Software Requirements and Design Document

for

Enigma Hospital Management System

By Saad Bin Farooq & Umer Mukhtar

FAST NUCES

27/11/202

1. Introduction

1.1.Purpose

This document contains details specifications for the Enigma hospital management system version 1.0. The domain of the project is limited to hospitals, hospitals with multiple branches can use the same system. This SRS covers all parts of the system.

1.2. Product Scope

The project covers doctor and patient management, but will be expanded to cover emergency, ICU, and pharmacy management as well. Except for a few numbers of hospitals, most hospital management systems are limited to only some of these features while a vast majority of hospitals in Pakistan, particularly government hospitals, rely on simplistic, basic management systems with no advanced functionality. This product aims to include more advanced functionally compared to other hospital management systems.

1.3. Title

Enigma Hospital Management System

1.4.Objectives

- Complete digitization of patient and doctor data
- Efficient patient management
- Automated prescription for patients and digital prescription record storage
- Automated billing
- Improve Transparency
- Efficient time management
- Improved inter-department communications in hospital

1.5.Problem Statement

Most Hospitals and clinics in Pakistan (particularly government hospitals) rely on traditional, manual file-based storage of data related to both patient and doctor. Even where some form of Hospital Management Systems exists, it is often limited to basic features and outdated. This leads to several problems within hospitals such as time delays, loss of records, outdated software usage and ultimately not a good experience for the patients and in some cases the doctors as well.

Apart from the above-mentioned issues, several hospitals in Pakistan also suffer from several other issues as well. Hospitals often rely on staff to send messages across departments, which is slow and ineffective, the delays cause problems for both patients and doctors. Reliance on such staff leads to issues in transparency, since the communication is done manually it is entirely reliant on the hospital

staff alone. Storage of previous patient records, such as what doctors they have met with before as well as their prescriptions are often not stored in the database, particularly the prescriptions. In most cases the doctor manually writes the prescription, but this is not usually recorded in most hospital management systems. One recurring issue with doctors is that their patient has consulted some other doctor in the hospital prior to their meeting, and the patient has lost or did not bring their prescription from the previous doctor. Our management system aims to avoid this problem. The doctors can access the entire patient history and prescription, as can the patient in case it is needed for another hospital.

Our system will rely on a database what can be used across multiple hospital branches, this ensures that patients can go to any hospital branch and their data can be updated and accessed by them and the doctors of that branch. The management system will maintain records of patients such as the doctors they visited and their medical prescriptions, it will also have a record for doctors. The system can be used to update patient status based on what department they need to go to and when, as well as their future appointments with doctors. The system can automatically store patient bill records as well. Patient data can be accessed by the patient.

2. Overall Description

2.1.Product Perspective

Hospital Management Systems are commonly used in hospital around the world, Enigma Hospital Management System (EHMS) is an improvement over existing hospital management system. Enigma hospital management system aims to improve on in use hospital systems in Pakistan, most hospital management systems in use throughout Pakistan have limited features compared to those this product aims to offer.

2.2.Product Functions

- Automated Patient Billing
- Hospital Ward Management
- Patient Prescription Management
- Patient Appointment Automation
- Digitized and editable Doctor schedules
- Operation Theatre management
- Hospital Staff Management

2.3.List of Use Cases

- Prescribe Medicine
- Manage Doctors
- Discharge Ward Patient
- Register Patient
- Update Doctor Schedule
- Manage Ward Patients
- Assign Doctor

- Register Staff/Doctor

2.4.Extended Use Cases

<table>
<tr><td>Use Case Name</td><td colspan="2">Prescribe Medicine</td></tr>
<tr><td>Scope</td><td colspan="2">Enigma Hospital Management System (EHMS)</td></tr>
<tr><td>Level</td><td colspan="2">User Goal</td></tr>
<tr><td>Primary Actor</td><td colspan="2">Doctor</td></tr>
<tr><td>Stake Holders & Interests</td><td colspan="2">Hospital, Patient</td></tr>
<tr><td>Main Success Scenario</td><td colspan="2"><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Doctor enters the patient ID into the system.</td><td></td></tr>
<tr><td></td><td>2. System validates the patient credentials.
3. System allows doctor to add a prescription to the patient.</td></tr>
<tr><td>4. Doctor prescribes the medicines to the patient on system.</td><td></td></tr>
<tr><td></td><td>5. System updates the patient data with the newly prescribed medicines.
6. System marks the scheduled doctor meeting with patient as complete</td></tr>
</table></td></tr>
<tr><td>Extensions</td><td colspan="2">1. Invalid Patient ID was entered
1.1. System Prompts the doctor to recheck patient ID and enter again
1.2. Doctor enters the patient ID again
1.3. System validates entered patient ID
1.4. Entered patient ID was valid
1.4.1. Patient ID is now valid, System continues from step (2) of main success scenario
1.5. Entered patient ID was invalid
1.5.1. System tells doctor to inform staff for assistance
1.5.2.Doctor Informs staff
2. System runs into an error
2.1. System outputs error message along with error code
2.2. System informs doctor to contact IT staff or admin for support</td></tr>
<tr><td>Preconditions</td><td colspan="2">1. Patient has been registered
2. Doctor has been registered
3. Doctor is signed into the system</td></tr>
<tr><td>Postconditions</td><td colspan="2">1. Patient goes to pharmacy for medicine
2. Database moves from one consistent state to another</td></tr>
</table>

<table>
<tr><td>Use Case Name</td><td>Manage Doctors</td></tr>
<tr><td>Scope</td><td>Enigma Hospital Management System (EHMS)</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Staff</td></tr>
<tr><td>Stake Holders & Interests</td><td>Doctor, Hospital</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Staff enters the doctor ID into the system</td><td></td></tr>
<tr><td></td><td>2. System validates the doctor ID
3. System returns a list of possible operations:
A. Update doctor timings
B. Update doctor data
C. Remove doctor</td></tr>
<tr><td>4. Staff selects one of the options to manage doctors</td><td></td></tr>
<tr><td></td><td>5. System updates data and informs the user of completion</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1. System runs into an error
1.1. System outputs error message along with error code
1.2. System informs doctor to contact IT staff or admin for support
2. System is unable to verify Doctor ID
2.1. System informs staff to re-enter Doctor ID
2.2. Staff enters Doctor ID again
2.3. System validates Doctor ID
2.3.1.Invalid Doctor ID was entered
2.3.1.1. System informs staff of authentication failure and suggests re-entering ID or contacting IT support or admin
2.3.2.Valid Doctor ID was entered
2.3.2.1. System continues from step (3) of the main success scenario</td></tr>
<tr><td>Preconditions</td><td>1. Doctor has been registered
2. Staff has been registered</td></tr>
<tr><td>Postconditions</td><td>1. Updated doctor data shown when staff checks doctor data
2. New Patients can now only meet doctor according to new timings</td></tr>
</table>

<table>
<tr><td>Use Case Name</td><td>Discharge from Ward</td></tr>
<tr><td>Scope</td><td>Enigma Hospital Management System (EHMS)</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Ward Staff</td></tr>
<tr><td>Stake Holders & Interests</td><td>Staff, Doctor, Patient</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Ward staff inputs the patient ID into the system</td><td></td></tr>
<tr><td></td><td>2. System validates the patient ID
3. System asks for confirmation to discharge the patient after displaying patient details along with ward number and billing</td></tr>
<tr><td>4. Ward staff collects payment/bill from patient or guardian
5. Ward staff confirms patient discharge in the system</td><td></td></tr>
<tr><td></td><td>6. System checks for confirmation
7. System updates the patient data
8. System marks the ward as free and informs ward staff</td></tr>
<tr><td>9. Patient leaves the ward</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1. Invalid Patient ID was entered/Patient ID does not exist
1.1. System Prompts the ward staff to recheck patient ID and enter again
1.2. Ward staff enters the patient ID again
1.3. System validates entered patient ID
1.4. Entered patient ID was valid
1.5. Patient ID is now valid, System continues from step (3) of main success scenario
1.5.1.Entered patient ID was invalid
1.5.2. System informs ward staff to enter ID again or to call IT staff for assistance
1.5.3.Doctor Informs IT staff or repeat of steps 1.1-1.5
2. Patient condition is not feasible for discharge
2.2. Ward staff informs patient/guardian regarding medical conditions
2.3. Patient is not discharged from the ward
3. Payment has not been made or payment error (in case of payment by card)
3.1. Ward staff informs guardian or patient regarding payment error
3.2. Ward staff asks for payment again
3.3. Payment is successfully made by patient/guardian
3.3.1.Payment Problems persist/ Guardian or Patient decides to pay later
3.3.1.1. Patient is not discharged
3.4. Patient is discharged</td></tr>
<tr><td>Preconditions</td><td>1. Patient has been registered
2. Patient had been assigned a ward</td></tr>
</table>

	3. Ward staff is logged into the system
Postconditions	1. Ward staff can now assign ward to other patients 2. Ward patient must leave the ward

Use Case Name	Register Patient
Scope	Enigma Hospital Management System (EHMS)
Level	User Goal
Primary Actor	Receptionist
Stake Holders & Interests	Hospital staff, Patient, Doctor
Main Success Scenario	

Actor Action	System Response
1. Patient provides data to the receptionist 2. Receptionist inputs the data into the system	
	3. System saves patient data into the database after ensuring all relevant data has been provided 4. System creates a unique ID for the patient
5. Receptionist provides patient with the generated ID (printout receipt)	
6. Receptionist asks patient regarding which services they require	
7. Patient informs receptionist regarding their required services(department) 8. Receptionist redirects patient towards the specific department reception	

Extensions	1. Patient is already registered 1.1. System informs receptionist that the patient has already been registered 1.2. System provides the patient ID to the receptionist 1.3. Continuation of main success scenario from step (5) 2. System encounters error/Failure 2.1. System outputs error message along with error code 2.2. System informs receptionist to contact IT staff or admin for support 3. No department with specified name is found 3.1. Receptionist informs patient that no such service/department is available in hospital 3.2. Receptionist suggests alternate departments if any are viable
Preconditions	1. Patient is not registered 2. Receptionist is registered 3. Logged in as receptionist in the system
Postconditions	1. Patient selects doctor based on timings specified by receptionist 2. Doctor is assigned to the patient if patient selects the doctor

<table>
<tr><td>Use Case Name</td><td>Update Schedule</td></tr>
<tr><td>Scope</td><td>Enigma Hospital Management System (EHMS)</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Doctor</td></tr>
<tr><td>Stake Holders & Interests</td><td>Patient, Staff, Doctor</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Doctor selects option to edit schedule</td><td></td></tr>
<tr><td></td><td>2. System displays current doctor weekly timings
3. System displays doctor patient appointments/meetings</td></tr>
<tr><td>4. Doctor selects any meeting/appointment from those listed by the system</td><td></td></tr>
<tr><td></td><td>5. System displays option for cancellation or timing update</td></tr>
<tr><td>6. Doctor selects options and inputs new time in case of timing update being selected</td><td></td></tr>
<tr><td></td><td>7. System updates the relevant meetings/appointments as specified by doctor; doctor is informed regarding changes that have taken place</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1. System encounters error/Failure
1.1. System outputs error message along with error code
1.2. System informs receptionist to contact IT staff or admin for support
2. Invalid new appointment/meeting time entered
2.1. System informs doctor regarding invalid timing input
2.2. System asks for input again or asks for cancelation
2.3. Doctor enters time & date again
2.4. Doctor enters the date and time
2.4.1. Doctor cancels updating operation
2.4.2. No changes made by the system, Doctor is informed regarding no change
2.5. System validates timing and date
2.6. If invalid time is entered, steps 2.1-2.5 are repeated</td></tr>
<tr><td>Preconditions</td><td>1. Logged in as doctor in the system</td></tr>
<tr><td>Postconditions</td><td>2. Patient appointments or meetings are updated according to changes made by doctor</td></tr>
</table>

<table>
<tr><td>Use Case Name</td><td>Manage Ward Patients</td></tr>
<tr><td>Scope</td><td>Enigma Hospital Management System (EHMS)</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Ward Manager</td></tr>
<tr><td>Stake Holders & Interests</td><td>Hospital, Patient</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Ward Manager opens the portal.</td><td></td></tr>
<tr><td></td><td>2. System shows Ward Manager, information about the patients.</td></tr>
<tr><td>3. Ward Manager views information about patients in the ward.
4. Ward Manager updates, adds, or deletes patients’ information in the ward.</td><td></td></tr>
<tr><td></td><td>5. System validates patients’ credentials.
6. System updates the ward database according to the action taken by Ward Manager.</td></tr>
<tr><td>7. Ward Manager logs out of the portal.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>3. Invalid Patient ID was entered
3.1. System Prompts the Ward Manager to recheck patient ID and enter again
3.2. Ward Manager enters the patient ID again
3.3. System validates entered patient ID
3.4. Entered patient ID was valid
3.4.1. Patient ID is now valid, System continues from step (6) of main success scenario
3.5. Entered patient ID was invalid
3.5.1. System tells Ward Manager to inform staff for assistance
3.5.2. Ward Manager Informs staff
4. System runs into an error
4.1. System outputs error message along with error code
4.2. System informs Ward Manager to contact IT staff or admin for support
5. Ward is full and Ward Manager wants to add patient
5.1. System informs Ward Manager that the ward is full.
5.2. Ward Manager informs patient to get admitted in a private ward outside the hospital till a room/bed becomes available.</td></tr>
<tr><td>Preconditions</td><td>4. Patient has been registered
5. Ward Manager has been registered
6. Ward Manager has been logged-in to the portal</td></tr>
</table>

Postconditions	1. Action performed by Ward Manager is registered and implemented. 2. Ward Database is updated and moved from one consistent state to another.

Use Case Name	Assign Doctor
Scope	Enigma Hospital Management System (EHMS)
Level	User Goal
Primary Actor	Department Receptionist
Stake Holders & Interests	Doctor, Patient

Main Success Scenario

Actor Action	System Response
1. Patient approaches Department Receptionist and provides the receipt 2. Department Receptionist inputs patient ID into the system.	
	3. System validates the patient ID 4. System returns patient's details
5. Receptionist asks patient of their preference for their previous doctor, any specific doctor. 6. Patient replies with their preference. 7. Receptionist checks Doctor's availability.	
	8. System checks that the Doctor is currently available
9. Receptionist asks for payment. 10. Patient pays for the consultation fee. 11. Receptionist assigns the patient to the Doctor.	
	12. System forwards patient's details and record to the Doctor's computer 13. System adds the patient to the waiting list of the Doctor.
14. Patient leaves the reception	

Extensions

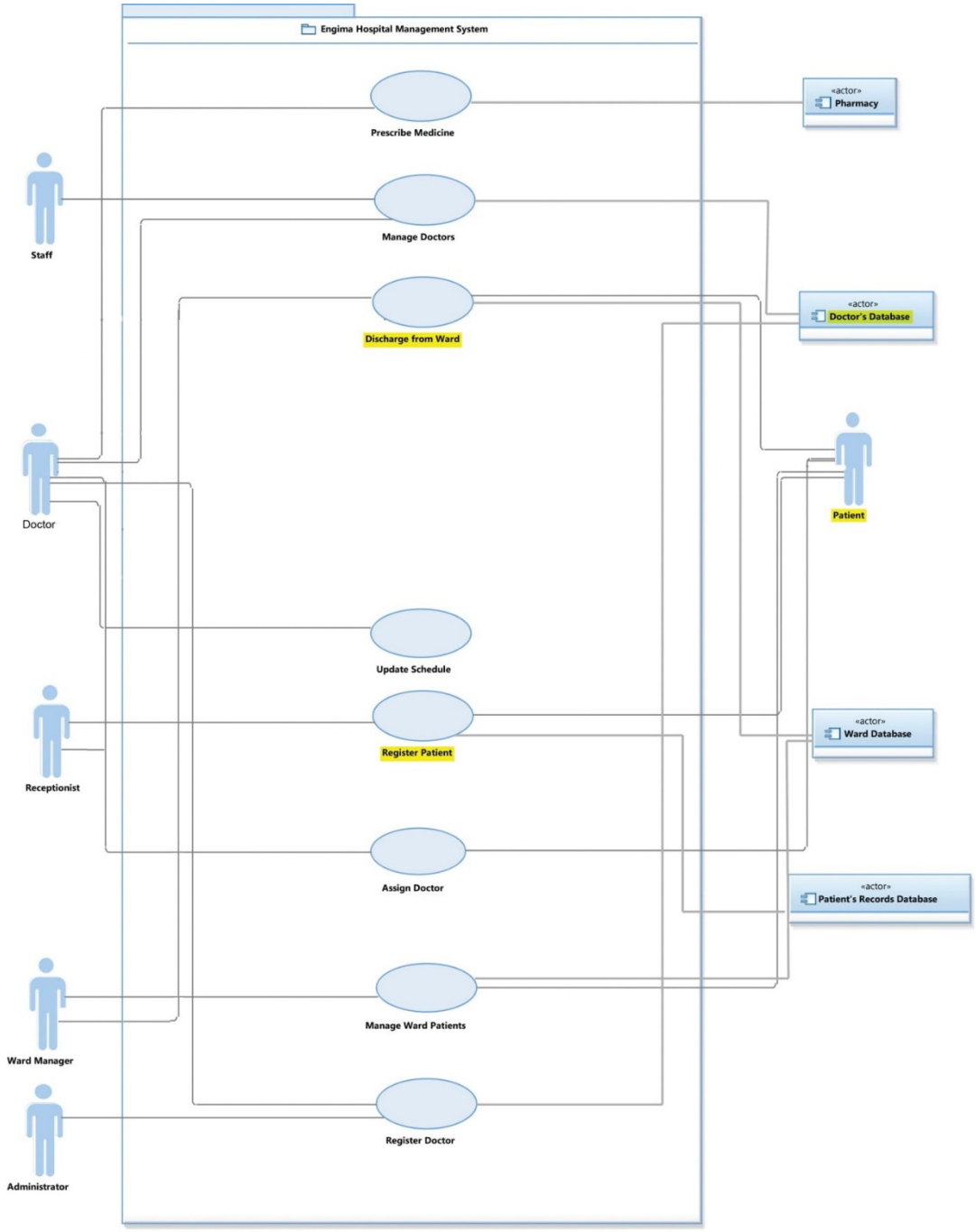
Patient doesn't have a receipt
1.1. Patient is asked to go to the Front Desk to get a receipt.
Patient has no preference of the Doctors
2.1. Patient is assigned any random Doctor that is available in the OPD
Invalid Patient ID was entered/Patient ID does not exist
3.1. System Prompts the Receptionist to recheck patient ID and enter again
3.2. Receptionist enters the patient ID again
3.3. System validates entered patient ID
3.4. Entered patient ID was valid

	3.4.1. Patient ID is now valid, System continues from step (4) of main success scenario 3.5. Entered patient ID was invalid 3.5.1. System informs Receptionist to enter ID again or to call IT staff for assistance 3.5.2.Receptionist Informs IT staff 4. Payment has not been made or payment error (in case of payment by card) 6.1. Receptionist informs guardian or patient regarding payment error 6.2. Receptionist staff asks for payment again 6.2.1. Payment is successfully made by patient/guardian 6.2.1.1. Patient is assigned the Doctor 6.2.2. Payment Problems persist/ Guardian or Patient decides to pay later 6.2.2.1. Patient is refused Doctor Consultation
Preconditions	1. Patient has been registered 2. Receptionist is logged into the system
Postconditions	1. Doctor is assigned to the patient and patient's details and history is forwarded to the Doctor. 2. Payment is registered in the database. 3. Patients visit details are updated in the patient's record.

Use Case Name	Register Staff/Doctor
Scope	Enigma Hospital Management System (EHMS)
Level	User Goal
Primary Actor	Administrator
Stake Holders & Interests	Hospital Staff, Doctor
Main Success Scenario	**Actor Action** / **System Response** 1. Administrator receives data of the Doctor/Staff which is to be added. 2. Administrator inputs the data into the system 1. System saves data into the database after ensuring all relevant data has been provided. 2. System creates a unique ID for the Staff/Doctor 3. System forwards Staff/Doctors data to the relevant departments. 4. Administrator informs the Staff/Doctor that they have been registered
Extensions	1. Doctor is already registered 1.1. System informs receptionist that the Staff/Doctor has already been registered 1.2. System provides the Staff/Doctor ID to the Admin 1.3. Admin informs relevant department that the Staff/Doctor has already been registered
Preconditions	1. Administrator has been logged-in to the portal 2. Staff/Doctor has been hired and must be registered 3. Information of Staff/Doctor has been verified

Postconditions	1. Actions performed by Administrator must be registered and implemented. 2. Database of the Hospital is updated and moved from one consistent state to the other

2.5. Use Case Diagram



3. Other Nonfunctional Requirements

3.1. Performance Requirements

- No specific performance requirements for the product

3.2. Safety Requirements

- Hospital needs to ensure patient health, over reliance on EHMS may lead to possible harm to patients' health.

3.3. Security Requirements

- Ensuring the safety of the database, particularly with regards to its physical location needs to ensure.
- The hospital must ensure a reliable person oversees the system (System Administrator), to prevent data from database from being shared.

- User authentication details need to be restricted to only those permitted access.

3.4. Software Quality Attributes

- **High Usability:** The software is easy to use, its menus and GUI are highly intuitive. No advanced training is required to use the software.

- **Reliable Software:** The software can be relied upon to work in most cases, it is highly tested prior and is less prone to run into error.

- **Correctness:** Data is logically checked before being updated in software, it is checked both syntactically and semantically. This ensures the software always provides and stores correct data.

3.5. Business Rules

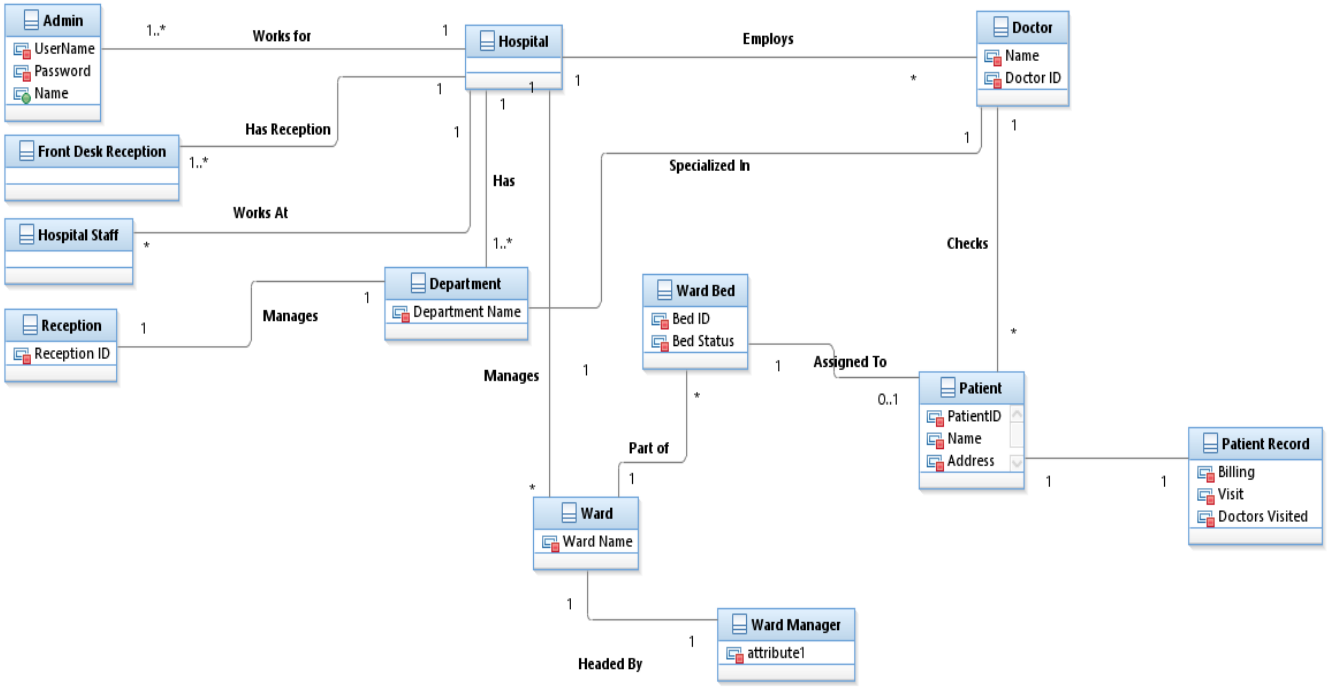
The system is to be used by Doctors, Hospital Staff, System Administrators and Personal Assistants. Doctors can only edit their own data and update some parts of the hospital patient data. Hospital staff have access to and are allowed to modify almost all the data excluding their own. Personal assistants should be limited only to patient and operation data of their respective doctors.

3.6. Operating Environment

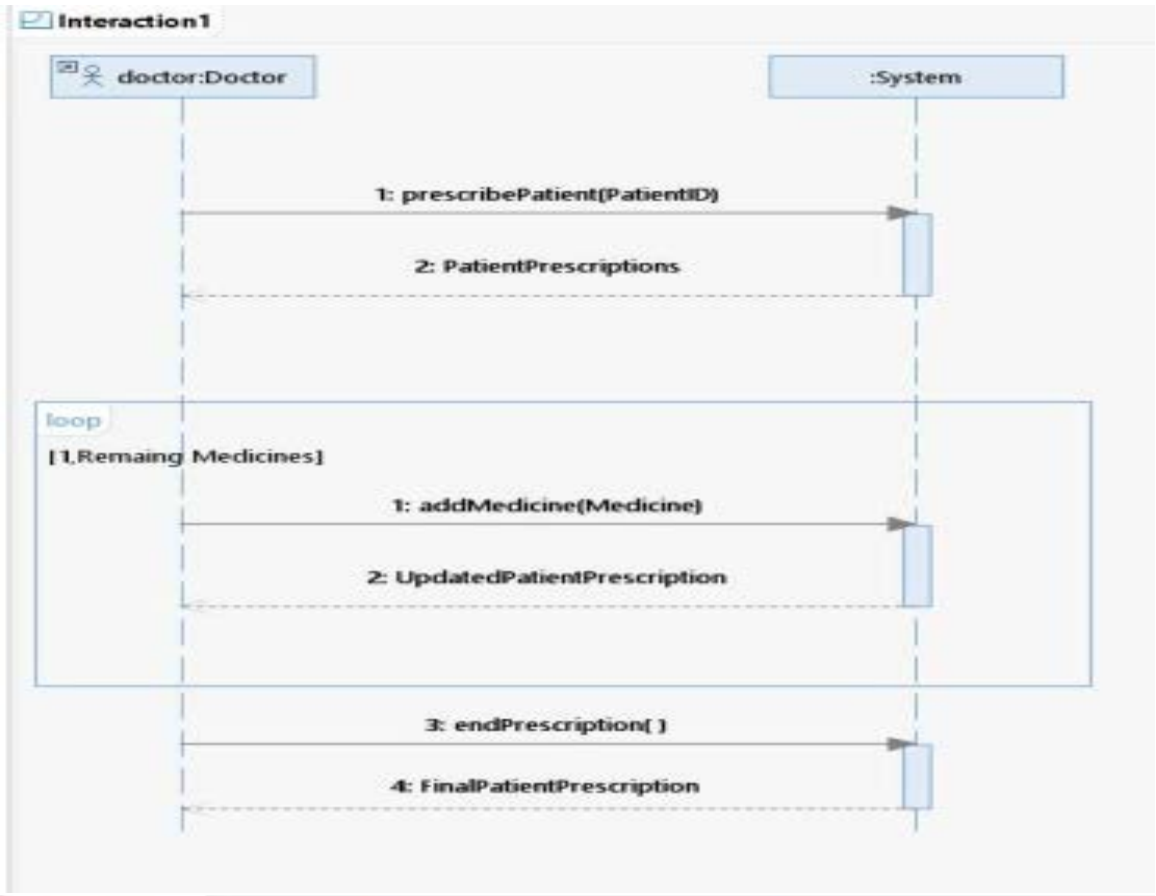
The product requires the following specifications:

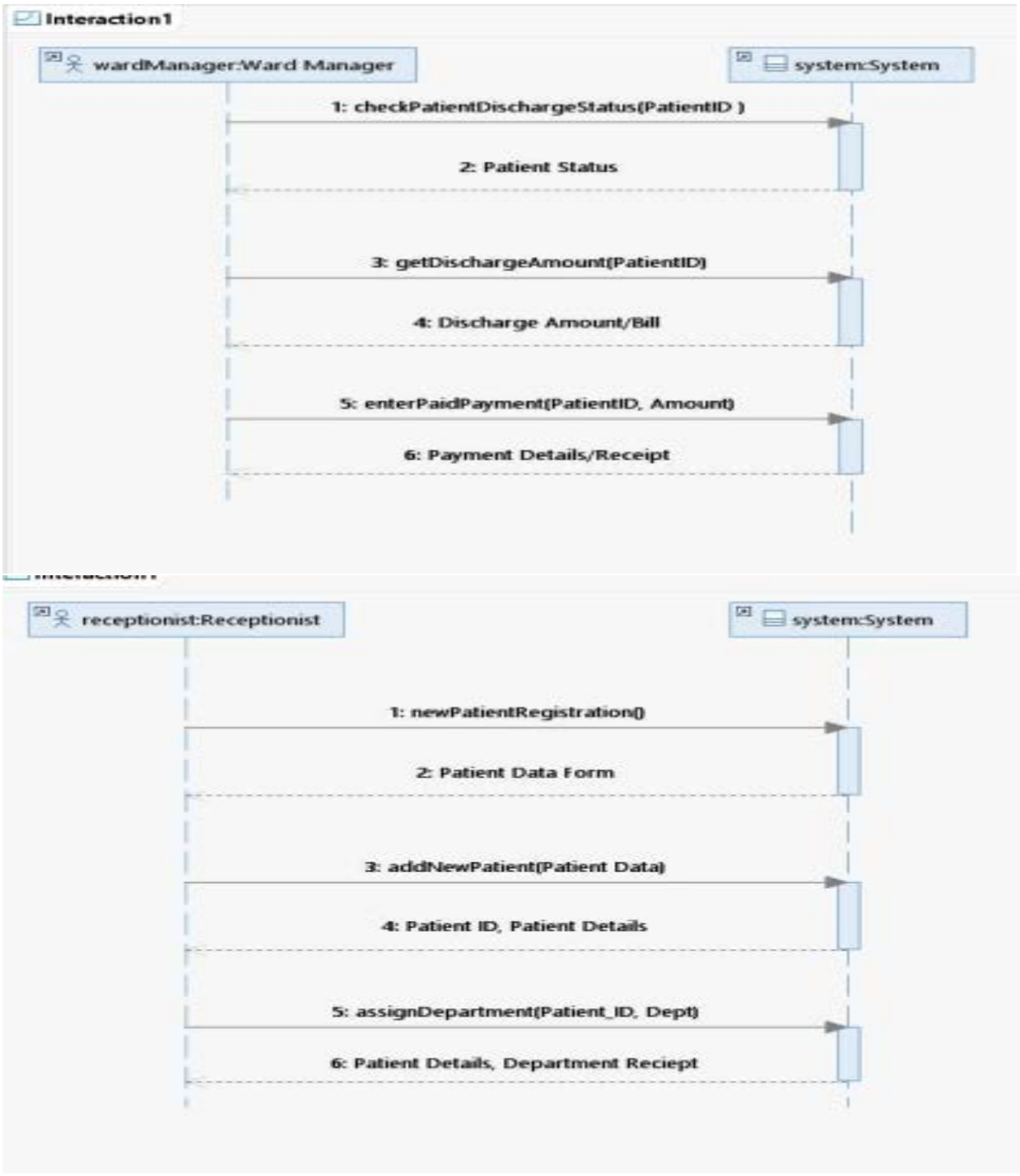
- PC with Monitor
- Windows Installation in PC
- Java RTE
- Keyboard and Mouse

4. Domain Model

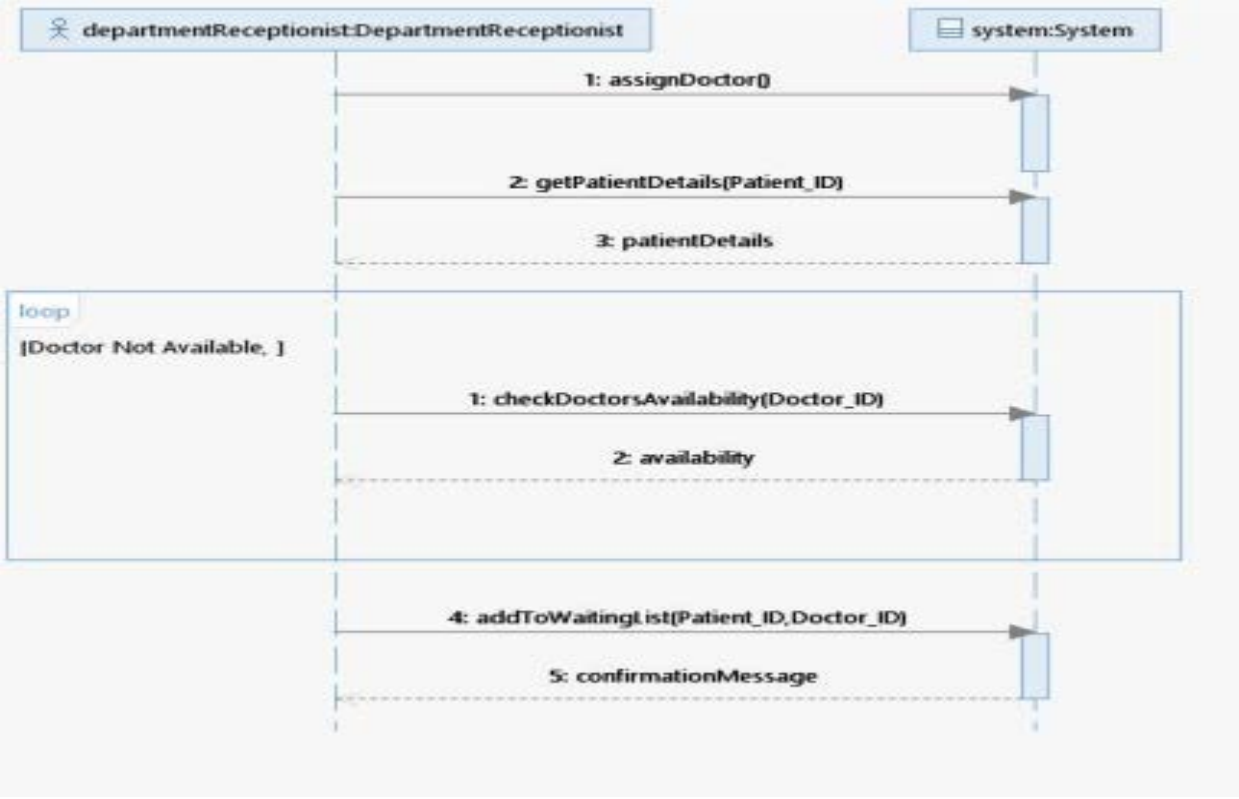


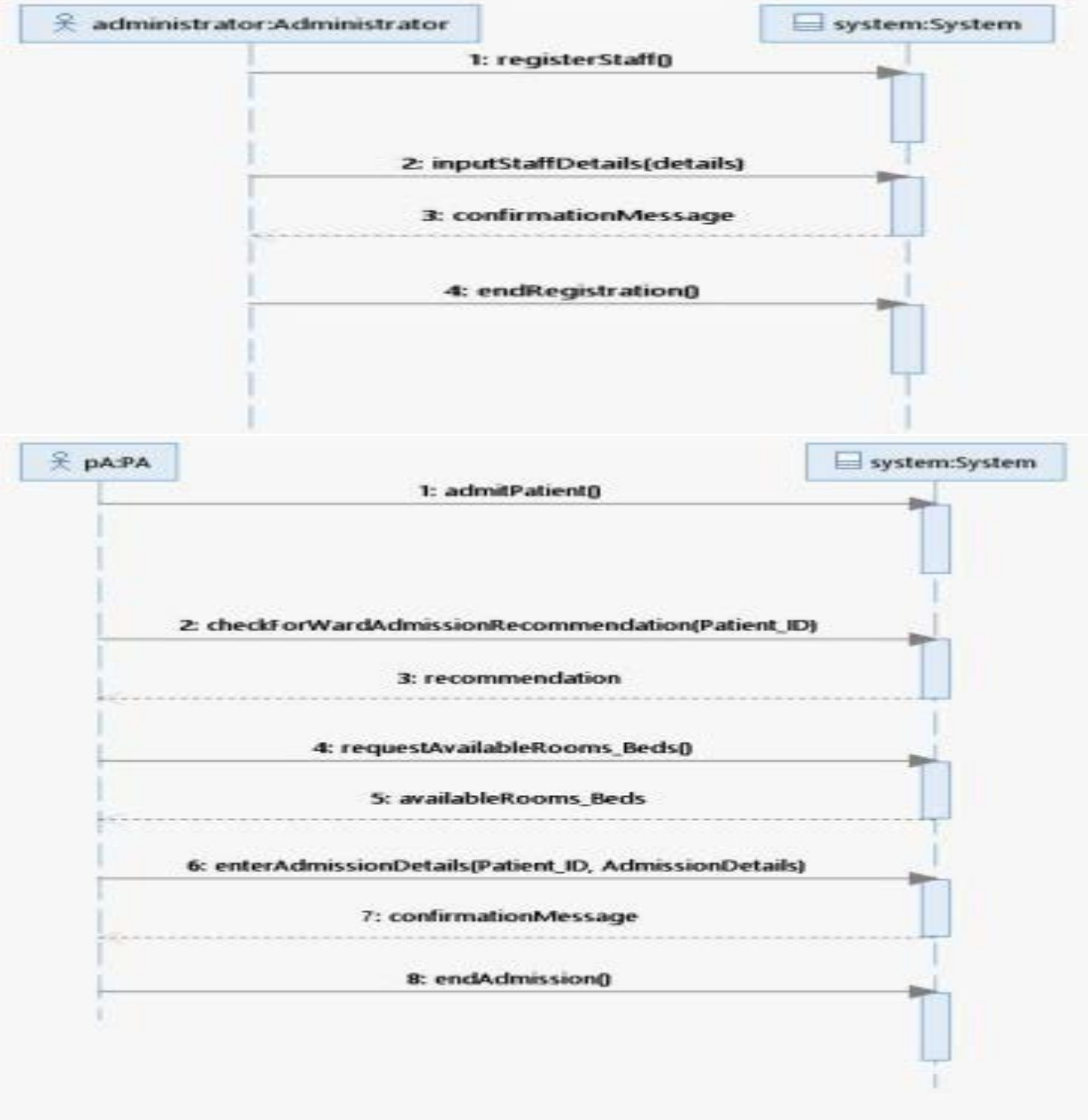
5. System Sequence Diagram



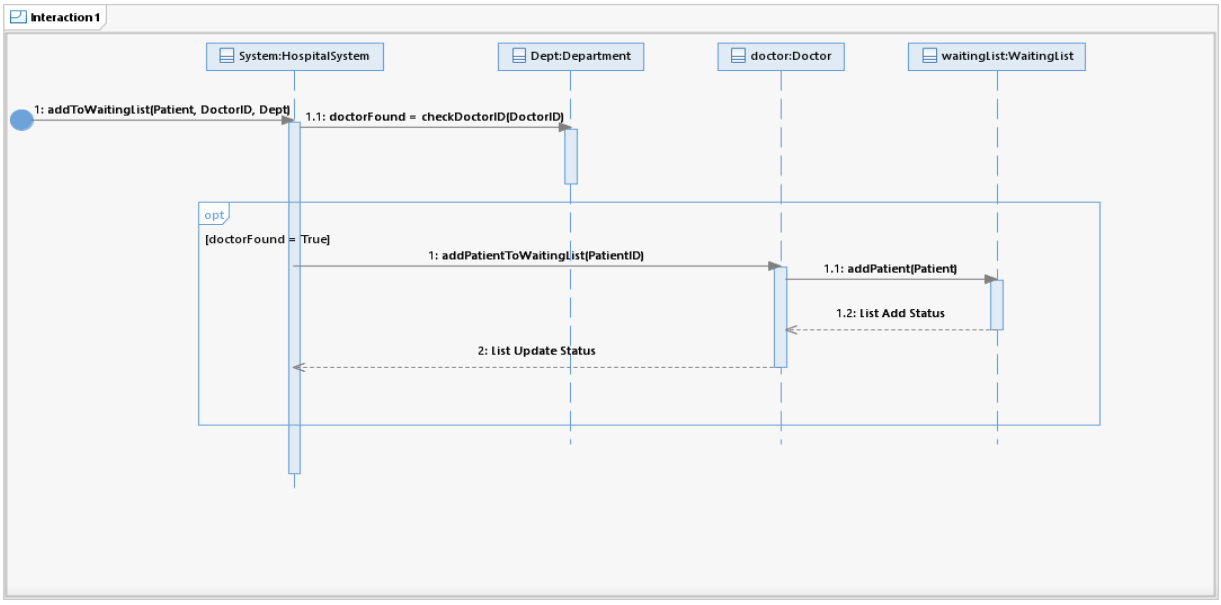

Interaction1
wardManager:Ward Manager
system:System
1: checkPatientDischargeStatus(PatientID)
2: Patient Status
3: getDischargeAmount(PatientID)
4: Discharge Amount/Bill
5: enterPaidPayment(PatientID, Amount)
6: Payment Details/Receipt
receptionist:Receptionist
system:System
1: newPatientRegistration()
2: Patient Data Form
3: addNewPatient(Patient Data)
4: Patient ID, Patient Details
5: assignDepartment(Patient_ID, Dept)
6: Patient Details, Department Reciept

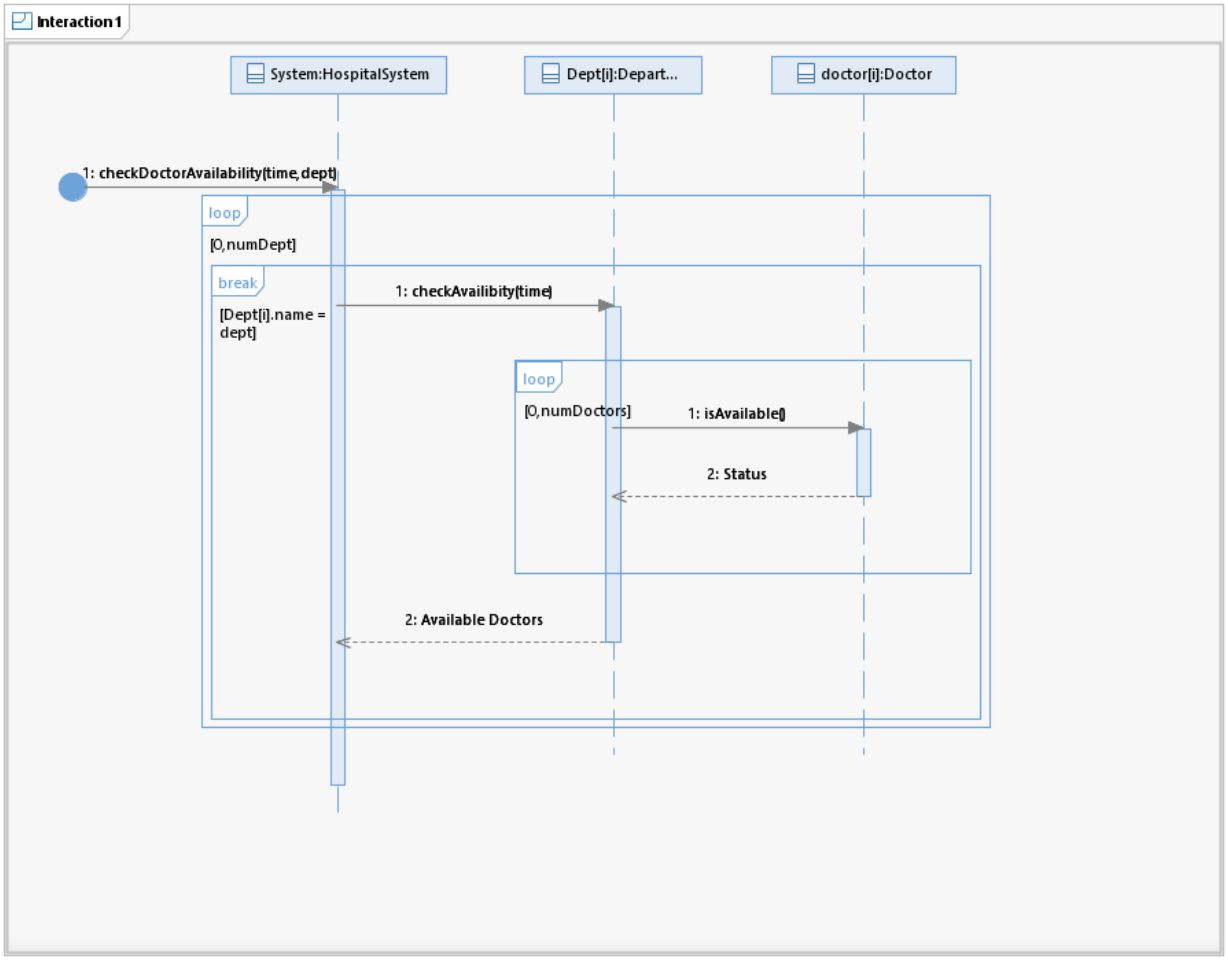

doctor:Doctor
system:System
1: editSchedule( )
2: Doctor Schedule
3: selectAppointment(Appointment)
4: Appointment Details, Update Options
5: applyUpdate(Operation, Appointment)
6: Doctor Schedule

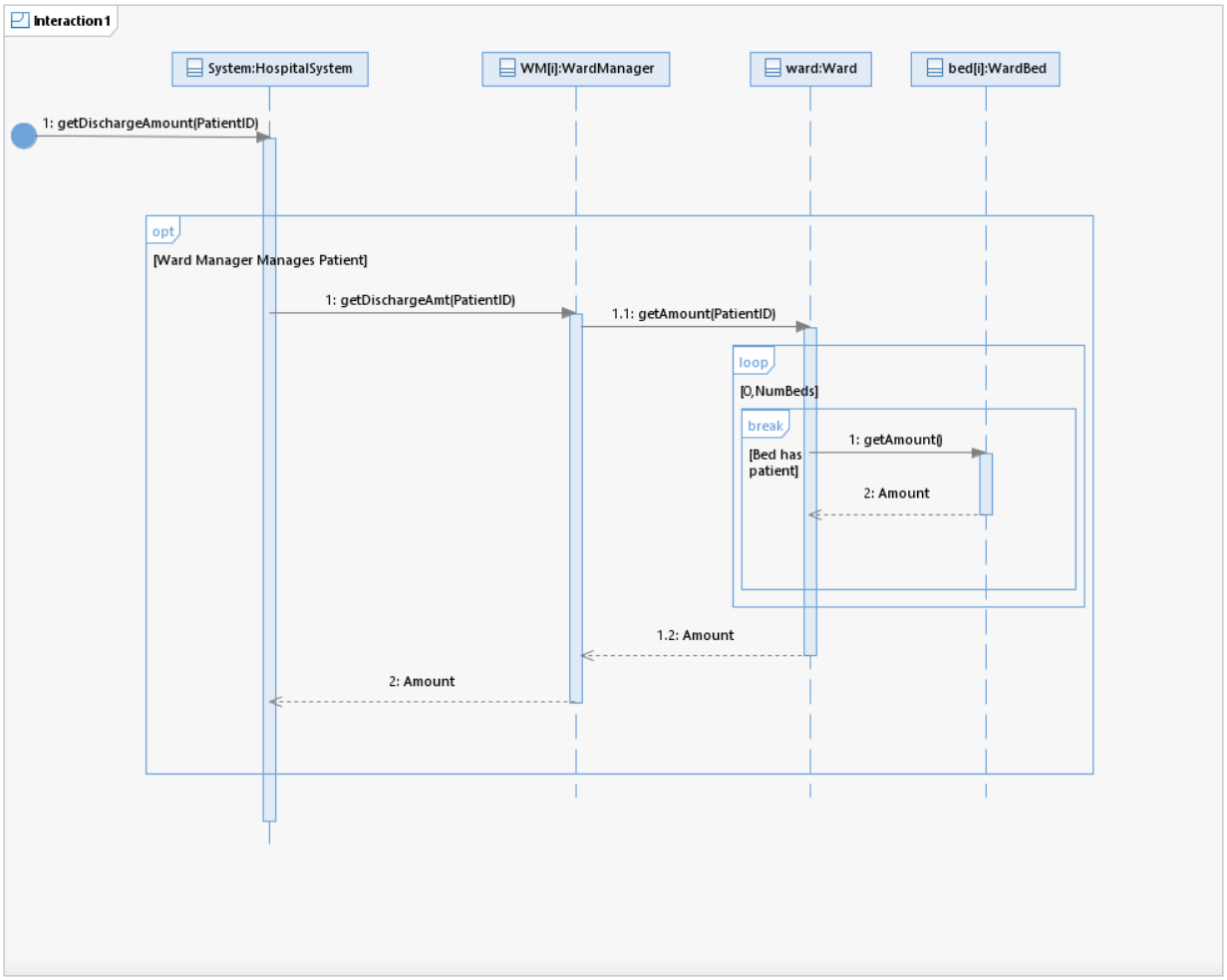

departmentReceptionist:DepartmentReceptionist
system:System
1: assignDoctor()
2: getPatientDetails(Patient_ID)
3: patientDetails
loop
[Doctor Not Available,]
1: checkDoctorsAvailability(Doctor_ID)
2: availability
4: addToWaitingList(Patient_ID,Doctor_ID)
5: confirmationMessage

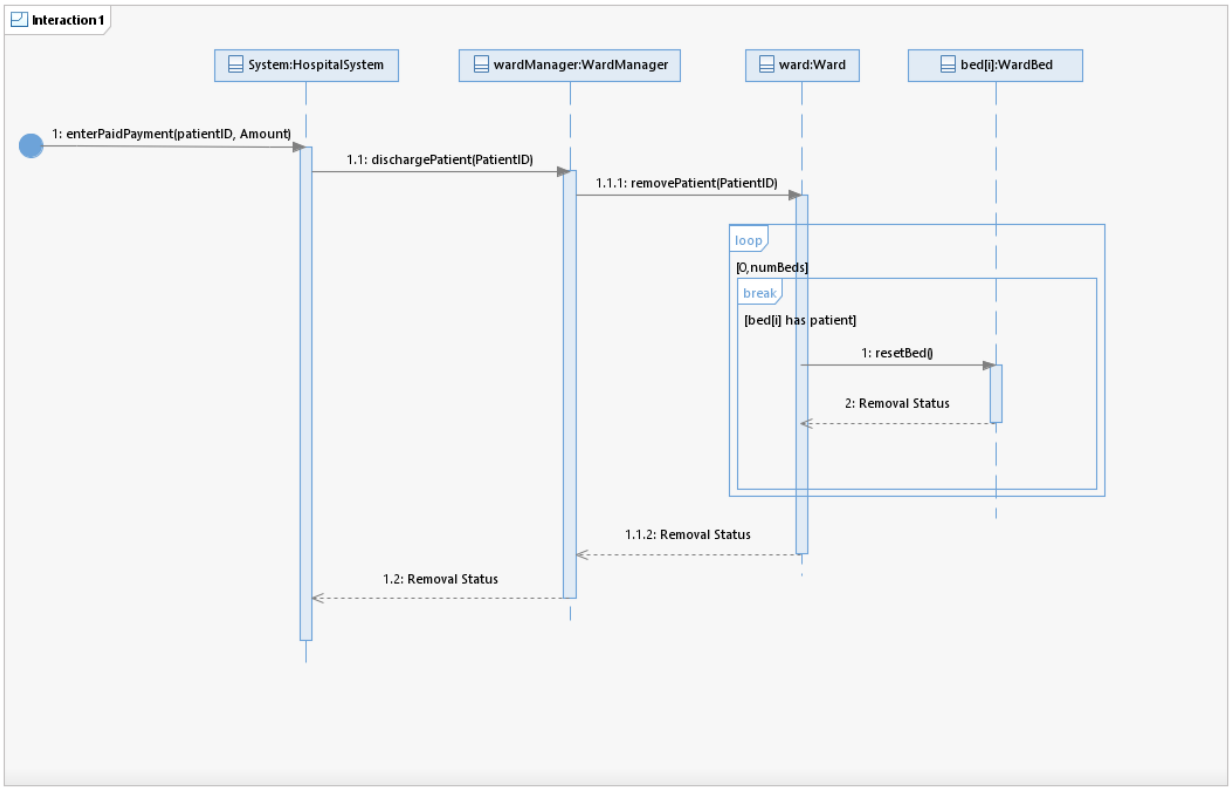

administrator:Administrator
system:System
1: registerStaff()
2: inputStaffDetails(details)
3: confirmationMessage
4: endRegistration()
pA:PA
system:System
1: admitPatient()
2: checkForWardAdmissionRecommendation(Patient_ID)
3: recommendation
4: requestAvailableRooms_Beds()
5: availableRooms_Beds
6: enterAdmissionDetails(Patient_ID, AdmissionDetails)
7: confirmationMessage
8: endAdmission()

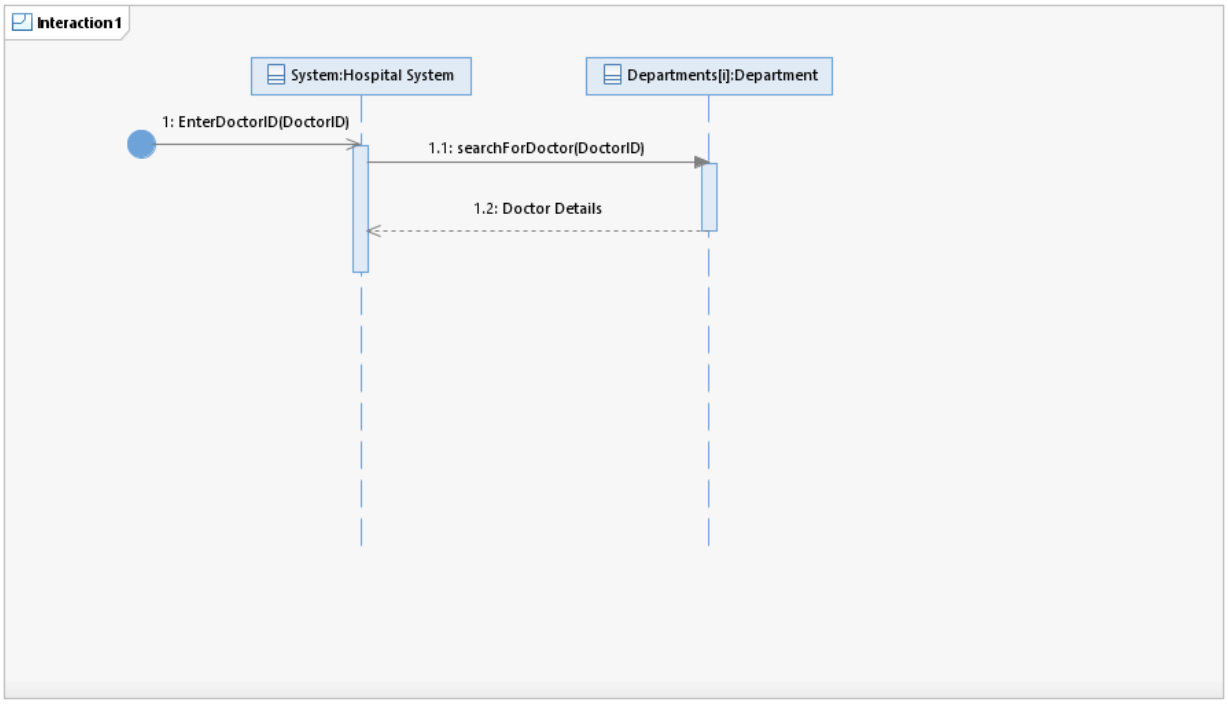
6. Sequence Diagram

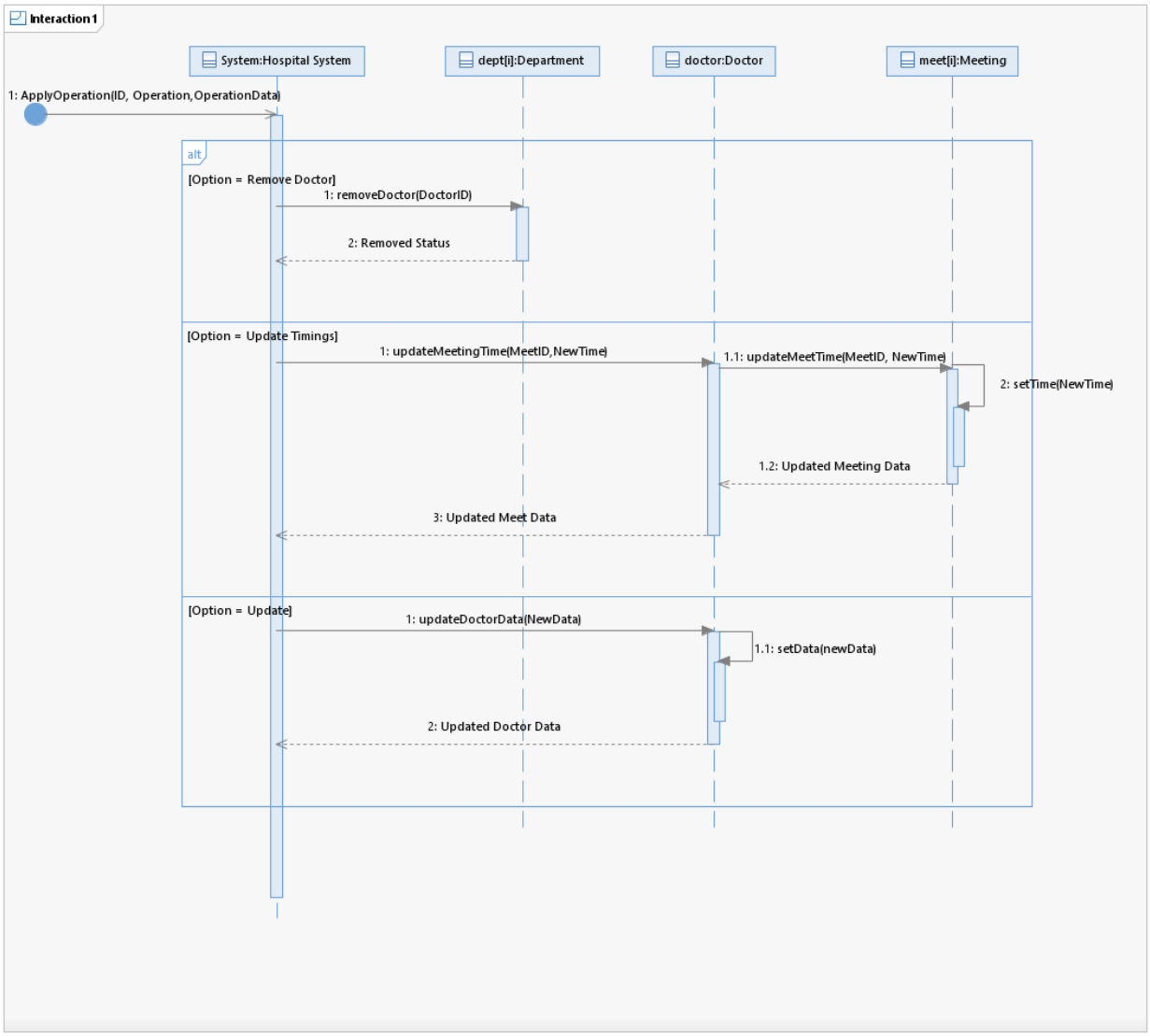


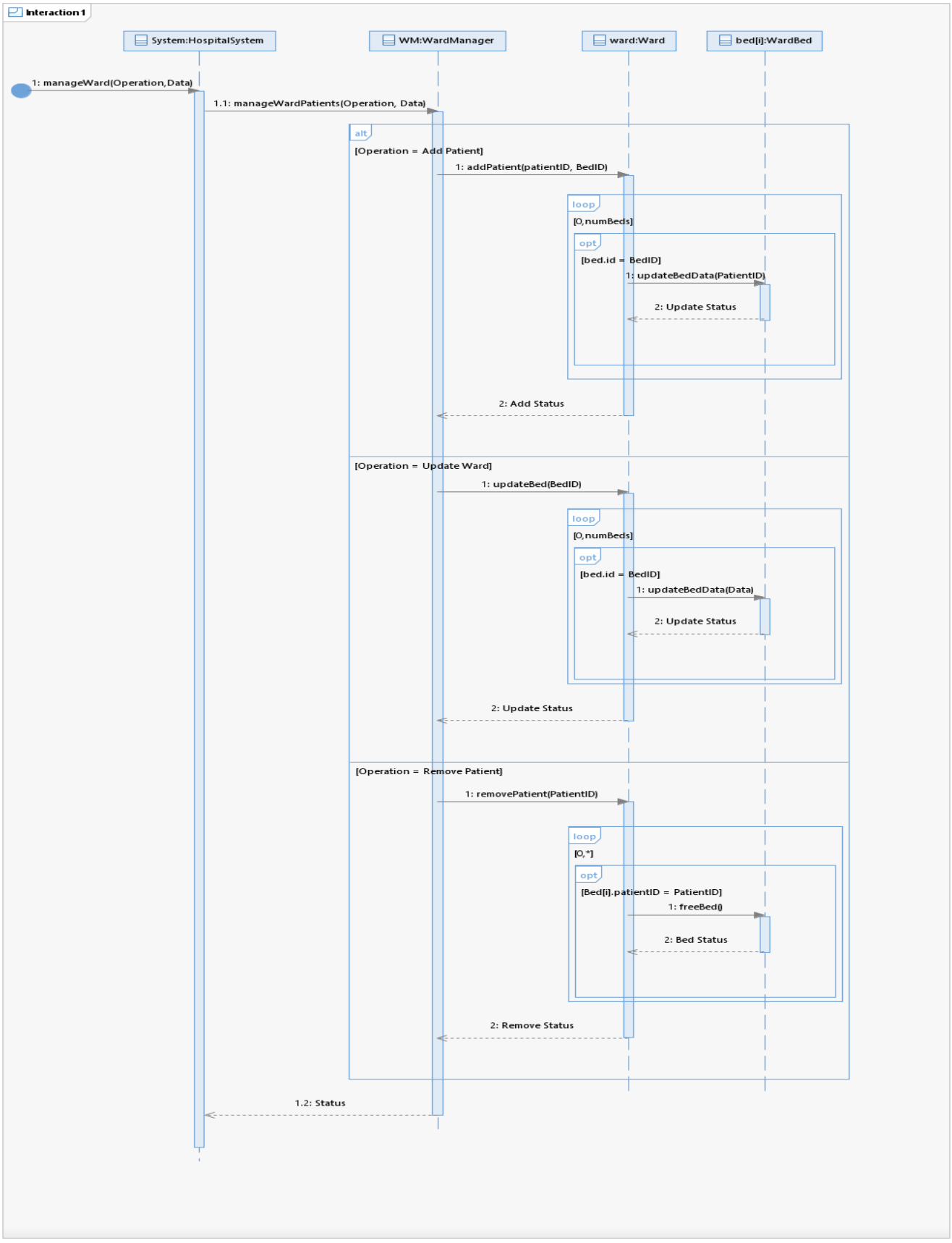


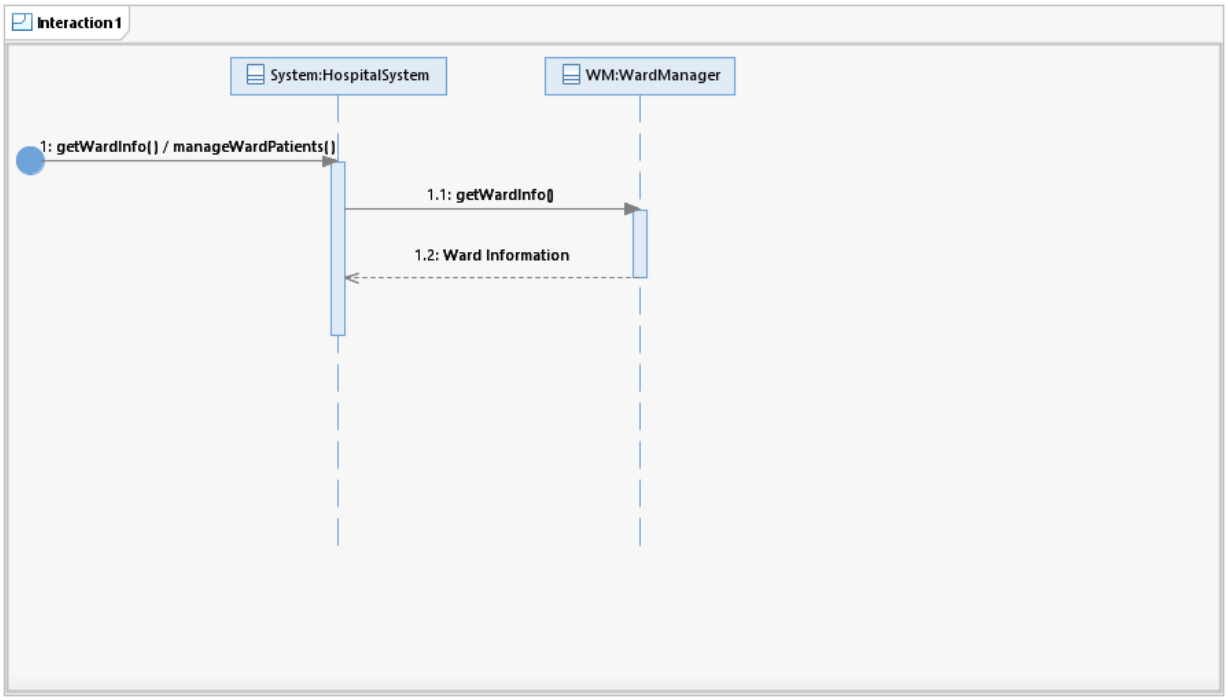

Interaction 1
System:HospitalSystem
WM[i]:WardManager
ward:Ward
bed[i]:WardBed
1: getDischargeAmount(PatientID)
opt
[Ward Manager Manages Patient]
1: getDischargeAmt(PatientID)
1.1: getAmount(PatientID)
loop
[0,NumBeds]
break
[Bed has patient]
1: getAmount()
2: Amount
1.2: Amount
2: Amount

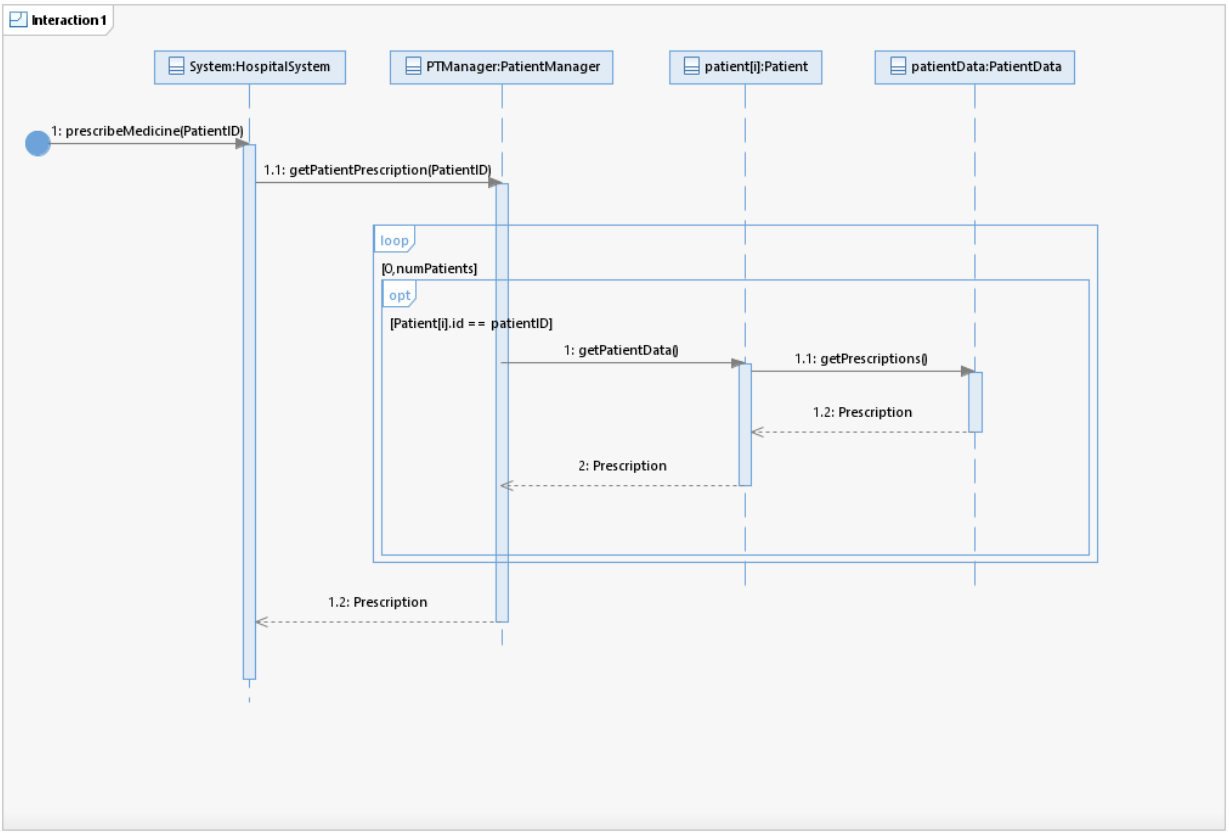

Interaction1
System:HospitalSystem
wardManager:WardManager
ward:Ward
bed[i]:WardBed
1: enterPaidPayment(patientID, Amount)
1.1: dischargePatient(PatientID)
1.1.1: removePatient(PatientID)
loop
[0,numBeds]
break
[bed[i] has patient]
1: resetBed()
2: Removal Status
1.1.2: Removal Status
1.2: Removal Status

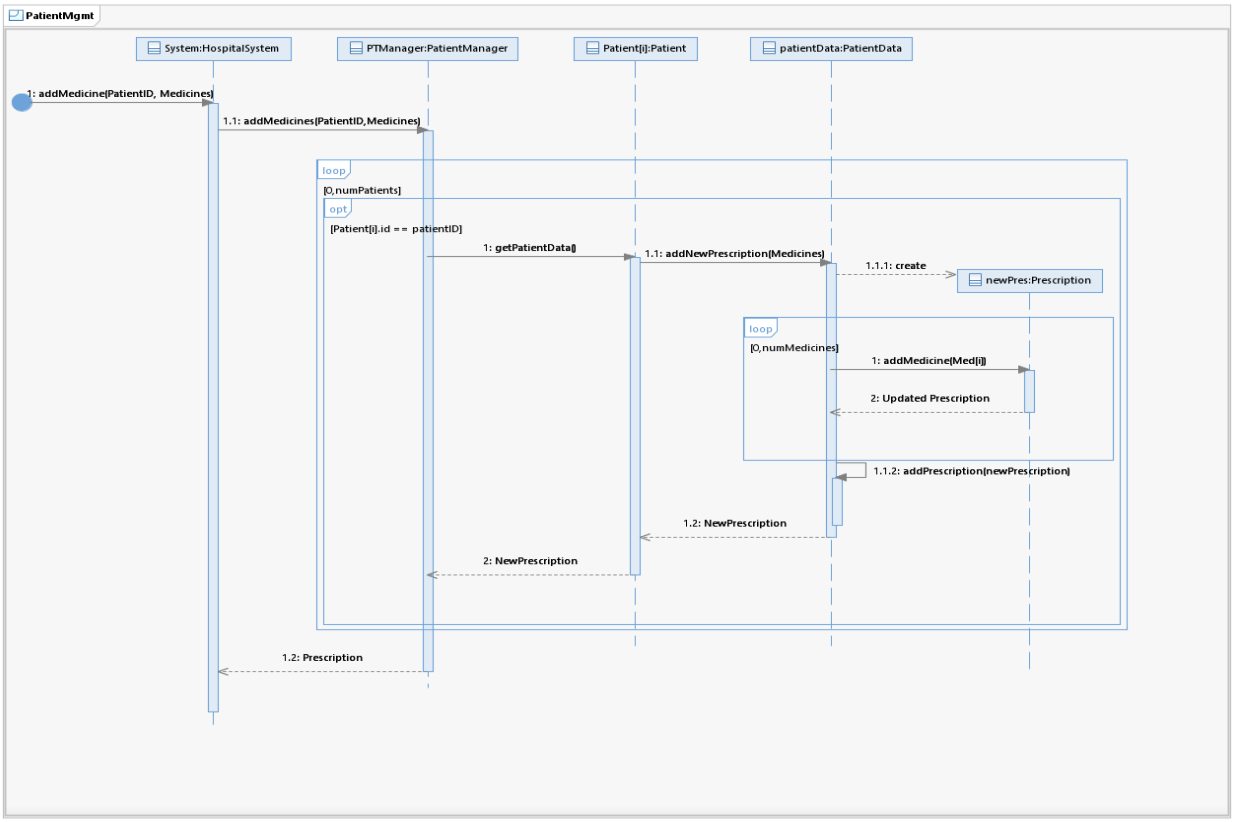

Interaction1
System:Hospital System
Departments[i]:Department
1: EnterDoctorID(DoctorID)
1.1: searchForDoctor(DoctorID)
1.2: Doctor Details

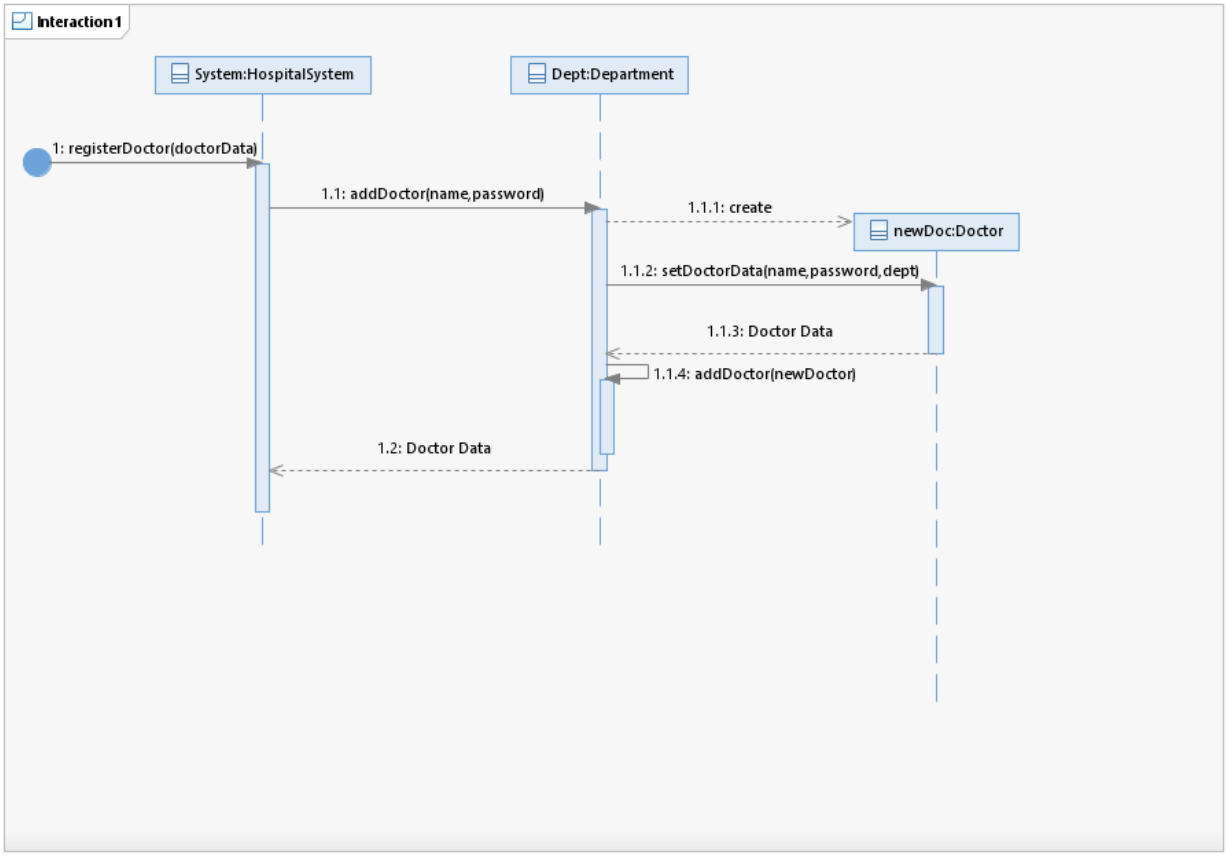

Interaction1
System:Hospital System
dept[i]:Department
doctor:Doctor
meet[i]:Meeting
1: ApplyOperation(ID, Operation,OperationData)
alt
[Option = Remove Doctor]
1: removeDoctor(DoctorID)
2: Removed Status
[Option = Update Timings]
1: updateMeetingTime(MeetID,NewTime)
1.1: updateMeetTime(MeetID, NewTime)
2: setTime(NewTime)
1.2: Updated Meeting Data
3: Updated Meet Data
[Option = Update]
1: updateDoctorData(NewData)
1.1: setData(newData)
2: Updated Doctor Data

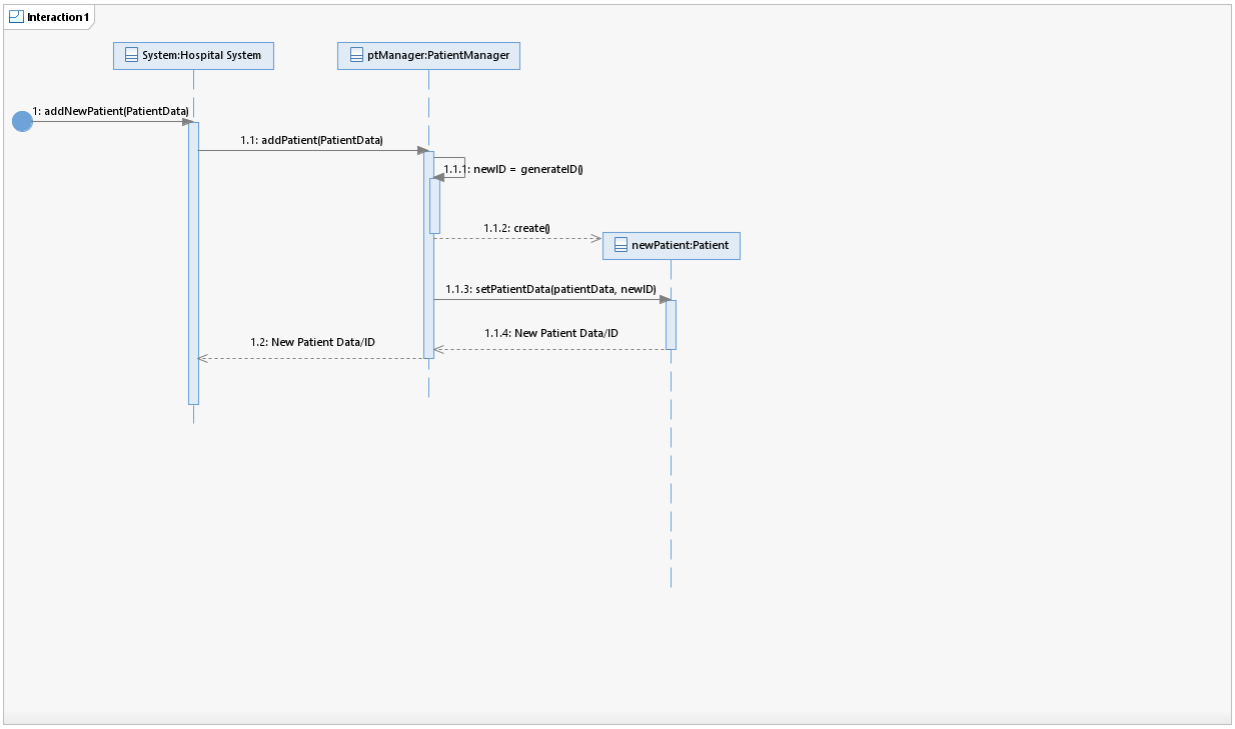

Interaction 1
System:HospitalSystem
WM:WardManager
ward:Ward
bed[i]:WardBed
1: manageWard(Operation,Data)
1.1: manageWardPatients(Operation, Data)
alt
[Operation = Add Patient]
1: addPatient(patientID, BedID)
loop
[0,numBeds]
opt
[bed.id = BedID]
1: updateBedData(PatientID)
2: Update Status
2: Add Status
[Operation = Update Ward]
1: updateBed(BedID)
loop
[0,numBeds]
opt
[bed.id = BedID]
1: updateBedData(Data)
2: Update Status
2: Update Status
[Operation = Remove Patient]
1: removePatient(PatientID)
loop
[0,*]
opt
[Bed[i].patientID = PatientID]
1: freeBed()
2: Bed Status
2: Remove Status
1.2: Status

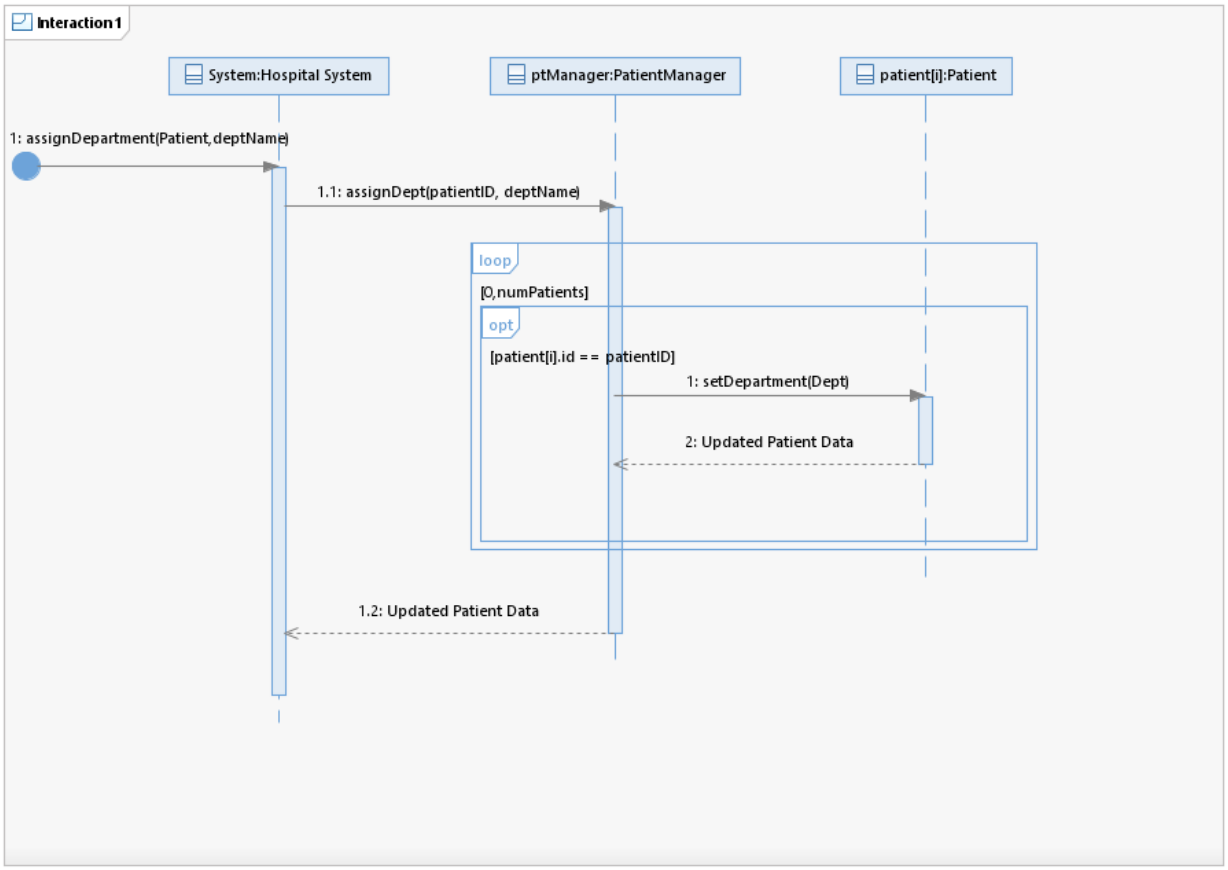

Interaction1
System:HospitalSystem
WM:WardManager
1: getWardInfo() / manageWardPatients()
1.1: getWardInfo()
1.2: Ward Information

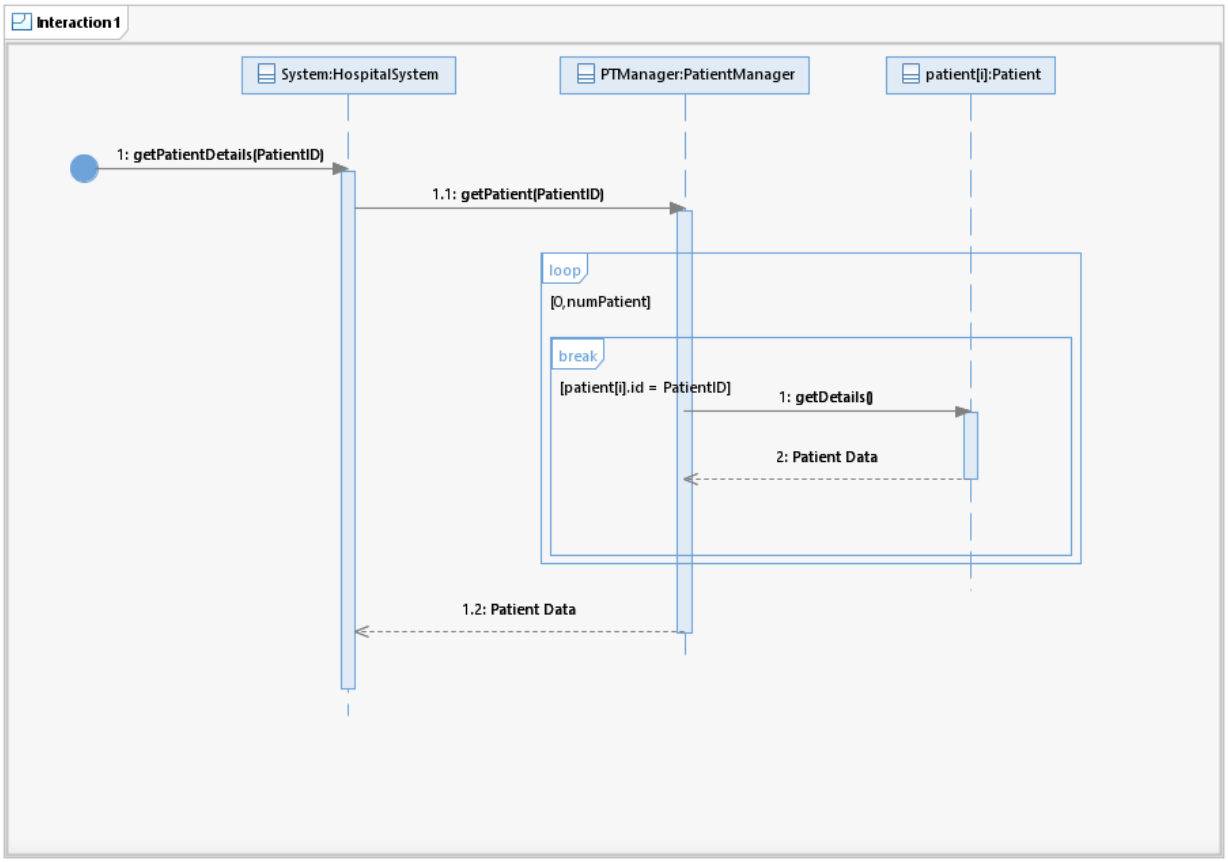

Interaction1
System:HospitalSystem
PTManager:PatientManager
patient[i]:Patient
patientData:PatientData
1: prescribeMedicine(PatientID)
1.1: getPatientPrescription(PatientID)
loop
[0,numPatients]
opt
[Patient[i].id == patientID]
1: getPatientData()
1.1: getPrescriptions()
1.2: Prescription
2: Prescription
1.2: Prescription


PatientMgmt
System:HospitalSystem
PTManager:PatientManager
Patient[i]:Patient
patientData:PatientData
1: addMedicine(PatientID, Medicines)
1.1: addMedicines(PatientID,Medicines)
loop
[0,numPatients]
opt
[Patient[i].id == patientID]
1: getPatientData()
1.1: addNewPrescription(Medicines)
1.1.1: create
newPres:Prescription
loop
[0,numMedicines]
1: addMedicine(Med[i])
2: Updated Prescription
1.1.2: addPrescription(newPrescription)
1.2: NewPrescription
2: NewPrescription
1.2: Prescription


Interaction 1
System:HospitalSystem
Dept:Department
1: registerDoctor(doctorData)
1.1: addDoctor(name,password)
1.1.1: create
newDoc:Doctor
1.1.2: setDoctorData(name,password,dept)
1.1.3: Doctor Data
1.1.4: addDoctor(newDoctor)
1.2: Doctor Data


Interaction1
System:Hospital System
ptManager:PatientManager
1: addNewPatient(PatientData)
1.1: addPatient(PatientData)
1.1.1: newID = generateID()
1.1.2: create()
newPatient:Patient
1.1.3: setPatientData(patientData, newID)
1.1.4: New Patient Data/ID
1.2: New Patient Data/ID


Interaction1
System:Hospital System
ptManager:PatientManager
patient[i]:Patient
1: assignDepartment(Patient, deptName)
1.1: assignDept(patientID, deptName)
loop
[0,numPatients]
opt
[patient[i].id == patientID]
1: setDepartment(Dept)
2: Updated Patient Data
1.2: Updated Patient Data


Interaction 1
System:HospitalSystem
PTManager:PatientManager
patient[i]:Patient
1: getPatientDetails(PatientID)
1.1: getPatient(PatientID)
loop
[0,numPatient]
break
[patient[i].id = PatientID]
1: getDetails()
2: Patient Data
1.2: Patient Data

7. Class Diagram