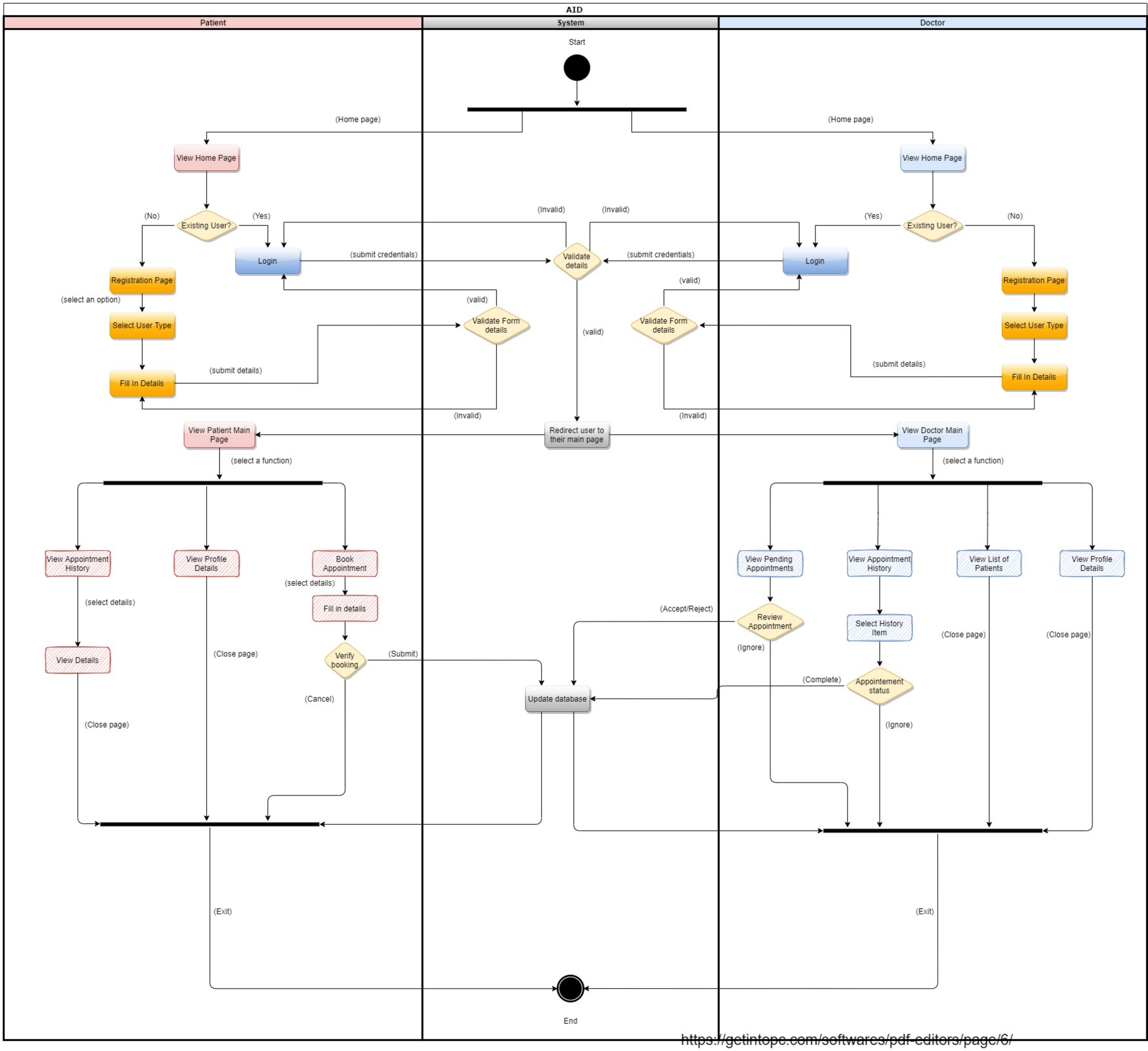
Activity Diagram

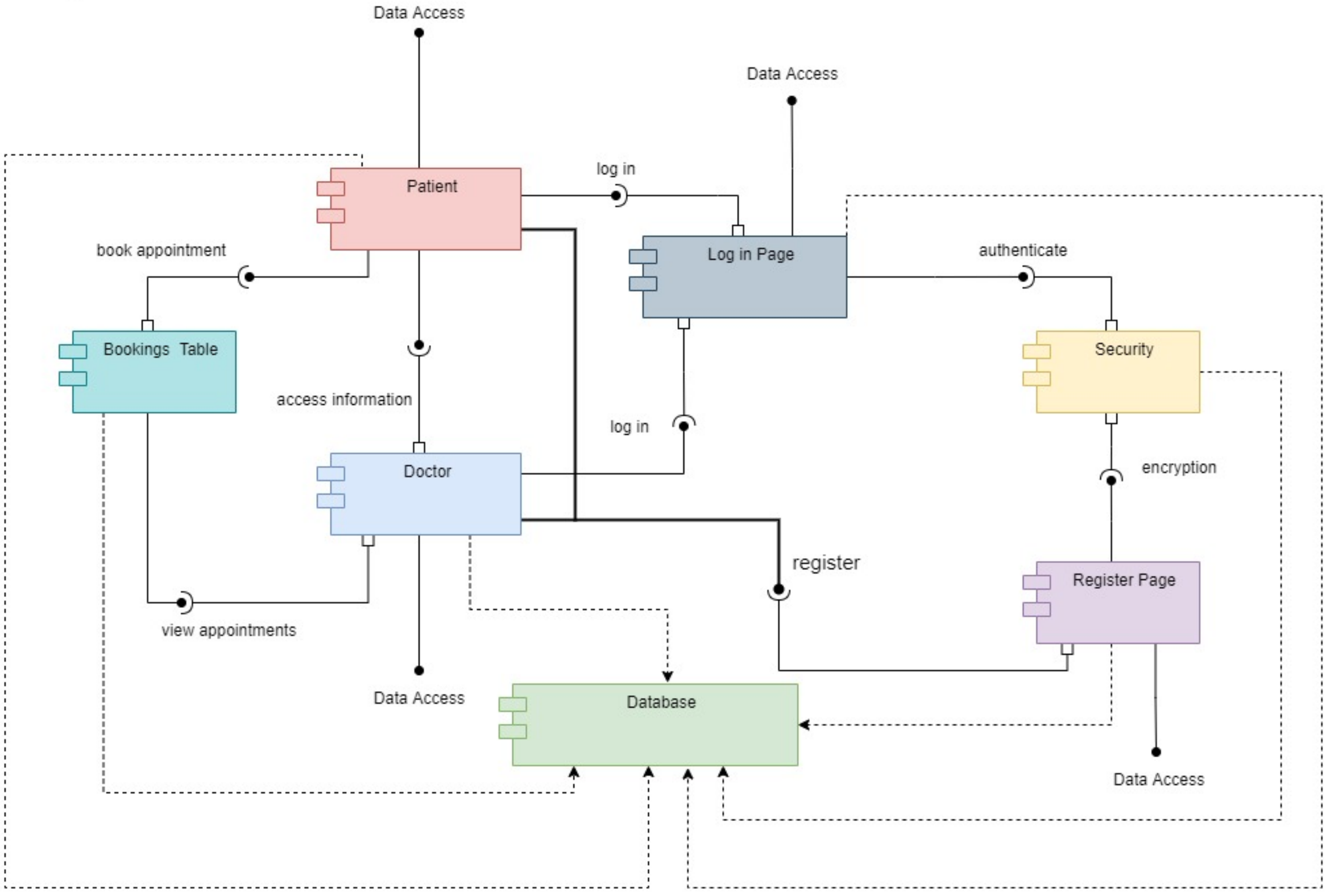


Description

Process view: The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. The process view addresses concurrency, distribution, integrator, performance, and scalability, etc. UML diagrams to represent process view include the sequence diagram, communication diagram, activity diagram.

Here with out activity diagram we have show all the activities that the two users will engage in. we have 3 main streams, which is the patient, doctor, and the system itself or the application. Both user are able to log in or create new accounts if they don't have one yet. They all have functionalities that are provided for their respective profiles. The doctor can view pending appointments, appointment history, list of patients and view their profile details. the patient can view their profile details, view appointment history and make a new booking. The system also illustrate how it interacts with actions caused by these two users. the system also redirects and validate passwords and help validate certain things like form fields. It also updates/ alter the table in the database.

Component Diagram



Description

The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML Component diagram to describe system components.

In our diagram we have a mixture of interfaces, databases, tables and our concerned 2 users.

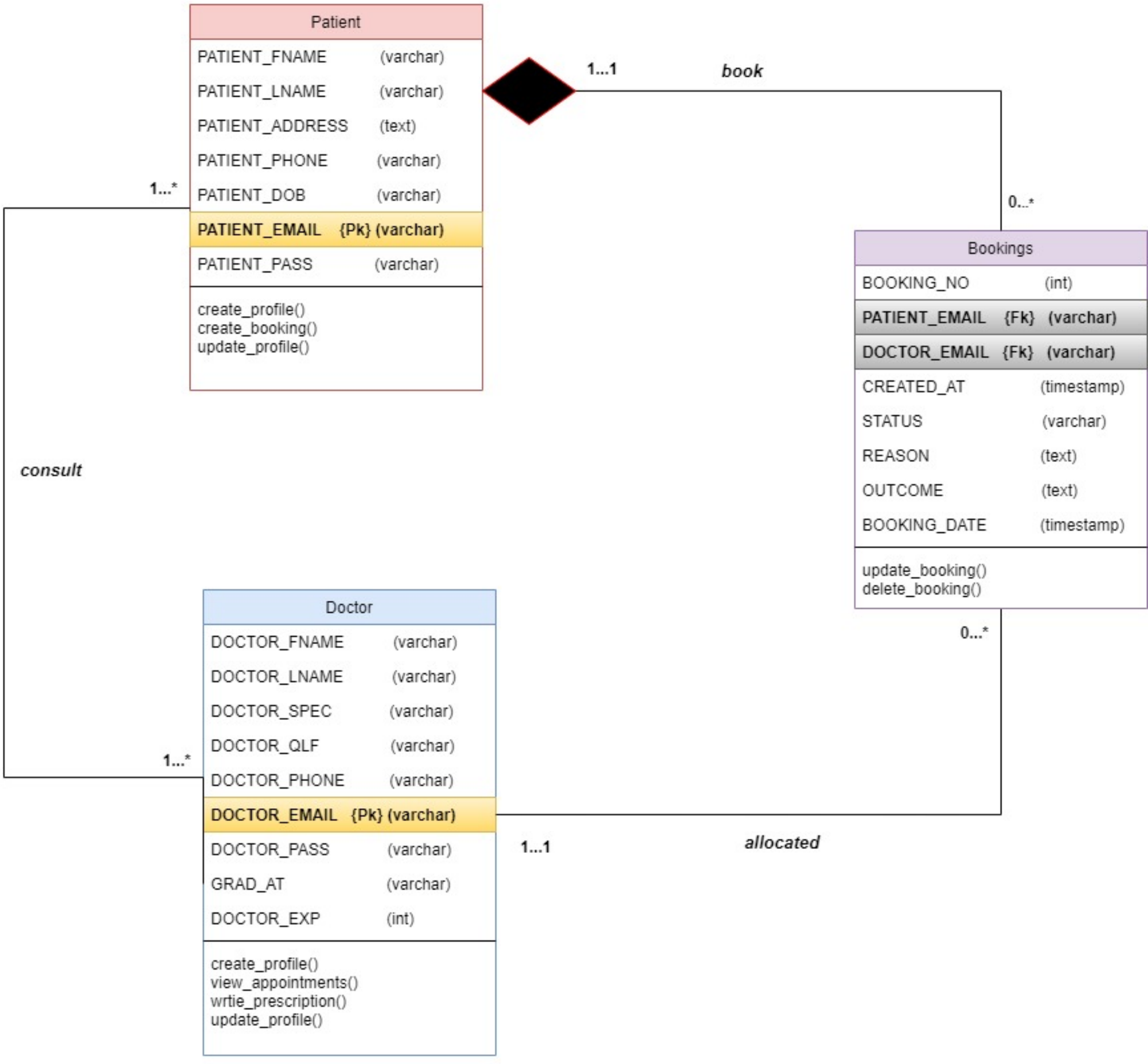
A doctor can can access information from a patient such as viewing their details and getting information. The patient provides the data and the doctor access it. Both users for log in they provided the app interface with their data to login.

A patient is able to book appointments, hence they will have to put in appointment information requested to be stored in a bookings table. The table request the data from the patient via the interface.

For registering both users will have to give information to the registration page which and the information from the registration page will have to be feed to the security interface in the backend where passwords are encrypted

https://en.wikipedia.org/wiki/4%2B1_architectural_view_model

Class Diagram



Description

Logical view: The logical view is concerned with the functionality that the system provides to end-users. UML diagrams are used to represent the logical view, and include class diagrams, and state diagrams.[2]

In total we have only 3 tables in our database. as illustrated by the diagram the primary keys and foreign keys have been created. The doctor table contains all information related to doctors and also the patient table contains all information related to the patient. The bookings tables has all the created appointments and a super key containing 2 foreign keys from all users.

Relationships.

- 1 and 1 only patient can make a single booking.0 to many bookings can be made by 1 and only 1 patient.

This means that multiple patients can not be assigned to the same booking.

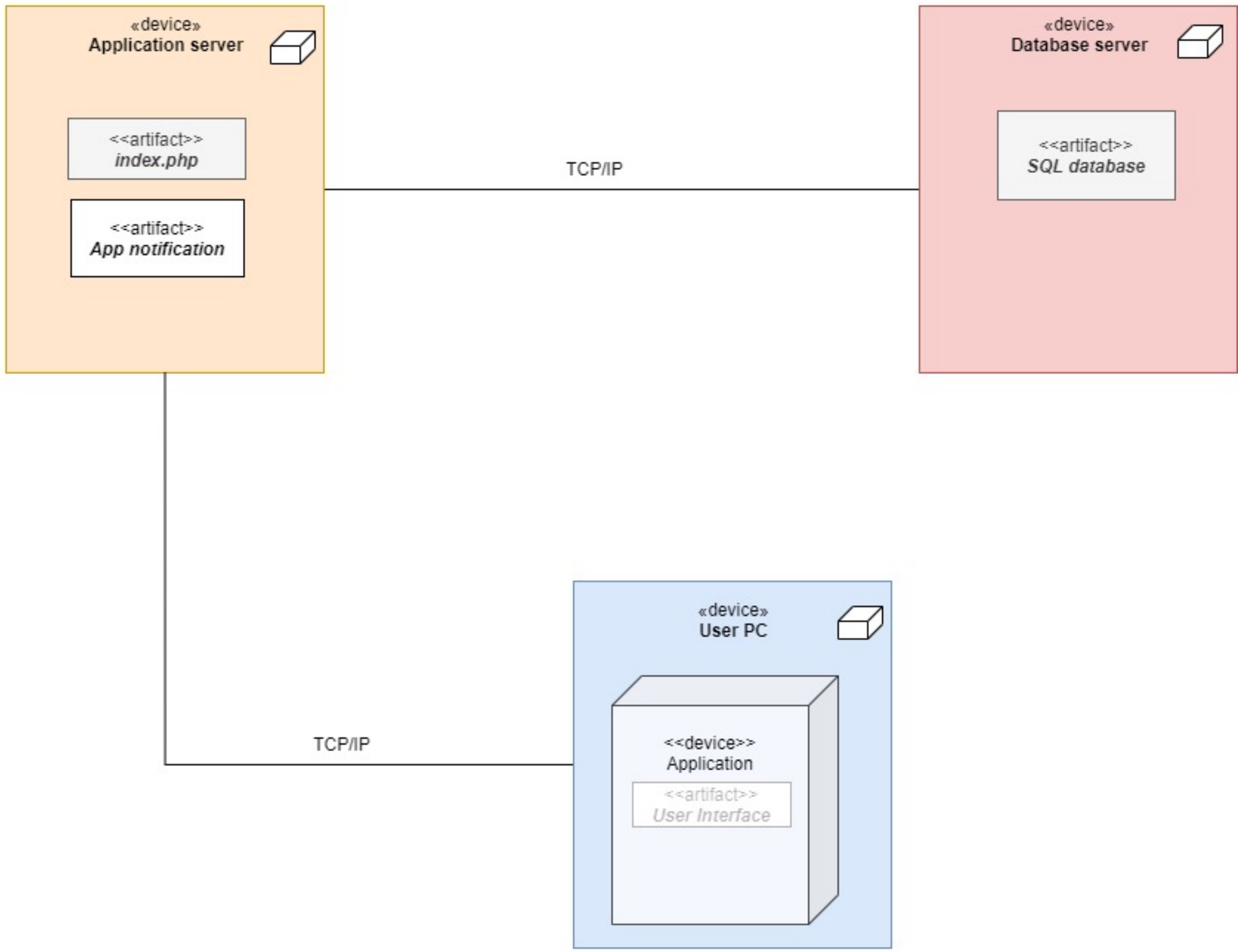
- 1 and only 1 doctor can be allocated to 0 or many bookings. 0 to many bookings can be allocated to one and only one doctor.

This means that multiple doctors can not be assigned to the same allocated booking.

1 and many patients can consult to 1 and many doctors. **!** to many doctors can consult to 1 and many patients.

This means that a patient can consult to more than one different doctor, also vice versa is also true

Deployment Diagram



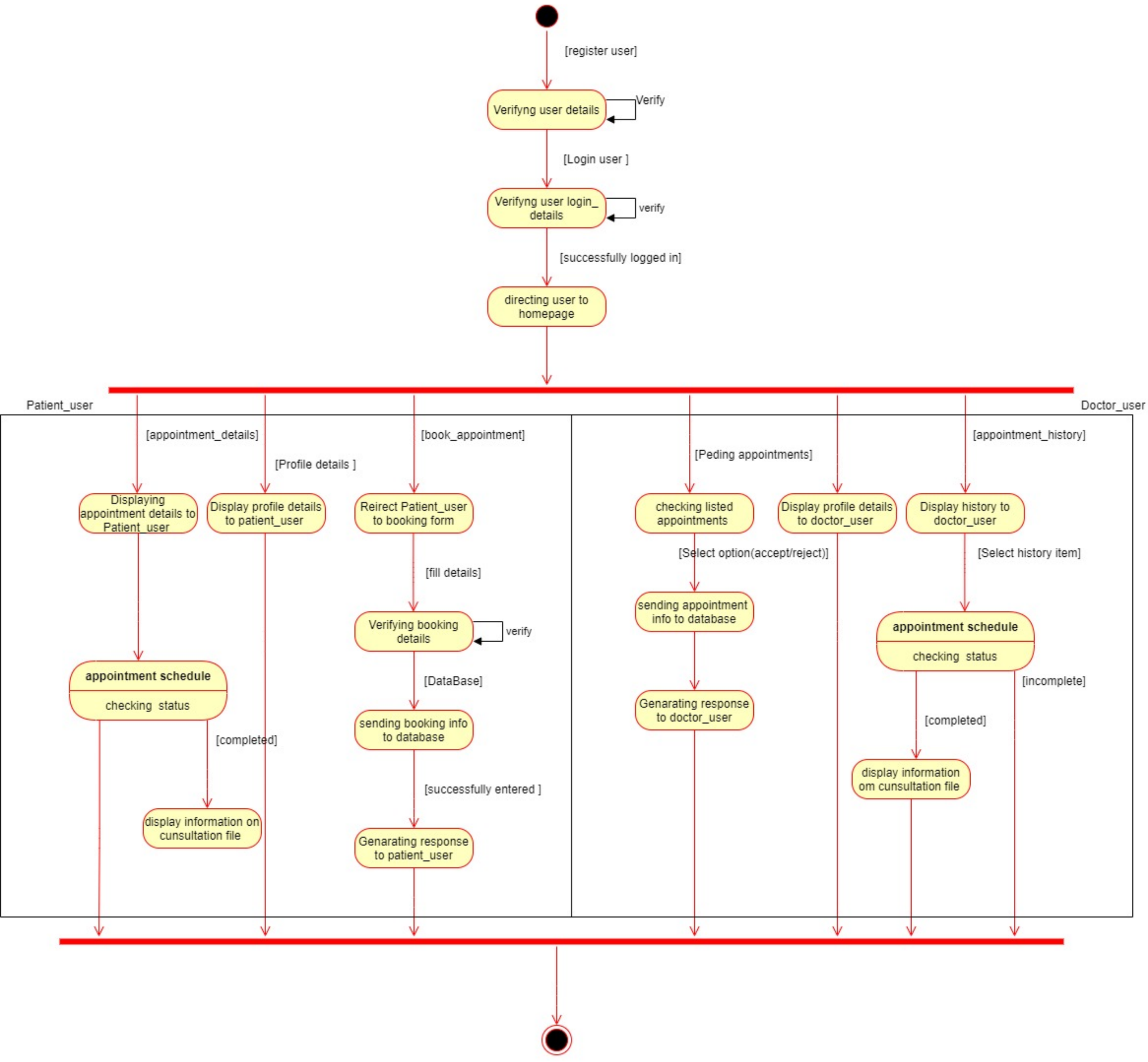
Description

The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer as well as the physical connections between these components. This view is also known as the deployment view. UML diagrams used to represent the physical view include the deployment diagram.

In our diagram e have 3 core aspects. We have the main database which is where we store all our data. Via a TCP/IP connection we have the databse communicating with the application server. This is where we have the application being rendered on. The finally we have the end user device. This is where the application will be installed and run at with the Gpu of the device and its hardware in general.

https://en.wikipedia.org/wiki/4%2B1_architectural_view_model

State diagram

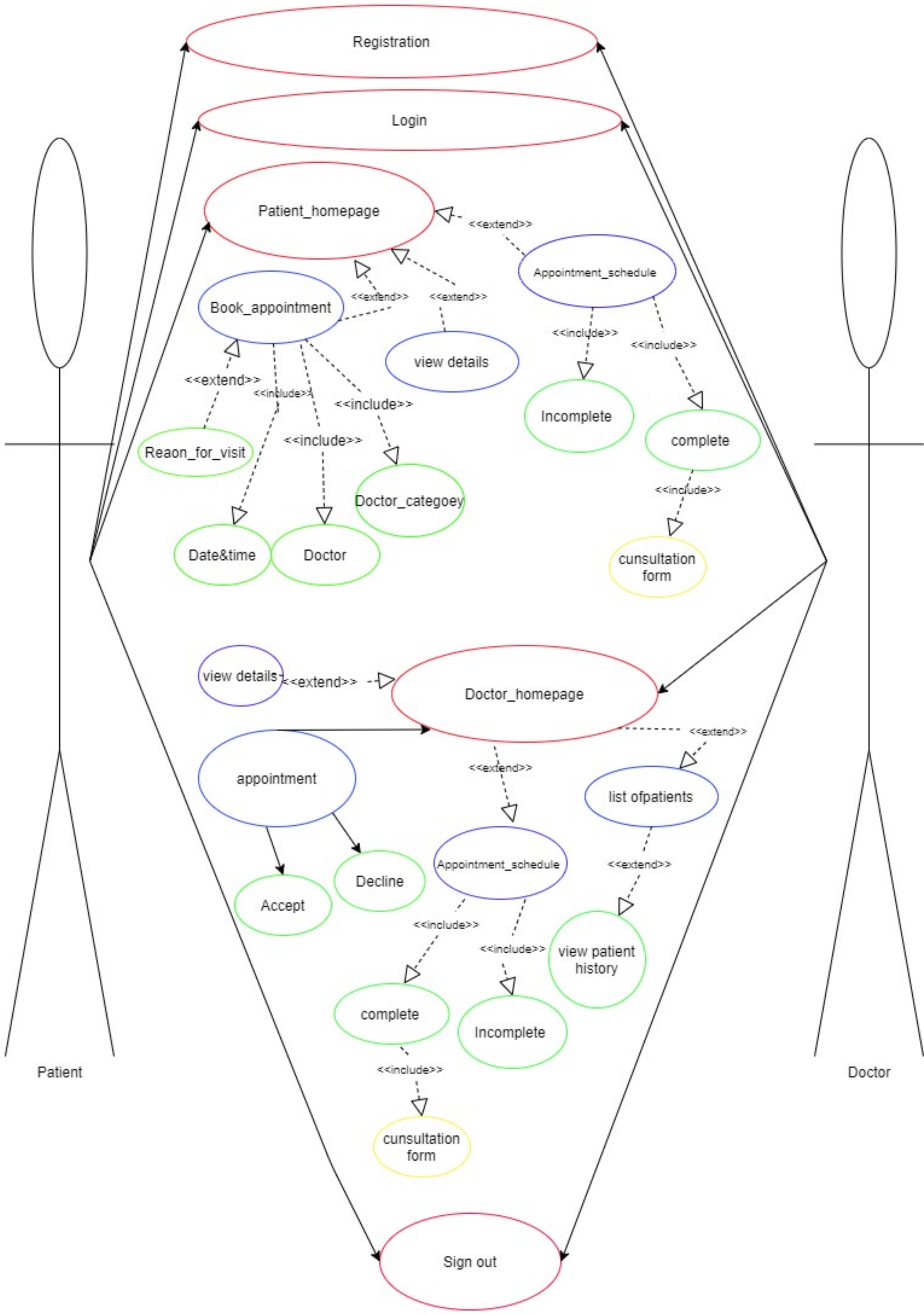


Description

Process view: The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. The process view addresses concurrency, distribution, integrator, performance, and scalability, etc. UML diagrams to represent process view include the sequence diagram, communication diagram, activity diagram

https://en.wikipedia.org/wiki/4%2B1_architectural_view_model

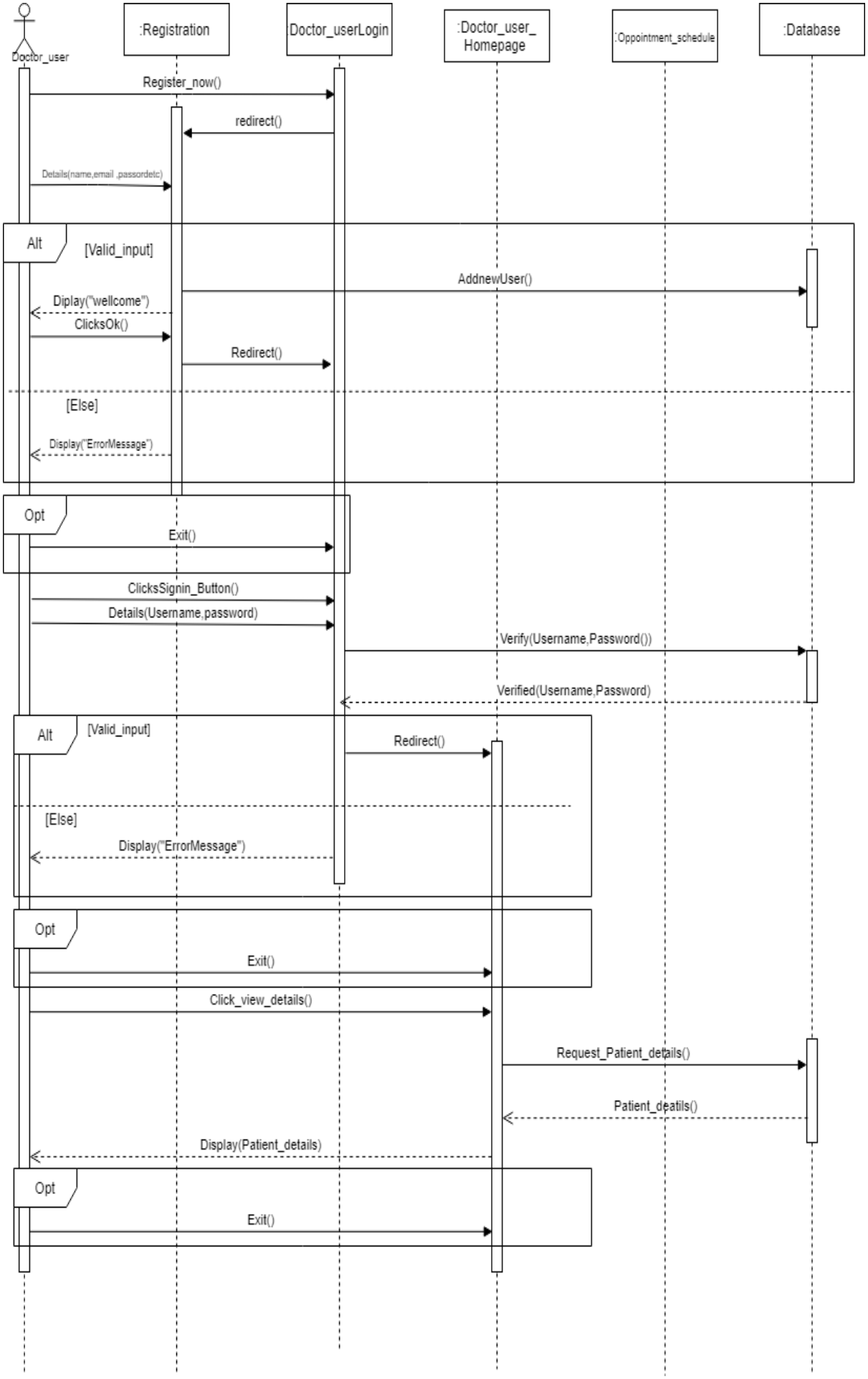
Use Case diagram

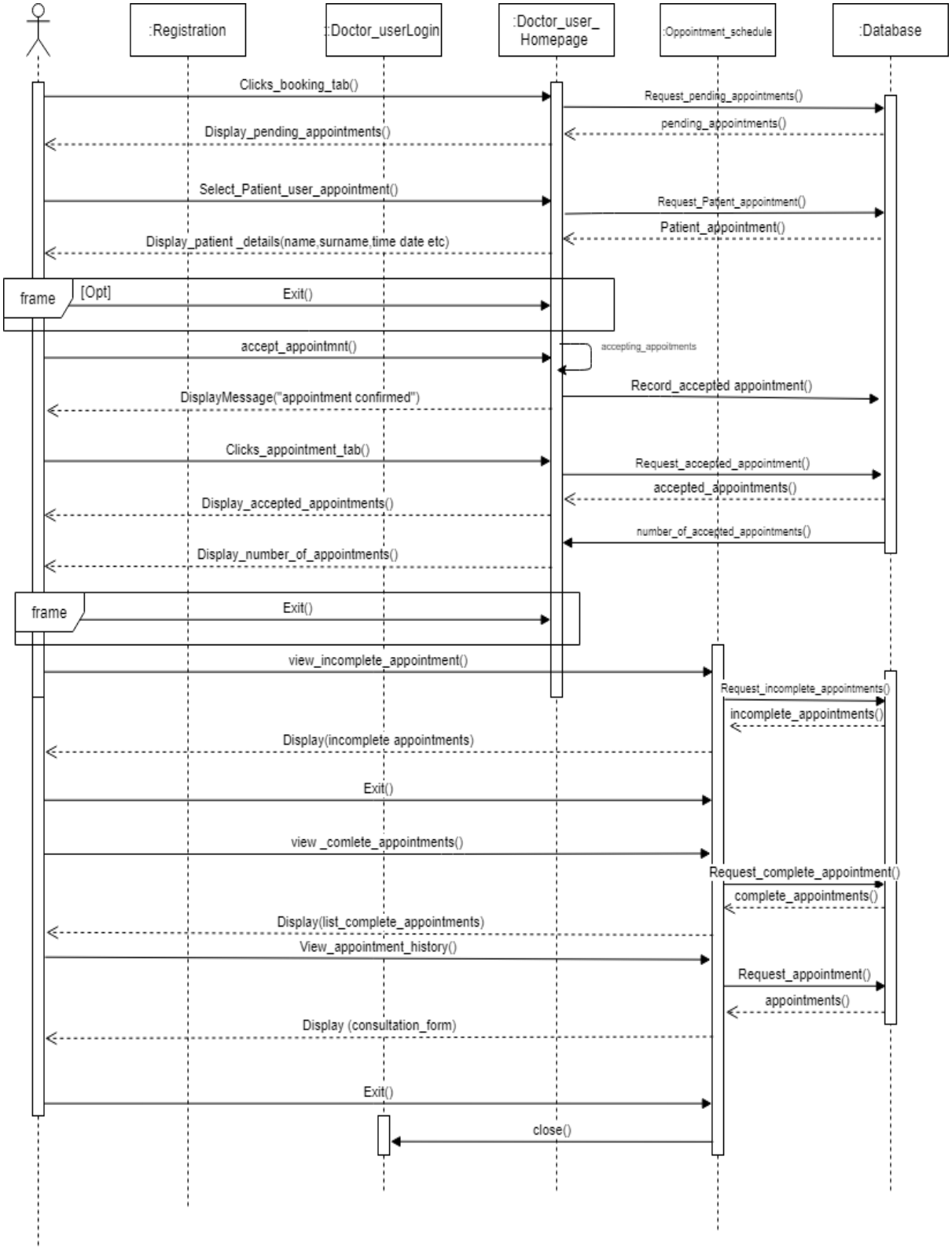


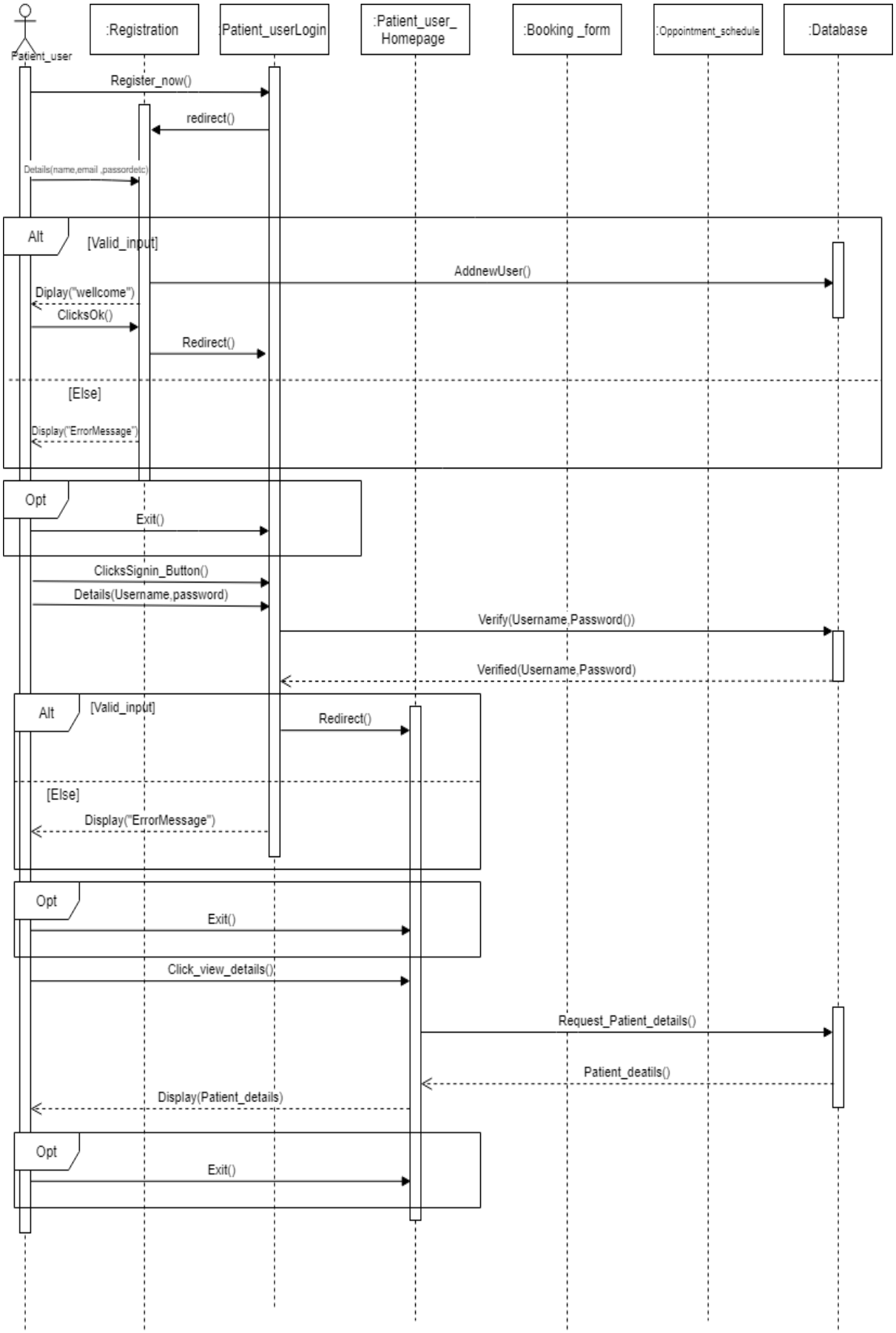
Description

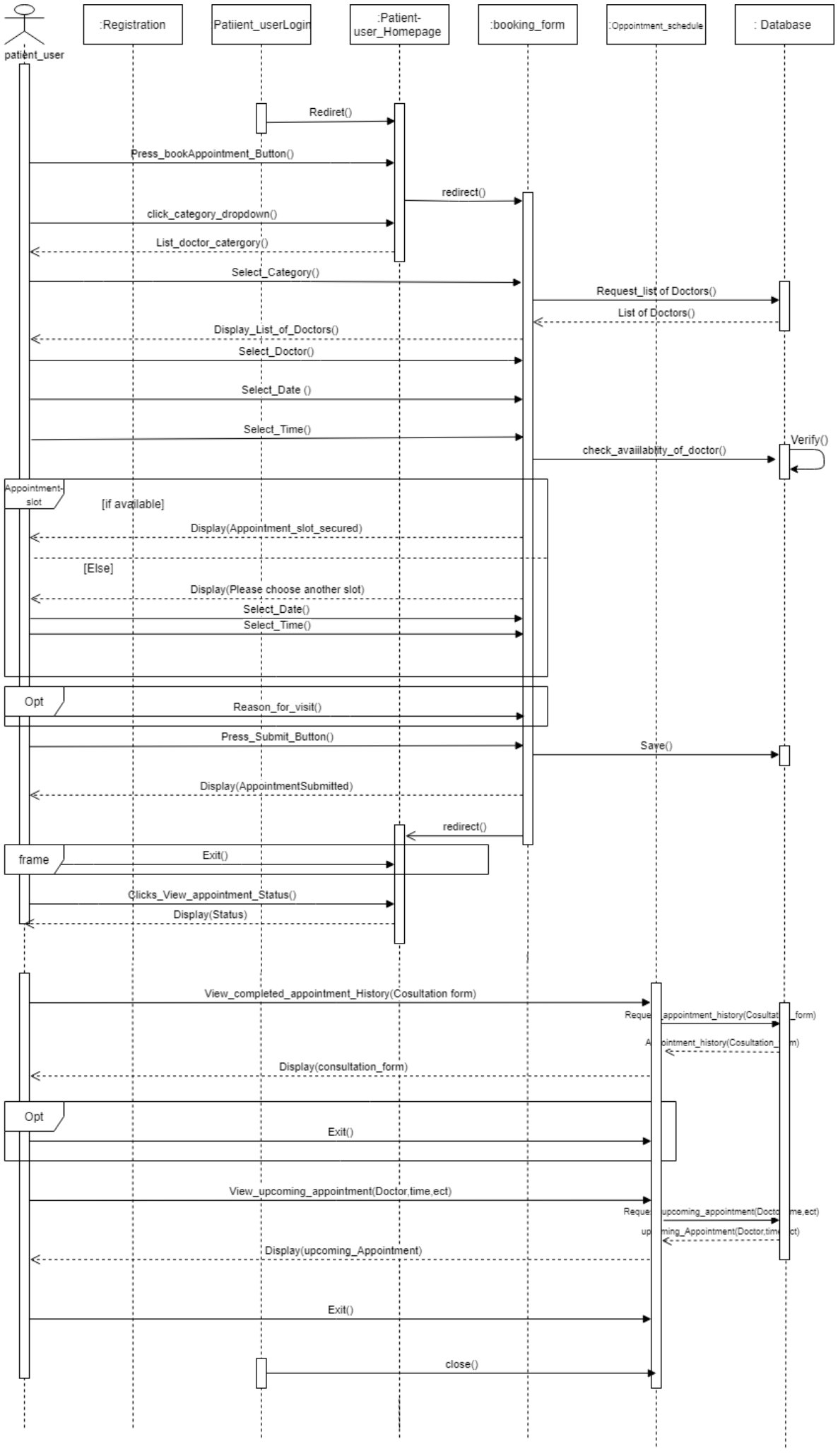
.***Scenarios***: The description of an architecture is illustrated using a small set of use cases, or scenarios, which become a fifth view. The scenarios describe sequences of interactions between objects and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design. They also serve as a starting point for tests of an architecture prototype. This view is also known as the **use case view**.

https://en.wikipedia.org/wiki/4%2B1_architectural_view_model


Doctor_user
:Registration
Doctor_userLogin
:Doctor_user_ Homepage
:Oppointment_schedule
:Database
Register_now()
redirect()
Details(name,email ,passordetc)
Alt
[Valid_input]
AddnewUser()
Diplay("wellcome")
ClicksOk()
Redirect()
[Else]
Display("ErrorMessage")
Opt
Exit()
ClicksSignin_Button()
Details(Username,password)
Verify(Username,Password())
Verified(Username,Password)
Alt
[Valid_input]
Redirect()
[Else]
Display("ErrorMessage")
Opt
Exit()
Click_view_details()
Request_Patient_details()
Patient_deatils()
Display(Patient_details)
Opt
Exit()


:Registration
:Doctor_userLogin
:Doctor_user_ Homepage
:Oppointment_schedule
:Database
Clicks_booking_tab()
Request_pending_appointments()
pending_appointments()
Display_pending_appointments()
Select_Patient_user_appointment()
Request_Patient_appointment()
Patient_appointment()
Display_patient _details(name,surname,time date etc)
frame
[Opt]
Exit()
accept_appointmnt()
accepting_appoitments
Record_accepted appointment()
DisplayMessage("appointment confirmed")
Clicks_appointment_tab()
Request_accepted_appointment()
accepted_appointments()
Display_accepted_appointments()
number_of_accepted_appointments()
Display_number_of_appointments()
frame
Exit()
view_incomplete_appointment()
Request_incomplete_appointments()
incomplete_appointments()
Display(incomplete appointments)
Exit()
view _comlete_appointments()
Request_complete_appointment()
complete_appointments()
Display(list_complete_appointments)
View_appointment_history()
Request_appointment()
appointments()
Display (consultation_form)
Exit()
close()


Patient_user
:Registration
:Patient_userLogin
:Patient_user_ Homepage
:Booking _form
:Oppointment_schedule
:Database
Register_now()
redirect()
Details(name,email ,passordetc)
Alt
[Valid_input]
AddnewUser()
Diplay("wellcome")
ClicksOk()
Redirect()
[Else]
Display("ErrorMessage")
Opt
Exit()
ClicksSignin_Button()
Details(Username,password)
Verify(Username,Password())
Verified(Username,Password)
Alt
[Valid_input]
Redirect()
[Else]
Display("ErrorMessage")
Opt
Exit()
Click_view_details()
Request_Patient_details()
Patient_deatils()
Display(Patient_details)
Opt
Exit()


patient_user
:Registration
Patiient_userLogin
:Patient-user_Homepage
:booking_form
:Oppointment_schedule
: Database
Rediret()
Press_bookAppointment_Button()
redirect()
click_category_dropdown()
List_doctor_catergory()
Select_Category()
Request_list of Doctors()
List of Doctors()
Display_List_of_Doctors()
Select_Doctor()
Select_Date ()
Select_Time()
Verify()
check_availiability_of_doctor()
Appointment-slot
[if available]
Display(Appointment_slot_secured)
[Else]
Display(Please choose another slot)
Select_Date()
Select_Time()
Opt
Reason_for_visit()
Press_Submit_Button()
Save()
Display(AppointmentSubmitted)
redirect()
frame
Exit()
Clicks_View_appointment_Status()
Display(Status)
View_completed_appointment_History(Cosultation form)
Display(consultation_form)
Opt
Exit()
View_upcoming_appointment(Doctor,time,ect)
Display(upcoming_Appointment)
Exit()
close()