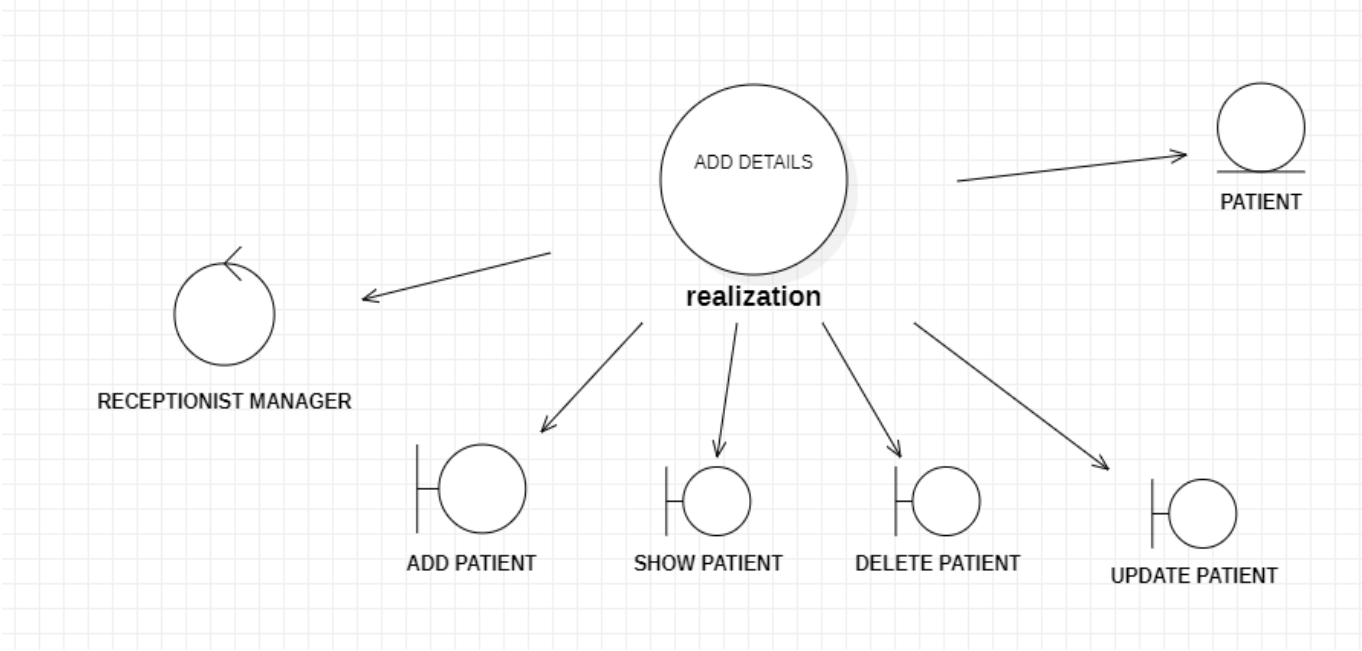
b) ***Usage Scenario/ Use case Description/ Specification:***

Description	[make | drop | view | update] a reservation to the user's account
Inputs	Doctor name, patient name, receptionist name
Source	1. User inputs from city, to city, seat type, travel date, return date and time 2. Press Button …
Alternate Case	
Outputs	Added | Deleted | Viewed | Modified reservation
Destination	Computer screen Reservation database Passenger Account database
Precondition	Valid information; train route and tickets available; user does not have another reservation at the same time
Post Condition	Reservation added to passenger account
Side Effects	User's current reservations adjusted Balance due adjusted

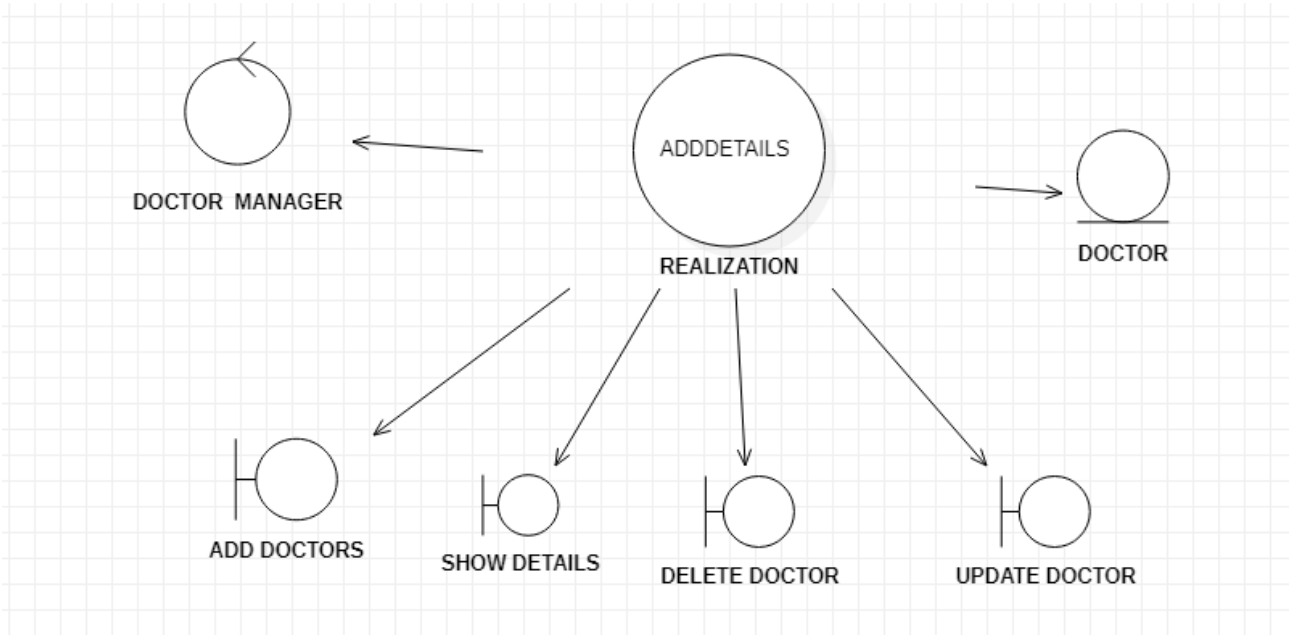
c) Use case Diagram:



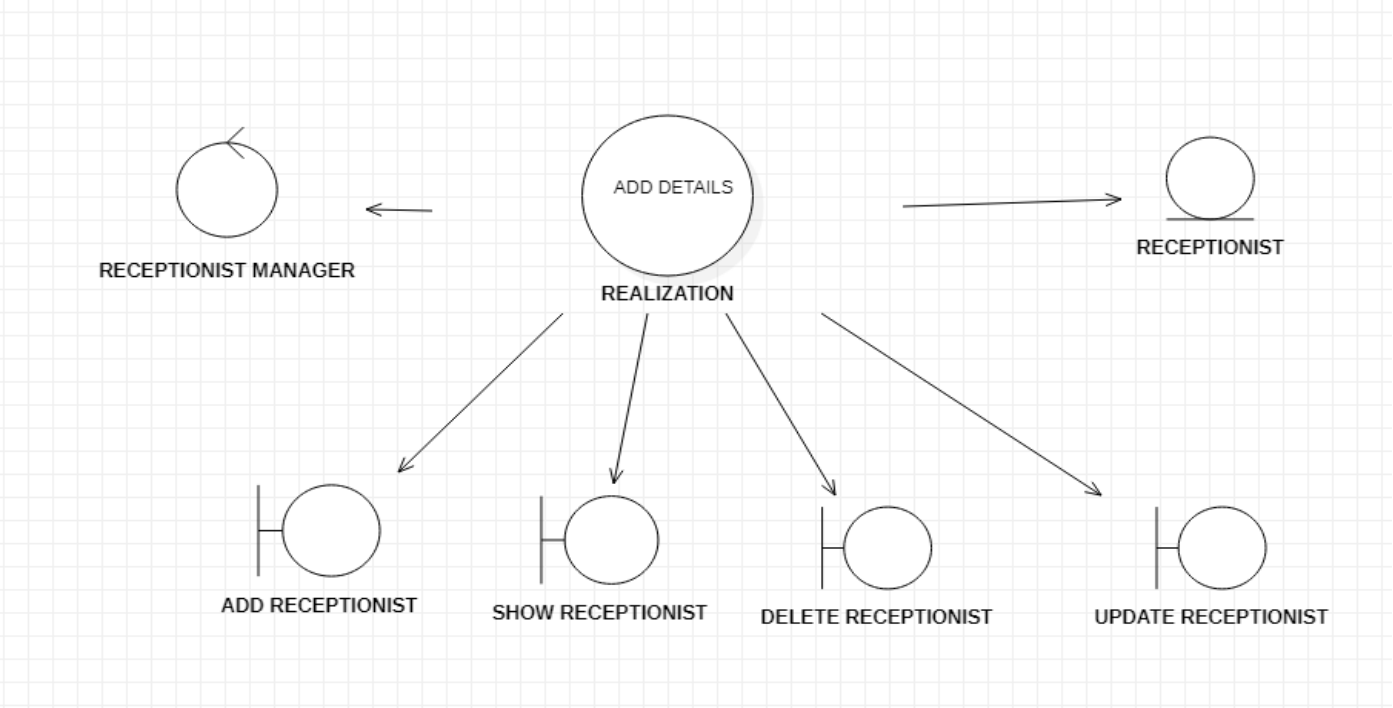
d) Use case Realization:



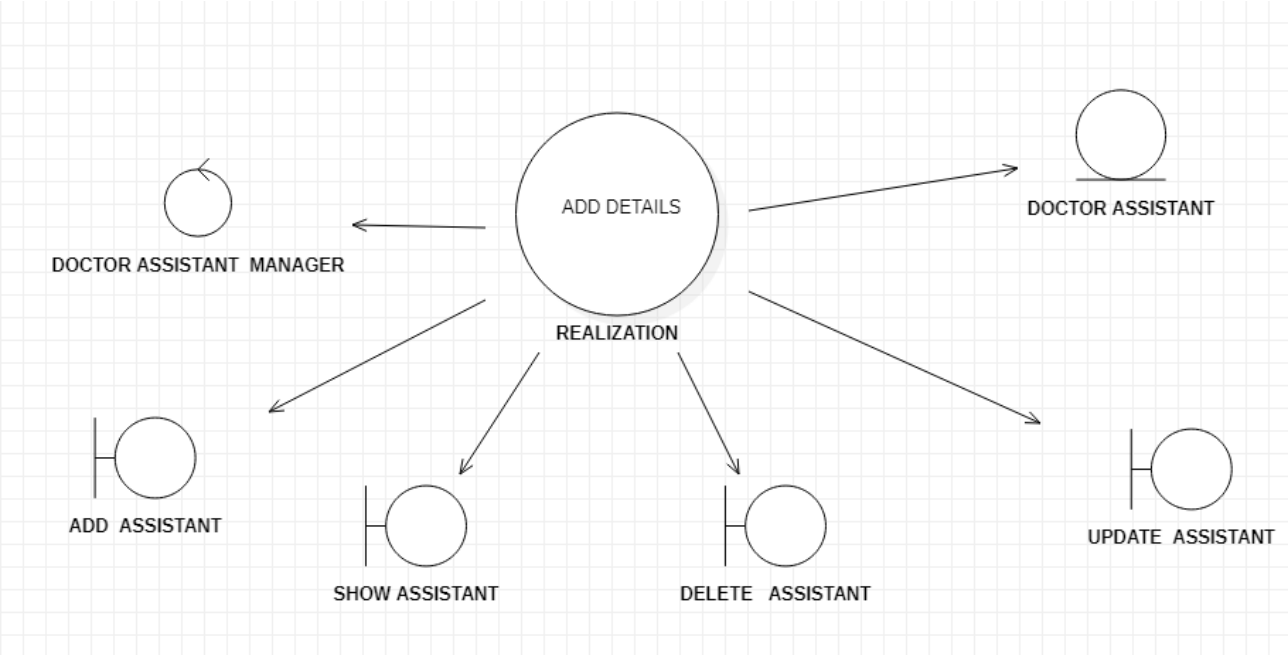
Doctor:



ADD RECEPTIONIST:

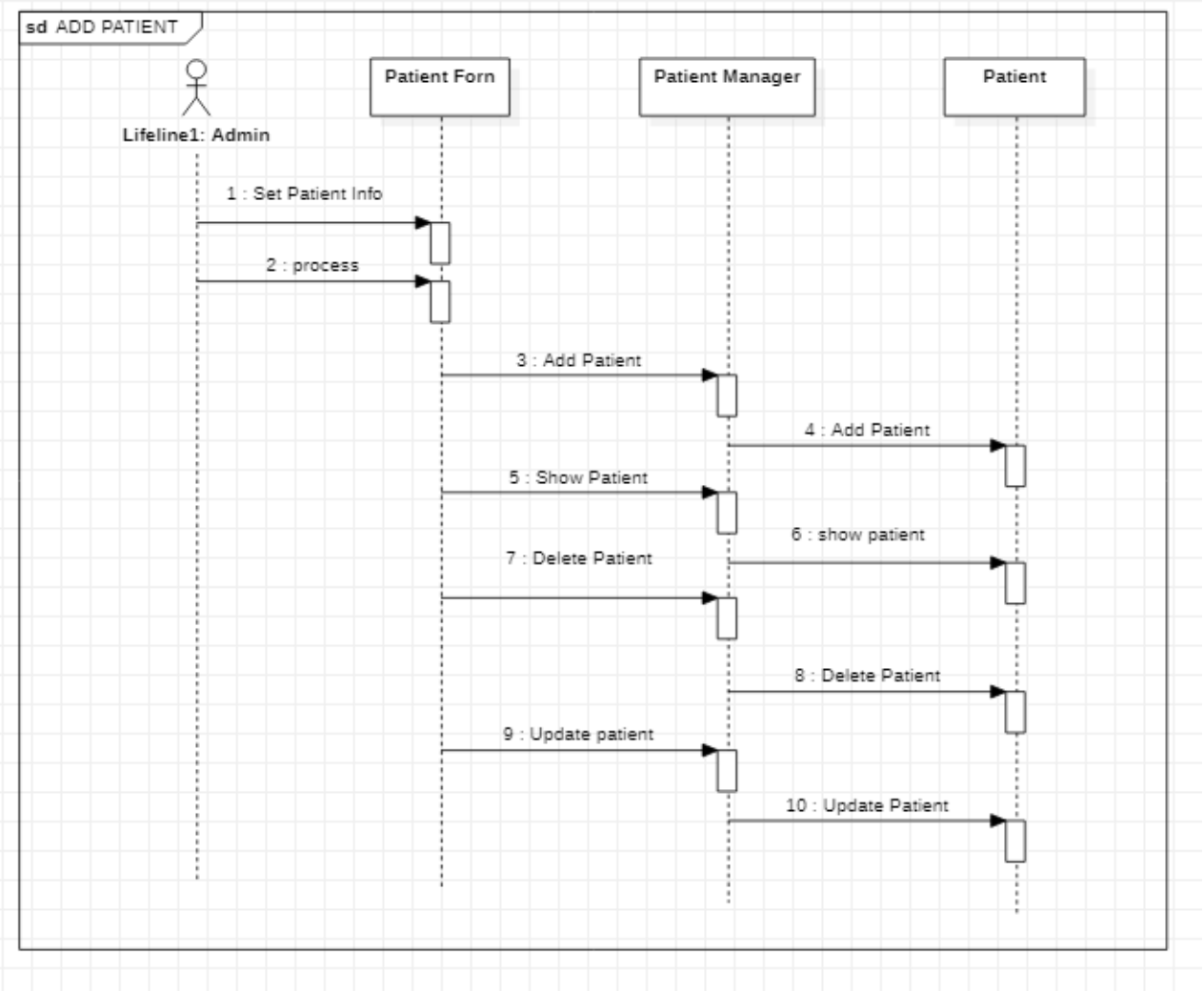


ADD DOCTOR ASSISTANT:

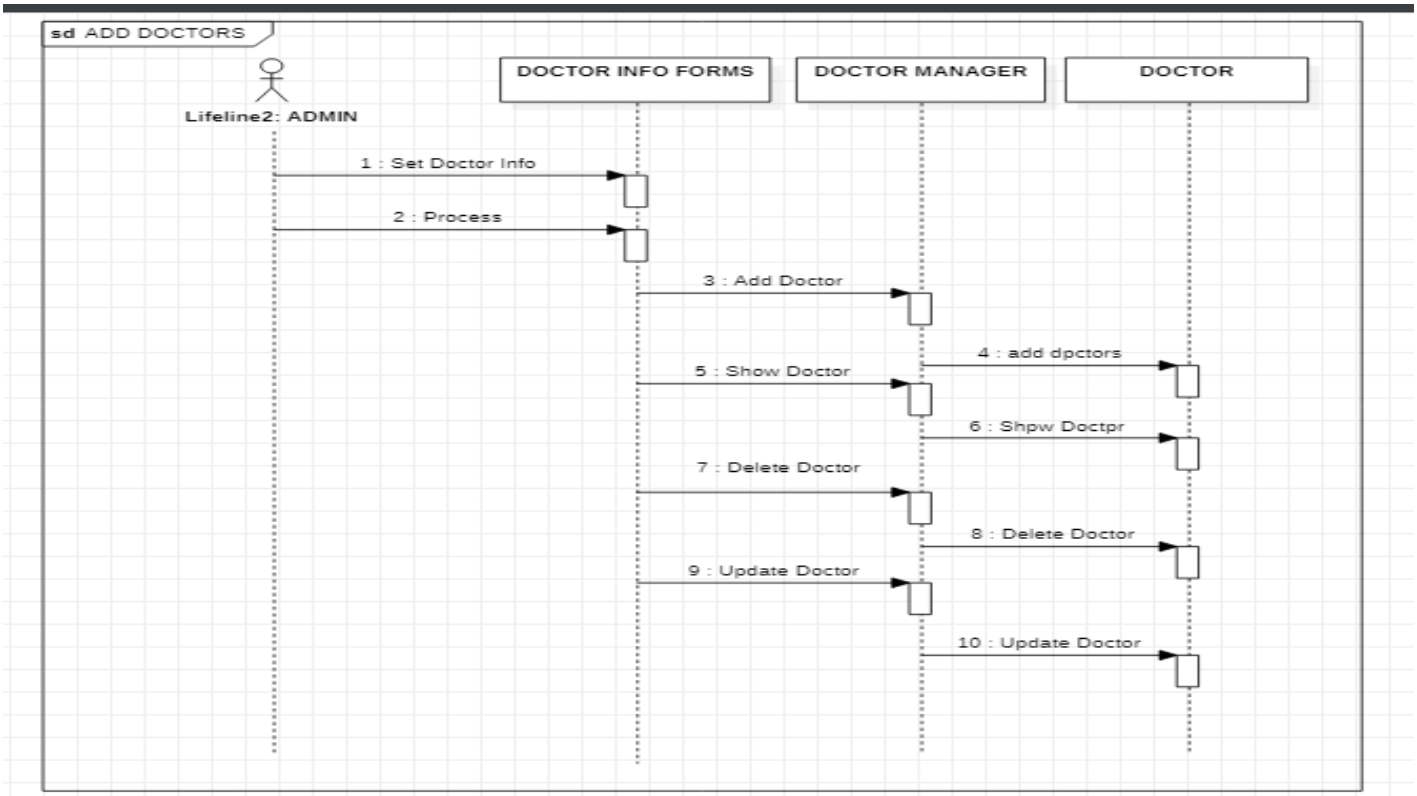


e) Flow of Event or Data Flow Diagram:

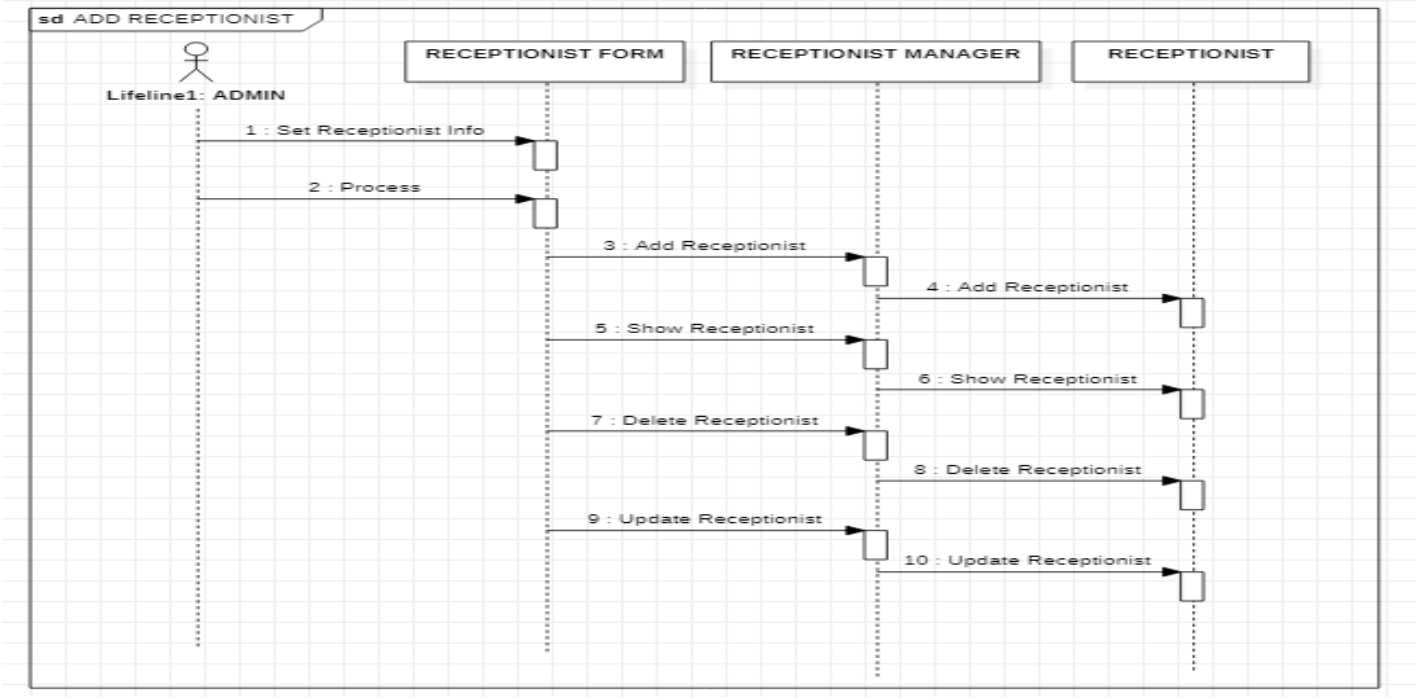
f) ***Sequence Diagram:***

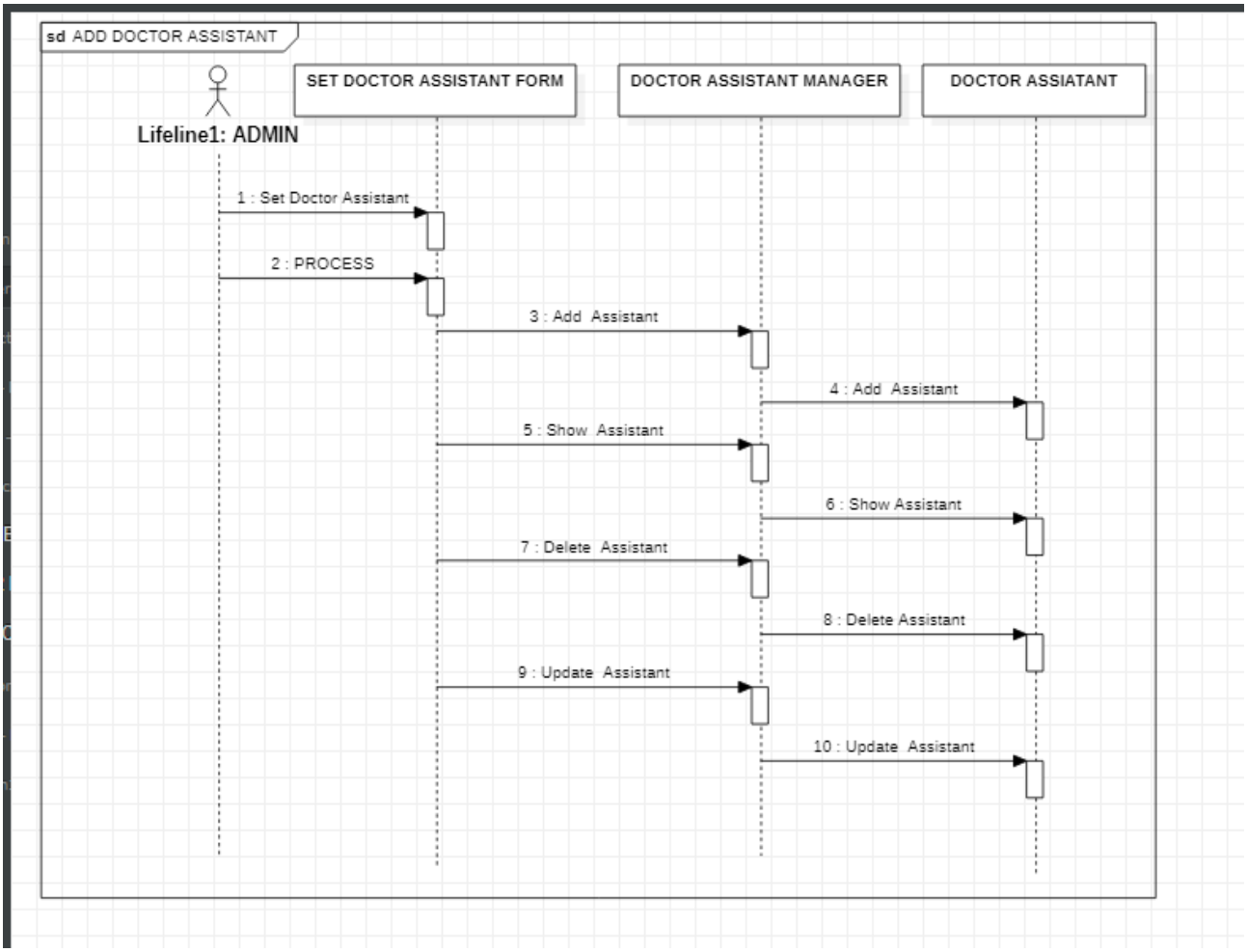

sd ADD PATIENT
Lifeline1: Admin
Patient Forn
Patient Manager
Patient
1 : Set Patient Info
2 : process
3 : Add Patient
4 : Add Patient
5 : Show Patient
6 : show patient
7 : Delete Patient
8 : Delete Patient
9 : Update patient
10 : Update Patient

ADD DOCTOR:



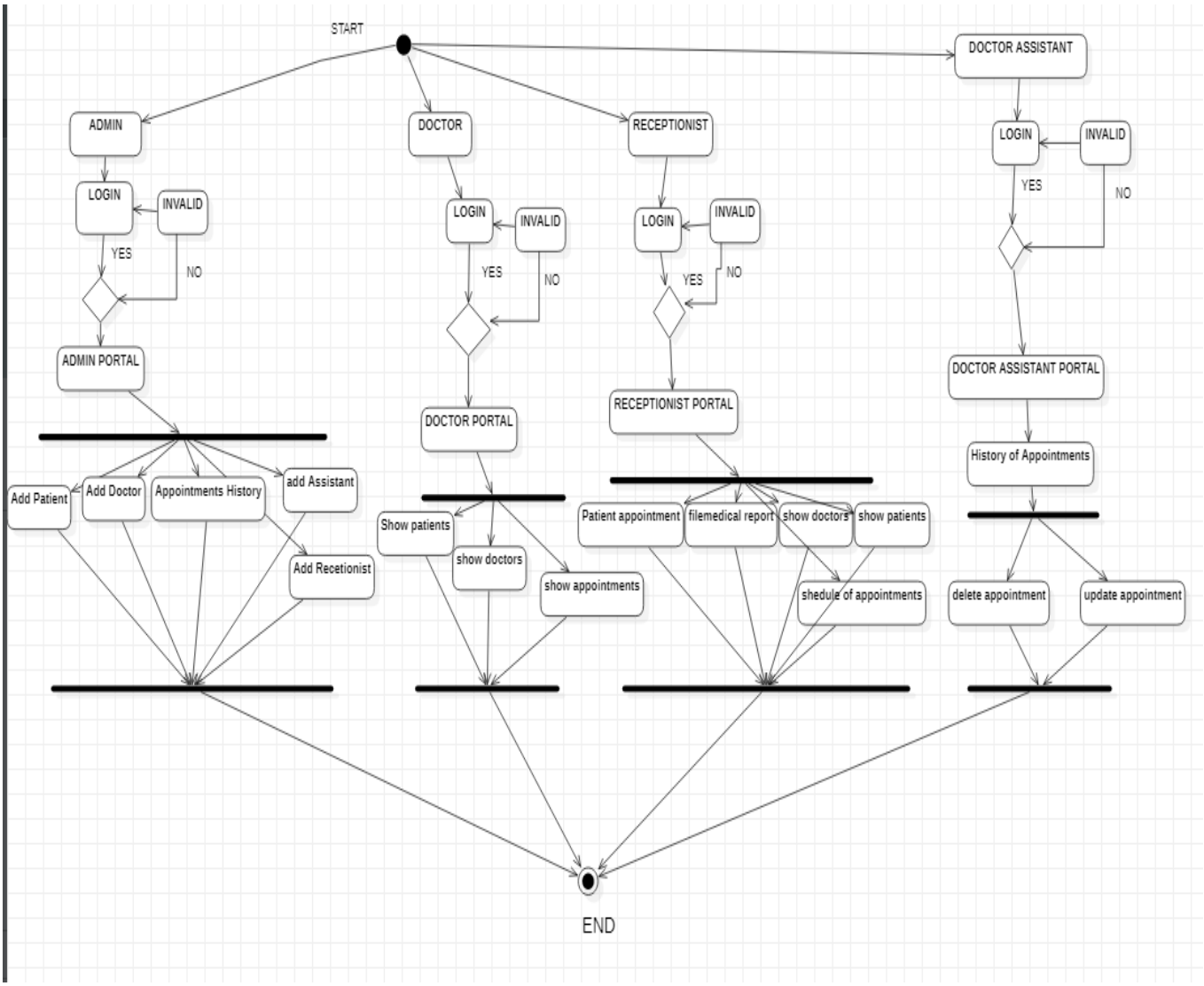
ADD RECEPTIONIST:



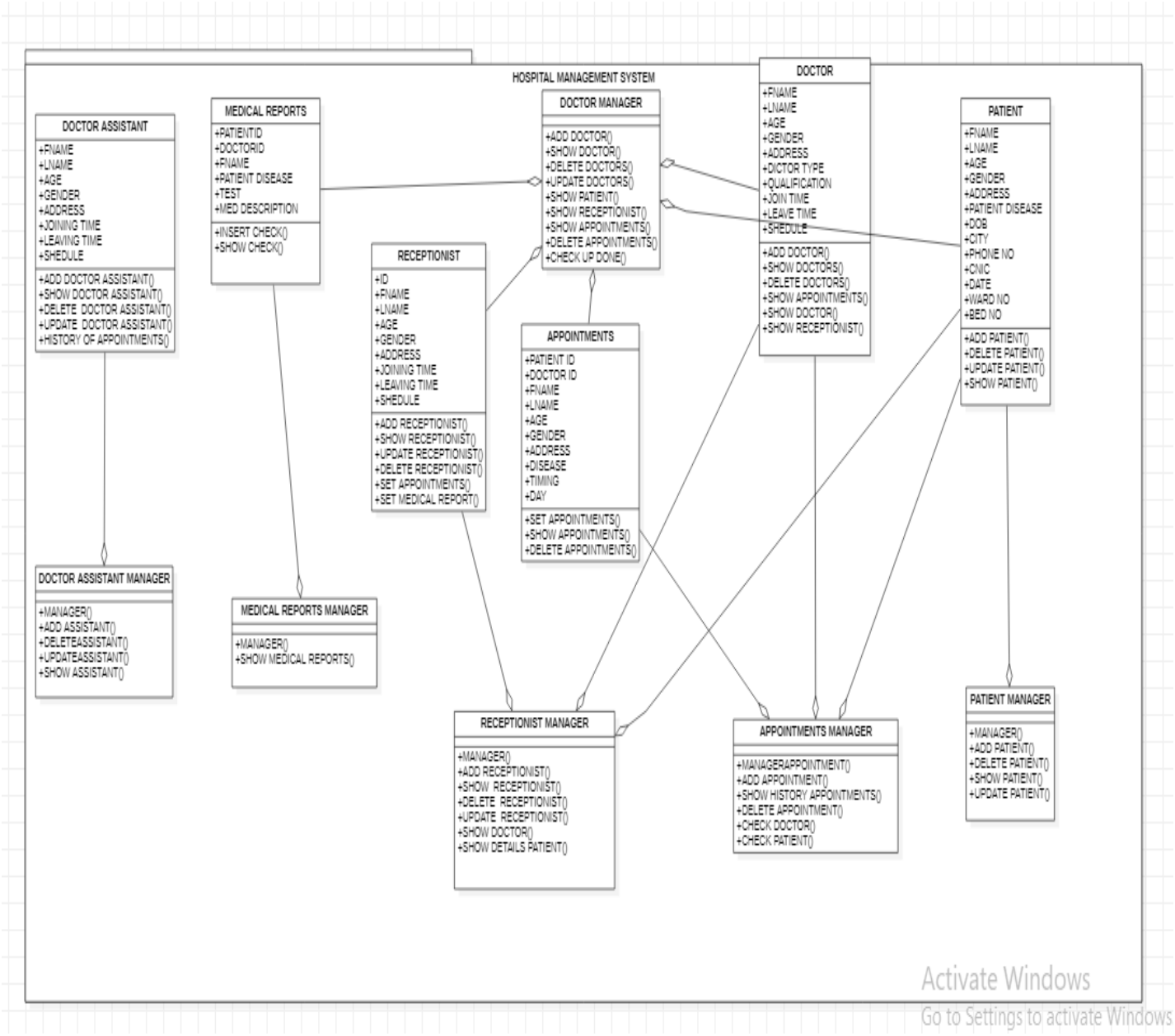


g) Collaboration Diagram:

h) Activity Diagram:



i) Class Diagram:



j) State Chart Diagram:

3.1.3 Module 3 complete CRUD Make a Reservation Function

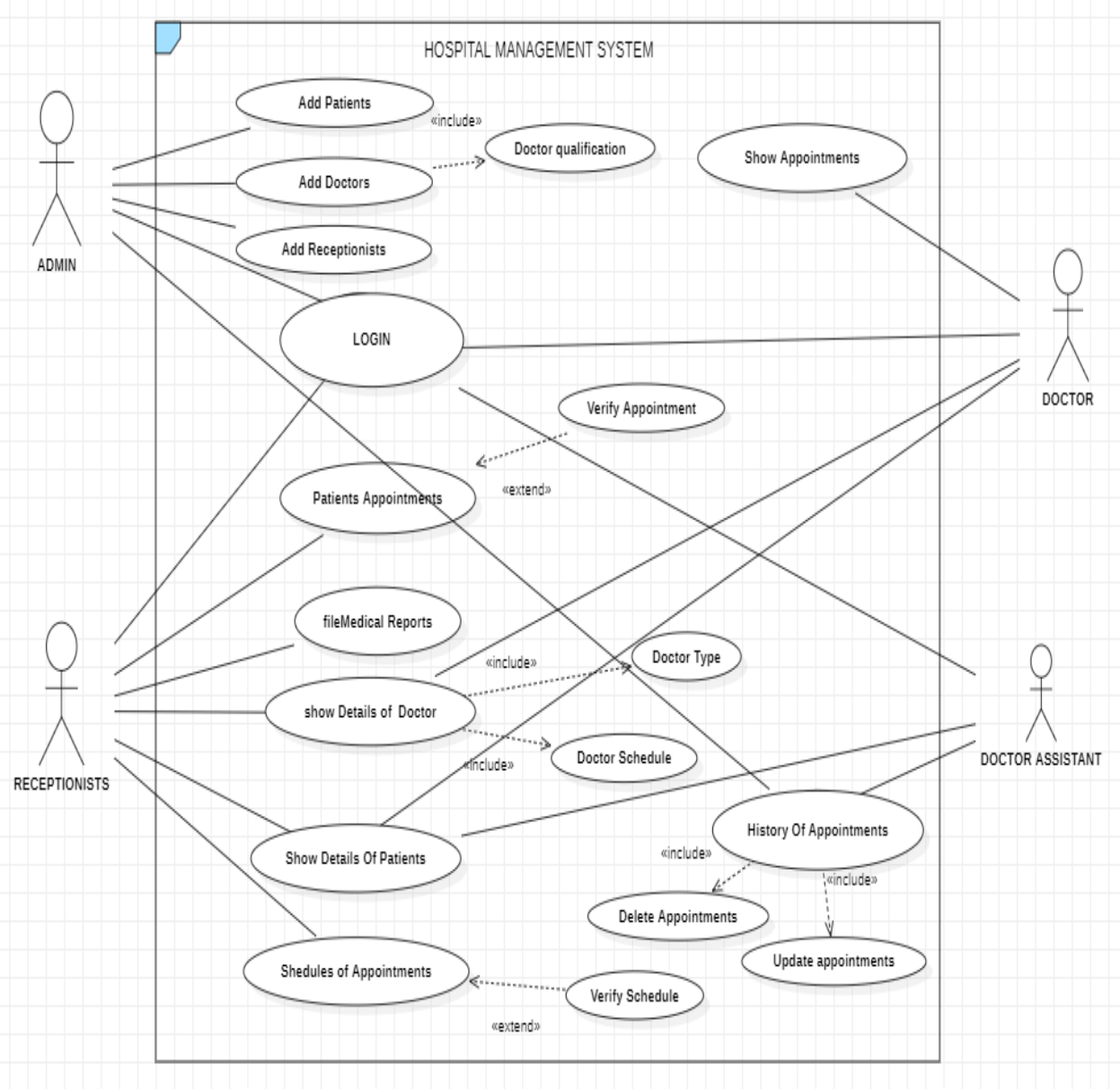
3.1.3 Module 2 complete CRUD Make a Doctor Info

k) ***Description:*** This function allows the Doctor to a show the patient ,Appointment , receptionist details and doctor treat them . If the patient does not already have a appointment, then a new appointment is created. If the patient appointment already has a previous reservation, a new reservation is added to the list of current reservations. And last doctor give the tests and medicine after treatment

k) ***Usage Scenario/ Use case Description/ Specification:***

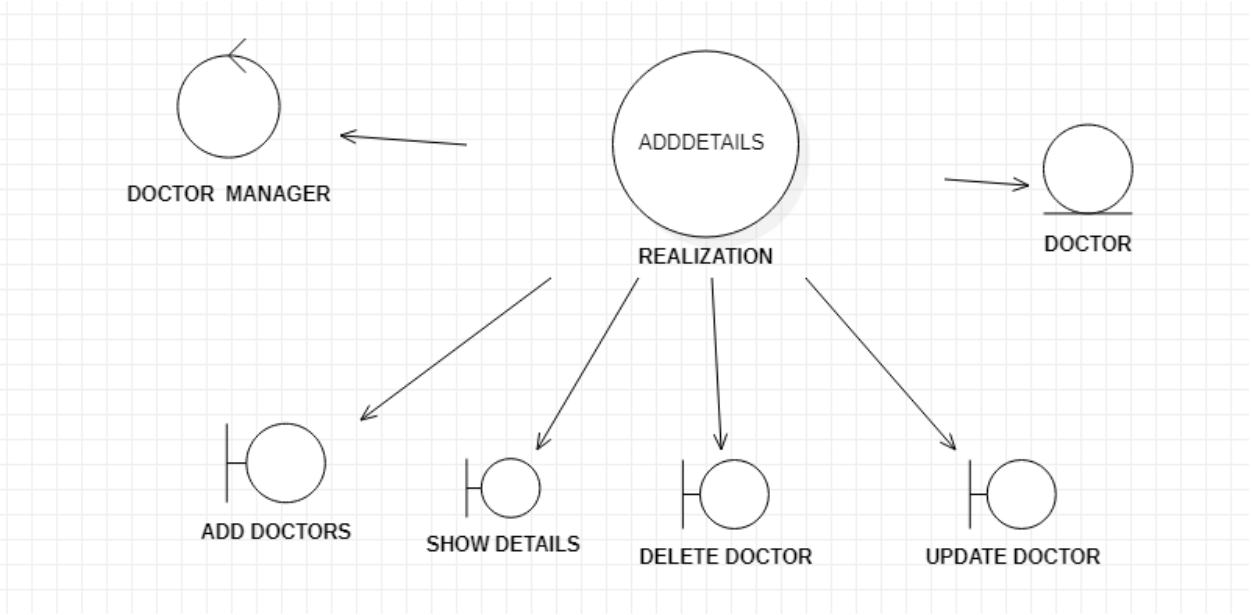
Description	Allows access to hospital management system
Inputs	Username, password
Source	3. User inputs username and password 4. Press Login Button
Alternate case	
Outputs	Successful login; unsuccessful login
Destination	None
Precondition	Authorized User, check appointment, patients
Post Condition	No change to hospital Database
Side Effects	Failures and successful logins are sent to hospital Database

l) Detailed Use case Diagram for Doctor: optional

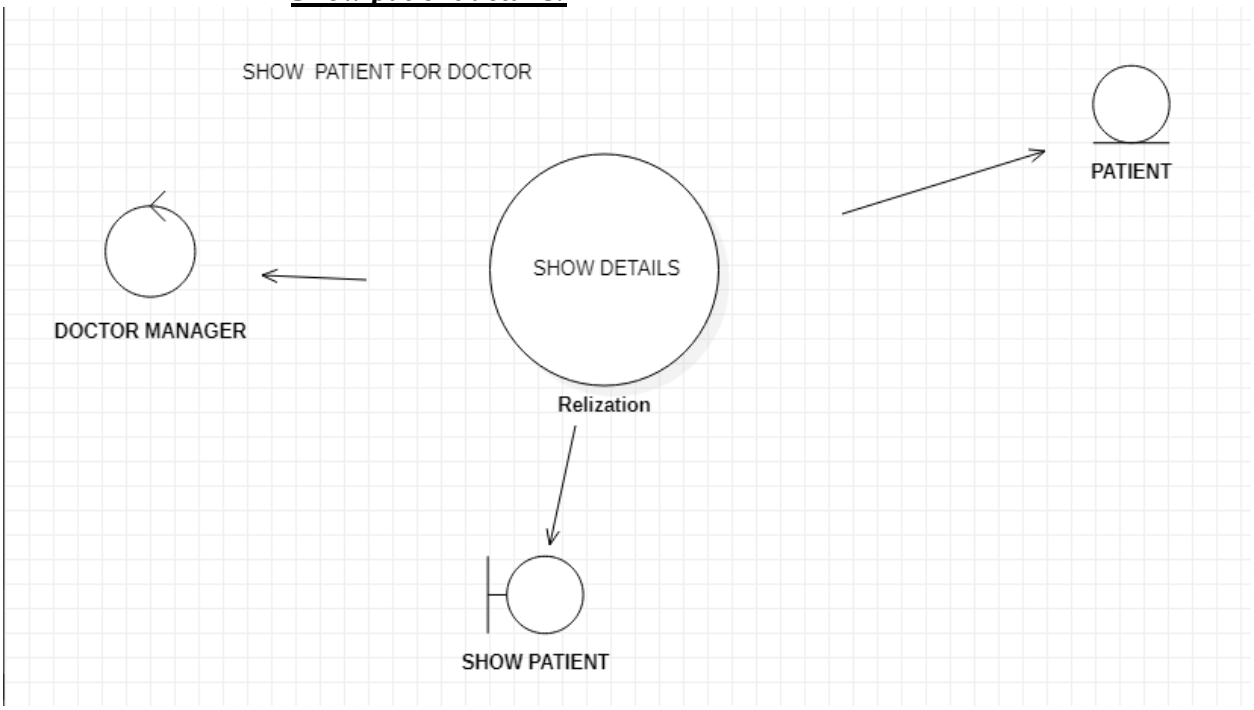


m) Use case Realization for Doctor: optional

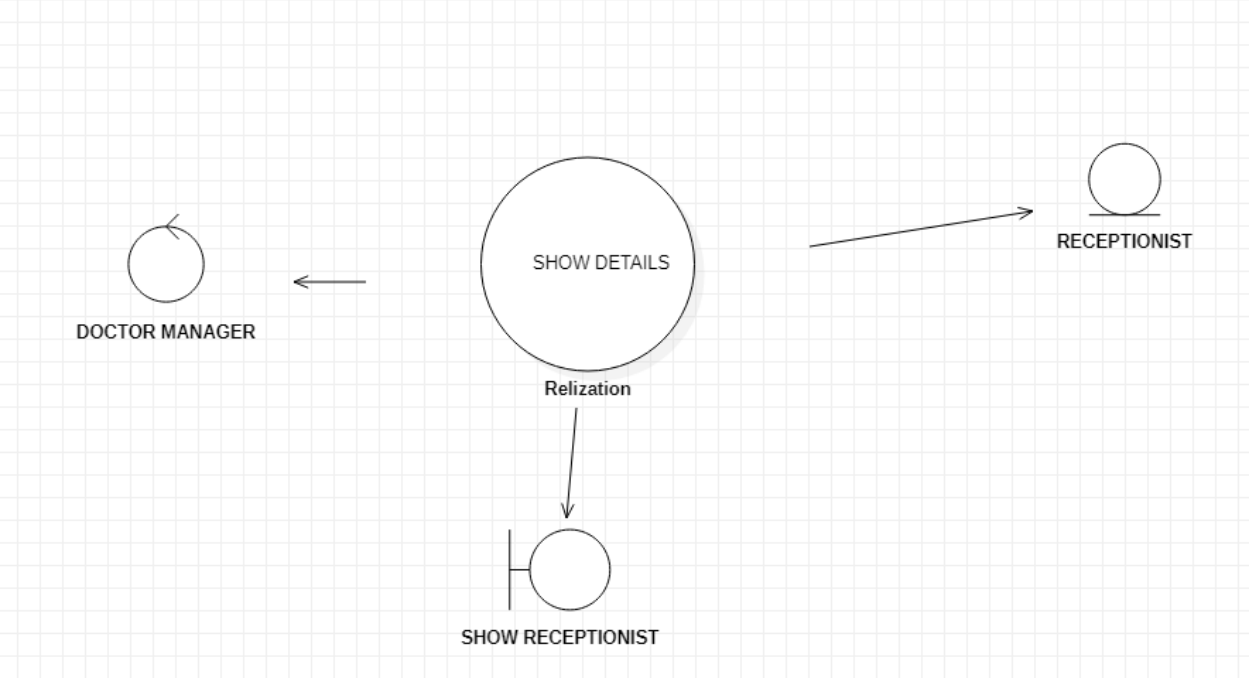
Admin add the doctor details:



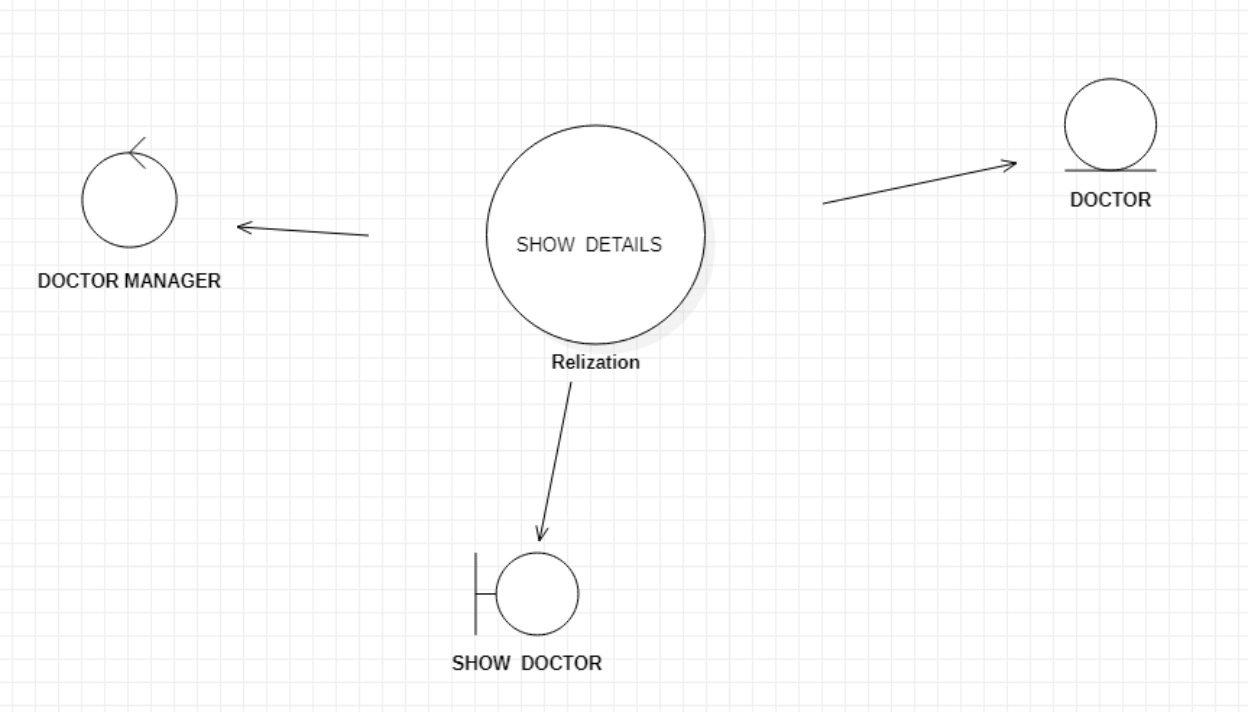
Show patient details:



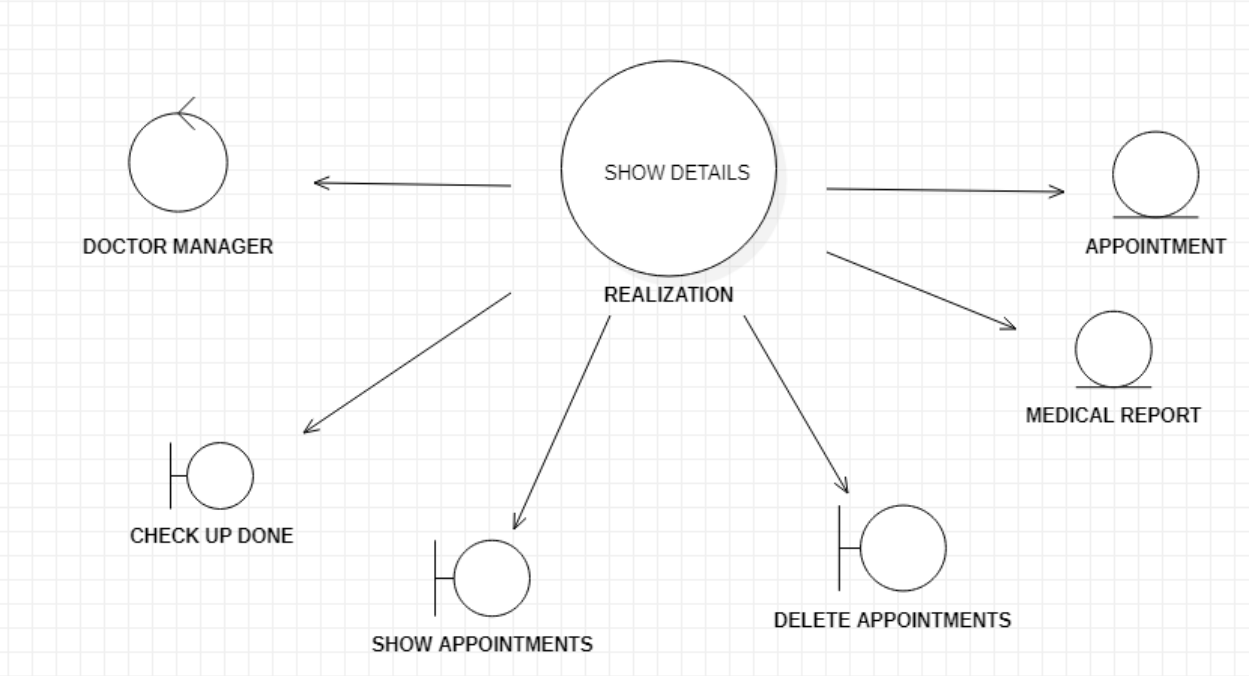
Show Receptionist Details:



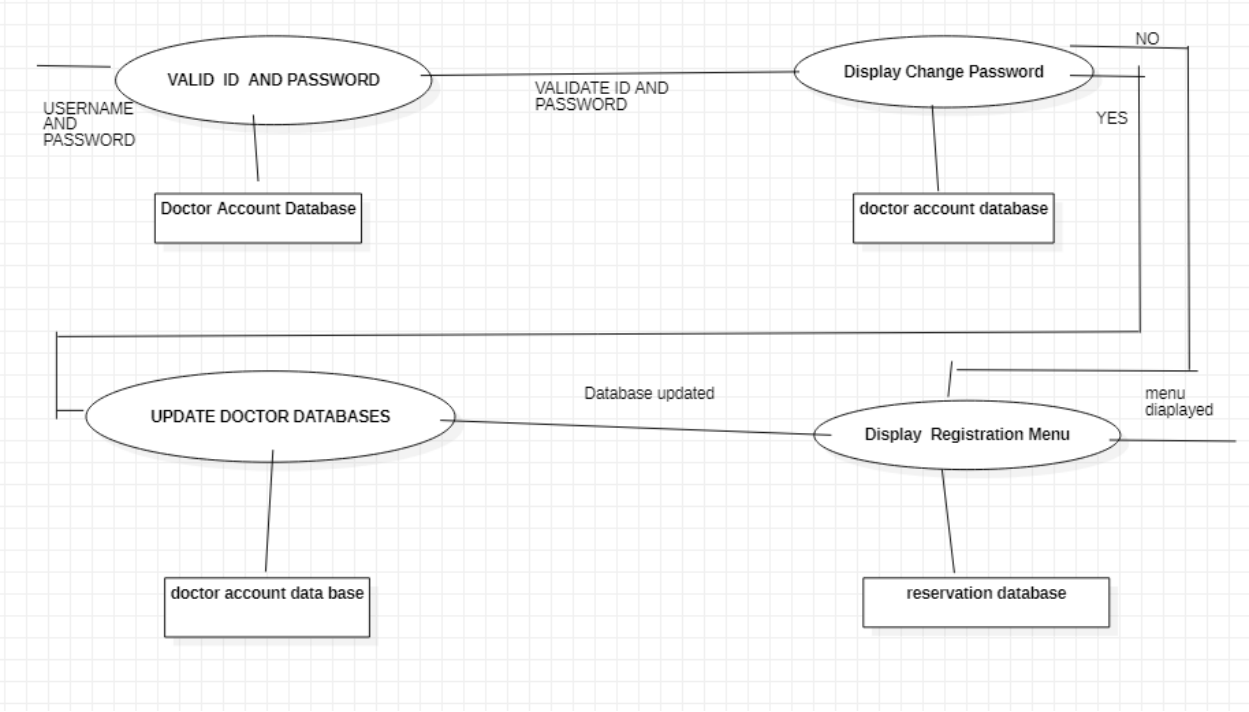
Show Doctor Details:



Show Appointments Details:



n) Flow of Event or Data Flow Diagram for Doctor: optional



Flow Of Events Of Doctor:

Precondition:

The sub flows is to maintain the all records & information which is related to the hospital.

It can also maintain all appointments of patients .It must execute before this use case begins. And the details of patient will also show to

Main flow:

This use case begins with the doctor login onto registration system and enter his/her password the system verifies that password is valid and prompts the doctor to see the patient and their appointments details .

Sub Flows:

i. Show details of patient

The doctor can see the patient data. Which Includes name, father name and treat them

ii. Appointments

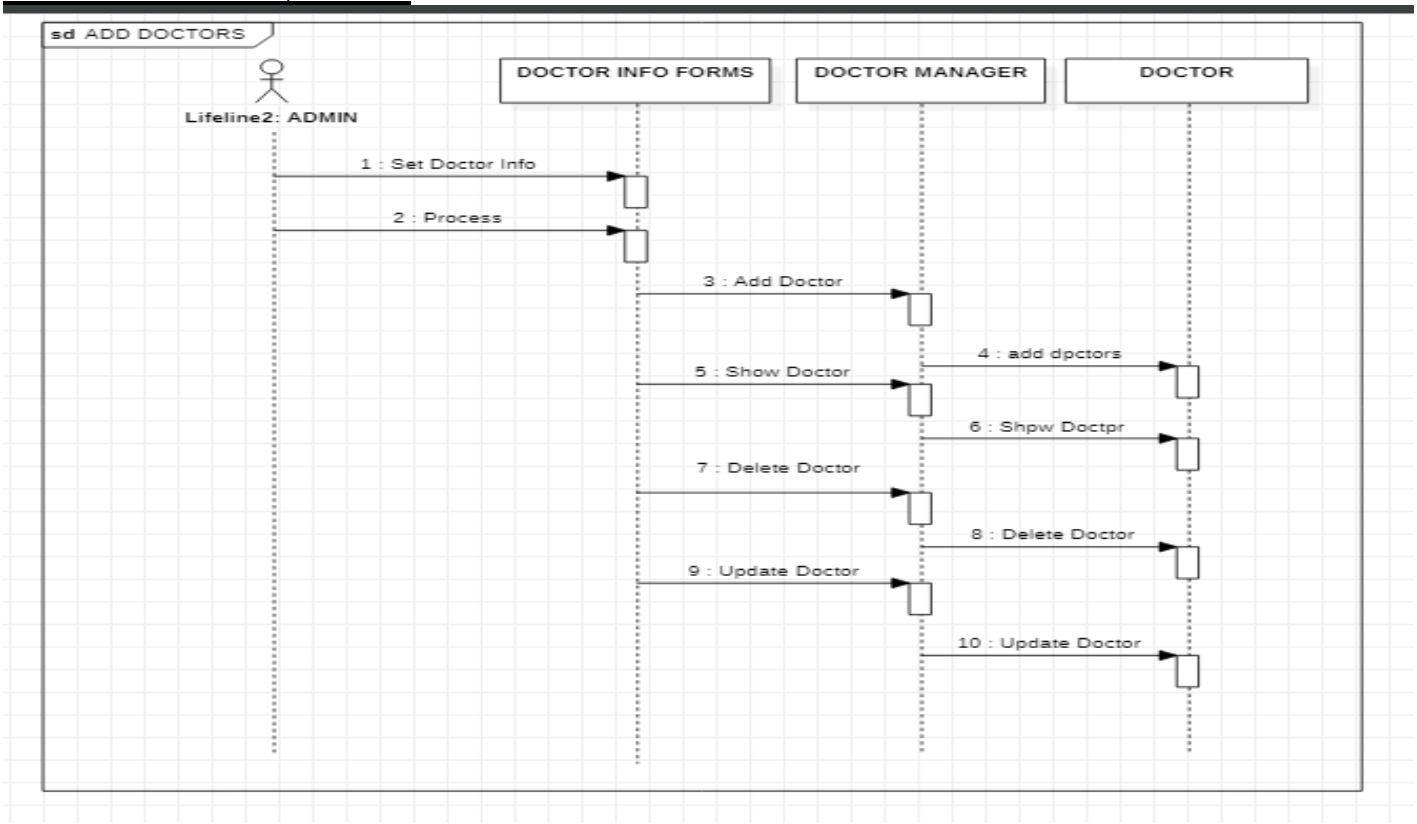
The doctor can also check their appointments of patient and diagnose their diseases after knowing their past history. The schedule of doctors are diff with names, qualification, date of joining and leaving

iii. Alternative Flows:

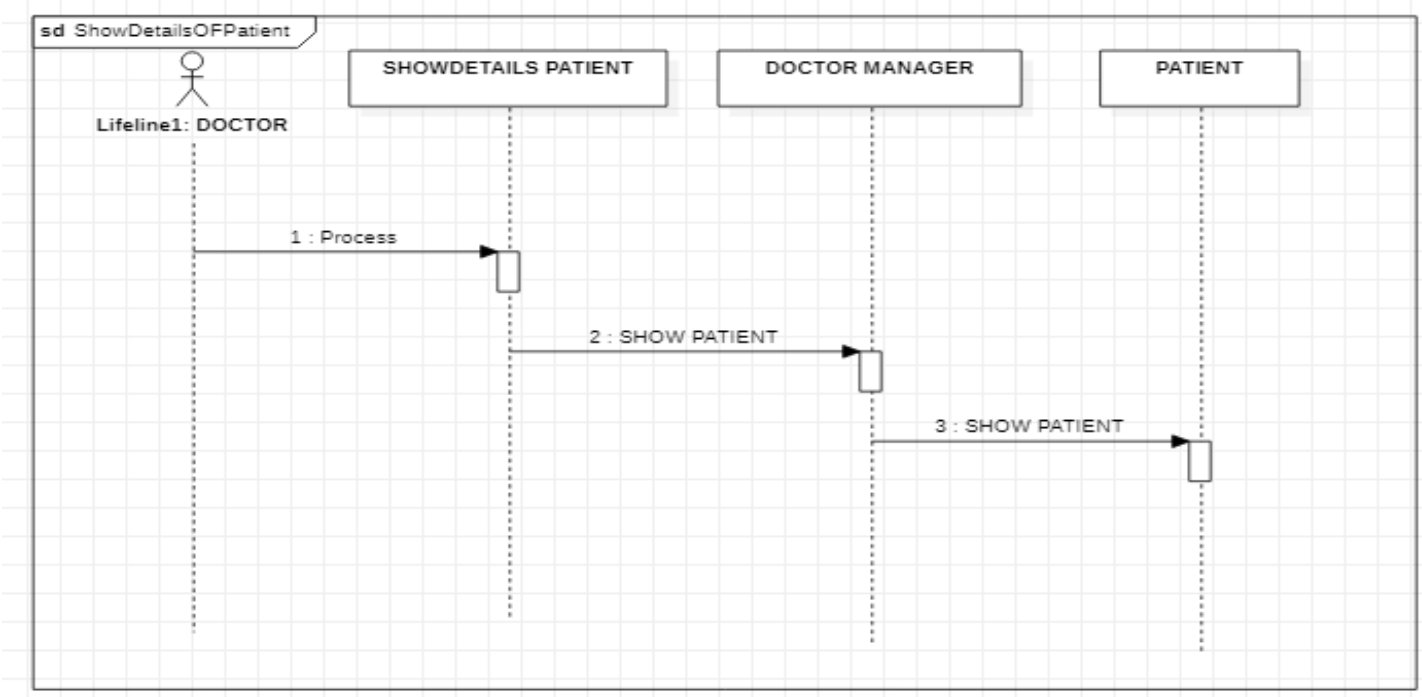
If the invalid doctor id is enter. The doctor can renter the doctor id
If the invalid patient name is enter they can reenter the whole data again
If the invalid appointments are given then reenter whole appointment again

o) ***Sequence Diagram for Login: optional***

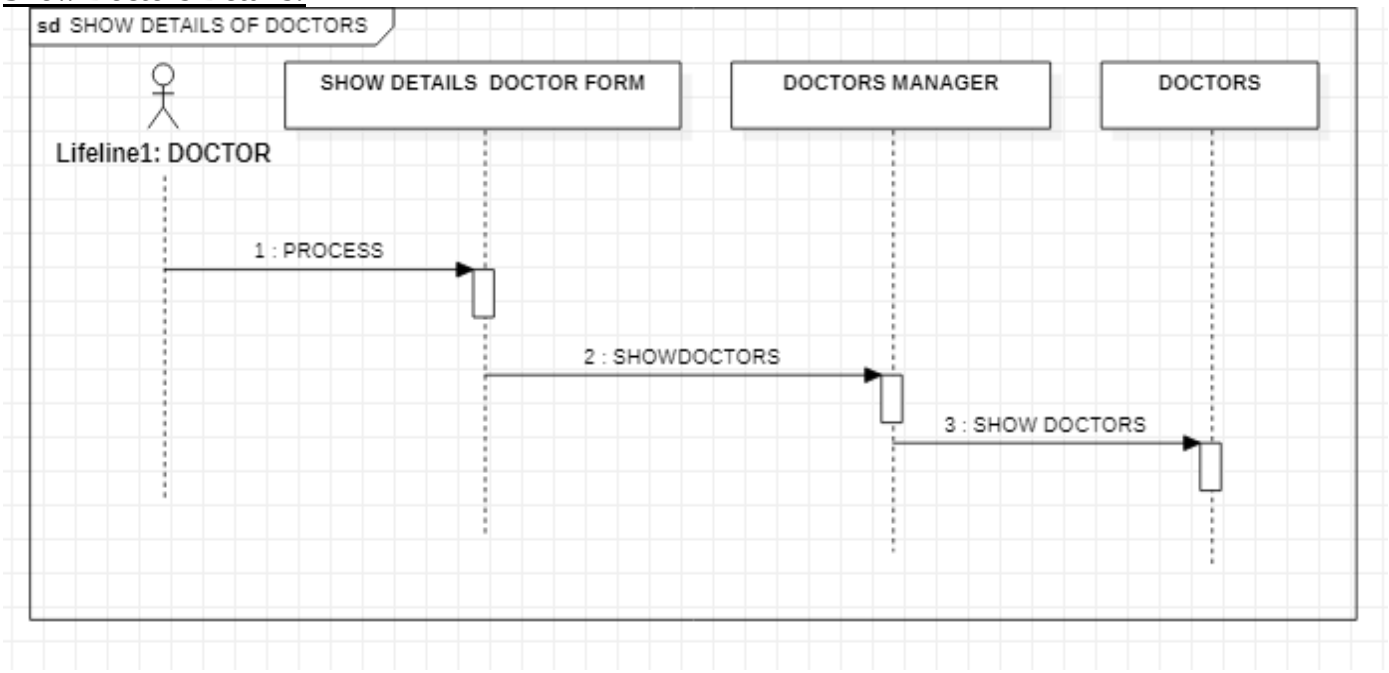
Add Dotor Details By Admin:



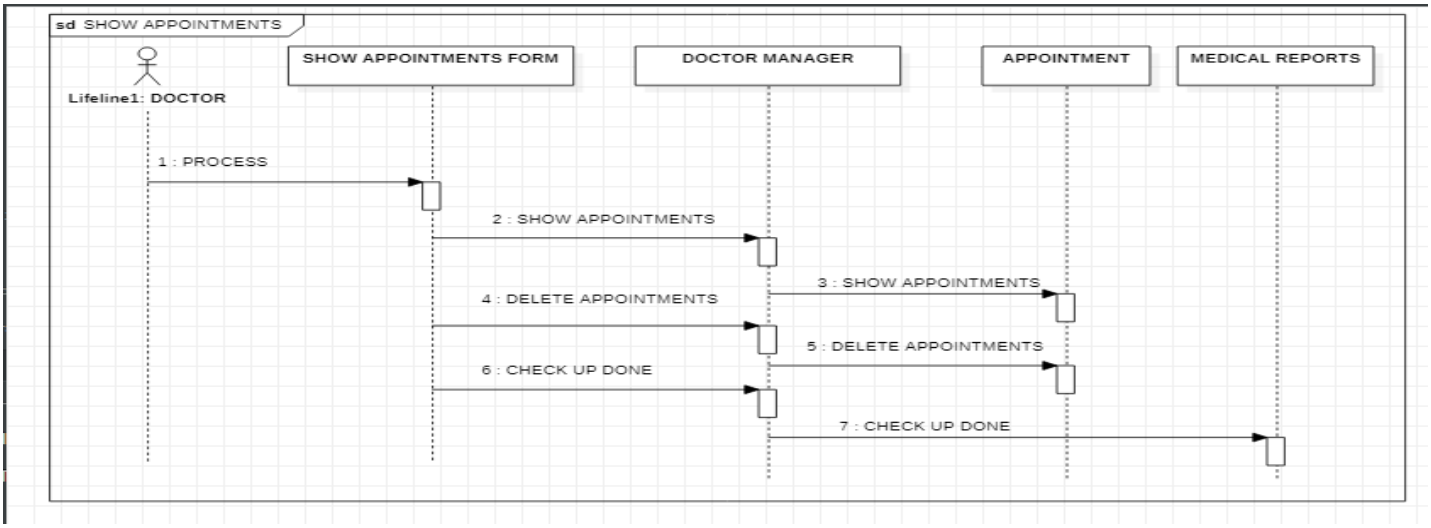
Show Details Of Patient:



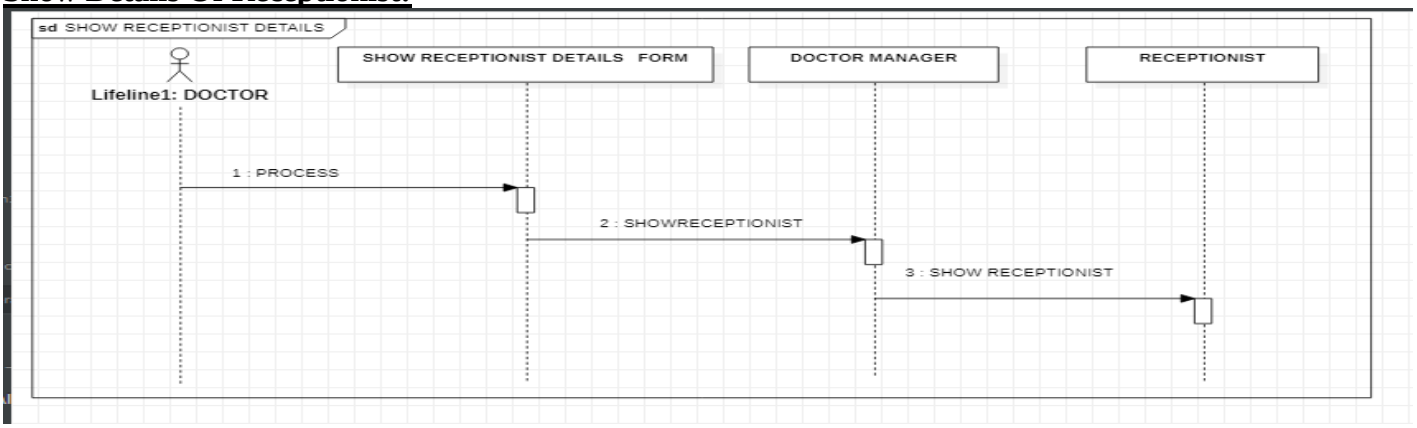
Show Doctors Details:



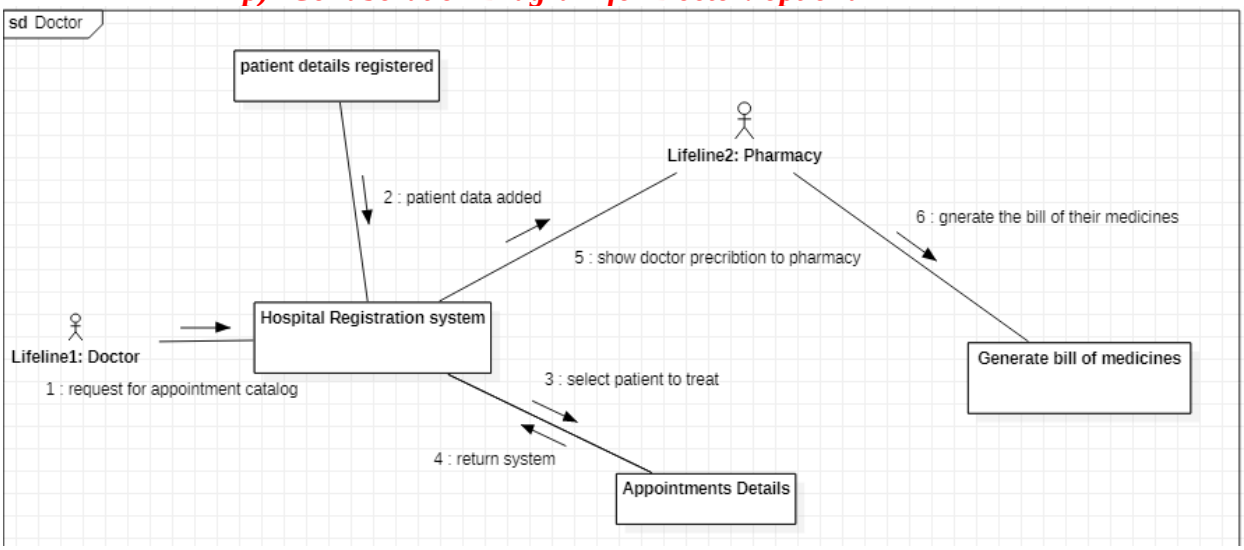
Show Appointments:



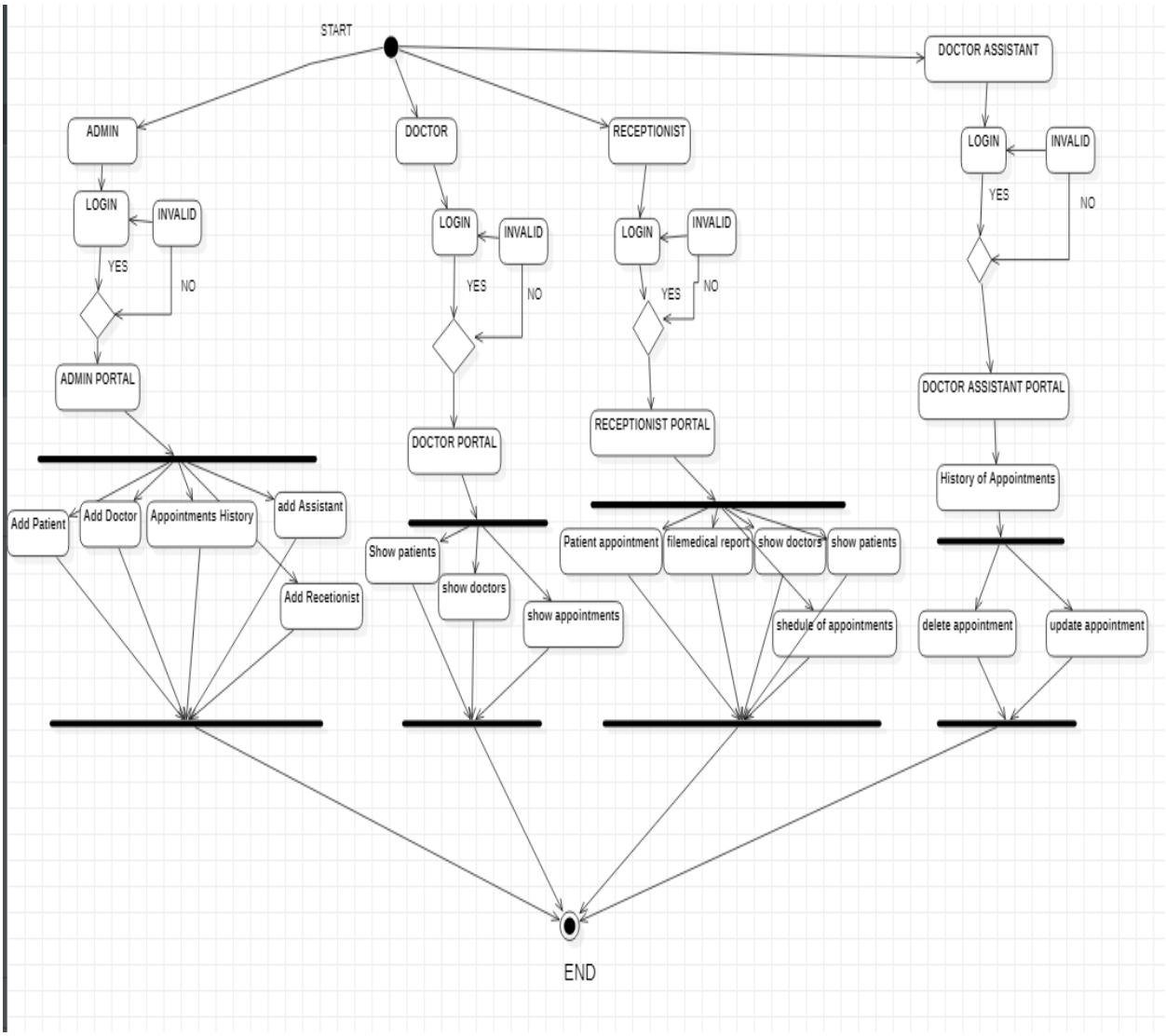
Show Details Of Receptionist:



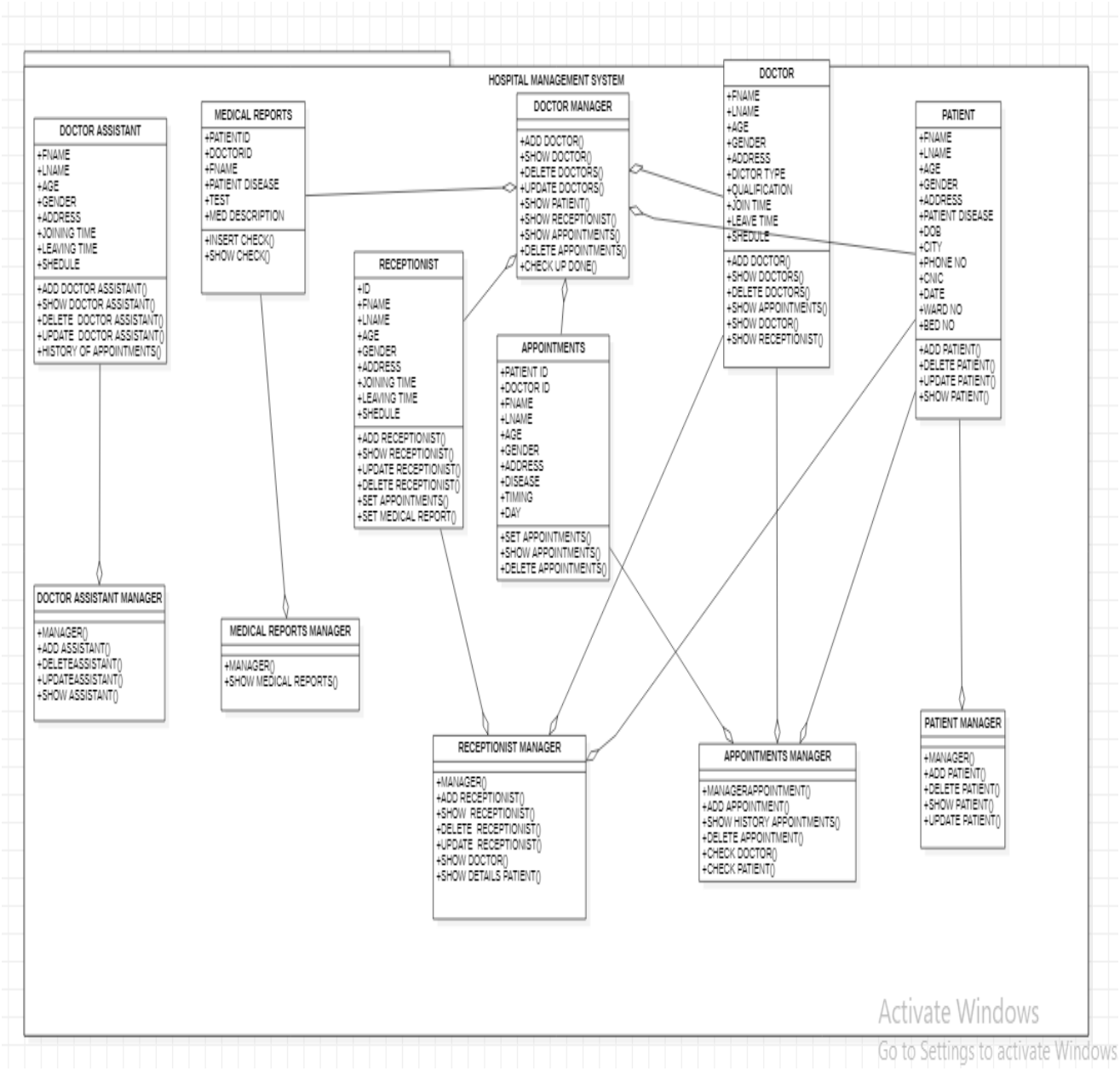
p) Collaboration Diagram for Doctor: optional



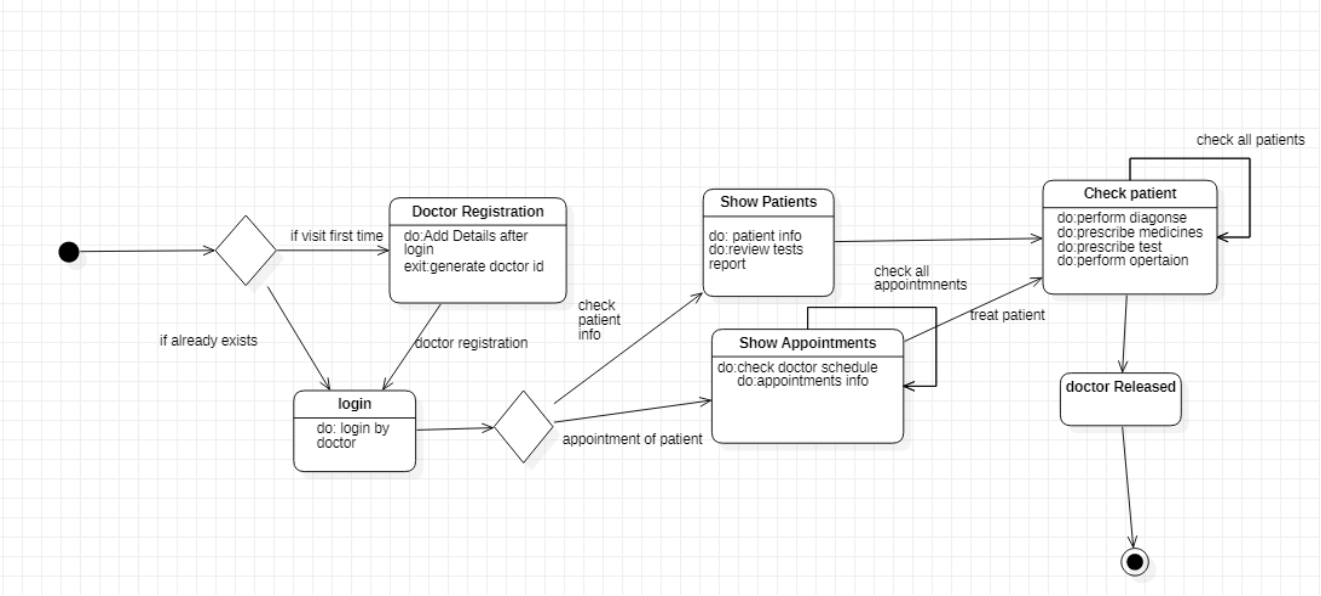
q) Activity Diagram for Doctor: optional



r) Class Diagram for Login: optional



s) State Chart Diagram:



3.1.4 Module 4 complete CRUD Appointment

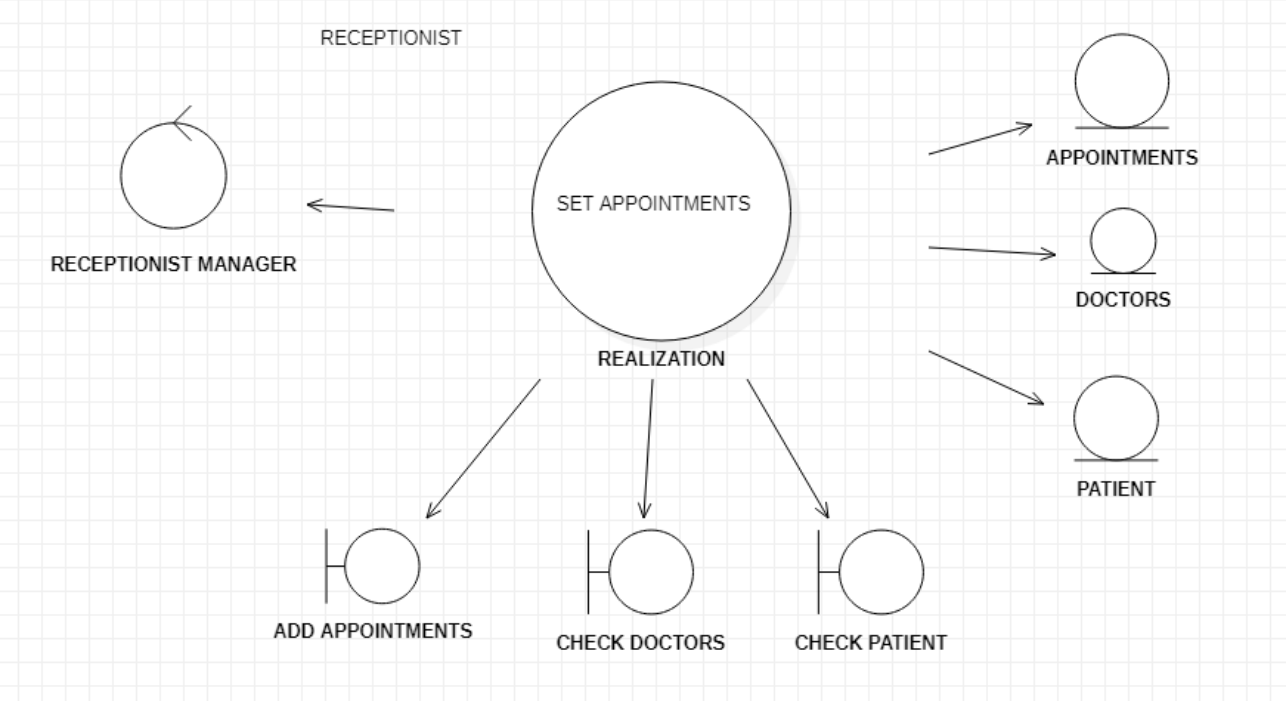
l) ***Description:*** Its is a receptionist in which we have method of add, delete, show, and update receptionist and all the date is store in SQL server database , receptionist also have duty to set the appointments by which doctor can treat the patient according to their given appointments list and also maintain the medical reports of each patient.

t) ***Usage Scenario/ Use case Description/Specification:***

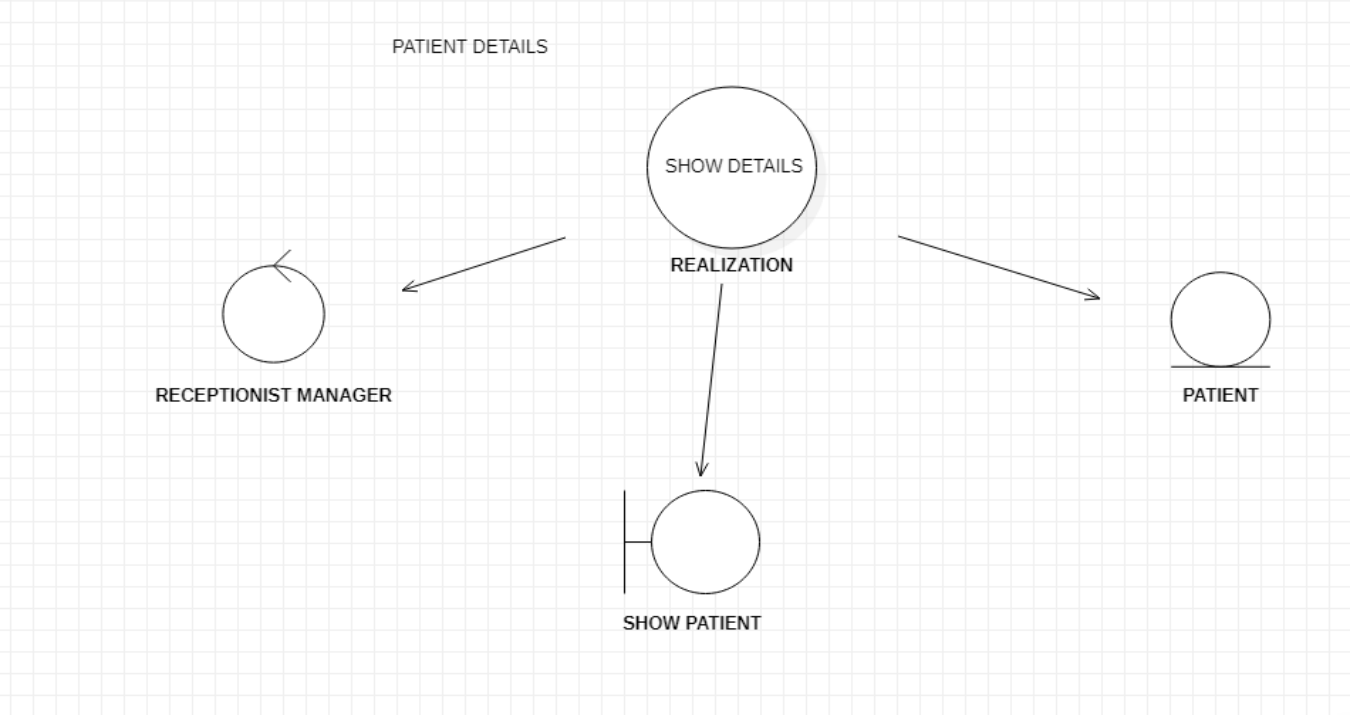
Description	Allows access to hospital management system
Inputs	Username, password
Source	5. User inputs username and password 6. Press Login Button
Alternate case	
Outputs	Successful login; unsuccessful login
Destination	None
Pré condition	Authorized User, check appointment, patients
Post Condition	No change to hospital Database
Side Effects	Failures and successful appointment are sent to hospital Database

u) Use case Realization for receptionist: optional

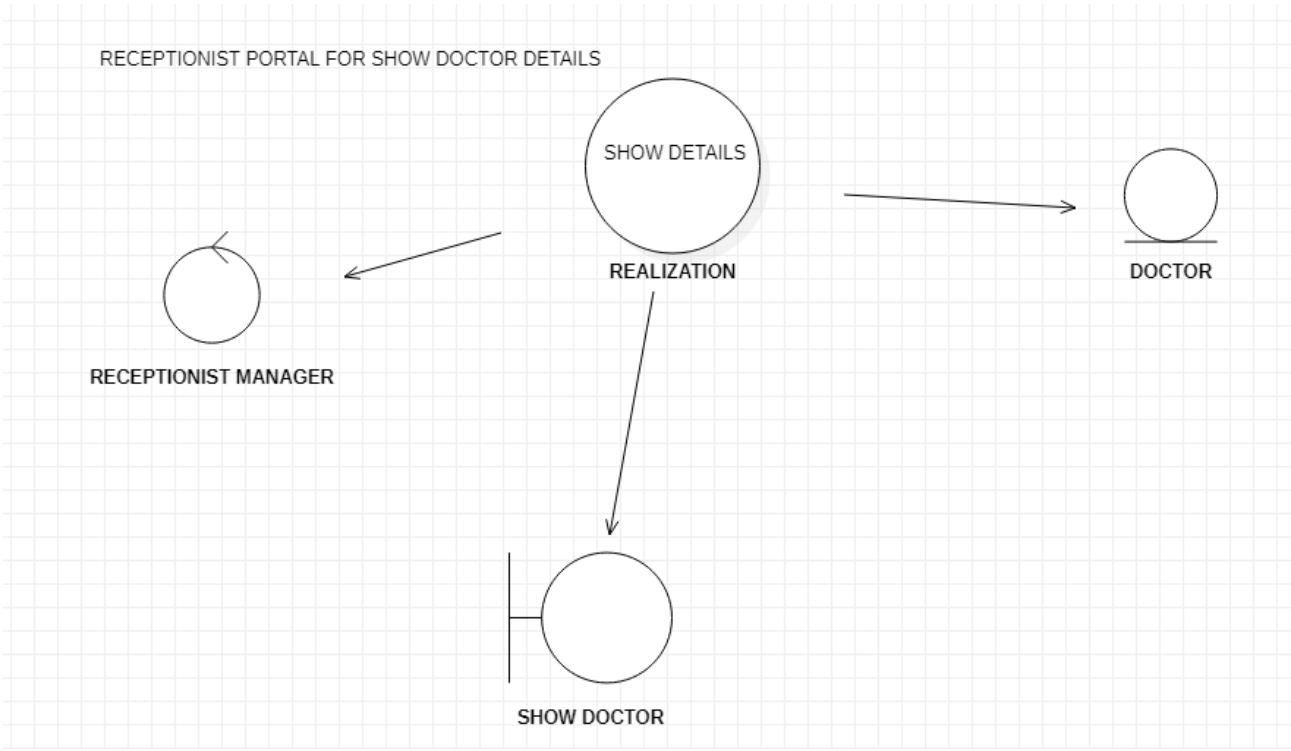
RECEPTIONIST:
SET APPOINTMENTS:



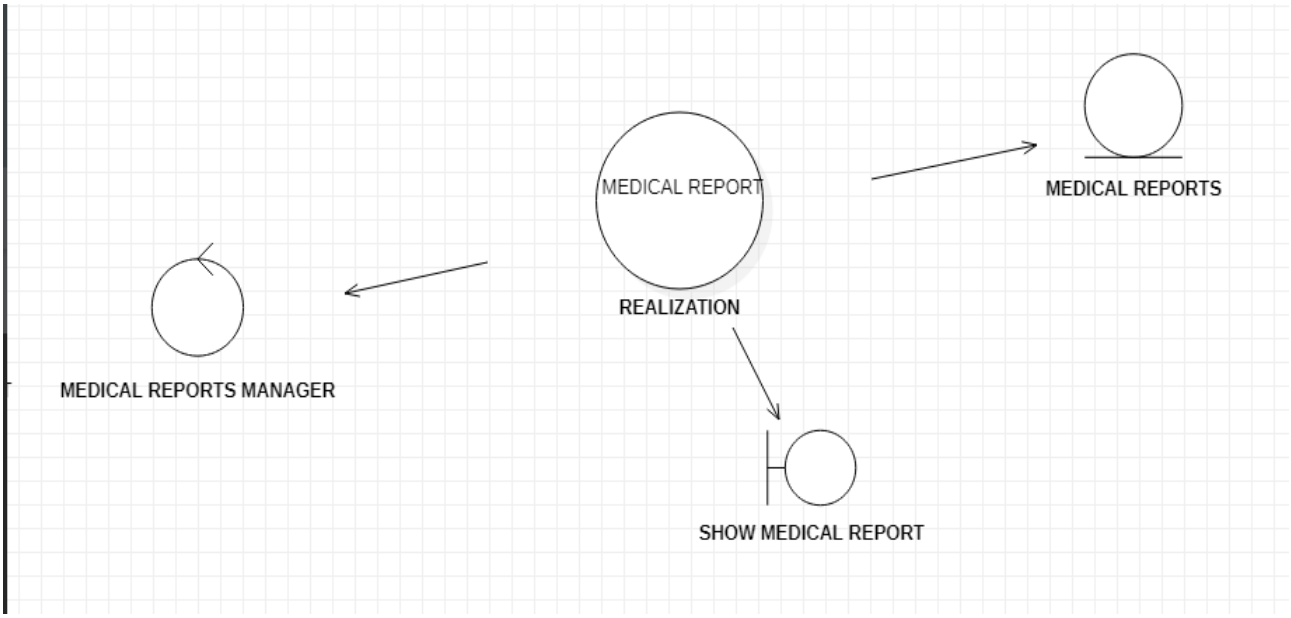
SHOW PATIENT DETAILS



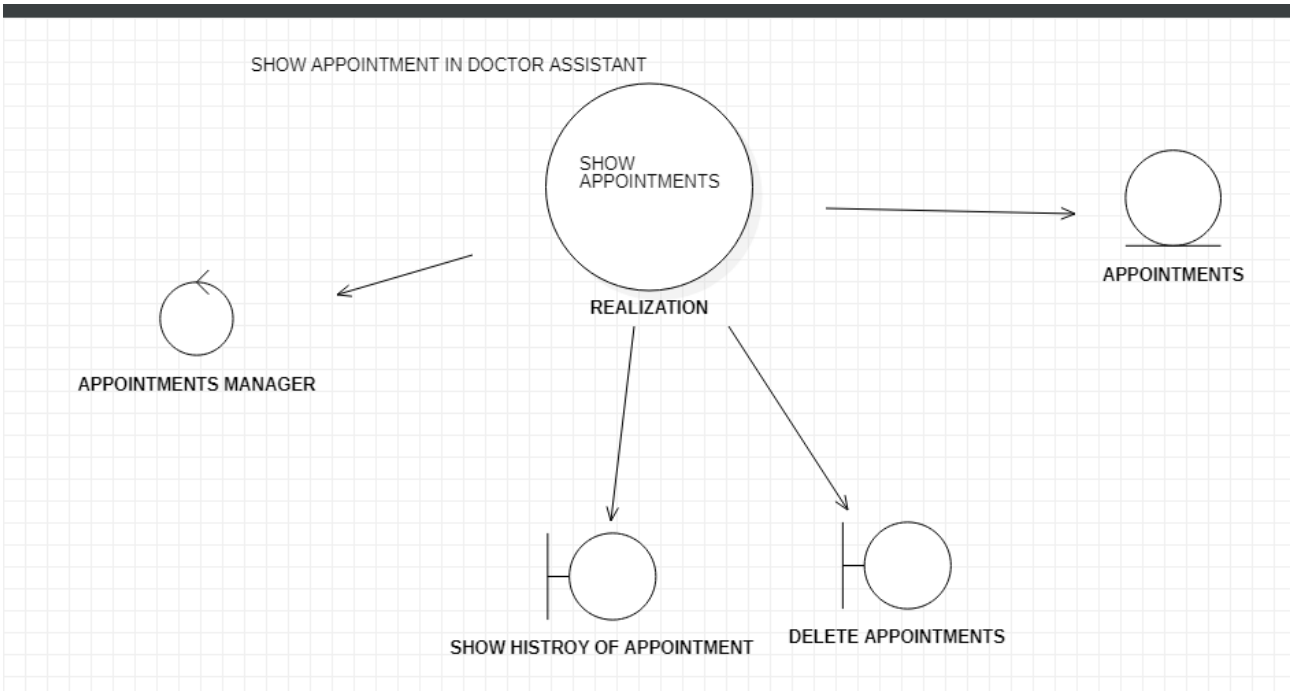
SHOW DOCTOR DETAILS:



SET MEDICAL REPORTS:



DOCTOR ASSISTANT:
HISTORY OF APPOINTMENTS:



v) Flow of Event or Data Flow Diagram for appointment : optional

- **Precondition:**

  The sub flows is to maintain the all records &information which is related to the hospital.
  It can also maintain all appointments and main file for each patient .It must execute before this use case begins.receptionist set all the patient appointment s and set medical reports of each

- **Main flow:**

  This use case begins with the receptionist login onto registration system and enter his/her password the system verifies that passwordis valid and prompts the receptionist to add patient and giving appointments to each patient. Receptionist can also show the details of the patients

- **SUB FLOWS:**

  Show details of patient

  The receptionist can see the patient data. Which Includes name, father name, disease, word no, bed no and edit them (delete, update)

  Appointments

  The receptionist can also maintain and giving appointments to the patient after knowing doctors schedule. The schedule of doctors includes names, qualification, date of joining and leaving

  ALTERNATIVE FLOWS:

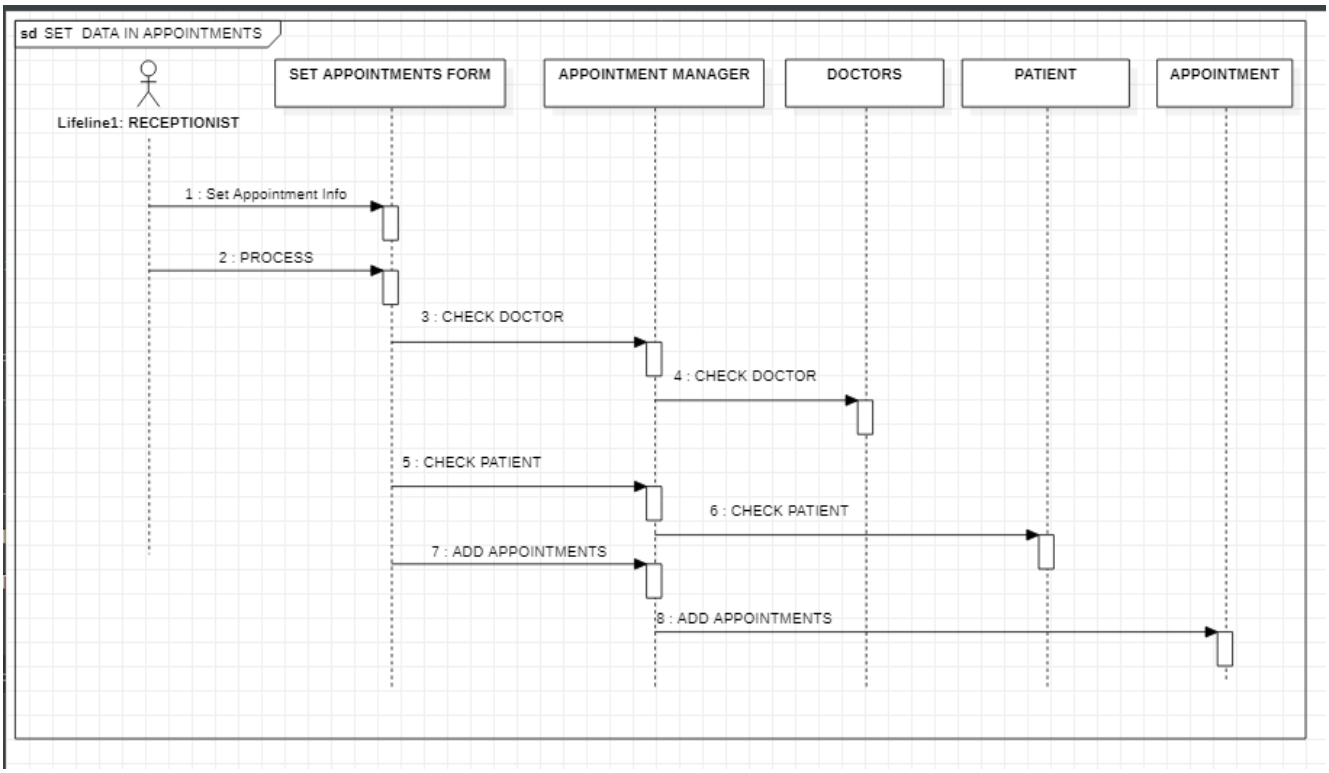
If the invalid receptionist id is enter. The admin can renter the admin id

If the invalid patient name is enter they can reenter the whole data again. If the invalid appointments are given then reenter whole appointment again.

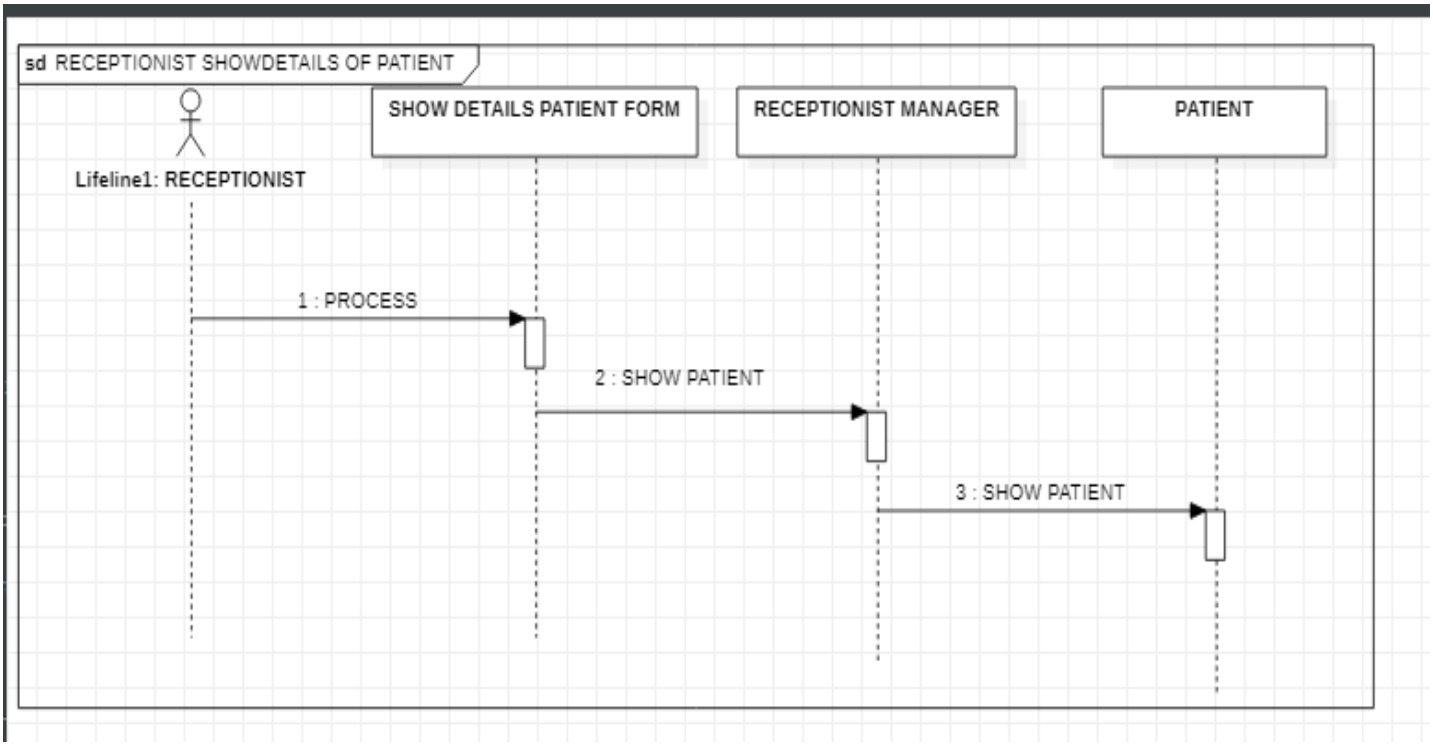
w) ***Sequence Diagram for receptionist: optional***

RECEPTIONIST:

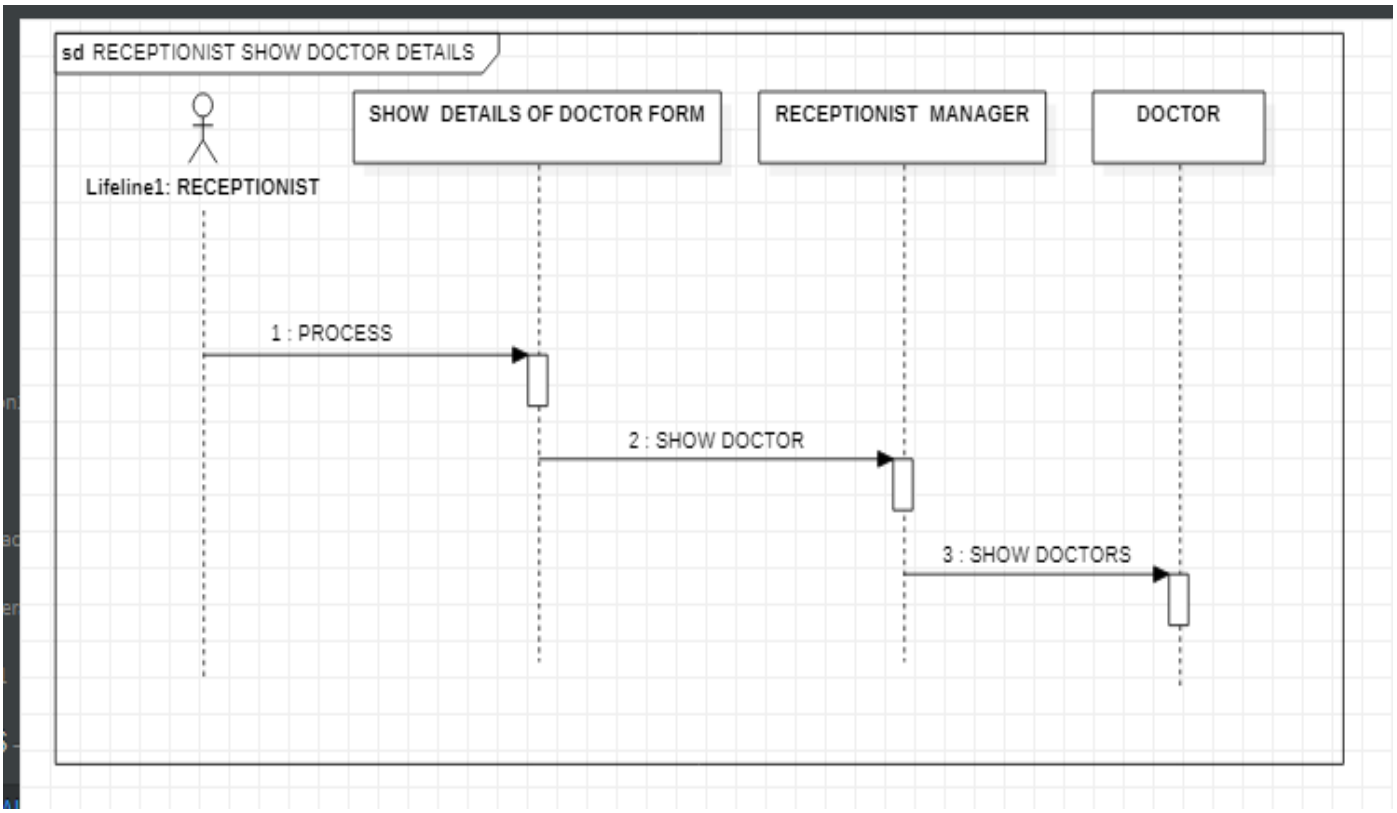
SET APPOINTMENTS:



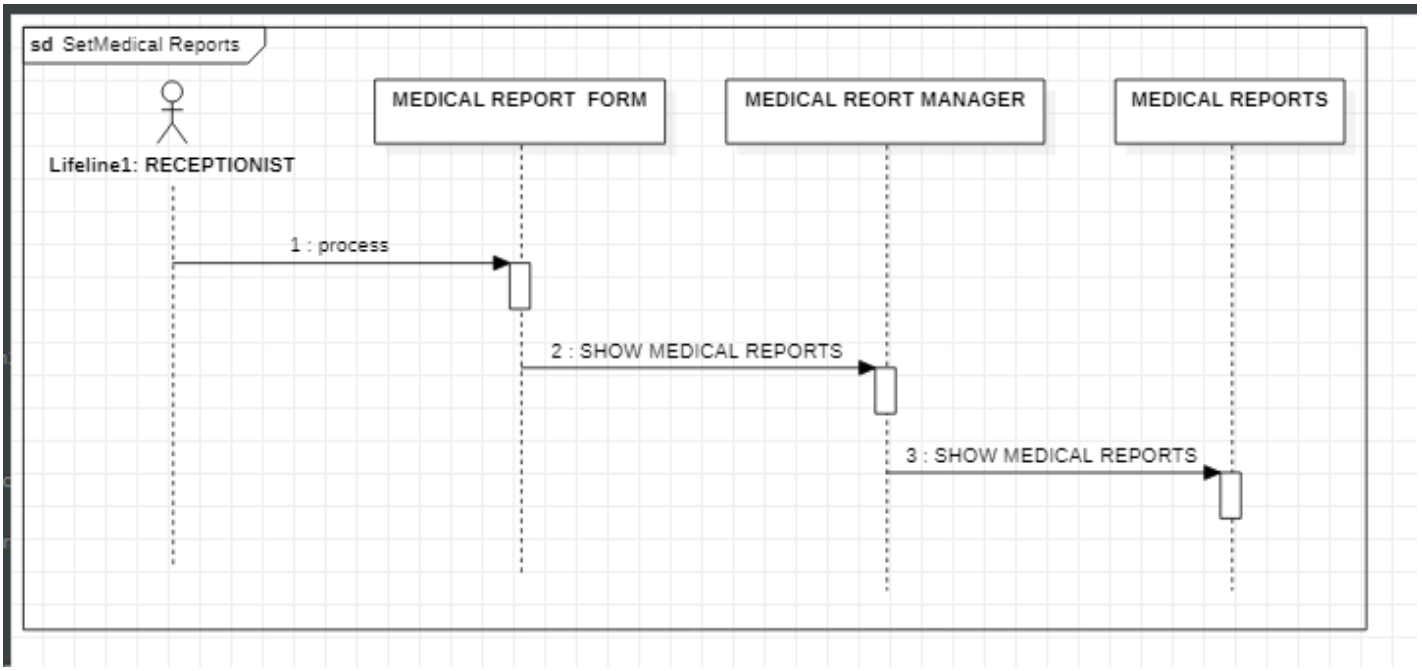
SHOW PATIENT DETAILS:



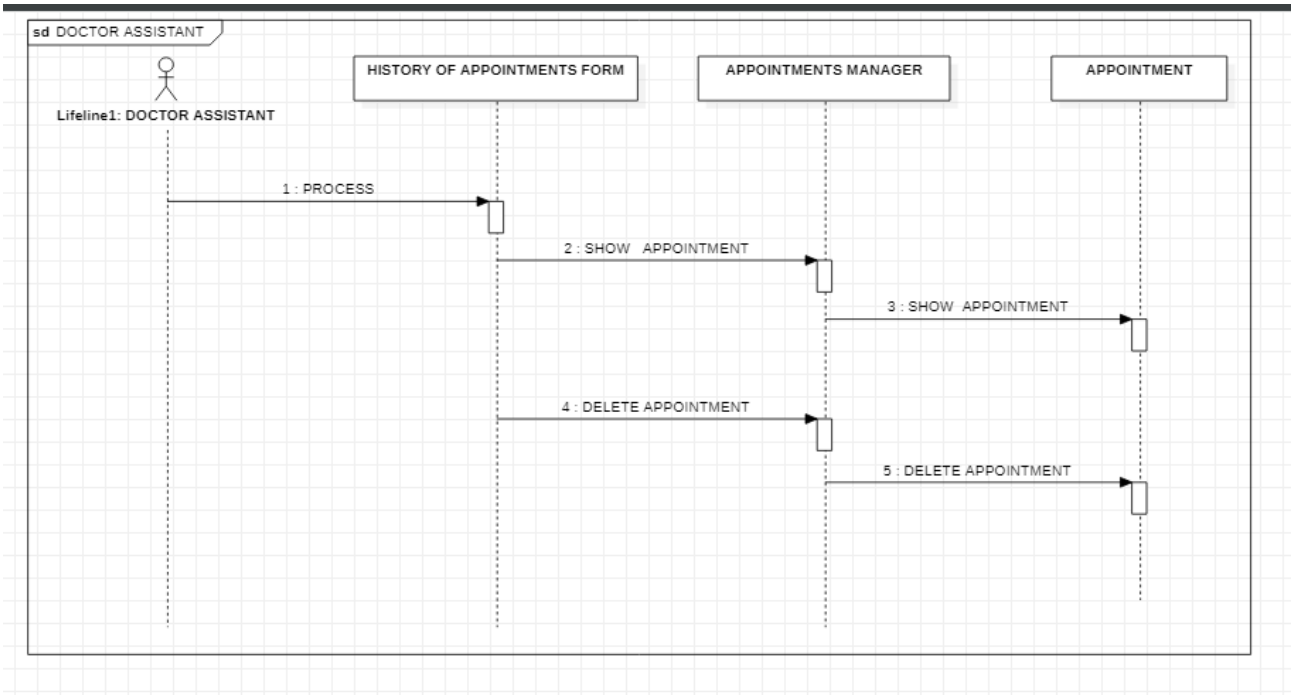
SHOW DETAILS OF DOCTORS:



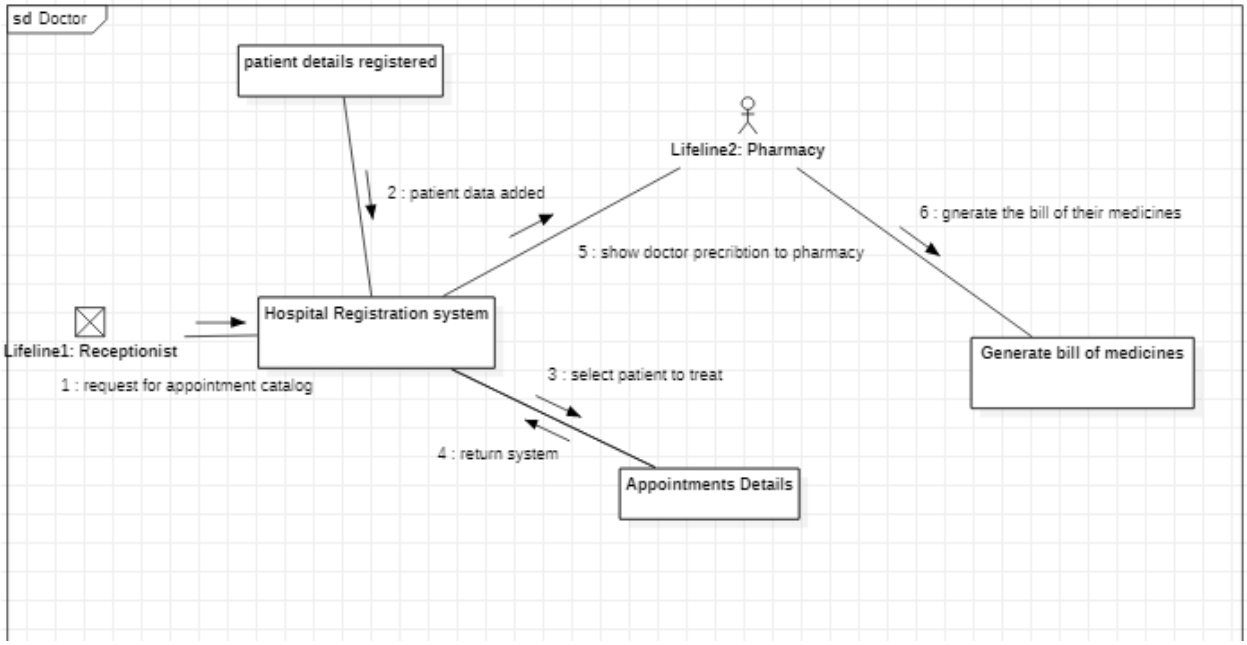
SET MEDICAL FILE:



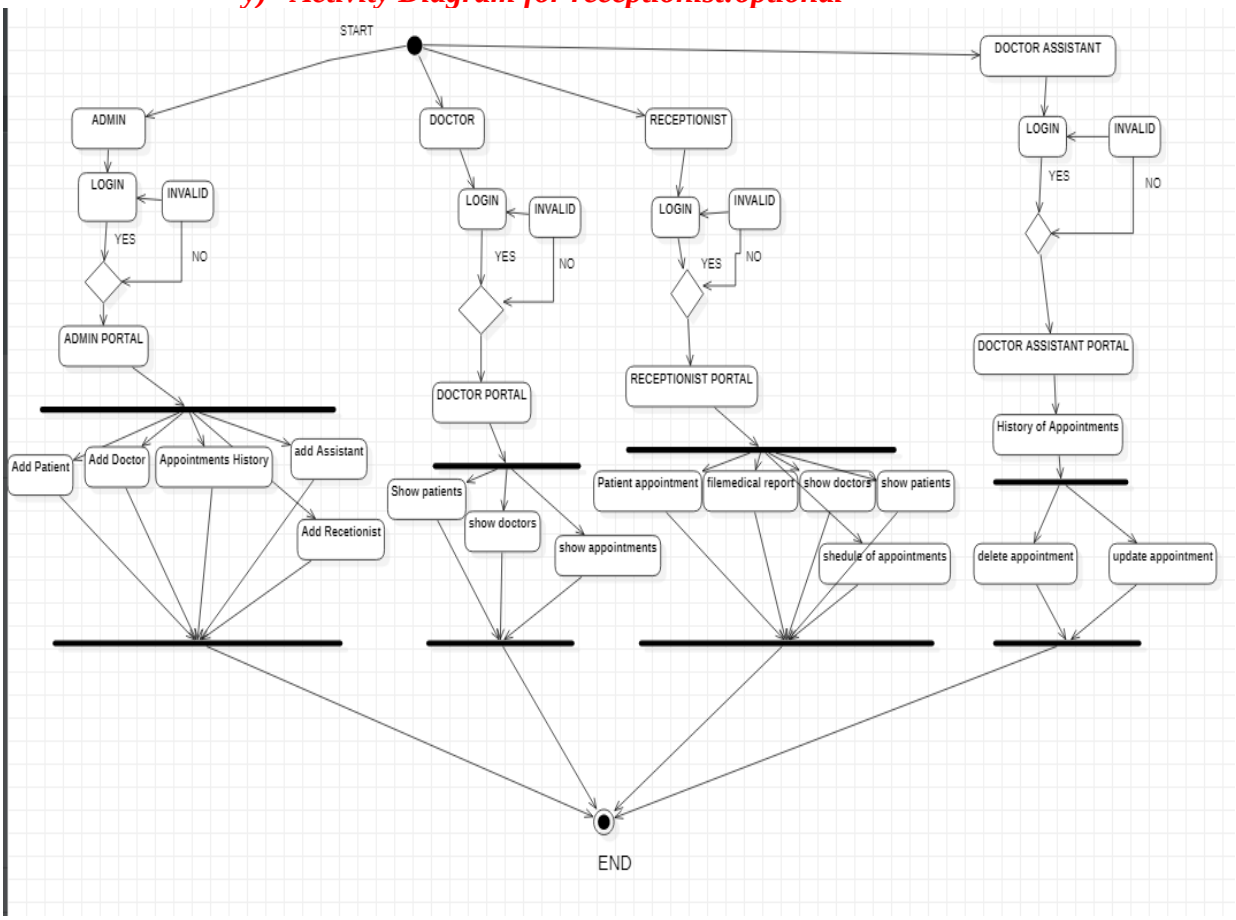
DOCTOR ASSISTANT:



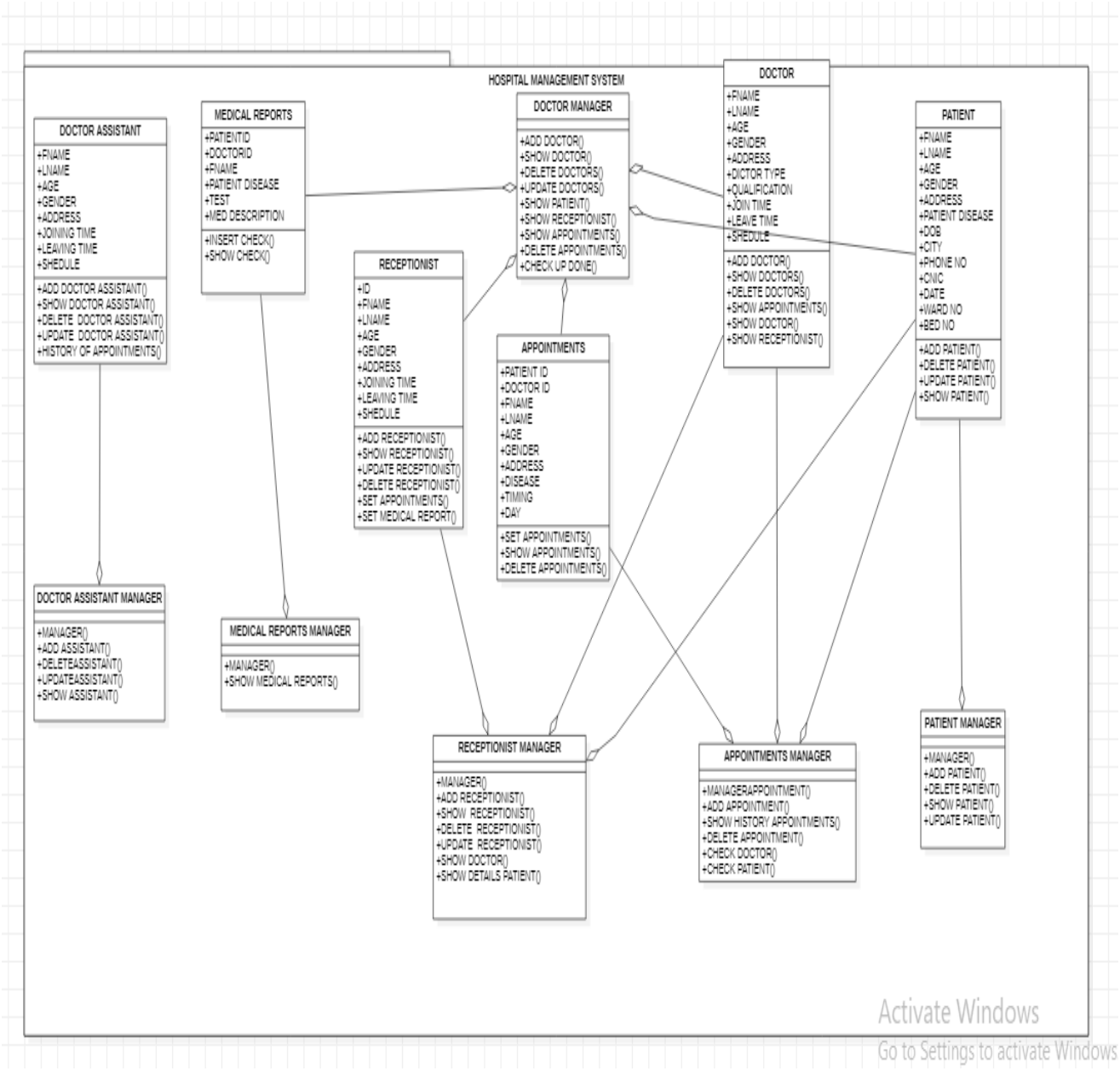
x) Collaboration Diagram for receptionist: optional



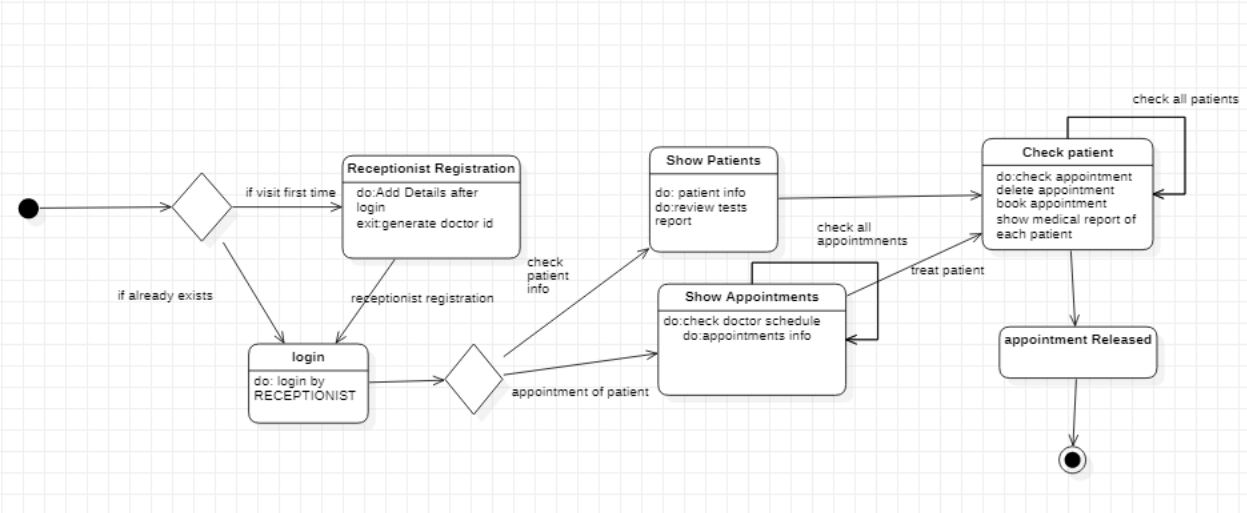
y) Activity Diagram for receptionist:optional



z) Class Diagram for receptionist: optional



aa) State Chart Diagram:



3.1.4 Module 5 complete CRUD Doctor Schedule

Description: This use case begins with the login onto registration system and enter his/her password the system verifies that password is valid and prompts the receptionist to delete patient appointments and maintain the history of appointments to each patient. doctor assistant can also check the details of the patients and take history of their disease and then he present the history of each patient to the doctor and after that doctor can treat the patients according to their diseases.

bb)

cc) ***Usage Scenario/ Use case Description/ Specification:***

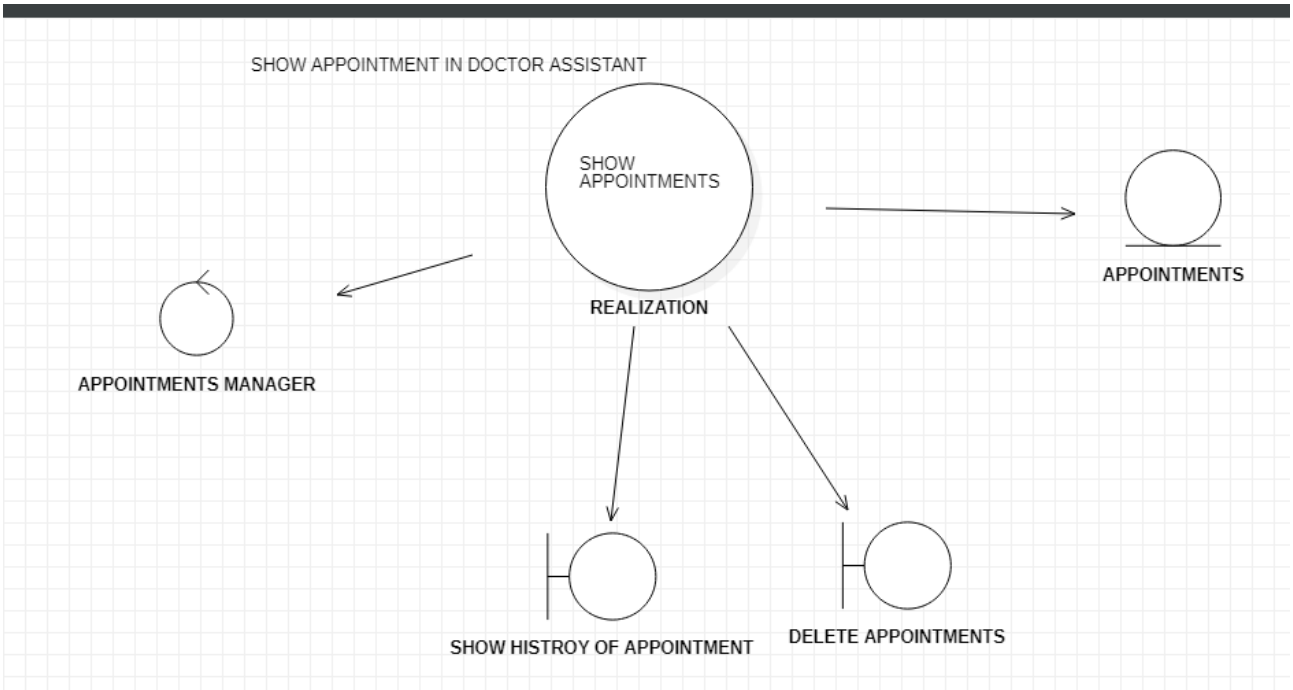
Description	Allows access to online HMS
Inputs	Username, password
Source	7. User inputs username and password 8. Press Login Button
Alternate case	
Outputs	Successful login; unsuccessful login
Destination	None
Precondition	Authorized User
Post Condition	No change to Passenger Accounts Database
Side Effects	Failures and successful logins are sent to Reservation Database

dd) Detailed Use case Diagram for* Doctor Schedule*: optional

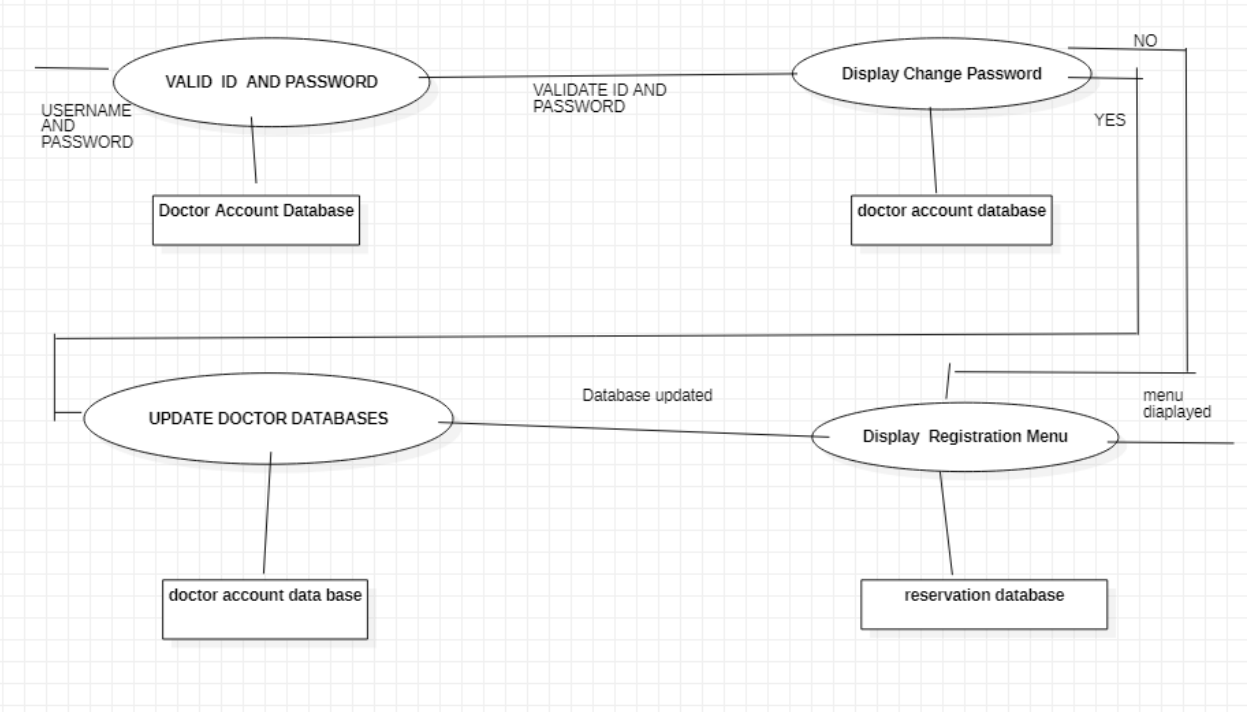
ee) Use case Realization for* Doctor Schedule*: optional

DOCTOR ASSISTANT:

HISTORY OF APPOINTMENTS:

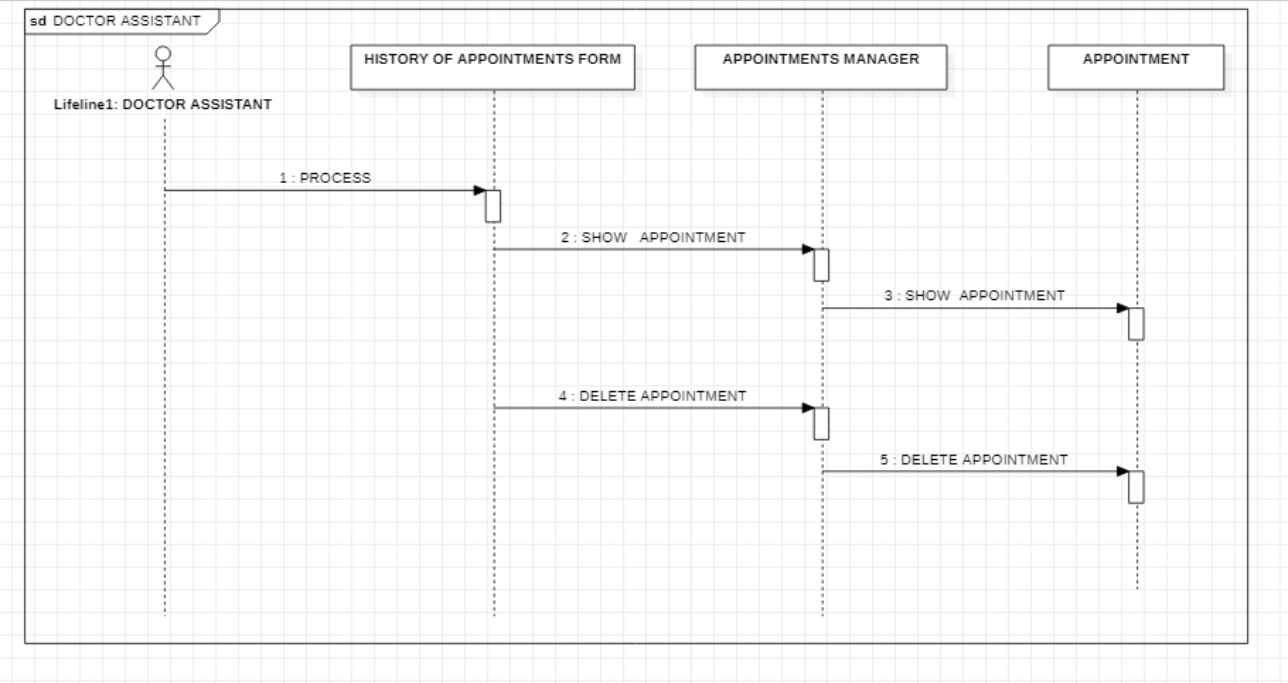


ff) Flow of Event or Data Flow Diagram for Doctor Schedule*:*



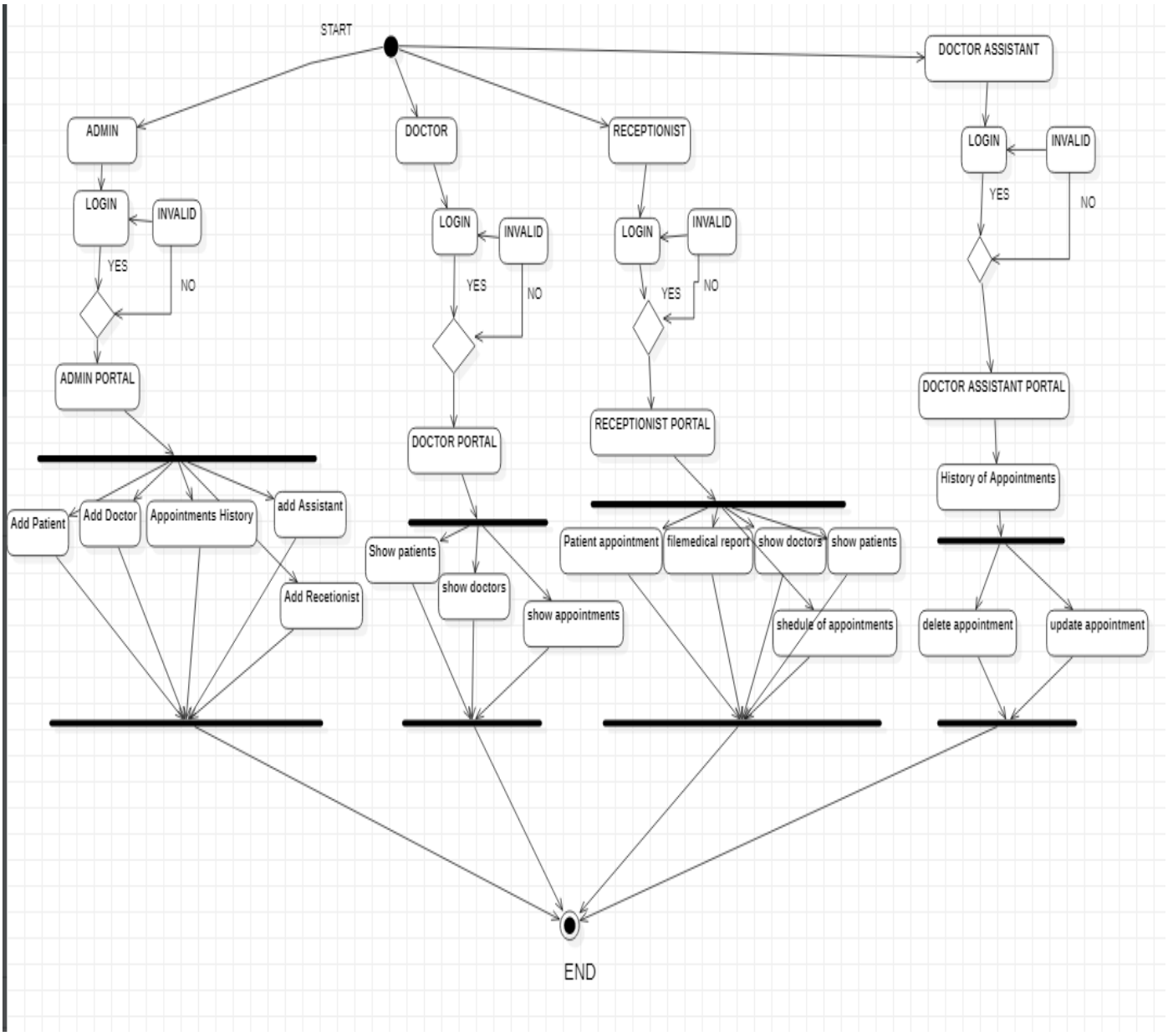
gg) ***Sequence Diagram for*** **Doctor Schedule*:***

DOCTOR ASSISTANT:

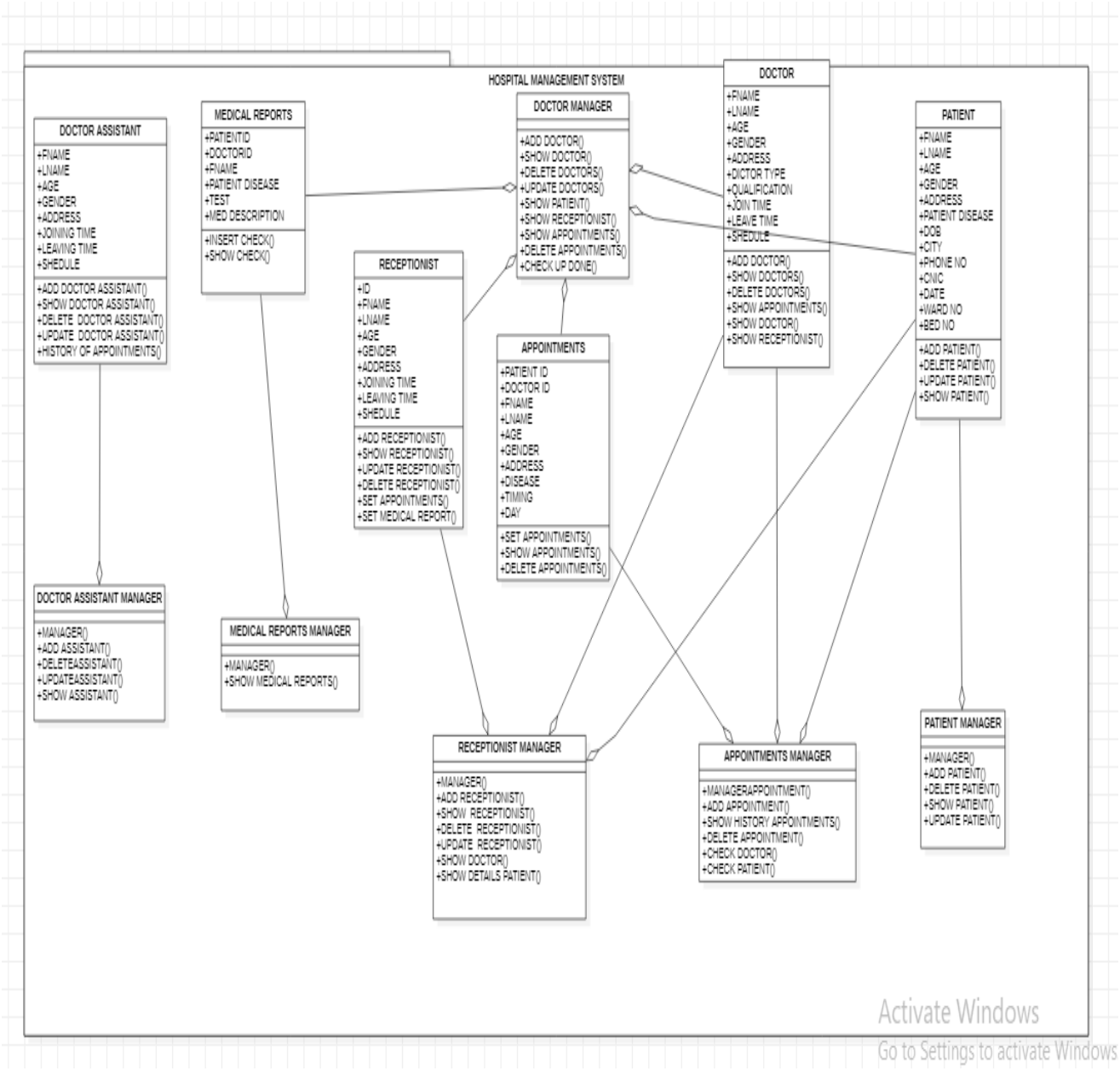


hh)Collaboration Diagram for **Doctor Schedule*: optional***

ii) Activity Diagram for **Doctor Schedule*: optional***



jj) Class Diagram for Doctor Schedule*: optional*



kk) State Chart Diagram for* Doctor Schedule*: optional

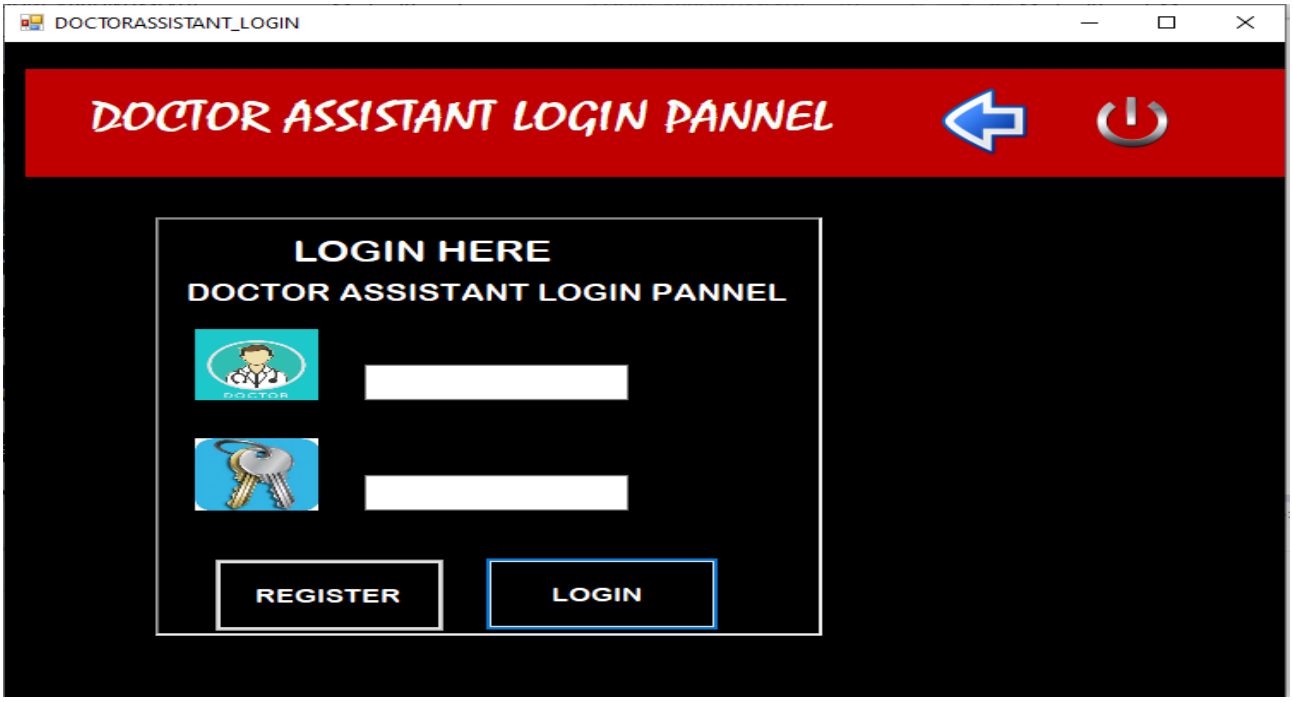
3.2. External Interface Requirements

3.2.1 User Interfaces

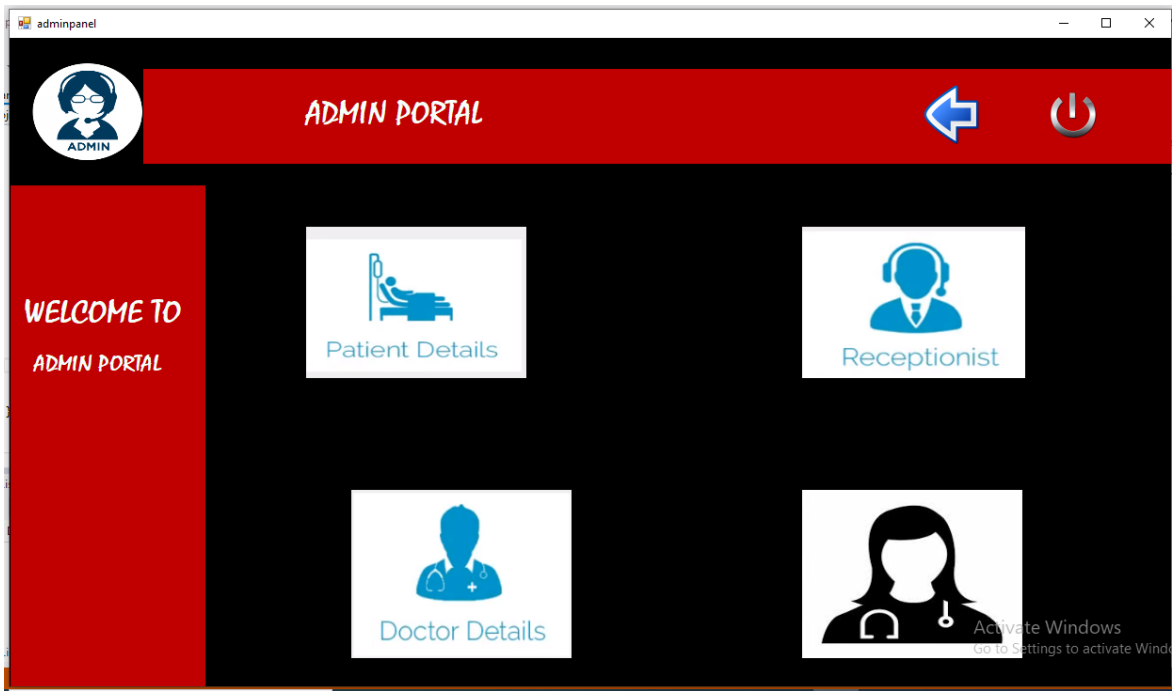
The user interfaces are divided into two major components. One part includes the user accessing the system using a cell phone. The other portion involves accessing the system through a remote site or at a particular location specifically designed to access the system. For instance, the clerks and the access the reservation system from the reservation office.

The diagrams and explanations below demonstrate the major transition from one user interface to another. This is a brief description. However, a more detailed demonstration is done in the prototype. The purpose of this interaction is to illustrate the overall view of the HMS.

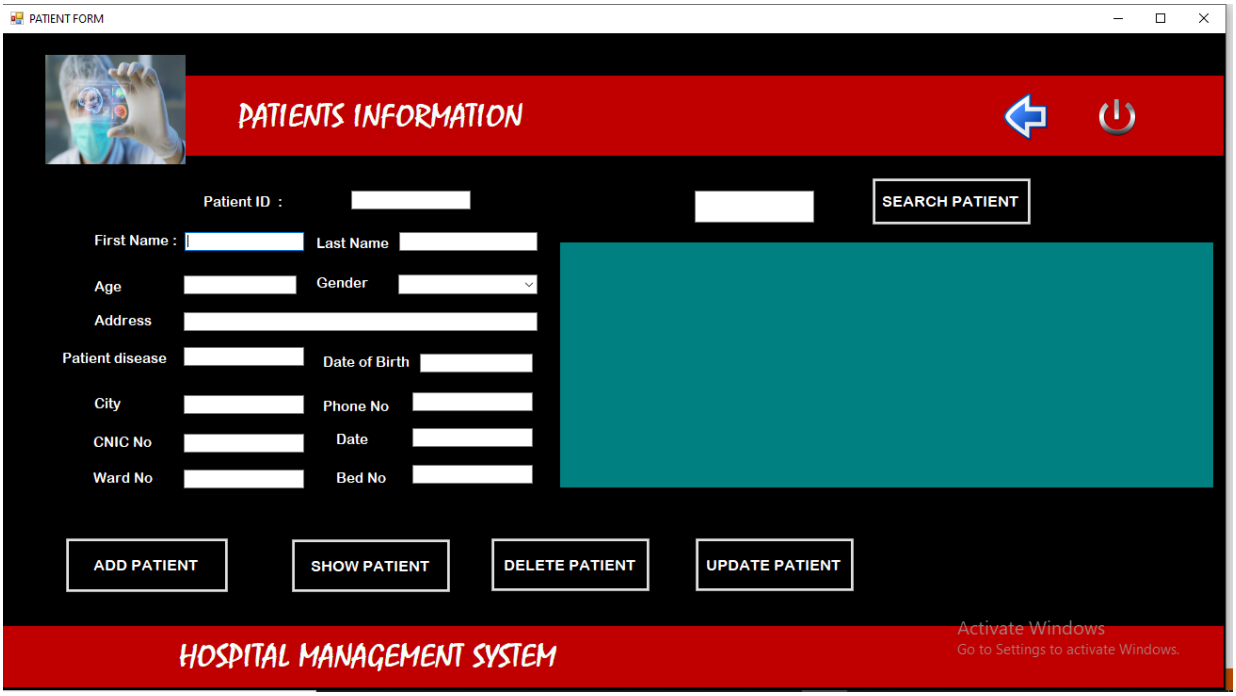
The diagram below illustrates the four **major functionalities or modules** . These functionalities will be displayed depending on the user. For instance, the CRM will see all four functionalities while the normal user and the clerks will only see the appointment Reservation and the Passenger Account.



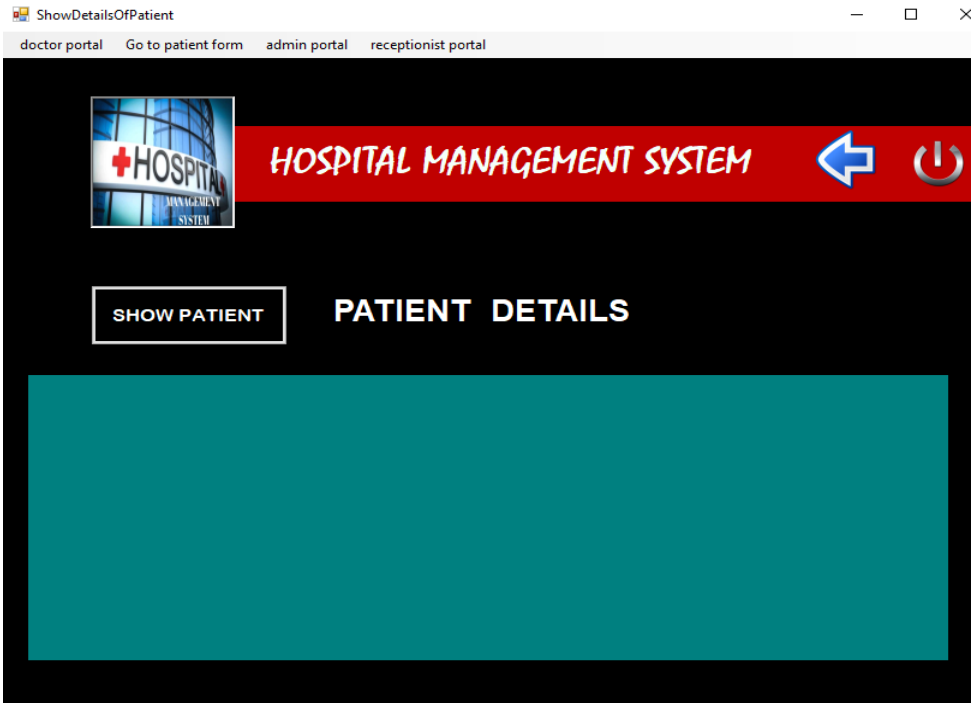
Selecting one of these functions will take the user to a different user interface. For instance, choosing appointment Reservation will display the following web page. The title of this page is consistent with the function selected, and since the appointment Reservation was selected, the title displays appointment Reservation. The purpose of this is to allow the user know what part of the system they are accessing. Furthermore, the user can select any of the four functions.



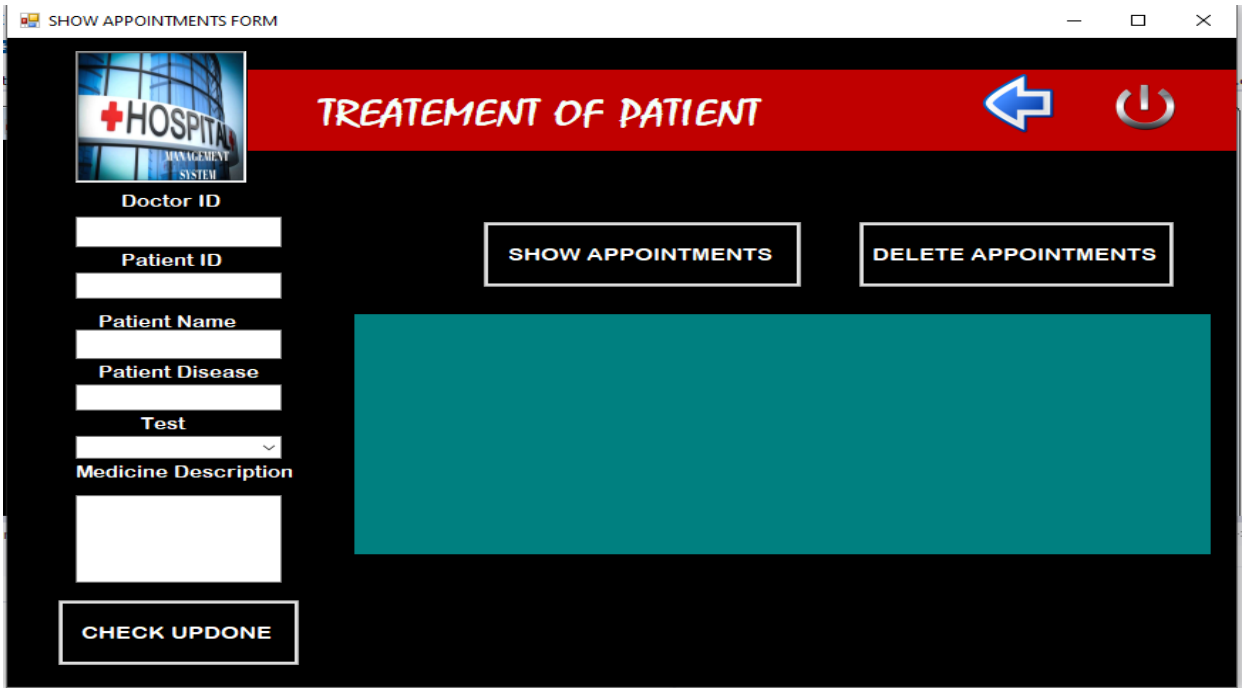
The user can select any of the four functionalities. For the sake of this demonstration, if the user clicks on the Make Reservation function the diagram below is displayed. Once again the title is the same as the main function and a subtitle indicates the second function selected. In addition, the person can fill up the following information and the date of travel or return if he/she wishes. The three buttons allow the user to navigate through the interfaces. For instance, the back button will take the user to the above page, and the clear button will clear the form of any selection he/she made before. The Display Available displays the available trains, seats and what city they want to travel to. However, before we get to the next page when clicking Display Available the picture below illustrates the Make Reservation function.



The Display Available function displays all the trains traveling from one city to another and the seats available on that train. Furthermore, the last list displays the number of tickets available for the particular train on the selected route. The back button will take the user to the above picture, and the confirm button takes the person to the payment page.



The following page allows the user to pay for the ticket as appropriate. Now, this page is part of the Passenger Account function, and it is used here to make payment for the ticket selected. This makes it easier for the user since they do not have to go back to the main menu and to access their account.



3.2.2 Hardware Interfaces

The HMS includes two major hardware components: cellular phones and regular PC's. The cell phones require WAP (wireless application protocol) network protocol, which is already programmed in the latest phones.

The second component involves the regular PC's, which communicate with the server. The server then communicates with the database. The protocol involved between the PC's and the server is the HTTP protocol, which allows communication between the PC's and the Server. The remote PC's, such as someone accessing the HMS from home using the Internet, are able access the information through the CGI. The requests come in through the HTTP protocol, and using an ODBC the database results are returned and processed using Perl to give an HTML web page. The format of the output is displayed as web pages.

3.2.3 Software Interfaces

An Oracle DBMS will be used to manage the database and any changes made to it. Furthermore, the DBMS will make regular backups of the database and generate reports regularly so that they can be accessed by the CRM. The Apache server between the client and the database will handle all communication, and the server will run on a Linux operating system. Furthermore, the HTML pages must be implemented such that they can be displayed on two common browsers: Netscape and Internet Explorer.

Information about the products used for the HMS:

(1) Name: Oracle
(2) Mnemonic: Oracle
(3) Version Number: ?
(4) Source: Oracle

(1) Name: Linux
(2) Mnemonic: Linux
(3) Version Number: 6.2
(4) Source: Unix

(1) Name: Internet Explorer
(2) Mnemonic: IE
(3) Version Number: 5.00
(4) Source: Microsoft

(1) Name: Apache
(2) Mnemonic: Apache
(3) Version Number: 1.3.14
(4) Source: Apache Software Foundation

3.3 Performance Requirements

The following sections list the performance requirements for the system.

3.3.1 User Requirements

User Requirements	**Description of Requirement For Design Environment**
Location(s) and Number(s) of Users	Karachi, Pakistan
Expected Growth in Number of Users	
After 1 Year	50%
After 2 Years	TBD
After 3 Years	TBD
User Expectation	
Interactivity	User expect that it provides a very easy to use graphical user interface
Reliability	For some applications, reliability must be 100% during the application session
Adaptability	Network must adapt to user additions, deletions and changes
Security	Encryption software would be used for Credit Card transactions
Cost / Funding	Less than $250K

3.3.2 Application Requirements

Since no specified service is indicated, then we have listed the applications as best – efforts. This may change as we learn more about the application.

The communication package is determined to be bursty in nature, with small data sizes and frequent transmissions. We can consider this application to be interactive-burst, while the database transaction-processing application is described by the CRM as transferring large amounts of data (initial estimates are 1 MB/transaction), we have listed this application as interactive-bulk.

Categorizing Applications	Best-Efforts	Application Locations
Communication	100 Kb/s	Guangzhou and Nanjing
Database Access	400 Kb/s	All Locations
Database Transaction processing	1.5 Mb/s	All Locations

3.3.3 Host Requirements

	Type of Host or Equipment	Numbers and Locations
Host A	PC	Guangzhou (10), Nanjing(7), Shanghai(10)
Host B	Database	Shanghai

	Server	
Host C	Application Server	Nanjing

3.4.1 Standards Compliance

There are no design constraints that can be imposed by other standards limitations.

3.4.2 Software Limitations

- must be able to run Internet Explorer or Netscape Communicator web browsers to access the system.
- must have cell-phone web based capability to access the system from a mobile phone.

3.4.3 Hardware Limitations

- Input/Output: One or two-button mouse, keyboard, cell-phone, or touch screen required.
- Network card required at thin-client terminals to make communication with server possible.

3.5 Quality Characteristics

There are a number of quality characteristics that apply to the HMS software system.

3.5.1 Portability

The HMS system will be developed using HTML and Java so that it can be accessed from any type of system using just a regular web browser. It will also be available to users that have web access on their cellular phones. The system will be tested on all types of hardware before being released to ensure that is it compliant with this requirement.

3.5.2 Reliability

The system should be capable of processing a given number of reservations within a give time frame with no errors and the system should be available and operational all the time. During the development of the prototype for the 3 cities, the system will be tested in its actual environment to ensure that it can handle the load of reservations that occur during a regular workday.

3.5.3 Usability

The HMS system will be developed so that it is an easy to use system that requires the least amount of user input possible. Every input will be validated. The user should only have general computer use knowledge. Error messages will be displayed if the user enters an invalid value or tries to access a function without the required permissions. An easy and well-structured user manual will be provided to the CRM and the system will include descriptive help for all operations allowed.

3.5.4 Correctness

The HMS system will be considered correct when the CRM approves the prototype presented and agrees that all the functions they require are implemented as stated in the Software Requirements Specification.

3.5.5 Flexibility

The HMS system should be developed in such a way that it is easily customizable. If new functions are required by CRM, there will be little effort required to update the system to support new cities or new transactions.

3.5.6 Security

The HMS system should not compromise the customer information at any time. The user information will never be sold to other parties and will be kept secure at all times. Users will be authenticated to ensure that no unauthorized users gain access to private information.

3.5.7 Maintainability

The HMS source code will be kept well structure and documented so that it is easier to maintain and extend the system. All changes to the system shall be documented.

3.6 Other Requirements

Certain requirements may, due to the nature of the software, the user organization, etc., be placed in separate categories such as those below.

3.6.1 Data Base

The Automate Railway Reservation System will have two main databases. One is the Reservation Database, and another is the Passenger Account Database. These database will be created with Oracle8i (Client/Server) version 8.1.6.0.0 Release 2. The following are the requirements for these databases that are to be developed as part of the product. They include:

Reservation Database

Types of information	Schedule information for the trains, including date, time, departure city, destination city, ticket cost and ticket availability for a particular train
Frequency of use	Depends on the passenger demand, which may reach 25,000 per day during peak periods
Accessing capabilities	The database should allow access to at least 1,000 people at once; the users will have a

	general access to the information about the train schedule, and a secure access to the reports (available only to CRM officials) using a username and a password
Data element and file descriptions	To be determined
Relationship of data elements, records and files	To be determined
Static and dynamic organization	To be determined
Retention requirements for data	Train schedule information will be available as long as the train for a particular route is in use and at least one year after the train has been cancelled. The reports information will be available at least for 5 years

Passenger Account Database

Types of information	Passenger account information including their name, address, phone numbers, last reservations, balance owed, credit card number (if they paid by a credit card)
Frequency of use	Depends on the passenger demand, which may reach 25,000 per day during peak periods
Accessing capabilities	The database should allow access to at least 500 people at once; the users will have a secure access to the database using a username and a password
Data element and file descriptions	To be determined
Relationship of data elements, records and files	To be determined
Static and dynamic organization	To be determined
Retention requirements for data	Passenger account will be available for as long as a passenger is using the account, and at least for 6 month since the passenger logged on last time.

3.6.2 Operations

The normal operations required by the user can be viewed as the following:

User-initiated Operations:

These operations include the login operation, which is initiated by the users. Also, the process of becoming a new user is in this category. Building, changing, and viewing itineraries, as well as paying for the itinerary are all initiated by the users. The user initiates the report generation activity, as well as changing train schedules.

Interactive Operations and Unattended Operations:

The users initiate all the operations mentioned above, and almost all of them are somehow interactive. Displaying the train schedule is non-interactive. The report display is a non-interactive operation, although selecting the desired reports will require user input.

Data Processing Support Functions:

The user account data is used to create new accounts, as well as to validate user id's during login functions. For building itineraries, user input, user account data, and train schedule data are used, and processed. User data along with final results of user interaction (whether the user purchased a trip, number of tickets bought, etc.) are collected, and used for report generation purposes. Administrative users' inputs are collected in order to modify and present schedules.

Backup and Recovery Operations:

Both databases used (passenger account database and reservations database) are production databases. The main operation used for the backup and recovery is Oracle's built-in cold backup, which is also known as the "archive mode". Depending on the customer's needs and budget, additional redundancy can be added using systems like RAID 5 and tape backup.

3.6.3 Site Adaptation Requirements

There are no site adaptation requirements for this project.

4. Supporting Information.

There is no supporting information required for this project.