

Patient Tracker System

Final Report

Siara Saylor

INFO-C451

Table of Contents

Customer Problem Statement	3
Glossary of Terms	5
System Requirements	6
Functional Requirements.....	6
Nonfunctional Requirements.....	7
User Interface Requirements	8
Functional Requirement Specification.....	15
Stakeholders	15
Actors and Goals	15
Use Cases	16
User Effort Estimation	27
Domain Analysis	28
Domain Model.....	28
System Operation Contracts	30
Interaction Diagrams	39
Class Diagram.....	50
Class Diagram	50
Data Types	51
Operation Signatures	52
Traceability Matrix.....	58
System Architecture	60
Architectural Styles	60
Identifying Subsystems	60
Mapping Subsystems to Hardware	60
Persistent Data Storage.....	61
Network Protocol	61
Global Control Flow.....	61
Execution Order.....	61
Time Dependency	61
Concurrency.....	61
Hardware Requirements.....	61
Algorithms and Data Structures	63
Data Structure.....	63
User Interface Design and Implementation.....	63
Design of Tests	64
History of Work, Current Status, and Future Work.....	65
Project Coordination and Progress Report	65
History of Work	65
Accomplishments in the Project.....	65
Gantt chart.....	66
Summary of Changes	67
References	68

Customer Problem Statement

Patients at hospitals and doctor's offices have to call to make an appointment, or they leave the appointment and can't remember what the doctor said. This has happened to many patients, and it can be frustrating playing phone tag with the doctor's office staff, just to get information about their appointment or schedule a new appointment.

With this system application, it is mainly focused around patients and their needs. The patients can schedule their appointments online, fill out the pre appointment paperwork ahead of time, and even be able to see what happened in their prior appointments. This application will allow shorter waiting times for appointments, due to the pre appointment paperwork already being filled out before the patient shows up to the appointment. Patients can also update their personal information through the system. Patients having the ability to do this themselves, will save time for the office staff from updating patient information. Along with scheduling an appointment, patients can also cancel or reschedule an appointment.

The office staff will have more free time to focus on the other tasks that they are assigned to, versus handing out information for the patient to fill out, updating the system with the new information, scheduling appointments, and updating patients on their appointments. This application will not remove all of these tasks entirely, because there may be some people that do not prefer to use the system, but it will help cut down on the redundancy of doing these tasks. Office staff will have to add any new patients into the system, but it will create them a username and password so the patient will have the opportunity to view and update their information themselves.

One of the amazing features of this application is allowing patients and staff to view prior appointment details. This actively demonstrates that a patient or staff can see the patient's test

results; weight, height, the reason for the appointment, the date of the appointment, the provider, and information on any of the prescriptions they may have been prescribed. By having this option in the application, the patient can view this at any time after the appointment. Sometimes when people go to the doctor they don't feel well, which can cause them to not focus as clearly on the instructions the doctor is providing. With this application, the patient can view the notes after the appointment as many times as they like, so they will not forget right when the doctor leaves the room. The nurses and doctors will keep their notes for the patient's appointment in the system as well. Then once the appointment is over, they click update and the patient receives an email that they can review the appointment.

Each person, staff or patient, will have a username and password in order to enter the application. This actively demonstrates that not anyone will be allowed to view patient records. For the exact purpose of this application, there will be an option for patients to download the application onto their computers. This application will not be available to anyone, it has to be sent out for users to download.

The future updates to this application will include medication refill requests, and a myHealth page. The medication refill requests will allow patients to request refills on prescriptions without having to call the doctor's office and having them send it over. The user has the ability to request any the prescriptions they have available and send the request to the doctor's office. The nurses and doctors will have the ability to view the requests and either approve or deny the request. If the request is approved, the prescription will be sent to the pharmacy of the patient's choice for pick up.

The myHealth page will combine all the information taken from the patient's previous appointments and allow the patient to view their progress. Charts will be provided to allow the

patient to view how their body has changed from appointment to appointment. Nurses and doctors will have visibility of patient myHealth pages as well, so they too can see the patient's development over time.

To summarize everything that has been stated so far, patients need a system that will allow them to access their medical records, create appointments, update their information, and view the status of their health. This system will also allow the staff to cut down on front desk phone calls and are able to focus on other tasks they may still have. Overall, this system will be very beneficial to not only the patients, but to the staff as well.

Glossary of Terms

Insurance Provider
Placeholder
Membership Number
Placeholder

Known Allergies
Type here
List of Medications
Type here

Pre Appointment Information

- **Pre Appointment Information** – This information will include confirming insurance information, any medications the patient is taking, and any known allergies.

- **Medication Refills** – This occurs when patients are prescribed medication that will allow refills. This could include, but not limited to allergy medication, headache medication, birth control, etc. Not all medications will be available for refills. A list of available medication refills will be provided when the

patient goes to request it.

- **Usernames** – Each user will be provided with a username in order to access the application. Usernames will be created using the first letter of the patients name, their last name, and their birthday.

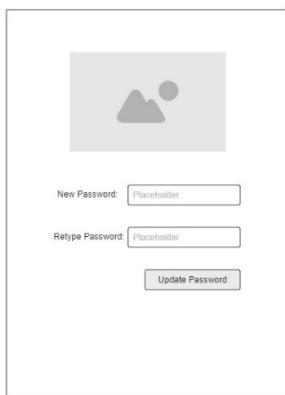
Choose the medication you would like refilled.
Medication 1
Medication 2

Medication Refills

First Name	Jane
Last Name	Doe
Birthday	Jan 1 2000

Username	jdoe01012000
----------	--------------

Usernames

*Passwords*

- **Passwords** – Each user will be provided with a temporary password. This will allow the user to log into the application. If it is their first time signing in, a screen will pop up where the user will have to set a password.

System Requirements

Functional Requirements

No.	Priority Weight	Description
REQ-1: Schedule Appointment	HIGH	This will allow patients to schedule their appointments online.
REQ-2: Update Information	HIGH	This will allow patients and staff to update patient information.
REQ-3: Cancel Appointment	HIGH	Allows patients the ability to cancel already scheduled appointments.
REQ-4: Reschedule Appointment	HIGH	Patients have the ability to reschedule already scheduled appointments.
REQ-5: Log In	HIGH	All users will have to have a username and password in able to log into the application.
REQ-6: Pre Appointment Information	MEDIUM	Available to patients to fill out before their appointment. This will allow the check in process be faster.
REQ-7: Staff Patient Search	HIGH	Staff will be able to search the system for a patient to view their information.
REQ-8: Staff Appointment Update	HIGH	Staff will have the ability to enter the information and test result from the appointment and save it in the system.

REQ-9: Adding New Patient	HIGH	Staff can add a new patient into the system, which will generate a username and password for the patient.
REQ-10: Prior Appointments	HIGH	Patients and staff will be able to review prior appointment notes from past appointments.
REQ-11: Staff View Patient Schedule	HIGH	The staff can view the schedule for the week or month and see all patient appointments.
REQ-12: Medication Refills	MEDIUM	Patients will have the ability to request medication refills online.
REQ-13: Medication Approvals	MEDIUM	Staff will have the ability to approve or deny the medication refills as well as review each request.
REQ-14: View My Health Page	MEDIUM	Staff and patients can view the My Health page to review the timeline of the patient's health.
REQ-15: Staff View Staff Schedule	LOW	The staff can view the schedule for other staff members and which doctors are working.

**Note that any requirements highlighted in gray will be completed as an update to the application. **

Nonfunctional Requirements

1. *Supportability* – support will be available during office hours, updates will be completed around midnight when traffic is lower.
2. *Functionality* – capability the system will be able to withstand the size and functions of the system.
3. *Reliability* – the system will be able to maintain performance during all times of the day.
4. *Usability* – the system will be easy to use and many training documentations will be provided.
5. *Performance* – the response time and speed will be high. With more information being added, updates may have to be included to improve performance.

User Interface Requirements:

The user should be able to run this on a Windows or iOS computer. The application will just be a desktop icon, which will open a Log in Screen. Each user will have their own username and password that will be provided by the administrator. This will be sent via Email upon downloading the application. Once logged in the user will have a menu of buttons at the top of the screen, where they can navigate through the application. Below is a sketch and description of all the screens that will be included along with where the buttons, text boxes, and menus will be located.

Log In Screen

This sketch shows a simple log-in interface. At the top is a logo labeled "LOGO". Below it are two text input fields: "Username" and "Password". Underneath these are two blue links: "Forgot Username" and "Forgot Password". At the bottom center is a red button labeled "Log in". Red arrows point from the text labels to their respective input fields and from the link labels to their respective links.

1. Log In Screen – This screen will be available for any user logging into the system. A logo of the health clinic or hospital will be included at the top. Each user will type in their username and password. Links will be provided if the user has forgotten either username or password.

Below the Log In Screen, there is a description of the Staff Page. It states that staff can view their schedule and which doctor's they are working with this week. There is also a calendar for patient appointments for the week below it. Clicking the link for the customer name will send the user to the *Patient Info* Page. There is also a menu of buttons at the top, which will allow the staff to navigate through the application.

Patient Search

This sketch shows a search interface. At the top is a logo labeled "image". Below it is a menu bar with three items: "calendar", "Patients", and "Prescription Approvals". To the right of the menu is a red arrow pointing up to the word "menu". Below the menu is a text input field labeled "Staff Name". Below the search area are three text input fields: "First Name", "Last Name", and "Date of Birth". To the right of these fields is a red arrow pointing up to the word "menu". Below these fields is a red button labeled "Search".

3. Patient Search – This page is where the staff can search for the patient. They can search by first name, last name, and date of birth. Below will be a list of all the possible options for that search. The staff can also click on NEW PATIENT to add a new patient to the system.

4. New Patient – This page allows the staff to add a new patient into the system. The staff enters the patient's name, Social Security Number, birthday, phone number, email, insurance information, and emergency contact. A username and temporary password are

Staff Page

This sketch shows a staff dashboard. At the top is a logo labeled "image". Below it is a menu bar with three items: "calendar", "Patients", and "Prescription Approvals". To the right of the menu is a red arrow pointing up to the word "menu". Below the menu is a section titled "Staff Schedule" with a grid of days (M-F) and times (9-5). Below that is a section titled "Patients Schedule" with a grid of days (M-F) and names (Sarah Jones, 11-11-15). A red arrow points from the "Sarah Jones" entry to the text "link to patient page".

New Patient

This sketch shows a form for adding a new patient. At the top is a logo labeled "Logo". Below it is a menu bar with three items: "calendar", "Patients", and "Prescription Approvals". To the right of the menu is a red arrow pointing up to the word "menu". The form includes fields for "First Name", "Middle Initial", "Last Name", "SSN", "Birthday" (with dropdown menus for Month, Day, Year), "Phone", "Email", "Insurance Provider", "Membership #", "Emergency Contact", "First Name", "Last Name", "Phone", "Email", "Username" (with a red arrow pointing down to a note "auto populated"), and "Temporary Password". A red arrow points from the "Add Patient" button to the "Username" field.

auto populated. The username will normally include the first initial of the patient's first name, their last name, and birthday. Once all the information is set up, the staff member can hit Add Patient. This will send an email to patient with their username and password if they wish to log in and fill out the rest of the paperwork on the *Profile Page*.

5. **Profile Page** – This screen will be first thing patients can view when they enter the system. This is known as the Profile Page. It includes the patient's name, contact information, emergency contact information, home address, and user authentication. The patient has an option to upload a photo of themselves for their profile. They also can update any of their information and save the update by hitting the update button at the bottom of the screen. User Authentication would allow the patient to update their username or change their password.

At the top of the screen are 4 options the user can do. This includes scheduling an appointment, refilling medication, checking their health, and contacting the office. Each button will direct the patient to a different screen.

The Appointment Page interface includes:

- A header section with a placeholder "Patient Name" and a "Schedule New Appointment" button.
- An "Upcoming Appointments" section showing a table with columns for Date/Time, Doctor, and Location. It includes "Pre-Apppt" and "cancel/resch." buttons.
- An "Previous Appointments" section showing a table with columns for Date/Time, Doctor, and Location. It includes a "details" button and a note: "more listed if available".

Appointment Page

6. **Appointment Page** – When the user clicks on the Schedule Appointment button at the top of the screen, it will direct them to the Appointments page. This will allow the user to create a new appointment, view any upcoming appointments, reschedule or cancel the upcoming appointments, complete the pre-appointment questionnaire, or view any previous appointments.

The Profile Page interface includes:

- A header section with a placeholder "Patient Name" and a "Schedule Apt.", "refill meds", "my health", and "contact" menu.
- A "Personal Info" section with fields for First, M.I., Last name; Sex (radio buttons M/V); Birthday (dropdown year); and Address (Street, City, State, Zip).
- An "Emergency Contact" section with fields for First, Last name, Phone, and Email.
- A "User Authentication" section with fields for Email, Username, and Password (with "change password" checkbox and "update" button).

Profile Page

menu of buttons

Patient Name

Schedule Appointment

Provider Select ✓ drop down menu

Network Select ✓

Location Select ✓

Available Times

Datetime	Doctor	Location

button → Schedule

button → Next

Schedule New Appointment

This will be a menu button option, where the user can select any of the three options: Phone, Zoom, or In Person appointments. Finally, the user will type in the text area, the reason they are wanting the appointment. The user will also have the option to go back if they decide they don't want that appointment, or continue making the appointment. If the user clicks back it will direct them back to the Appointments Page.

7. Schedule New Appointment – When the user clicks on the Schedule New Appointment button, the user will be able to choose a provider, network, and location using the available options in the drop down menu. To the right is a calendar, highlighting today's date. The user can pick any future date and view any available appointments at the location they choose. When they find an appointment that works for them, they can click schedule, which will select that time slot. The user will then click next to continue the sign up process.

8. Schedule New Appointment

Page 2 – After the user clicks NEXT, they will come to the screen where they can choose the kind of appointment they need. Button options will be available based on the type of appointments that location has available. In this instance Preventive is an option. Once that is selected they can choose the appointment type. This

menu of buttons

Patient Name

What kind of visit do you need?

Preventive

Select Appointment Type

Phone Zoom In Person horizontal menu option

what is the reason for appointment?

text area

back next

Schedule New Appointment 2

9. New Appointment Page 3 – This page is where the user will review and confirm their appointment. They have the options to go back to the previous screen or confirm the appointment at the top of the screen and the bottom. The user will review their contact information, which will be auto populated from their profile. If the user finds this information is incorrect, they can manually update it and it will also save on their profile as well. Below their contact information is the appointment information. This will include the patient's name, the visit type, the location, the date and time of the appointment, the provider's information, and then the details they provided on the previous screen. Once confirmed, the user can hit confirm appointment. They will receive an email confirming the appointment along with the information about the appointment including all the information provided on this page.

button menu

Profile refill medi my health contact

Patient Name

buttons

Review the following:

contact info: Email John.Doe@gmail.com
*text boxes auto populated

cell phone 789-4321 home phone 123-4567

Patient Name

Appointment Date

Time

Provider Name

Details

information populated

map

back Confirm Appt.

Schedule New Appointment 3

Pre Appointment Questionnaire

10. Pre-Appointment Page – On the *Appointment Page*, there is a button for the Pre-Appointment questionnaire that will need to be completed before the appointment. This will include the insurance information, any allergies or medications that the patient might have or be using. Filling this out will save time for the patient and staff, so it will not need to be completed while waiting in the waiting room. Once all the information is confirmed and filled out, the user can submit the questionnaire. If this patient has already had an appointment then this information will be auto filled in, from their previous visit. They can update it if they wish. If this is the patient's first time, they will need to fill this out. It will be saved and will auto populate for their next appointment.

Cancel/Reschedule Page

Prior Appointment Details

11. Cancel/Reschedule Page

– Also on the *Appointment Page* is a button to cancel or reschedule an appointment. When the user clicks this button, they are redirected to this page. At the top the user will have an option to cancel their appointment. They will need to provide a reason for their cancellation, then click cancel. This will send a notification to the office staff that an appointment has been cancelled and it will delete it from the calendar. If the patient wishes to reschedule, they can choose a provider, network,

location, and day on the calendar. They can view available times and select the one that works for them. Once all the information is filled out, they will be redirected to the review page to confirm their rescheduled appointment ****Schedule New Appointment 3****. With cancelling an appointment, depending on the provider's policies, there may be restrictions to how early a patient can cancel the appointment.

12. Prior Appointment Details – This page is found by clicking the Details button on the Appointment Page next to the previous appointments. When the user clicks on details they are sent to a page that will show the date and time of the appointment, the provider and all the health information collected from that appointment. This will include test results, weight, height, any doctor's notes, and the reason for the visit, and any medications prescribed by the doctor at that time.

Staff Calendar View

Info screen if they would like to view that. Below is a section for prior appointments, where the staff can view the *Prior Appointment Details* page. There is also Next Appointments, which has the patient's next appointments listed. When it is time for their appointment, the staff will hit the View button

then a button to the Patient Info screen if they would like to view that. Below is a section for prior appointments, where the staff can view the *Prior Appointment Details* page. There is also Next Appointments, which has the patient's next appointments listed. When it is time for their appointment, the staff will hit the View button

to take them to the *Staff Appointment Update* page. There is also a button at the bottom to view the patient's health page, which is known as *My Health Staff View*.

Staff Appointment Update

13. Staff Calendar View – This page includes a staff view or a patient view for appointments that month. This is a toggle view, so that patients and staff do not confuse the two. The patient view will provide links to the *Patient Info* page.

14. Patient Info

Page – This page can be located from the Staff Calendar View or the Staff Page. When a patient comes in for an appointment, the staff member can click on the link to this page. This will show the patient's name, date of birth, insurance information,

Patient Info Page

then a button to the Patient Info screen if they would like to view that. Below is a section for prior appointments, where the staff can view the *Prior Appointment Details* page. There is also Next Appointments, which has the patient's next appointments listed. When it is time for their appointment, the staff will hit the View button

to take them to the *Staff Appointment Update* page. There is also a button at the bottom to view the patient's health page, which is known as *My Health Staff View*.

15. Staff Appointment Update – This page is where the staff can update the patient's information from the appointment. This would include, their weight, height, BMI, blood pressure, the details from the appointment like the reason behind it, any doctor notes, and add any medications they were prescribed for the appointment. Once the information has been completed, the staff would just click UPDATE. This will update the *My Health* page in the Patient view, so they can review their results of their appointment.

16. Medication Refill – When a user wants a refill on one of their prescriptions, they can click Medication Refill option at the top of the page. This will direct them to this page. The user will have options on medications they can refill. They can click on the medications, then choose the pharmacy they would like the medication sent to. If the user has had medication sent there before, this will be listed. They also have an option to search for another pharmacy. Once a pharmacy is selected, the user can submit the request. This will send a notification to the office staff to approve the refill. The user will receive an email that their request has been sent and that they will be notified within 24 hours if it was approved. Once approved, the user will receive an email with a time the prescription will be ready for pick up. The patient will also be able to see how many more refills they have available underneath the button of medication option

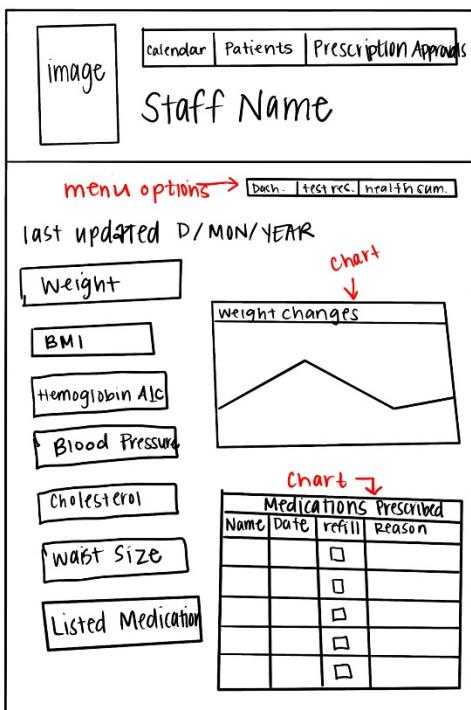
image	calendar Patients Prescription Approvals			
Staff Name				
Prescription Approvals				
Name	Medication	Prior Refill	Approve	Deny
Patient Name ↑ link to patient info	Name of medication	Date/time	<input checked="" type="checkbox"/> button	<input type="checkbox"/> checkbox boxes
submit				

Medication Approvals

Medication Refill Page

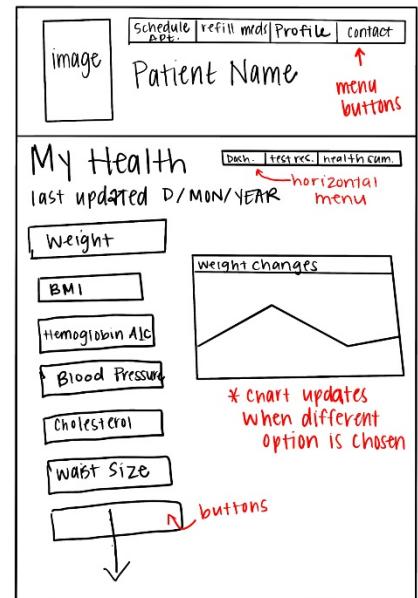
17. Medication Approvals – When the patient submits a request for medication refills, the request gets added to a queue on the Prescription Approvals tab for the staff. The staff will have to review the request, by seeing when it was previously refilled, check to see who prescribed it by clicking the link and reviewing the Patient Info Screen, then check the box to either approve or deny the request. Once they are done, they can hit submit at the bottom. If approved, this will send an email to the pharmacy and the user letting them know it has been confirmed. If denied, the patient will receive an email to call the office to discuss the refill request.

18. My Health – This page is available by clicking My Health on the menu buttons. This page will show the user their test results, health summary, and their health results based of all their appointments. When on the dashboard, the user can see their prior weight, BMI, Cholesterol, and all the variables that are taken during an appointment. When each one of those are clicked, a line graph will show up on the right, which will allow the user to see all their results. The test results option, will allow the user to view any blood tests that they recently have taken, and their health summary will show their progress over time.



My Health Staff View

that zip code and city. Below will be a detailed list with the locations name, phone, address, and hours.

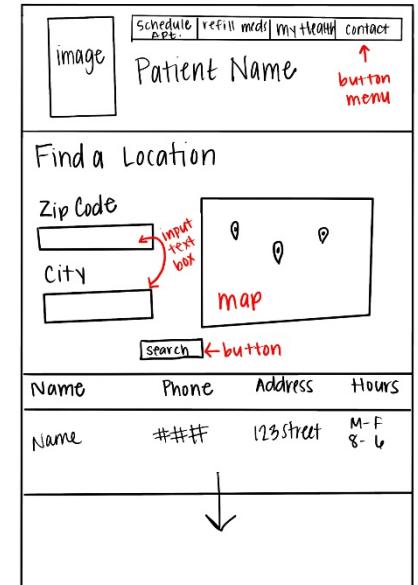


My Health Page

19. My Health Staff

View - This page can be found by the staff by viewing the *Patient Info Page* and clicking the button for Patient Health at the bottom. There is a toggled list to the left side, where the staff can view a chart seeing the progress and prior results of the patient. They can also view the patient's listed medications, and any medications that were prescribed during that appointment and if it can be refilled or not. This is very similar to the *My Health* page, except for the medications prescribed and listed medication options.

20. Contact – This page is for contacting and finding a close location. The user can enter their zip code and city, then click search. The map on the right will update to show all the locations available in



Contact Page

Functional Requirement Specification

Stakeholders: Below is a list of stakeholders that would be interested in this application.

- Nurses
- Doctors
- Front Desk Staff
- Patients
- Customer Support
- Project Manager
- VPN Connection Company
- Two Factor Authentication Company

Actors and Goals: Below is a chart which shows the roles of people/devices that will interact directly with the system.

Actors	Goals
VPN Connection Company	<ul style="list-style-type: none"> ○ Allow the users to sign into VPN ○ Only allow users signed into VPN to access the application
Two Factor Authentication Company	<ul style="list-style-type: none"> ○ Allow the users to have an extra set of authentication ○ Only allow users access to application if it passes Two Factor Authentication.
Patients	<ul style="list-style-type: none"> ○ Update personal information ○ Log Into System ○ Log out of system ○ View personal information ○ Schedule a New Appointment ○ View Prior Appointment Details ○ Complete Pre Appointment Paperwork ○ Cancel Appointments ○ Reschedule Appointments ○ View health records/progress on My Health page ○ Request prescription refills ○ Search for offices in the area ○ View contact information for offices near them. ○ Create New Password
Nurses	<ul style="list-style-type: none"> ○ Update patient information ○ Log into system ○ Log out of system ○ View patient information ○ View calendar of patient appointments ○ View calendar of staff schedule ○ View patient's health records/progress ○ View patient's prior appointment details ○ Update patient appointment details ○ Approve/deny prescription refills ○ Search for a patient

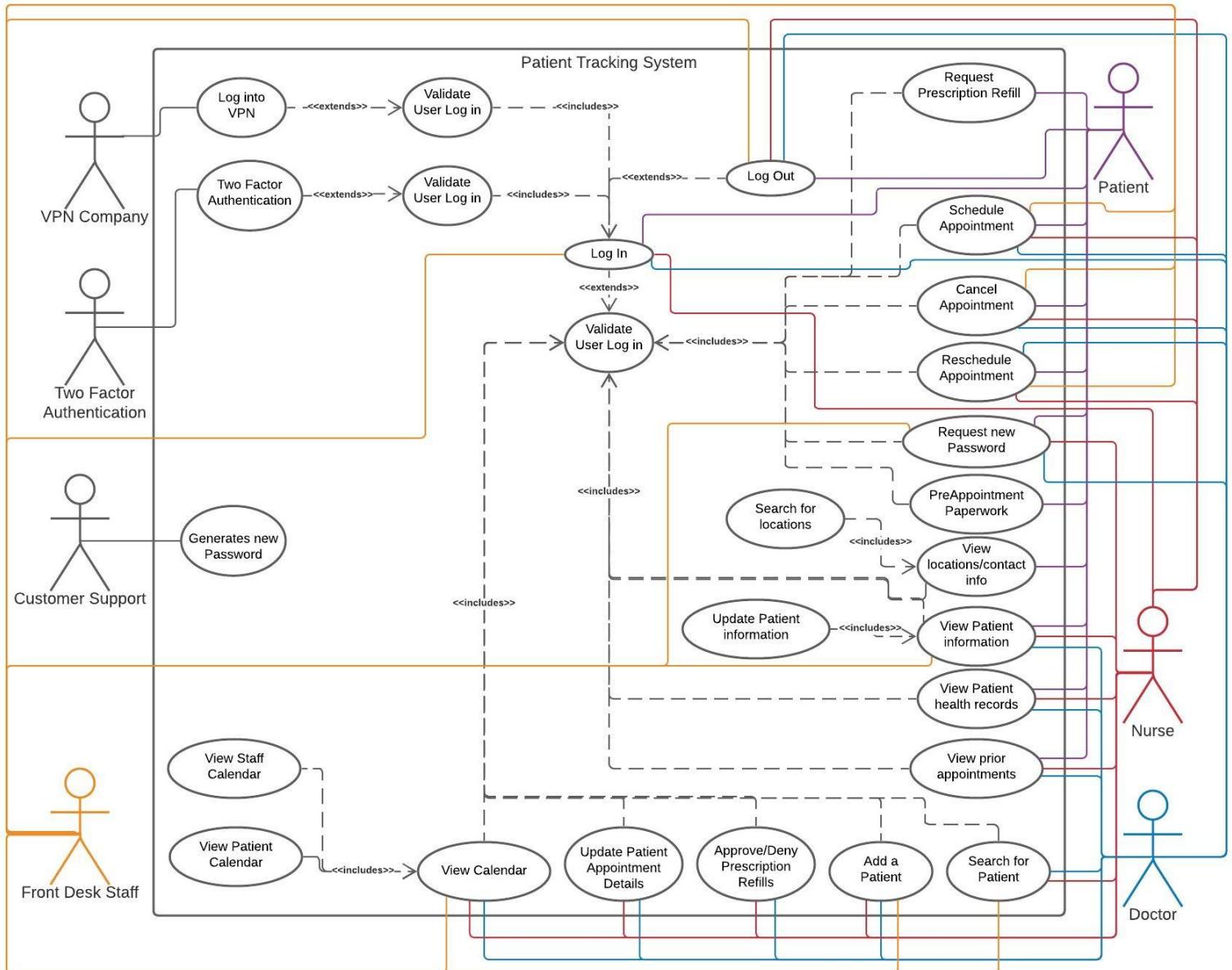
	<ul style="list-style-type: none"> <input type="radio"/> Add new patient <input type="radio"/> Schedule an appointment <input type="radio"/> Cancel an appointment <input type="radio"/> Reschedule an appointment <input type="radio"/> Create New Password
Doctors	<ul style="list-style-type: none"> <input type="radio"/> View patient information <input type="radio"/> Update Patient Information <input type="radio"/> Log into system <input type="radio"/> Log out of system <input type="radio"/> View calendar of patient appointments <input type="radio"/> View calendar of staff schedule <input type="radio"/> View patient's health records/progress <input type="radio"/> View patient's prior appointment details <input type="radio"/> Update patient appointment details <input type="radio"/> Approve/deny prescription refills <input type="radio"/> Search for a patient <input type="radio"/> Add new patient <input type="radio"/> Schedule an appointment <input type="radio"/> Cancel an appointment <input type="radio"/> Reschedule an appointment <input type="radio"/> Create New Password
Front Desk Staff	<ul style="list-style-type: none"> <input type="radio"/> Log into System <input type="radio"/> Log out of system <input type="radio"/> View calendar of patient appointments <input type="radio"/> View calendar of staff schedule <input type="radio"/> View patient contact information <input type="radio"/> Update patient information <input type="radio"/> Search for a patient <input type="radio"/> Schedule an appointment <input type="radio"/> Cancel an appointment <input type="radio"/> Reschedule an appointment <input type="radio"/> Create New Password
Customer Support	<ul style="list-style-type: none"> <input type="radio"/> Creates new password

Use Cases:

System Requirement	Description
REQ-1: Log In	All users will have to have a username and password in able to log into the application.
REQ-2: Update Information	This will allow patients and staff to update patient information.
REQ-3: Cancel Appointment	Allows patients the ability to cancel already scheduled appointments.

REQ-4: Reschedule Appointment	Patients have the ability to reschedule already scheduled appointments.
REQ-5: Schedule Appointment	This will allow patients to schedule their appointments online.
REQ-6: Pre Appointment Information	Available to patients to fill out before their appointment. This will allow the check in process be faster.
REQ-7: Staff Patient Search	Staff will be able to search the system for a patient to view their information.
REQ-8: Staff Appointment Update	Staff will have the ability to enter the information and test result from the appointment and save it in the system.
REQ-9: Adding New Patient	Staff can add a new patient into the system, which will generate a username and password for the patient.
REQ-10: Prior Appointments	Patients and staff will be able to review prior appointment notes from past appointments.
REQ-11: Staff View Patient Schedule	The staff can view the schedule for the week or month and see all patient appointments.
REQ-12: Log Out	This allows the user to log out of the application.
REQ-13: Medication Refills	Patients will have the ability to request medication refills online.
REQ-14: Medication Approvals	Staff will have the ability to approve or deny the medication refills as well as review each request.
REQ-15: Staff View Staff Schedule	The staff can view the schedule for other staff members and which doctors are working.
REQ-16: View My Health Page	Staff and patients can view the My Health page to review the timeline of the patient's health.
REQ-17: Request new password	This allows the user to request a new password if they forgot theirs.
REQ-18: Create new password	This allows the support team to create a new password if the user requests one.

**Note that any requirements highlighted in gray will be completed as an update to the application. **

Use Case Diagram

Traceability Matrix

Req-ID	Requirement	Priority Weight	Test Case ID	Test Case Description	Status
1	Log In	HIGH	TC01	Login with invalid username and password	Passed
			TC02	Login without inputting username and password	Passed
			TC03	Login with valid credentials	Passed
			TC04	Login as patient	Passed
			TC05	Login as Front Desk Staff	Passed
			TC06	Login as Nurse	Passed
			TC07	Login as Doctor	Passed
2	Update Information	HIGH	TC08	Patient updates information	Passed
			TC09	Front Desk updates information	Passed
			TC10	Nurse updates information	Passed
			TC11	Doctor updates information	Passed
3	Cancel Appointment	HIGH	TC12	Patient cancels appointment	Passed
			TC13	Front Desk cancels appointment	Passed
			TC14	Nurse cancels appointment	Passed
			TC15	Doctor cancels appointment	Passed
4	Reschedule Appointment	HIGH	TC16	Patient reschedules appointment	Passed
			TC17	Front desk reschedules appointment	Passed
			TC18	Nurse reschedules appointment	Passed
			TC19	Doctor reschedules appointment	Passed
5	Schedule Appointment	HIGH	TC20	Patient schedules appointment	Passed
			TC21	Front desk schedules appointment	Passed
			TC22	Nurse schedules appointment	Passed
			TC23	Doctor schedules appointment	Passed
6	Pre-Appointment Information	MED	TC24	Patient completes pre-appointment information	Passed
7	Staff Patient Search	HIGH	TC25	Front desk searches for patient	Passed
			TC26	Nurse searches for patient	Passed
			TC27	Doctor searches for patient	Passed
8	Staff Appointment Update	HIGH	TC28	Nurse updates patient appointment page	Passed
			TC29	Doctor updates patient appointment page	Passed
9	Adding New Patient	HIGH	TC30	Front desk adds a new patient	Passed
			TC31	Nurse adds a new patient	Passed
			TC32	Doctor adds a new patient	Passed

10	Prior Appointments	HIGH	TC33	Patient views prior appointments	Passed
			TC34	Nurse views prior appointments	Passed
			TC35	Doctor views prior appointments	Passed
11	Staff View Patient Schedule	HIGH	TC36	Front desk views patient schedule	Passed
			TC37	Nurse views patient schedule	Passed
			TC38	Doctor views patient schedule	Passed
12	Log Out	HIGH	TC39	Click CLOSE on login menu	Passed
			TC40	Patient clicks Exit in application menu	Passed
			TC41	Patient clicks Log Out in application menu	Passed
			TC42	Front Desk clicks Exit in application menu	Passed
			TC43	Front Desk clicks Log Out in application menu	Passed
			TC44	Nurse clicks Exit in application menu	Passed
			TC45	Nurse clicks Log Out in application menu	Passed
			TC46	Doctor clicks Exit in application menu	Passed
			TC47	Doctor clicks Log Out in application menu	Passed
13	Request new password	MED	TC48	user clicks forgot password link	Future Update
14	Create new password	MED	TC49	Customer support generates temp password	Future Update
			TC50	Customer support sends email link to create new password	Future Update
15	Medication Refills	MED	TC51	Patient requests medication refills	Future Update
16	Medication Approvals	MED	TC52	Nurse approves medication refills	Future Update
			TC53	Nurse denies medication refills	Future Update
			TC54	Doctor approves medication refills	Future Update
			TC55	Doctor denies medication refills	Future Update
17	View My Health Page	MED	TC56	Patient views My Health Page	Future Update
			TC57	Nurse views My Health Page	Future Update
			TC58	Doctor views My Health Page	Future Update
18	Staff View Staff Schedule	LOW	TC59	Front desk view staff schedule	Future Update
			TC60	Nurse views staff schedule	Future Update
			TC61	Doctor views staff schedule	Future Update

Fully-Dressed Description

Req-ID 1 Login: This requirement allows the user to log into the application using a username and password. Depending on the username, the application can tell if the user is a patient, nurse, doctor, or front desk staff person. Each type of user has a different view, so logging into the system is a really important step. If the username begins with a letter, then the patient will be logged in as

a patient and have the patient view. The front desk staff have more restrictions in their view compared to the doctor or nurse. For staff users, depending on the first number of their username, that will tell the application which view the user should be logged into. If the username begins with 1, the user will be logged in as a doctor; if the username begins with 2, the user will be logged in as a nurse; if the username begins with 3, the user will be logged in as a front desk staff. The login screen will alert the user if they have the correct credentials to login. It will also notify them if nothing has been entered and if the credentials are incorrect. The user has an opportunity to hit the login button, or click enter after typing their password.

Req-ID 3 Cancel Appointment: If a patient would like to cancel their appointment, they would be able to do this on the Schedule Appointment page. They can view their Next Appointments and can click Cancel/Reschedule button. This action will direct the patient to the Cancel/Reschedule Page. At the top of the page, is where the user can cancel their appointment. All the patient has to do is explain why they are cancelling their appointment and then click cancel. Once they click cancel, it will remove the appointment from their view, and add it to the prior appointment with the status of Cancelled. The patient will also receive an email that their appointment has been cancelled. This option will be available depending on the office policy. If a patient can only cancel up to 24 hours before the appointment, then the option to cancel will not be available for the patient.

If a staff member would need to cancel an appointment for a patient, they can click on the calendar view and locate the appointment. When they click view, to the right they will be able to edit the appointment and click cancel. This will remove it from their calendar view and add it to the patient's prior appointments with the status of Cancelled.

Req-ID 4: Reschedule Appointment: If a patient would like to reschedule their appointment, they would be able to do this on the Schedule Appointment page. They can view their Next Appointments and can click Cancel/Reschedule button. This will direct the patient to the Cancel/Reschedule Page. At the bottom of this page, the patient can reschedule their appointment, by choosing the provider, network, and location in the drop down menus. To the right, they can choose a new date and view all available appointments. Once they decide they can click Schedule. This will direct them to the review page for the appointment. They will have to verify that their contact information is correct, and can make updates if needed. Any updates made will update their profile page. They will also need to review the appointment details including the provider name, date and time, they type of visit, location, and then the details they provided on the previous page. Once confirmed, they can hit the CONFIRM APPOINTMENT button. Once the appointment is confirmed, the appointment will update the staff calendar and an email will be sent to the patient confirming the appointment. It will also update the Next Appointments section. This option will be available depending on the office policy. If a patient can only cancel up to 24 hours before the appointment, then the option to cancel will not be available for the patient.

If a staff member would need to reschedule an appointment for a patient, they will click on the calendar and location the patient's current appointment. From there, they can click view and it will show up to the right side of the screen. The staff member can click the edit button and choose a different date, time, and provider for the patient. Once they click save it will update the staff calendar and the patient's Next Appointments section with updated information.

Req-ID 5 Schedule Appointment: For the patient to schedule an appointment, they will need to click the New Appointment button on the Schedule Appointment page. This will direct the user to a new page. They will have to use the dropdown menu to select an available provider, another dropdown for network, and another dropdown for location. To the right will be a calendar where the user can select a date they are interested in having the appointment. When a date is clicked, all the available times for that day will populate at the bottom of the screen. Once the patient finds the appointment they are interested in, they will click schedule and it will direct them to the next set of instructions. The patient will have to choose the kind of appointment, the type of appointment, and then enter the reason for the appointment. Once they are satisfied they can click NEXT. If they decide not to proceed, they can click BACK and it will direct them back to the New Appointment page. When the patient clicks NEXT, it will direct them to the Review page. They will have to verify that their contact information is correct, and can make updates if needed. Any updates made will update their profile page. They will also need to review the appointment details including the provider name, date and time, they type of visit, location, and then the details they provided on the previous page. Once confirmed, they can hit the CONFIRM APPOINTMENT button. If they decide not to proceed, they can click the BACK button and it will redirect them back to the previous page. Once the appointment is confirmed, the appointment will be added to the staff calendar and an email will be sent to the patient confirming the appointment. It will also populate on the Next Appointments section, where the patient has access to the Pre-Appointment Information and can either cancel the appointment or reschedule it.

If the staff would like to create an appointment for a patient, they can do so by clicking on the calendar option on their menu. When the staff member is on the calendar page, they can view the calendar and add appointments for the patients by clicking ADD APPOINTMENT on the day of the week the patient would like their appointment. They will enter the patient's name, provider, type, start time, and reason for the appointment. Once it is added to the calendar, the patient will receive an email confirming the appointment. It will also populate on the Next Appointments section, where the patient has access to the Pre-Appointment Information and can either cancel the appointment or reschedule it.

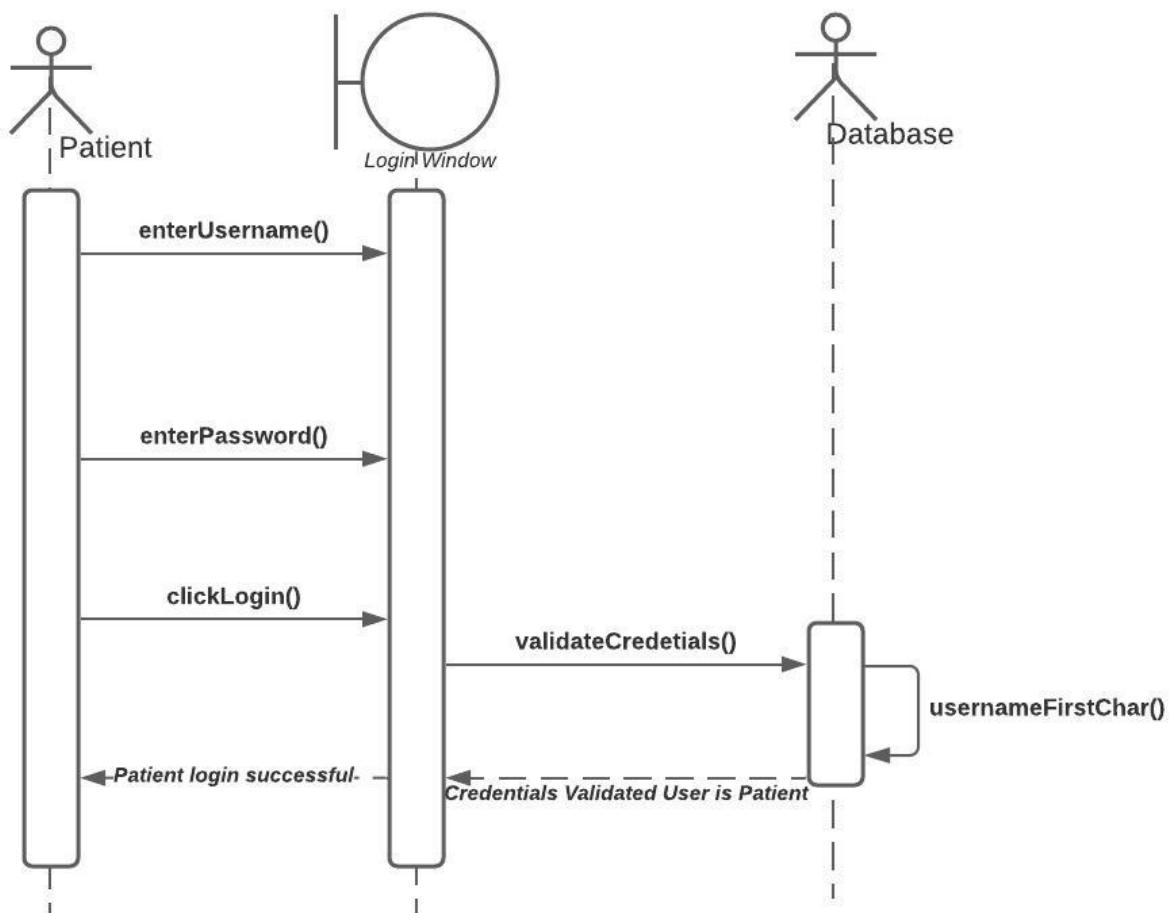
Req-ID 7 Staff Patient Search: All staff members have an opportunity to search for a patient. Each staff member can enter the first name, last name, and use the drop down menus for the date of birth of the patient and hit the search button. If any of the patients have that name and date of birth they will populate at the bottom of the screen in a table. This table will provide the patient's name, date of birth, address, phone number, and email. There will also be a button to take the staff member to the Patient Profile page where they can view their contact information. This page also has a button called Add New Patient. This button will direct the staff to a new page where they can create a new patient for the system.

Req-ID 8 Staff Appointment Update: This page is available for only nurses and doctors. For the patient's appointment, the nurse or doctor will have to click on the patient's appointment on the calendar. This will direct them to the Patient Information Page. In the next appointments section of that page, the nurse and doctor will click view. Once they land on this page, this is where the nurse will input all the information taken during the screening, before the doctor comes into the room. This includes their weight, height, BMI, Blood Pressure, etc. There will be a section that pulls from the patient where it specifies the reason for the appointment. There will be a section for

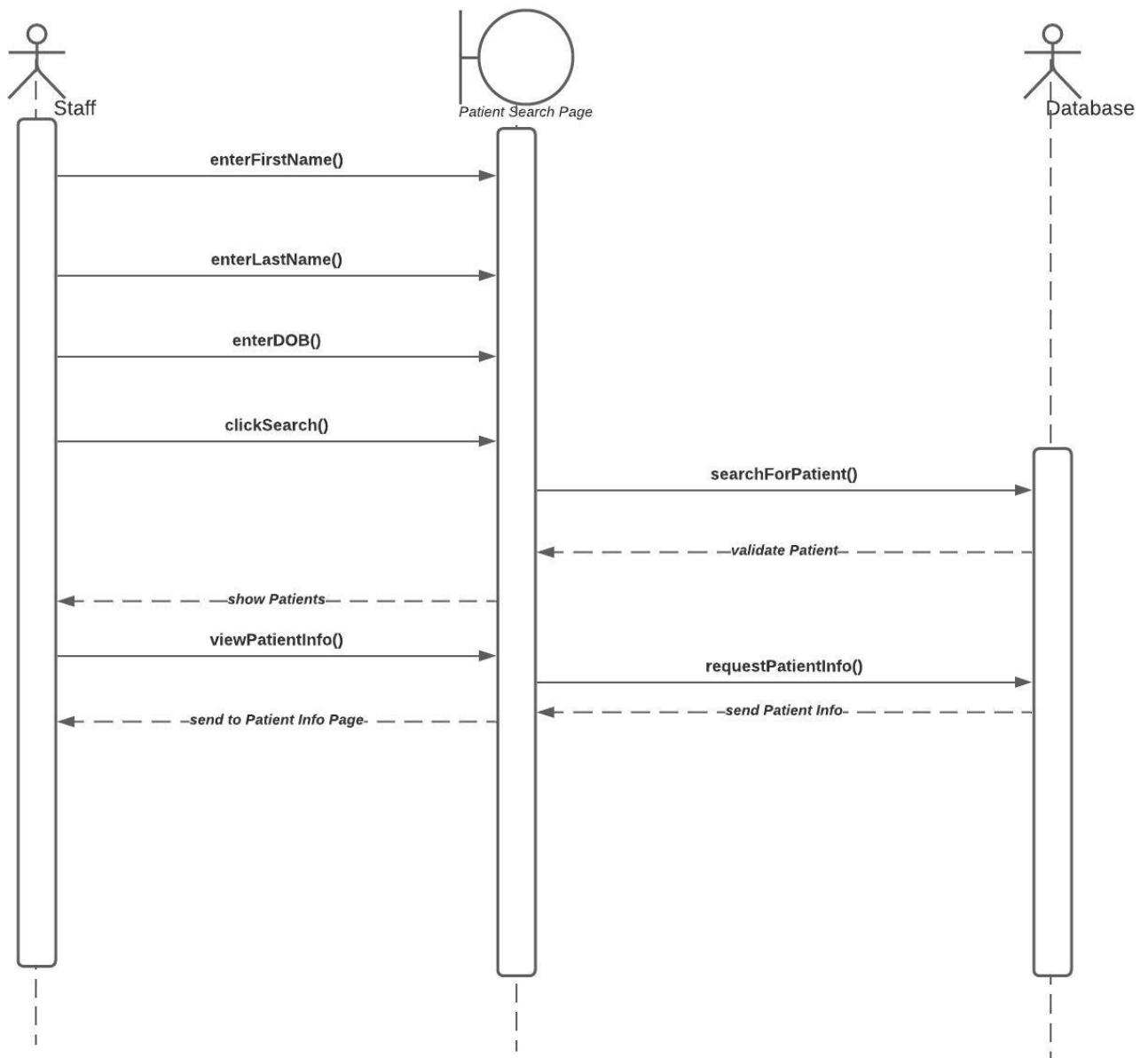
nurses to take and leave any additional notes for the doctor. When the nurse gets ready to leave, they will click SAVE. This will save the information on the page for the doctor to come and edit. Once the doctor logs in, the information from the screening should already be filled out. The doctor can then add any additional notes in the doctor section. Below that will be a table where the doctor can add prescriptions. Only the doctor has the ability to add prescriptions to this page. Once the appointment has concluded, the doctor will click Update and it will save that information on the page. That will close the appointment and move it to the Prior Appointments section.

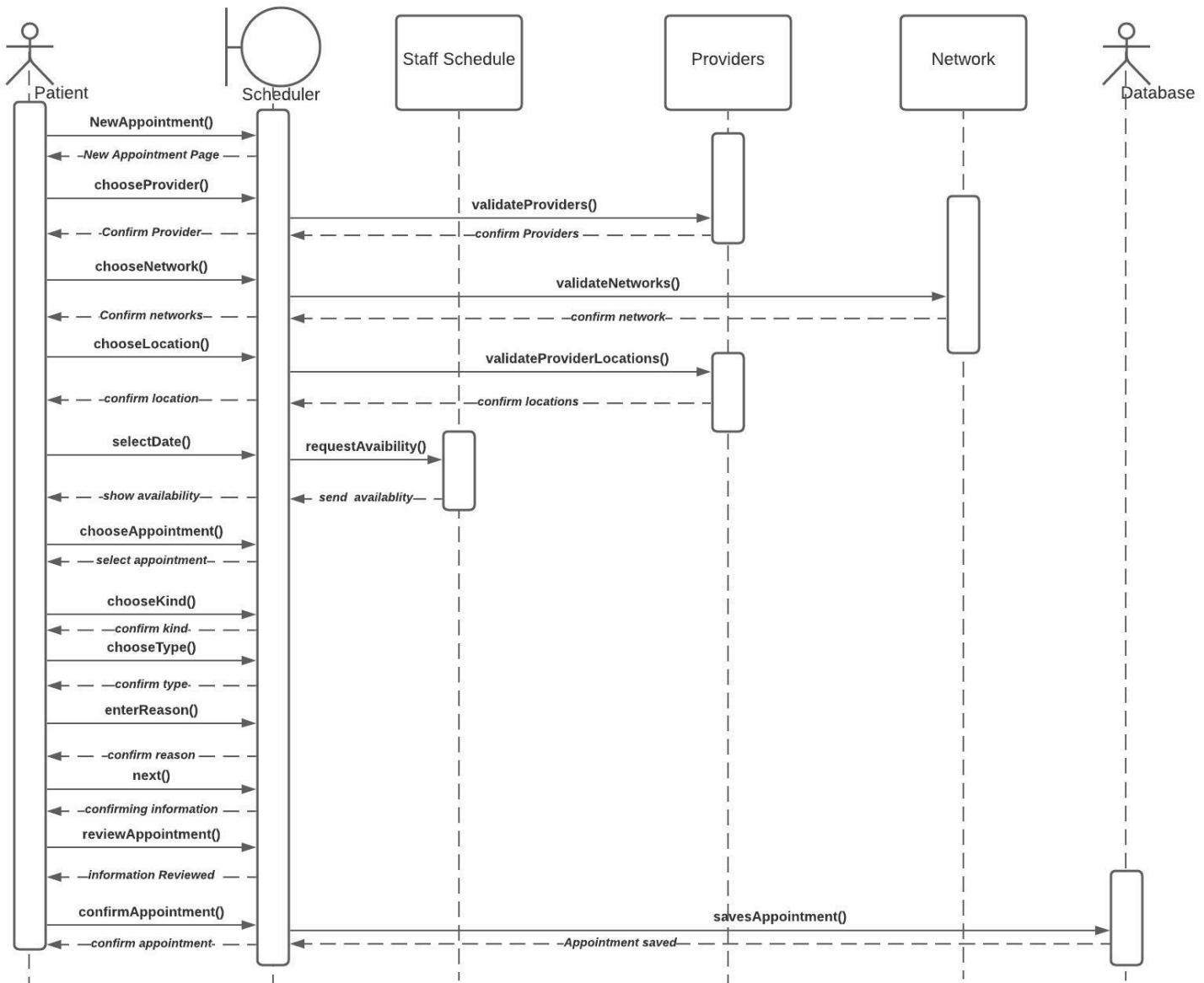
System Sequence Diagrams

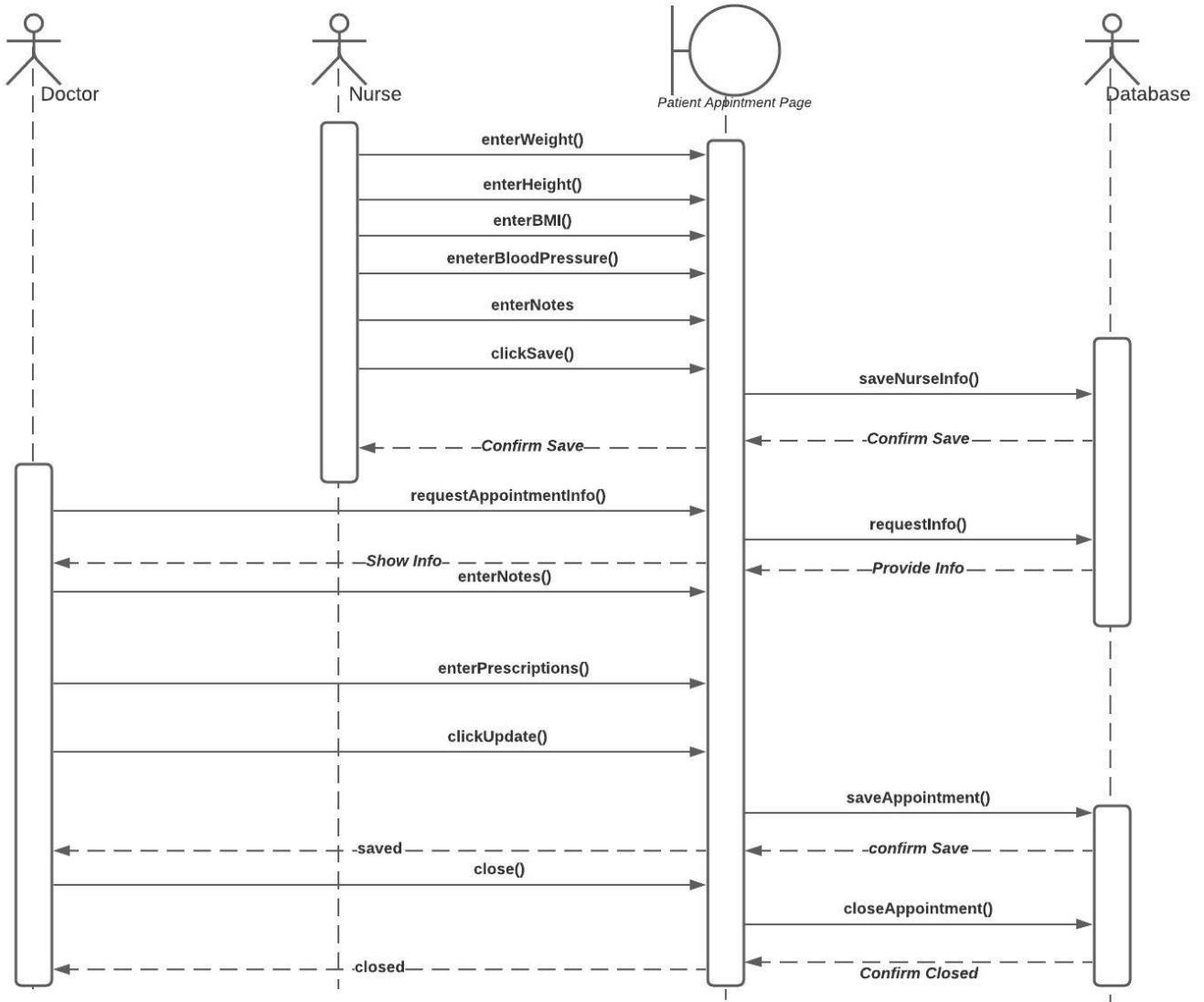
Login as Patient



Patient Search as any Staff Member



Schedule Appointment as Patient

Staff Appointment Update

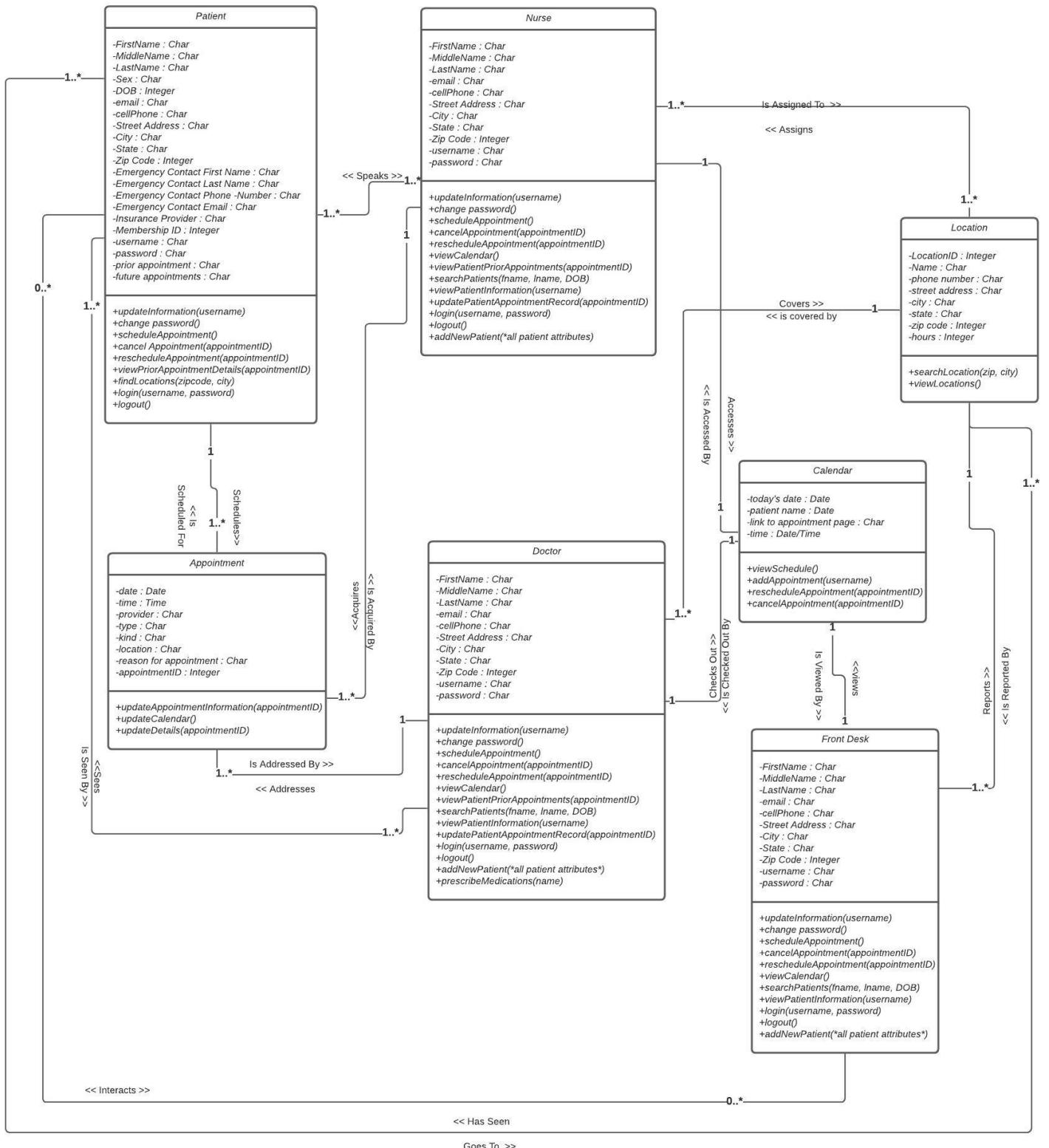
User Effort Estimation

Below is a chart that will select some of the major use cases and the number of clicks needed to complete each scenario.

Use Case	Least Amount of Clicks
Log In Screen	3
Search for Patient	4/5
Add a New Patient	16
Schedule New Appointment (Patient)	11
Schedule New Appointment (Staff)	7
Cancel an Appointment (Patient)	2
Reschedule and Appointment (Patient)	6
Cancel an Appointment (Staff)	3
Reschedule an Appointment (Staff)	5
Staff Appointment Update (Nurses)	8
Staff Appointment Update (Doctors)	6
Medication Refills – pharmacy was previously used	3
Medication Refills – new pharmacy	6
Prescription Approvals	3
Find a Location	3

**Note that any requirements highlighted in gray will be completed as an update to the application. **

Domain Analysis



Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
1	5	X											
2	5		X										
3	5			X									
4	5				X								
5	5					X							
6	3						X						
7	5							X					
8	5								X				
9	5									X			
10	5										X		
11	5											X	
12	5												X
Max PW	5	5	5	5	5	3	5	5	5	5	5	5	5
Total PW	5	5	5	5	5	3	5	5	5	5	5	5	5

The traceability matrix is based off of the 12 requirements and 12 use cases that were implemented during this process. The breakdown of what each requirement is can be viewed in the traceability matrix in the use case section of the report. The requirements and use cases go hand in hand with the domain analysis above to break down the methods that are being utilized for every class.

System Operation Contracts

Login Use Case

Contract CO1: enterUsername	
Operation	enterUsername(username: UserName)
Cross Reference	Use Case: Login
Preconditions	User is trying to log into system
Post conditions	<ul style="list-style-type: none"> • Username is associated with patient account (<i>association formed</i>) • Username is associated with front desk account (<i>association formed</i>) • Username is associated with nurse account (<i>association formed</i>) • Username is associated with doctor account (<i>association formed</i>)

Contract CO2: enterPassword	
Operation	enterPassword(password: PassWord)
Cross Reference	Use Case: Login
Preconditions	User is trying to log into system
Post conditions	<ul style="list-style-type: none"> • Password is associated with user's username (<i>association formed</i>)

Contract CO3: clickLogin	
Operation	clickLogin(username: UserName, password: Password)
Cross Reference	Use Case: Login
Preconditions	User is trying to log into system
Post conditions	<ul style="list-style-type: none"> • Username instance un is created (<i>instance created</i>) • Username is associated with user account (<i>association formed</i>) • Password instance pd is created (<i>instance created</i>) • Password is associated with user's username (<i>association formed</i>)

Contract CO4: validateCredentials	
Operation	validateCredentials(username: UserName, password: PassWord)
Cross Reference	Use Case: Login
Preconditions	User is trying to log into system
Post conditions	<ul style="list-style-type: none"> • Username and password are associated with account in database (<i>association formed</i>)

Contract CO5: usernameFirstChar	
Operation	usernameFirstChar(username: UserName)
Cross Reference	Use Case: Login
Preconditions	User is trying to log into system
Post conditions	<ul style="list-style-type: none"> • Username is associated with patient account (<i>association formed</i>) • Username is associated with front desk account (<i>association formed</i>) • Username is associated with nurse account (<i>association formed</i>) • Username is associated with doctor account (<i>association formed</i>)

Schedule Appointment Use Case

Contract CO1: newAppointment	
Operation	newAppointment()
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • newAppointment instance na is created (<i>instance created</i>) • Attributes of na were initialized

Contract CO2: chooseProvider	
Operation	chooseProvider(provider : pvdr)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • Provider became provider chosen (<i>attribute modification</i>)

Contract CO3: validateProviders	
Operation	validateProviders(provider : pvdr)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • provider is associated with doctor account (<i>association formed</i>)

Contract CO4: chooseNetwork	
Operation	chooseNetwork (network : Net)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • network became network chosen (<i>attribute modification</i>)

Contract CO5: validateNetworks	
Operation	validateNetworks (network : Net)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • network is associated with doctor account (<i>association formed</i>)

Contract CO6: chooseLocation	
Operation	chooseLocation (provider: prvdr, network : Net, location : loc)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • location became location chosen (<i>attribute modification</i>)

Contract CO7: validateProviderLocations	
Operation	validateProviderLocations(provider: prvdr, network : Net, location : loc)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • location is associated with location (<i>association formed</i>)

Contract CO8: selectDate	
Operation	selectDate (provider: prvdr, network : Net, location : loc)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • selectDate is associated with calendar (<i>association formed</i>) • selectDate.date/time became dat and time chosen (<i>attribute modification</i>)

Contract CO9: requestAvailability	
Operation	requestAvailability (provider: prvdr, network : Net, location : loc, date : dt)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> • requestAvailability is associated with calendar (<i>association formed</i>)

Contract CO10: chooseKind	
Operation	chooseKind (kind : knd)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> kind became kind chosen (<i>attribute modification</i>)

Contract CO11: chooseType	
Operation	chooseType (type : tp)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> Type became type chosen (<i>attribute modification</i>)

Contract CO12: enterReason	
Operation	enterReason (reason : rsn)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> reason became reason provided (<i>attribute modification</i>)

Contract CO13: next	
Operation	Next (kind: knd, type : tp, reason : rsn)
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> appointment instance apt was created (<i>instance created</i>) Attributes of apt were initialized

Contract CO14: reviewAppointment	
Operation	reviewAppointment
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> Apt was associated with appointment (<i>association formed</i>)

Contract CO15: confirmAppointment	
Operation	confirmAppointment
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> Apt was associated with appointment (<i>association formed</i>)

Contract CO16: savesAppointment	
Operation	savesAppointment
Cross Reference	Use Case: Schedule Appointment
Preconditions	User is trying to schedule new appointment
Post conditions	<ul style="list-style-type: none"> Apt was associated with appointment <i>(association formed)</i>

Staff Appointment Update

Contract CO1: enterWeight	
Operation	enterWeight(weight : Weight)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> Weight became weight entered <i>(attribute modification)</i>

Contract CO2: enterHeight	
Operation	enterHeight(height : Height)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> height became height entered <i>(attribute modification)</i>

Contract CO3: enterBMI	
Operation	enterBMI(bmi : BMI)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> bmi became BMI entered <i>(attribute modification)</i>

Contract CO4: enterBloodPressure	
Operation	enterBloodPressure(bloodPressure : bp)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> Blood Pressure became blood pressure entered <i>(attribute modification)</i>

Contract CO5: enterNotes	
Operation	enterNotes (notes : Notes)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Notes became notes entered (<i>attribute modification</i>)

Contract CO6: clickSave	
Operation	clickSave (weight : Weight, height : Height, bloodPressure : bp, notes : Notes)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record instance pr was created (<i>instance created</i>)

Contract CO7: saveNurseInfo	
Operation	saveNurseInfo (weight : Weight, height : Height, bloodPressure : bp, notes : Notes)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record instance pr was created (<i>instance created</i>)

Contract CO8: requestAppointmentInfo	
Operation	requestAppointmentInfo (weight : Weight, height : Height, bloodPressure : bp, notes : Notes)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record is associated with patient account (<i>association formed</i>)

Contract CO9: enterNotes	
Operation	enterNotes (notes : Notes)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Notes became notes entered (<i>attribute modification</i>)

Contract CO10: enterPrescriptions	
Operation	enterPrescriptions(name : Name, refillnum : Refills, dosage : dose, date : Date)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • prescription became prescription entered (<i>attribute modification</i>)

Contract CO11: clickUpdate	
Operation	clickUpdate (notes : Notes, prescriptions : ps)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record instance pr was created (<i>instance created</i>)

Contract CO12: saveAppointment	
Operation	saveAppointment (notes : Notes, prescriptions : ps)
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record instance pr was created (<i>instance created</i>)

Contract CO13: closeAppointment	
Operation	closeAppointment ()
Cross Reference	Use Case: Staff Appointment Update
Preconditions	Nurse and Doctor are updating patient appointment information
Post conditions	<ul style="list-style-type: none"> • Patient Record instance pr was created (<i>instance created</i>)

Patient Search

Contract CO1: enterFirstName	
Operation	enterFirstName(firstname : fname)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> • First Name became First Name entered (<i>attribute modification</i>)

Contract CO2: enterLastName	
Operation	enterLastName(lastName : lname)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> Last Name became Last Name entered (<i>attribute modification</i>)

Contract CO3: enterDOB	
Operation	enterDOB(month : m, day : dd, year : yyyy)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> Month became Month entered (<i>attribute modification</i>) Day became Day entered (<i>attribute modification</i>) Year became Year entered (<i>attribute modification</i>)

Contract CO4: clickSearch	
Operation	clickSearch(firstName : fname, lastName : lname, month : m, day : dd, year : yyyy)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> First Name is associated with patient account (<i>association formed</i>) Last Name is associated with patient account (<i>association formed</i>) Month is associated with patient account (<i>association formed</i>) Day is associated with front desk account (<i>association formed</i>) Year is associated with nurse account (<i>association formed</i>) Patient instance pat was created (<i>instance created</i>)

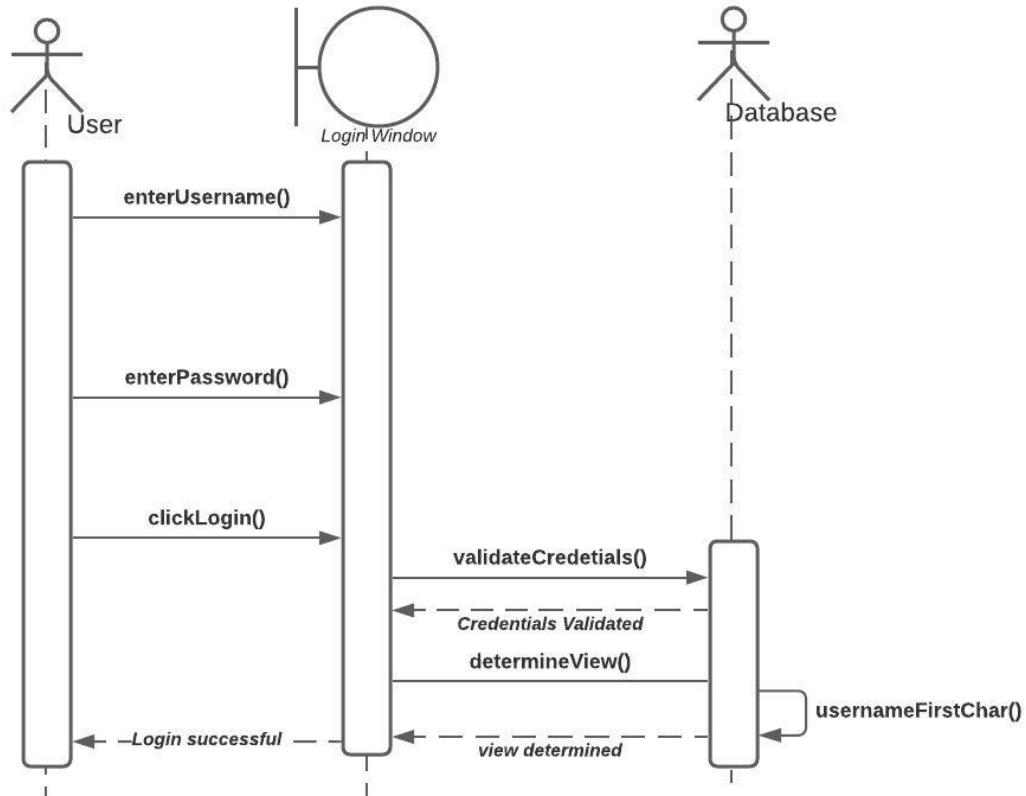
Contract CO5: searchForPatient	
Operation	searchForPatient (firstName : fname, lastName : lname, month : m, day : dd, year : yyyy)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> • First Name is associated with patient account (<i>association formed</i>) • Last Name is associated with patient account (<i>association formed</i>) • Month is associated with patient account (<i>association formed</i>) • Day is associated with front desk account (<i>association formed</i>) • Year is associated with nurse account (<i>association formed</i>)

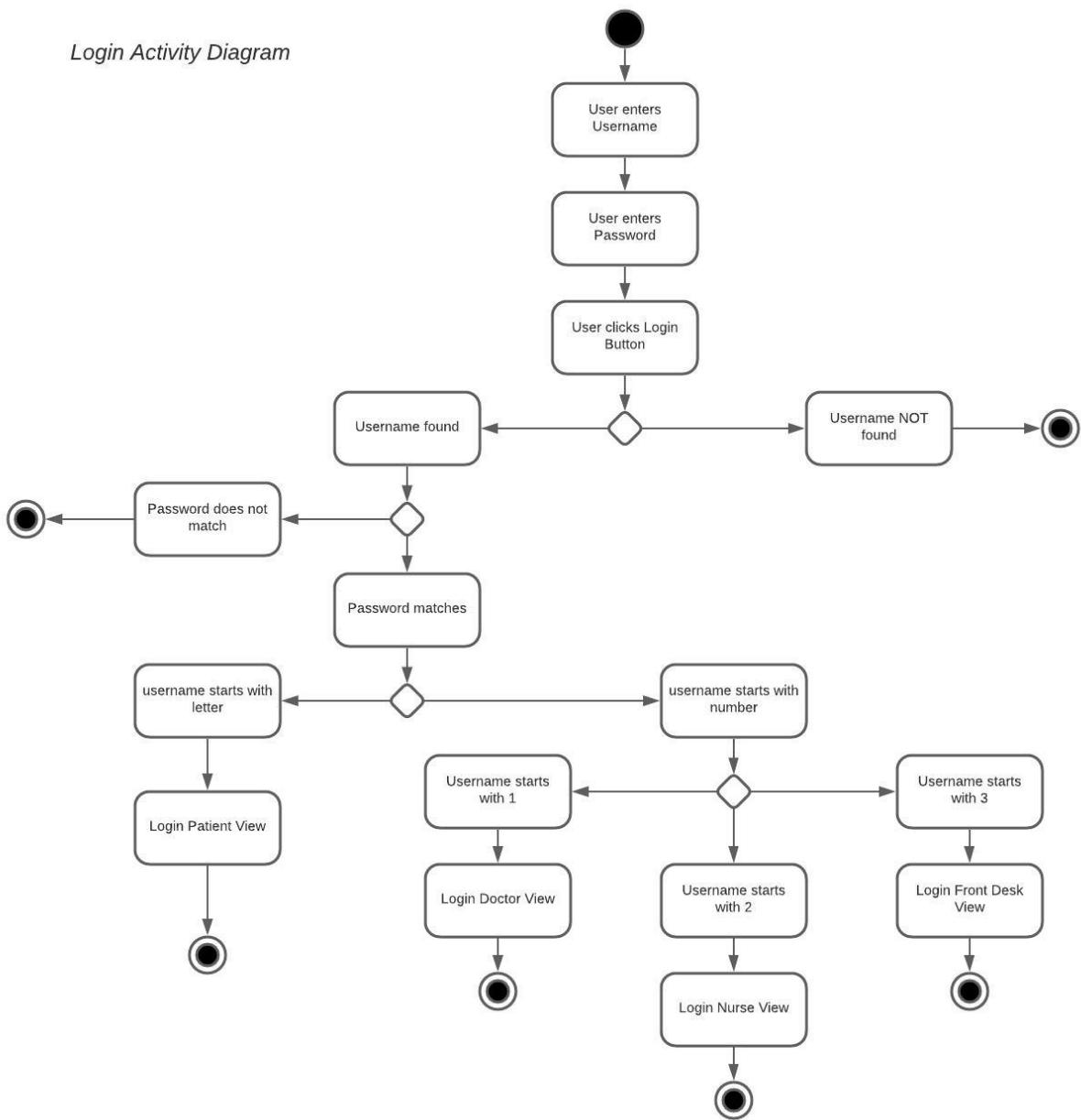
Contract CO6: viewPatientInfo	
Operation	viewPatientInfo (patientInfo : patinfo)
Cross Reference	Use Case: Patient Search
Preconditions	Staff Member is searching for a patient
Post conditions	<ul style="list-style-type: none"> • Patient info instance patinfo was created (<i>instance created</i>) • Patinfo is associated with Patient, based on clickSearch (<i>associated formed</i>)

Interaction Diagrams

Req-ID 1 Login: Using the login function, there are four different views; patient, front desk staff, nurse, and doctor. All staff will have usernames that begin with a number and patients will have a username that begins with a letter. In order to not duplicate code by coding each staff page for each specific view, the program will utilize the user's username to constantly determine which view they need to be in. Every page will extend the staff menu class, to help determine the view to display. This will utilize the open-closed principle by having one class for the staff menu extended to all the staff pages. This will allow each staff page to be viewed by the specific staff member based on their username. The login window will be extended to this staff menu class, so it will be able to reference the user's username throughout the duration the user is logged into the program.

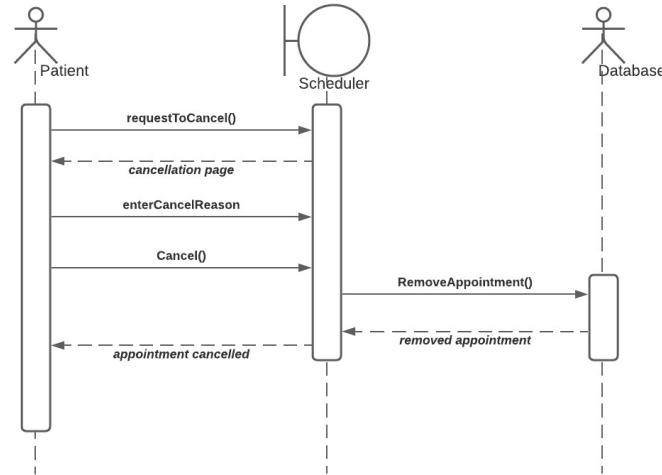
Login Sequence Diagram





Req-ID 3 Cancel Appointment: For a patient to cancel an appointment, they will click the cancel/reschedule button. This will redirect the patient to the cancel/reschedule page. To cancel an appointment, the patient will just enter a reason to cancel and then click the cancel button. The system will then remove the appointment from the staff calendar and the upcoming appointments page for the patient. The design principle that will be utilized will be the single-responsibility principle. This principle will be perfect in this case due to cancelling an appointment is done by clicking a single button.

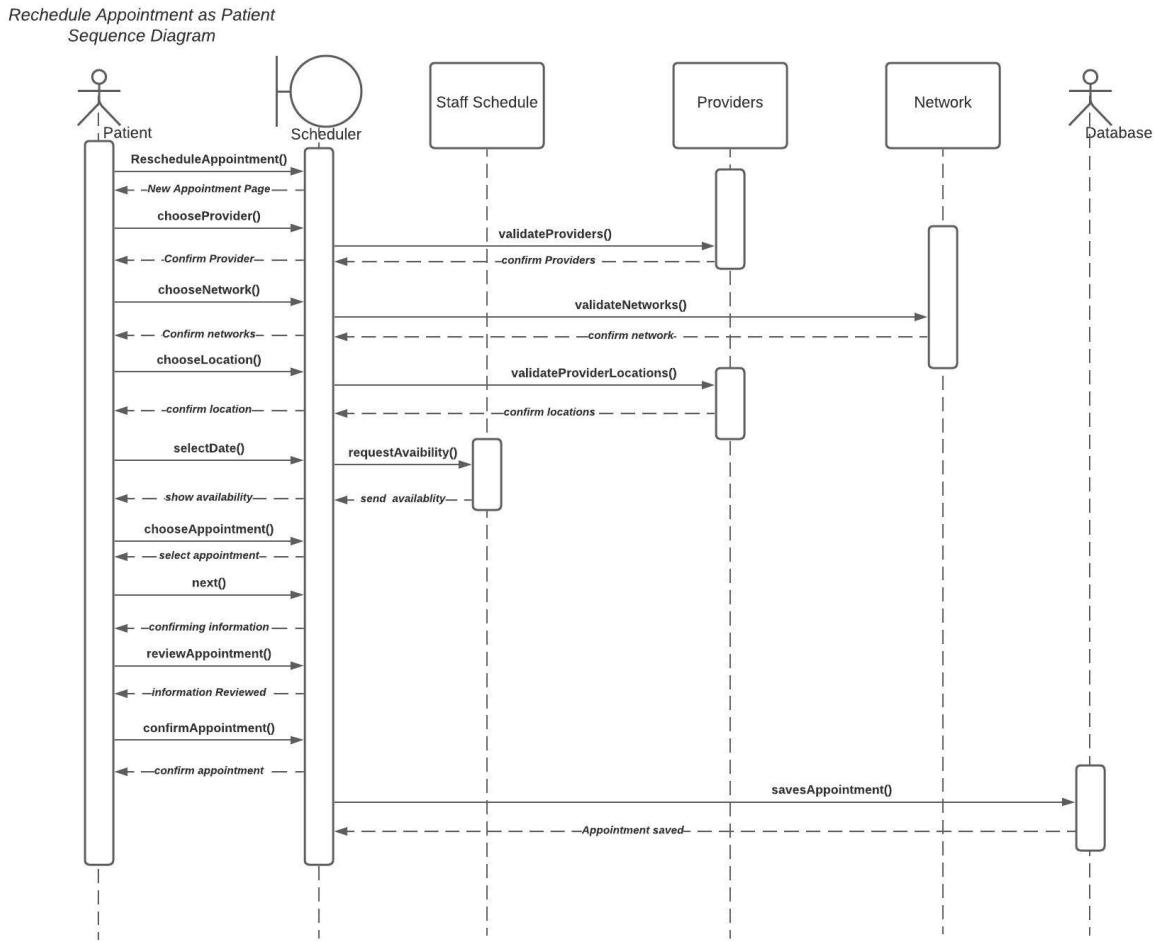
*Cancel Appointment as a Patient
Sequence Diagram*



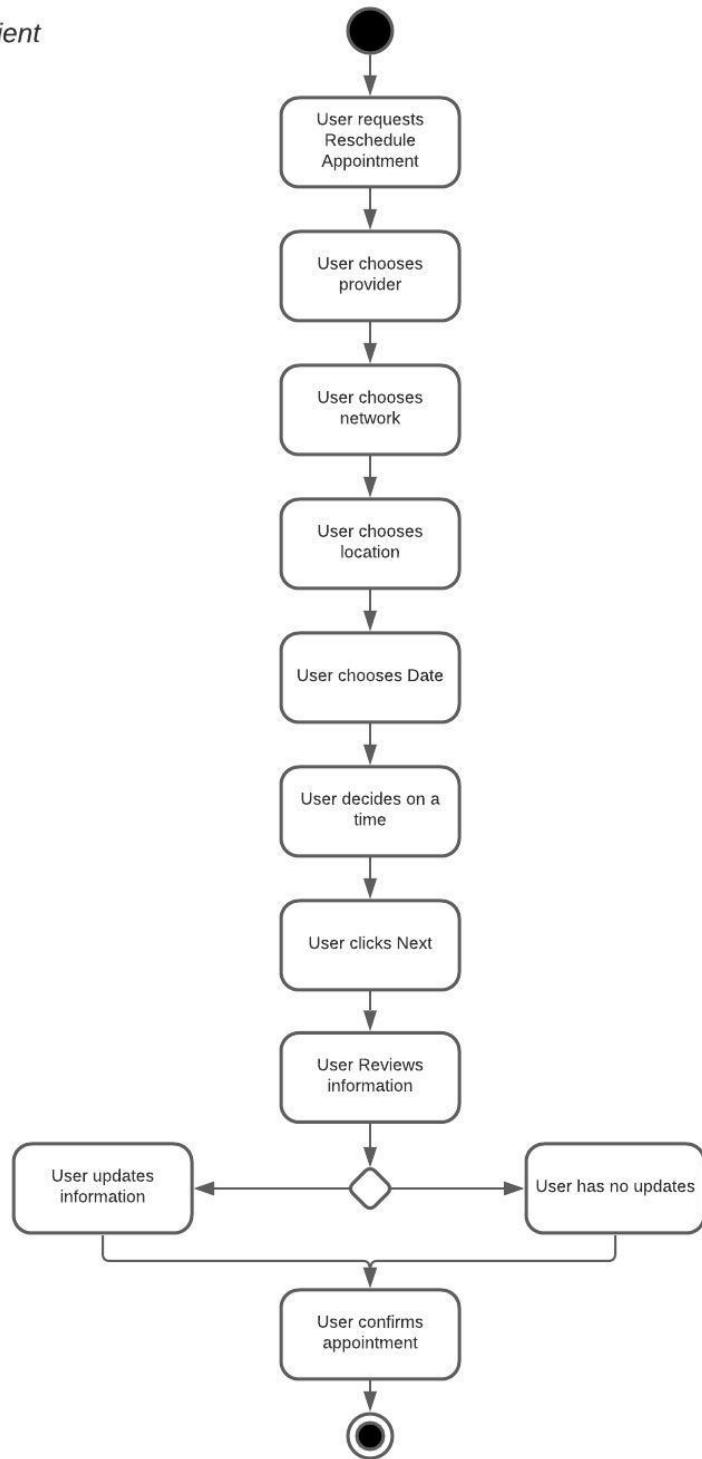
*Cancel Appointment as a Patient
Activity Diagram*



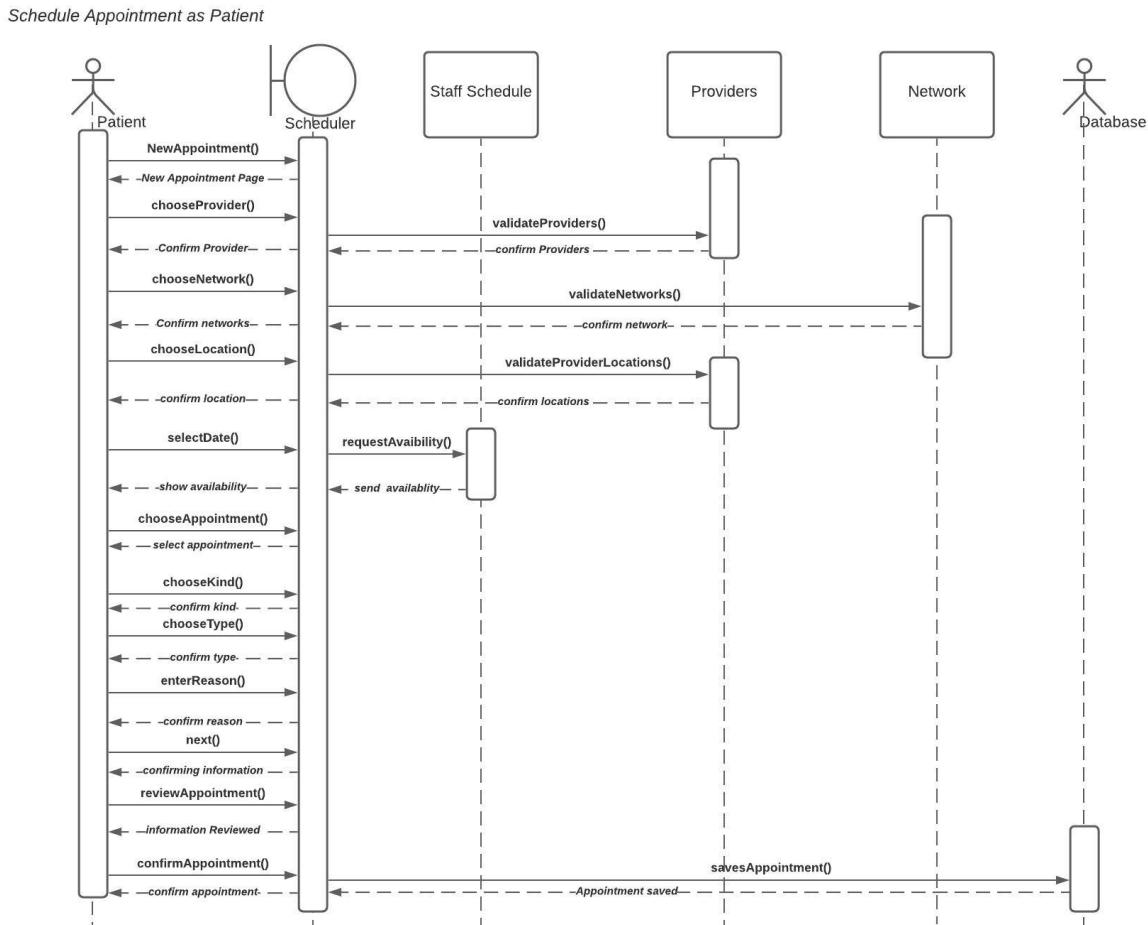
Req-ID 4 Reschedule Appointment: To reschedule an appointment, the user will click the cancel/reschedule appointment. The patient will then be redirected to the cancel/reschedule page. The second half of the page is set up similar to the schedule appointment page. The user will then choose a provider, network, location, date and time available. From there they will then be redirected to the review page where they can keep the kind of appointment, type of appointment, and reason for the appointment. Once confirmed the date will update the appointment to the new date. This will utilize the open-closed principle due to updating an appointment that has already been created.

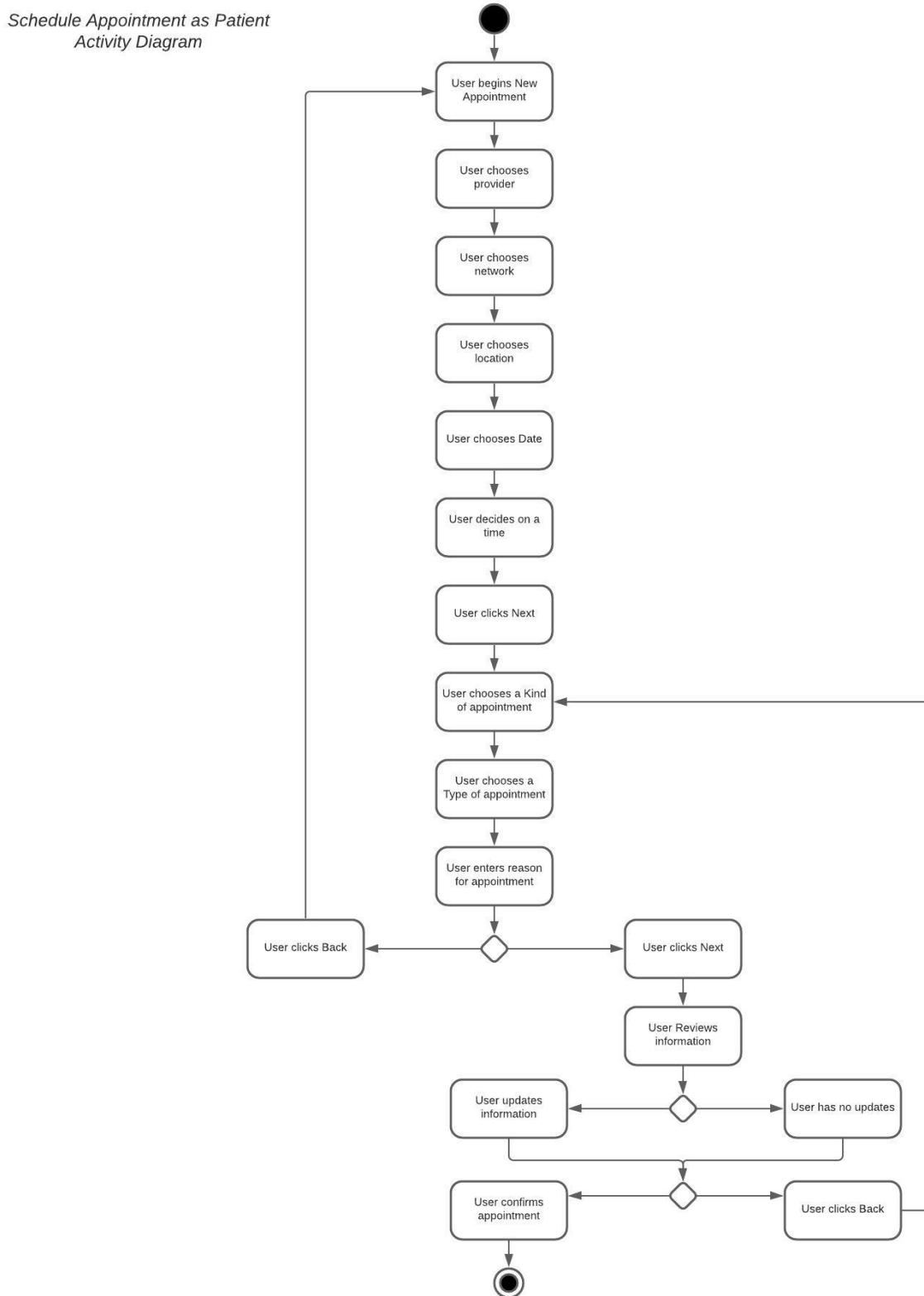


Reschedule Appointment as Patient Activity Diagram



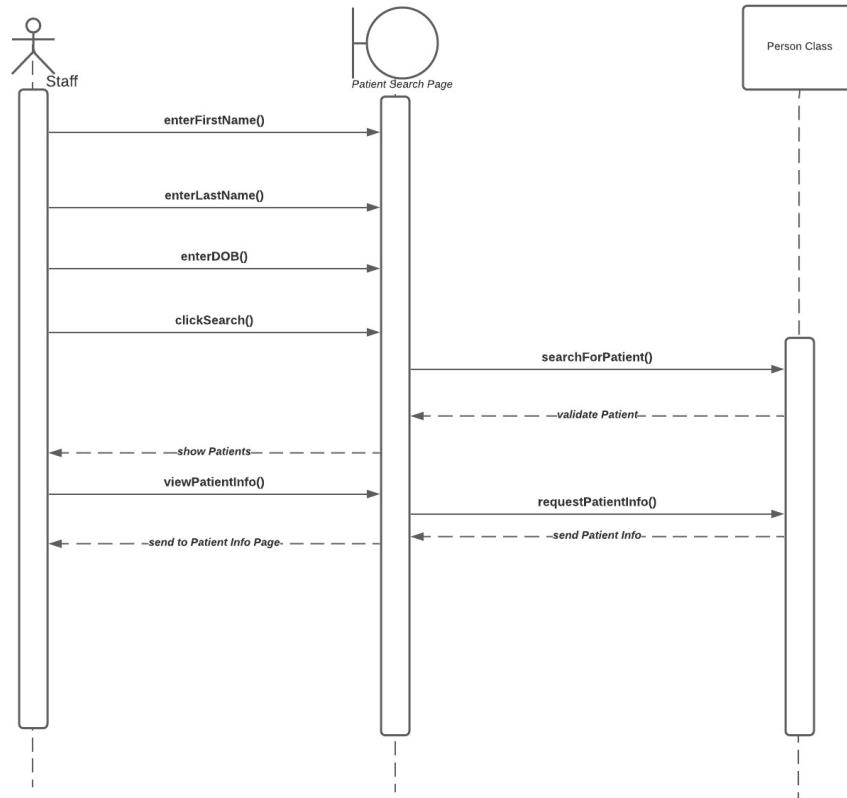
Req-ID 5 Schedule Appointment: To schedule an appointment as a patient, the user must first choose a provider, network, location, and date. They can then choose an available time based off of the staff schedule. The patient will then click next, which will take them to another screen where they can choose the kind of appointment, type of appointment and the reason for their appointment. If they click the back button they will be redirected to the previous page, the next button will direct the patient to the review page, where they can update their information or review it then confirm the appointment. For the purpose of scheduling an appointment in the patient view, we will be utilizing the interface-segregation principle (ISP). ISP will help break down the steps of scheduling an appointment into three sections broken down into screens. This will help the patients be able to focus on what is needed on that one screen versus seeing everything on one screen.

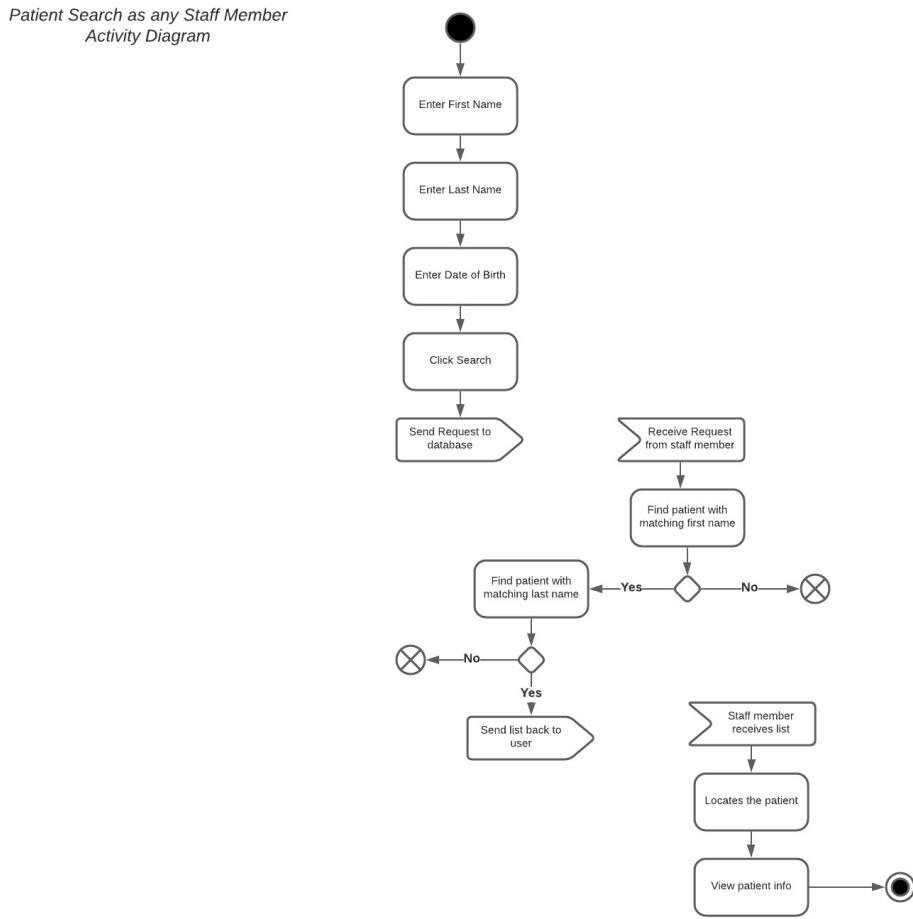




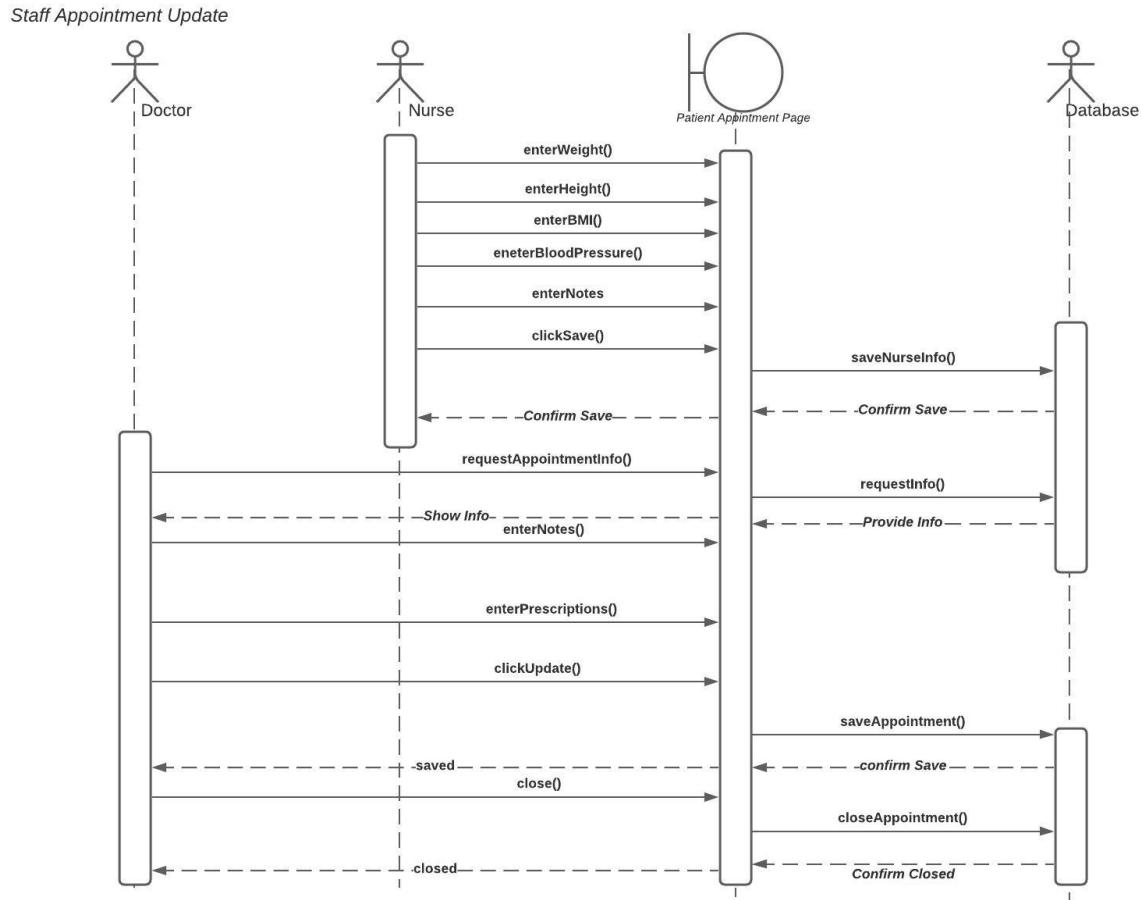
Req-ID 7 Patient Search: The patient search page is available to all the staff members and allows them to be able to search for a patient using their first name, last name, and date of birth. This page will utilize the single responsibility principle, which will have the program search a text document for the person.

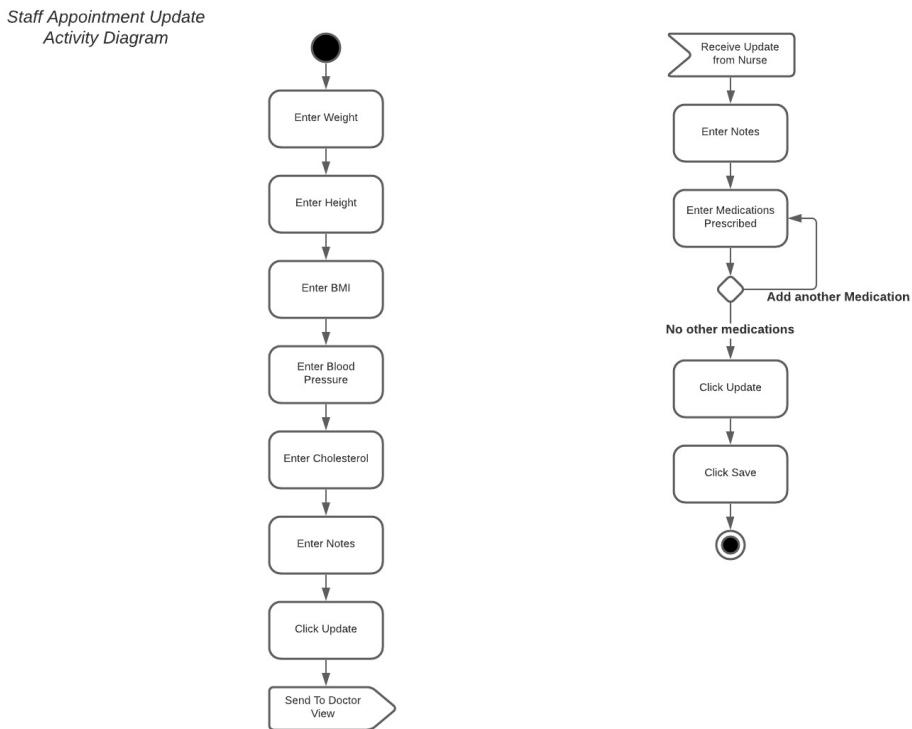
Patient Search as any Staff Member
Sequence Diagram



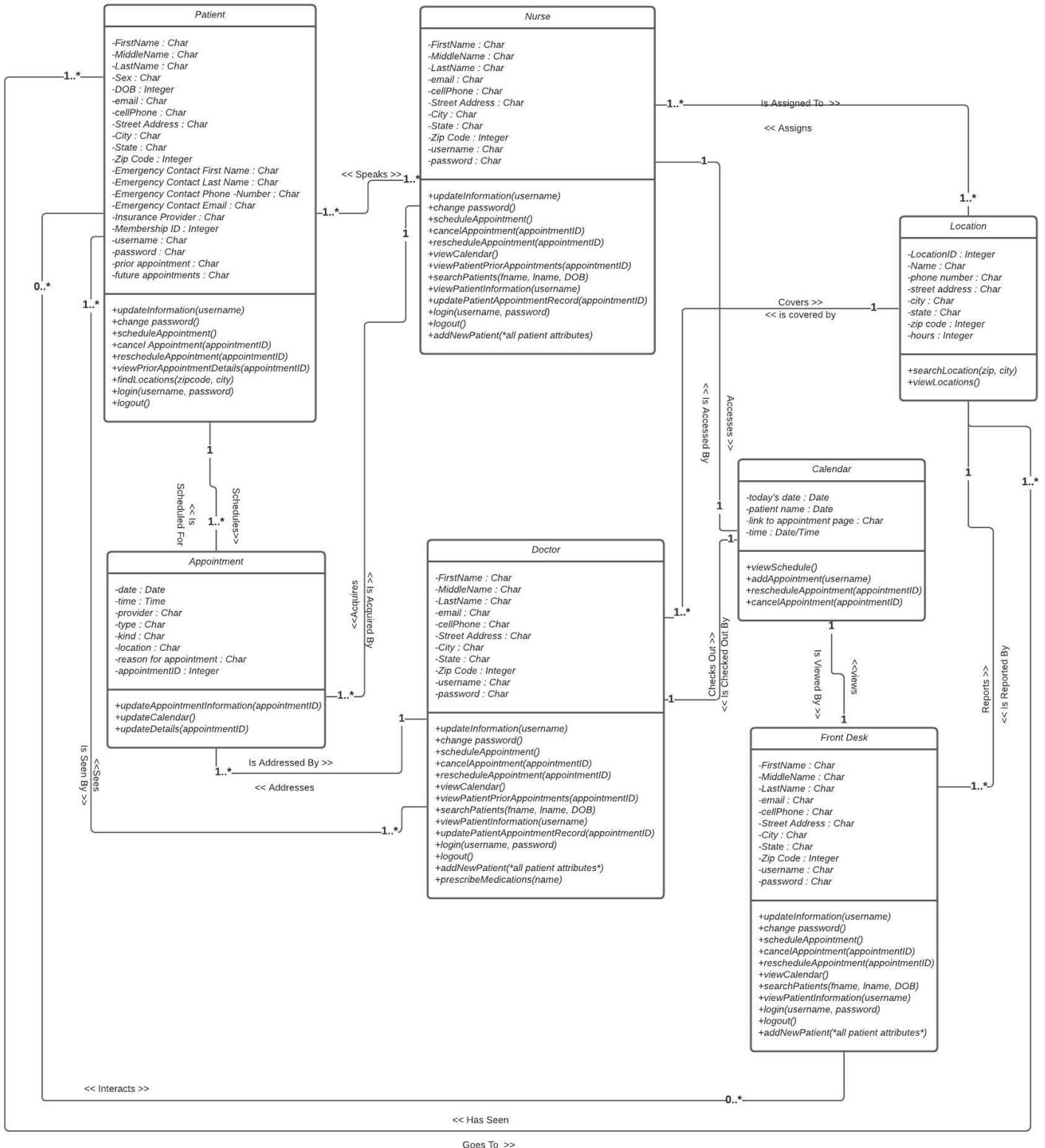


Req-ID 8 Staff Appointment Update: The staff appointment page allows the nurse and doctor to put in all the information gathered from the appointment. To be able to utilize the nurse view and the doctor view of the staff appointment update page, we will be utilizing the Liskov substitution principle. This principle will allow the nurse to only view what they are able to update. Once the nurse clicks the update button, the program will send the update to the doctor view, where the doctor will be able to view and add any additional notes and any prescribed medications.





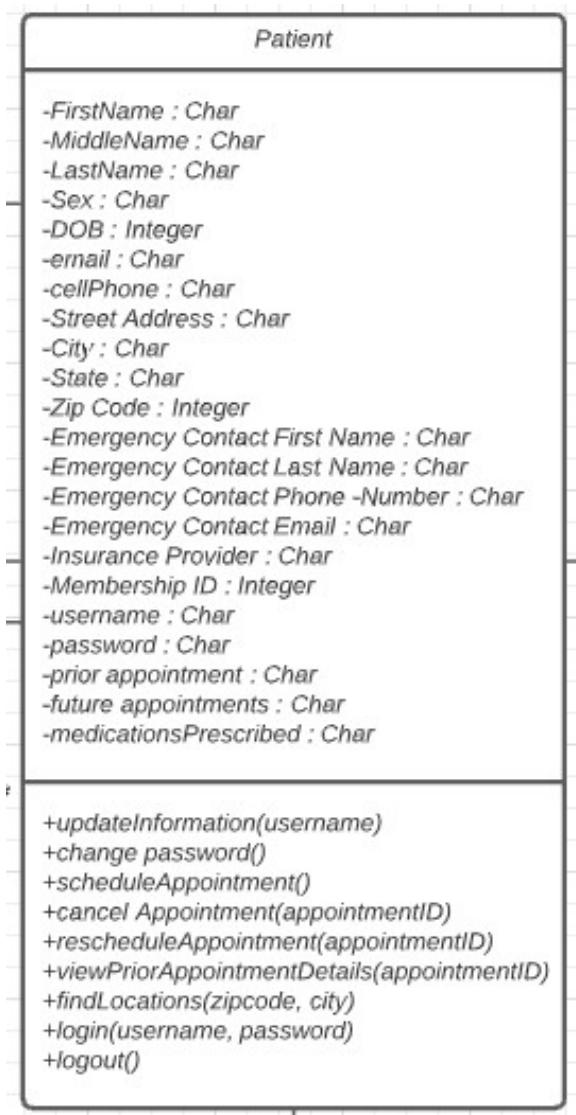
Class Diagram and Interface Specification



Data Types and Operation Signatures

UML Notation	Nature of the Relationship	Class1	Class2	Description
1..*	One-to-many	Patient	Doctor	A patient can have one doctor or many doctors.
1..*	One-to-many	Doctor	Patient	A doctor can have one patient or many patients.
1..*	One-to-many	Patient	Nurse	A patient can see one nurse or many nurses.
1..*	One-to-many	Nurse	Patient	A nurse can have one patient or many patients.
1..*	One-to-many	Patient	Appointment	A patient can have one appointment or many appointments.
1	One-and-Only-One	Appointment	Patient	An appointment can only have one patient.
0..*	Zero-to-many	Patient	Front Desk	A patient may interact with zero front desk staff to many front desk staff.
0..*	Zero-to-many	Front Desk	Patient	A front desk staff may interact with zero patients to many patients.
1	One-to-many	Patient	Location	A patient may go to one location or many locations.
1..*	One-to-many	Location	Patient	A location can have one patient to many patients.
1	One-and-Only-One	Front Desk	Calendar	A front desk staff will only have access to one calendar.
1	One-and-Only-One	Calendar	Front Desk	A calendar is only available to one front desk staff.
1	One-and-Only-One	Front Desk	Location	A front desk staff only works at one location.
1..*	One-to-many	Location	Front Desk	A location can have one to many front desk staff employees.
1..*	One-to-many	Nurse	Appointment	A nurse can have one appointment or many appointments.
1	One-and-Only-One	Appointment	Nurse	An appointment can only have one nurse.
1	One-and-Only-One	Nurse	Calendar	A nurse can view only one calendar.
1	One-and-Only-One	Calendar	Nurse	A calendar is only available for one nurse.
1..*	One-to-many	Nurse	Location	A nurse can work at one to many locations.

1..*	One-to-many	Location	Nurse	A location can have one to many nurses employed.
1..*	One-to-many	Doctor	Appointment	A doctor can have one to many appointments.
1	One-and-Only-One	Appointment	Doctor	An appointment can only have one doctor.
1	One-and-Only-One	Doctor	Calendar	A doctor can view only one calendar.
1	One-and-Only-One	Calendar	Doctor	A calendar is only available for one doctor.
1..*	One-to-many	Doctor	Location	A doctor can work at one to many locations.
1..*	One-to-many	Location	Doctor	A location can have one to many doctors employed.



1. Patient Class: This class is for every patient in the system.

Attributes: The attributes for this class are mainly characters (Strings in Java). This includes First Name, Middle Name, Last Name, Sex, Email, Cell Phone Number, Street Address, City, State, Emergency Contact First Name, Emergency Contact Last Name, Emergency Contact Phone Number, Emergency Contact Email, Insurance Provider, username, password, Prior Appointments, Future Appointments, Medications Prescribed, and Previous Pharmacy Location. The date of birth (DOB), zip code, and membership ID will all be integers.

Methods:

- The updateInformation(username) method allows the user to update their information in the system.
- The changePassword() method allows the user to change the password of the user. This will update their information as well.
- scheduleAppointment() method allows the patient to schedule an appointment.
- cancelAppointment(appointmentID) method allows the patient to cancel a future appointment. What is needed for this is the appointmentID to make sure the correct appointment is being cancelled.
- rescheduleAppointment(appointmentID) will allow the patient to reschedule a future appointment. What is needed for this is the appointmentID to make sure the correct appointment is being rescheduled.

- vi. `findLocations(zipcode, city)` allows the patient to search for a location by entering the zipcode and city of the location the patient is searching for.
- vii. `Login(username, password)` allows the patient to log into the system by entering their username and password.
- viii. `Logout()` this allows the patient to log out of the system.
- ix. `viewPriorAppointmentDetails(appointmentID)` allows the patient to view a prior appointment that they have chosen. What is needed for this is the appointmentID to verify which appointment the patient wishes to view.



2. Front Desk Class: This is for every front desk staff member that has access to the application.

Attributes: Most of the attributes for the front desk staff are characters. These attributes include first name, middle name, last name, email, cell phone, street address, city, state, username and password. The only attribute that is an Integer is the zip code.

Methods:

- i. `updateInformation(username)` allows the front desk staff to update their information and patient information.
- ii. `changePassword()` allows the front desk staff to change their password.
- iii. `scheduleAppointment()` allows the front desk staff to schedule an appointment for a patient.
- iv. `cancelAppointment(appointmentID)` allows the front desk staff to cancel an appointment for a patient. What is needed for this is the appointmentID to make sure the correct appointment is being cancelled.
- v. `rescheduleAppointment(appointmentID)` allows the front desk staff to reschedule an appointment for a patient. What is needed for this is the appointmentID to make sure the correct appointment is being rescheduled.
- vi. `viewCalendar()` allows the front desk staff to view the calendar.
- vii. `searchPatients(fname, lname, DOB)` this allows the front desk staff to search for a patient using their first name, last name, and date of birth (DOB).
- viii. `viewPatientInformation(username)` this allows the front desk staff to view the patient's contact information by using the patient's username.
- ix. `Login(username, password)` this allows the front desk staff to log into the system using a username and password.
- x. `Logout()` this allows the front desk staff to log out of the system.

- xi. `addNewPatient(*all patient attributes*)` this allows the front desk staff to create a new patient and add them to the system. The staff member will need the patient's first name, middle name, last name, DOB, phone number, email, SSN, insurance provider, membership ID, street address, city, state, zip code, emergency contact first and last name, phone, email, username, and password.

3. Nurse Class:

This is for every nurse that has access to the application.

Attributes: The attributes are mostly characters. The attributes that are characters include the nurse's first name, middle name, last name, email, cell phone, street address, city, state, username, and password. The only attributes that is an Integer is the zip code.



Methods:

- i. `updateInformation(username)` allows the nurse to update the information of their account or a patient's account.
- ii. `changePassword()` allows the nurse to update their password.
- iii. `scheduleAppointment()` allows the nurse to schedule an appointment for a patient.
- iv. `cancelAppointment(appointmentID)` allows the nurse to cancel an appointment for a patient. The appointmentID will have to be passed in order to cancel the right appointment.
- v. `rescheduleAppointment(appointmentID)` allows the nurse to reschedule an appointment for a patient. The appointmentID will have to be passed in order to reschedule the correct appointment for the patient.
- vi. `viewCalendar()` allows the nurse to view the calendar.
- vii. `viewPatientPriorAppointments(appointmentID)` allows the nurse to view a patient's prior appointment by passing the appointmentID for the appointment the nurse wants to view.
- viii. `searchPatients(fname, lname, DOB)` this allows the nurse to search for a patient using their first name, last name, and date of birth (DOB).
- ix. `viewPatientInformation(username)` this allows the nurse to view the patient's contact information by using the patient's username.
- x. `updatePatientAppointmentRecord(appointmentID)` allows a nurse to update the appointment page for the patient's appointment. The appointment ID will be passed in order for the nurse to update the correct appointment.
- xi. `Login(username, password)` this allows the nurse to log into the system using a username and password.
- xii. `Logout()` this allows the nurse to log out of the system.

- xiii. addNewPatient(*all patient attributes*) this allows the nurse to create a new patient and add them to the system. The staff member will need the patient's first name, middle name, last name, DOB, phone number, email, SSN, insurance provider, membership ID, street address, city, state, zip code, emergency contact first and last name, phone, email, username, and password.

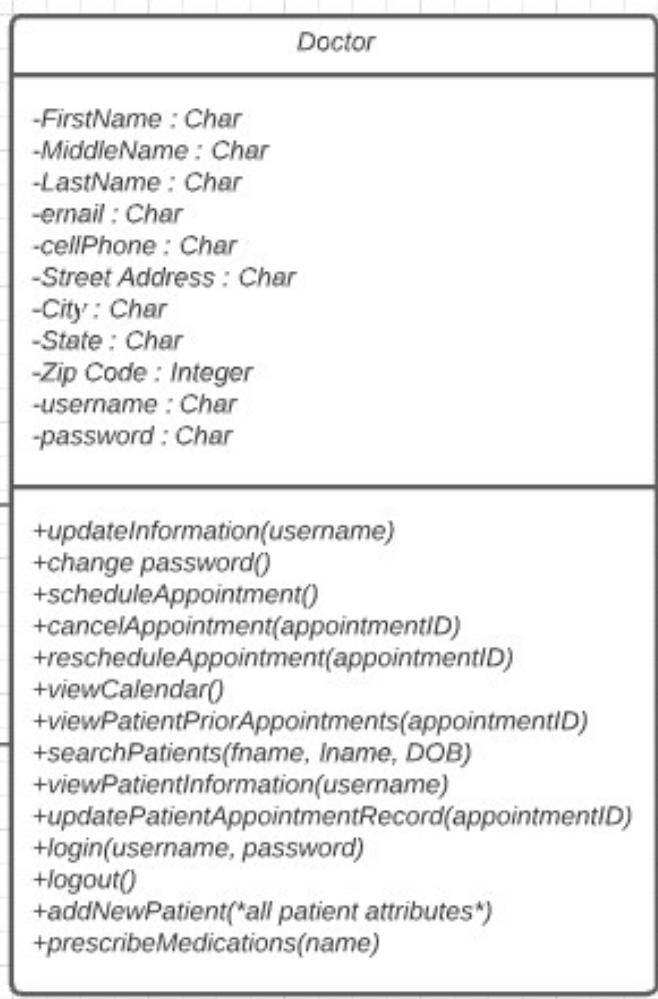
4. Doctor Class: This is for every doctor that has access to the application.

Attributes: The attributes are mostly characters. The attributes that are characters include the doctor's first name, middle name, last name, email, cell phone, street address, city, state, username, and password. The only attributes that is an Integer is the zip code.

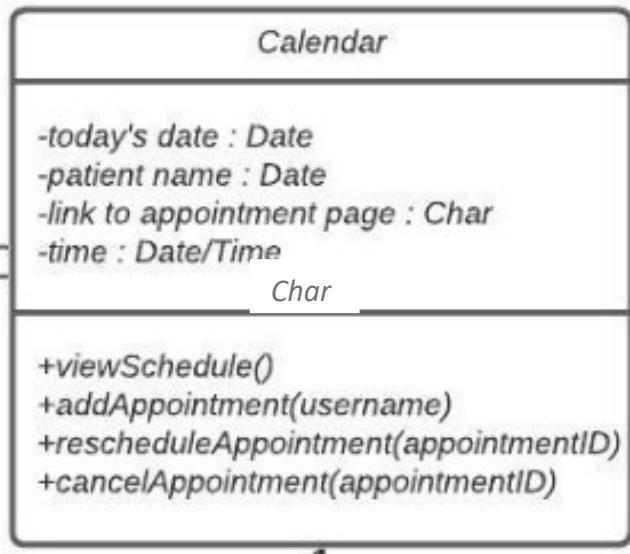
Methods:

- i. updateInformation(username) allows the doctor to update the information of their account or a patient's account.
- ii. changePassword() allows the doctor to update their password.
- iii. scheduleAppointment() allows the doctor to schedule an appointment for a patient.
- iv. cancelAppointment(appointmentID) allows the doctor to cancel an appointment for a patient. The appointmentID will have to be passed in order to cancel the right appointment.
- v. rescheduleAppointment(appointmentID) allows the doctor to reschedule an appointment for a patient. The appointmentID will have to be passed in order to reschedule the correct appointment for the patient.
- vi. viewCalendar() allows the doctor to view the calendar.
- vii. viewPatientPriorAppointments(appointmentID) allows the doctor to view a patient's prior appointment by passing the appointmentID for the appointment the doctor wants to view.

- viii. searchPatients(fname, lname, DOB) this allows the doctor to search for a patient using their first name, last name, and date of birth (DOB).
- ix. viewPatientInformation(username) this allows the doctor to view the patient's contact information by using the patient's username.
- x. updatePatientAppointmentRecord(appointmentID) allows a doctor to update the appointment page for the patient's appointment. The appointment ID will be passed in order for the doctor to update the correct appointment.
- xi. Login(username, password) this allows the doctor to log into the system using a username and password.



- xii. Logout() this allows the doctor to log out of the system.
- xiii. addNewPatient(*all patient attributes*) this allows the doctor to create a new patient and add them to the system. The staff member will need the patient's first name, middle name, last name, DOB, phone number, email, SSN, insurance provider, membership ID, street address, city, state, zip code, emergency contact first and last name, phone, email, username, and password.
- xiv. prescribeMedications(name) allows a doctor to prescribe medications to patients on the appointment page. Only the name will of the prescription will have to be passed.

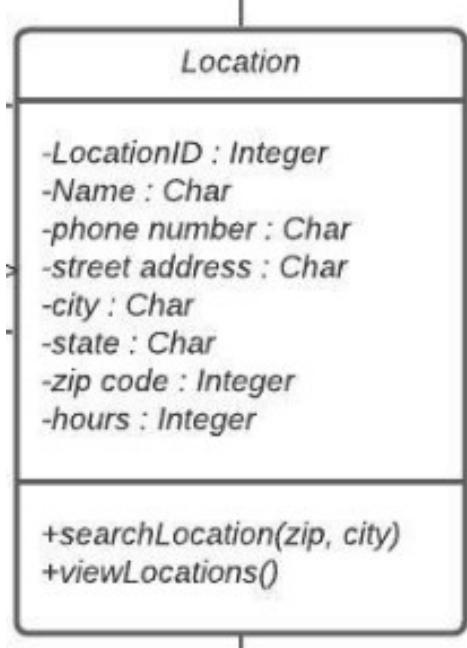


5. Calendar Class: This is a class that is available for all three staff views and updates as the appointments are added, removed, or rescheduled.

Attributes: The attributes for the calendar class include today's date as a Date. Patient's name and link to appointment page are both characters. Finally, time is going to be Date/Time.

Methods:

- i. viewSchedule() allows the staff member to view the calendar and see all the appointments for the month.
- ii. addAppointment(username) allows the staff member to create an appointment for a patient using the patient's username. This will add the appointment to the calendar.
- iii. rescheduleAppointment(appointmentID) allows the staff member to reschedule an appointment using the appointment ID. This will update the appointment on the calendar.
- iv. cancelAppointment(appointmentID) allows the staff member to cancel an appointment using an appointment ID. This will remove the appointment from the calendar.

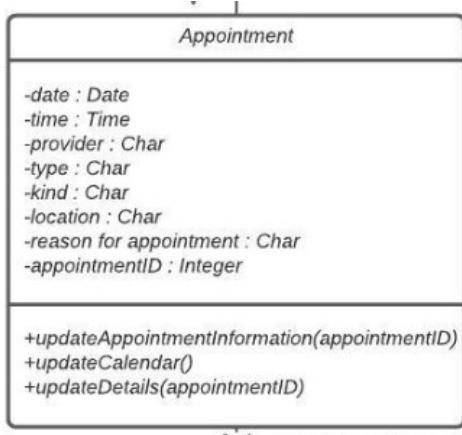


6. Locations Class: This class is for each location available for patient's to make appointments and for staff to access the application.

Attributes: The location ID will have a data type of an integer. This will allow each location to have a unique title. The location's name, phone number, street address, city, and state will be character data types. The zip code and hours will be integers.

Methods:

- i. `searchLocation(zip, city)` is a method that searches for a location using the zip code and city that is entered by the patient.
- ii. `viewLocations()` shows the locations information for any in that area.



7. Appointment Class: This class is for each appointment that is created and identified uniquely by the appointment ID.

Attributes: The date is going to be a date data type and the time is going to be a date/time data type. The appointment ID is going to be an integer data type. The remaining attributes including provider, type, kind, location, and reason for appointment are all going to be character data types.

Methods:

- i. `updateAppointmentInformation(appointmentID)` is a method that allows nurses and doctors to update the appointment with the information collected during the appointment. The appointment ID is passed in order to update the correct record.
- ii. `updateCalendar()` is a method that updates the calendar with the date, time, and patient name to the calendar when an appointment is created.
- iii. `updateDetails(appointmentID)` when there are changes to an appointment, `updateDetails` method will update the appointment information using the appointment ID to make sure the correct appointment is being updated.

Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
1	5	X											
2	5		X										
3	5			X									
4	5				X								
5	5					X							
6	3						X						
7	5							X					
8	5								X				
9	5									X			
10	5										X		
11	5											X	
12	5												X
Max PW		5	5	5	5	5	3	5	5	5	5	5	5
Total PW		5	5	5	5	5	3	5	5	5	5	5	5

Description of Requirements with Test Cases

Req-ID	Requirement	Priority Weight	Test Case ID	Test Case Description
1	Log In	HIGH	TC01	Login with invalid username and password
			TC02	Login without inputting username and password
			TC03	Login with valid credentials
			TC04	Login as patient
			TC05	Login as Front Desk Staff
			TC06	Login as Nurse
			TC07	Login as Doctor
2	Update Information	HIGH	TC08	Patient updates information
			TC09	Front Desk updates information
			TC10	Nurse updates information
			TC11	Doctor updates information
3	Cancel Appointment	HIGH	TC12	Patient cancels appointment
			TC13	Front Desk cancels appointment
			TC14	Nurse cancels appointment
			TC15	Doctor cancels appointment
4	Reschedule Appointment	HIGH	TC16	Patient reschedules appointment
			TC17	Front desk reschedules appointment
			TC18	Nurse reschedules appointment
			TC19	Doctor reschedules appointment
5	Schedule Appointment	HIGH	TC20	Patient schedules appointment
			TC21	Front desk schedules appointment
			TC22	Nurse schedules appointment

			TC23	Doctor schedules appointment
6	Pre-Appointment Information	MED	TC24	Patient completes pre-appointment information
7	Staff Patient Search	HIGH	TC25	Front desk searches for patient
			TC26	Nurse searches for patient
			TC27	Doctor searches for patient
8	Staff Appointment Update	HIGH	TC28	Nurse updates patient appointment page
			TC29	Doctor updates patient appointment page
9	Adding New Patient	HIGH	TC30	Front desk adds a new patient
			TC31	Nurse adds a new patient
			TC32	Doctor adds a new patient
10	Prior Appointments	HIGH	TC33	Patient views prior appointments
			TC34	Nurse views prior appointments
			TC35	Doctor views prior appointments
11	Staff View Patient Schedule	HIGH	TC36	Front desk views patient schedule
			TC37	Nurse views patient schedule
			TC38	Doctor views patient schedule
12	Log Out	HIGH	TC39	Click CLOSE on login menu
			TC40	Patient clicks Exit in application menu
			TC41	Patient clicks Log Out in application menu
			TC42	Front Desk clicks Exit in application menu
			TC43	Front Desk clicks Log Out in application menu
			TC44	Nurse clicks Exit in application menu
			TC45	Nurse clicks Log Out in application menu
			TC46	Doctor clicks Exit in application menu
			TC47	Doctor clicks Log Out in application menu

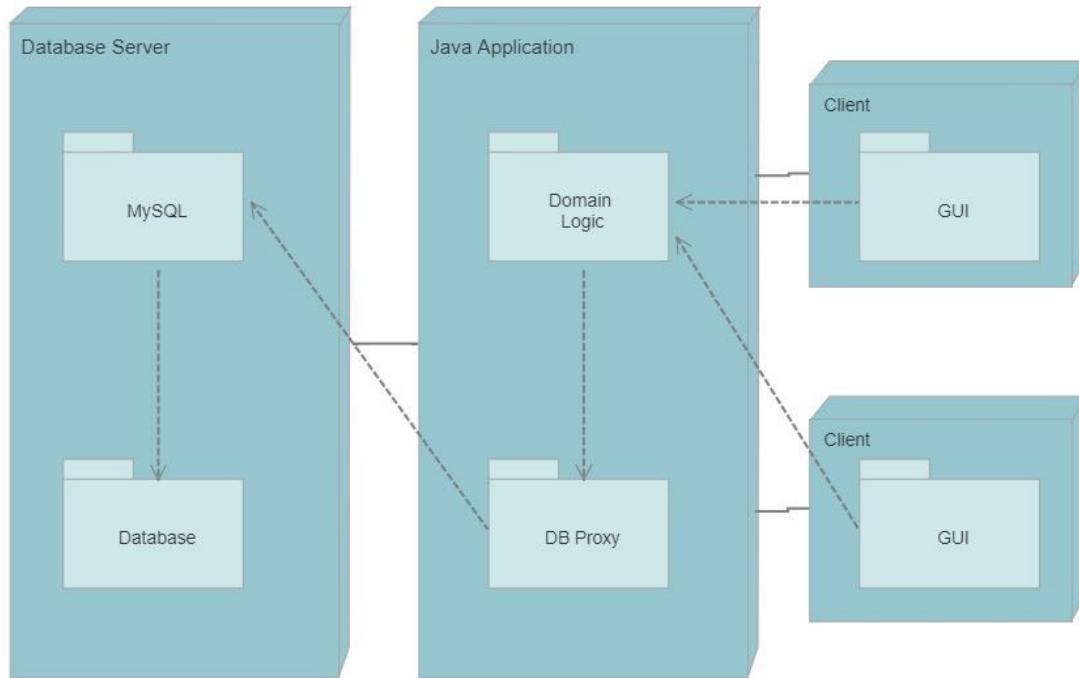
System Architecture and System Design

Architectural Styles

The architectural style for the patient tracker system will be the client-server model. All of the data of the appointments, patient information, and health records will be kept on a server. All the users that have access to the desktop application will be able to connect to MySQL database to be able to access the information.

Identifying Subsystems

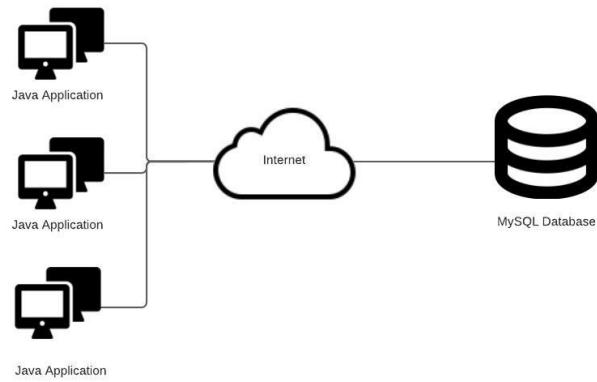
UML Package Diagram



Each client will access the GUI for the Java Application. The Java Application includes all the domain logic, which allows the client to make the decisions on what they plan to execute. This would include, updating information, scheduling an appointment, cancelling an appointment, or rescheduling an appointment. The domain logic depends on the DB Proxy, which connects to the MySQL Database Server. MySQL will access the database containing all the data for the application.

Mapping Subsystems to Hardware

This application will be downloaded on any computer that wants access to the application. There will be a MySQL database running on one computer that all the other computers will have access to by connecting to the database. All the computers that have downloaded the application will connect to the internet, where they will be able to log in and access the database data for the program. Below is an image on what this would look like.



Persistent Data Storage

This application does require saving data constantly. From the user updating information, to creating appointments, the application will require frequent saving. For persistent data storage, the application will use relational databases in order to keep up with the saving and updates to the data. MySQL will be the relational database used in this application.

Network Protocol

The network protocol being used by this application is Java JDBC. This network protocol was chosen because Java JDBC works well with MySQL. MySQL will allow the application to be stored in a database on the server. Java JDBC will be able to connect to MySQL to access the information.

Global Control Flow

Execution Order

This application is event driven. The application will only act if the user clicks button or enters information.

Time Dependency

This application is real-time. As updates are created, the data will automatically do those updates. This also is the same for creating appointments, adding patients, and updating the appointments. If multiple users are looking at the same page and someone updates the information, the other user may have to refresh the page in order for the information to update, but overall, the time dependency should be real-time.

Hardware Requirements

The user will need a color display using Windows (Vista to Windows10), Max OS X, or Linux. Below is a broken down list depending on the operating system.

Windows

- Windows Server 2008 R2 SP1 (64-Bit)
- Windows Server 2012 and 2012 R2 (64-Bit)
- RAM: at least 128 MB
- Disk Space: at least 124 MB **note for Java Update the user will need 2MB
- Processor: at least 266 MHz processor

Mac OS X

- Administrator privileges for installation
- 64-Bit browser

Linux

- Oracle Linux 5.5+¹, 6.x, or 7.x
- Red Hat Enterprise Linux 5.5+¹, 6.x, or 7.x
- Suse Linux Enterprise Server 10 SP2+, 11.x or 12.x
- Ubuntu Linux 12.04 LTS, 13.x , 14.x, 15.04, 15.10

Algorithms and Data Structures

Data Structures

This system currently uses arrays to read information from a text file until MySQL database is able to connect to Java. Currently the text file is set up as all the information listed with commas in between each index. The code will then search the file for a specific index. Each line starts a new array, which helps with looking for a username and password in order to enter the program. This text file is only temporary until MySQL is updated and ready to connect, then the code will be able to search through the database versus a text file.

All the appointments are set up as array objects as well that include all the information gathered from scheduling an appointment. These will save to a text file (for testing) and eventually automatically create a new record in the MySQL database.

Arrays were more flexible to use in order to search for specific information. Having a username for the person text file as the first index helps when searching for a specific person. The code also sets up how it will process in MySQL by being able to search records to find specific people or appointments.

User Interface Design and Implementation

The layout design hasn't changed too drastically from the mock ups in Report #1. However, the calendar page is better than the design. Overall this will help make the calendar page more user friendly by having add buttons on every date. Also, when an appointment is made, the number of appointments for that day will increase. Once the staff member clicks the View button, the information regarding all the appointments that day will show up on the right hand column.

Also, to schedule the appointment, I originally had buttons for the patient to choose the kind and type into radio buttons. This will allow the data to be saved more easily as well as more user friendly for the user so they can see which option is selected.

Design of Tests

Demo 2 Test Cases

Req-ID	Requirement	Priority Weight	Test Case ID	Test Case Description	Status
3	Cancel Appointment	HIGH	TC12	Patient cancels appointment	Passed
			TC13	Front Desk cancels appointment	Passed
			TC14	Nurse cancels appointment	Passed
			TC15	Doctor cancels appointment	Passed
4	Reschedule Appointment	HIGH	TC16	Patient reschedules appointment	Passed
			TC17	Front desk reschedules appointment	Passed
			TC18	Nurse reschedules appointment	Passed
			TC19	Doctor reschedules appointment	Passed
6	Pre-Appointment Information	MED	TC24	Patient completes pre-appointment information	Passed
7	Staff Patient Search	HIGH	TC25	Front desk searches for patient	Passed
			TC26	Nurse searches for patient	Passed
			TC27	Doctor searches for patient	Passed
8	Staff Appointment Update	HIGH	TC28	Nurse updates patient appointment page	Passed
			TC29	Doctor updates patient appointment page	Passed
12	Prior Appointments	HIGH	TC38	Patient views prior appointments	Passed
			TC39	Nurse views prior appointments	Passed
			TC40	Doctor views prior appointments	Passed

The above chart shows the test cases that will be tested during the second demo. Currently, everything seems to be working and does the task.

The bottom-up integration strategy will be used for this demo. The program is based on restricted access, so to implement that, the log in views of patient, front desk staff, nurse, and doctor were all implemented first. This program is focused on storing patient information and storing it with the access to edit and update information. Patients are also able to schedule their own appointments through this application as well. Staff will also have the capability to schedule appointments for patients. Staff also has the ability to search for a patient and add a new patient to the system. Going through the demo I will start with implementing the patient side of the program then move onto the front desk staff, nurse, and then end with the doctor side.

History of Work, Current Status, and Future Work

Project Coordination and Progress Report

The following Use Cases have been implemented for the second demo.

Cancel Appointment
Reschedule Appointment
Pre-Appointment Information
Staff Patient Search
Staff Appointment Update
Prior Appointments

All of the above use cases are fully functional with a few minor kinks that need worked out. Future updates will include a medication refill request by the patient, a myHealth page for the patient, nurse, and doctor to view the patient's development over time through appointments. There will also be a "forgot password" link for users who forgot their password can request a new one. Originally these were going to be implemented for demo two, but due to time constraints and unexpected challenges in the process, these were pushed out as an update.

History of Work

Due to complications with implementing some of the functions, the lower priority use cases were pushed into an update instead of part of the next demo. This was so, more time could be focused on the higher priority use cases versus just trying to get everything done with half the effort.

Accomplishments in the Project

- I was able to implement a program with 4 different user views.
- I was able to manipulate text documents using Java
- I was able to create a calendar that showed all appointments for certain days.
- I was able to search a document for a patient's name and a location.
- I was able to update user data.
- I was able to schedule appointments for all views.
- I was able to pull up information all based on the username of the patient.
- I was able to gain more confidence in coding in Java.

Plan of Work Patient Tracker

Summary of Changes

1. The use cases were reordered from highest priority to lowest priority.
2. The project objectives altered a tad from the original concept. It was more the final project included less than expected, but with future updates, the whole project objective will be implemented.
3. The domain analysis had data types for the attributes and type of attribute included.
4. Customer Problem Statement was updated in the section regarding the myHealth and medication refills section, that they would be incorporated as future updates. Originally they were set up to be in the final demo, but were pushed out due to timing constraints.
5. System Requirements were updated to have the requirements that were not being implemented highlighted in gray and added to the bottom of the list of the requirements. It is noted below the chart that those items will be implemented in a future update.
6. The use cases were also broken down and rearranged by what was being implemented in the second demo. Anything highlighted in gray was noted that those functions will be implemented in a future update.
7. The first tractability matrix was rearranged by the highest priority and noted any additional requirements that would be implemented in a future update.
8. The User Effort Estimation still included three of the features that were going to be implemented in a future update, and were just highlighted gray with a note mentioning that they are going to be implemented at a later date.
9. The class diagram removed the prescription class and the myHealth class. All the methods and attributes that were involved in those processes were removed from the patient, nurse, and doctor classes.
10. The Gantt chart was also updated to include the new schedule that occurred and a future section as well.

References

Baxter, S. (2021, 02 6). Nurse at Jane Pauley.

I interviewed Shaunee Baxter, who is a nurse at Jane Pauley. She helped indicate the type of information that nurses look for at the beginning of appointments like weight, height, blood pressure, heart rate, pulse, and temperature for vitals. This information was helpful to the Appointment Details pages.

Marathon Health. (2021, 02 06). Retrieved from Marathon Health Portal:

https://member.ourhealth.org/sign_in

This page was an example of how a patient portal should be used. Referencing the information needed for appointments, and the steps to set up appointments was key in understanding how to create an application that will be doing this task. I used my parent's login information in order to see how their portal worked and how to incorporate their basic concepts into my own application.

Nishadha. (2021, March 9). *UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples*. Retrieved from Creately:

<https://creately.com/blog/diagrams/uml-diagram-types-examples/>

This reference assisted me in researching different types of UML diagrams other than sequence diagrams. I found Activity Diagrams and was able to implement those to go along with each sequence diagram.

Wikipedia. (2021, Feb 24). *SOLID*. Retrieved from Wikipedia:

<https://en.wikipedia.org/wiki/SOLID>

This reference was used to be able to research more design principles. I was able to research which ones I could use for my project and more details on how to implement them.