

Hospital Management System

Group Members:

- Sebastian Silva Benites
- Syed Ikramuddin
- Hardipsinh Matroja
- Mustafa Ali Taleb Al-Omari
- Louie Go
- Kamalkumar, Patel
- Neethu Dominic
- Belove Nareshkumar Patel



Part A: Requirements Analysis Document

Section 1: Problem Statement

1.1.a. Problem and Need

Dealing with hospitals and doctors is no easy task, getting a date for your appointment, forgetting when you made the appointment, receiving your prescription and forgetting where you put it. It may not seem so difficult but with the increasing speed of day-to-day life it gets more complicated every time. Also, the hospital may have some problems, confusing dates, not registering appointments or not delivering medicines in the appropriate date.

Therefore, we will make a software where patients, doctors and hospital staff will have their account where they can manage appointments and patient's history and prescription medicines. Patients can book appointments online whilst the doctor and hospital admins can see, manage or reschedule appointments. Patients can also consult the doctors using the video interface from anywhere. Moreover, patients can see prescriptions, doctor's advice and review their own medical history.

1.1.b. Capabilities and Benefits

(i) Capabilities

- Manage Doctors Appointments
- Display and Store Patient's History
- Display Doctors Available hours
- Display and Store Patients Prescriptions
- Manage Patient's personal information
- Online consultation through video call

(ii) Benefits

- Minimize time on looking up records for a certain patient
- Remove the paper-based system completely
- Create a more systematic and organized way of keeping and storing patient's records
- Create an easier way to make an appointment with a doctor.
- Patients will receive quicker way to check lab results

1.2. Stakeholders and their roles

- Admin: full access to the system.
- Staff: interacts with the customers and users of the application.
- Receptionist: check appointment information
- Doctors: Manage appointments and add data to patient's history
- Staff: Manage users
- Lab Technicians: Manage Lab Procedures appointments and send results
- Pharmacist: Gets prescriptions and manage the delivery of said prescription
- Patient: Uses the application to get appointments and review their medical history

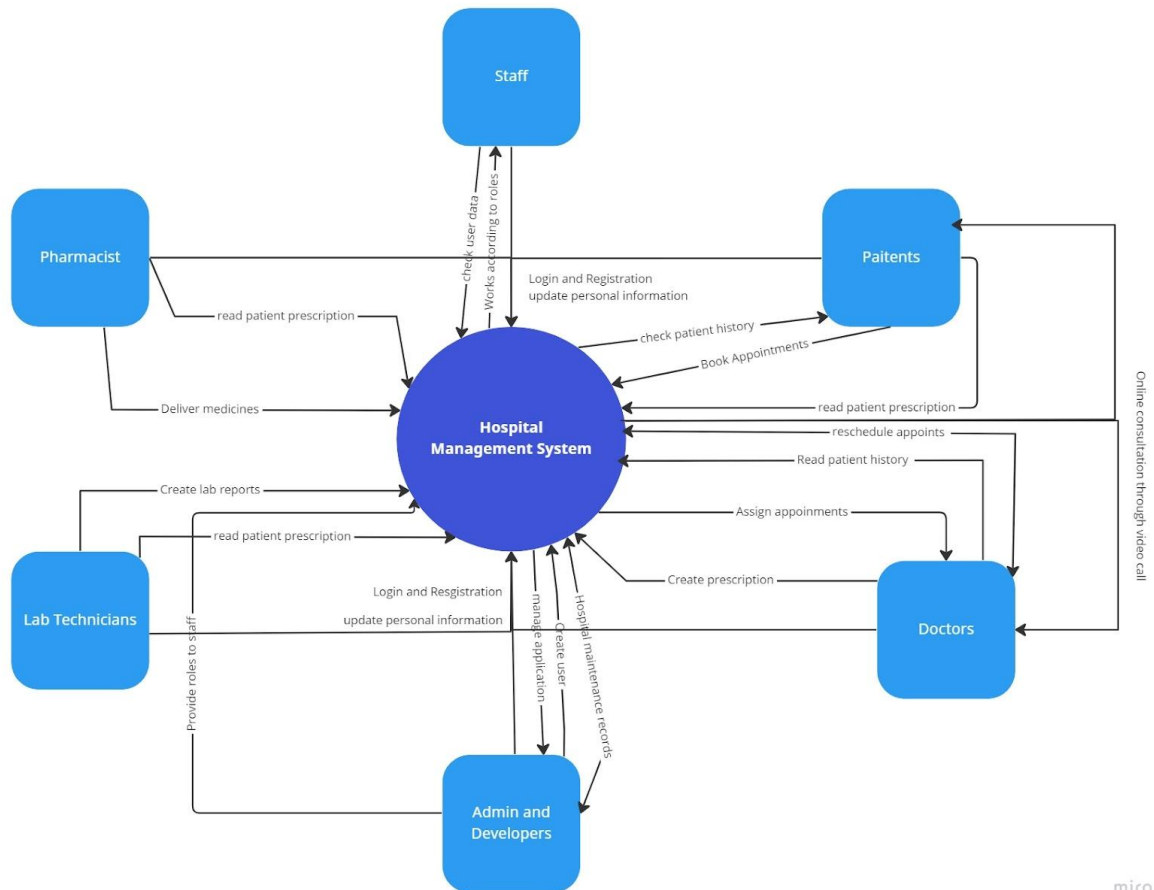
1.3. Subsystems of the application

- Login and Registration Subsystem
- Consultation Subsystem
- Pharmacy Subsystem
- Lab Management Subsystem
- Administration Subsystem

1.4. Intended User of the SRS documentation

- Developers
- UX Designers
- Admin Management
- IT Management

Section 2: Context Flow Diagram



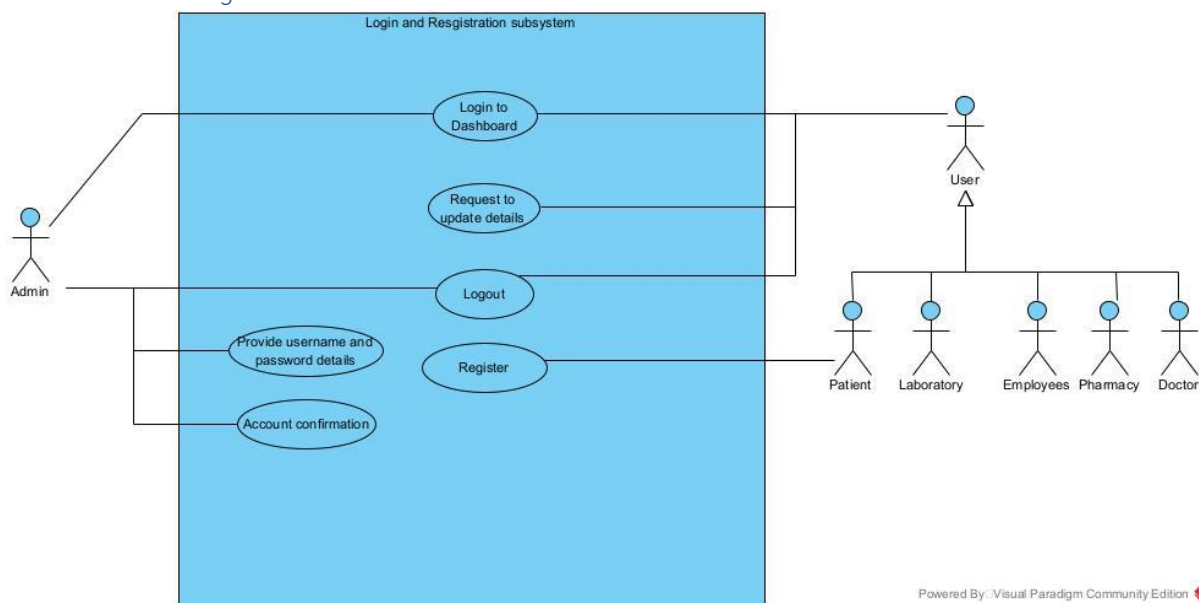
Section 3: Requirements

3.1. Login and Registration Subsystem

3.1.1. Goal Use Cases

#FR	Goal Use case	Role	Description
FR001	Login into Dashboard	All Roles	Login into the corresponding Dashboard
FR002	Change Password	All Roles	Allow all roles to change their current password
FR003	Register	Patient	Allow new patients to create their account
FR004	Create hospital user	IT Department, Admin	Allow the IT and Admin personnel to create new hospital users
FR005	Logout of account	All Roles	Allow the user to logout from the system

3.1.2. Use Case Diagram



3.1.3. User Stories

US-001. AS a Patient, I WANT to be able to create an account by inserting my name, last name, email, phone number and a password SO THAT I can use this account to access the application.

Acceptance Criteria:

- When I click the register link, then the application will redirect me to the registration page.
- When I have inserted all the required fields and pressed the register button, then I will receive a confirmation email and be redirected to the login page.

US-002. AS a User, I WANT to access the application by inputting my email and my password, SO THAT I can access the application.

Acceptance Criteria:

- When I click the login link, the application will redirect me to the login page.
- When I enter the email and password and click the login button, then I will enter to my home page

US-003. AS a System Admin, I WANT to create employee's users, SO THAT this newly created users can interact with the application.

Acceptance Criteria:

- When I click on the add link, The application will redirect me to the add user's page.
- After I enter all the required details and click on add button, the application will give me a confirmation.

US-004. AS a User, I WANT to logout of my account, SO THAT I can maintain my privacy if I leave the computer.

Acceptance Criteria:

- When I click on the logout link, the application will ask for confirmation.
- When I enter the confirmation, then I will be redirected to the login page.

US-005. AS a User, I WANT to change the password of my account, SO THAT I can maintain my data secure.

Acceptance Criteria:

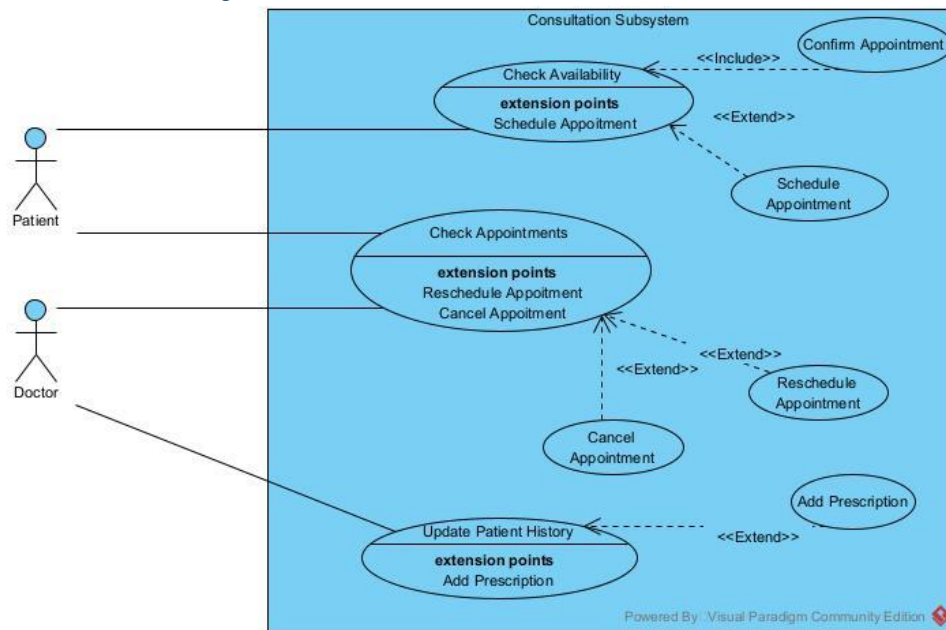
- When I click on the change password link, the application will ask me the current password and new password.
- When I enter the details and click on the change button, the application will give me a confirmation.

3.2. Consultation Subsystem

3.2.1. Goal Use Cases

#FR	Goal Use case	Role	Description
FR006	Check for Availability	Outpatient	Allow the user to search an available date for an appointment
FR007	Online Consultation	Outpatient, Doctor	Allow the patient to consult the doctor using an online video interface.
FR008	Confirm an Appointment	Doctor	Allow the user to confirm an appointment time and date.
FR009	Reschedule an Appointment	Outpatient, Doctor, Admin	Allow the user to change or suggest a time and date change.
FR010	Cancel Appointment	Outpatient, Doctor, Admin	Allow the user to cancel a scheduled appointment
FR011	Send Prescription	Doctor	Allow the user to add a prescription to a patient
FR012	Request Patient's History	Doctor, Outpatient, Inpatient	Allow the user to request a patient's history
FR013	Update Patient's prescription	Doctor	Allow the user to update patient prescription

3.2.2. Use Case Diagram



3.2.3. User Stories

US-006. AS an Outpatient, I WANT to get a list of available physicians of a certain especially, SO THAT I can find a time and date that will suit me for an appointment.

Acceptance Criteria:

- When I enter to the consultation scheduling interface, I can select a specific speciality and a date, if both are correct the application will show me all the available doctors.
- When I entered a speciality and a date were there are not available doctors, the application will show me a prompt that tells me that there are no available doctors on that date.
- When I Select a doctor on the available doctors list, the application will show all the available hours of the doctor.
- When I select an available hour, the application will show a confirmation message, and if successful, it will do the reservation for the consultation.

US-007. AS an Outpatient, I WANT to request an appointment with a physician at a certain date and time, SO THAT I can get an appointment on that date.

Acceptance Criteria:

- When I click on the request link, the application will redirect me to that page.
- When I enter required details and click on the request button, the application will give a confirmation of request

US-008. AS a Doctor, I WANT to confirm an appointment, SO THAT the patient can be notified and come in the specified date and time.

Acceptance Criteria:

- When I click on the confirm button, the application will ask me for confirmation.
- When I click on the confirm button, the application will give me a confirmation

US-009. AS a doctor or an Outpatient, I WANT to request a reschedule of an existing appointment, SO THAT I can move an established appointment in case of an emergency.

Acceptance Criteria:

- When I click on the reschedule link, the application will show me the list of available appointments.
- When I select and confirm my option, the application will give me a confirmation.

US-010. AS a doctor or an Outpatient, I WANT to cancel an existing appointment, SO THAT date can be used by anyone else.

Acceptance Criteria:

- When I click on the cancel link, the application will ask for a confirmation.
- When I confirm it, the application will give me a confirmation.

US-011. AS a Doctor, I WANT to add a prescription to the history of the appointment, SO THAT the patient can use that prescription to get his or her medicines.

Acceptance Criteria:

- When I click on the add link, the application will redirect me to a page where I can add a prescription.
- When I add and click on the save button, the application will give me a confirmation.

US-012. AS a Doctor, I WANT to request a specific patient's history, SO THAT I can read and know what the patient has been taking, what other illnesses he or she has, or any other important information that may be present in the history.

Acceptance Criteria:

- When I Search for a patients name or Id, the basic information of the patient is displayed.
- When I click on the show patient history of a selected patient, it redirects me to a table display of said patient's history.

US-013. AS a Doctor, I WANT to add information to the patient's history, SO THAT anything prescribed or found in the appointment can be used or seen by other doctors and by the patient.

Acceptance Criteria:

- When I click on the add link, the application will redirect me to a page where I can add information.

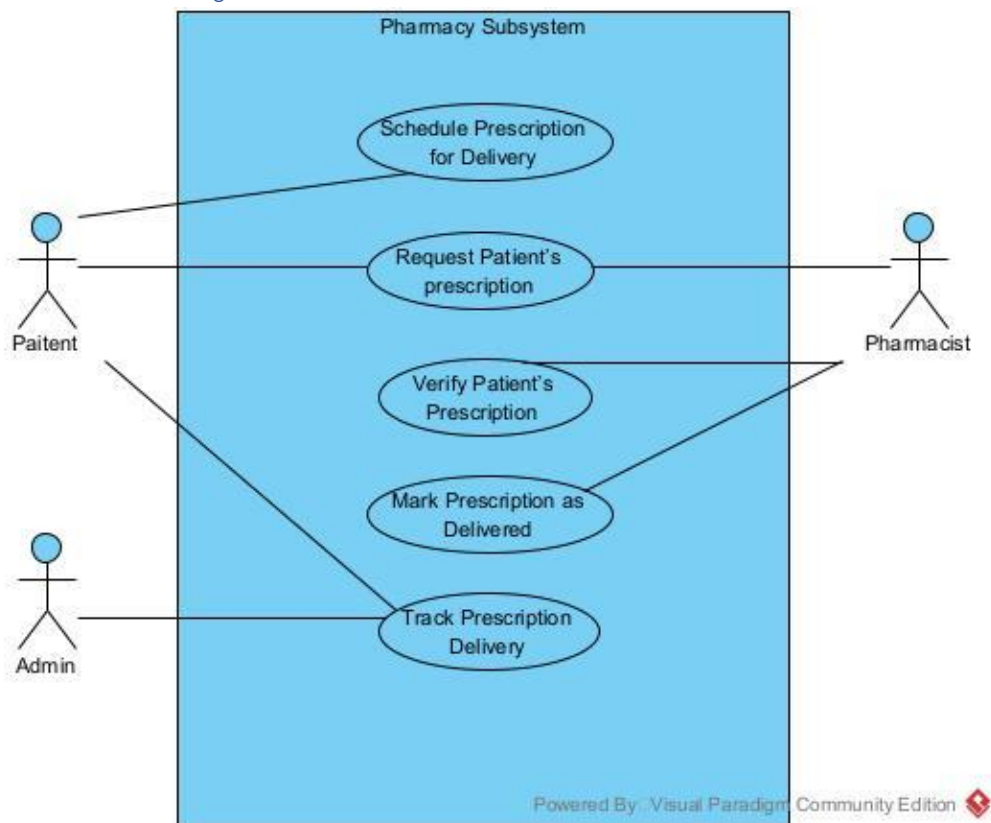
- When I add and click on the save button, the application will give me a confirmation.

3.3. Pharmacy Subsystem

3.3.1. Goal Use Cases

#FR	Goal Use Case	Role	Description
FR014	Request Patient's prescription	Pharmacist, Patient	Allows the user to retrieve a specific patient's prescription
FR015	Verify Patient's Prescription	Pharmacist	Allow the user to verify the information of a prescription
FR016	Mark Prescription as Delivered	Pharmacist	Allow the user to add the approved check to a prescription
FR017	Schedule Prescription for Deliver	Patient	Allow the user to schedule a delivery date for his or her prescription
FR018	Track Prescription Delivery	Patient, Admin	Allow the user to get the status of a prescription delivery

3.3.2. Use Case Diagram



3.3.3. User Stories

US-014. As a Pharmacist, I want to get a prescription from a patient that can easily get by the user and give them medicine accordingly.

Acceptance Criteria:

- When I click on a patient's application then it is attached to their application, and I can see prescription with that application.
- When I click the Done button, it will close the patient prescription and go into the history of the patient's application.

US-015. AS a Pharmacist, I WANT to verify the number, code, doctor or medicines that are in a prescription, SO THAT I can provide the correct medicines to the patient.

Acceptance Criteria:

- When I enter to the verification page, I can input the number code, doctor and medicines that are in the prescription, and the system will show me if it's all fine.

US-016. AS a Pharmacist, I WANT to change the status of a prescription to “delivered”, SO THAT the given prescription can’t be re-claimed.

Acceptance Criteria:

- When I enter the code of a prescription it shows me the verification page.
- When I click on the delivered button on the verification page, the state of that prescription is changed to delivered.

US-017. AS a Patient, I WANT to schedule my prescription for delivery, SO THAT I can receive my medicines on the scheduled date.

Acceptance Criteria:

- When I select a prescription that hasn’t been delivered yet, a scheduling interface appears.
- When I select a day, a confirmation prompt will appear to confirm the delivery date.

US-018. AS a Patient or an Admin, I WANT to check the delivery status of a specific prescription, SO THAT I can know if the delivery is going to take place.

Acceptance Criteria:

- When I enter a prescription Id or select a prescription, I will be redirected to the prescription information page.
- When I am in a prescription information page, I can see the prescription status.

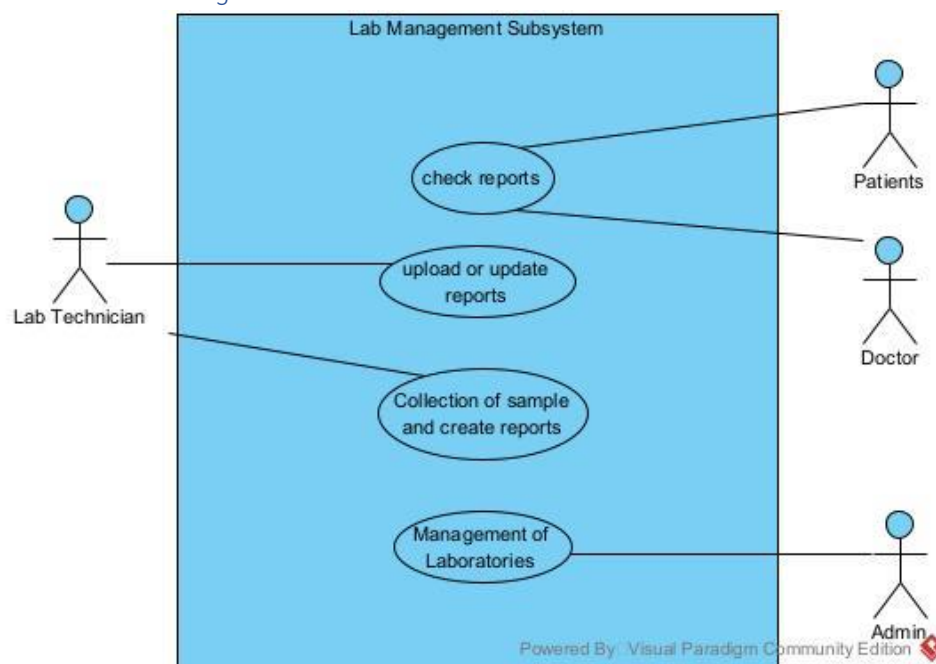
3.4. Lab Management Subsystem

3.4.1. Goal Use Cases

#FR	Goal Use Case	Role	Description
FR19	Upload or update reports	Lab Technician	Lab technician uploads or creates the created reports.

FR20	Management of laboratories	Lab Admin	Admin can manage Laboratory users and patient's lab data for future reference.
FR21	Check Reports	Patients, Doctor	Patients can request for the collection of reports, Also the Lab technician can upload the test reports or can make changes if required.
FR22	Collection of Sample and create reports	Lab Technician	Lab technicians can collect the sample from patients and can create reports to verify with the doctors.

3.4.2. Use Case Diagram



3.4.3. User Stories

US-019. As a lab technician, I can create a patient's report and upload it to and send it to the lab admin after the test is completed.

Acceptance Criteria:

- After the lab test is completed, I can upload it into the lab management system and send it to the lab admin.
- I can request for lab inventory to the lab admin by clicking the request button.

US-20. As a patient or Doctors, I WANT to get reports, SO THAT I can check the values of each test.

Acceptance Criteria:

- When in hub I select the check lab reports, and I will be redirected to a lab reports list page.
- When a doctor clicks on search lab result by patient, I can input the name, id or email of the user to search for a specific lab result of a patient.

US-21. As a lab technician I can create a patients report after a patient sample drop in, SO THAT the data for the lab test can be updated.

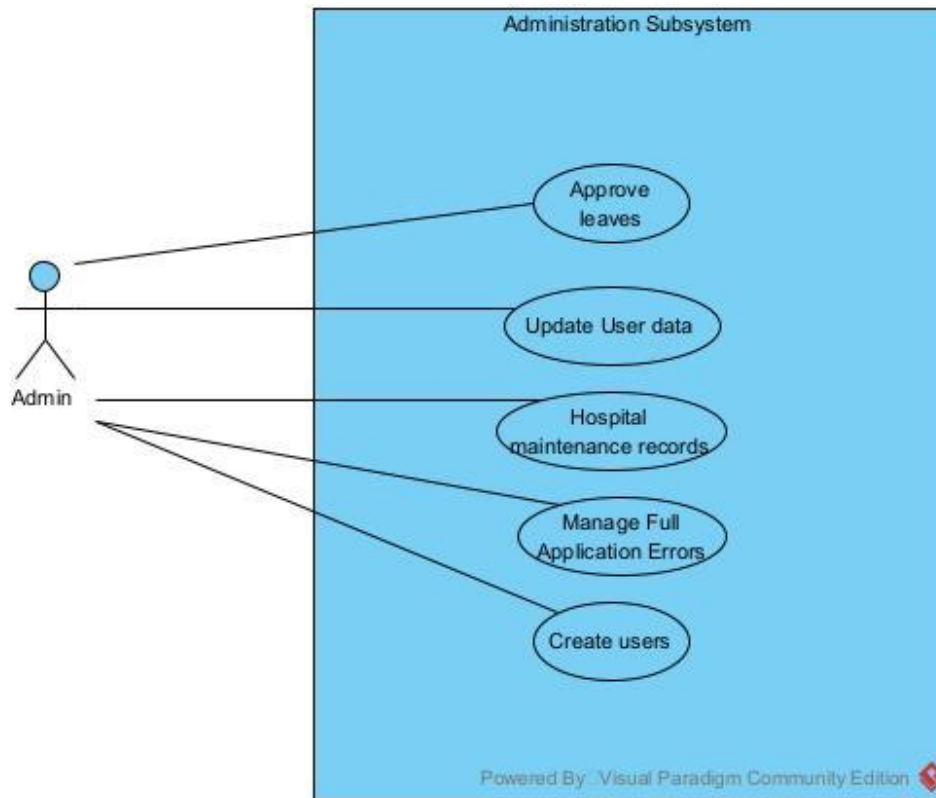
- When I click the edit button, I can add a report or data to a test resquest.

3.5. Administration Subsystem

3.5.1. Goal Use Cases

#FR	Goal Use Case	Role	Description
FR23	Hospital Maintenance Records	Admin	Admin has the records of hospital which can accessed on portal.
FR24	Manage Full Application Errors	Admin	Admin can manage and deal with errors that may occur.
FR25	Update user Data	Admin	Admin can update details of user.
FR26	Create Users	Admin	Create accounts for users like doctors, pharmacists, lab technicians, etc.
FR27	Approve Leaves	Admin	Admin can approve leaves using application.

3.5.2. Use Case Diagram



3.5.3. User Stories

US-022. As Admin or Staff, I want to get the information of patients, doctors, or staff, so that I can verify them.

Acceptance Criteria:

- I can access the User search interface
- When I input a correct ID or username in the search bar, it will show me the corresponding user information page.
- When I input an incorrect ID or username, the interface will show me a message that says: "No valid user with that ID or username".

US-022. AS Admin or Staff, I want to be able to see all the application errors and mark them as solve, or pending, so that everyone can keep track of the errors to be resolved.

Acceptance Criteria:

- When I enter the errors tickets page, the interface will show me a table of un-marked errors.
- When I press the marked as pending error or the marked as solved, the application will change the status of that error.

US-023. AS Admin, I WANT to update user's details, SO THAT I can keep all the information up to date.

Acceptance Criteria:

- When I am in a valid user search information, and it is from a staff member, I can click the update button.
- When I click the update button, I will be redirected to an update information page.

US-024. AS Admin, I WANT to create new users, SO THAT we can have a valid user for a new staff member.

Acceptance Criteria:

- When I want to create a user, I select the create a new user button.
- When I am in the create user page, I can input the data of the specific user that has to be created.
- When I click in create, a prompt will appear telling me that the creation was a success, and I will be redirected to the home page.

US-025. AS Admin or staff, I WANT to approve leaves, SO THAT patients can leave when all the protocols are done.

Acceptance Criteria:

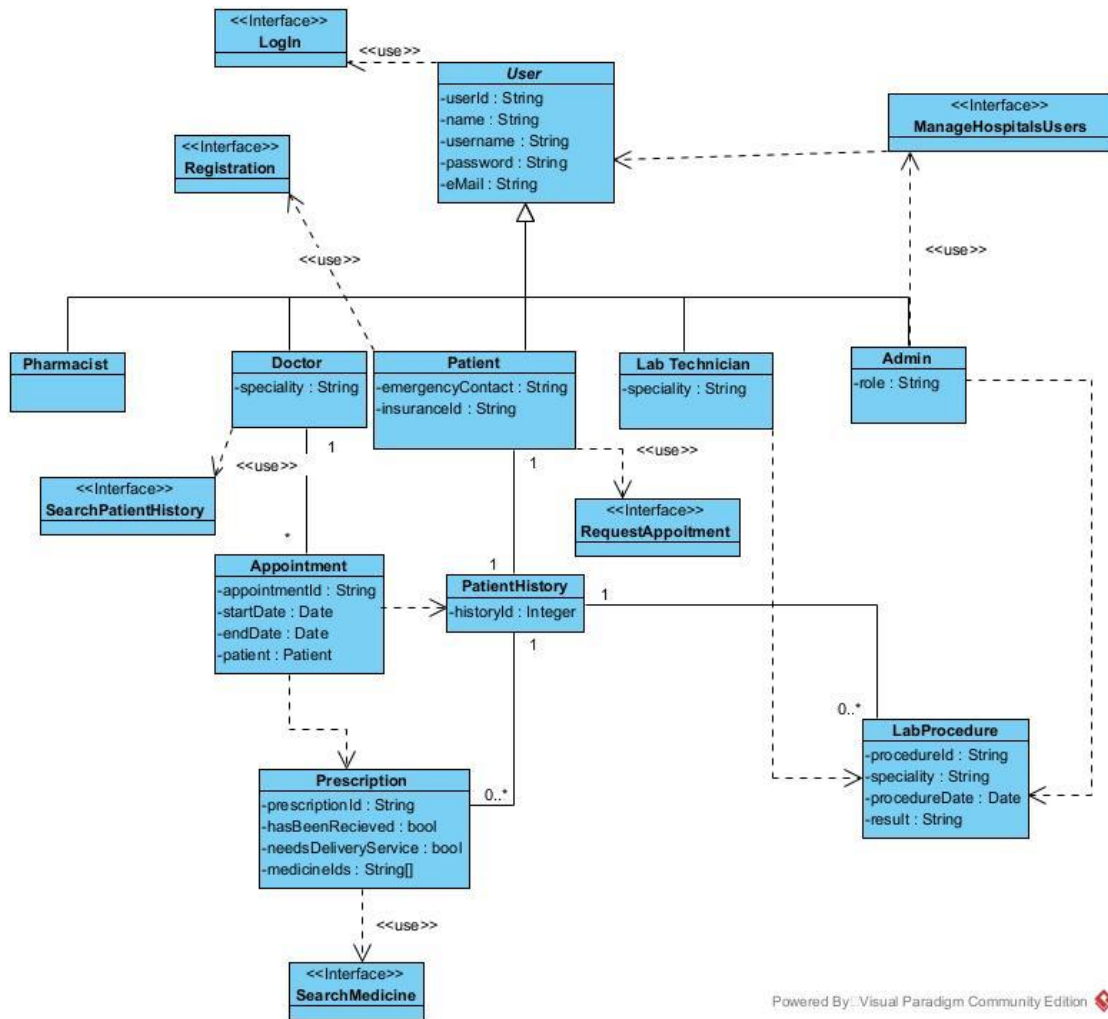
- When I search for a valid inpatient, I can click on the approve leave button and the state of the patient will change.

Section 4: UML Domain Class Diagram

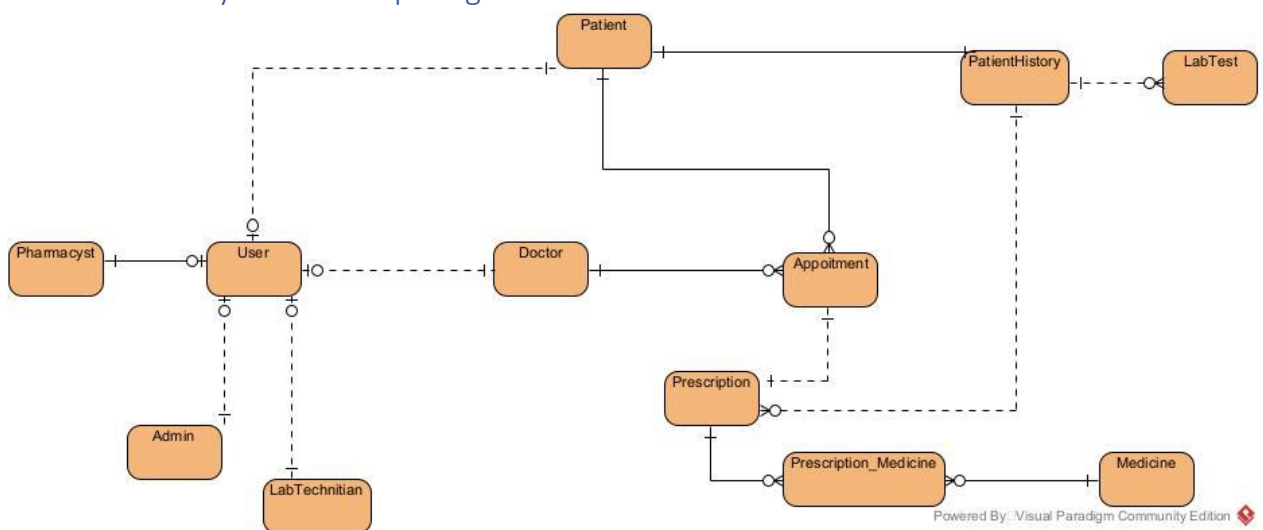
4.1. Classes

- User
- Doctor
- Patient
- Admin
- Patient History
- Prescription
- Pharmacist
- Lab
- Lab Technician
- Lab Procedure
- Appointment

4.2. Domain Class Diagram

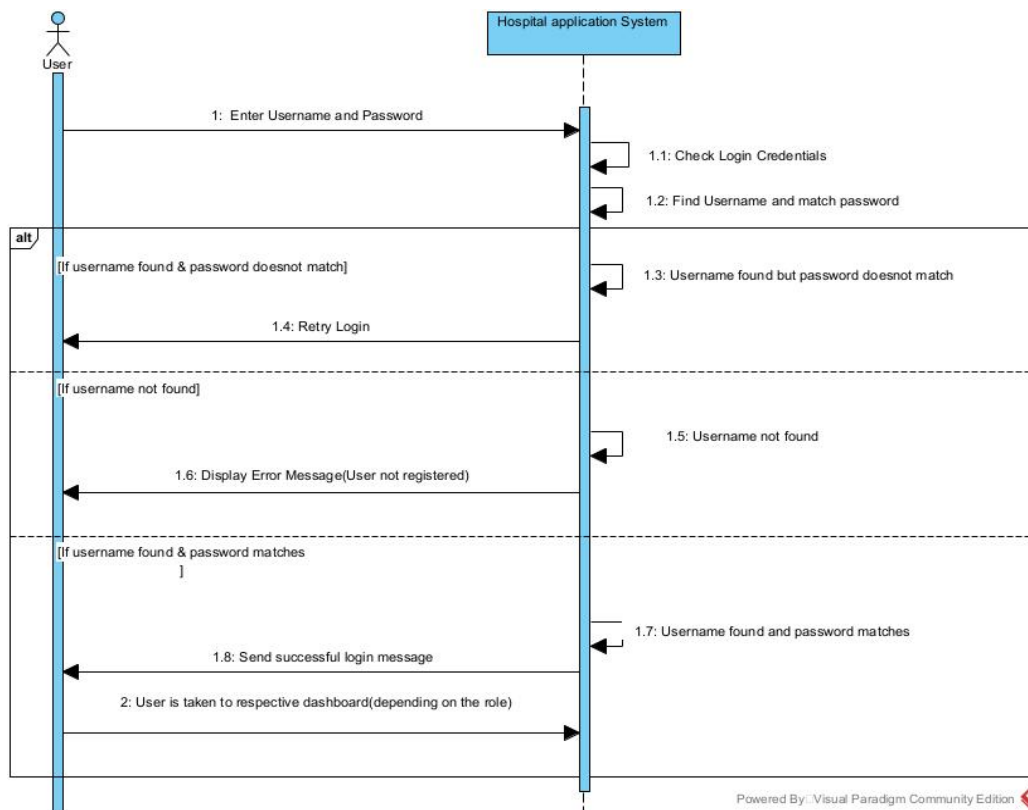


Section 5: Entity Relationship Diagram

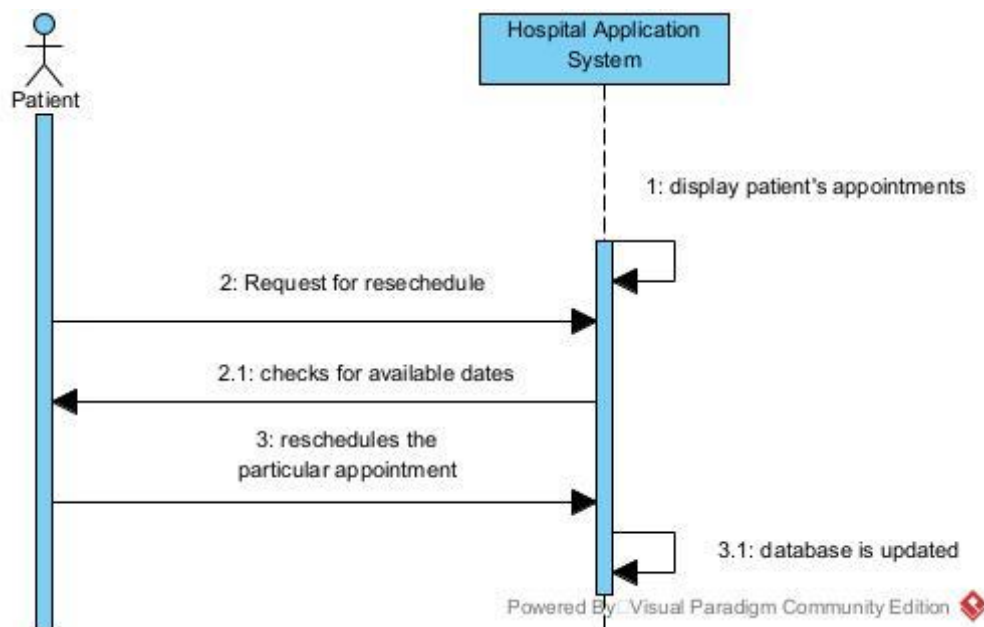


Section 6: System Sequence Diagrams

6.1 Login system sequence diagram

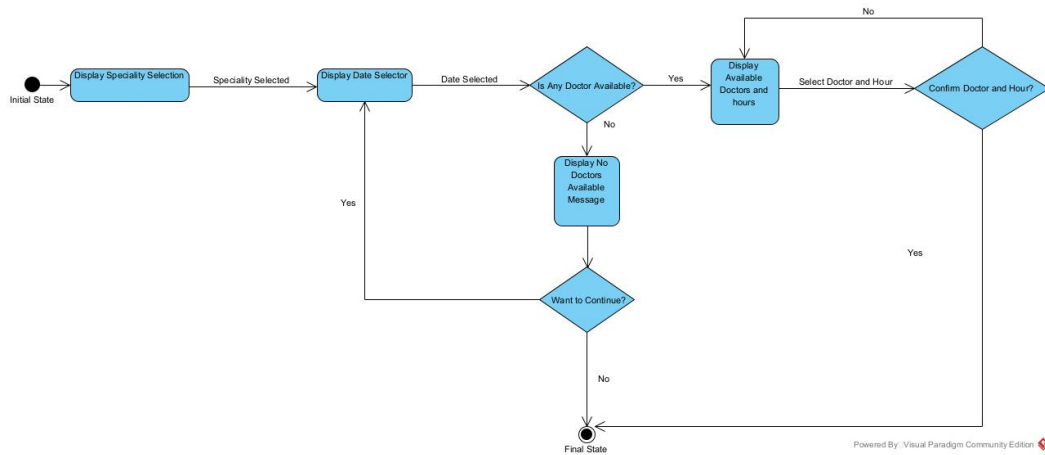


6.2 Reschedule an appointment system sequence diagram

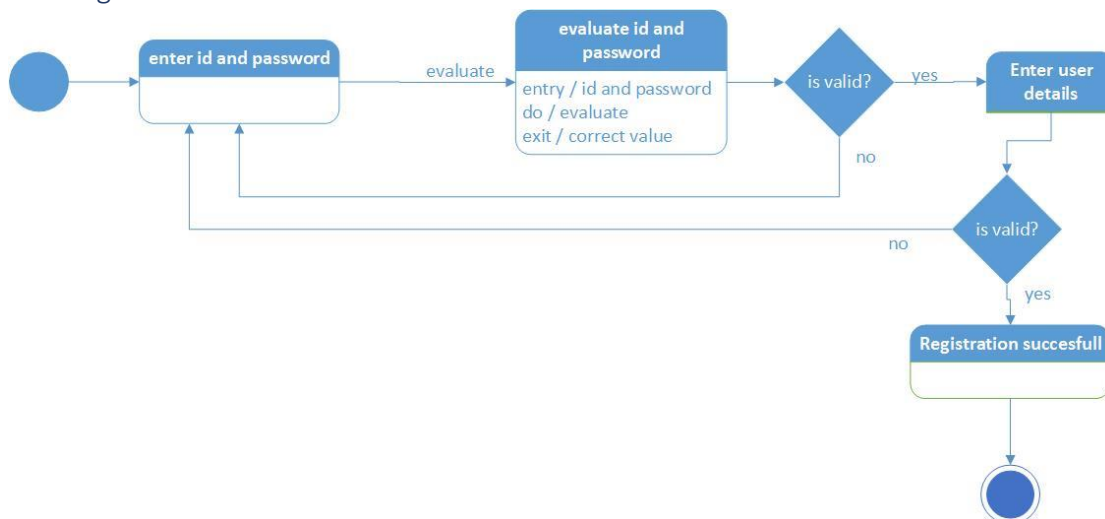


Section 7: State Diagrams

7.1. Consultation – Schedule an Appointment



7.2. Registration



Section 8: Technologies

This system will be implemented as a Web Application, that in the long run will be converted to a desktop application using Electron or Tauri.

8.1. Client Side:

- Angular
- Typescript
- NodeJS

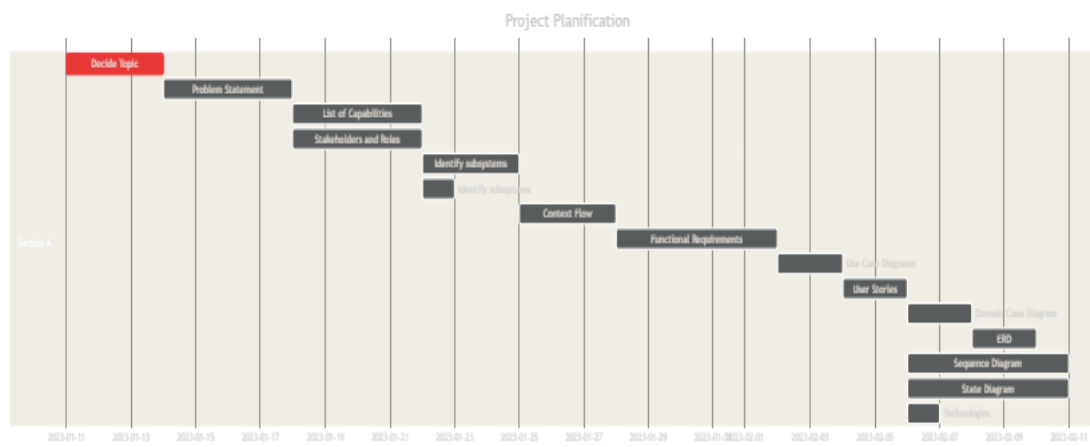
8.2. Business Logic

- Java
- Spring Boot

8.3. Data Side

- MySQL
- Redis

Section 9: Project Management – Gantt Diagram



Part B: Software Design Architecture

Section 1: Requirements Edits to Part A

1.1 Corrections on Problem Statement (Section 1.1a)

- Corrected the grammatical mistakes in 1.1 a

1.2 Stakeholders and their roles (Section 1.2)

- Removed hospital management employees
- Removed IT management and receptionist
- Removed the types of patients
- Made an order of the stakeholders

1.3 Use Case Diagram (Section 3.2.2)

- Patients can check the availability and confirm or schedule the appointment
- Doctors can check the appointment and cancel or reschedule it.
- Doctors can update the patient's history and add prescription.

1.4 Use Case Diagram (Section 3.3.2)

- Corrected the grammatical mistakes

1.5 Use Case Diagram (Section 3.5.2)

- Removed staff
- Removed management of staff data
- Corrected check user data to update user data

1.6 Domain Class Diagram (Section 4.2)

- has insurance removed from the patient

1.6 Domain ER Diagram (Section 5)

- Diagram has been redrawn.

1.7 Domain ER Diagram (Section 6.2)

- Removed the doctor
- Doctors reschedules has been removed

1.8 Project Management - Gantt Diagram (Section 9)

- Redrew the diagram without cuts

Section 2: Overview Model

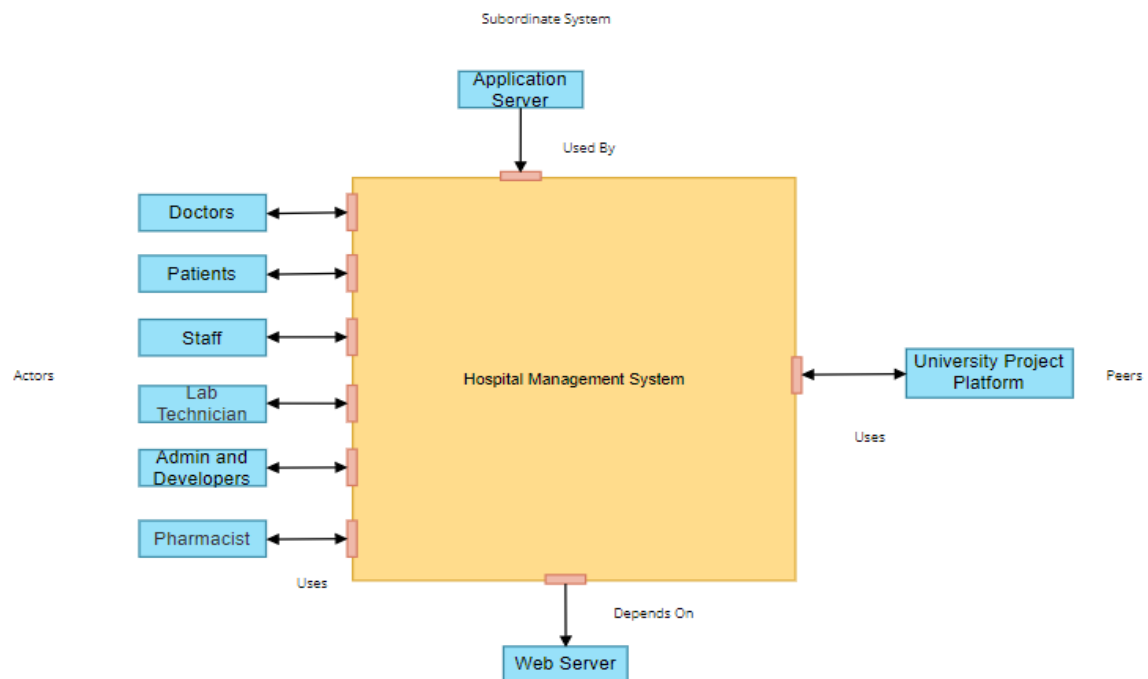
2.1. Intended users of the SDD document

The purpose of the System Design Document (SDD) is to provide a complete description of the system design so that software developers can have a full understanding of what needs to be built. Therefore, the intended audience for this document is the developers and managers. They are responsible for the system development.

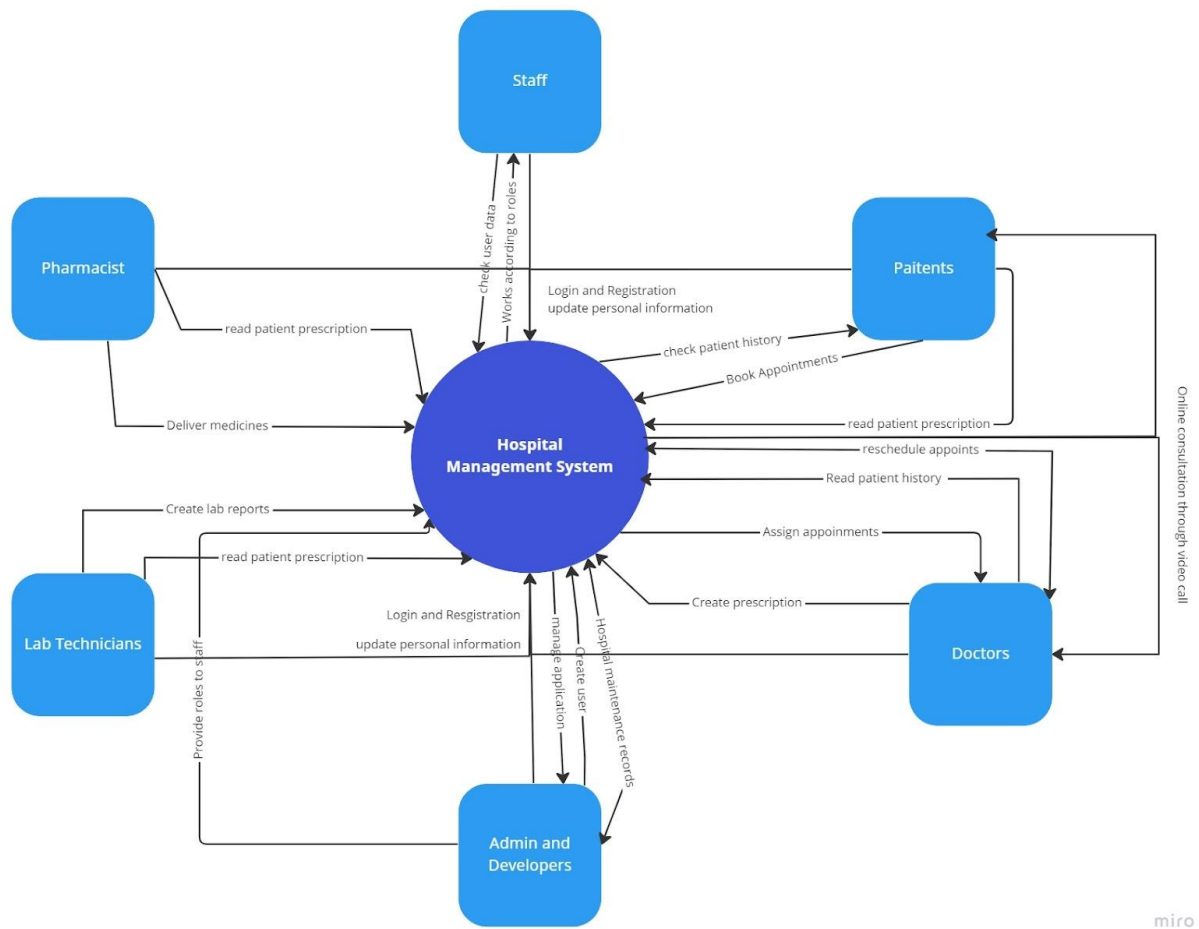
- Developers
- Managers
- UI/UX Designers
- Product Owner
- Scrum Master

2.2. Architectural Context diagram (ACD) versus Context Flow Diagram (CFD)

2.2.1 Architectural Context Diagram (HOW Overview)



2.2.2 Architectural Context Diagram (WHAT Overview)

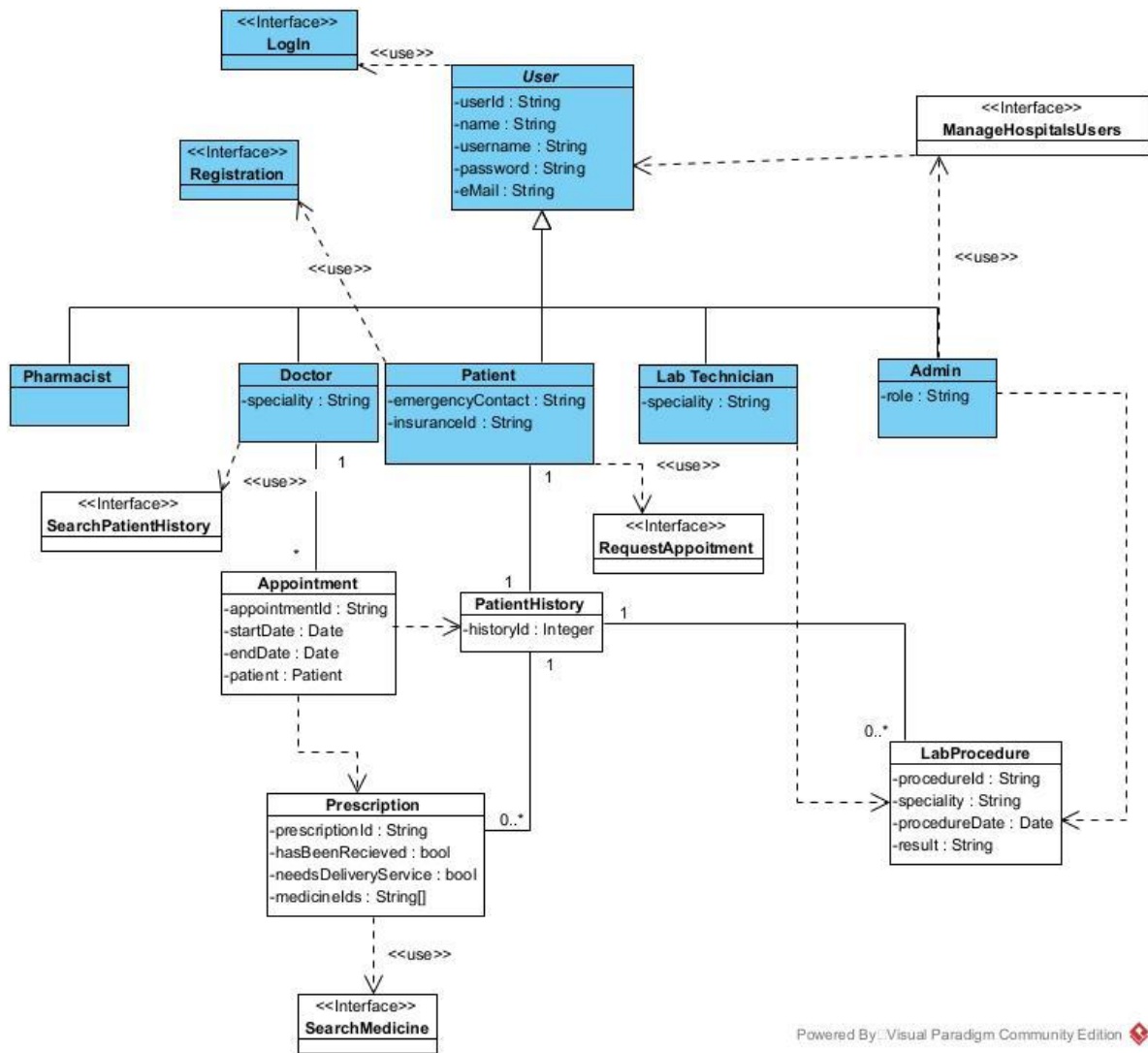


Section 3: Modularization

3.1. Partition the analysis model

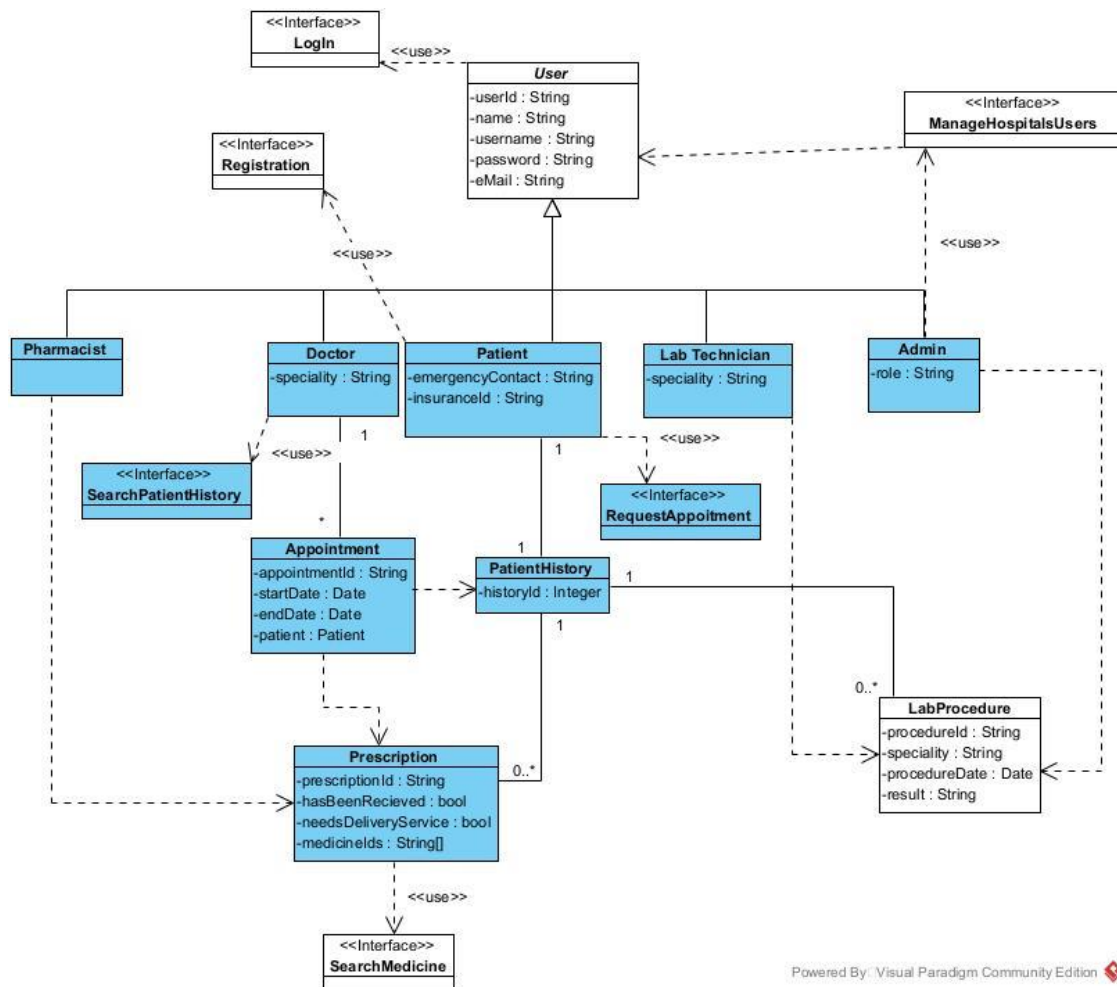
3.1.1. Login and Registration Subsystem

- Login Interface
- Registration Interface
- User
- Pharmacist
- Doctor
- Patient
- Lab Technician
- Admin

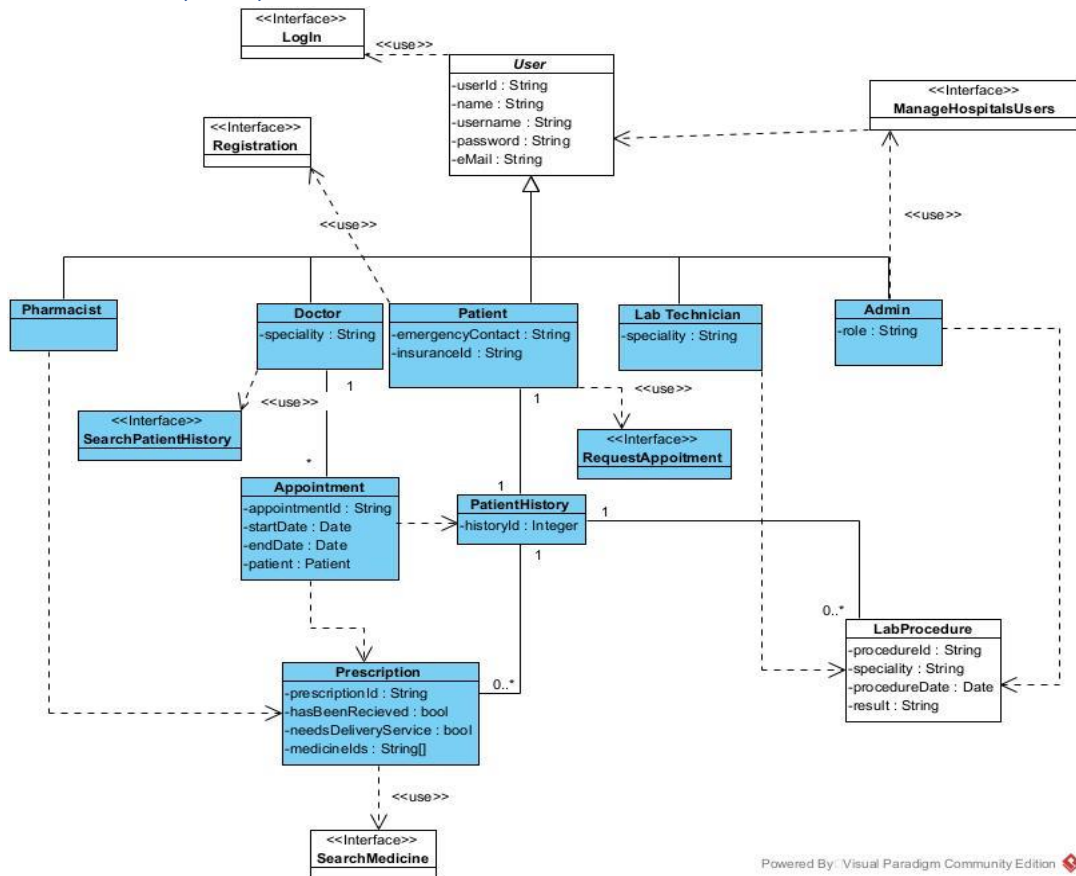


3.1.2. Consultation Subsystem

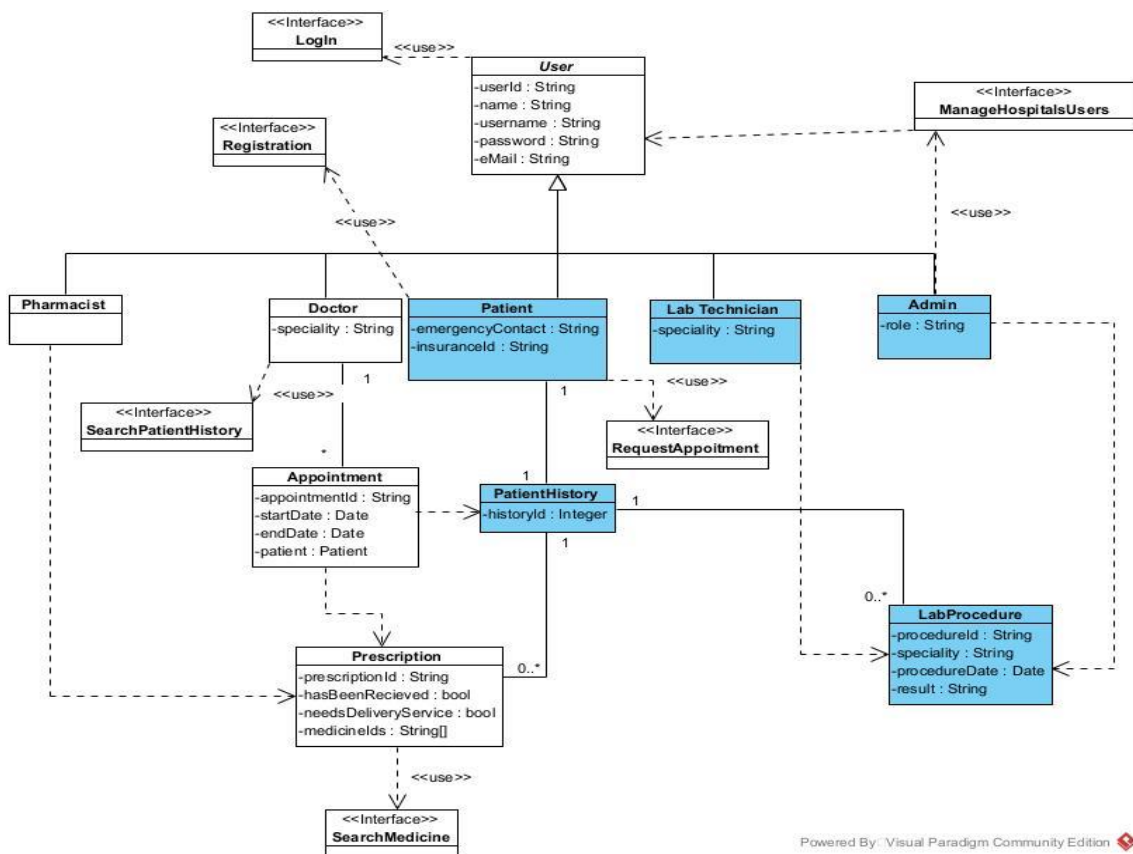
- Patient
- Doctor
- Search Patient History Interface
- Appointment
- Patient History
- Request Appointment Interface
- Prescription



3.1.3. Pharmacy Subsystem



1.4. Lab Management Subsystem



3.2 Class Responsibility Collaboration (CRC)

3.2.1 Login and Registration Subsystem

Login Interface Super Classes: Sub Classes: User Description: Provides login functionalities to the users Attributes: Name Description Responsibilities: Name Collaborator Login into your account Logout of your account specify user role Change the Password Change the UserID	Registration Interface Super Classes: Sub Classes: Patient Description: Allows Registration functionality to patient class Attributes: Name Description Responsibilities: Name Collaborator Fetch the UserDetails Confirm the Account Change UserDetails	User Super Classes: Sub Classes: Doctor, Patient, Admin, Lab Technician, Pharmacist Description: identifies the user and provides user information to login Attributes: Name Description userid username password password email email of user Responsibilities: Name Collaborator Get the Username Change the Username Fetch Email Address Change the Email Update the Details	Admin Super Classes: User Sub Classes: Description: creates Admin object and its functionalities Attributes: Name Description role role of the administration Responsibilities: Name Collaborator Provide UserID And Validate the User Details Specify the RoleOfAdmin	Pharmacist Super Classes: User Sub Classes: Description: creates Pharmacist object and its functionalities Attributes: Name Description Responsibilities: Name Collaborator	Lab Technician Super Classes: User Sub Classes: Description: create Lab Technician object and its functionalities Attributes: Name Description specialty the specialty of lab Responsibilities: Name Collaborator
--	---	---	--	---	--

Powered By Visual Paradigm Community Edition

3.2.2 Consultation Subsystem

Doctor Super Classes: User Sub Classes: Description: creates Doctor object and its functionalities Attributes: Name Description specialty gives the specialty of Doctor Responsibilities: Name Collaborator Receive the Appointment	PatientHistory Super Classes: Sub Classes: Description: allows to interact with database to store and receive patient history Attributes: Name Description historyId gives the Id for patient history Responsibilities: Name Collaborator Add the Prescription	RequestAppointment Interface Super Classes: Sub Classes: Appointment Description: gives functionality to patient and doctor through appointment class Attributes: Name Description Responsibilities: Name Collaborator Reschedule the Appointment Confirm the Appointment Schedule the Appointment Cancel the Appointment	Patient Super Classes: User Sub Classes: Description: creates Patient object and its functionalities Attributes: Name Description emergencyContact contact of other person insuranceId insurance number Responsibilities: Name Collaborator Get AppointmentDetails Get the Prescription	Appointment Super Classes: Sub Classes: Description: creates appointment and its details Attributes: Name Description appointmentId id of appointment startDate endDate patient Responsibilities: Name Collaborator Provide AppointmentDetails receive PatientHistory Get the PatientDetails	SearchPatientHistory Interface Super Classes: Sub Classes: Patient History, Doctor Description: functionalities to doctor to access patient history Attributes: Name Description Responsibilities: Name Collaborator Search for PatientHistory Update PatientHistory Create the PatientHistory	Prescription Super Classes: Sub Classes: Description: creates prescription according to appointment Attributes: Name Description prescriptionId id of prescription hasBeenReceived boolean value if medicines are received or not needsDeliveryService boolean value if needs delivery or not medicineIds collection of required medicines of Patient Responsibilities: Name Collaborator Specify the Appointment Details
---	--	---	---	---	---	--

Powered By Visual Paradigm Community Edition

3.2.3 Pharmacy Subsystem

Pharmacist Super Classes: User Sub Classes: Description: creates Pharmacist object and its functionalities Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Select Delivery Service</td><td>Prescription</td></tr> <tr> <td>Get the Patient Prescription</td><td>Patient</td></tr> <tr> <td>Get Medicine Ids</td><td>Prescription</td></tr> <tr> <td>Specify if delivered</td><td>Prescription</td></tr> <tr> <td>Verify the Patient Prescription</td><td>PatientHistory</td></tr> <tr> <td>Track the Deliver</td><td></td></tr> </table>		Name	Description	Name	Collaborator	Select Delivery Service	Prescription	Get the Patient Prescription	Patient	Get Medicine Ids	Prescription	Specify if delivered	Prescription	Verify the Patient Prescription	PatientHistory	Track the Deliver	
Name	Description																
Name	Collaborator																
Select Delivery Service	Prescription																
Get the Patient Prescription	Patient																
Get Medicine Ids	Prescription																
Specify if delivered	Prescription																
Verify the Patient Prescription	PatientHistory																
Track the Deliver																	
SearchMedicine Interface Super Classes: Sub Classes: Description: allows pharmacist to check inventory Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Search for Medicine</td><td>Pharmacist</td></tr> <tr> <td>Add the Medicine</td><td>Pharmacist</td></tr> <tr> <td>Remove the Medicine</td><td>Pharmacist</td></tr> <tr> <td>Update the Medicine</td><td>Pharmacist</td></tr> </table>		Name	Description	Name	Collaborator	Search for Medicine	Pharmacist	Add the Medicine	Pharmacist	Remove the Medicine	Pharmacist	Update the Medicine	Pharmacist				
Name	Description																
Name	Collaborator																
Search for Medicine	Pharmacist																
Add the Medicine	Pharmacist																
Remove the Medicine	Pharmacist																
Update the Medicine	Pharmacist																
Patient Super Classes: User Sub Classes: Description: creates Patient object and its functionalities Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>emergencyContact</td><td>contact of other person</td></tr> <tr> <td>insuranceId</td><td>insurance number</td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Request for Delivery</td><td>Pharmacist</td></tr> <tr> <td>Provide the Prescription</td><td>PatientHistory</td></tr> <tr> <td>Request For Medicines</td><td>Pharmacist</td></tr> </table>		Name	Description	emergencyContact	contact of other person	insuranceId	insurance number	Name	Collaborator	Request for Delivery	Pharmacist	Provide the Prescription	PatientHistory	Request For Medicines	Pharmacist		
Name	Description																
emergencyContact	contact of other person																
insuranceId	insurance number																
Name	Collaborator																
Request for Delivery	Pharmacist																
Provide the Prescription	PatientHistory																
Request For Medicines	Pharmacist																
PatientHistory Super Classes: Sub Classes: Description: allows to interact with database to store and receive patient history Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>historyId</td><td></td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Get the PatientHistory</td><td></td></tr> </table>		Name	Description	historyId		Name	Collaborator	Get the PatientHistory									
Name	Description																
historyId																	
Name	Collaborator																
Get the PatientHistory																	
Prescription Super Classes: Sub Classes: Description: creates prescription according to appointment Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>prescriptionId</td><td>id of prescription</td></tr> <tr> <td>hasBeenReceived</td><td>boolean value if medicines are received or</td></tr> <tr> <td>needsDeliveryService</td><td>boolean value if needs delivery or not</td></tr> <tr> <td>medicinesIds</td><td>collection of medicines</td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> </table>		Name	Description	prescriptionId	id of prescription	hasBeenReceived	boolean value if medicines are received or	needsDeliveryService	boolean value if needs delivery or not	medicinesIds	collection of medicines	Name	Collaborator				
Name	Description																
prescriptionId	id of prescription																
hasBeenReceived	boolean value if medicines are received or																
needsDeliveryService	boolean value if needs delivery or not																
medicinesIds	collection of medicines																
Name	Collaborator																

Powered By Visual Paradigm Community Edition

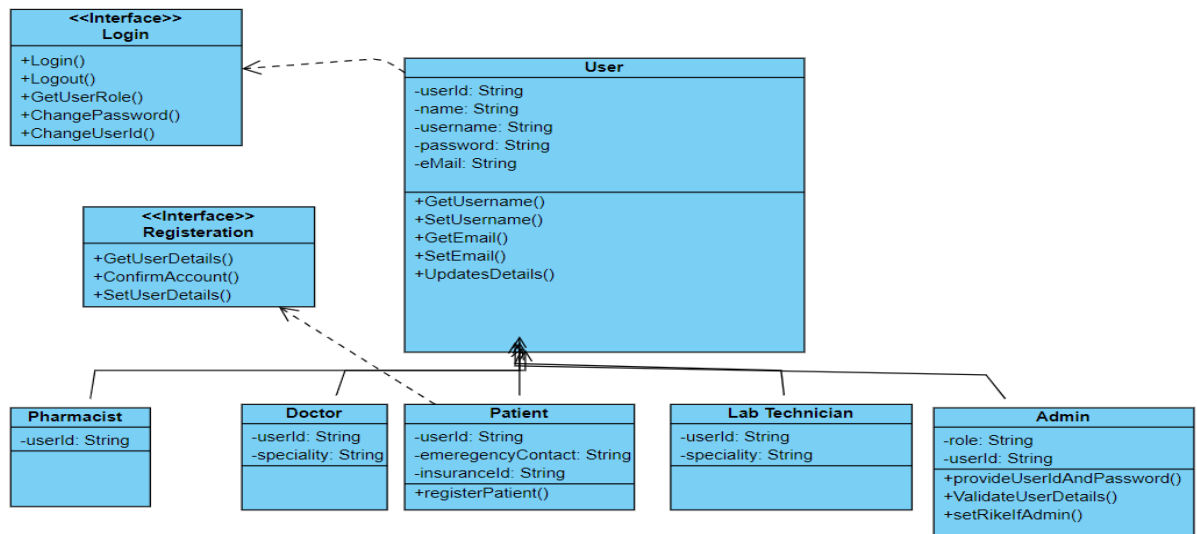
3.2.4 Lab Management Subsystem

Patient Super Classes: User Sub Classes: Description: creates Patient object and its functionalities Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>emergencyContact</td><td>contact of other person</td></tr> <tr> <td>insuranceId</td><td>insurance number</td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Get the LabProcedure</td><td>PatientHistory</td></tr> </table>		Name	Description	emergencyContact	contact of other person	insuranceId	insurance number	Name	Collaborator	Get the LabProcedure	PatientHistory						
Name	Description																
emergencyContact	contact of other person																
insuranceId	insurance number																
Name	Collaborator																
Get the LabProcedure	PatientHistory																
LabProcedure Super Classes: Sub Classes: Description: creates a lab procedure object and its details which is later stored in patient history Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>procedureId</td><td>id of lab report</td></tr> <tr> <td>specialty</td><td>particular type of lab report</td></tr> <tr> <td>procedureDate</td><td>date of procedure produced</td></tr> <tr> <td>result</td><td>result of procedure</td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Add the LabProcedure Details</td><td></td></tr> </table>		Name	Description	procedureId	id of lab report	specialty	particular type of lab report	procedureDate	date of procedure produced	result	result of procedure	Name	Collaborator	Add the LabProcedure Details			
Name	Description																
procedureId	id of lab report																
specialty	particular type of lab report																
procedureDate	date of procedure produced																
result	result of procedure																
Name	Collaborator																
Add the LabProcedure Details																	
PatientHistory Super Classes: Sub Classes: Description: allows to interact with database to store and receive patient history Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>patientHistory</td><td></td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Add the LabProcedure</td><td>Lab Procedure</td></tr> <tr> <td>Provide the LabProcedure Detials</td><td>Patient</td></tr> </table>		Name	Description	patientHistory		Name	Collaborator	Add the LabProcedure	Lab Procedure	Provide the LabProcedure Detials	Patient						
Name	Description																
patientHistory																	
Name	Collaborator																
Add the LabProcedure	Lab Procedure																
Provide the LabProcedure Detials	Patient																
Lab Technician Super Classes: User Sub Classes: Description: creates Lab Technician object and its functionalities Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> <tr> <td>specialty</td><td>the specialty of lab</td></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Create LabProcedure</td><td>LabProcedure</td></tr> <tr> <td>Update the LabProcedure</td><td>LabProcedure</td></tr> <tr> <td>Collect the Samples</td><td></td></tr> <tr> <td>Update the Samples</td><td></td></tr> <tr> <td>Upload the LabProcedure</td><td></td></tr> </table>		Name	Description	specialty	the specialty of lab	Name	Collaborator	Create LabProcedure	LabProcedure	Update the LabProcedure	LabProcedure	Collect the Samples		Update the Samples		Upload the LabProcedure	
Name	Description																
specialty	the specialty of lab																
Name	Collaborator																
Create LabProcedure	LabProcedure																
Update the LabProcedure	LabProcedure																
Collect the Samples																	
Update the Samples																	
Upload the LabProcedure																	
Admin Super Classes: User Sub Classes: Description: creates Admin object and its functionalities Attributes: <table> <tr> <th>Name</th><th>Description</th></tr> </table> Responsibilities: <table> <tr> <th>Name</th><th>Collaborator</th></tr> <tr> <td>Manage Lab Technicians</td><td>Lab Technicians</td></tr> <tr> <td>Manage PatientForLabEntry</td><td>Patients</td></tr> </table>		Name	Description	Name	Collaborator	Manage Lab Technicians	Lab Technicians	Manage PatientForLabEntry	Patients								
Name	Description																
Name	Collaborator																
Manage Lab Technicians	Lab Technicians																
Manage PatientForLabEntry	Patients																

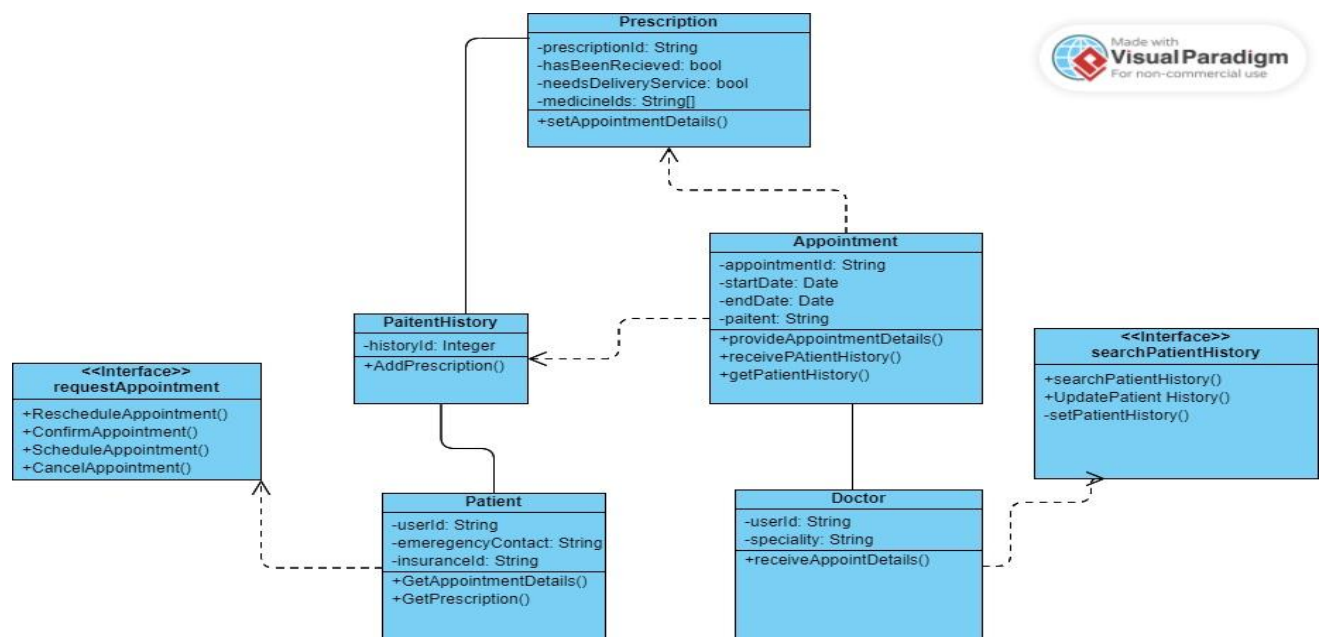
Powered By Visual Paradigm Community Edition

3.3 Design Classes diagrams

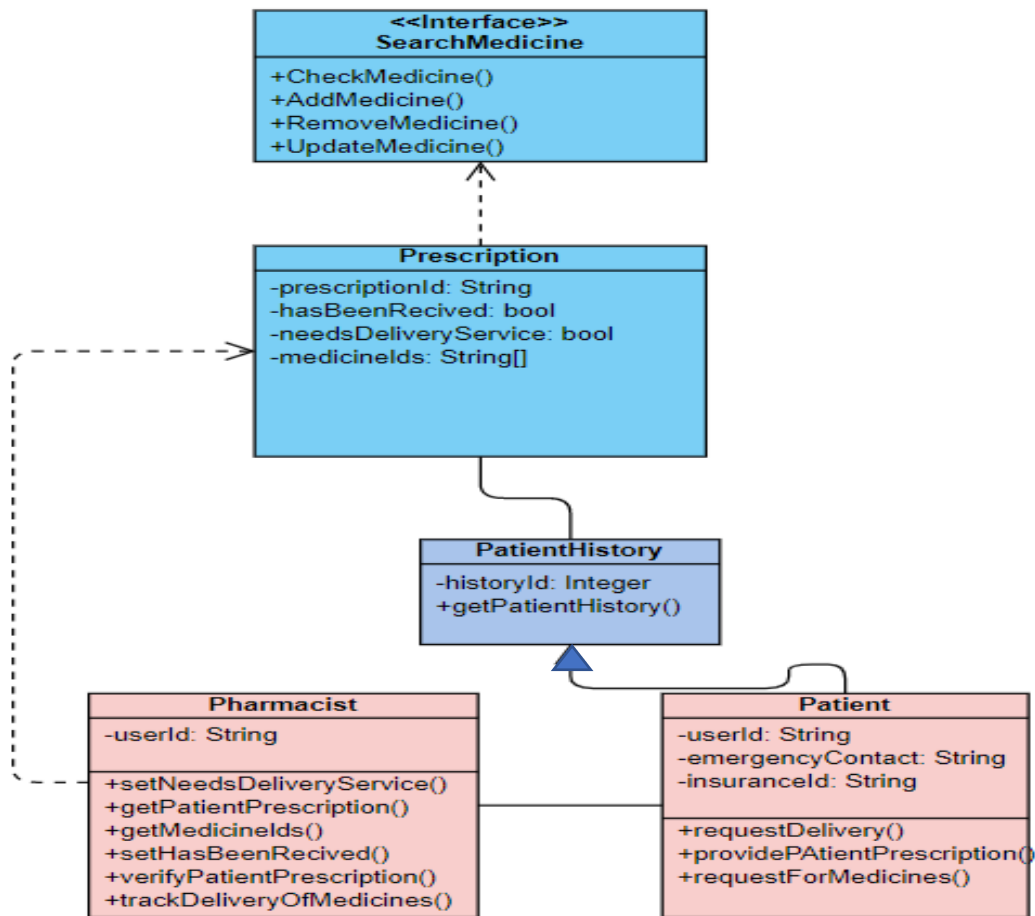
3.3.1 Login and Registration Subsystem



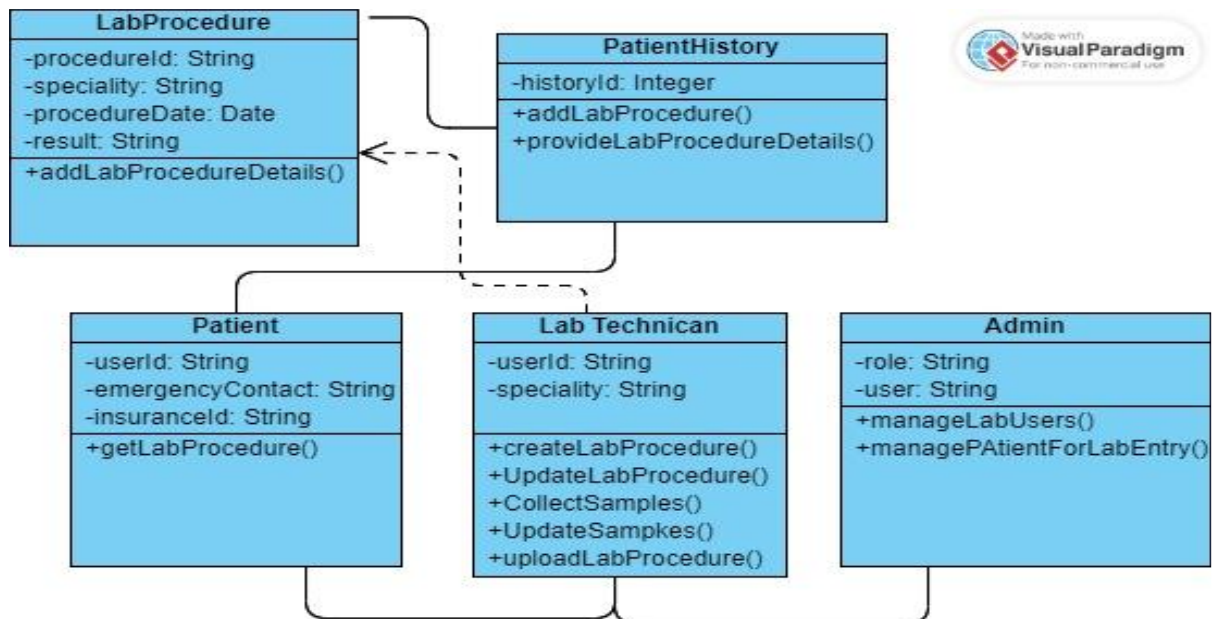
3.3.2 Consultation Subsystem



3.3.3 Pharmacy Subsystem



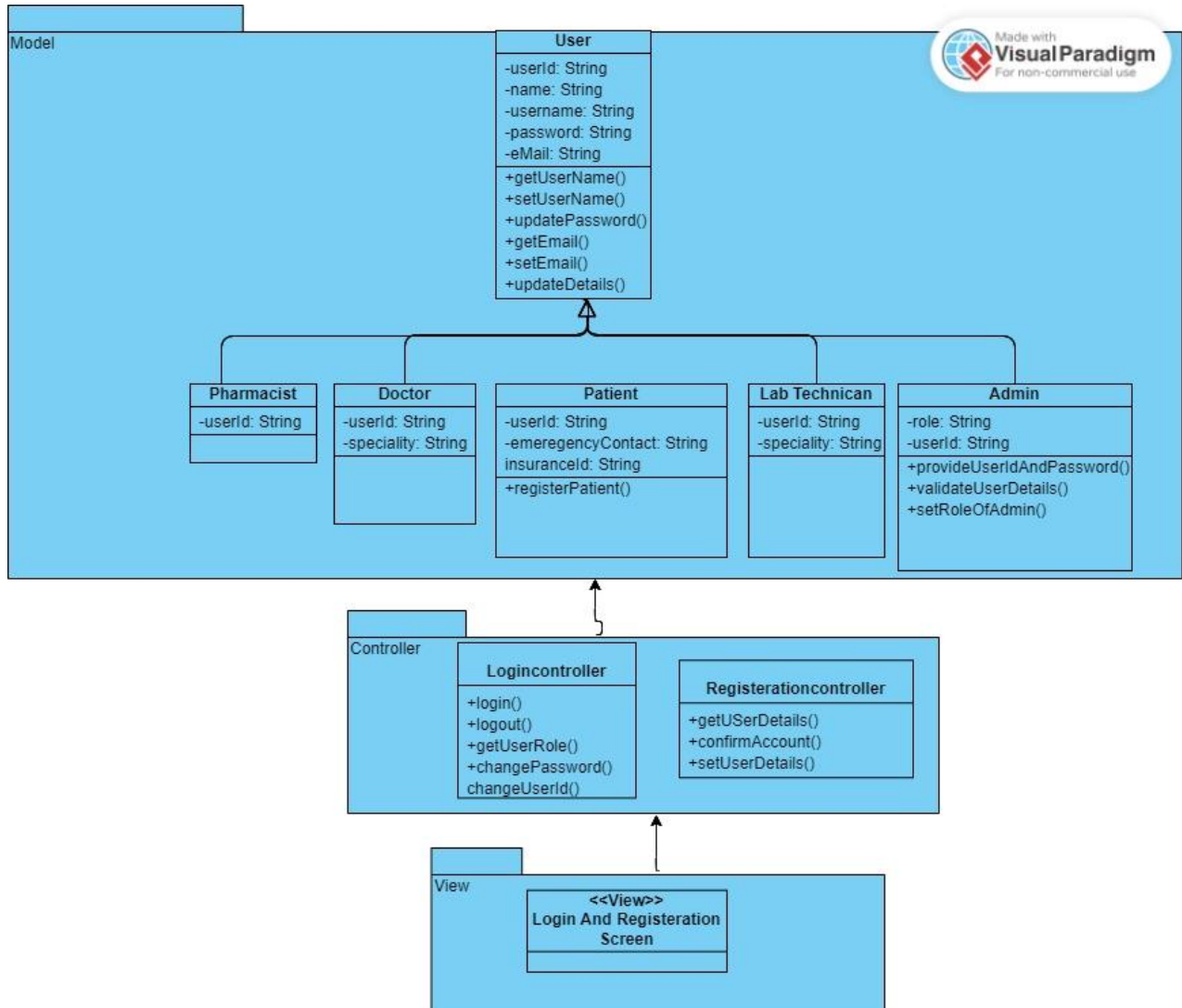
3.4.4 Lab Management Subsystem



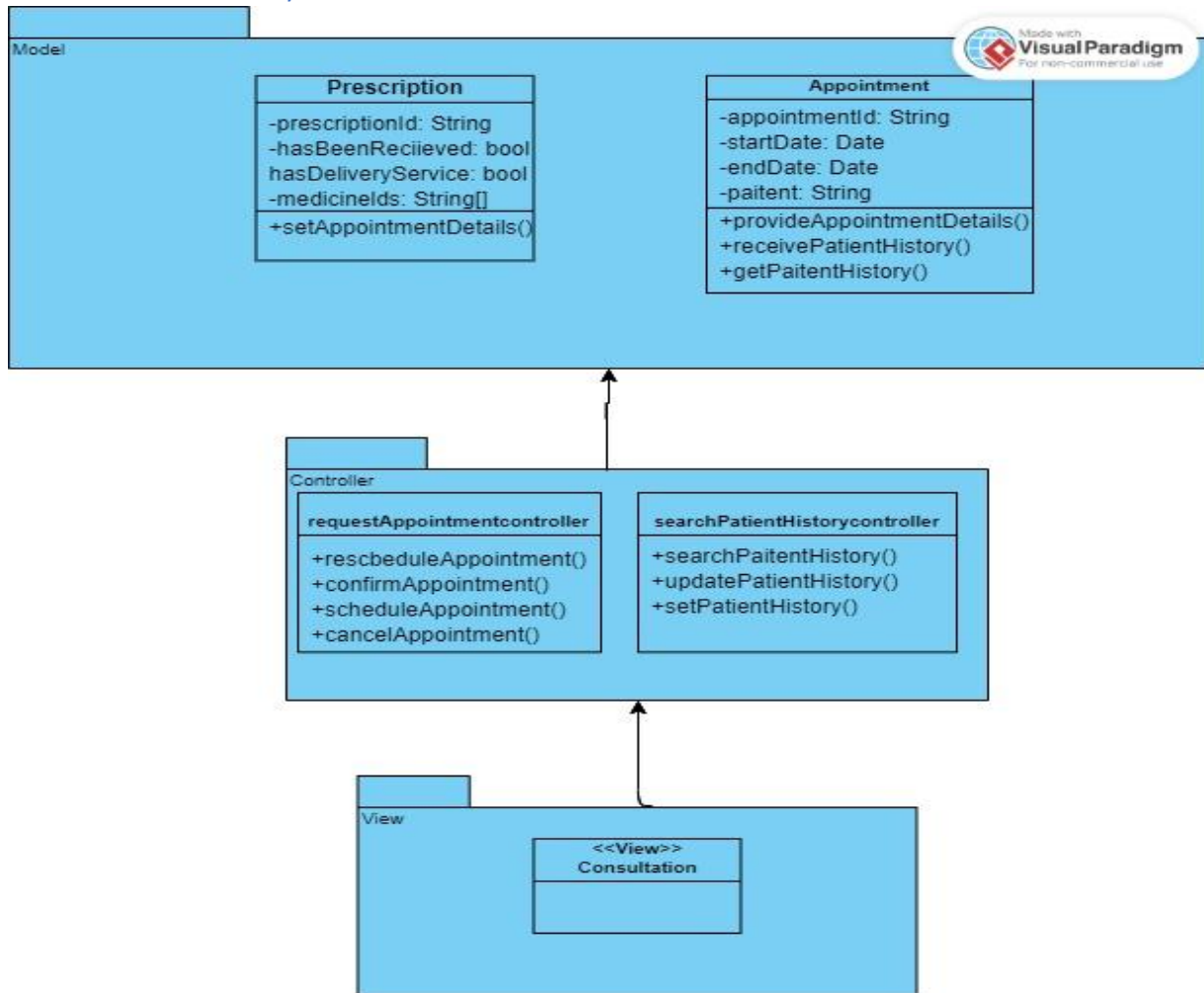
Section 4: Framework M(odel) V(iew) C(ontroller)

4.1 MVC pattern diagram to include

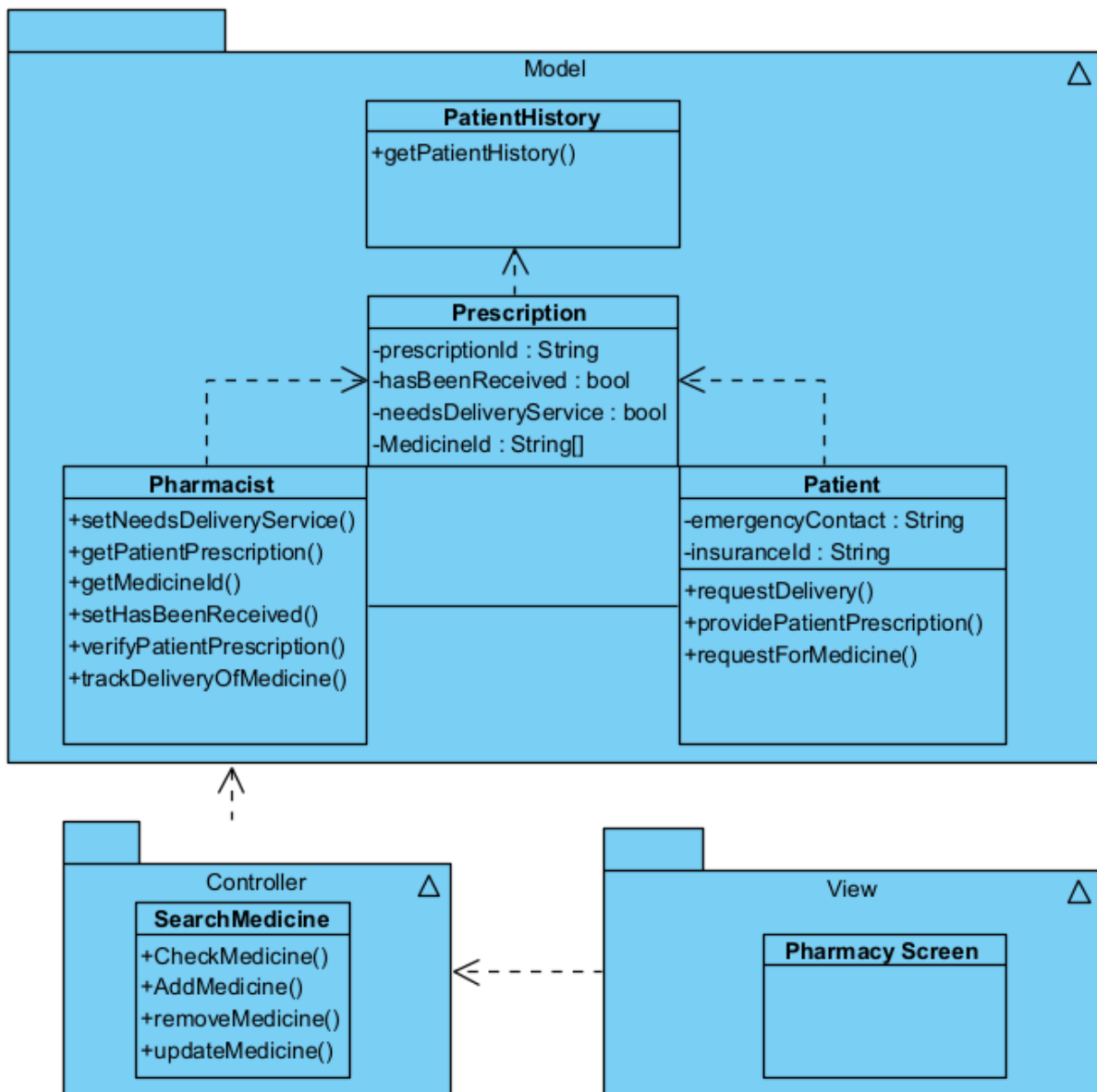
4.1.1 Login and Registration Subsystem



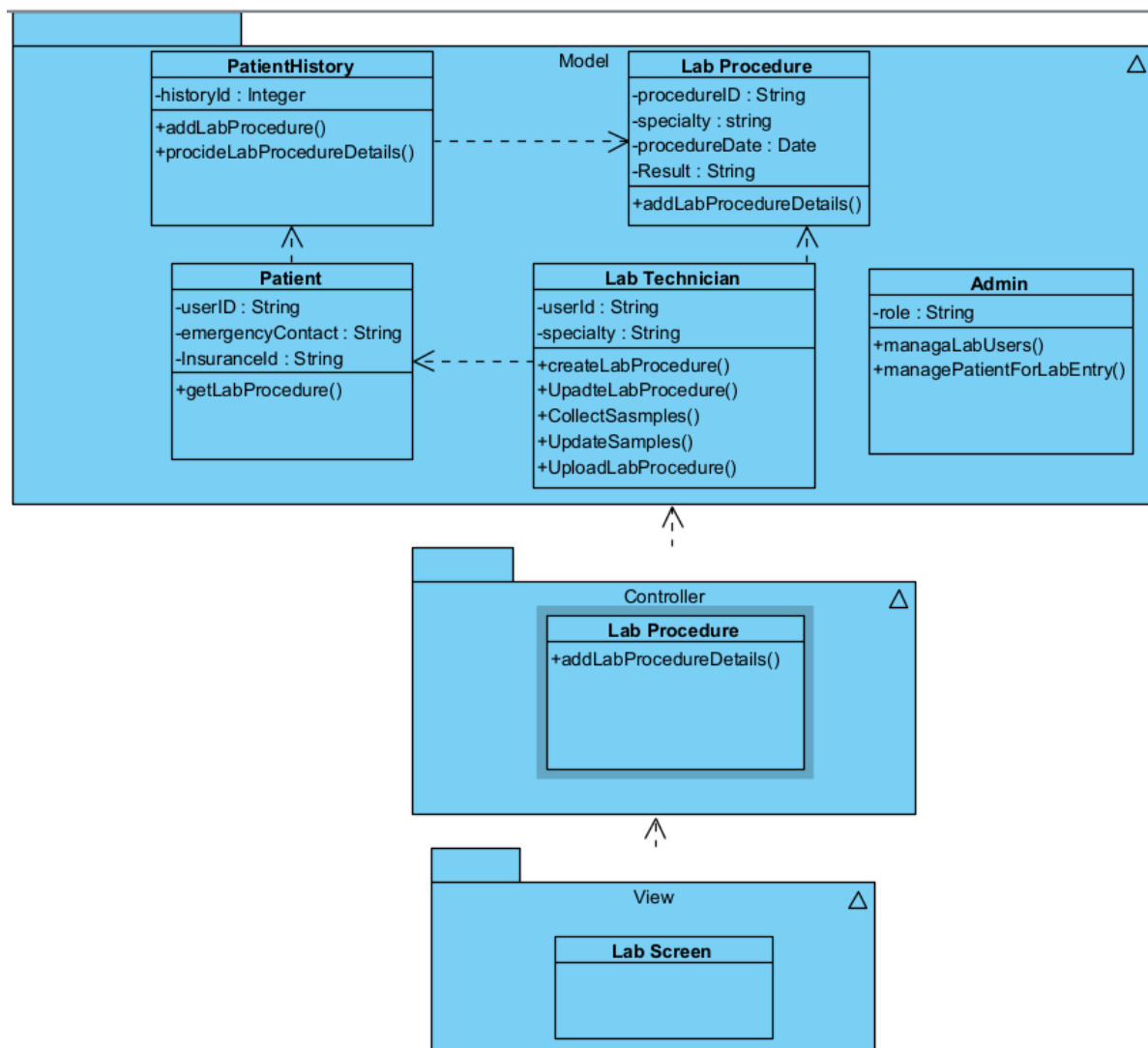
4.1.2 Consultation Subsystem



4.1.3 Pharmacy Subsystem

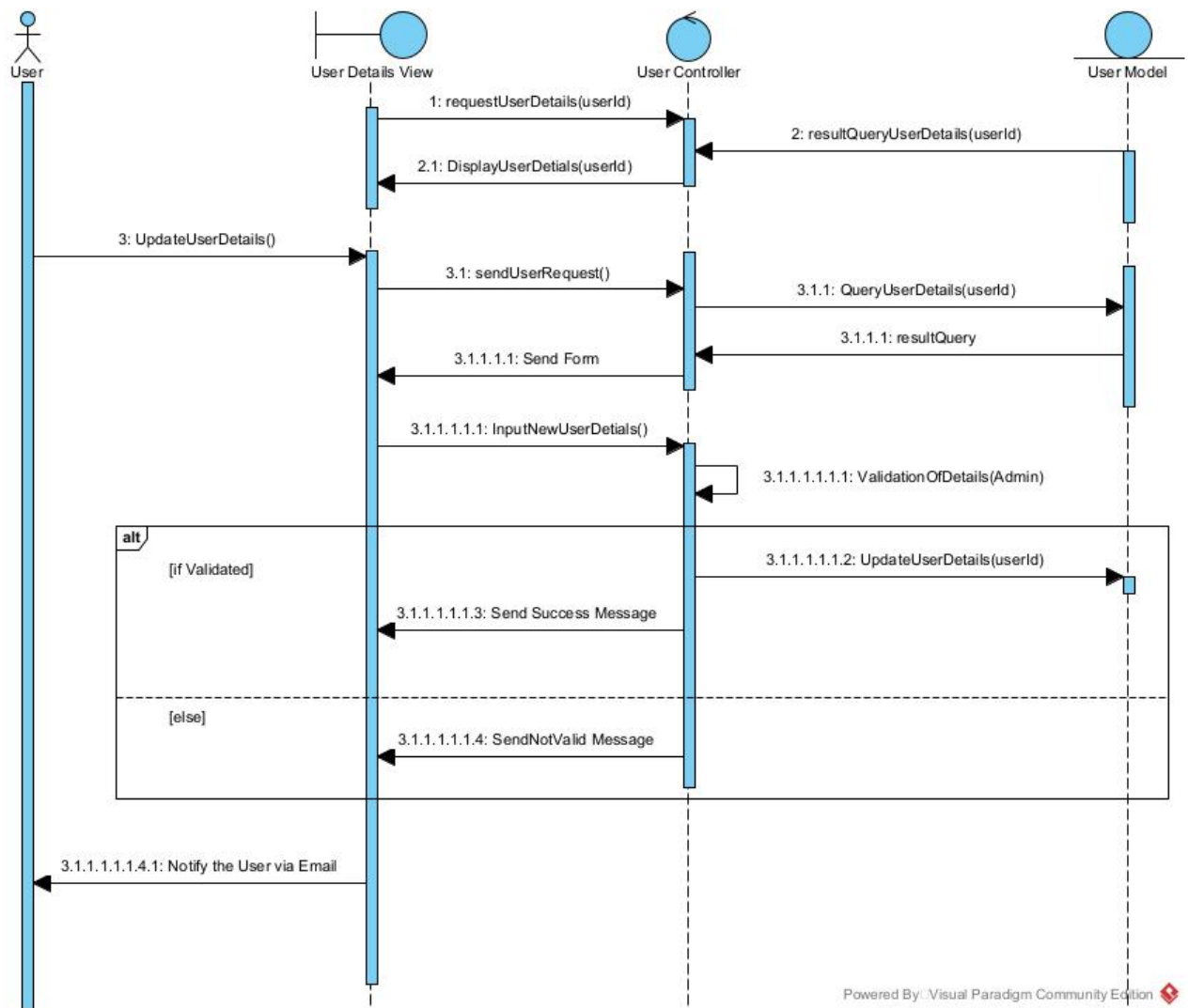


4.1.4 Lab Management Subsystem

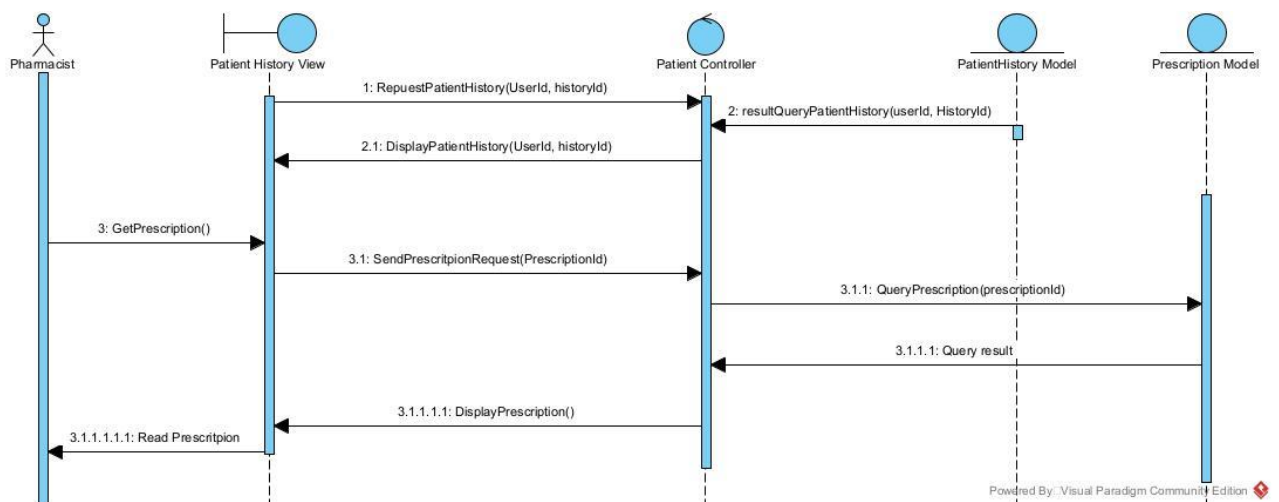


4.2 Full Sequence Diagrams

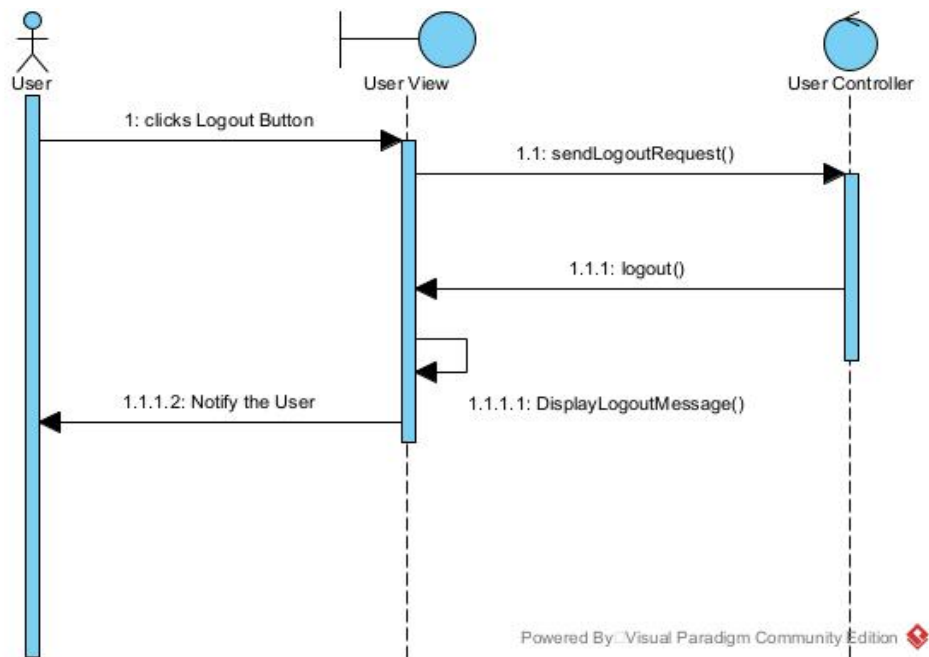
4.2.1 Use case: Request to update details (Login and Registration Subsystem)



4.2.2 Use case: Request Patient's prescription (Pharmacy Subsystem)

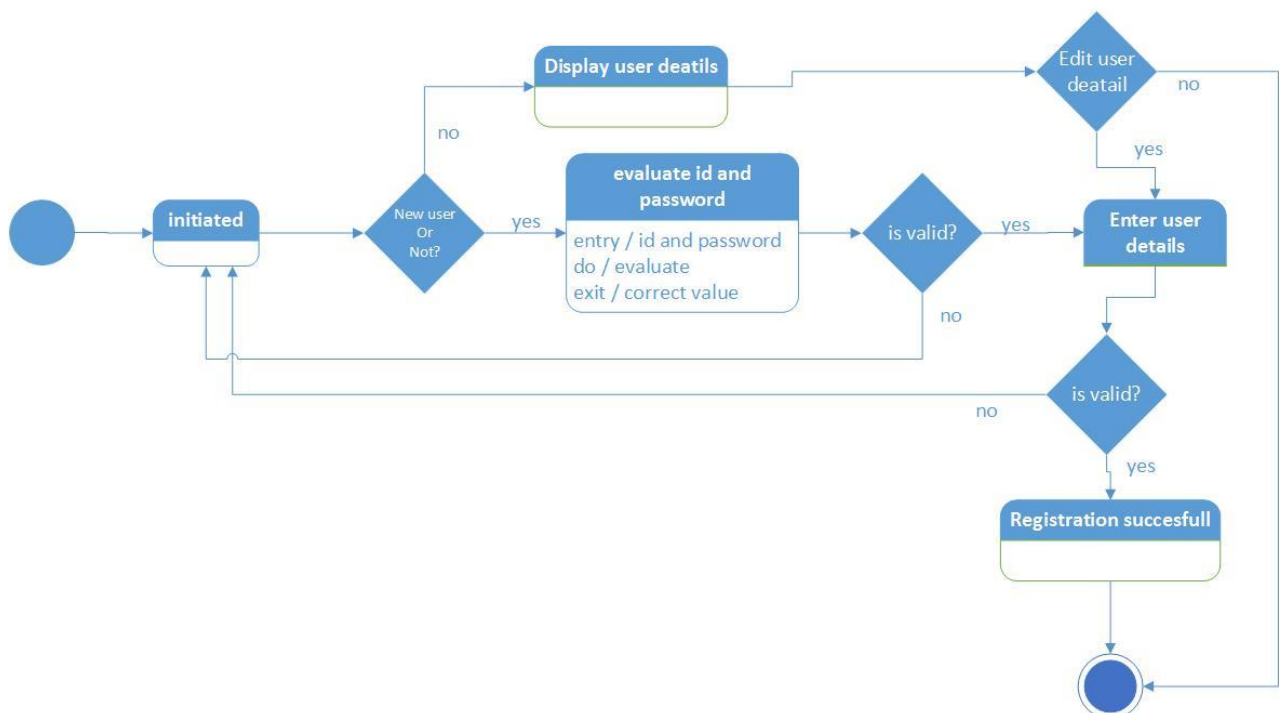


4.2.3 Use Case: Logout (Login and Registration Subsystem)

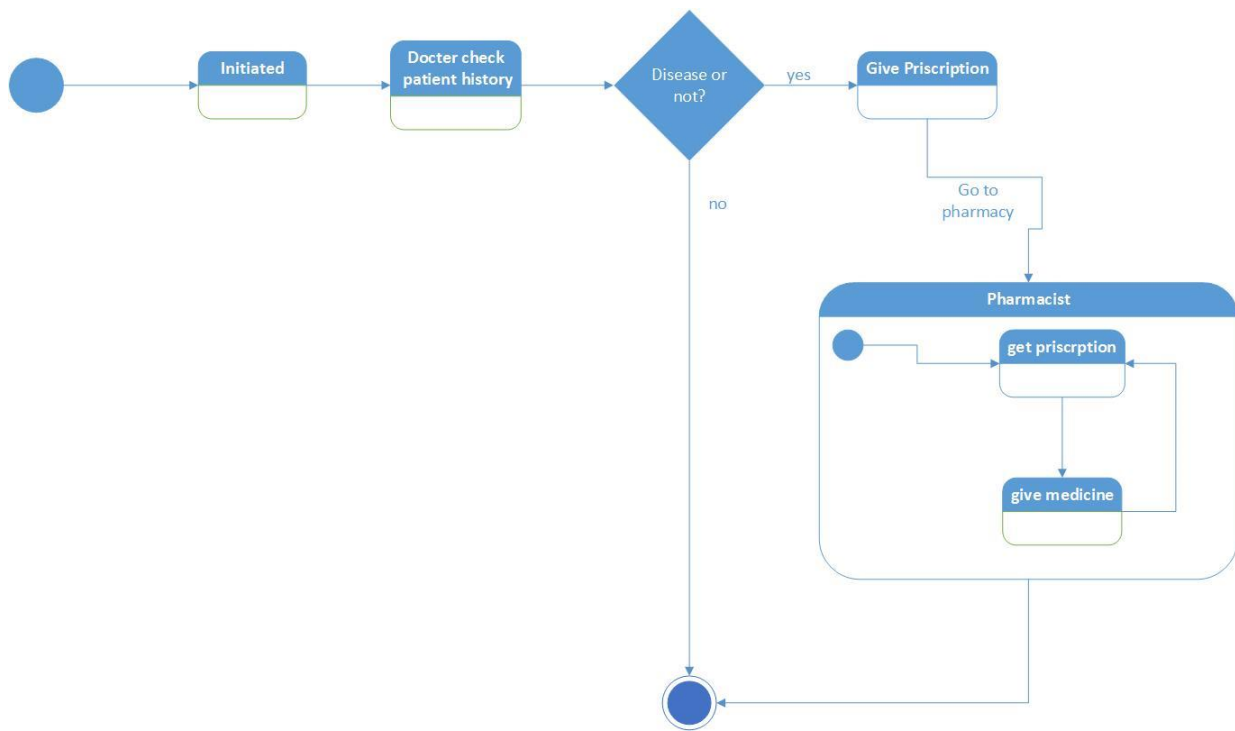


4.3 State Machine Diagrams

4.3.1 Update patient detail

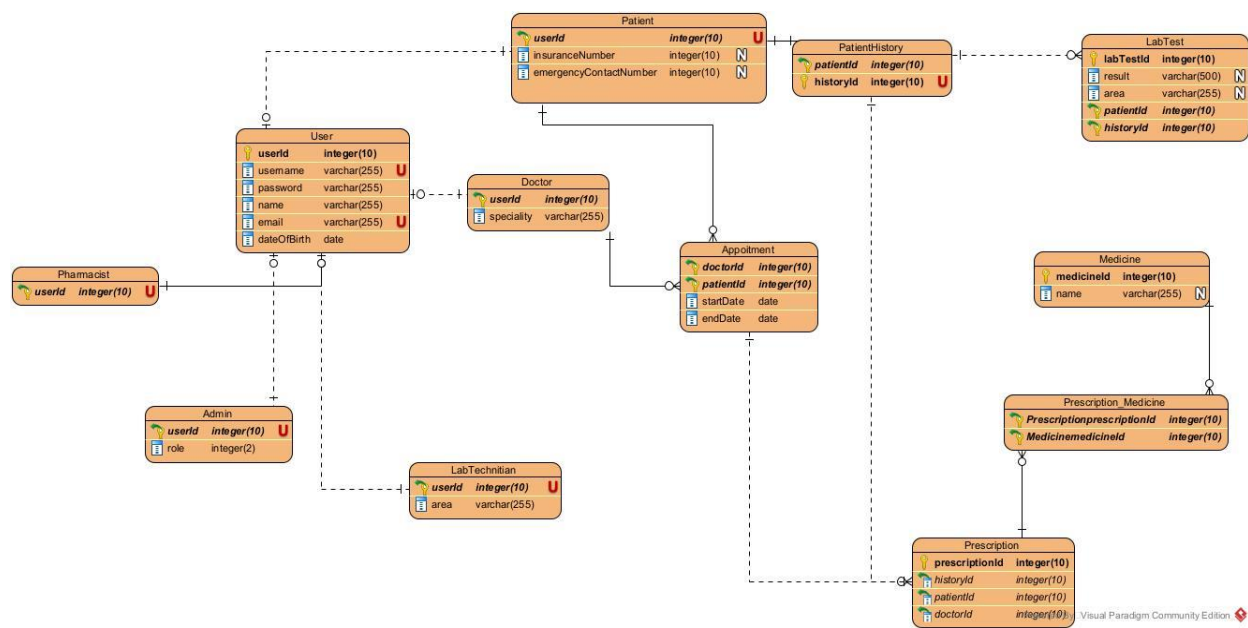


4.3.2 Update Prescription :



Section 5: Data Layer

5.1 Database Schema



5.2 Technology Update List

No updates required

Section 6: Gantt Chart Update

