Online Health Care Application

Project Description:

We have built an online healthcare application to book doctor and diagnostic lab appointments and manage digital health records. The application allows patients to search for the doctors in their vicinity as per specialization. It allows patients to look at available slots and book an appointment when required. The application also helps patients to digitally store all their medical records and associated history in a digital health record**.**

Each **location** has a number of **hospitals**. Hospitals have many **departments** based on the specialization: Dentist, Ophthalmologist, Dermatologist, Psychiatrist, Ear-Nose-Throat (ENT), Gynaecologist, Neurologist, Urologist etc. Multiple **doctors** work at each hospital departments. **Patients** book **doctor's appointment** online. Patients can book **diagnostic lab appointment**. Patients maintain a **digital health record** which can be accessed and updated by doctors and **diagnostic lab attendant**. The digital health record contains the medical history of the patient including previous diagnostic lab **test results** and previous doctor's **prescriptions**. The application also has an online **forum** where the patients can ask **health related questions** and doctors can give **answers**.

Patients can **select location** and search for the doctors in their vicinity. They can **select hospital** and department and choose from available doctors. Patients can **book doctor's appointment**. Doctor accesses patient's digital health record and can **recommend a diagnostic lab test**. Doctors can access digital health record and **post prescriptions**. Patient can search local diagnostic labs and **book diagnostic lab appointments**. Lab attendant **performs test** on patients and **posts the test results** to patient's digital health record. Patients can **give reviews** about their experience with each hospital or doctor. Patients can **ask health related questions** and doctors **give answers.**

Actors	**Use Cases**
Patient	Book doctor's appointment
Patient	Book diagnostic lab appointment
Patient	Gives review about doctors
Patient	Ask health related questions
Doctor	Access patient's digital health record and post prescriptions
Doctor	Access patient's digital health record and recommend diagnostic lab tests
Doctor	Gives answers to patient's health related questions
Lab attendant	Performs test on patients
Lab attendant	Posts the test results to patient's digital health record

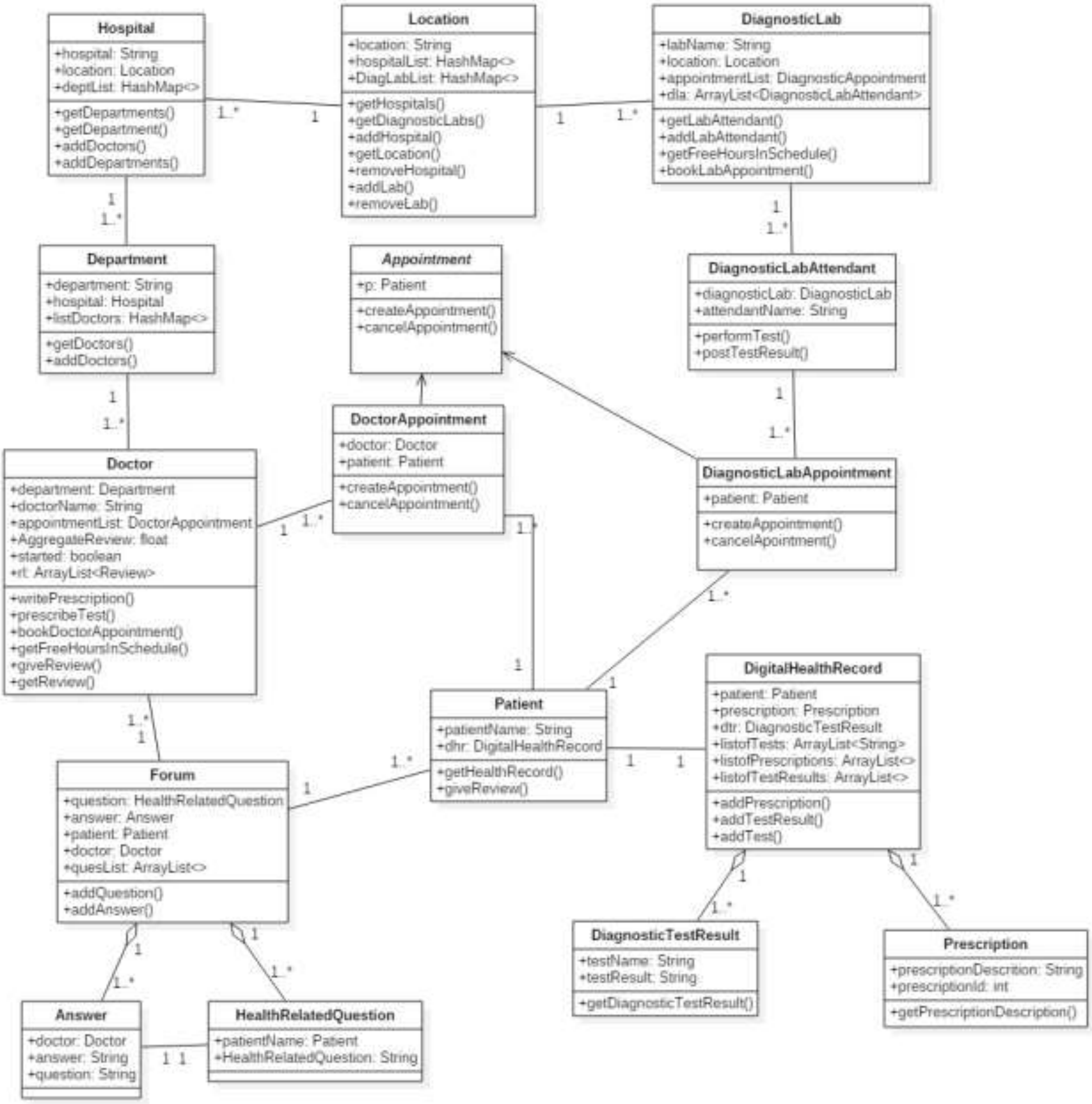
Operating Environment:

The system should allow only authorised personnel to access and modify patient information. Only doctors and patients can view the patient's digital health record. Only doctors can recommend test or post prescriptions to the digital health record. Lab attendants can only post test results to the patient's digital health record.

Ease-of-use:

Patient’s digital health records can be accessed anytime and thus it should be on a server that is always up and running (reliable) and they are likely to be accessed frequently and hence should have low response time.

Final UML Diagram:



Final Use Cases:

Use Case	Description
UC1	Patient book doctor’s appointment
UC2	Patient book diagnostic lab appointment
UC3	Patient Gives review about doctors
UC4	Patient ask health related questions on the Health Forum
UC5	Doctors access patient’s digital health record and post prescriptions
UC6	Doctors access patient’s digital health record and recommend diagnostic lab tests
UC7	Doctors give answers to patient’s health related questions
UC8	Lab Attendant performs test on patients and Posts the test results to patient’s digital health record

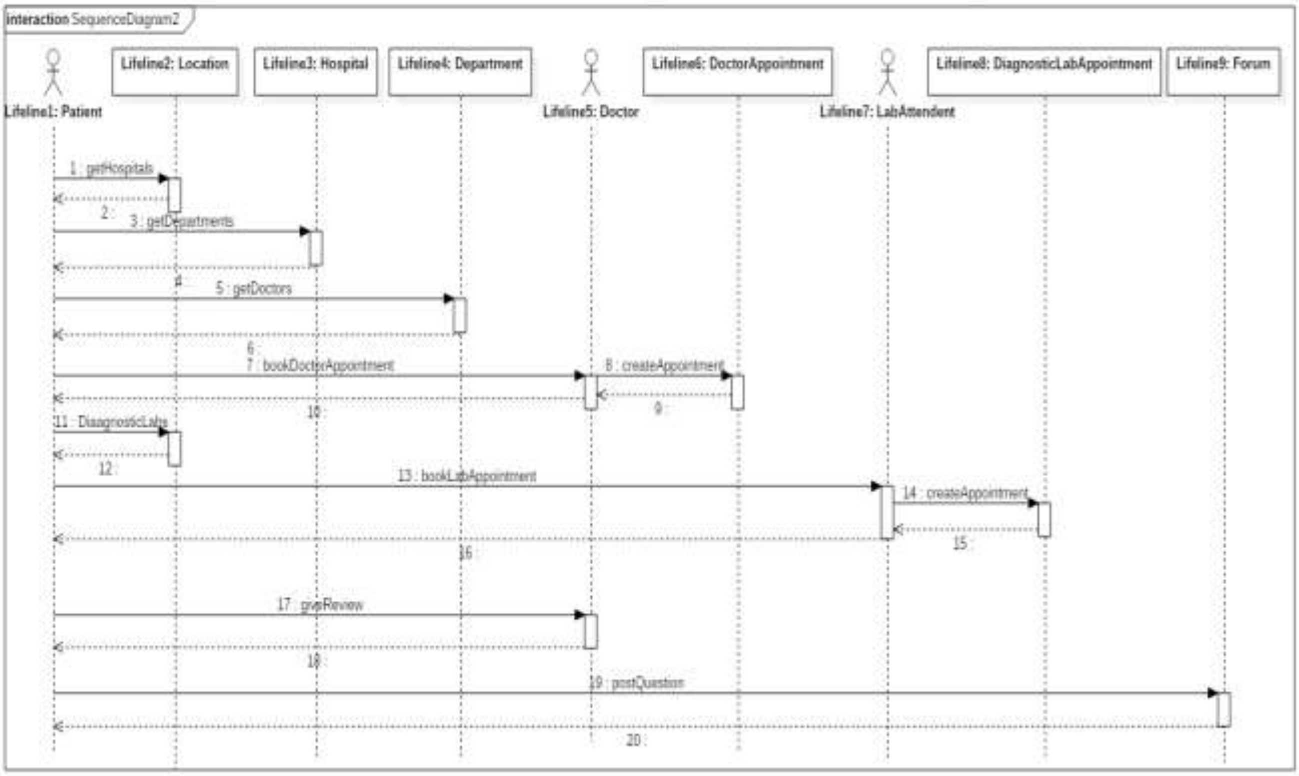
	Patient	Location	Hospital	Department	Doctor	DiagnosticLab
UC1	x	getLocation, getHospitals	getDepartments	getDoctors	getFreeHoursSchedule, bookDoctorAppintment	x
UC2	x	getLocation, getDiagnosticLabs	x	x	x	getFreeHoursSchedule,getLabAttendant
UC3	x	x	x	x	giveReview	x
UC4	x	x	x	x	x	x
UC5	getHealthRecord	x	x	x	writePrescription	x
UC6	getHealthRecord	x	x	x	prescribeTest	x
UC7	x	x	x	x	x	x
UC8	x	x	x	x	x	x

	DiagnosticLabAttendant	DoctorAppointment	DiagnosticLabAppointment	DigitalHealthRecord	Forum
UC1	x	createAppointment,cancelAppointment	x	x	x
UC2	x	x	bookAppointment,cancelAppointment	x	x
UC3	x	x	x	x	x
UC4	x	x	x	x	addQuestion
UC5	x	x	x	addPrescription	x
UC6	x	x	x	addTest	x
UC7	x	x	x	x	addAnswer
UC8	performTest, postTestResult	x	x	addTestResult	x

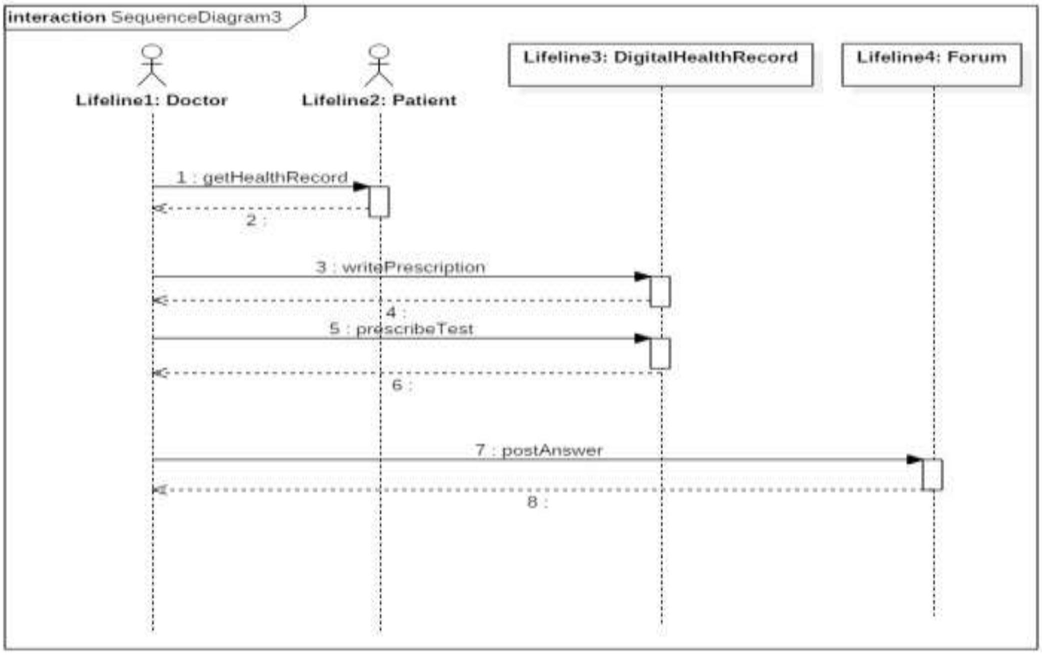
Design Time Sequence Diagram:

Given below are the list of sequence diagrams of all the combined use cases for each actor.

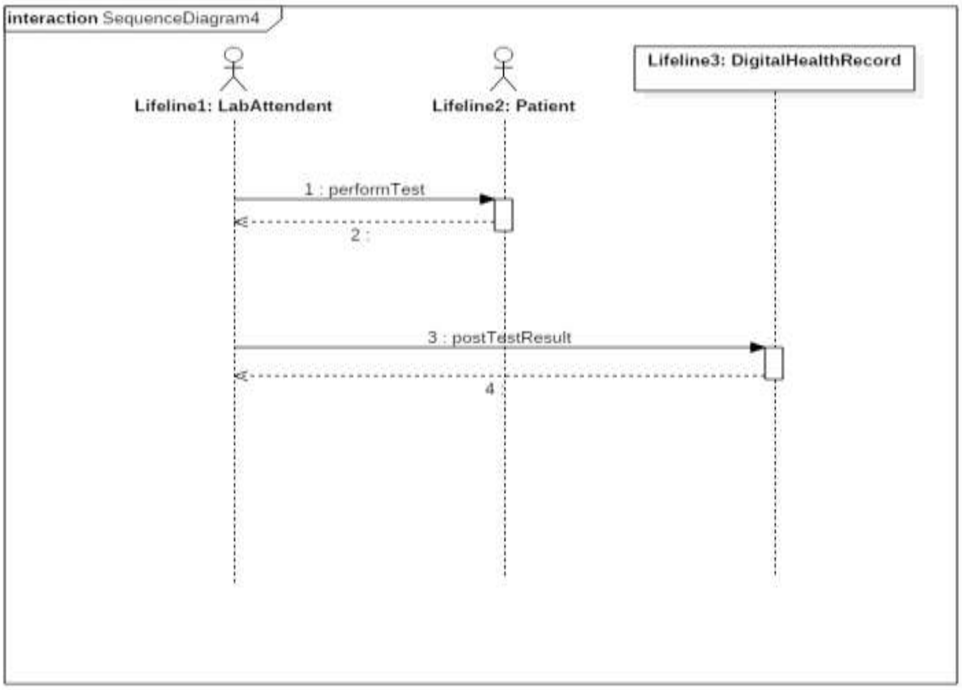
Actor – Patient:



Actor – Doctor:



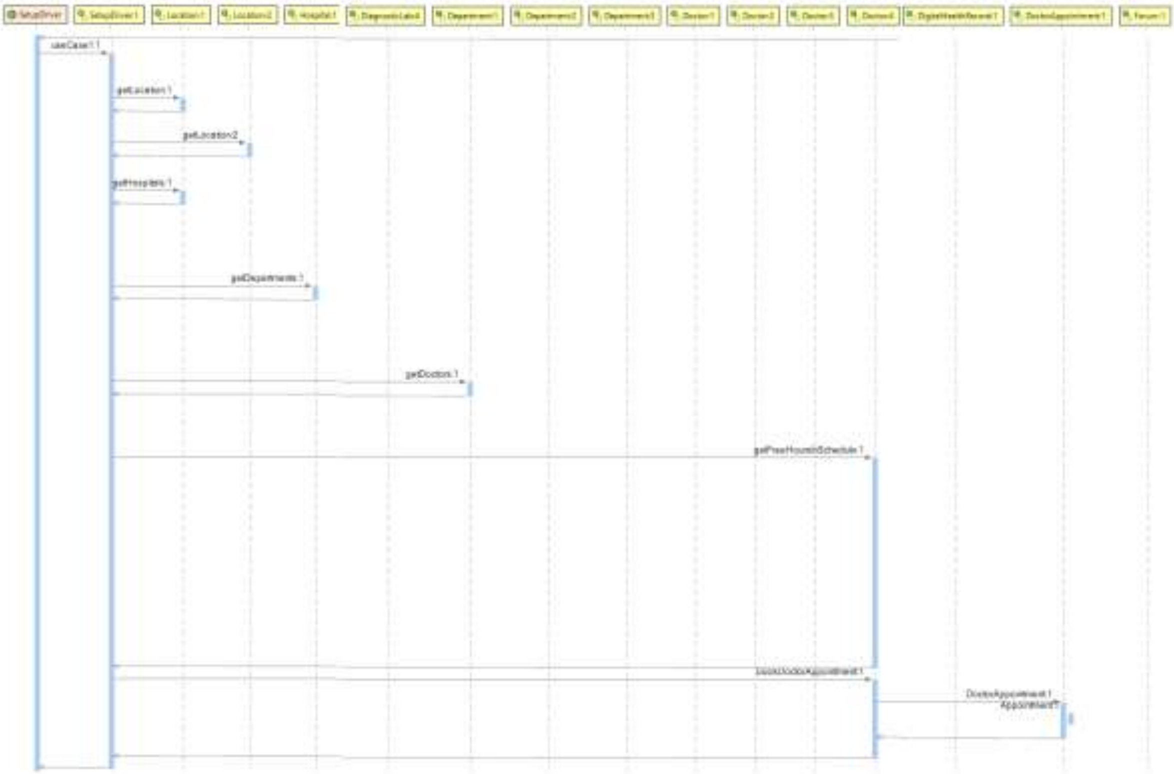
Actor – Diagnostic Lab Attendant



Current Sequence Diagrams:

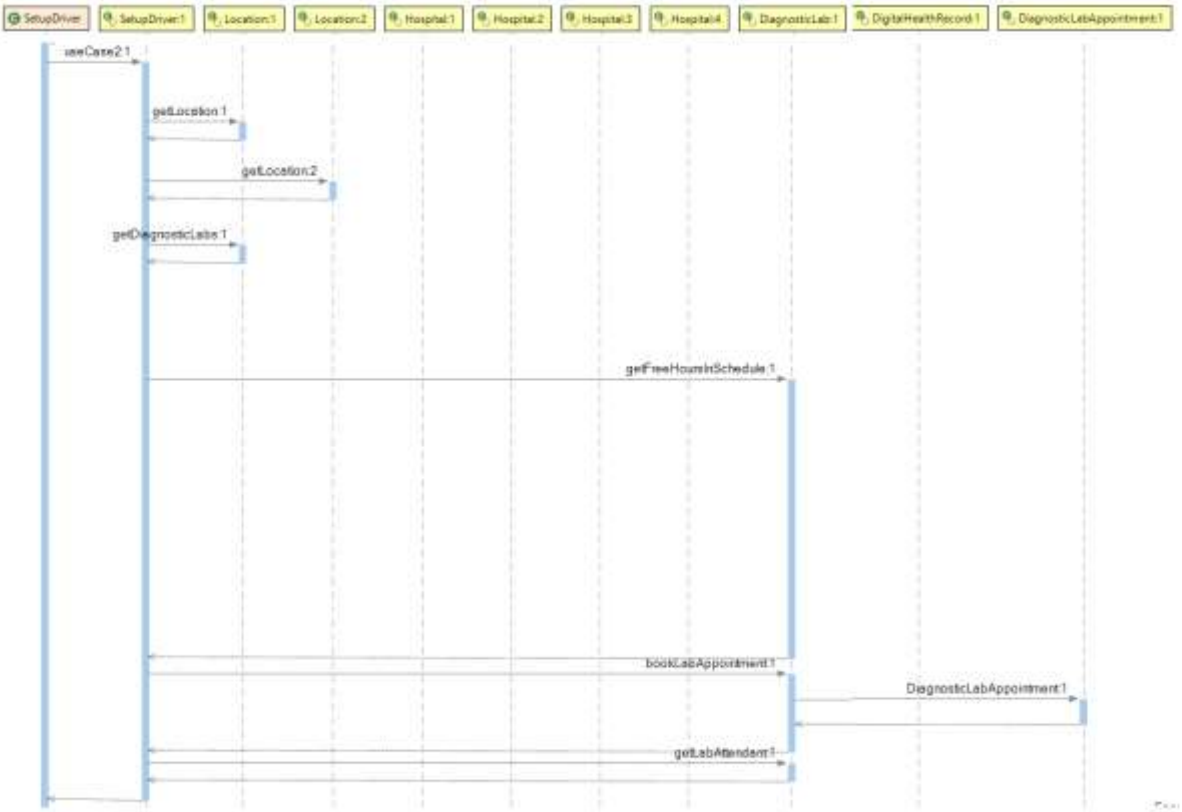
Use Case 1:

Patient check for availability of a doctor of his choice and books an appointment



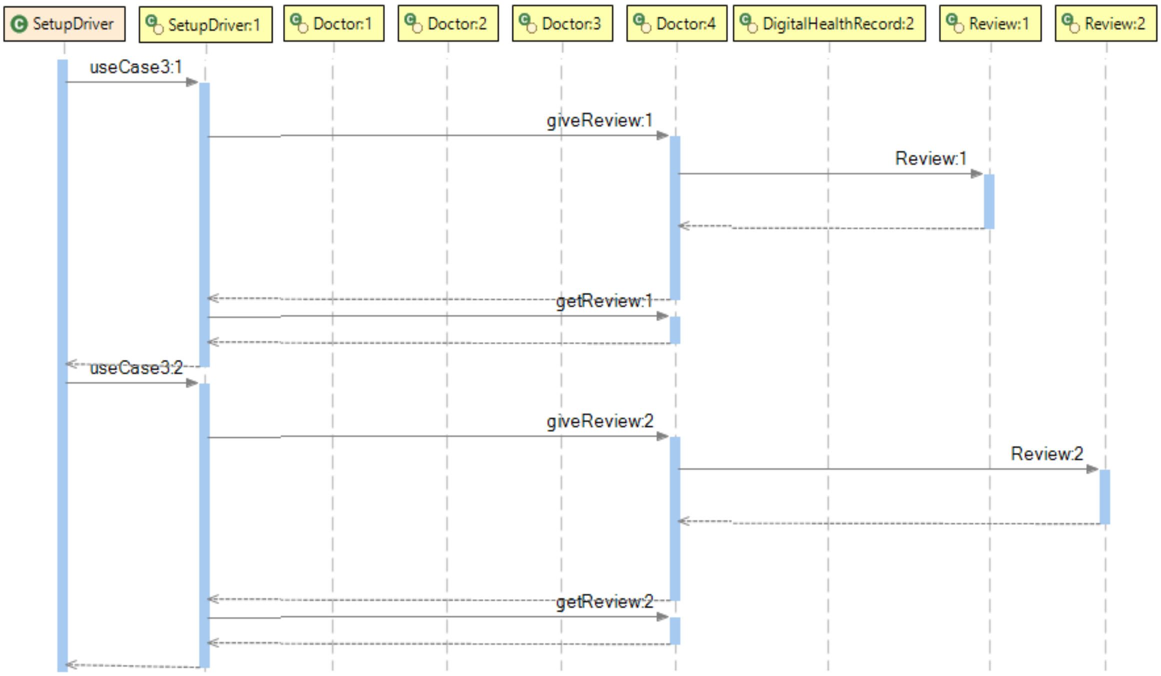
Use Case 2:

Patient looks for a Diagnostic lab of his/her choice and books an appointment



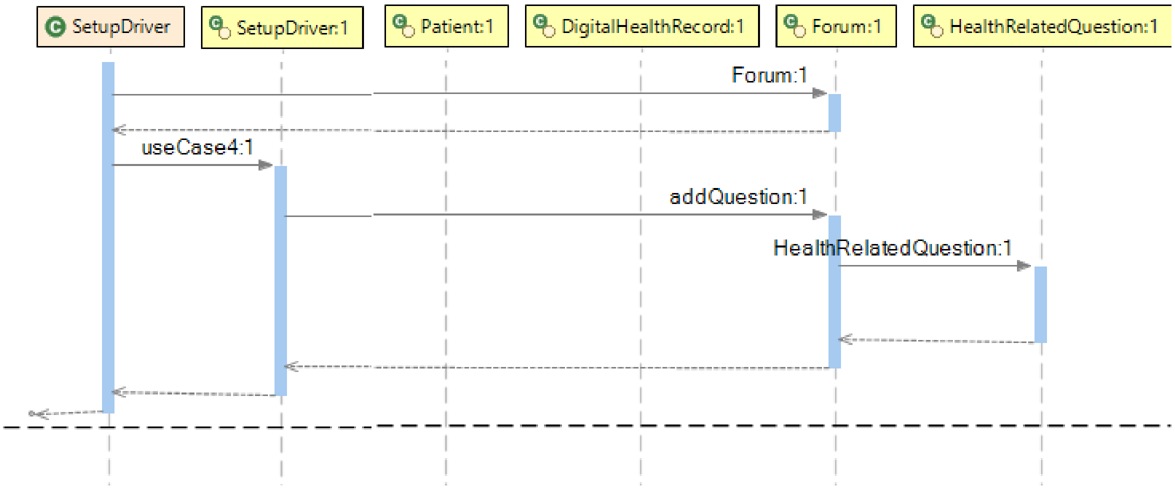
Use Case 3:

Patient gives a review about the doctor he/she visited



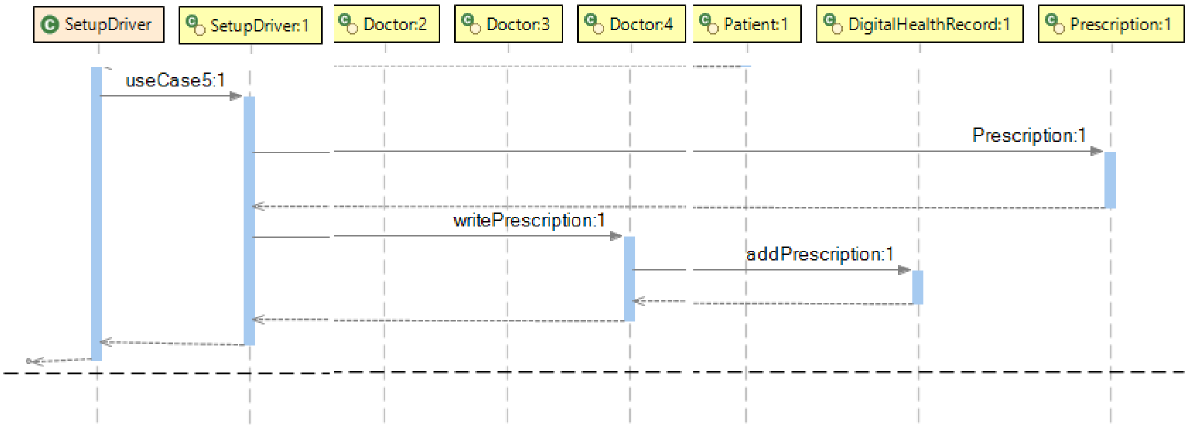
Use Case 4:

Patients ask Health related Questions in a forum for the doctors to answer.



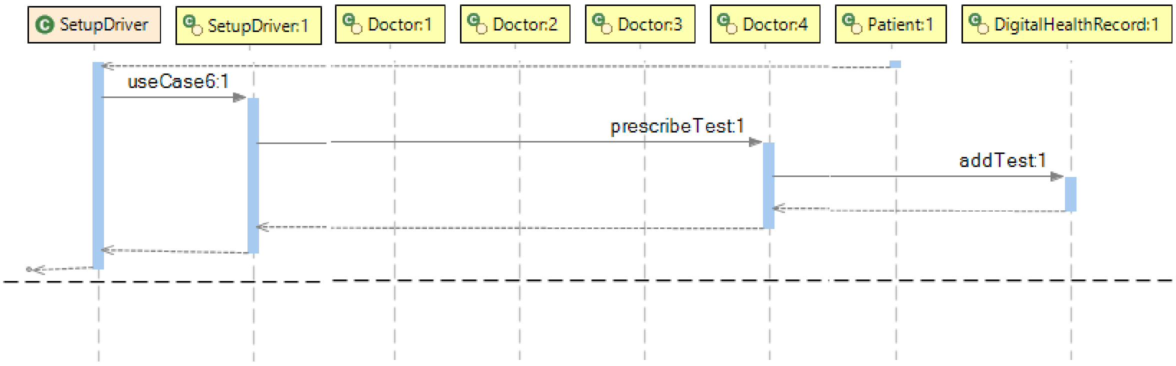
Use Case 5:

Doctors access patient’s digital health record and post prescriptions



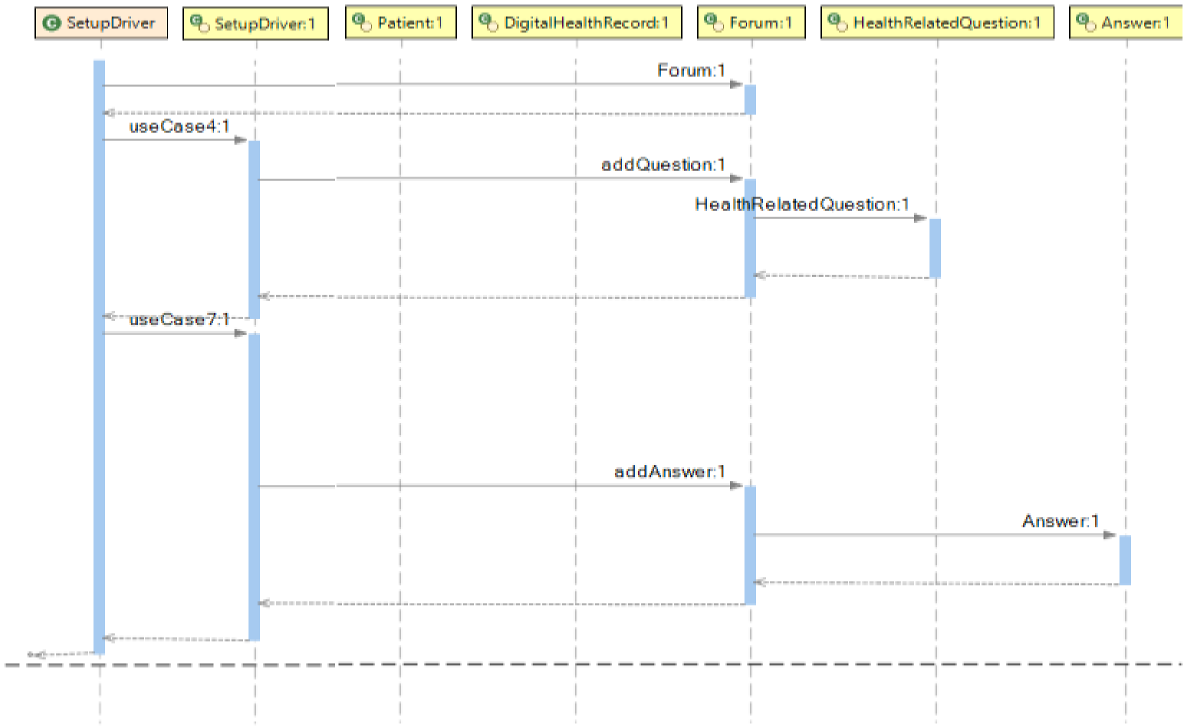
Use Case 6:

Doctors access patient’s digital health record and recommend diagnostic lab tests



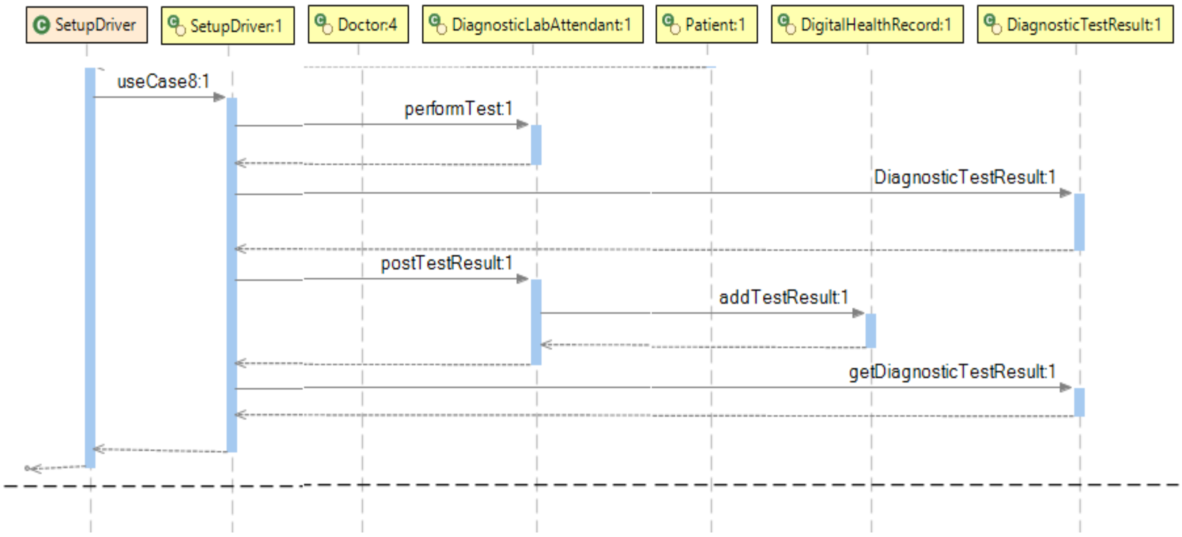
Use Case 7:

Doctor gives answers to patient’s health related questions



Use Case 8:

Lab attendant performs test on patients and posts the test results to patient’s digital health record.



Use-Case driven methodology

- Requirements evolve from the abstract to the very concrete. It helps us in removing ambiguity and preparing a clear picture of the requirements and corresponding implementations needed.
- Through the structural analysis we can foresee difficult problems we might face while development and plan accordingly.
- Identifying exceptions to a successful scenario early in the project saves a lot of time by finding subtle requirements.
- One important benefit of use case driven analysis is that it helps manage complexity, since it focuses on one specific usage aspect at a time.
- Use cases also encourage designers like us to envision outcomes before attempting to specify outcomes, and thereby they help to make requirements more proactive in system development.
- Depending on the size of the system in question, it can take a significant amount of time to complete a set of use cases. But the benefits that we reap from this approach far outweigh the efforts we put in.

Conclusion Remarks:

On a whole, after implementing use case driven methodology, we feel that such a designing paradigm is definitely useful for all the above mentioned reasons and we can honestly say it helped us to achieve our objectives in very structured and organised way.