Software Requirements and Design Document

for

CATALYST – INTEGRATED HOSPITAL MANAGEMENT SYSTEM

Farheen Malik

Project

3rd December, 2023

Table of Contents

- **Introduction**

  - **Purpose**

This project aims to develop an effective management system for the healthcare facilities in rural areas in order to increase productivity, ensure data security, and minimize errors. We will work on developing this software in best interests of the healthcare facilities.

  - **Product Scope**

We will develop this software for hospitals in the rural areas. We will work on the operating system, patient management, scheduling, reporting, and billing of the hospital operations. In regards with the future, we will be further expanding the data annalistic capabilities ensuring enhanced future growth.

  - **Title:**

Catalyst – Integrated Hospital Management System

  - **Objectives**

Our main goals are
1. To improve the management system of the hospitals in rural areas,

2. To create a user-friendly hospital management system to enhance the quality,
3. To improve staff productivity by automating routine task,
4. To enhance data accuracy and security, and
5. To minimize the error.

- **Problem Statement**

We chose this project because there is a pressing need to address the challenges faced by our healthcare facilities. In rural areas, there aren't any proper record keeping methods which have led to errors and delays. For smooth treatment of patients, a proper management system is required.

Our hospital management system will work toward eliminating these problems by creating a digitalized platform where patient records, appointment scheduling, and billing will be stored within a software, making the process easy and effective.
This will improve data accuracy and ensure security and privacy to confidential information.

- **Overall Description**

  - **Product Perspective**

The software is a standalone product designed to address the unique needs of hospitals in rural areas. It is not a replacement but an enhancement to existing hospital management systems

  - **Product Functions**

- Login
- Register
- ManagePatientRecord
- ScheduleAppointment
- InventoryManagement
- BillingAndPayments
- GenerateReports
- Feedback
- PatientCheckin/Checkout
- AdministerSystem

  - **List of Use Cases**

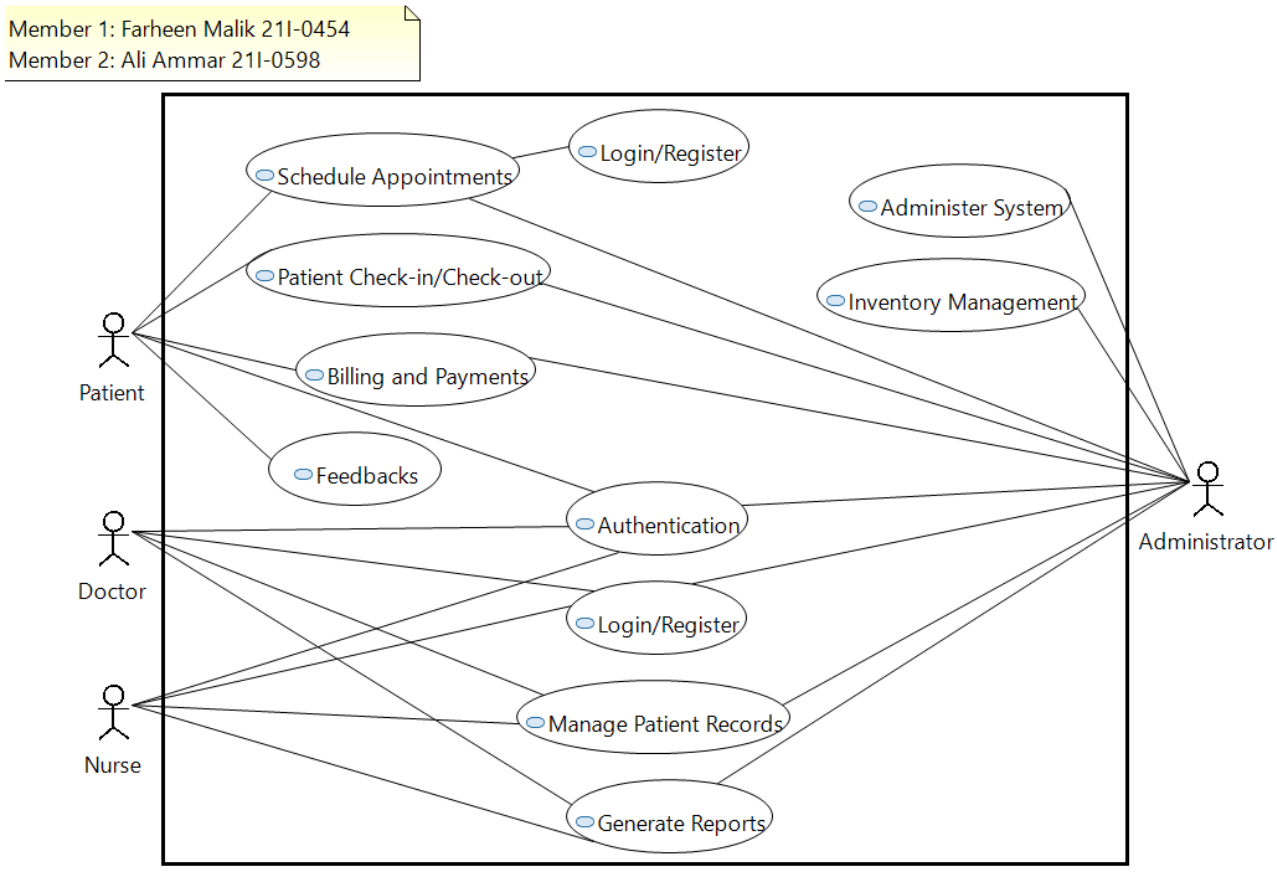
- Login
- Register
- ManagePatientRecord
- ScheduleAppointment
- InventoryManagement
- BillingAndPayments

- GenerateReports
- Feedback
- PatientCheckin/Checkout
- AdministerSystem

○ Extended Use Cases

Showed in delieverable 3.

○ Use Case Diagram



- **Other Nonfunctional Requirements**
  - **Performance Requirements**

The performance requirements are that the system must ensure that the expected load can be handled, respond swiftly to user actions, and have good running performance when in use.

Responsiveness

It is important to provide prompt response times to support fast user experience and real-time collaboration.

Scalability

The system must therefore be able to continue to perform to optimal levels as the number of users increases.

Task Loading Time

Fast loading speeds enhance the efficiency of task management and user productivity.

Data Retrieval Time

Fast data retrieval makes possible instant access to project information to prevent wasting time for users.

System Uptime

Ensures users have reliable availability so that they can have access to the system whenever needed and thereby limiting disruptions.

Integration Response Time

This includes providing for a good user experience and interoperability with other systems.

- **Safety Requirements**

In addition, we must have healthy encryption processes to protect the login credentials, as well as user information from other projects.

Periodic security audits and patching for vulnerabilities. The ability to comply with industry-standard encryption protocols.

Ensure data security by tightening access controls and periodical update of security policies.

- **Security Requirements**

Ensuring the security and privacy of sensitive data is a top priority. The software should adhere to the following security requirements:

- Data Encryption:

All communication between the client and the server should be encrypted using industry-standard protocols.

- User Identity Authentication:

Users must undergo secure authentication processes before accessing sensitive information.

- Regulatory Compliance:

The system must comply with relevant data protection regulations and standards.

- **Software Quality Attributes**

To meet the expectations of both customers and developers, the software should exhibit the following quality attributes:

- Reliability:

The system should operate consistently without unexpected failures.

- Maintainability:

Code should be well-documented, and the system should allow for easy updates and modifications.

- Usability:

The user interface should be intuitive, and users should be able to perform tasks with minimal training.

- Interoperability:

The software should seamlessly integrate with other healthcare systems and standards.

- **Business Rules**

The following business rules outline operating principles within the product:

- User Roles:

Different roles (e.g., administrator, medical staff) will have specific permissions and access levels.

- Appointment Scheduling:

Patients can schedule appointments, but only administrators can modify the master schedule.

- Billing Rules:

Only authorized personnel can initiate billing processes.

- **Operating Environment**

The software will operate in the following environment:

- Hardware Platform:

The system should run on standard x86 architecture with a minimum of 8GB RAM.

- Operating System:
Compatible with Windows Server 2016 and above, Linux Ubuntu 18.04 and above.

- Software Components:

Requires a relational database (e.g., MySQL) and Java Runtime Environment (Intellij) for server deployment.

- **User Interfaces**

Logical characteristics of the user interface include:

- GUI Standards:

The GUI will adhere to industry standards for healthcare software interfaces.

- Layout Constraints:

Screens will follow a consistent layout for ease of use.

- Standard Buttons:

Common functions (e.g., help, save) will be accessible through standard buttons.

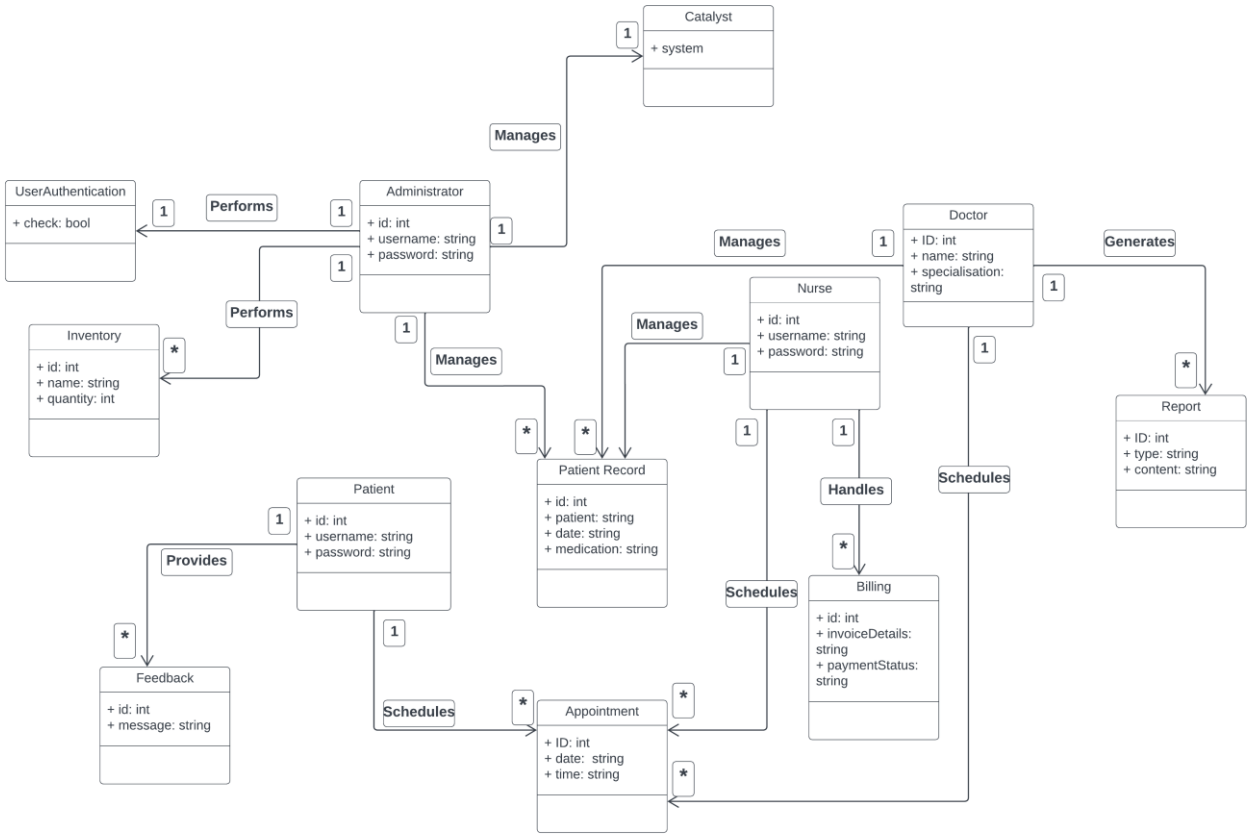
- Error Messages:

Errors will be displayed in a standard format, indicating the nature of the issue.

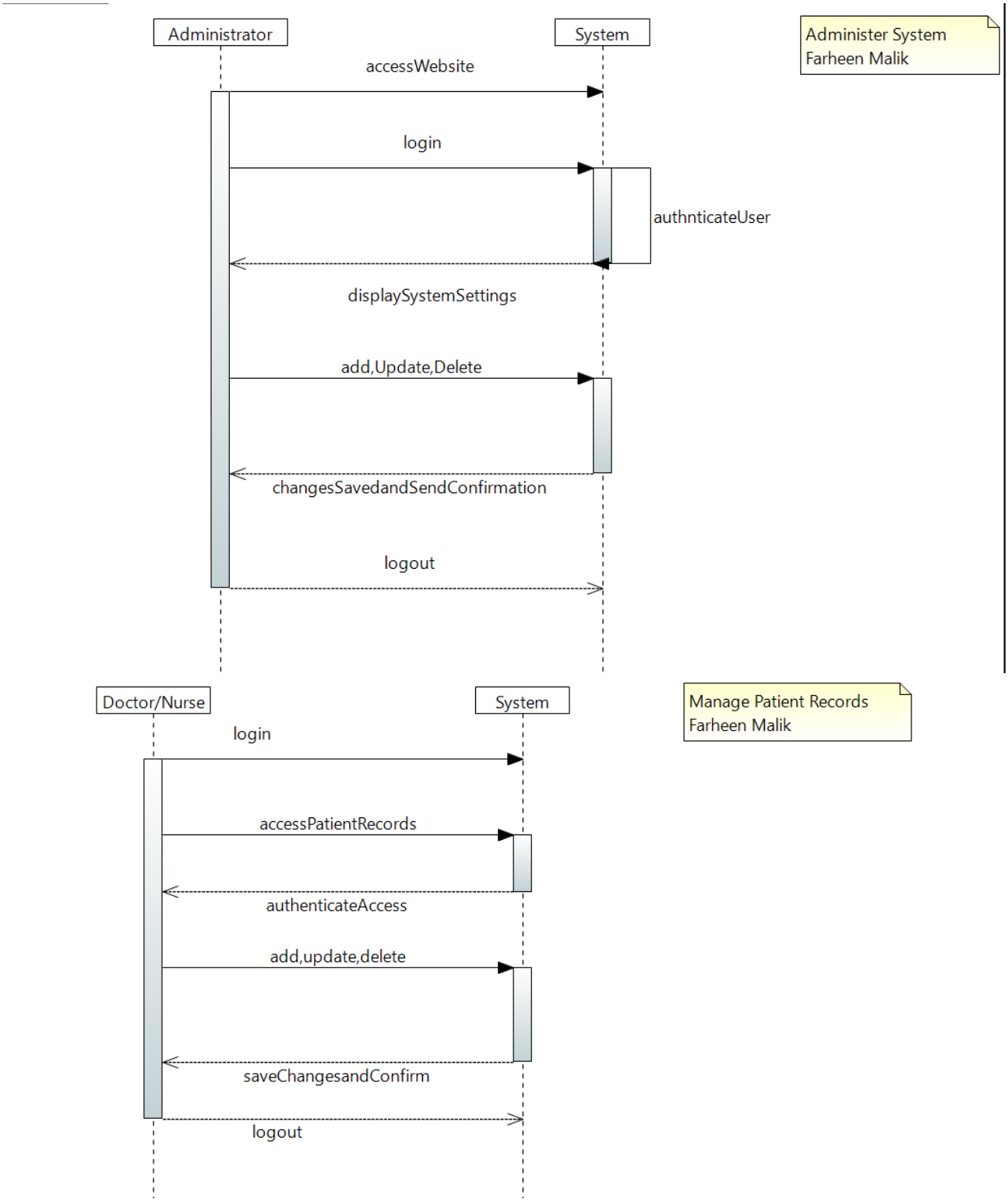
- User Interface Components:

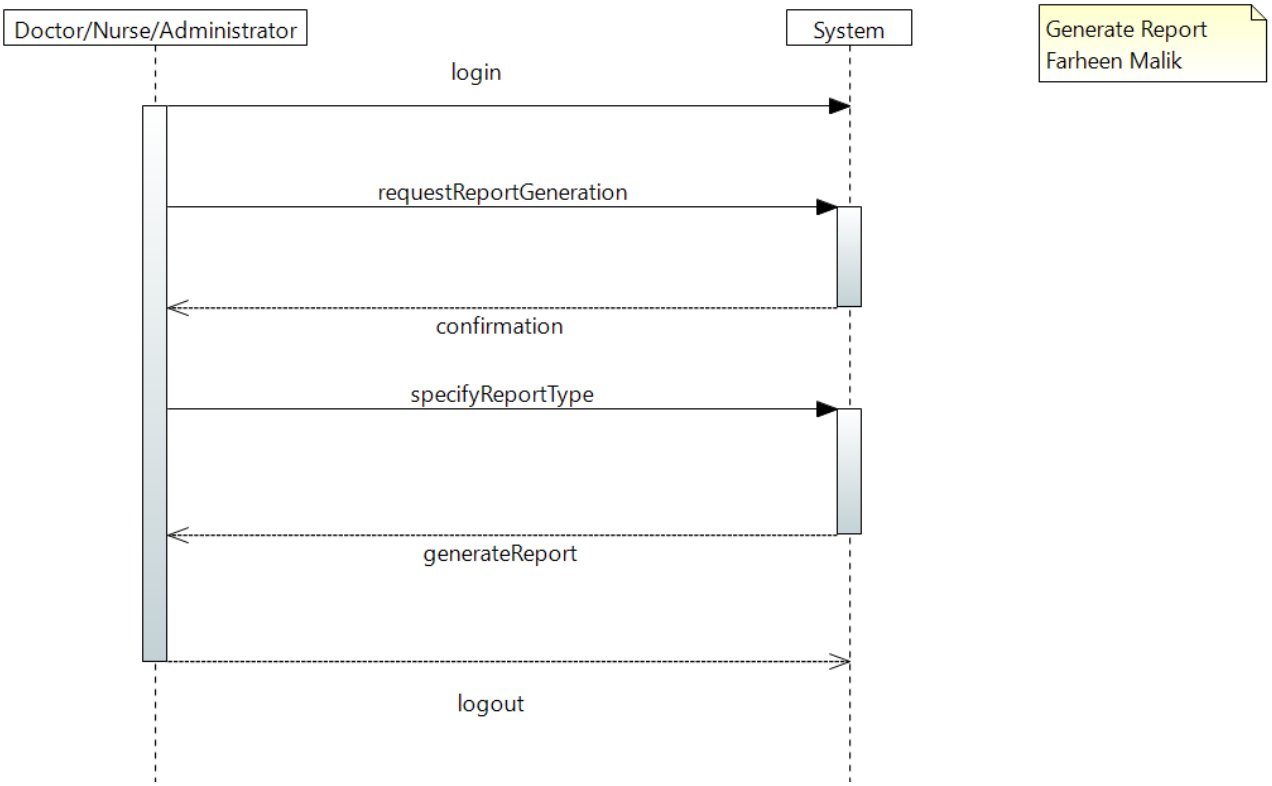
UI components include patient information forms, scheduling interfaces, and billing modules.

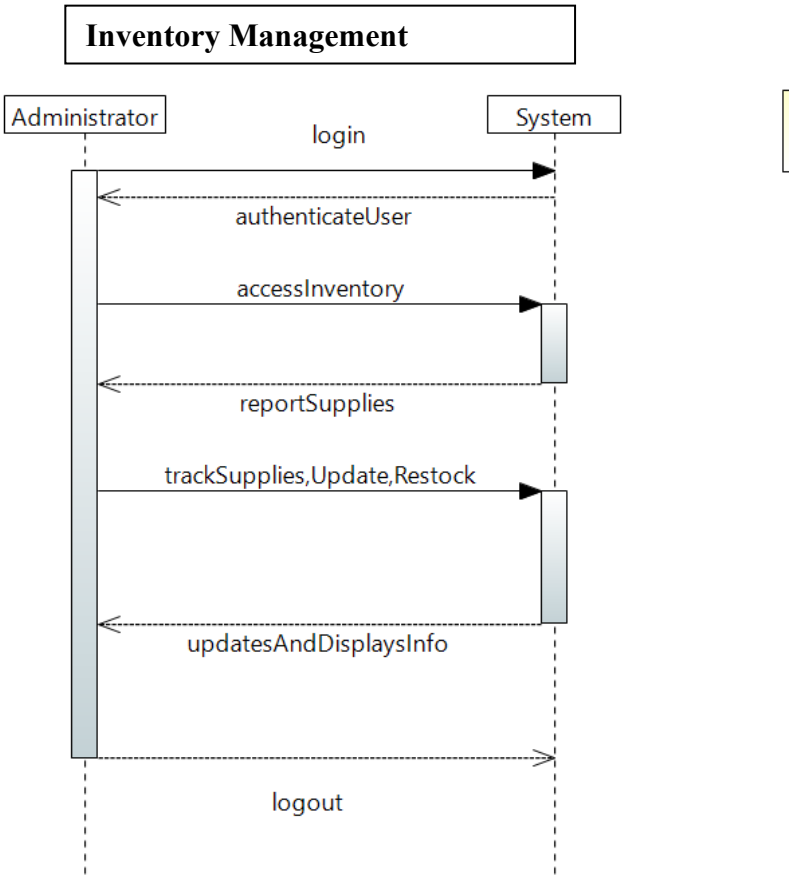
- **Domain Model**

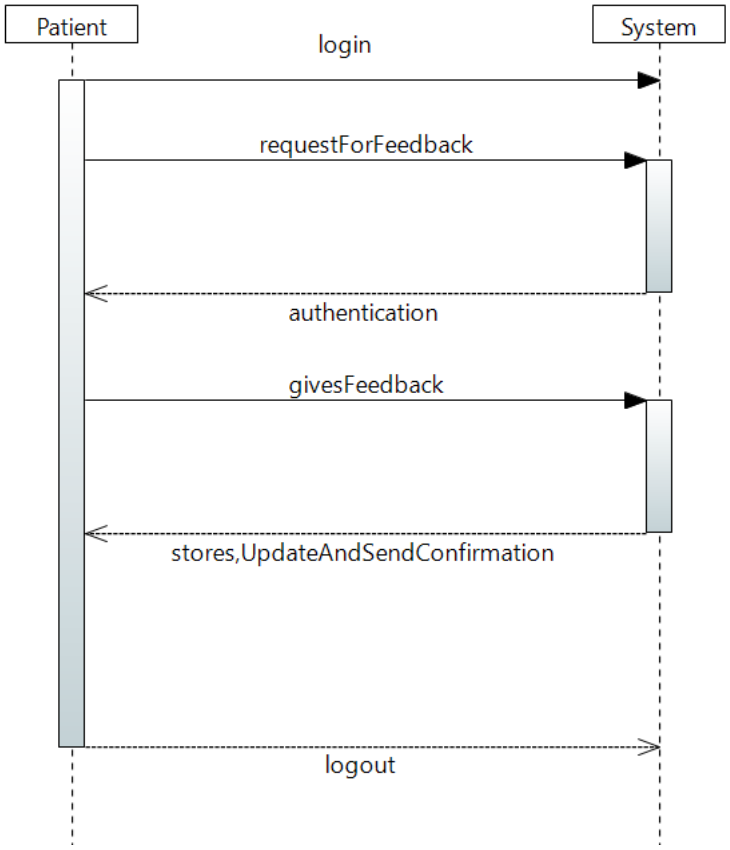


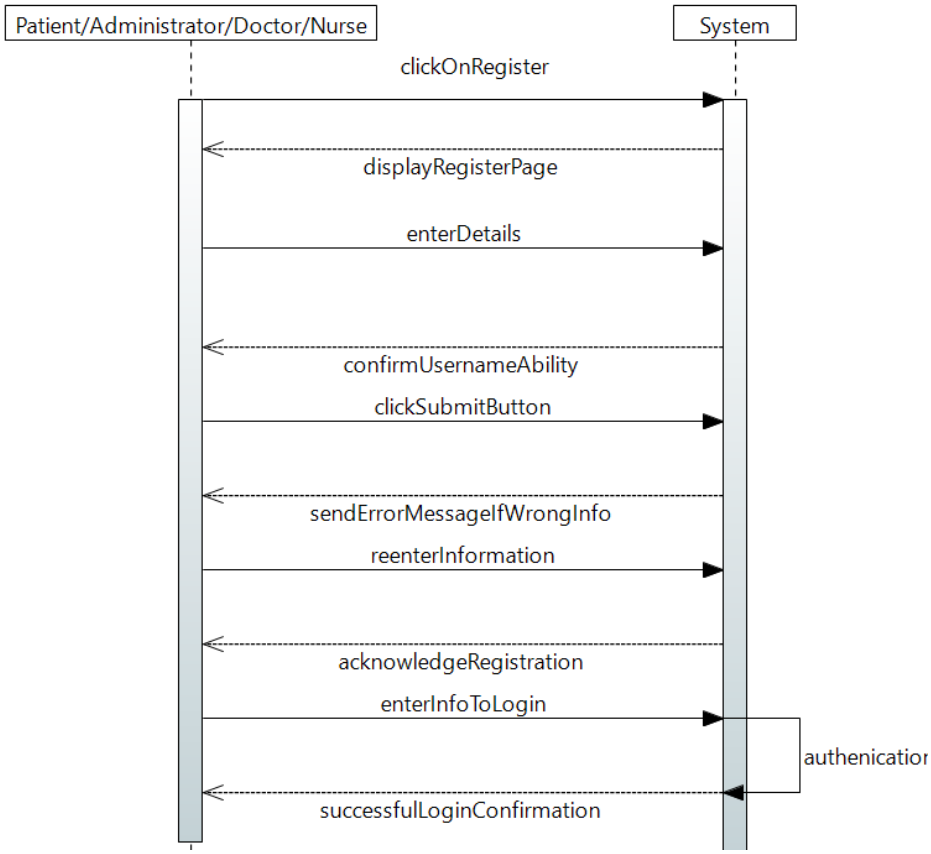
- **System Sequence Diagram**

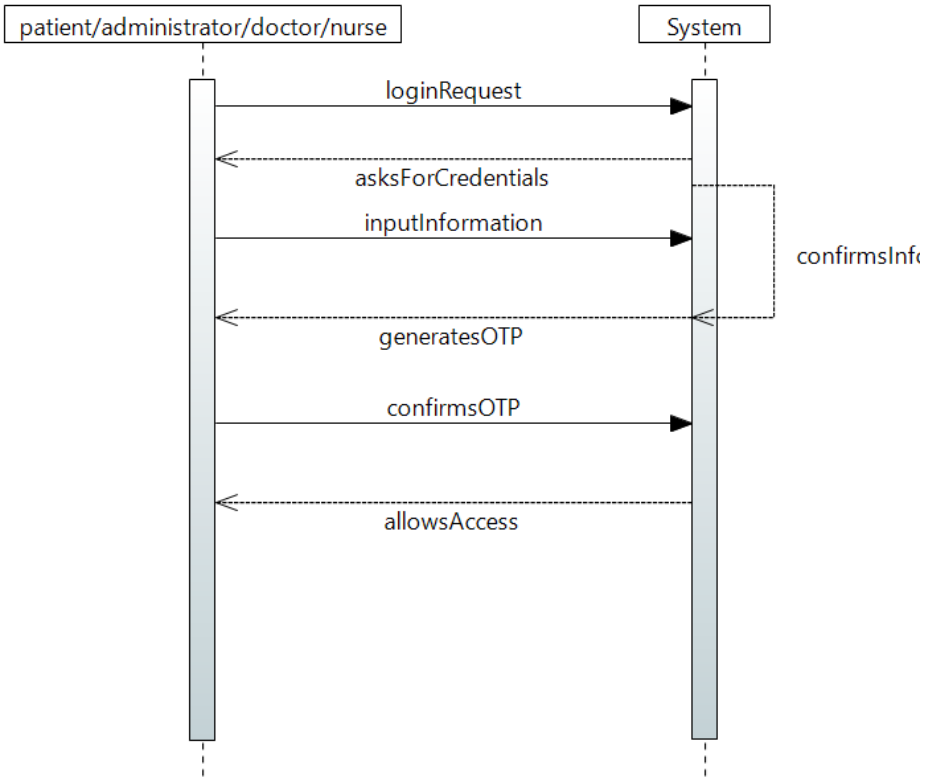

Administrator
System
Administer System
Farheen Malik
accessWebsite
login
authnticateUser
displaySystemSettings
add,Update,Delete
changesSavedandSendConfirmation
logout
Doctor/Nurse
System
Manage Patient Records
Farheen Malik
login
accessPatientRecords
authenticateAccess
add,update,delete
saveChangesandConfirm
logout


Doctor/Nurse/Administrator
System
Generate Report
Farheen Malik
login
requestReportGeneration
confirmation
specifyReportType
generateReport
logout

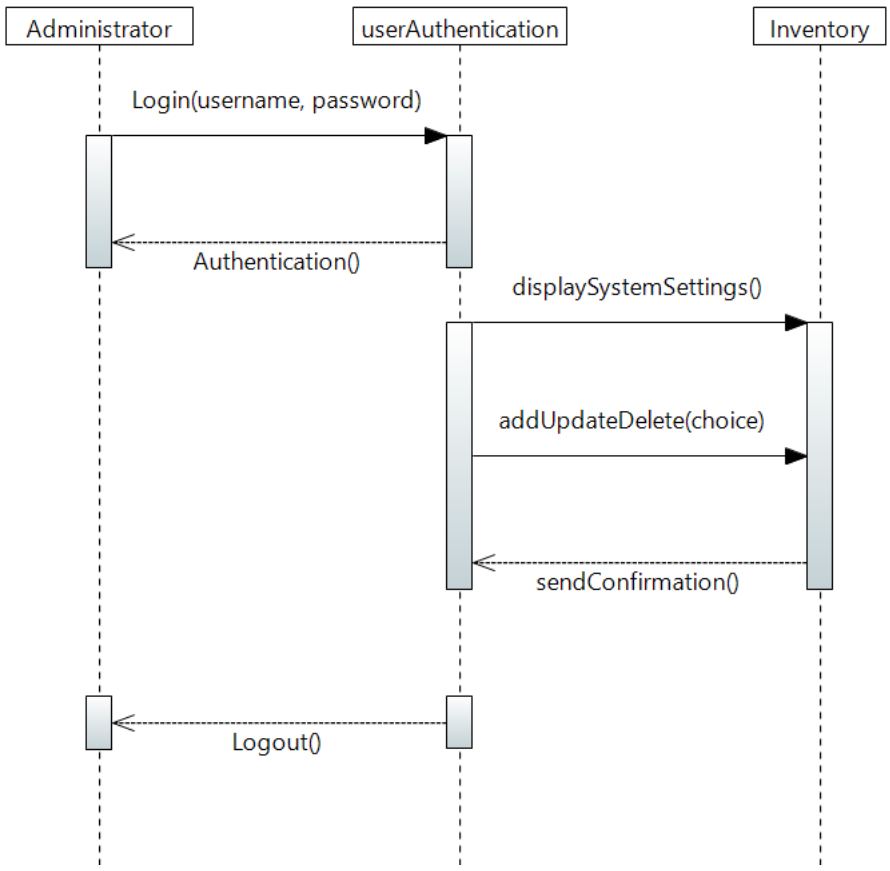

Inventory Management
Administrator
System
login
authenticateUser
accessInventory
reportSupplies
trackSupplies,Update,Restock
updatesAndDisplaysInfo
logout

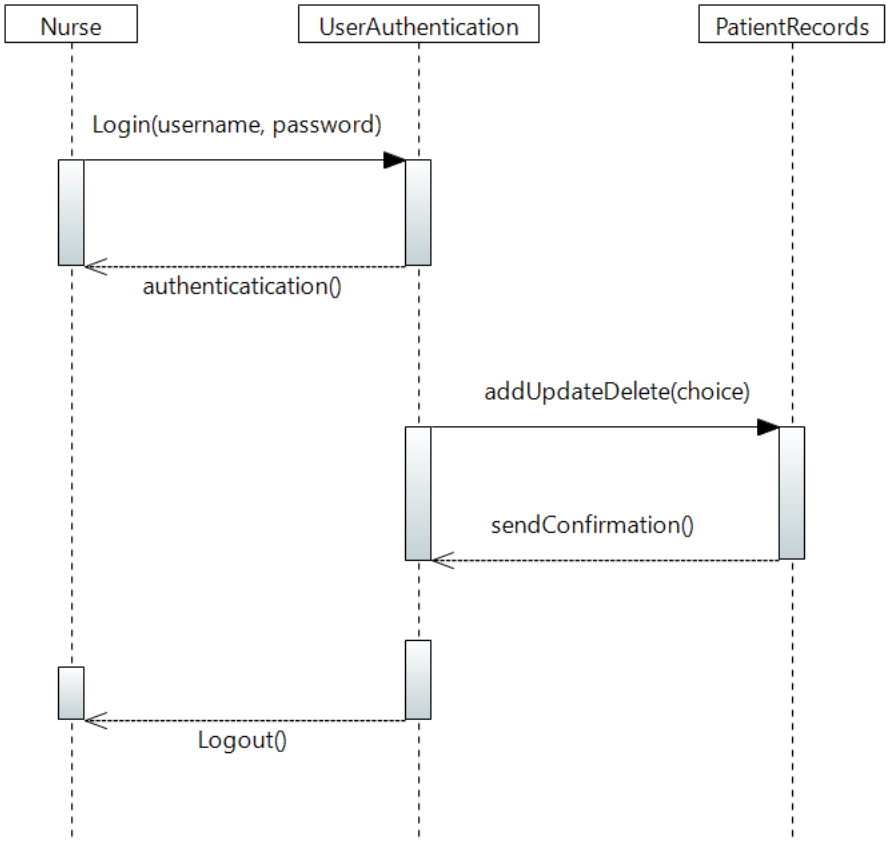

Patient
System
login
requestForFeedback
authentication
givesFeedback
stores,UpdateAndSendConfirmation
logout

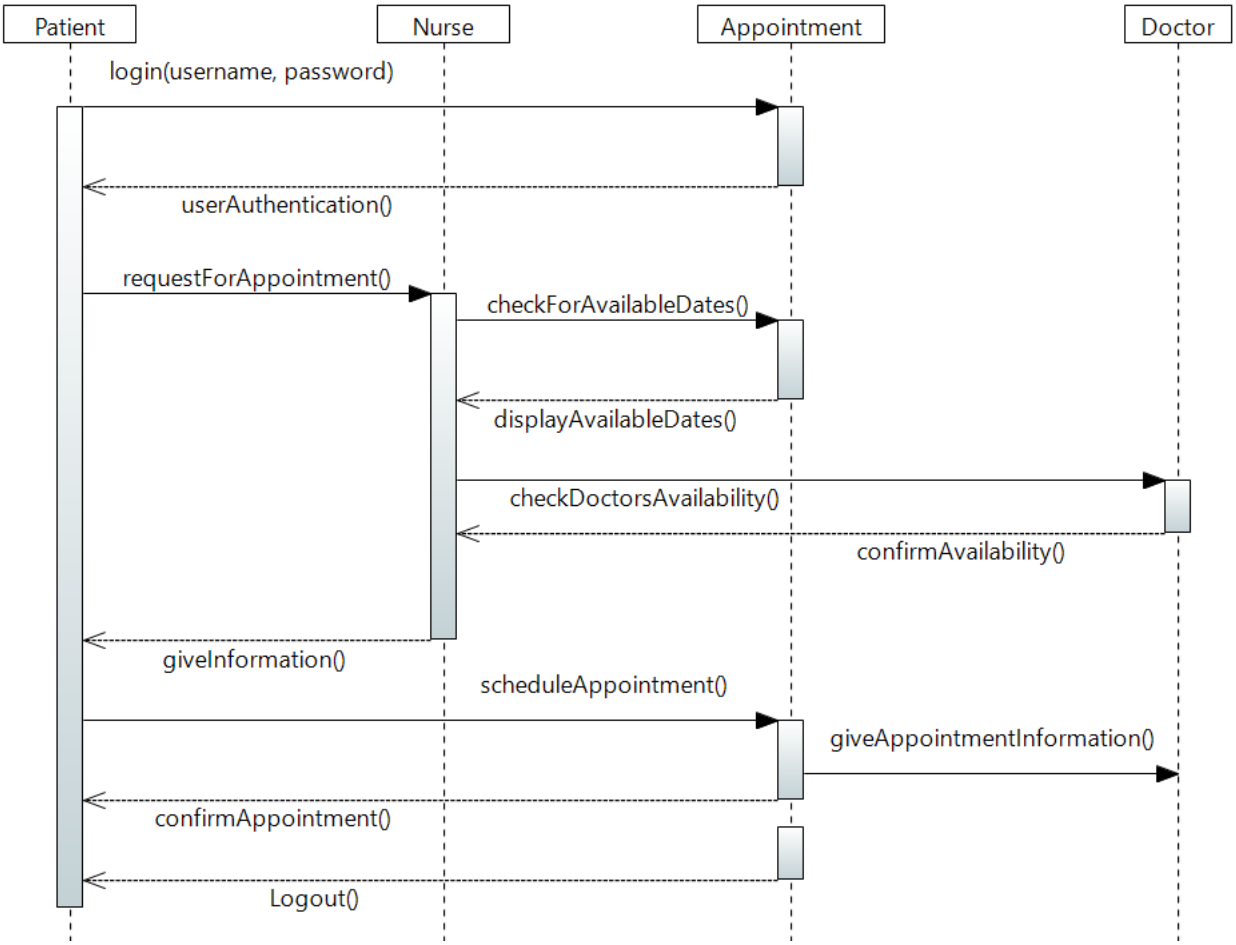

Patient/Administrator/Doctor/Nurse
System
clickOnRegister
displayRegisterPage
enterDetails
confirmUsernameAbility
clickSubmitButton
sendErrorMessageIfWrongInfo
reenterInformation
acknowledgeRegistration
enterInfoToLogin
authenicatior
successfulLoginConfirmation

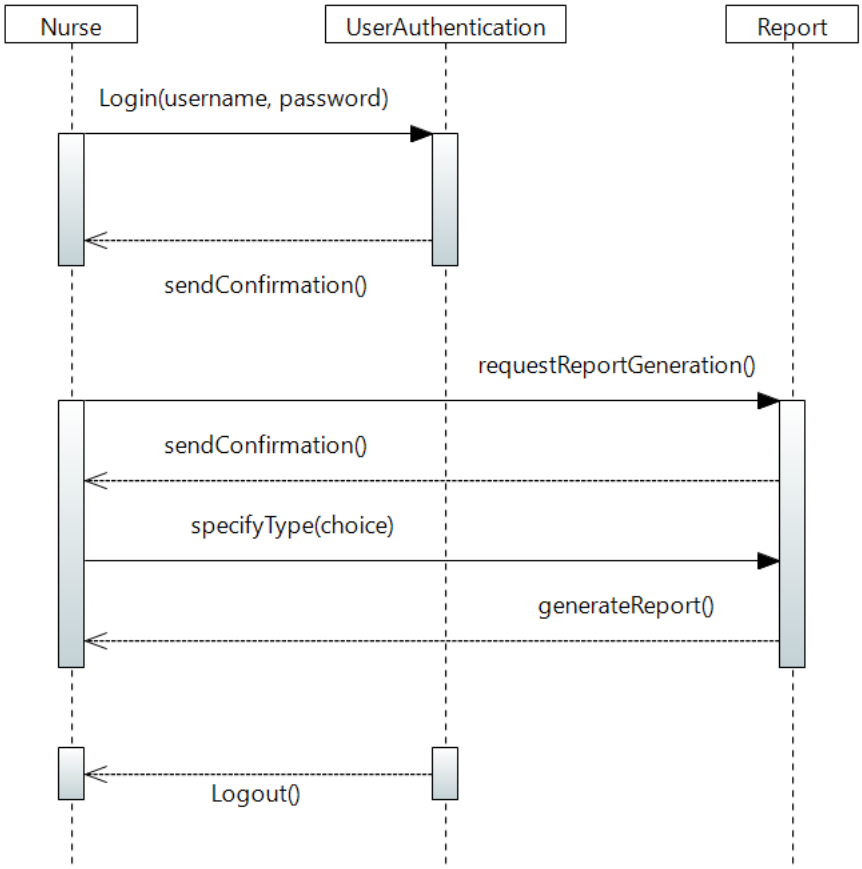

patient/administrator/doctor/nurse
System
loginRequest
asksForCredentials
inputInformation
confirmsInf
generatesOTP
confirmsOTP
allowsAccess

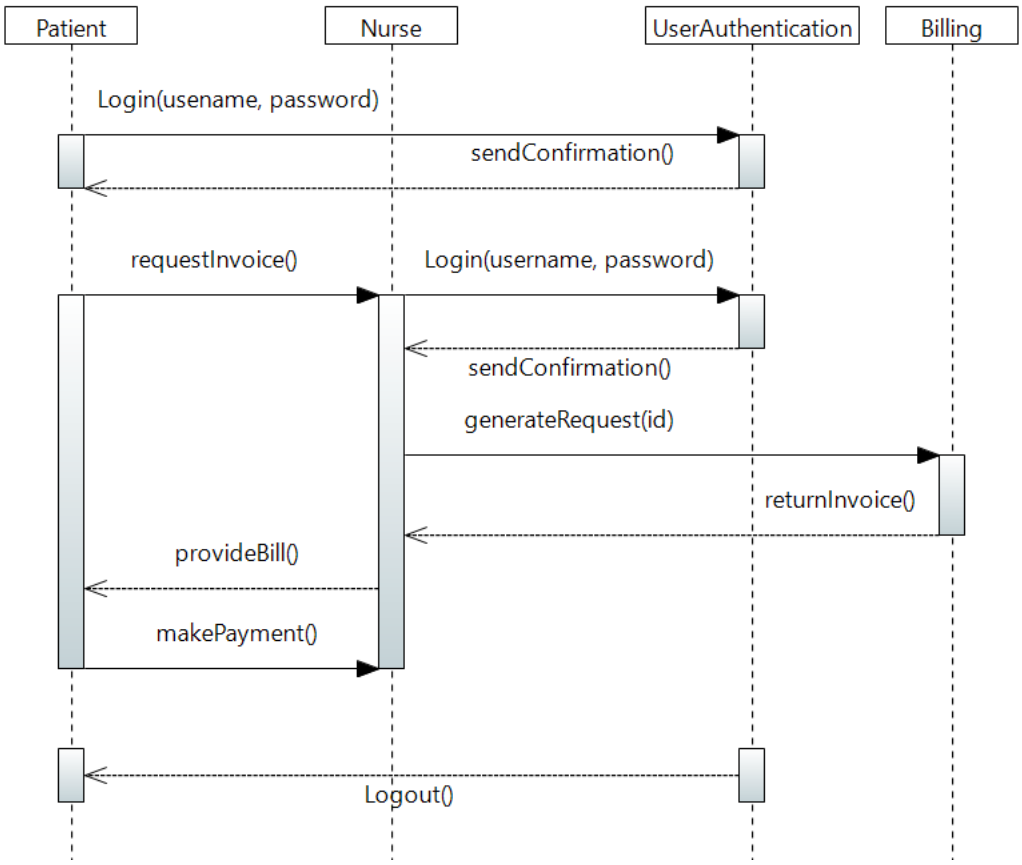
- **Sequence Diagram**

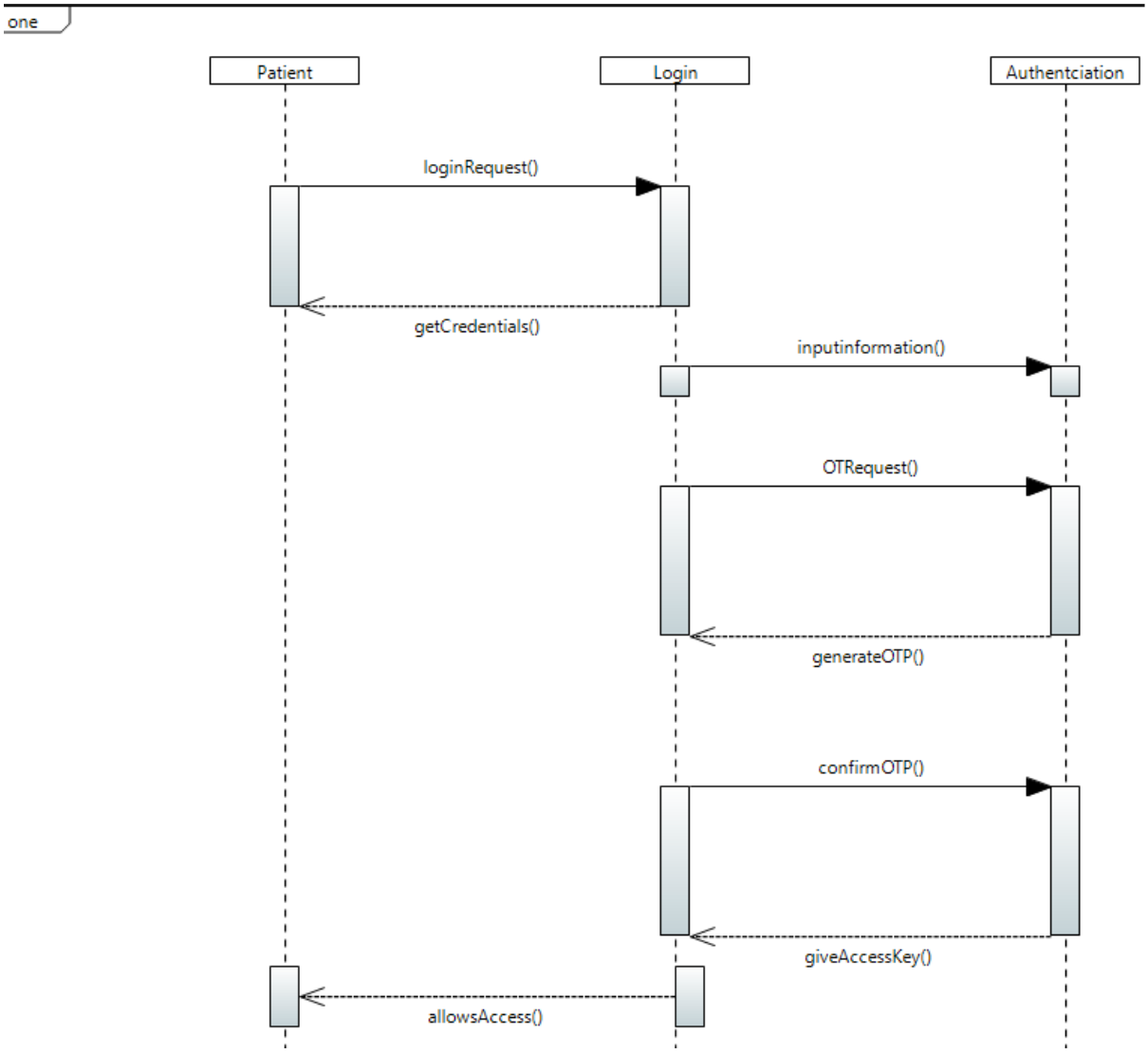

Administrator
userAuthentication
Inventory
Login(username, password)
Authentication()
displaySystemSettings()
addUpdateDelete(choice)
sendConfirmation()
Logout()

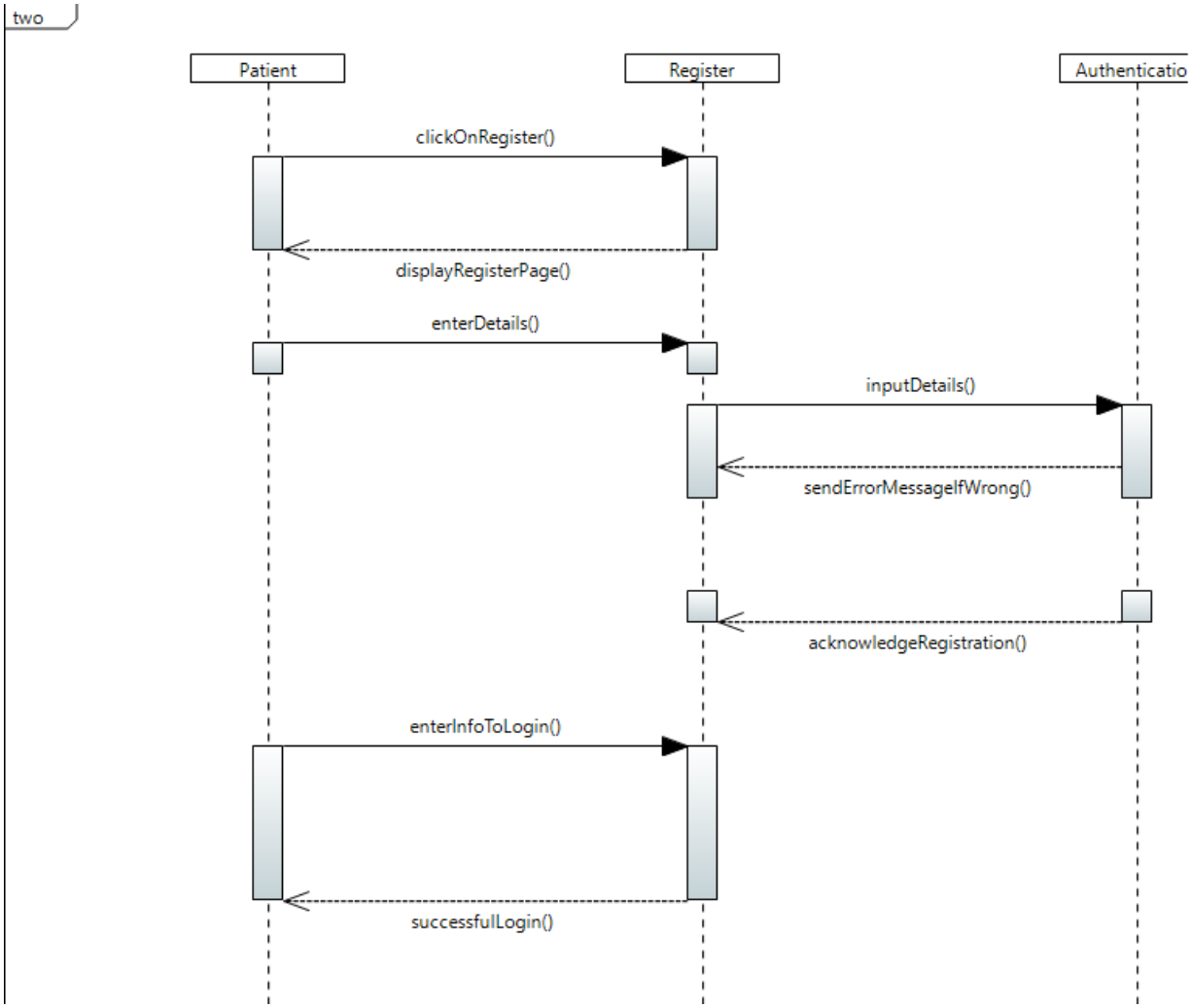

Nurse
UserAuthentication
PatientRecords
Login(username, password)
authenticatication()
addUpdateDelete(choice)
sendConfirmation()
Logout()

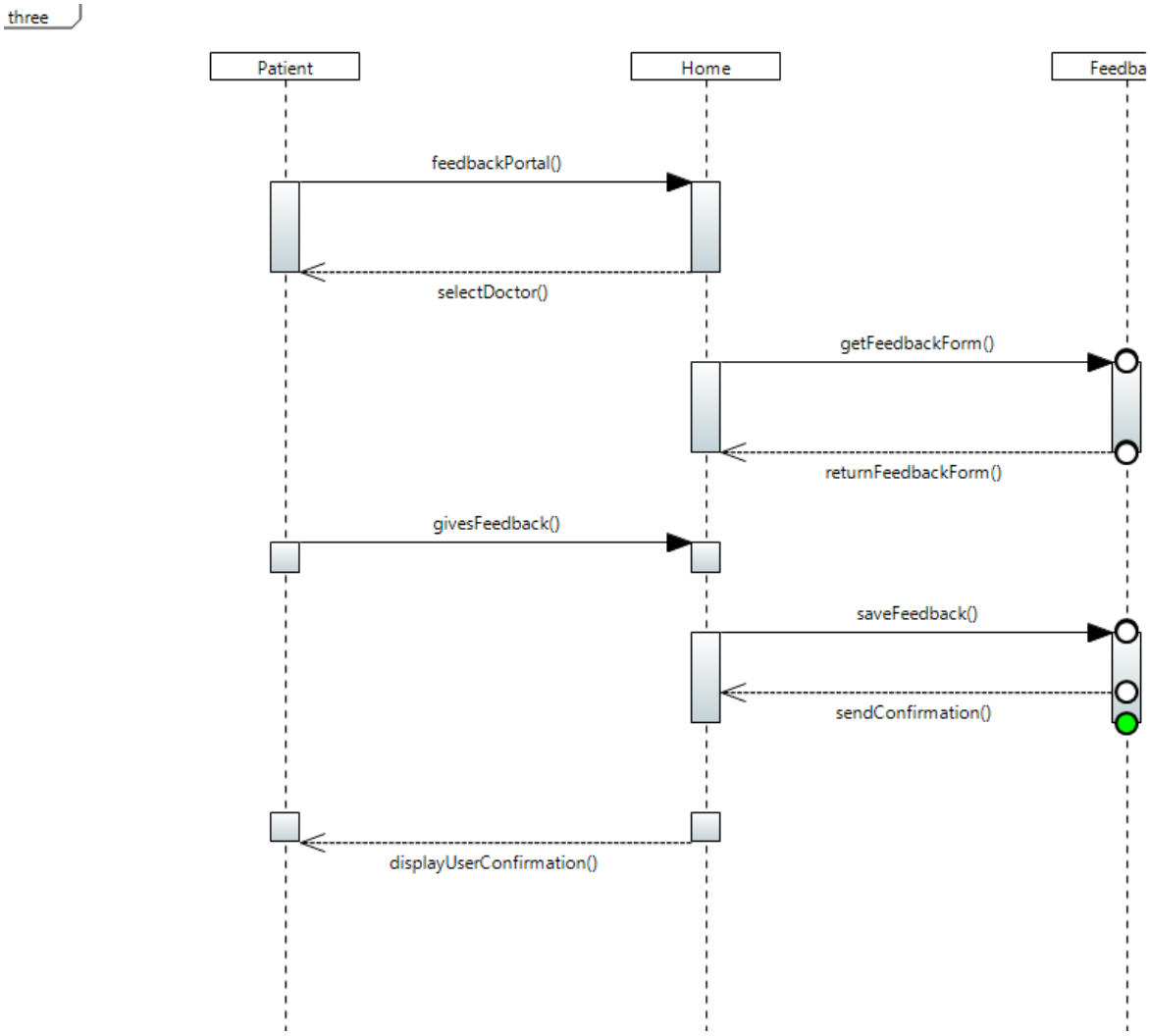

Patient
Nurse
Appointment
Doctor
login(username, password)
userAuthentication()
requestForAppointment()
checkForAvailableDates()
displayAvailableDates()
checkDoctorsAvailability()
confirmAvailability()
giveInformation()
scheduleAppointment()
giveAppointmentInformation()
confirmAppointment()
Logout()

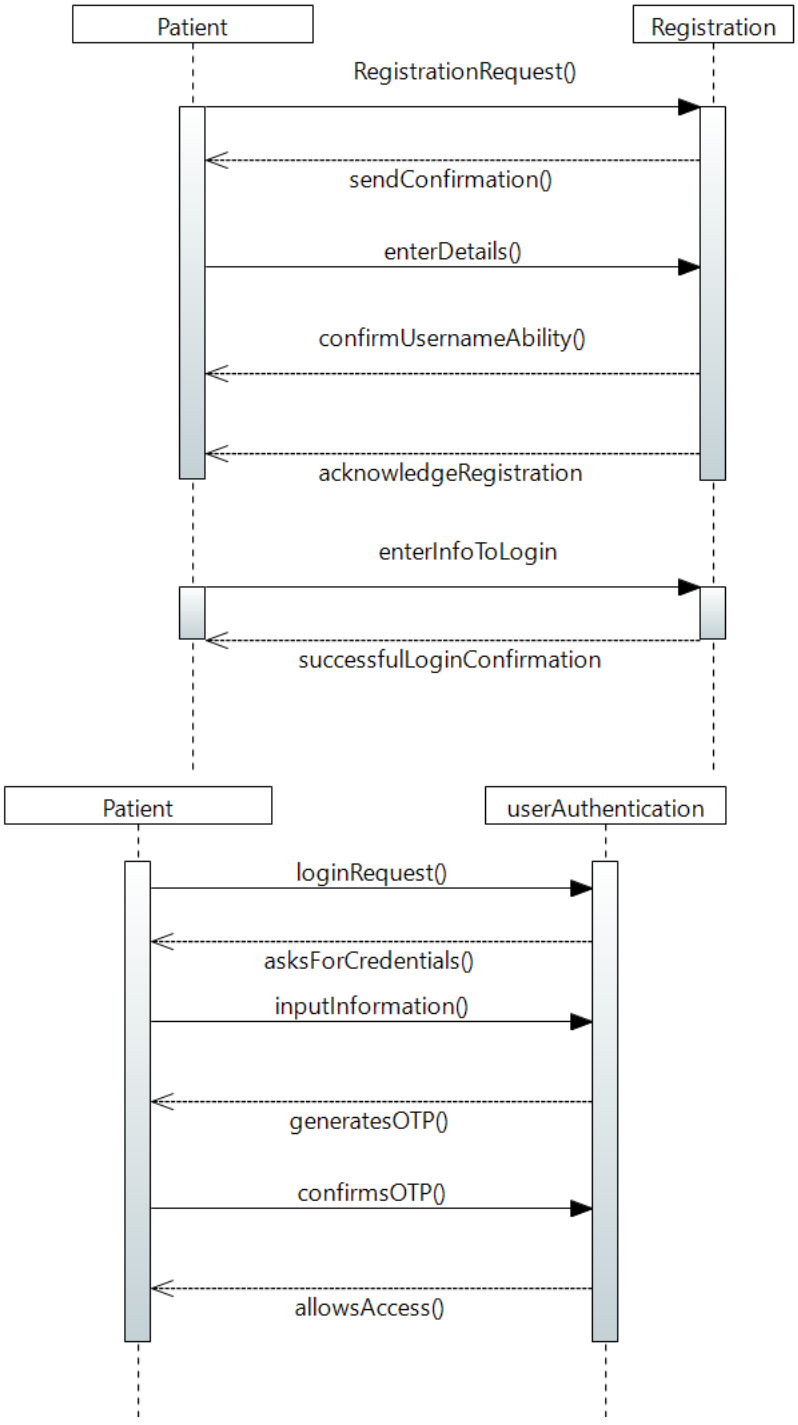

Nurse
UserAuthentication
Report
Login(username, password)
sendConfirmation()
requestReportGeneration()
sendConfirmation()
specifyType(choice)
generateReport()
Logout()


Patient
Nurse
UserAuthentication
Billing
Login(usename, password)
sendConfirmation()
requestInvoice()
Login(username, password)
sendConfirmation()
generateRequest(id)
returnInvoice()
provideBill()
makePayment()
Logout()


one
Patient
Login
Authentciation
loginRequest()
getCredentials()
inputinformation()
OTRequest()
generateOTP()
confirmOTP()
giveAccessKey()
allowsAccess()


two
Patient
Register
Authenticatio
clickOnRegister()
displayRegisterPage()
enterDetails()
inputDetails()
sendErrorMessageIfWrong()
acknowledgeRegistration()
enterInfoToLogin()
successfulLogin()


three
Patient
Home
Feedba
feedbackPortal()
selectDoctor()
getFeedbackForm()
returnFeedbackForm()
givesFeedback()
saveFeedback()
sendConfirmation()
displayUserConfirmation()


Patient
Registration
RegistrationRequest()
sendConfirmation()
enterDetails()
confirmUsernameAbility()
acknowledgeRegistration
enterInfoToLogin
successfulLoginConfirmation
Patient
userAuthentication
loginRequest()
asksForCredentials()
inputInformation()
generatesOTP()
confirmsOTP()
allowsAccess()

- # Class Diagram