CSE 5324 - 001: Software Engineering: Analysis, Design, and Testing

Electronic Health Record

Team 9: Fist

Team Members

Shubham Borhade (Captain)
Yash Karanje
Keerthi Patnaik
Preethi Subramanian
Shalaka Rane

Instructor

Dr. Michael F. Siok

05/01/2023
Iteration 3

TABLE OF CONTENTS

Electronic Medical Record (EMR)

Electronic Health Record (EHR) is an application that captures health data of patients, stores it, analyses it, provides reports, medications, handles accounting, and insurance coverage. As the 21st century is approaching towards digitization, there's a need to store patient health information, such that it must be useful in future of health analytics, research, and availability of data. EHR must be able to provide health analytics and availability of data which may be a difficult task without digitization of such important information. Today, the whole of North America continent uses an EHR, each facility (clinic and hospital) uses EHR in their daily work schedule. EHR has made it efficient to handle and manage health data and to provide accuracy in health operations. Hence, we propose to develop an android application that could be used by only one clinic/hospital to capture health information and do analysis over the same.

Functionalities:

a. **Function 1: Login/Sign-up**

- The application must have a login page so that the users in the application (Patient, Provider (Doctor), Laboratory, Pharmacy, Insurance, and Admin) should be able to login. The application must have a sign-up page only for the patients to register. The patients must be able to sign up by filling their demographics.

b. **Function 2: Update/View Profile**

- All the users in the application should be able to view and update their demographics and credentials.

c. **Function 3: Book/Manage Visit**

- Patients should be able to book a visit/appointment in the application by selecting a date/time slot and enter some patient notes.
- Patients should be able to cancel the appointment only before the scheduled time. Once the appointment is cancelled, it should not be seen in the application.
- The provider should be able to view the appointment in the application, patient details, also should be able to work on it by adding/updating the provider notes, symptoms, and diagnosis related to the respective appointment in the application.

d. **Function 4: Add/Manage Account Charges**

- Once a patient has been treated, the provider should be able to add the visit charges in the application.
- Once the patient pays a part of the visit charge, the balance amount should be displayed to the patient's insurer, and the respective insurance company pays the coverage for the patient visit in the application.
- Patients should be able to view their account charges.
- Patients should be able to add insurance, view insurance details, and disable the insurance if required.

e. **Function 5: Add/Manage Lab Test**

- Whenever a provider treats a patient and there's a need for the patient to be tested for something, in this case the provider must be able to add a lab test for the patient visit with one of the registered laboratories in the application.
- The laboratory should be able to view the assigned lab test details, view patient details, and add a written report of the test performed.
- Once the report is submitted, both the provider and patient should be able to view it in the application.

f. **Function 6: Add/Manage Medication**

- Whenever a provider treats a patient and there's a need for the patient to be prescribed medication, in this case the provider must be able to add the medication for the patient visit with one of the registered pharmacies in the application.

- Once the medication has been assigned, the corresponding pharmacy must be able to view the medication and patient details. Also, once the medicine has been handed over to the patient, the pharmacy should be able to update the handover status.

g. **Function 7: Add Users**

- Admin should be able to add users (Providers, Laboratory, Pharmacy, and Insurance companies in the application).

h. **Function 8: Business Aspect Functionalities**

- Provider should be able to view reports and perform analytics on the data captured in the system.

i. **Function 9: Logout**

- The users of the application should be able to logout of the application by clicking on the logout button which will then lead them back to the login page.

j. **Function 10: Forgot Password**

- If a user forgets the password, then the user should be able to use the forgot password functionality. The user must enter their email address, and the application must send the user's password on the corresponding email address.

Resources:

Android Studio, MySQL Database, UTA Cloud

Team Members Bios

Shubham Borhade:
Shubham has worked for 3.5 years at varied IT Companies. Specifically, he has worked on GIS, ORMB, Web Development, and HL7 Interface technologies. He has worked on several web-development and interfaces designing projects in the past. He has proficiency in varied programming languages like Java, C, C++, C#, PHP, Python, ReactJS, NodeJS, JavaScript, HTML, CSS, and SQL. He has attended android workshops and developed an event management android application in the past. He has experience working in agile software development and has knowledge about different SDLCs. Although, with only one android development experience he is excited and optimistic to take this challenge of developing an android application along with his teammates with a precise software development approach.

Yash Karanje:
Yash has developed an Android app before for Wholesale-Service-Management using Android Studio, Firebase Realtime Application and Java, mainly designing DB schema. Further, he converted the same project into a Desktop application. He has also contributed extensively to his other baccalaureate projects in Idea exploration, UML design and Implementation using language specific IDEs. Besides, Yash has corporate work experience of 2 years under e-commerce domain as SAP Application Developer in the Agile work environment. Yash is excited to have hands-on practice on Android; learning from his prior experience, he is curious to contribute to the best of his knowledge in every phase of the project.

Keerthi Patnaik:
Keerthi has 2.5 years of experience in web development using HTML, CSS, and JavaScript. She is familiar with using Visual Studio Code as an IDE for all her development work. She has experience in using the Agile methodology for project management and is comfortable working in a team environment. Despite having no prior experience in Android development, Keerthi is eager to learn and contribute to the team's project. Her background in web development, along with her proficiency in various programming languages (JavaScript, ReactJS & NextJS), will prove to be valuable assets for the team.

Shalaka Rane:
Shalaka has developed an E-commerce Android Application using Android Studio, React Native, Firebase Realtime DB and a simple World Clock Android Application using Flutter. She has also contributed to Web Applications using Html, CSS, JavaScript during her Undergraduate program and is familiar with web frameworks like Django, ReactJS, Angular and NodeJS. Furthermore, she has an industry experience of 1 year as a Software Engineering Professional and has worked on Java Full Stack Web Development for the duration of her employment. She is very excited to explore more of Android Development, learning from previous experiences and from her current project team’s knowledge; and contribute greatly to the project.

Preethi Subramanian:
Preethi has 2.9 years of work experience as a Software developer in the Investment banking technology domain. She is familiar with Java, Scala, HTML, CSS, Oracle application, and BDD approach using cucumber for test automation. During her Undergraduate, she also got experience designing UML as part of the software development process. Furthermore, she is familiar with using IntelliJ and Visual Studio Code as IDEs for most of her development work. She also has experience working in an agile environment and is familiar with agile ceremonies. Regardless of having no prior experience in Android development, she is intrigued and looking forward to learning and contributing to the best of her capacity to the team's project.

Requirements

Requirements	Requirement Statements	Line Reference
R1	The Electronic Health Record (EHR) shall have a login page.	13
R1.1	The EHR shall allow the registered user (Patient, Provider, Laboratory, Pharmacy, Insurance and Admin) to login.	13
R1.2	The EHR shall allow only the patient user to self-register in the application.	14
R1.3	The EHR shall allow the user to retrieve their current password using the forgot password link.	55
R1.4	The EHR shall display authentication errors whenever incorrect credentials are entered by the user.	Derived
R2	The EHR shall allow the user to manage their profile.	17
R2.1	The EHR Shall allow the user to view their profile.	17
R2.2	The EHR shall allow the user to update their profile details	17
R3	The EHR shall allow the patient to manage appointment	Derived
R3.1	The EHR shall allow the patient to book an appointment	19
R3.2	The EHR shall allow the patient to view appointment details.	Derived
R3.3	The EHR shall allow the patient to cancel an appointment only before the scheduled appointment time.	21
R4	The EHR shall allow the patient to manage insurance.	Derived
R4.1	The EHR shall allow the patient to view insurance details.	31
R4.2	The EHR shall allow the patient to add insurance details, only if there is no active insurance.	31
R4.3	The EHR shall allow the patient to disable the insurance details.	31
R5	The EHR shall allow the provider to manage appointment.	Derived
R5.1	The EHR shall allow the provider to view appointment details.	23
R5.2	The EHR shall allow the provider to add diagnosis, symptoms, and notes for an appointment.	23
R5.3	The EHR shall allow the provider to assign a lab test for an appointment with one of the registered laboratories.	33
R5.4	The EHR shall allow the provider to prescribe medication for an appointment with one of the registered pharmacies.	40
R5.5	The EHR shall allow the provider to add appointment charges.	27

Requirements (Contd.)

Requirements	Requirement Statements	Line Reference
R6	The EHR shall allow the provider to view patient details.	23
R7	The EHR shall allow the provider to perform analytics.	50
R8	The EHR shall allow the laboratory to manage lab test.	Derived
R8.1	The EHR shall allow the laboratory to view test details.	36
R8.2	The EHR shall allow the laboratory to add the test results.	36
R9	The EHR shall allow the laboratory to view the patient details.	36
R10	The EHR shall allow the pharmacy to manage prescribed medication.	Derived
R10.1	The EHR shall allow the pharmacy to view the prescribed medication.	43
R10.2	The EHR shall allow the pharmacy to update medication handover status.	43
R11	The EHR shall allow the pharmacy to view the patient details.	43
R12	The EHR shall allow the insurance company to manage insurance coverages.	Derived
R12.1	The EHR shall allow the insurance company to view insurance coverages.	Derived
R12.2	The EHR shall allow the insurance company to update the insurance coverage for the patient appointment.	28
R13	The EHR shall allow the insurance company to view the patient details.	Derived
R14	The EHR shall allow the admin to manage users (Provider, Laboratory, Pharmacy, and Insurance Company).	Derived
R14.1	The EHR shall allow the admin to add users.	47
R14.2	The EHR shall allow the admin to view users.	Derived
R15	The EHR shall allow the user to logout.	52

Use Cases

Use Case No.	Use Case Name	Patient	Provider	Laboratory	Pharmacy	Insurance Company	Admin
UC1	Login	X	X	X	X	X	X
UC2	Register Patient	X					
UC3	Forgot password	X	X	X	X	X	X
UC4	View profile	X	X	X	X	X	X
UC5	Update profile	X	X	X	X	X	X
UC6	Patient manages appointment	X					
UC 6.1	Book appointment	X					
UC 6.2	View appointment details	X					
UC 6.3	Cancel appointment	X					
UC7	Patient manages insurance	X					
UC7.1	View insurance details	X					
UC7.2	Add insurance details	X					
UC7.3	Disable insurance	X					
UC8	Provider manages appointment		X				
UC8.1	View appointment details		X				
UC8.2	Add diagnosis, symptoms, and notes		X				
UC8.3	Assign lab test		X				
UC8.4	Prescribe medication		X				
UC8.5	Add appointment charges		X				
UC9	Provider views patient details		X				
UC10	Provider performs analytics		X				
UC11	Laboratory manages lab test			X			
UC11.1	Laboratory view test details			X			
UC11.2	Laboratory add test results			X			
UC12	Laboratory views patient details			X			
UC13	Pharmacy manages prescribed medication				X		
UC13.1	Pharmacy view medication details				X		
UC13.2	Pharmacy update medication handover status				X		
UC14	Pharmacy views patient details				X		
UC15	Insurance company manages insurance coverage					X	
UC15.1	Insurance company views insurance coverage					X	
UC15.2	Insurance company updates insurance coverage					X	
UC16	Insurance company views patient details					X	
UC17	Admin manages users (Provider, Laboratory, Pharmacy, Insurance Company)						X
UC17.1	Admin view user details						X
UC17.2	Admin adds user						X
UC18	Logout	X	X	X	X	X	X

High Level Use Cases

UC1: User Login

- TUCBW user being displayed the login page.
- TUCEW user being logged in and redirected to the EHR home page.

UC2: Register Patient

- TUCBW patient opens the registration form page.
- TUCEW the patient is successfully registered.

UC3: Forgot Password

- TUCBW user clicks on the forgot password link.
- TUCEW user's current password being emailed to the registered email address.

UC4: User views profile

- TUCBW user clicks on the profile option from the menu bar.
- TUCEW user being able to view their profile details.

UC5: User updates profile

- TUCBW user clicks on the update profile button.
- TUCEW user profile data being updated in the application.

UC6: Patient manages appointment

- TUCBW patient clicks on the appointment schedule.
- TUCEW patient being able to view the appointment schedule.

UC6.1: Patient books an appointment

- TUCBW patient clicks on the book appointment button.
- TUCEW appointment being booked in the application.

UC6.2: Patient views an appointment

- TUCBW patient clicks on a particular appointment.
- TUCEW patient being able to view the appointment details.

UC6.3: Patient cancels an appointment

- TUCBW patient clicks on the cancel button.
- TUCEW appointment status being updated to cancel.

UC7: Patient manages insurance

- TUCBW patient clicks on the insurance menu.
- TUCEW patient being able to view all the insurance details.

UC7.1: Patient views insurance details

- TUCBW patient selects a particular insurance from the list of insurance subscriptions.
- TUCEW patient being able to view all the details of the specific insurance subscription.

UC7.2: Patient adds an insurance

- TUCBW patient clicks on the add insurance button.
- TUCEW insurance details being saved in the application.

UC7.3: Patient disables an insurance

- TUCBW patient clicks on the disable button.
- TUCEW patient's insurance status being updated to disabled.

UC8: Provider manages appointment

- TUCBW provider clicks on appointment schedule option from the menu bar.
- TUCEW provider being able to view the appointment schedule.

UC8.1: Provider views appointment details

- TUCBW provider selects an appointment.
- TUCEW provider being able to view all the details of the appointment.

UC8.2: Provider adds diagnosis, symptoms, and notes

- TUCBW provider enters the diagnosis, symptoms, and notes in the appointment general details form.

- TUCEW entered details being saved in the application.

UC8.3: Provider adds lab test
- TUCBW provider enters the test details in the appointment test details form.
- TUCEW lab test details being saved in the application.

UC8.4: Provider prescribes medication
- TUCBW provider enters the medication details in the appointment test details form.
- TUCEW medication details being saved in the application.

UC8.5: Provider adds appointment charges
- TUCBW provider enters the charges in the appointment charges form.
- TUCEW appointment charges being saved in the application.

UC9: Provider views patient Details
- TUCBW provider clicks on a particular patient entry.
- TUCEW provider being able to view the patient details.

UC10: Provider performs analytics
- TUCBW provider clicks on the analytics option from the menu bar.
- TUCEW provider being able to view the analytics report details.

UC11: Laboratory manages lab test
- TUCBW laboratory user opens assigned lab test page.
- TUCEW laboratory user being able to view all pending lab tests.

UC11.1: Laboratory view test details
- TUCBW laboratory user clicks on a specific lab test.
- TUCEW laboratory user being able to view the details of the lab test.

UC11.2: Laboratory add test results
- TUCBW laboratory user adds the written result of a specific lab test.
- TUCEW lab result being saved in the application.

UC12: Laboratory views patient Details
- TUCBW laboratory clicks on the patient's name from the list of lab tests assigned.
- TUCEW laboratory being able to view the patient details.

UC13: Pharmacy manages medication
- TUCBW pharmacy user opens assigned medication page.
- TUCEW pharmacy user being able to view all the assigned medication details.

UC13.1: Pharmacy views medication
- TUCBW pharmacy user clicks on a specific medication entry from the available list.
- TUCEW pharmacy user being able to view all the details of the medication.

UC13.2: Pharmacy updates the handover status
- TUCBW pharmacy user updates the medication handover status to Complete.
- TUCEW medication status being updated in the application.

UC14: Pharmacy views patient Details
- TUCBW pharmacy clicks on a patient entry.
- TUCEW pharmacy being able to view the selected patient details.

UC15: Insurance company manages insurance coverage
- TUCBW insurance company user opens insurance coverage page.
- TUCEW insurance company user being able to view all the insurance coverage details.

UC15.1: Insurance company views the insurance coverages
- TUCBW insurance company user clicks on a specific insurance coverage from the available list.
- TUCEW insurance company user being able to view all the details of the insurance coverage.

UC15.2: Insurance company updates the insurance coverage

- TUCBW insurance company user updates the coverage amount paid for the patient appointment.
- TUCEW insurance coverage amount being updated in the application.

UC16: Insurance company views patient Details

- TUCBW insurance company clicks on patient entry.
- TUCEW insurance being able to view the selected patient details.

UC17: Admin manages users

- TUCBW admin clicks on the dashboard icon.
- TUCEW admin being able to view all the users in the system.

UC17.1: View User Details (Provider, Laboratory, Pharmacy, and Insurance Company).

- TUCBW admin clicks on a user entry.
- TUCEW admin being able to view the user details.

UC17.2: Admin adds users (Patient, Provider, Laboratory, Pharmacy, and Insurance Company).

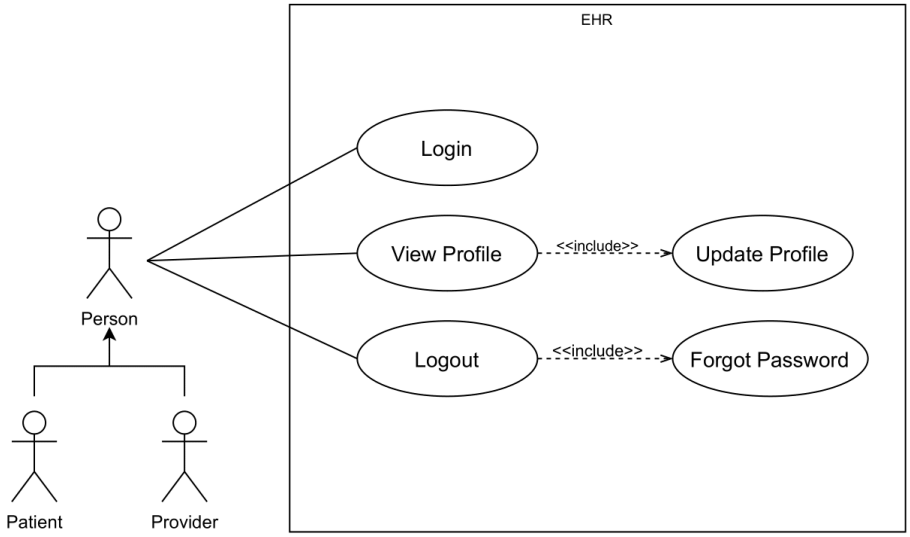
- TUCBW admin clicks on the add user form.
- TUCEW user details being added to the application.

UC18: User Logout

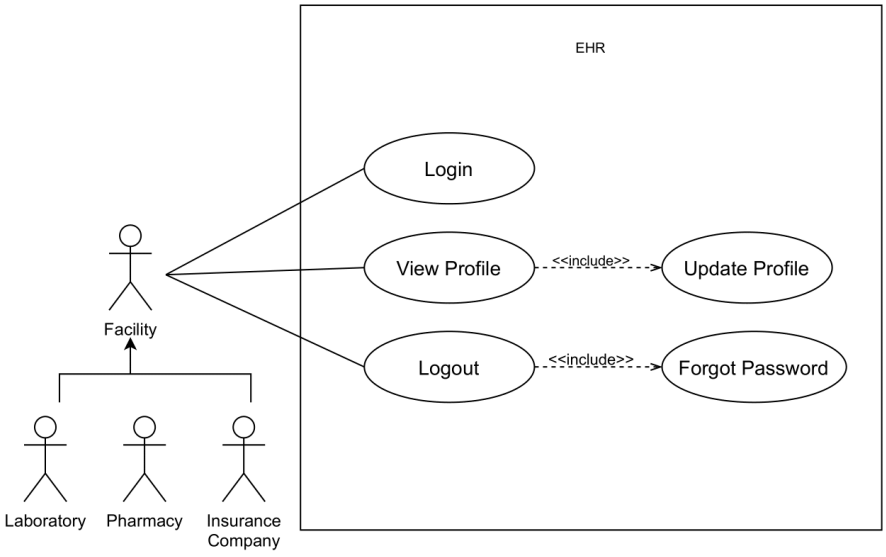
- TUCBW user clicks on the logout button.
- TUCEW user being logged out of the application.

Use Case Diagrams

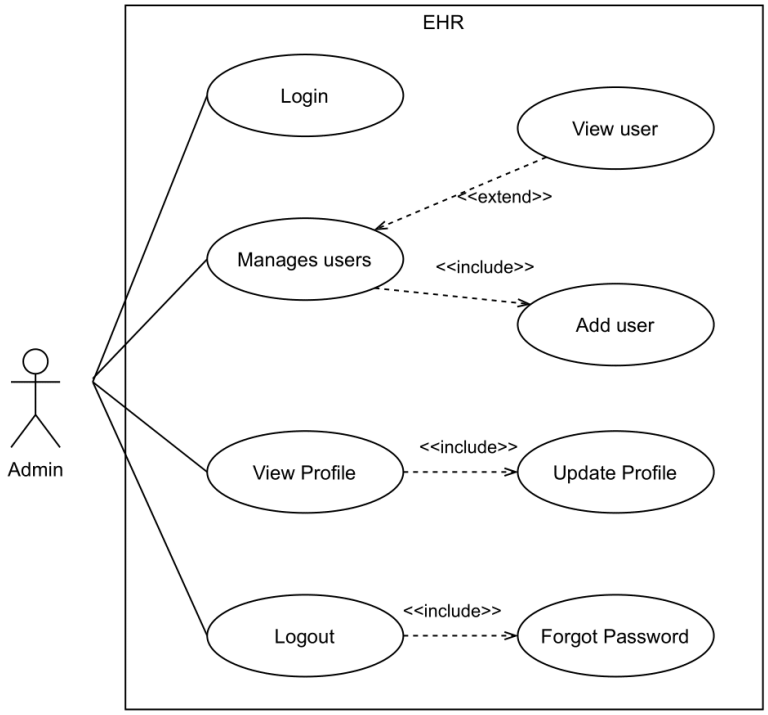
Person



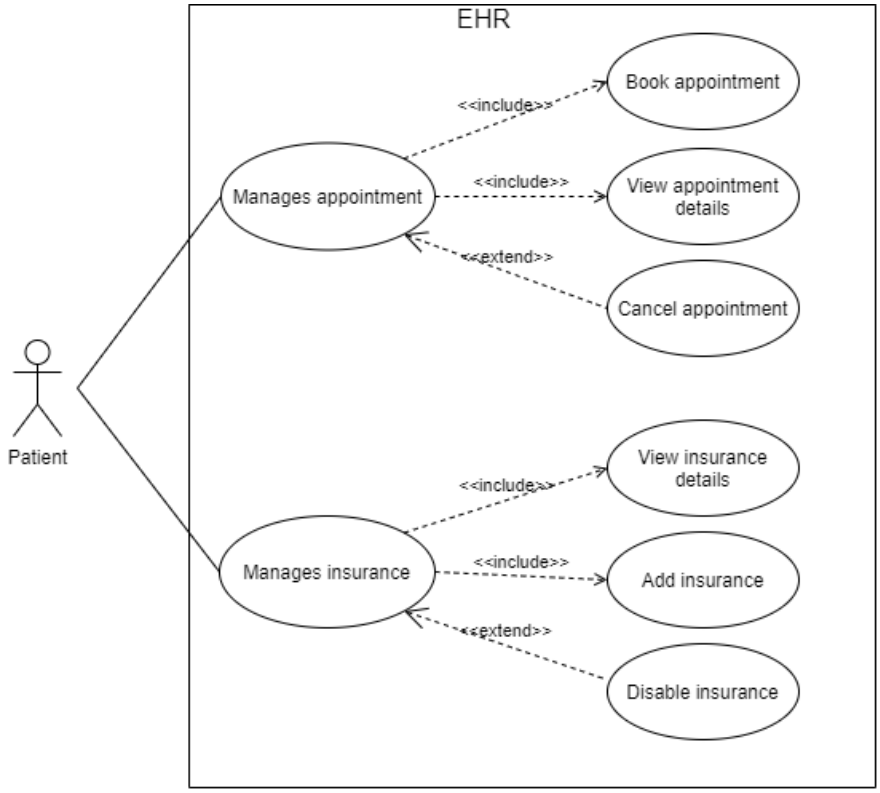
Facility



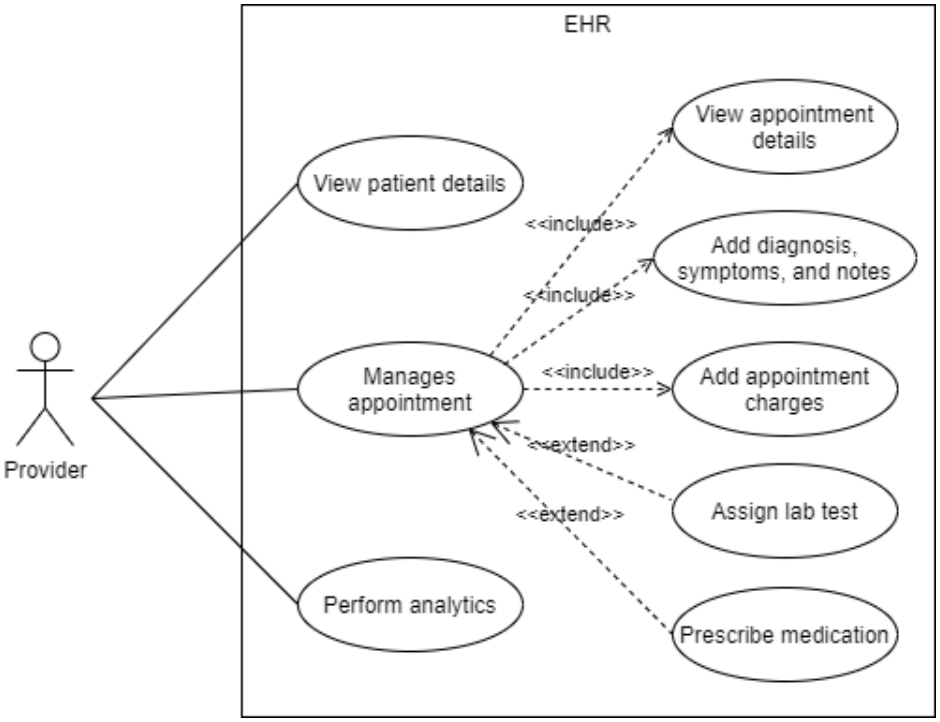
Admin



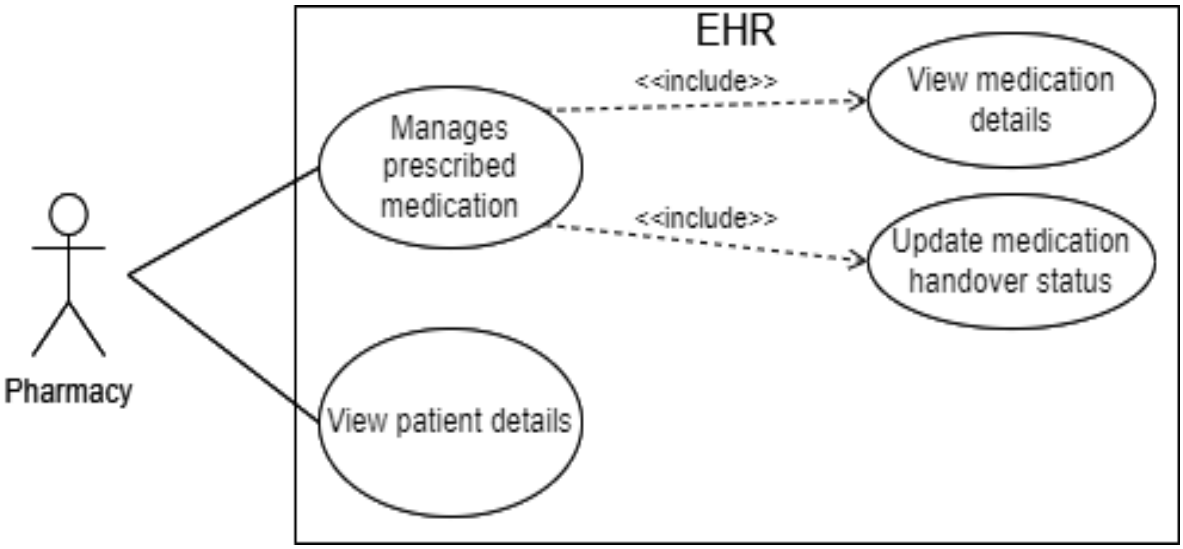
Patient



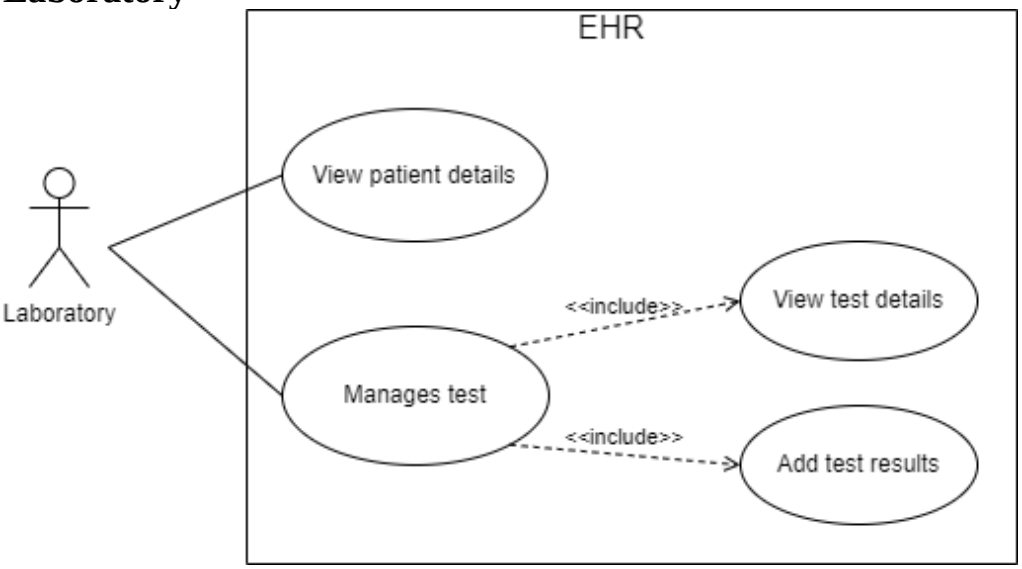
Provider



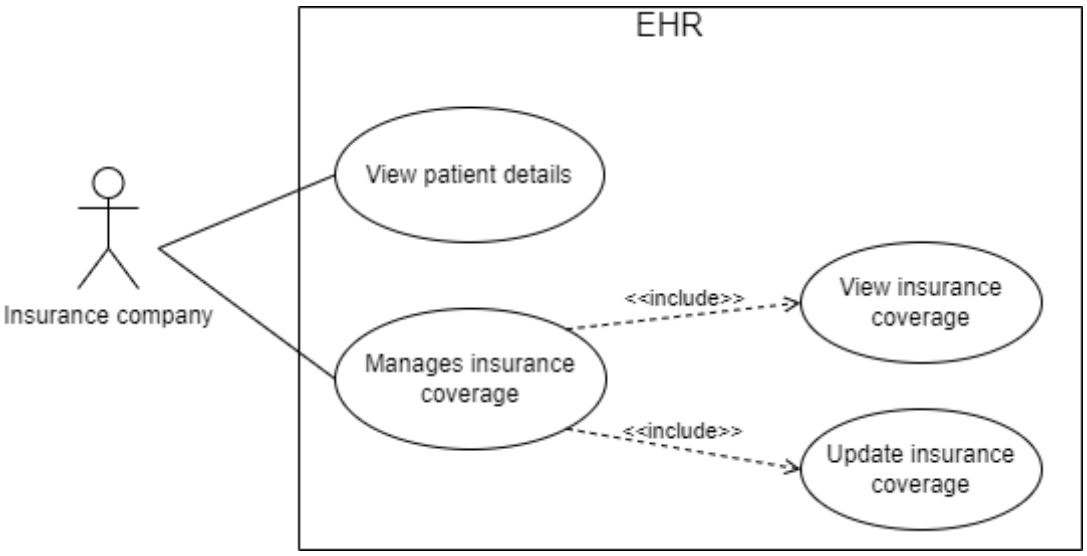
Pharmacy



Laboratory



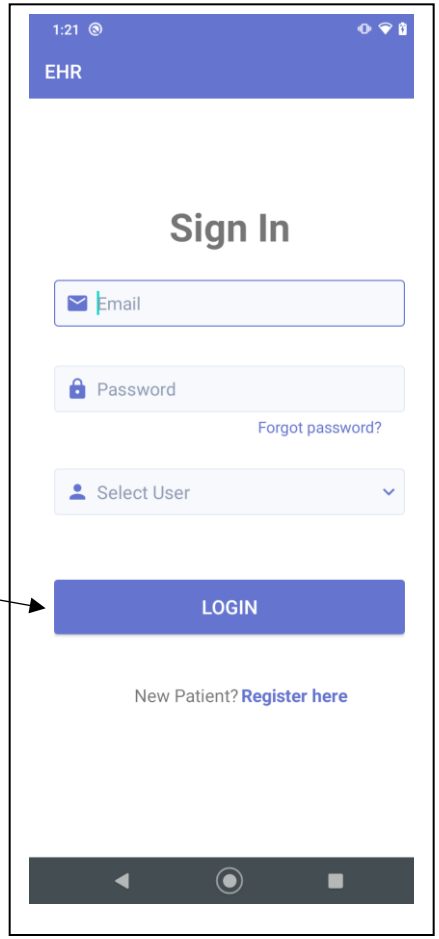
Insurance Company

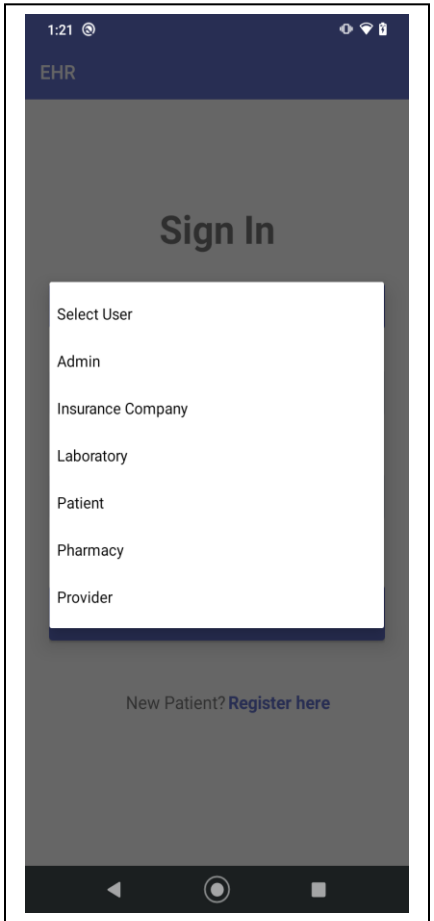


EUC/UIPs

EUC 1: Login.

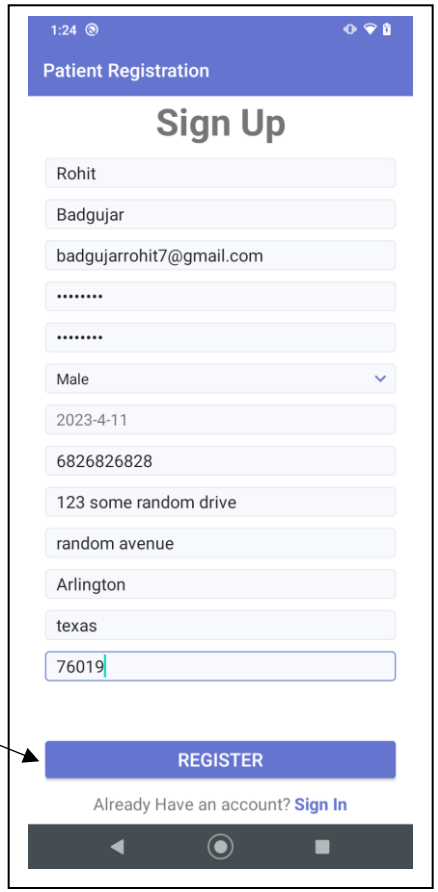
Precondition: This use case assumes that the user is on Login page.	
Actor: User	System: EHR
1. **TUCBW** the user enters the emailid, password, and selects the user type, and clicks on the 'Login' button. 3. **TUCEW** the user being logged into the system.	0. System displays the login page. *2. System will validate the user provided credentials: a. If the credentials are valid, then the system creates the user object and redirects the user to the respective home page. b. If the credentials are invalid, then the system displays 'Invalid Credentials' message.
Post Condition: The user being logged into the system and able to view the homepage.	

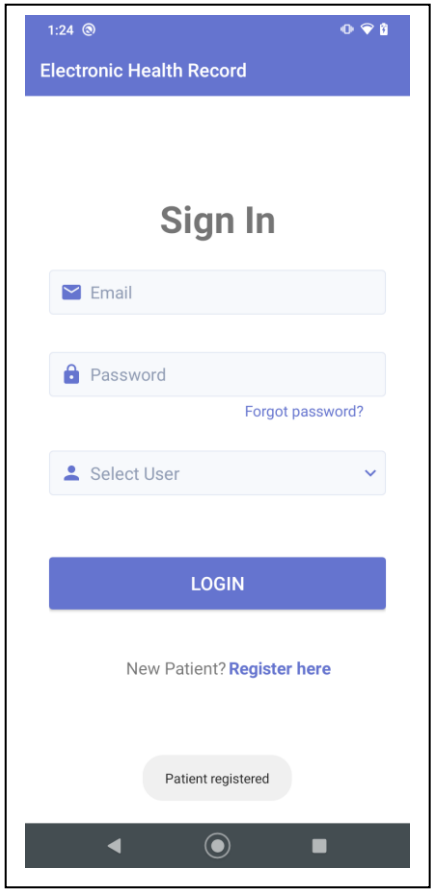




EUC 2: Register Patient

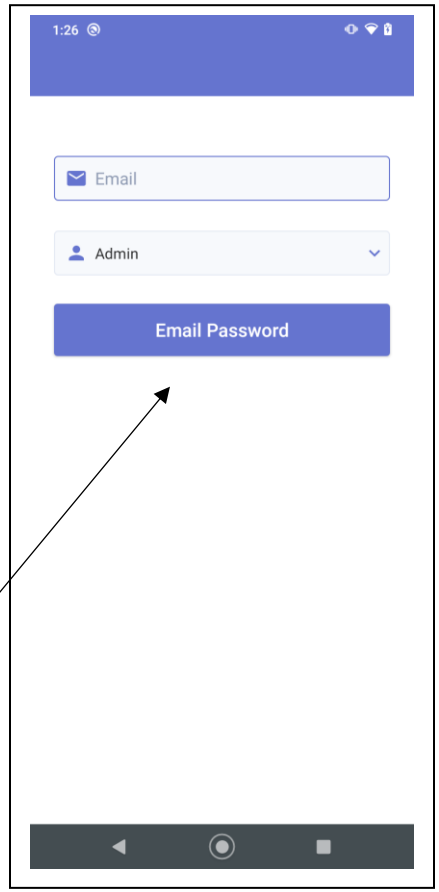
Precondition: This use case assumes that the user is on the patient registration page.	
Actor: Patient	System: EHR
1. **TUCBW** the patient enters their personal and demographic details, and clicks on the 'Register' button. 3. **TUCEW** the patient registered successfully.	0. System displays the patient registration page. *2. System will validate the entered details: a. If the entered emailid already exist in the system, then the system doesn't register the patient and displays 'Emailid already exist' message. b. Patient details are saved and the patient is registered successfully, and shows the 'Patient Registered' message.
Post Condition: Patient registered and able to login.	

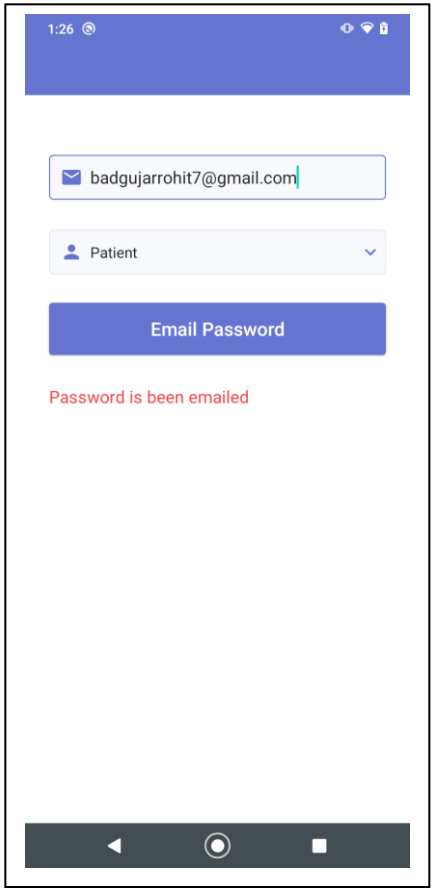




EUC 3: Forgot password

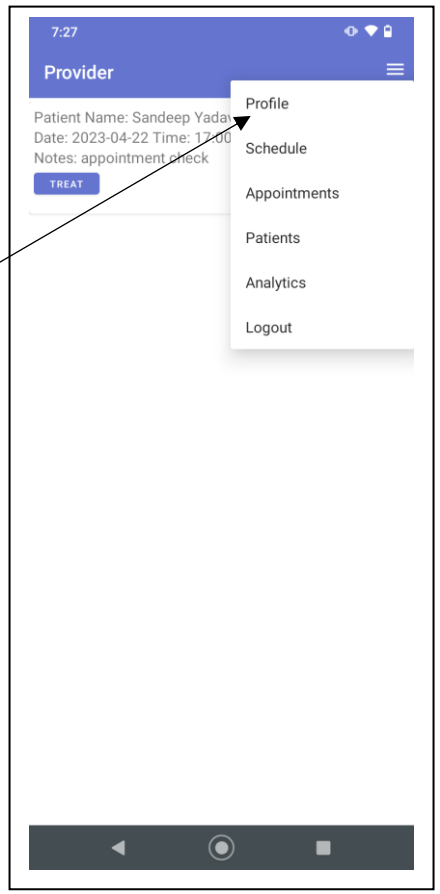
Precondition: This use case assumes that the user is on the forgot password page.	
Actor: User	System: EHR
1. **TUCBW** the user enters the registered email-id, selects the user type from the dropdown, and clicks on the 'Email Password' button. 3. **TUCEW** the user receives the password on their email account.	0. System displays the forgot password page with field to enter registered email-id and select the user type, and 'Email Password' button. *2. System checks if the email-id exists in the database and takes the appropriate action: a. If the email-id exists, then the system retrieves the password associated with the same and emails the password to the provided email-id. b. If the email-id doesn't exist, then system displays an error message.
Post Condition: The user can use the received password to log into the system.	

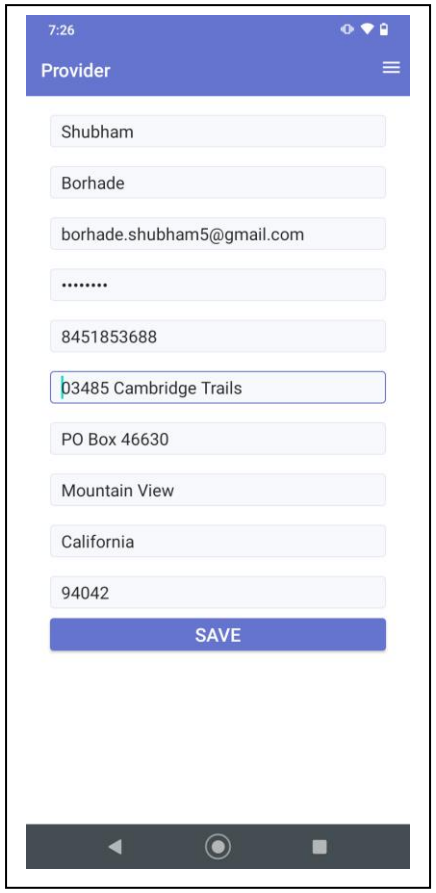




EUC 4: View profile

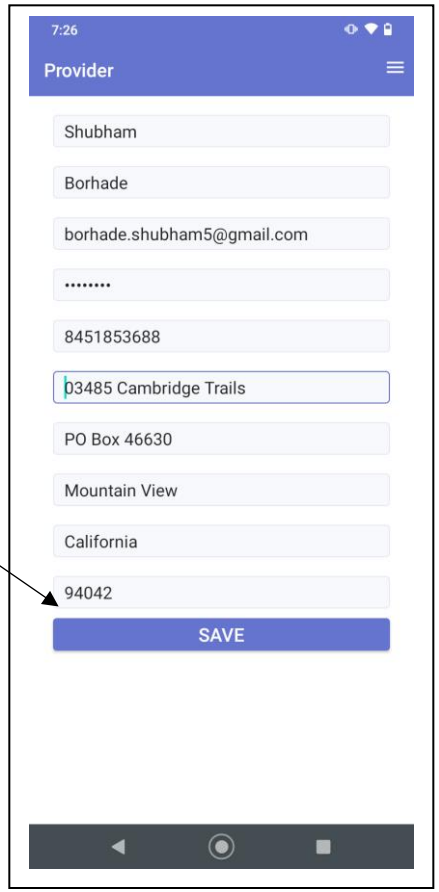
Precondition: This use case assumes that the user is logged into the system and on the homepage.	
Actor: User	System: EHR
1. **TUCBW** the user clicks on the 'Profile' from the menu bar. 3. **TUCEW** the user able to see the profile details.	0. System displays the user homepage. *2. System displays the user profile details.
Post Condition: User able to see to the profile details and can update the same.	

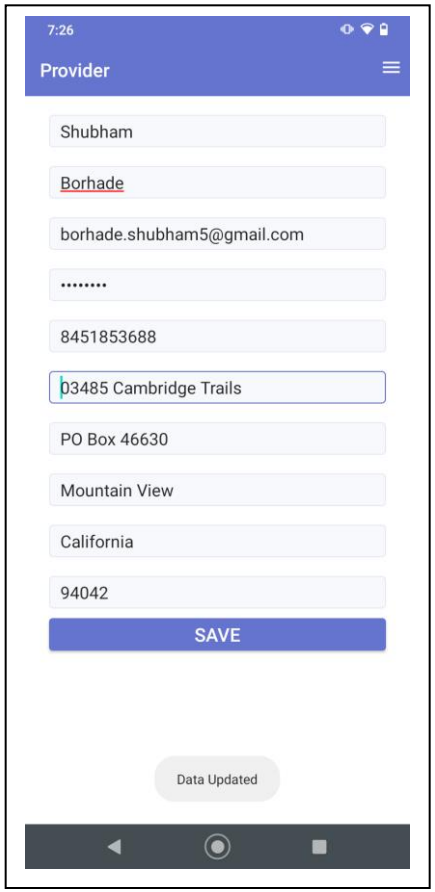




EUC 5: Update profile

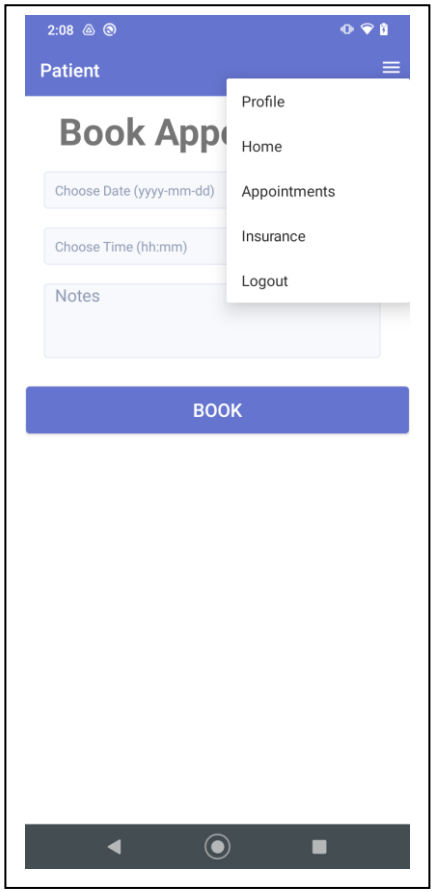
Precondition: This use case assumes that the user is logged into the system and on the profile page.	
Actor: User	System: EHR
1. **TUCBW** the user enters the details that they want to update in the profile page fields and clicks on the 'Save' button. 3. **TUCEW** the user able to see the updated profile details.	0. System displays the user profile page. *2. System saves the user entered details in the database and shows the updated profile details, and 'Data updated' message.
Post Condition: User able to see to the profile details.	

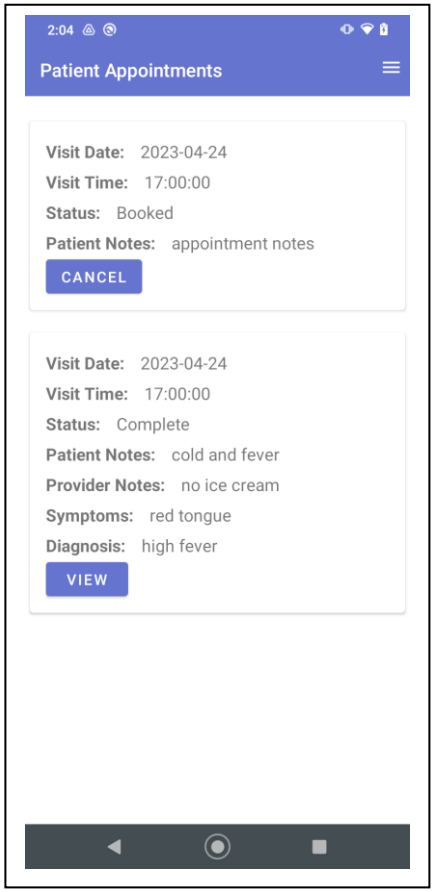




EUC 6: Patient manages appointment.

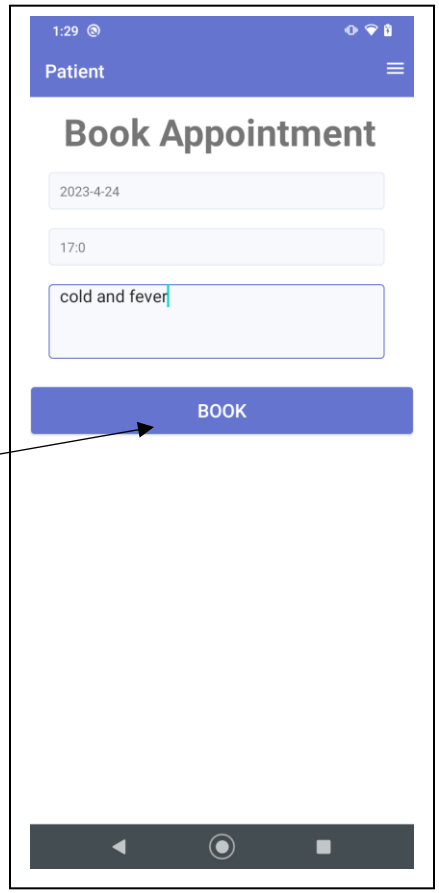
Precondition: This use case assumes that the Patient user is already logged into the system and is on the home page.	
Actor: Patient	System: EHR
1. **TUCBW** the patient may click on the 'Appointments' from the menu bar. 3. The patient user may click on the View button on a particular completed appointment. 5. **TUCEW** the patient user being able to view the appointments scheduled.	0. System displays the patient homepage. 2. System shall display all the appointments that belong to the patient. 4.System shall display the pending appointment details.
Post Condition: Patient user being able to view pending and completed appointments.	

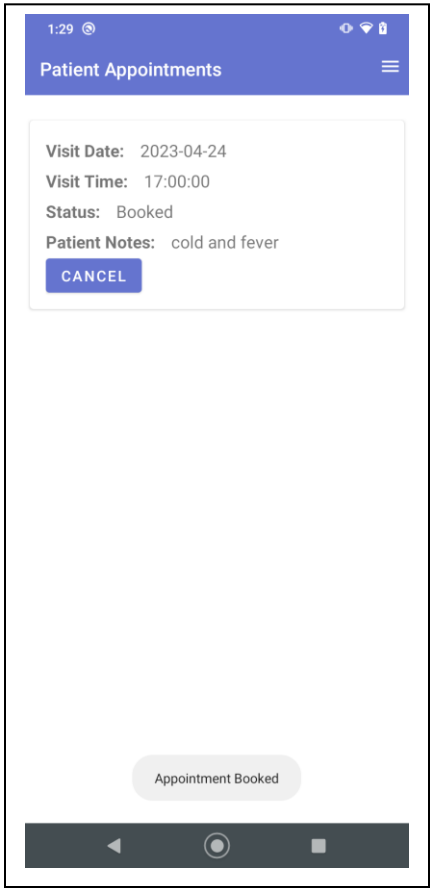




EUC 6.1: Patient books an appointment.

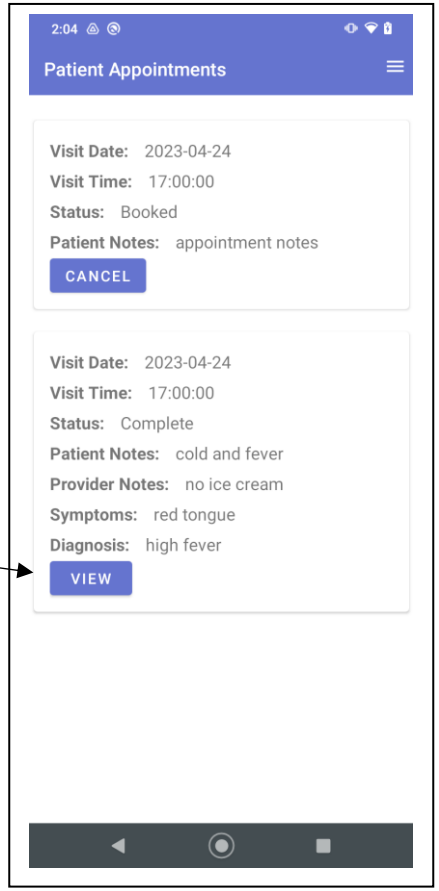
Precondition: This use case assumes that the Patient user is on the homepage.	
Actor: Patient	System: EHR
1. **TUCBW** the patient selects a date and time slot, enters notes, and clicks on the 'Book' button. 3. **TUCEW** the patient being able to view the confirmed appointment and the message.	0. System displays the patient homepage on which the form to book an appointment is displayed with fields to select a date, time slot, and a field to enter notes. *2. System creates an appointment and displays 'Appointment Booked' message.
Post Condition: The booked appointment details are visible to the patient and is available to cancel as well.	

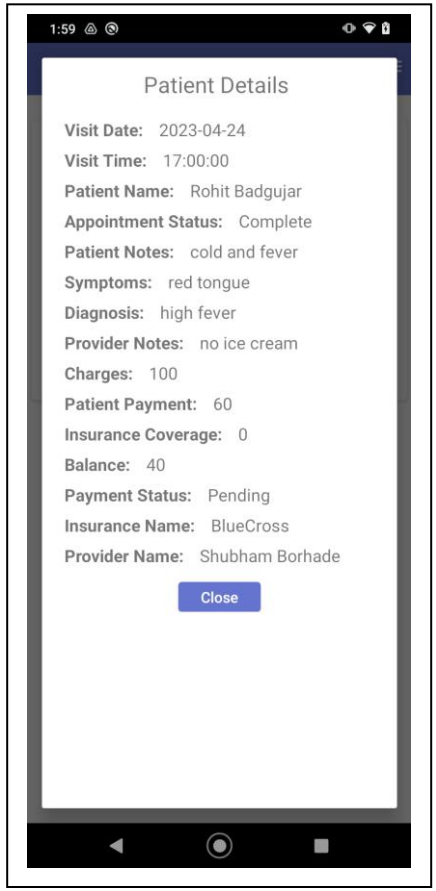




EUC 6.2: Patient views an appointment.

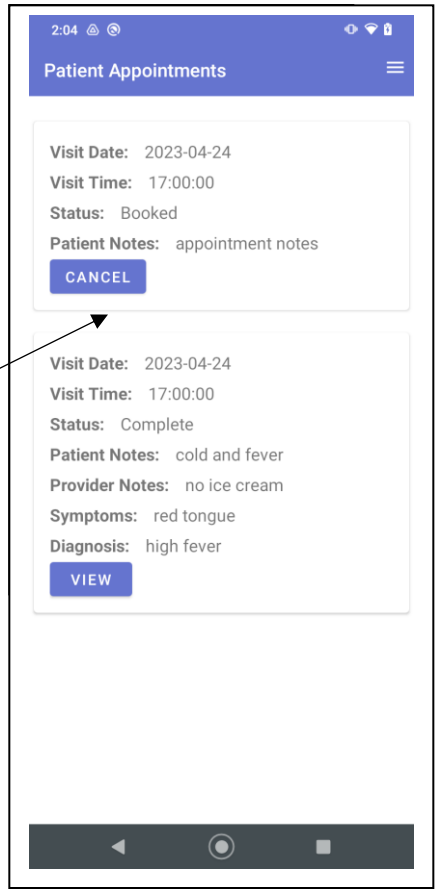
Precondition: This use case assumes that the patient is on the appointments page.	
Actor: Patient	System: EHR
1. **TUCBW** the patient clicks on the 'View' button of an appointment. 3. **TUCEW** the patient being able to view the details of an appointment.	0. System displays the appointment page on which all the appointments belonging to the patient are displayed. *2. System displays all the details associated with the appointment.
Post Condition: The appointment details are visible to the patient.	

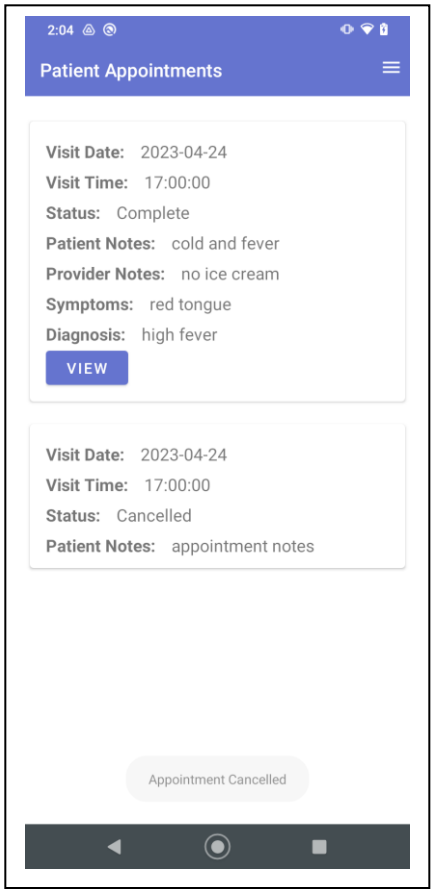




EUC 6.3: Patient cancels an appointment.

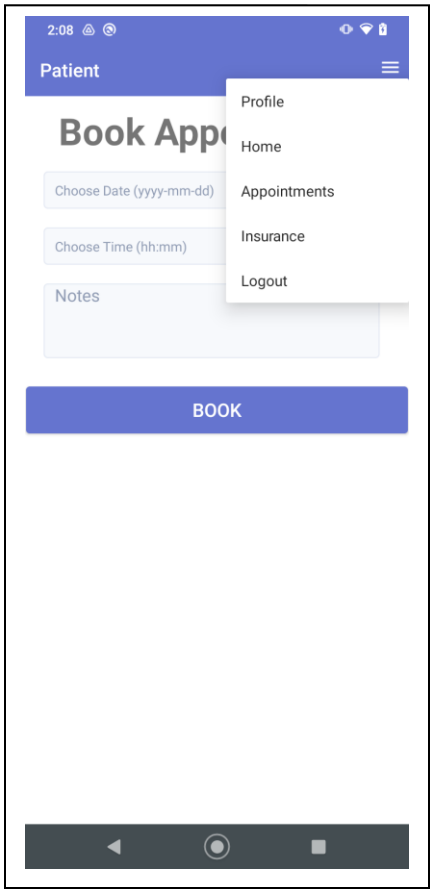
Precondition: This use case assumes that patient is on the appointment page.	
Actor: Patient	System: EHR
1. **TUCBW** patient clicks on the 'Cancel' button to cancel an appointment. 3. **TUCEW** the patient sees the appointment being cancelled.	0. System displays the appointment page. *2. Systems updates the appointment status to 'Cancelled' and shows the message 'Appointment Cancelled'.
Post Condition: The patient able to see that the appointment status being updated to Cancelled.	





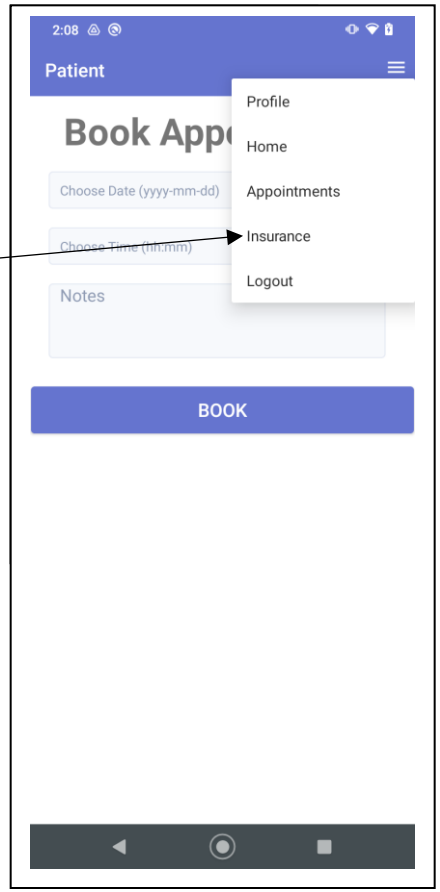
EUC 7: Patient manages insurance.

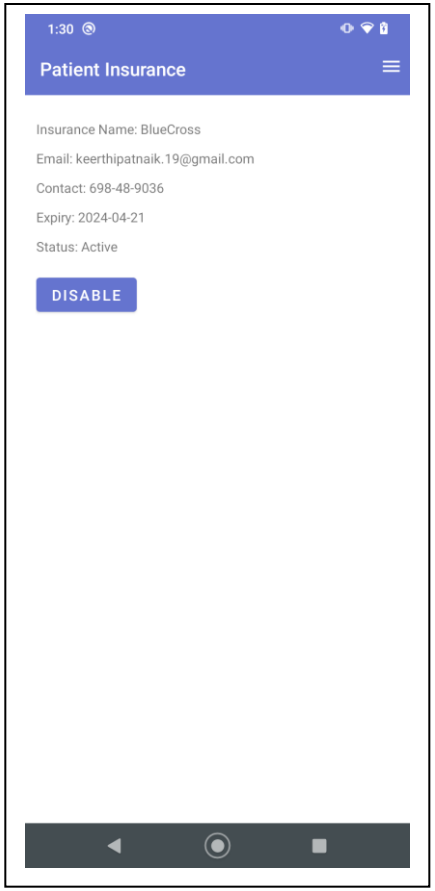
Precondition: This use case assumes that the Patient user is on the homepage.	
Actor: Patient	System: EHR
1. **TUCBW** patient may click on the ‘Insurance’ from the menu bar. 3. **TUCEW** the patient sees the insurance details.	0. System displays the home page. 2. System shall display the insurance details if the patient has an active insurance or display the add insurance form if the patient doesn’t have an insurance.
Post Condition: The patient able to add/view insurance details.	



EUC 7.1: Patient views insurance details.

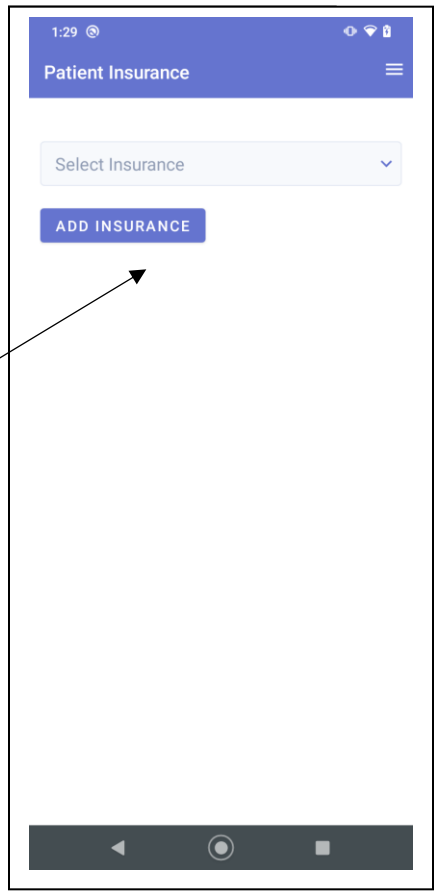
Precondition: This use case assumes that the Patient user is on the homepage.	
Actor: Patient	System: EHR
1. **TUCBW** patient clicks on the 'Insurance' from the menu bar. 3. **TUCEW** the patient sees the insurance details.	0. System displays the homepage. *2. System displays the insurance details.
Post Condition: The patient able to add/view insurance details.	

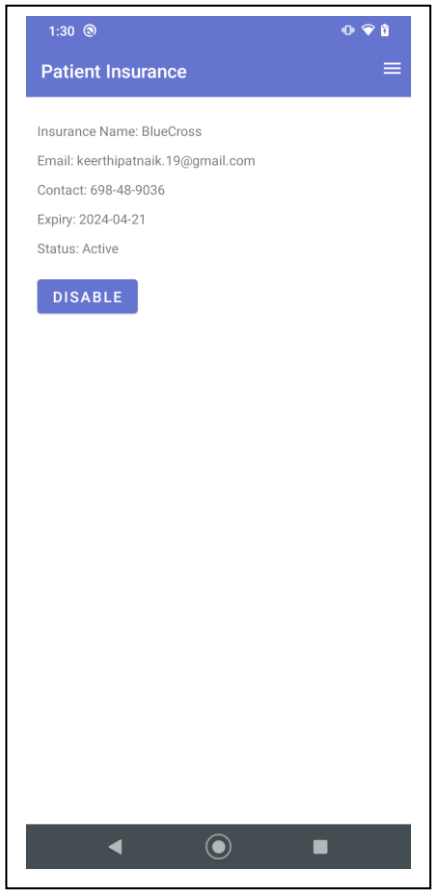




EUC 7.2: Patient adds insurance.

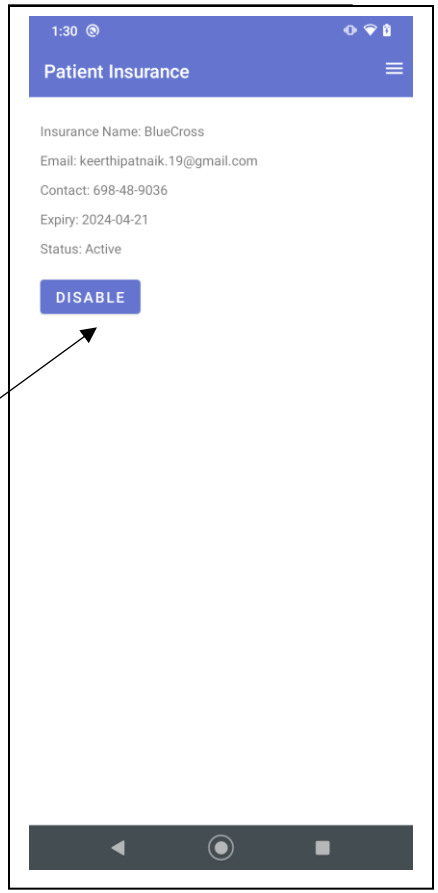
Precondition: This use case assumes that the Patient user is on the insurance page.	
Actor: Patient	System: EHR
1. **TUCBW** patient selects one of the insurance companies and clicks on the ‘Add Insurance’ button. 3. **TUCEW** the patient sees the saved insurance details.	0. System displays the insurance page. *2. System adds the insurance details for the patient in the database.
Post Condition: The patient able to view insurance details and disable the same.	

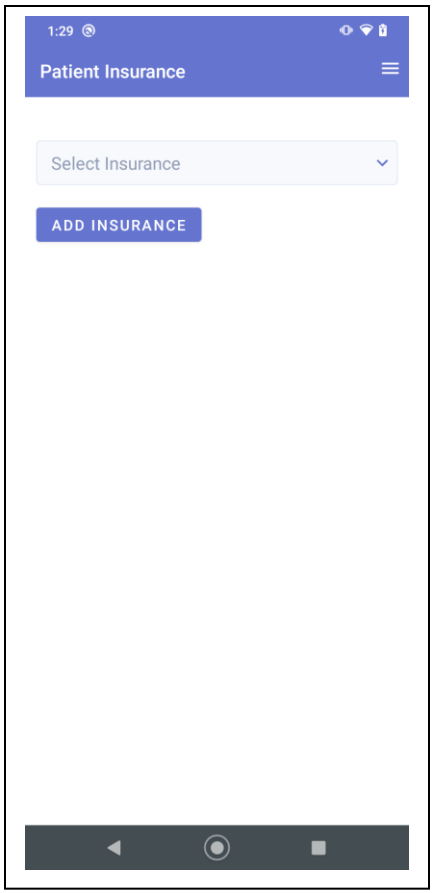




EUC 7.3: Patient disables insurance.

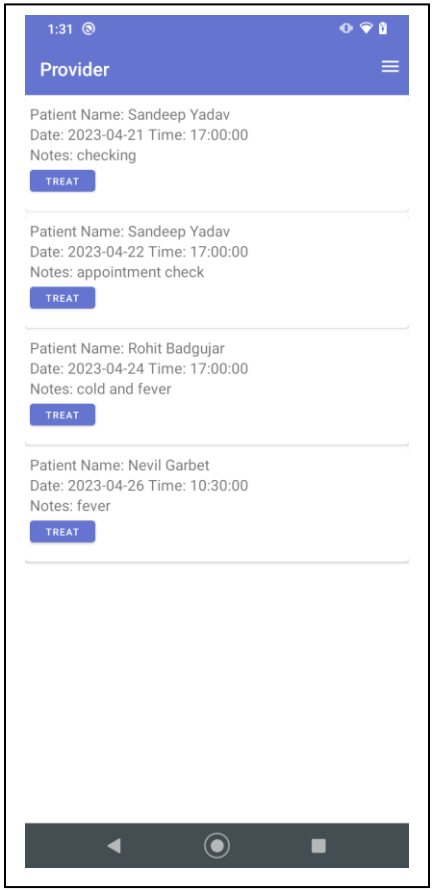
Precondition: This use case assumes that the Patient user is on the insurance page.	
Actor: Patient	System: EHR
1. **TUCBW** patient clicks on the 'Disable' button. 3. **TUCEW** the patient not able to see the insurance details.	0. System displays the insurance page. *2. System updates the insurance status to 'Inactive' in the database and shows the message 'Insurance Disabled'.
Post Condition: The patient able to add new insurance.	





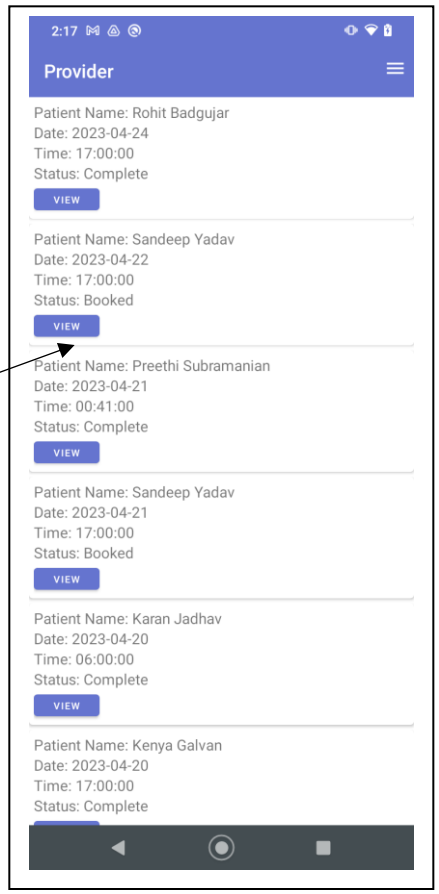
EUC 8: Provider manages appointment.

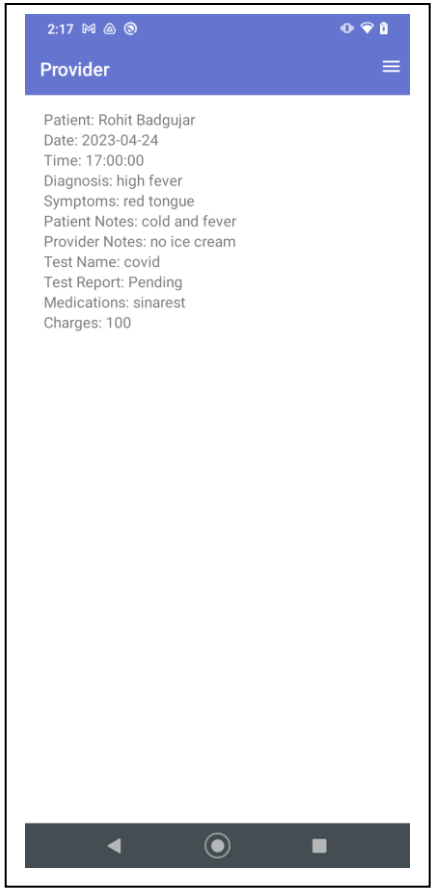
Precondition: This use case assumes that the Provider user is already logged into the system.	
Actor: Provider	System: EHR
1. **TUCBW** the provider may click on the Schedule option from the menu bar. 3. Provider may click on the Treat button for a particular entry of the scheduled visits. 5. **TUCEW** the provider viewing and adding visit information.	0. System displays the provider homepage. 2. System shall display the schedule page. 4. System shall display the forms to add data related to the visit.
Post Condition: Provider being able to manage the appointment by viewing and adding visit information under schedule page.	



EUC 8.1: Provider views appointment details.

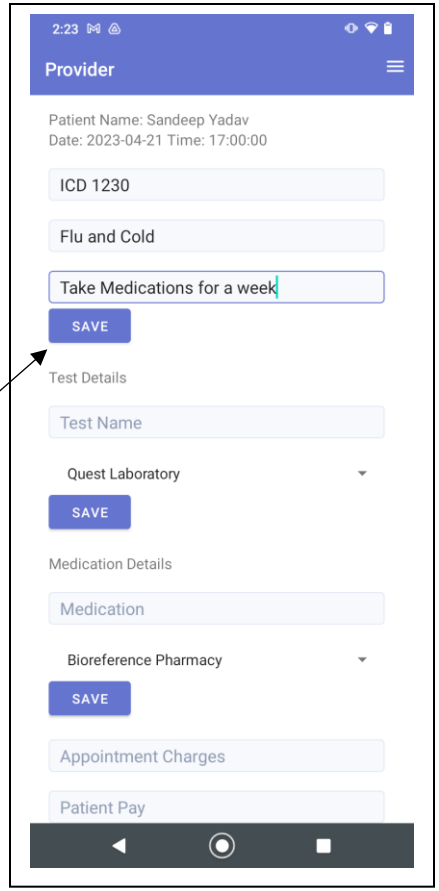
Precondition: This use case assumes that the provider is on the appointments page.	
Actor: Provider	System: EHR
1. **TUCBW** the provider clicks on the 'View' button of a particular appointment from the list of appointments. 3. **TUCEW** the provider being able to view the details of an appointment.	0. System displays the appointment page on which all the appointments are displayed. *2. System displays all the details associated with the appointment.
Post Condition: The appointment details are visible to the provider.	

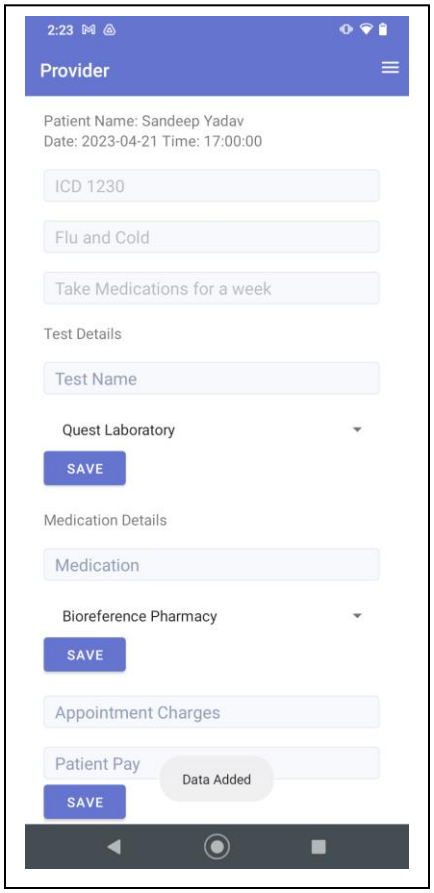




EUC 8.2: Provider adds diagnosis, symptoms, and notes.

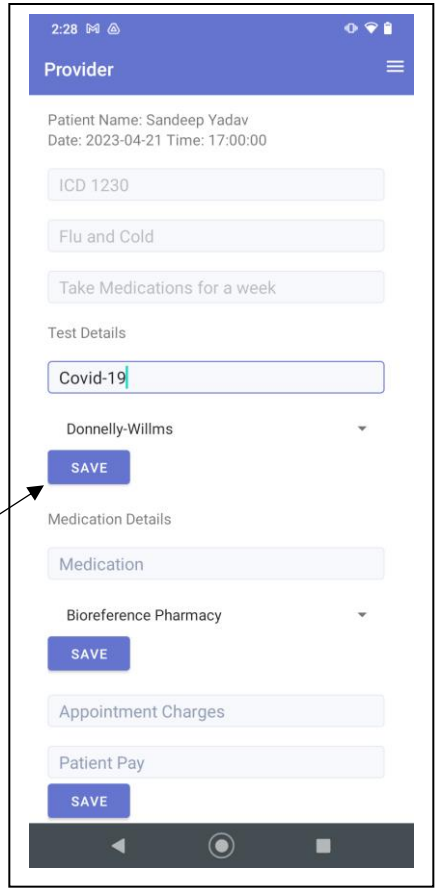
Precondition: This use case assumes that the provider has clicked the 'Treat' button on the schedule page and now system has displayed the appointment page that lists the fields to enter the appointment information.	
Actor: Provider	System: EHR
1. **TUCBW** the provider enters the symptoms, diagnosis, provider notes in the input fields, and clicks on the 'Save' button. 3. **TUCEW** the provider being able to see the system message and the data being saved in the system.	0. System displays the appointment page with fields to enter the information: Symptoms, Diagnosis, and Provider Notes. *2. System saves the information and displays a message 'Data Added'.
Post Condition: Provider being able to see the data being saved in the system.	

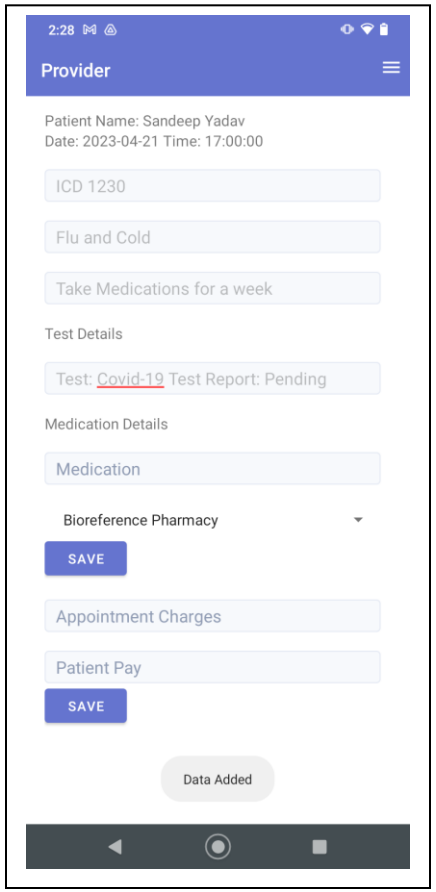




EUC 8.3: Provider assign lab test.

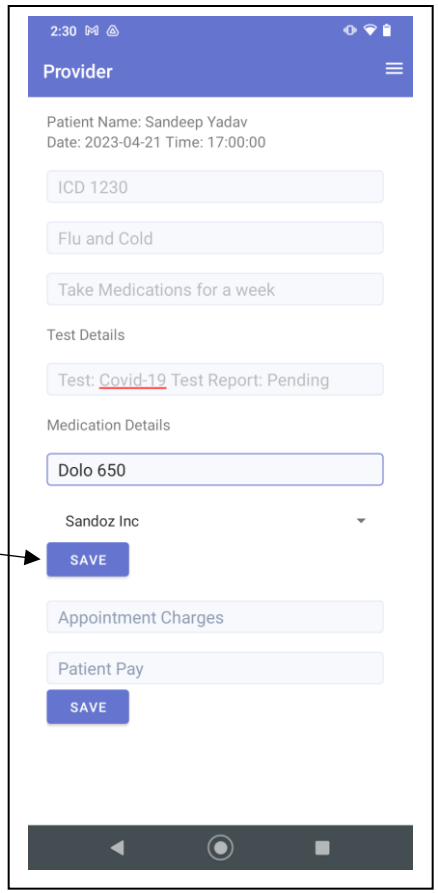
Precondition: This use case assumes that the provider has clicked the 'Treat' button on the schedule page and now system has displayed the appointment page that lists the fields to enter the appointment information.	
Actor: Provider	System: EHR
1. **TUCBW** the provider enters the test name, selects a laboratory from the dropdown, and clicks on the 'Save' button. 3. **TUCEW** the provider being able to see the system message and the data being saved in the system.	0. System displays the appointment page with fields to enter the information: Test Name and select the Laboratory. *2. System saves the information and displays a message 'Data Added'.
Post Condition: Provider being able to see the data being saved in the system.	

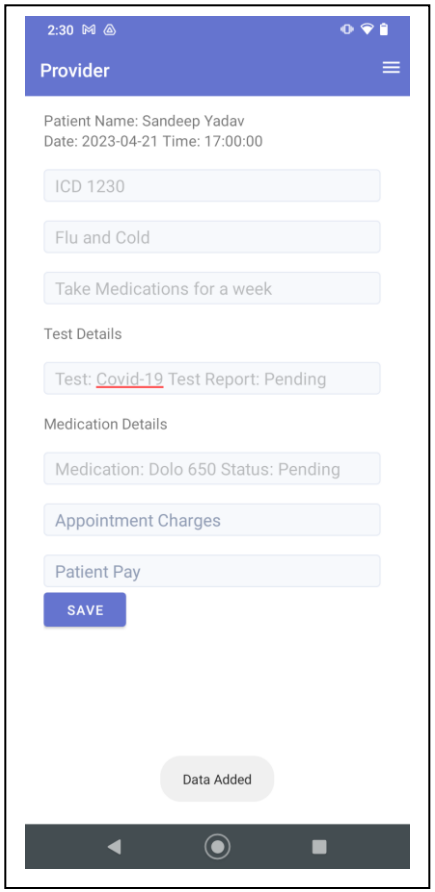




EUC 8.4: Provider prescribes medication.

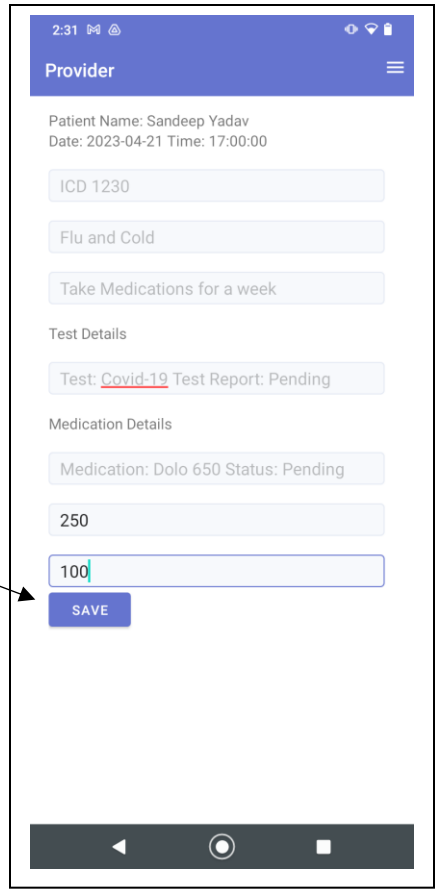
Precondition: This use case assumes that the provider has clicked the 'Treat' button on the schedule page and now system has displayed the appointment page that lists the fields to enter the appointment information.	
Actor: Provider	System: EHR
1. **TUCBW** the provider enters the medication, selects a pharmacy from the dropdown, and clicks on the 'Save' button. 3. **TUCEW** the provider being able to see the system message and the data being saved in the system.	0. System displays the appointment page with fields to enter the information: Medication and select the Pharmacy. *2. System saves the information and displays a message 'Data Added'.
Post Condition: Provider being able to see the data being saved in the system.	

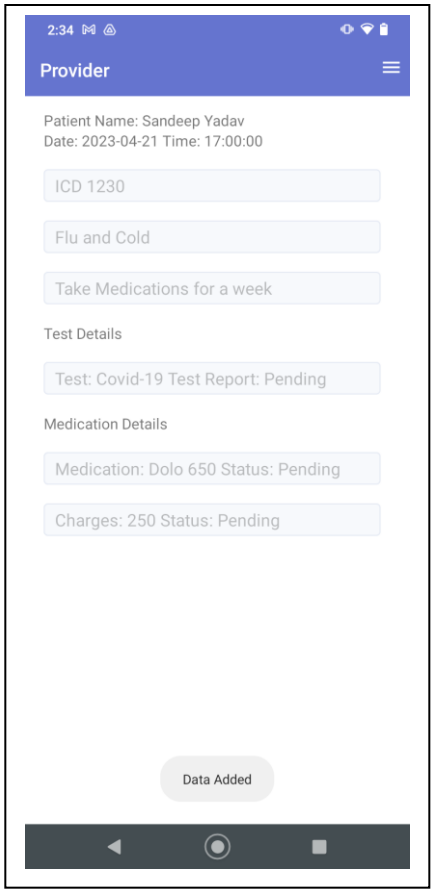




EUC 8.5: Provider adds appointment charges.

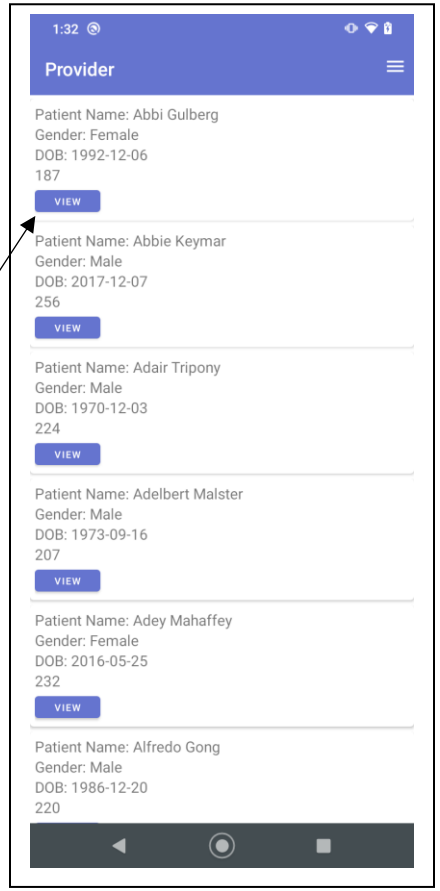
Precondition: This use case assumes that the provider has clicked the 'Treat' button on the schedule page and now system has displayed the appointment page that lists the fields to enter the appointment information.	
Actor: Provider	System: EHR
1. **TUCBW** the provider enters the charges and clicks on the 'Save' button. 3. **TUCEW** the provider being able to see the system message and the data being saved in the system.	0. System displays the appointment page with a field to enter the charges. *2. System saves the information and displays a message 'Data Added'.
Post Condition: Provider being able to see the data being saved in the system.	

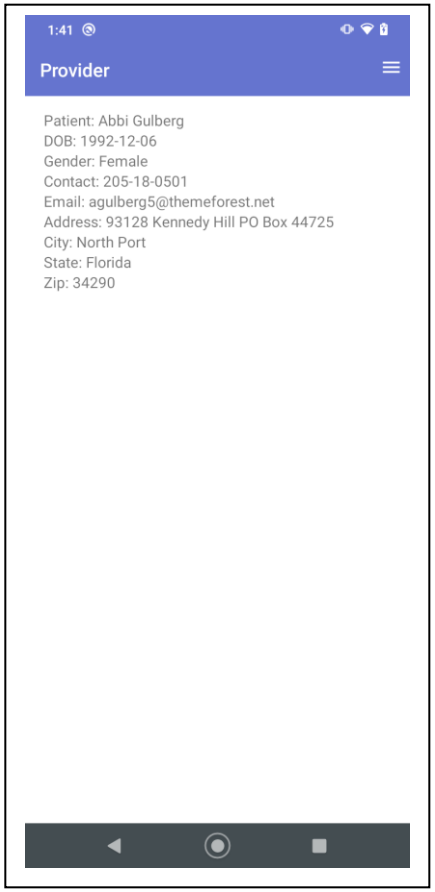




EUC 9: Provider views patient details

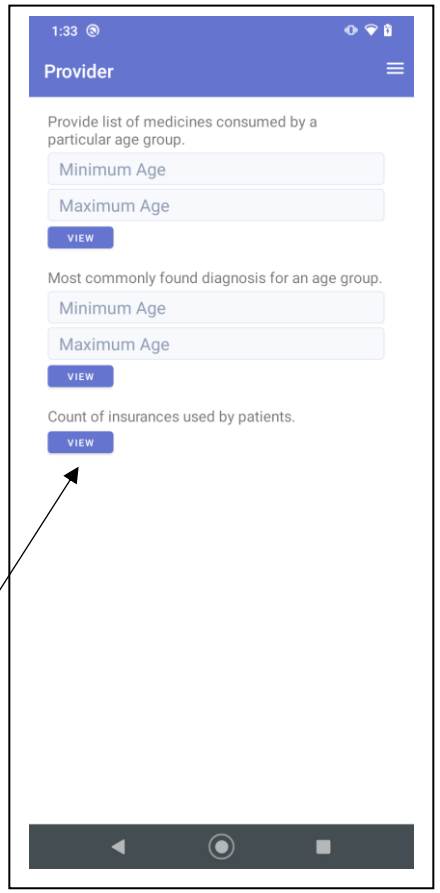
Precondition: This use case assumes that the provider is on the patients list page.	
Actor: Provider	System: EHR
1. **TUCBW** the provider clicks on the 'View' button of a particular patient entry from the list of patients. 3. **TUCEW** the provider being able to view the details of a patient.	0. System displays the patients list page on which all the patients are displayed. *2. System displays all the details associated with the patient.
Post Condition: The patient details are visible to the provider.	

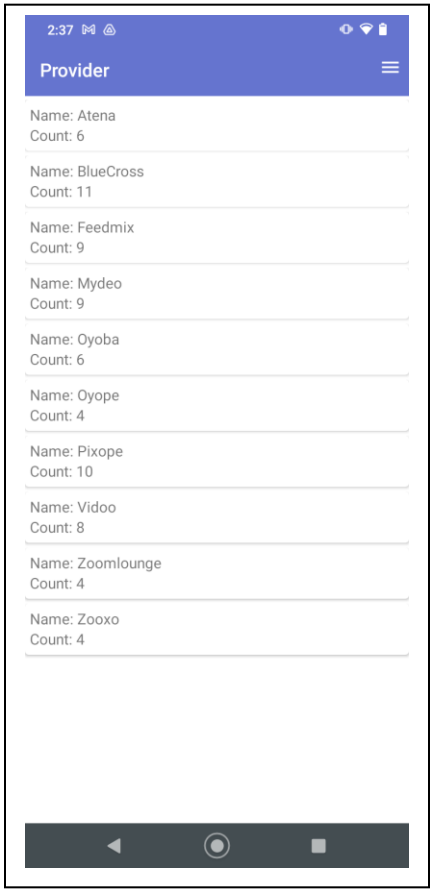




EUC 10: Provider performs analytics

Precondition: This use case assumes that the provider is on the analytics page.	
Actor: Provider	System: EHR
1. **TUCBW** the provider performs one of the following: a. Entering the minimum and maximum age to check the list of medicines prescribed for a particular age group and clicking on the 'View' button. b. Entering the minimum and maximum age to check the list of commonly found diagnosis for a particular age group and clicking on the 'View' button. c. Clicking on the 'View' button to view the count of insurances used by patients.	0. System displays the analytics page on which there are options to perform the following analytics: a. To view a list of medicines prescribed for a particular age group. b. Commonly found diagnosis for an age group. c. Count of insurances used by patients.
	*2. System displays all the details for the provider selected option.
3. **TUCEW** the provider being able to view the details for analytics selection.	
Post Condition: The provider being able to view the details for analytics selection	



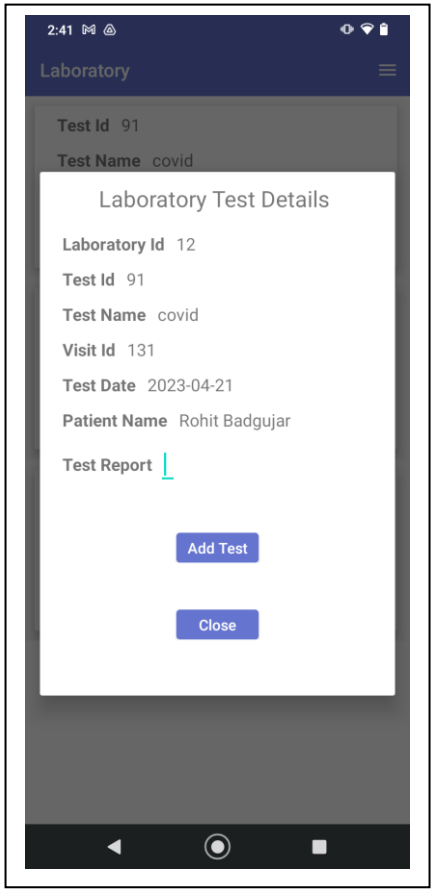


EUC 11 Laboratory manages lab test

Precondition: This use case assumes that the laboratory user is on the home page.	
Actor: Laboratory	System: EHR
1. **TUCBW** the laboratory user may click on the 'Tests' option from the menu bar.	0. System displays laboratory home page. 2. System shall display the pending and completed lab tests.
3. The laboratory user may click on the 'Home' option from the menu bar.	4.System shall display the pending lab tests.
5. **TUCEW** the laboratory user able to view the pending lab tests.	
Post Condition: The laboratory user being able to view the details of the lab tests.	

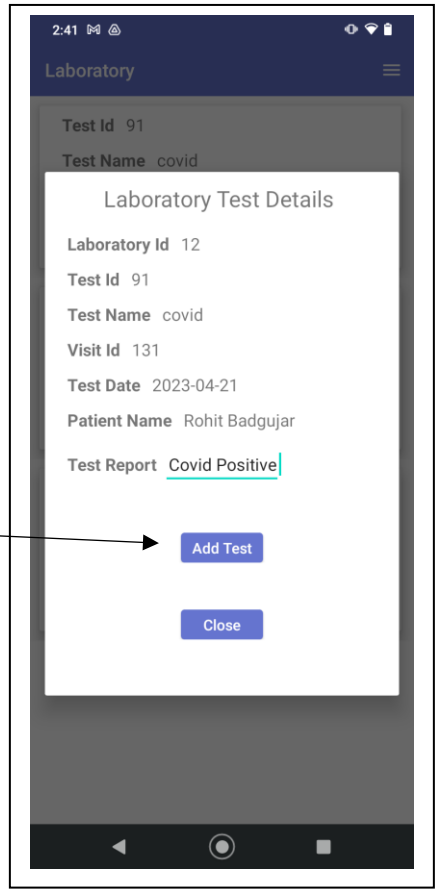
EUC 11.1 Laboratory view test details

Precondition: This use case assumes that the laboratory user is on the home page.	
Actor: Laboratory	System: EHR
1. **TUCBW** the laboratory user clicks on the view button for a particular pending lab test. 3. **TUCEW** the laboratory user being able to view the details of lab test.	0. System displays all the pending lab tests that are assigned to the laboratory on the homepage. *2. System displays a pop-up with all the details associated with the lab test.
Post Condition: The laboratory user being able to view the details of lab test.	



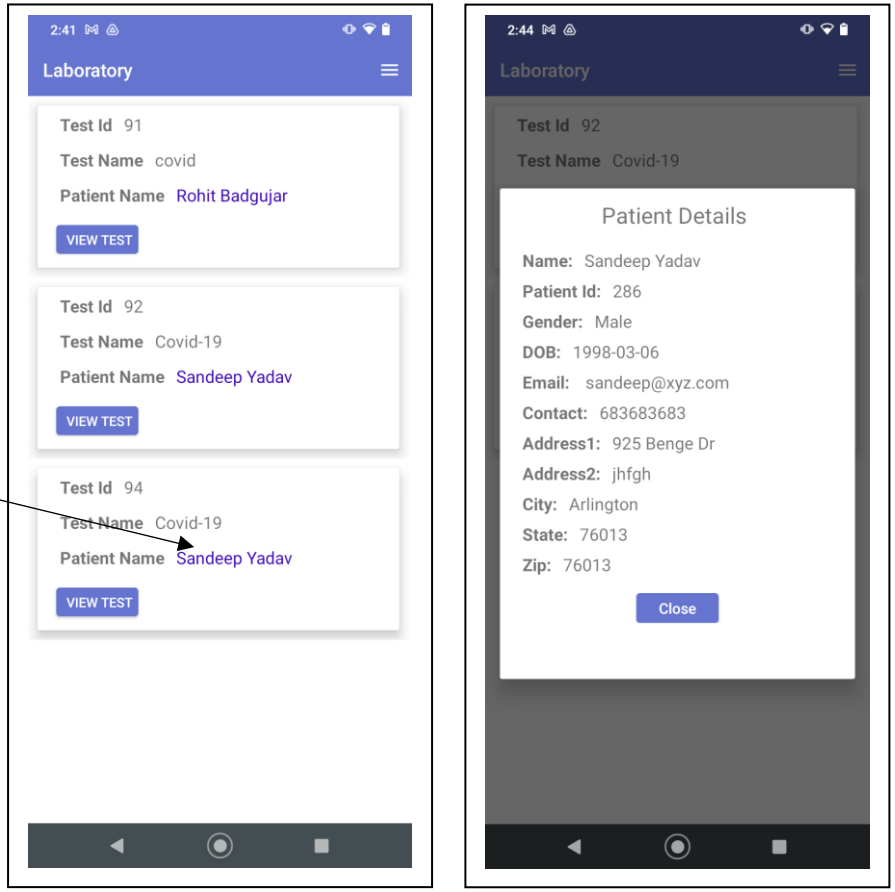
EUC 11.2 Laboratory add test results

Precondition: This use case assumes that the laboratory user has clicked the view button for a pending lab test and able to see the details.	
Actor: Laboratory	System: EHR
1. **TUCBW** the laboratory user enters the lab test results in the field and clicks on the 'Add Test' button. 3. **TUCEW** the laboratory user being able to view the message and doesn't see the lab test in the pending list.	0. System displays the details associated to the pending lab test with a field to enter the lab test results and button to save. *2. System saves the lab test results in the database and displays the message 'Test added successfully.'
Post Condition: The laboratory user being able to view the message and doesn't see the lab test in the pending list.	



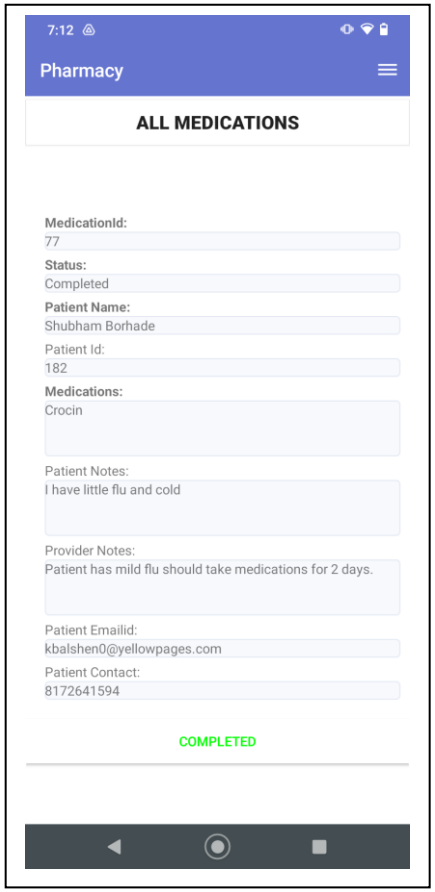
EUC 12 Laboratory views patient details

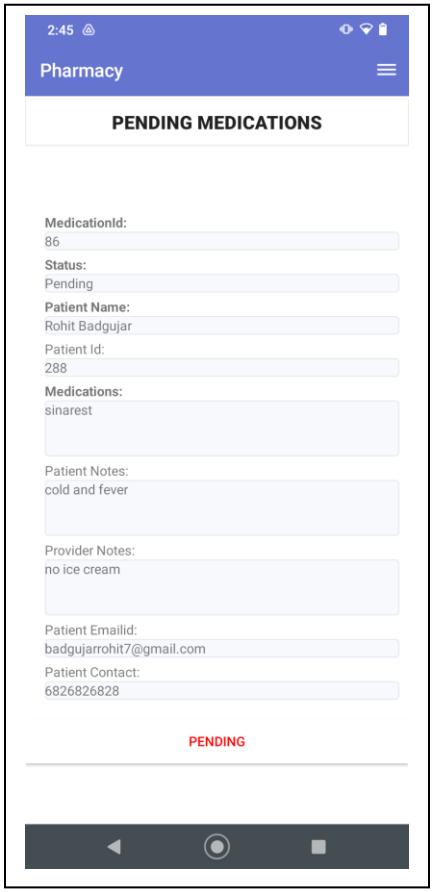
Precondition: This use case assumes that the laboratory user is on the home page.	
Actor: Laboratory	System: EHR
1. **TUCBW** the laboratory user clicks on patient name which is displayed on particular pending lab test card. 3. **TUCEW** the laboratory user being able to view the details of patient.	0. System displays all the pending lab tests that are assigned to the laboratory on the homepage. *2. System displays a pop-up with all the patient details.
Post Condition: The laboratory user being able to view the details of patient.	



EUC 13 Pharmacy manages prescribed medication

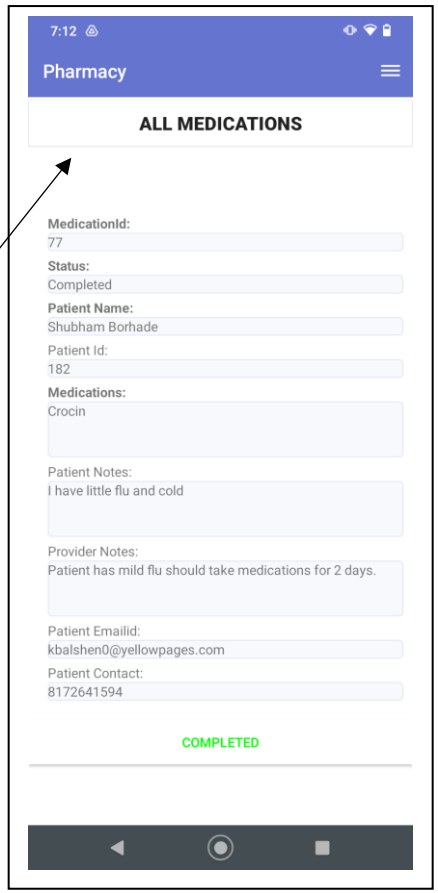
Precondition: This use case assumes that the pharmacy user is on the home page.	
Actor: Pharmacy	System: EHR
	0. System displays the home page.
1. **TUCBW** the pharmacy user viewing the home page.	
	2. System shall display all medication requests details.
3. The pharmacy user may click on the medication handover status button.	
	4. System shall update the status for the particular medication request.
5. **TUCEW** the pharmacy user able to perform viewing and updating the medication requests.	
Post Condition: The pharmacy user able to perform viewing and updating the medication requests.	

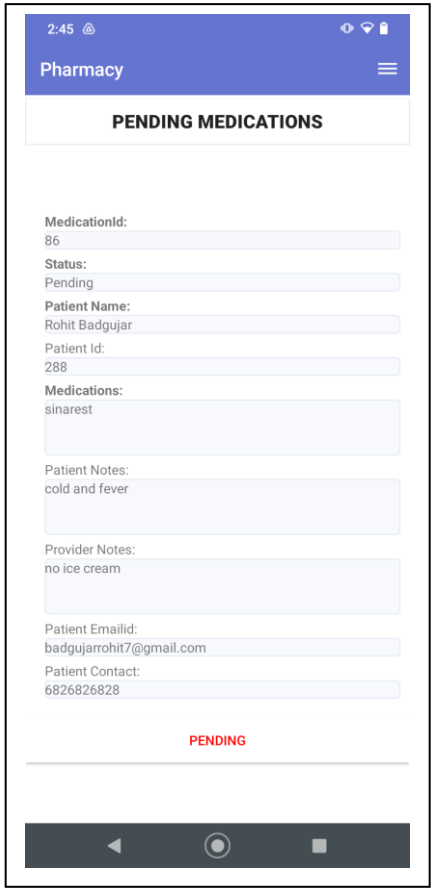




EUC 13.1 Pharmacy views medication details

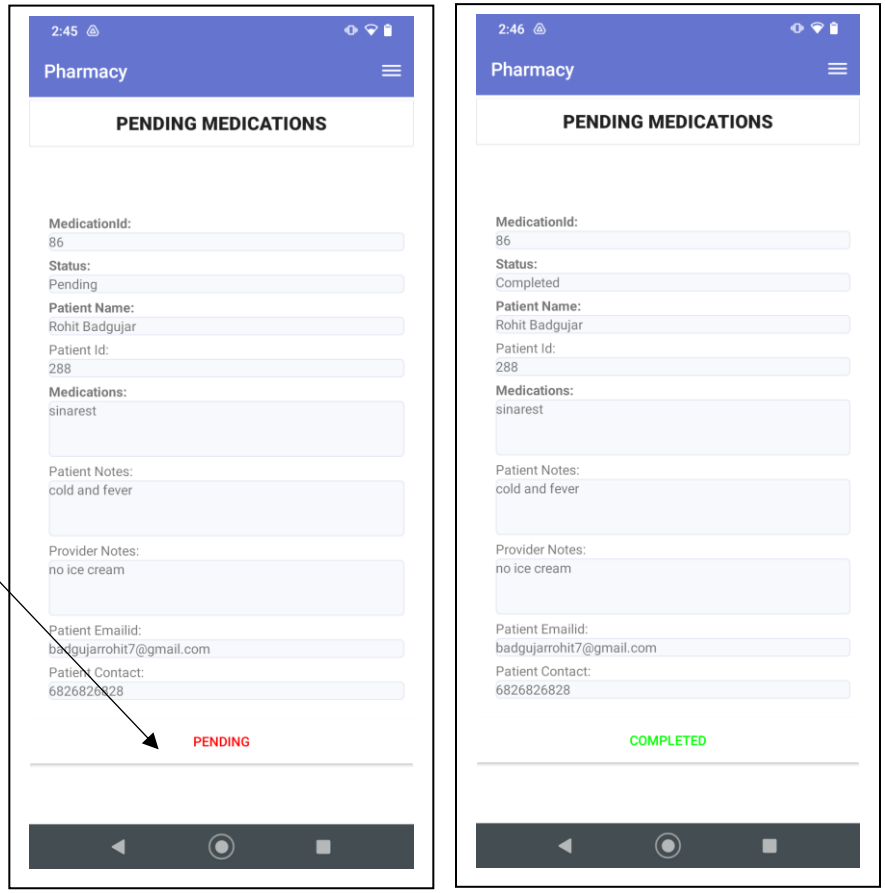
Precondition: This use case assumes that the pharmacy user is on the home page.	
Actor: Pharmacy	System: EHR
1. **TUCBW** the pharmacy user clicks on the 'All Medications' button. 3. **TUCEW** the pharmacy user able to see a list of pending medications requests.	0. System displays the pharmacy user home page. *2. System displays a list of all pending medications requests.
Post Condition: The pharmacy user able to see a list of pending medications requests.	





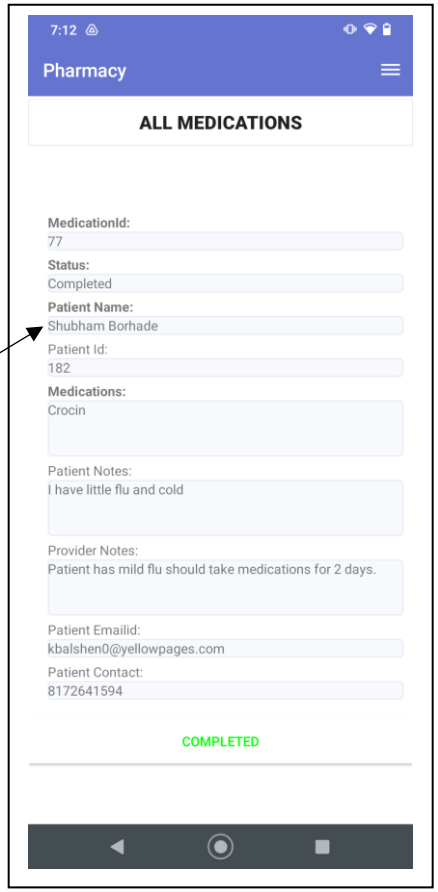
EUC 13.2 Pharmacy updates medication handover status

Precondition: This use case assumes that the pharmacy user is on the home page and have clicked on ‘All Medications’ button.	
Actor: Pharmacy	System: EHR
1. **TUCBW** the pharmacy user clicks on the ‘Pending’ button of a particular medication entry. 3. **TUCEW** the medication status being updated and pharmacy user being able to see the status as ‘Completed’	0. System displays a list of pending medication requests. *2. System updates the status to ‘Completed’ for that medication entry in the database. And displays the status as ‘Completed’ in the UI.
Post Condition: Pharmacy user being able to see the status as ‘Completed’ and the entry moves to the Pending tab.	



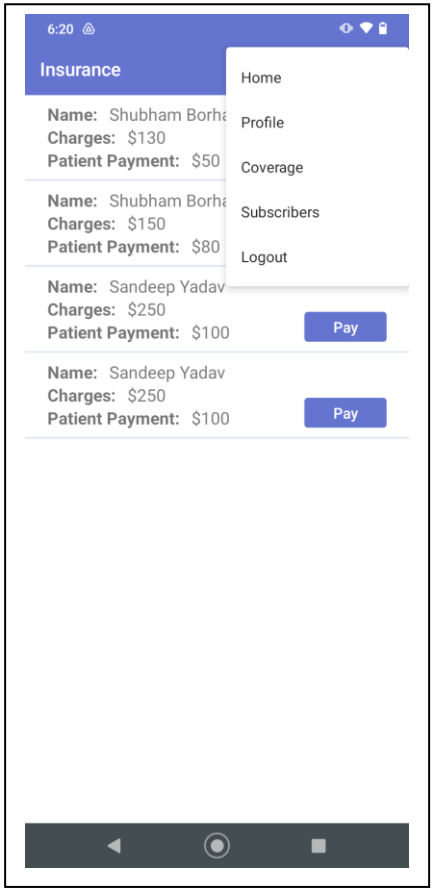
EUC 14 Pharmacy views patient details

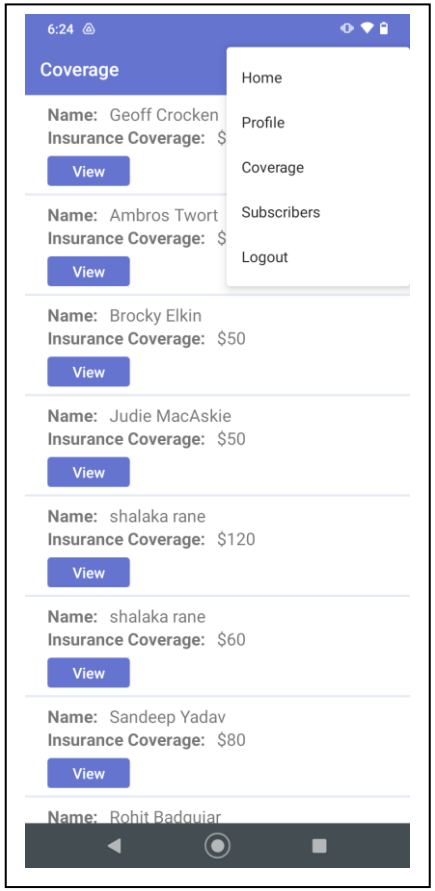
Precondition: This use case assumes that the pharmacy user is on the home page.	
Actor: Pharmacy	System: EHR
1. **TUCBW** the pharmacy user viewing the home page. 3. **TUCEW** the pharmacy user able to see patient details for each medication requests.	0. System displays the pharmacy user home page. *2. System displays patient details for each of the medication requests.
Post Condition: The pharmacy user able to see patient details for each medication requests.	



EUC 15 Insurance company manages insurance coverage

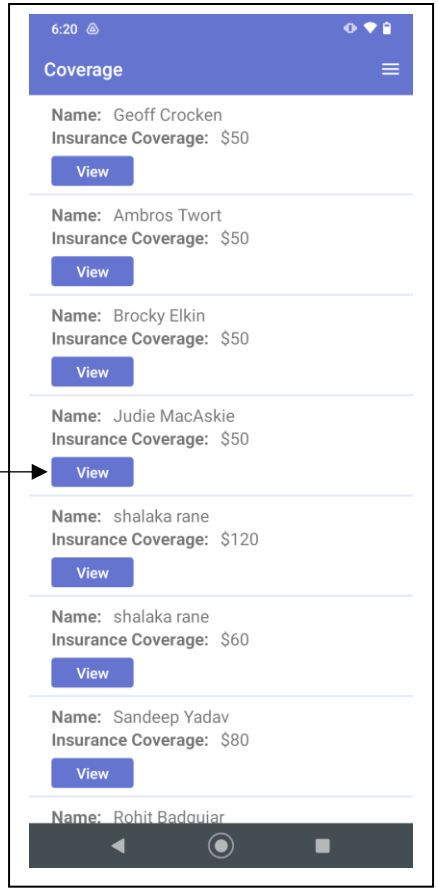
Precondition: This use case assumes that the Insurance Company user is on the home page.	
Actor: Insurance Company	System: EHR
1. **TUCBW** the insurance company user may click on the 'Coverage' option from the menu bar. 3. The insurance company user may click on the 'Home' option from the menu bar. 5. **TUCEW** insurance company able to see the pending coverages.	0. System displays the home page. 2. System shall display all the coverages. 4. System shall display the pending coverages.
Post Condition: insurance company user being able to see the insurance coverage details.	

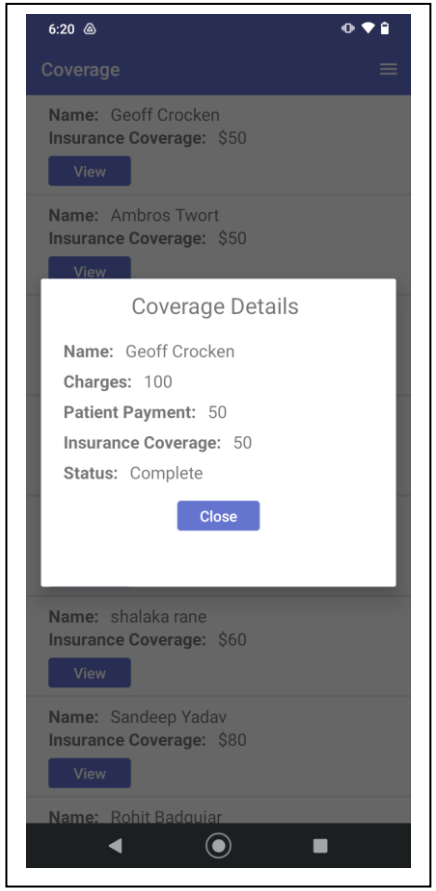




EUC 15.1 Insurance company views insurance coverage

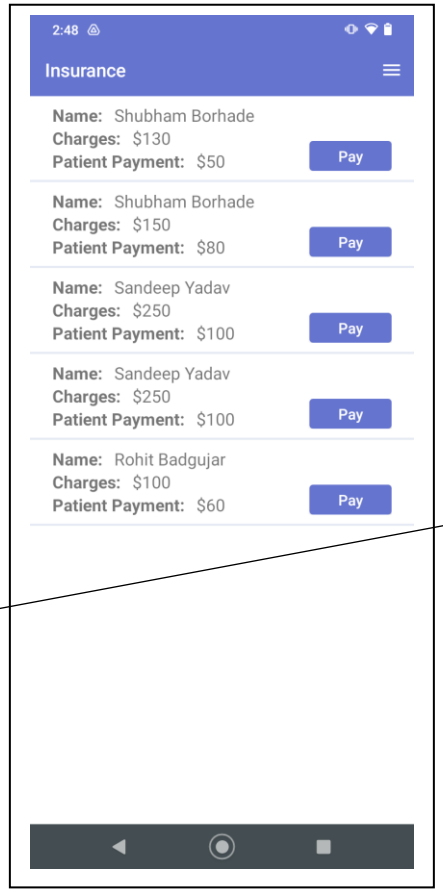
Precondition: This use case assumes that the Insurance Company user is on the coverage page.	
Actor: Insurance Company	System: EHR
1. **TUCBW** the insurance company user clicks on the 'View' button of a particular coverage entry. 3. **TUCEW** insurance company user being able to see all the details associated to the select coverage entry.	0. System displays the coverage page with list of coverages the insurance company has paid so far. *2. System displays a pop up with all the details associated to the select coverage entry.
Post Condition: insurance company user being able to see all the details associated to the select coverage entry.	

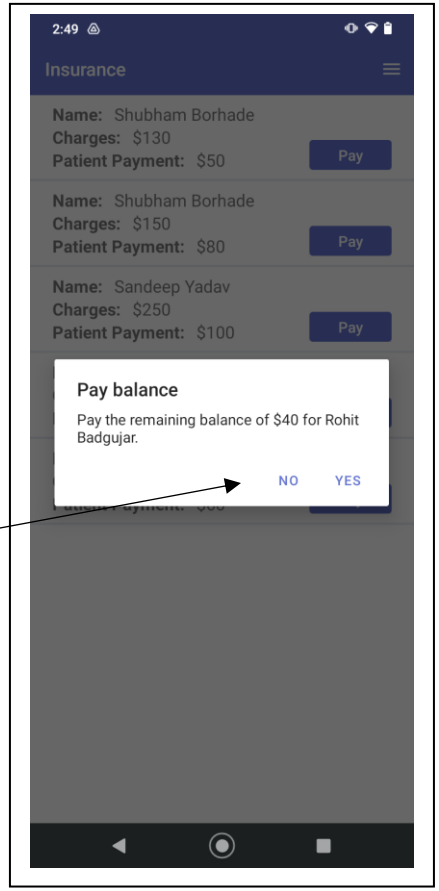




EUC 15.2 Insurance company updates insurance coverage

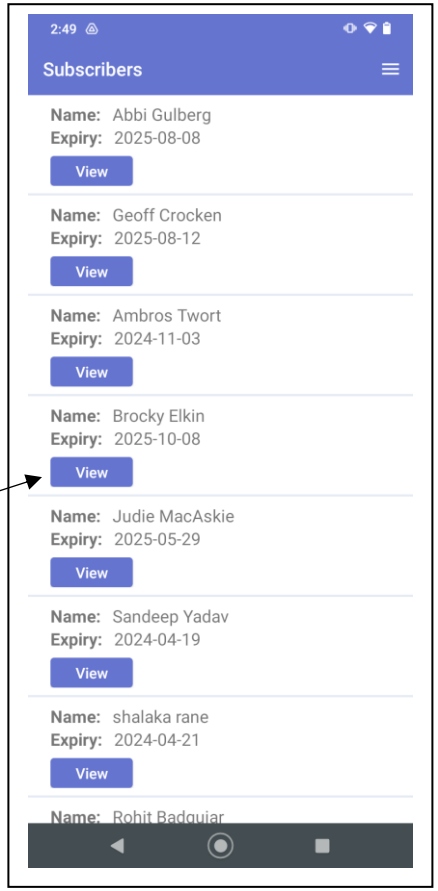
Precondition: This use case assumes that the Insurance Company user is on the homepage	
Actor: Insurance Company	System: EHR
1. **TUCBW** the insurance company user clicks on the 'Pay' button. 3. Insurance Company user able to see the balance amount and clicks on the 'Yes' option. 5. **TUCEW** insurance coverage and balance amounts updated in the database and the insurance company user not able to see the pending coverage on the homepage.	0. System displays the homepage with all the pending coverages that are yet to be paid. 2. System displays a dialog box with the balance amount that has to be paid by the insurance company and options 'No' & 'Yes'. *4. System updates the balance amount to 0 and insurance coverage in the database.
Post Condition: Insurance company user not able to see the pending coverage on the homepage	

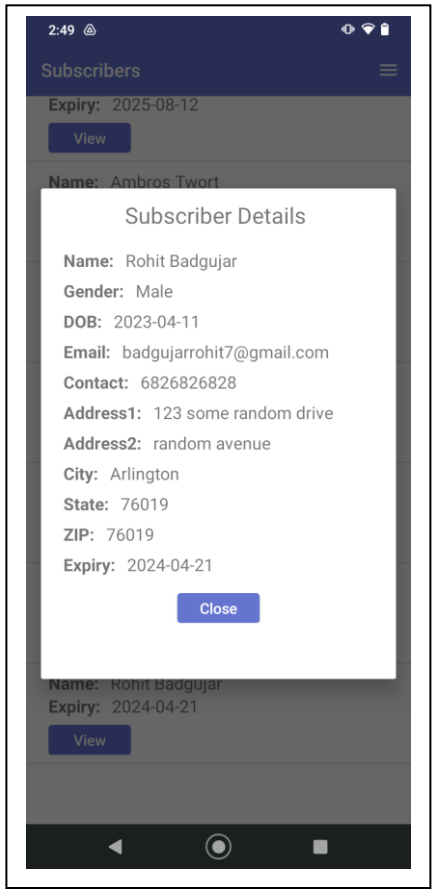




EUC 16 Insurance company views patient details

Precondition: This use case assumes that the Insurance Company user is on the subscribers page.	
Actor: Insurance Company	System: EHR
1.**TUCBW** the insurance company user clicks on the 'View' button of a particular patient entry. 3. **TUCEW** insurance company user being able to see all the details associated to the selected patient entry.	0. System displays the subscribers page with all the patients that have taken insurance from the respective insurance company. *2. System displays a pop up with all the details associated to the selected patient entry.
Post Condition: insurance company user being able to see all the details associated to the selected patient entry.	



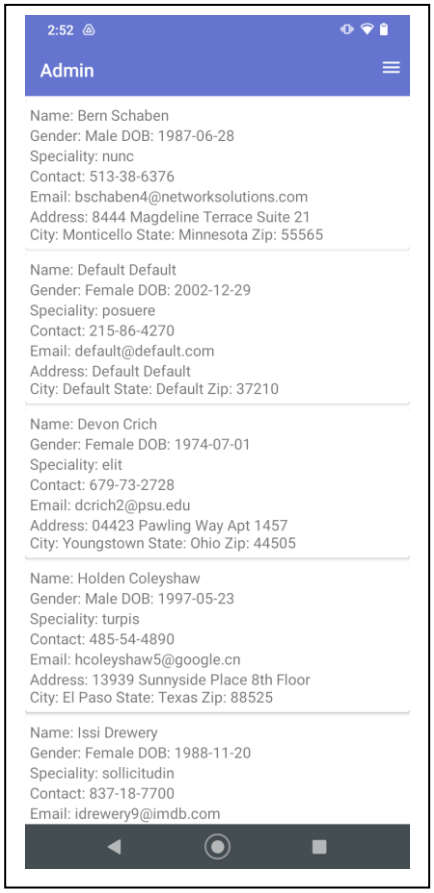


EUC 17 Admin manages users (Provider, Laboratory, Pharmacy, Insurance Company)

Precondition: This use case assumes that the Admin user is already logged into the system.	
Actor: Admin	System: EHR
1. **TUCBW** the admin clicks on the 'Add User' from the menu bar.	0. System displays the admin homepage.
	2. System displays the Add User page with options to add Provider and Facilities.
3. Admin able to see the options to add the desired user in the system. Admin clicks on the 'View User' from the menu bar.	
	4. System displays the View User page with options to view Provider, Laboratory, Pharmacy, Insurance Company.
5. **TUCEW** the admin being able to see the options and can select any of them to see the user details.	
Post Condition: Admin being able to manage the users.	

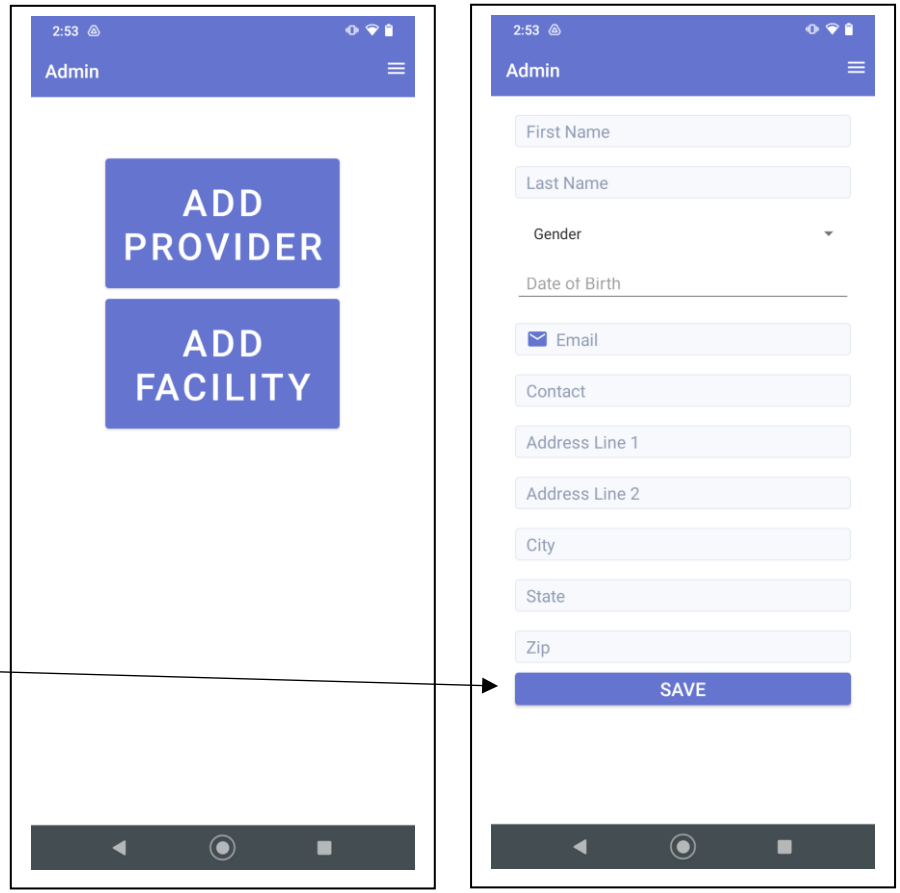
EUC 17.1 Admin views user details

Precondition: This use case assumes that the Admin user is on the View User page	
Actor: Admin	System: EHR
1. **TUCBW** the admin clicks on one of the following options: a. 'View Provider' button. b. 'View Laboratory button. c. 'View Pharmacy' button. d. 'View Insurance Company' button 3. **TUCEW** the admin able to see all the users and their details.	0. System displays the View User page with options to view Provider, Laboratory, Pharmacy, and Insurance Company. *2. System displays all the list of users and their details.
Post Condition: admin able to see all the users and their details.	



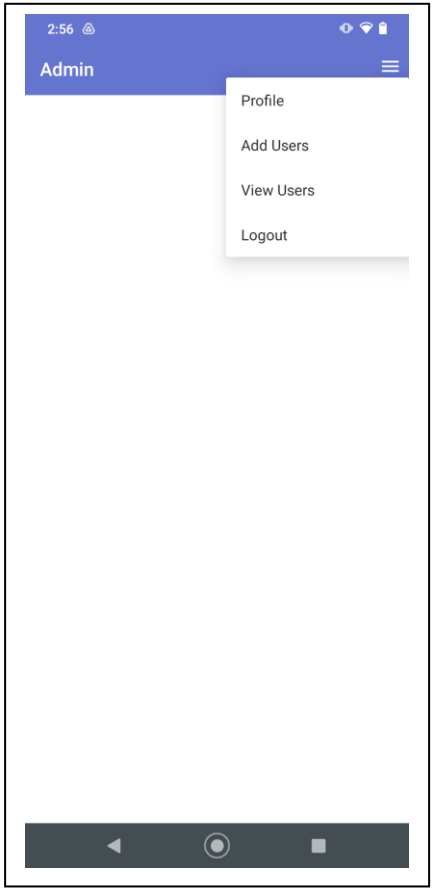
EUC 17.2 Admin adds user

Precondition: This use case assumes that the Admin user is on the Add User page	
Actor: Admin	System: EHR
1. **TUCBW** the admin clicks on one of the following options: a. 'Add Provider' button. b. 'Add Facility' button.	0. System displays the Add User page with options to add Provider and Facilities.
	2. System displays the add user form depending on the admin selected option.
3. Admin enters the user details in the add user form and clicks on the 'Save' button.	
	*4. System saves the user details in the system and displays a message 'Data saved'.
5. **TUCEW** the user details being saved in the system and admin able to see the system provided message.	
Post Condition: User details being saved in the system and admin able to see the system provided message.	



EUC 18: Logout

Precondition: This use case assumes that the user in one of the active activities.	
Actor: User	System: EHR
1. **TUCBW** the user clicks on the 'Logout' from the menu bar 3. **TUCEW** the user being logged out and able to see the login page.	0. System displays the menu. 2. System logs-out the user and redirects to the Login page.
Post Condition: User being able to see the Login page.	



Requirement to Use Case Traceability Matrix (RUCTM)

	Priority Weight	UC1	UC2	UC3	UC4	UC5	UC6	UC6.1	UC6.2	UC6.3	UC6.4	UC6.5	UC7	UC7.1	UC7.2	UC7.3	UC8	UC8.1	UC8.2	UC8.3	UC8.4	UC8.5	UC9	UC10	UC11	UC11.1	UC11.2	UC12	UC13	UC13.1	UC13.2	UC14	UC15	UC15.1	UC15.2	UC16	UC17	UC17.1	UC17.2	UC18
R1	3	X																																						
R1.1	3	X																																						
R1.2	3		X																																					
R1.3	3			X																																				
R1.4	3	X																																						
R2	5				X	X																																		
R2.1	5				X																																			
R2.2	5					X																																		
R3	1						X																																	
R3.1	1							X																																
R3.2	1								X																															
R3.3	1									X																														
R3.4	1										X																													
R3.5	1											X																												
R4	4												X																											
R4.1	4													X																										
R4.2	4														X																									
R4.3	4															X																								
R5	1																X																							
R5.1	1																	X																						
R5.2	1																		X																					
R5.3	1																			X																				
R5.4	1																				X																			
R5.5	1																					X																		
R6	4																						X																	
R7	3																							X																
R8	1																								X															
R8.1	1																									X														
R8.2	1																										X													
R9	5																											X												
R10	1																												X											
R10.1	1																													X										
R10.2	1																														X									
R11	5																															X								
R12	3																																X							
R12.1	3																																	X						
R12.2	3																																		X					
R13	5																																			X				
R14	1																																				X			
R14.1	1																																					X		
R14.2	1																																						X	
R15	5																																							X
	Score	9	3	3	10	10	1	1	1	1	1	1	4	4	4	4	1	1	1	1	1	1	4	3	1	1	1	5	1	1	1	5	3	3	3	5	1	1	1	5

Note: Priority 1 is of highest weightage, ranges between 1 to 5.

Increment Matrix

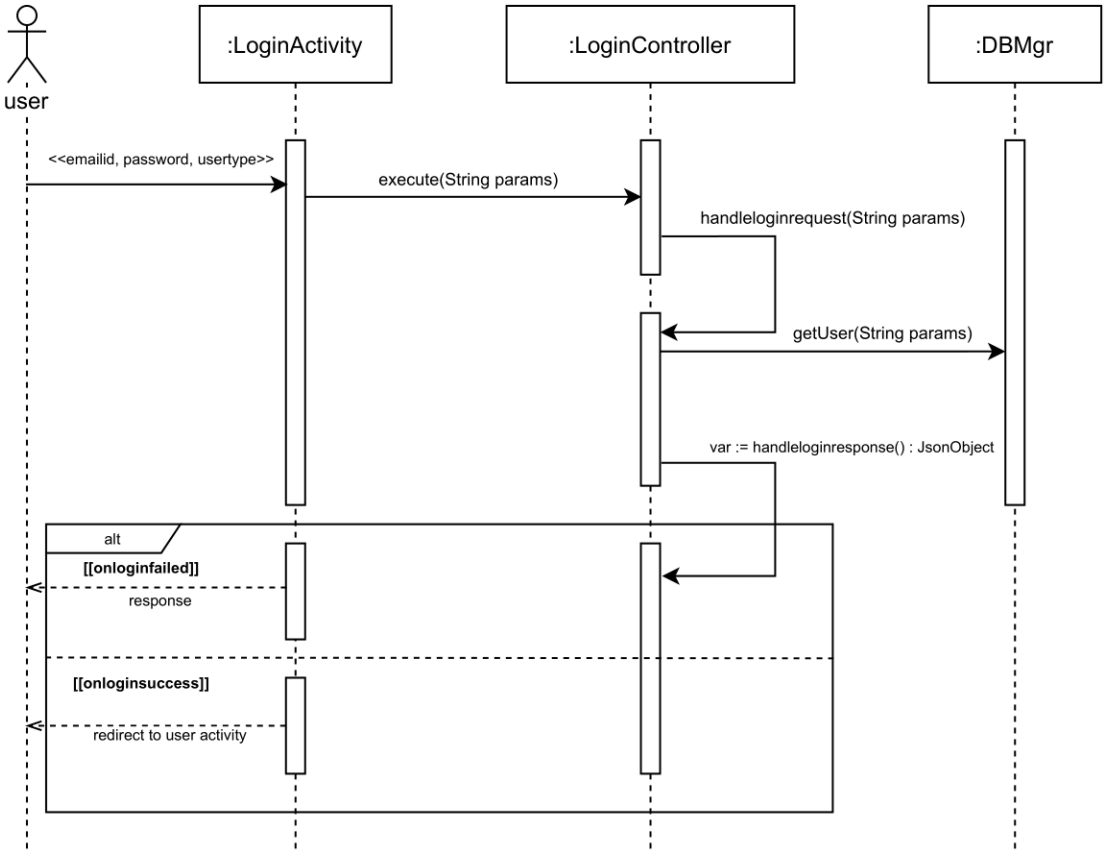
Use Case	Priority	Efforts (Person Week)	Depends On	Assigned To	Iteration 1 (03/10/2023)	Iteration 2 (03/29/2023)	Iteration 3 (05/01/2023)
UC1	9	1	UC2, UC17	All		1	
UC2	3	2	None	SR		2	
UC3	3	1	UC2, UC17	All			1
UC4	10	1	UC1	All		1	
UC5	10	2	UC4	All		2	
UC6	1	3	UC1	SR		3	
UC6.1	1	2	UC1			2	
UC6.2	1	1	UC1			1	
UC6.3	1	1	UC1			1	
UC7	4	1		SR		1	
UC7.1	4	1	UC1			1	
UC7.2	4	1	UC1			1	
UC7.3	4	1	UC1			1	
UC8	1	3	UC1, UC6	SB		3	
UC8.1	1	1	UC1, UC6			1	
UC8.2	1	1	UC1, UC6			1	
UC8.3	1	1	UC1, UC6			1	
UC8.4	1	1	UC1, UC6			1	
UC8.5	1	1	UC1, UC6			1	
UC9	4	1	UC1	SB			1
UC10	3	3	UC1	SB			3
UC11	1	3	UC1, UC6, UC8	PS			3
UC11.1	1	1	UC1, UC6, UC8				1
UC11.2	1	2	UC1, UC6, UC8				2
UC12	5	1	UC1	PS			1
UC13	1	3	UC1, UC6, UC8	YK			3
UC13.1	1	1	UC1, UC6, UC8				1
UC13.2	1	2	UC1, UC6, UC8				2
UC14	5	1	UC1	YK			1
UC15	3	3	UC1, UC6, UC8	KP			3
UC15.1	3	1	UC1, UC6, UC8				1
UC15.2	3	2	UC1, UC6, UC8				2
UC16	5	1	UC1	KP			1
UC17	1	3	None	SB			3
UC17.1	1	1	None				1
UC17.2	1	2	None				2
UC18	5	1	UC1	All		1	
Total Efforts		58				26	32

Note: 1 Person-Week = 5 hours

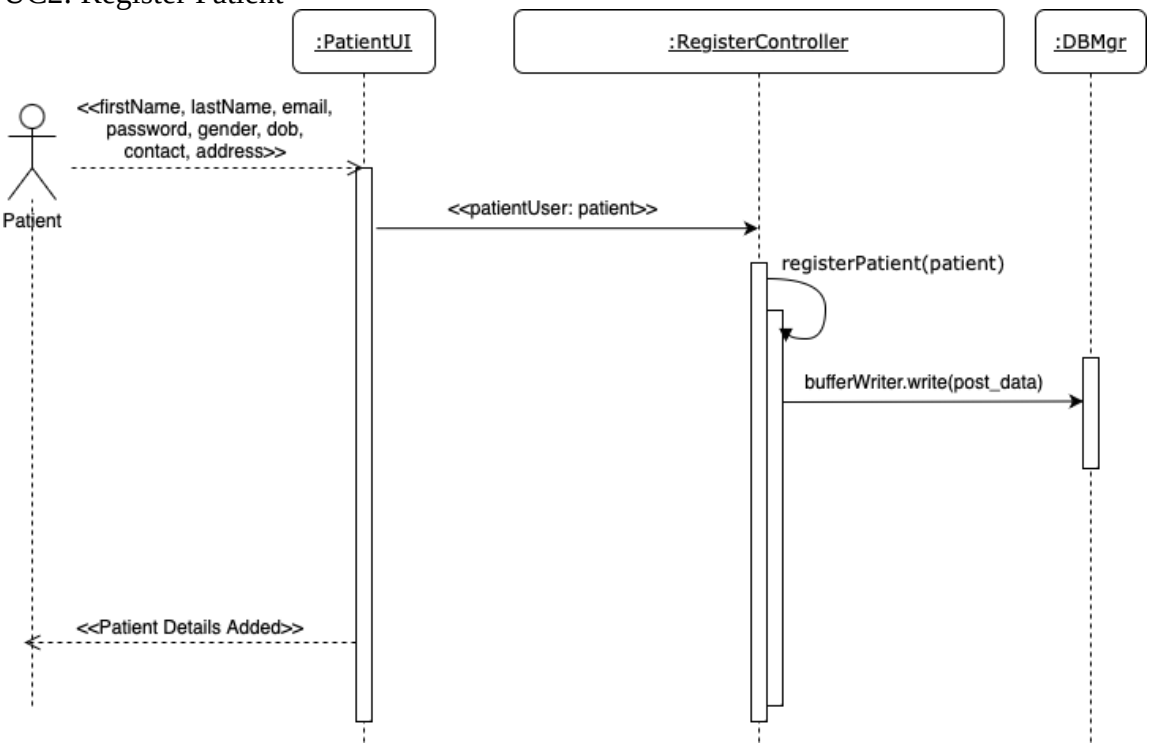
Team Members	
SB	Shubham Borhade
YK	Yash Karanje
KP	Keerthi Patnaik
PS	Preethi Subramanian
SR	Shalaka Rane

Sequence Diagram

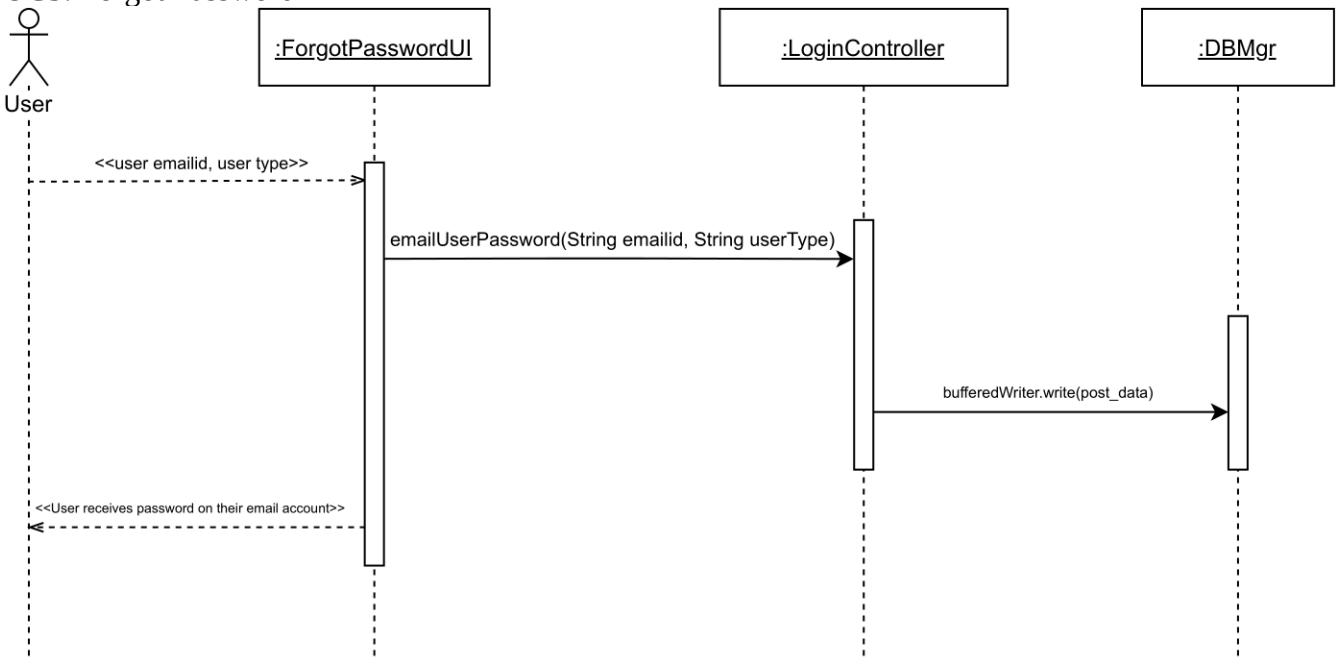
UC 1: Login



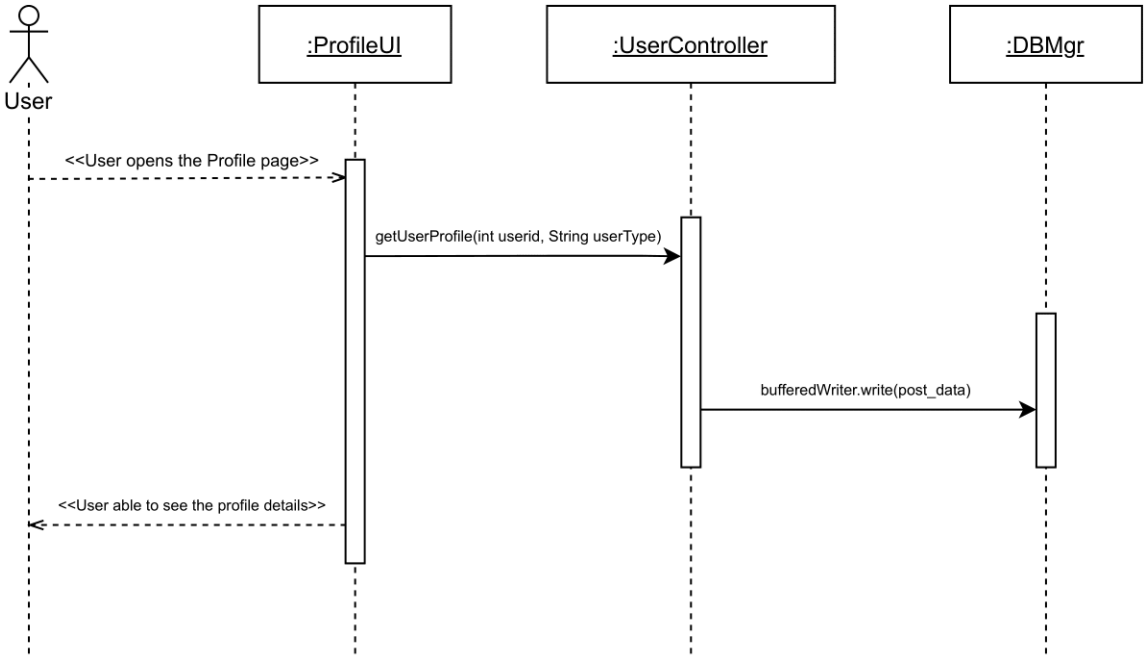
UC2: Register Patient



UC3: Forgot Password

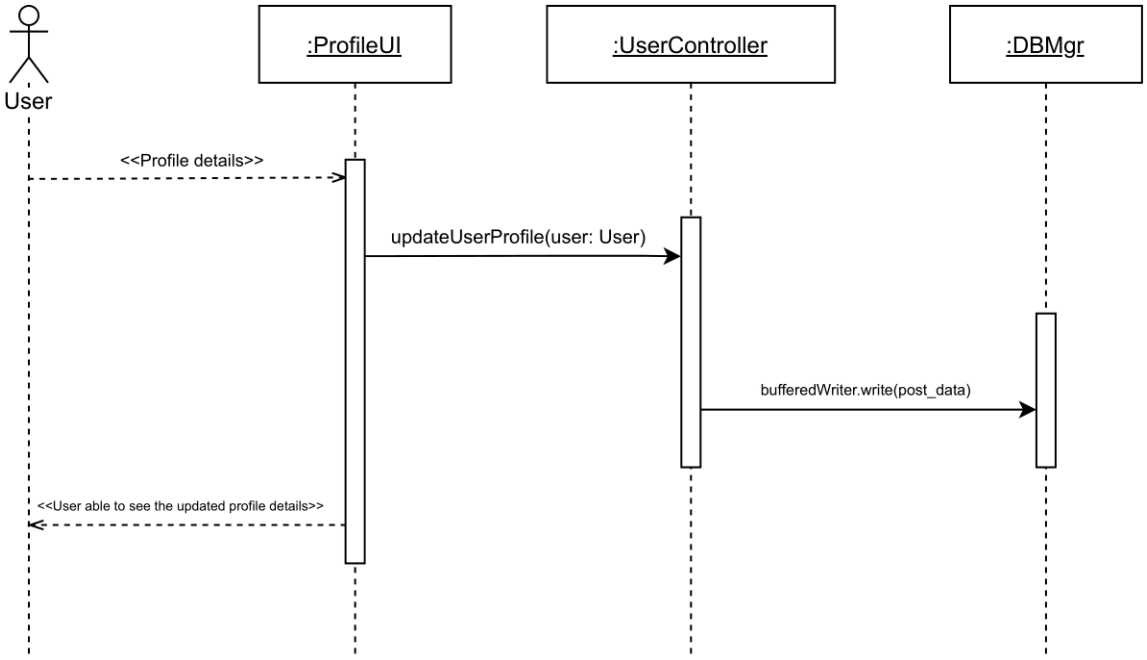


UC4: View Profile:



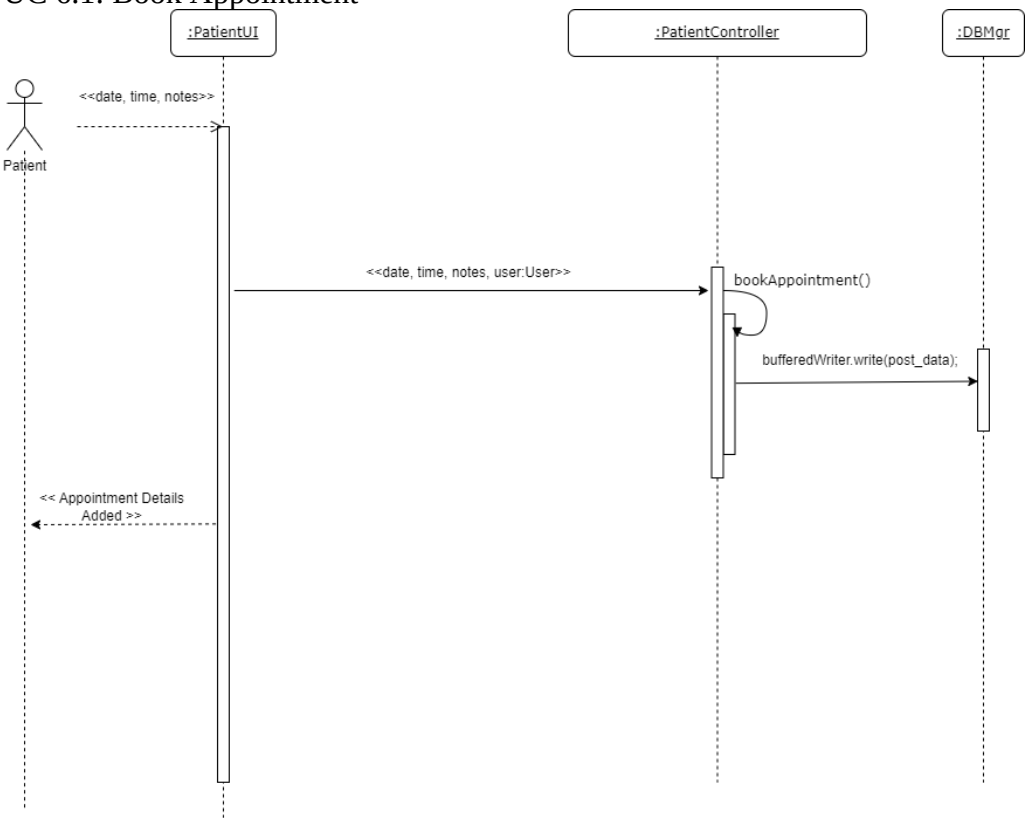
Note: UserController represents (PatientController, ProviderController, InsuranceController, PharmacyController, LaboratoryController, and AdminController)

UC5: Update Profile:

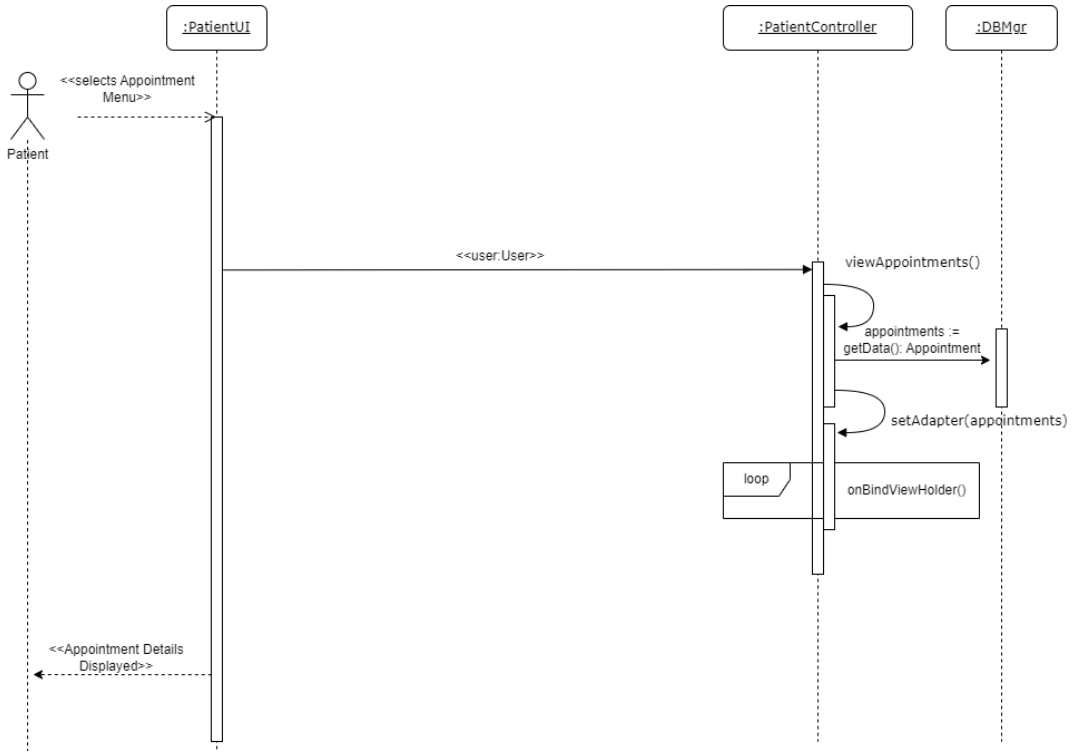


Note: UserController represents (PatientController, ProviderController, InsuranceController, PharmacyController, LaboratoryController, and AdminController)

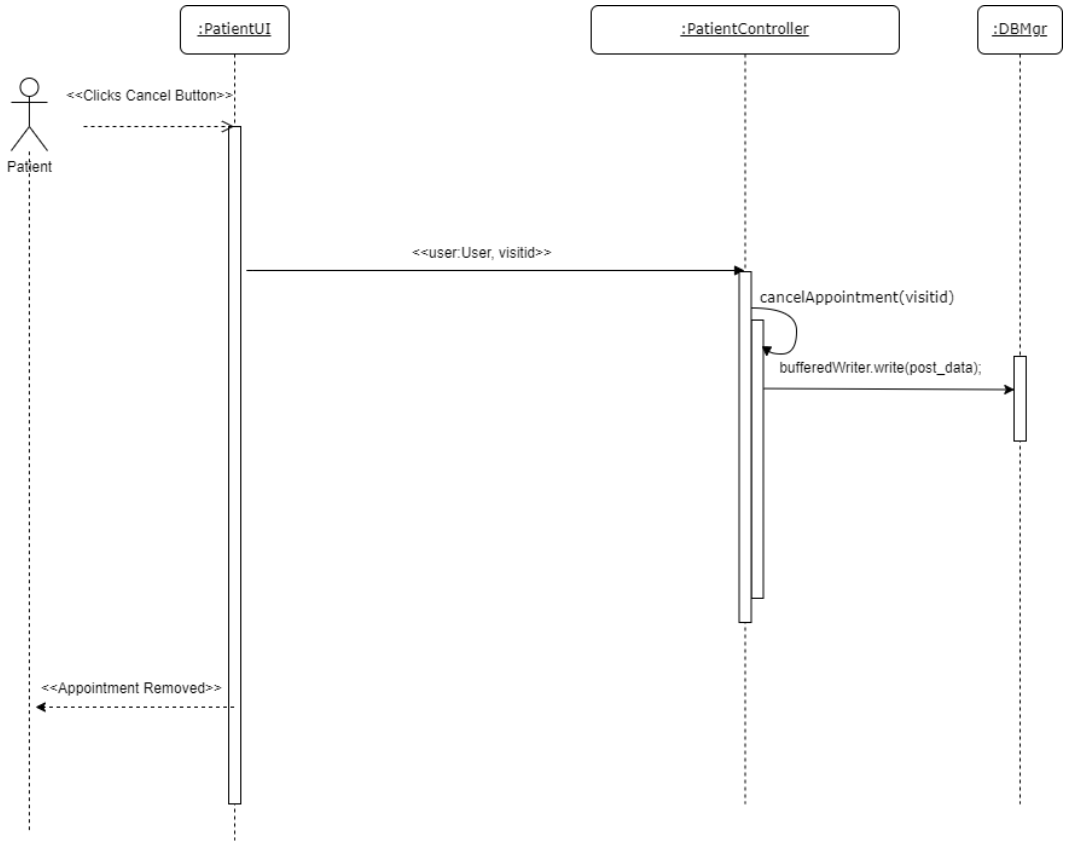
UC 6.1: Book Appointment



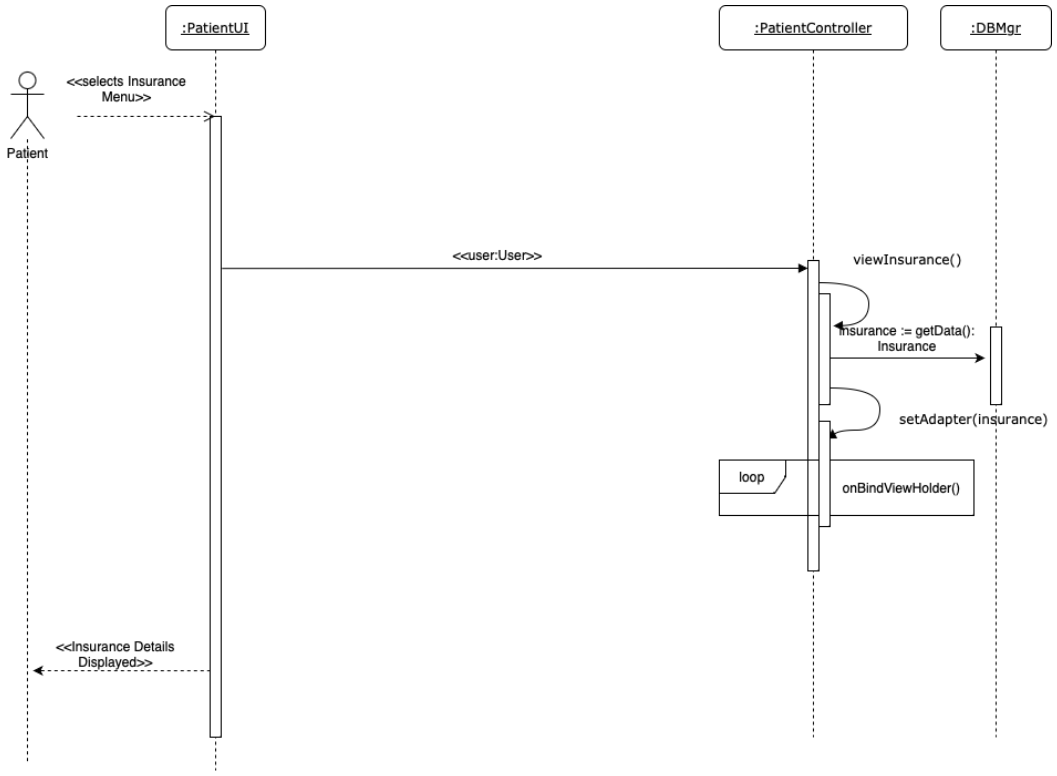
UC 6.2: View Appointment Details



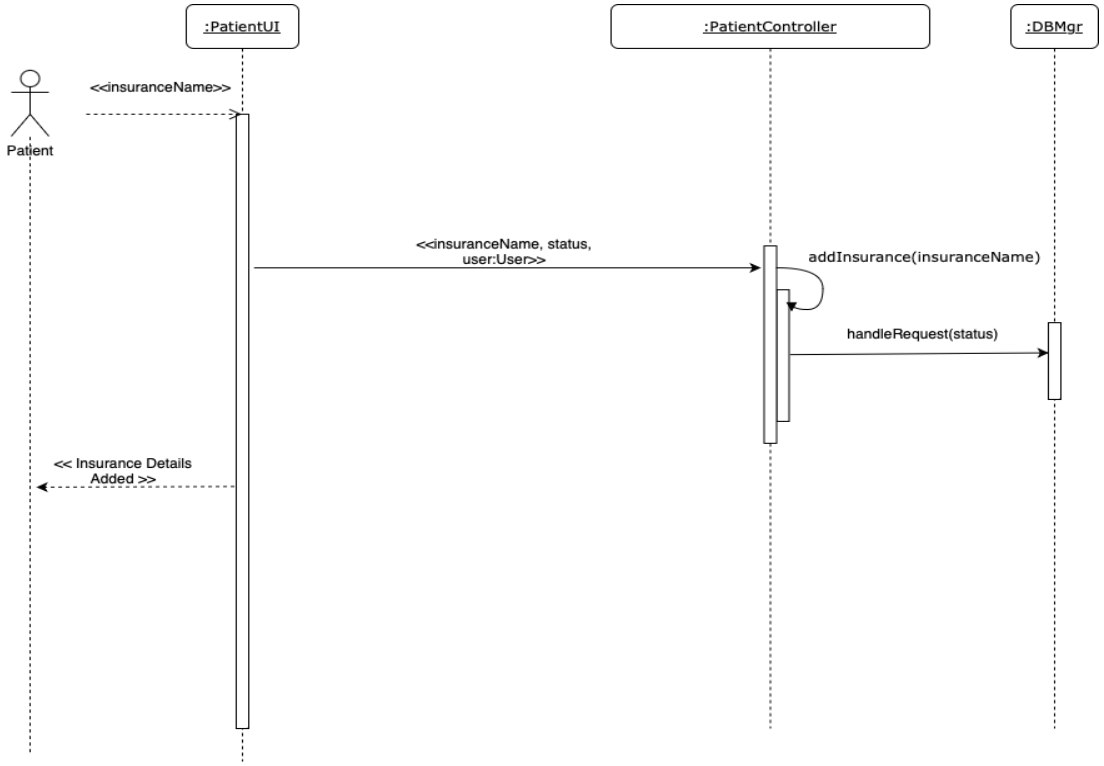
UC 6.3: Cancel Appointment



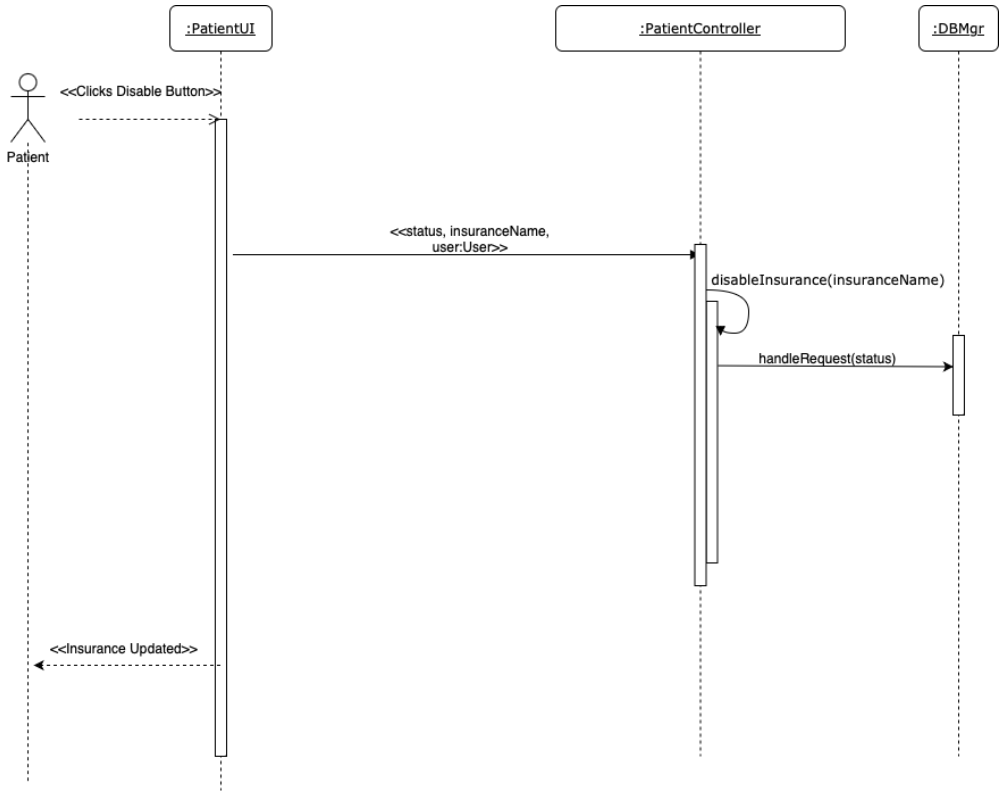
UC 7.1: View Insurance Details



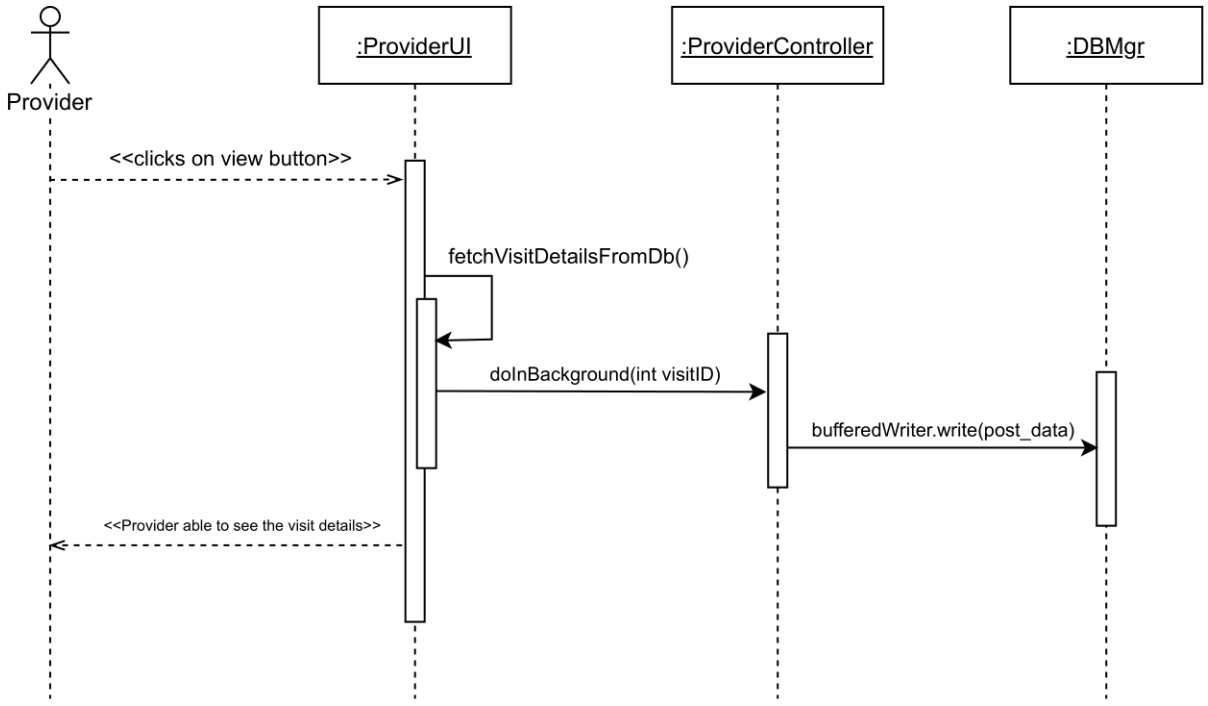
UC 7.2: Add insurance details.



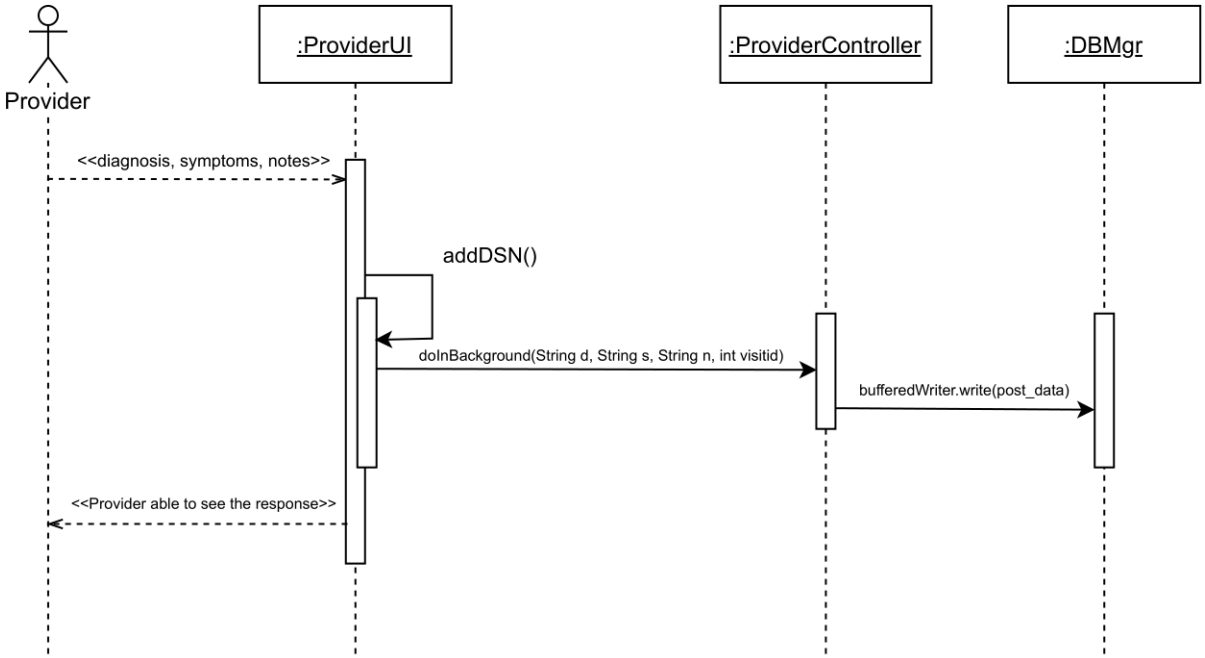
UC 7.3: Disable insurance.



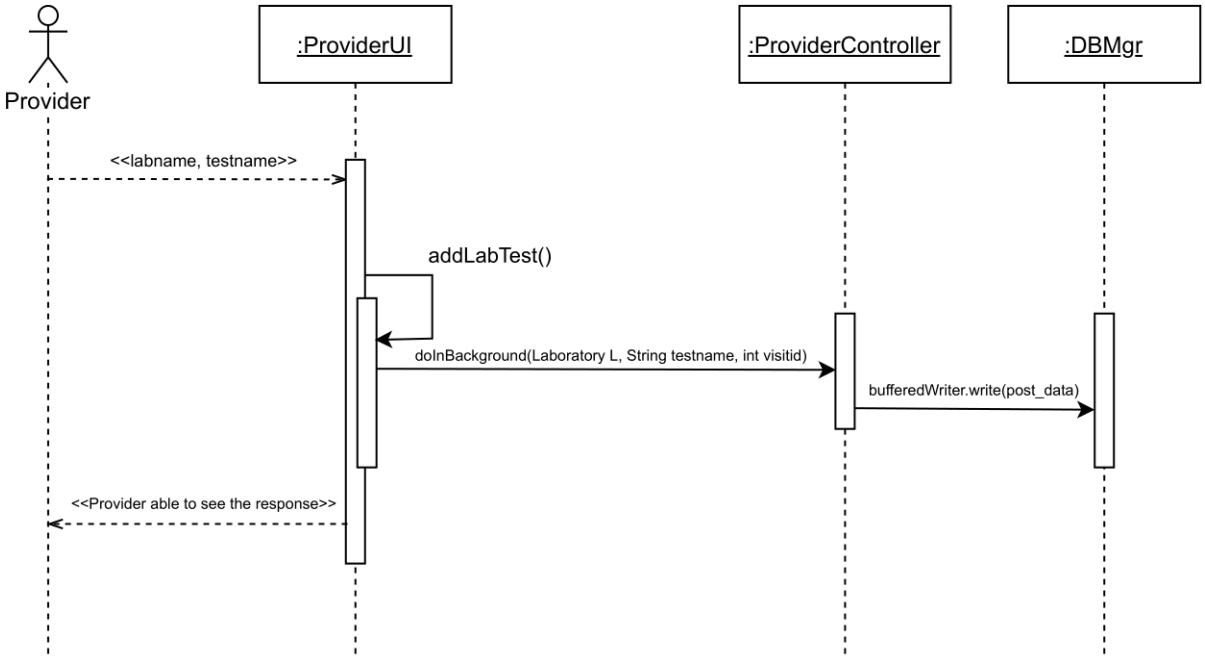
UC 8.1: Provider views appointment details



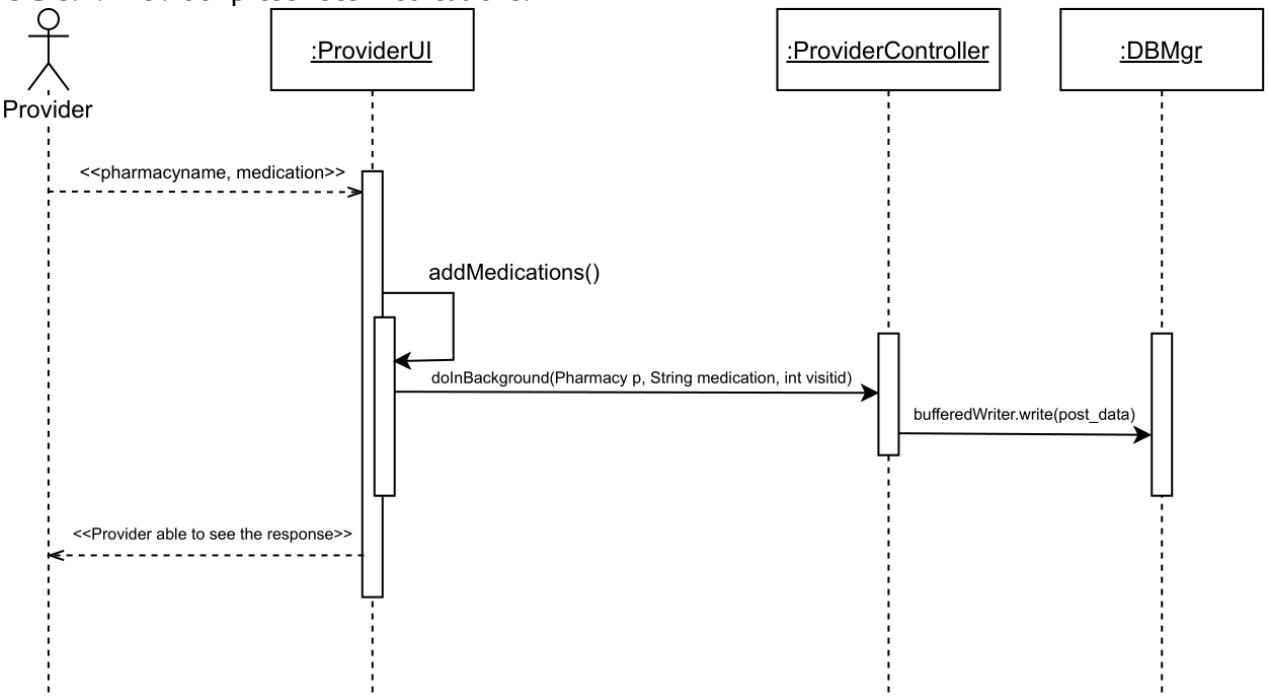
UC 8.2: Provider adds diagnosis, symptoms, and notes.



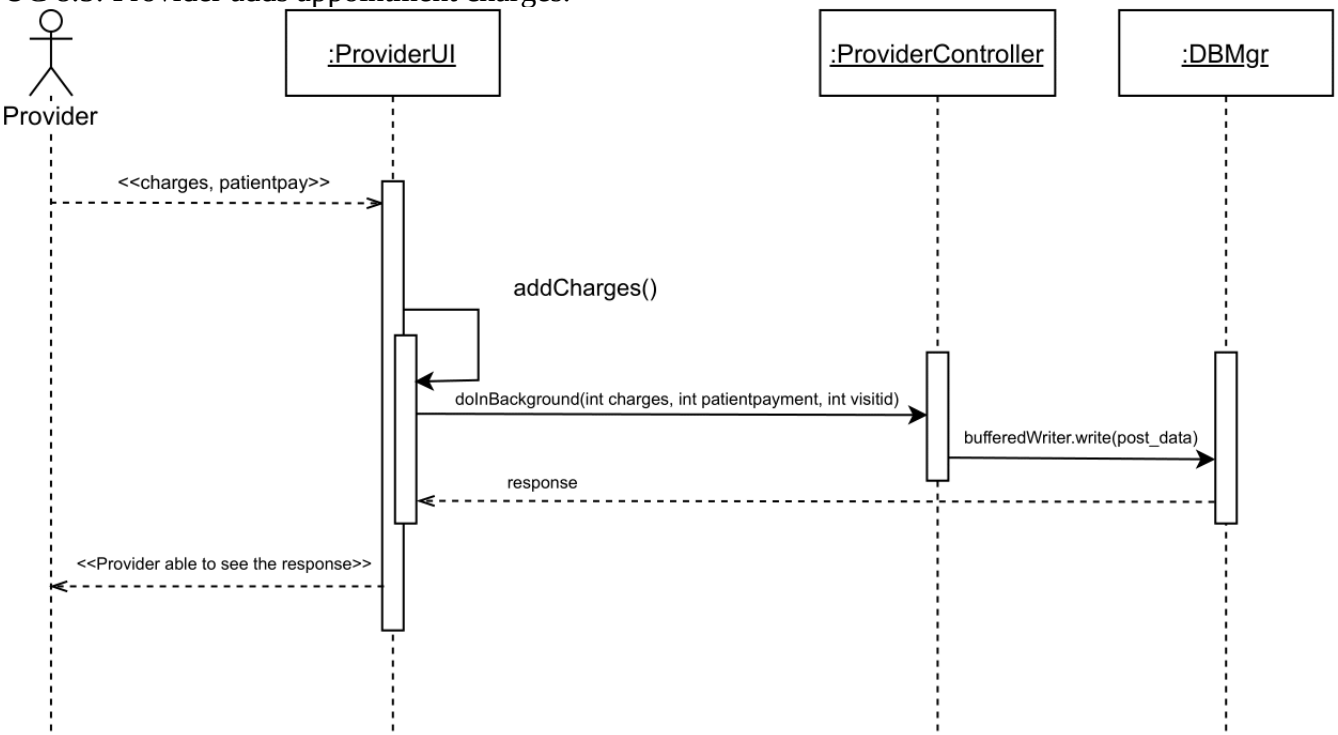
UC 8.3: Provider assign lab test.



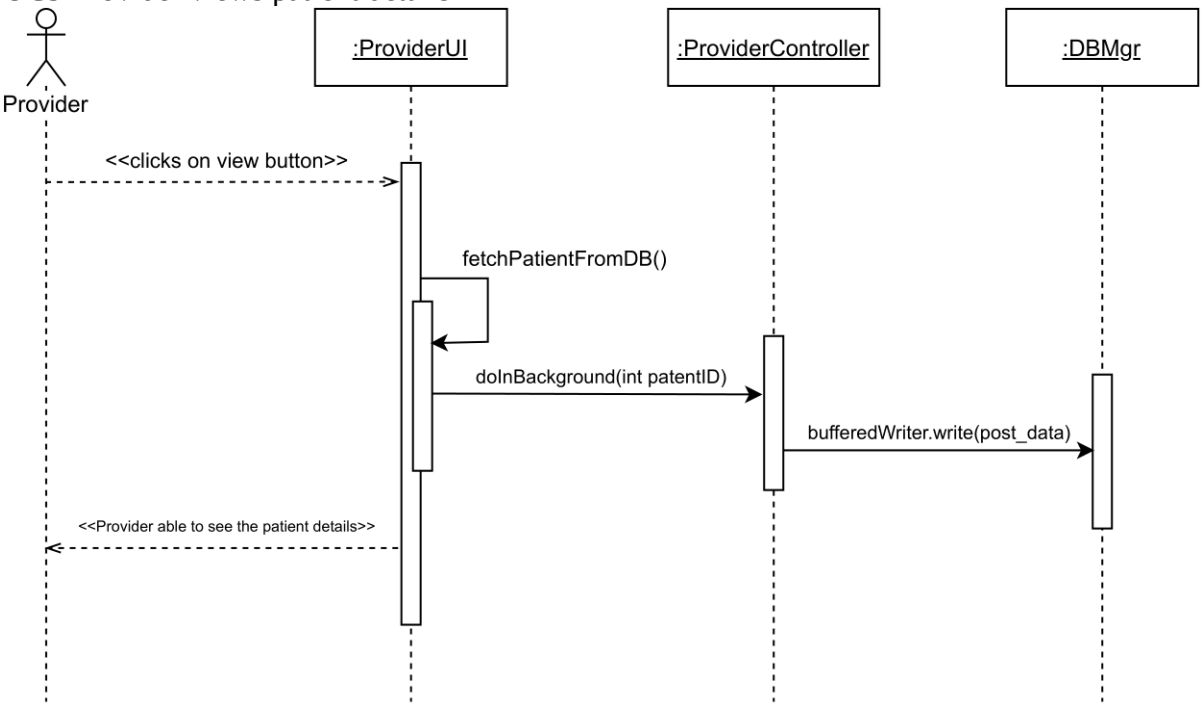
UC 8.4: Provider prescribes medications.



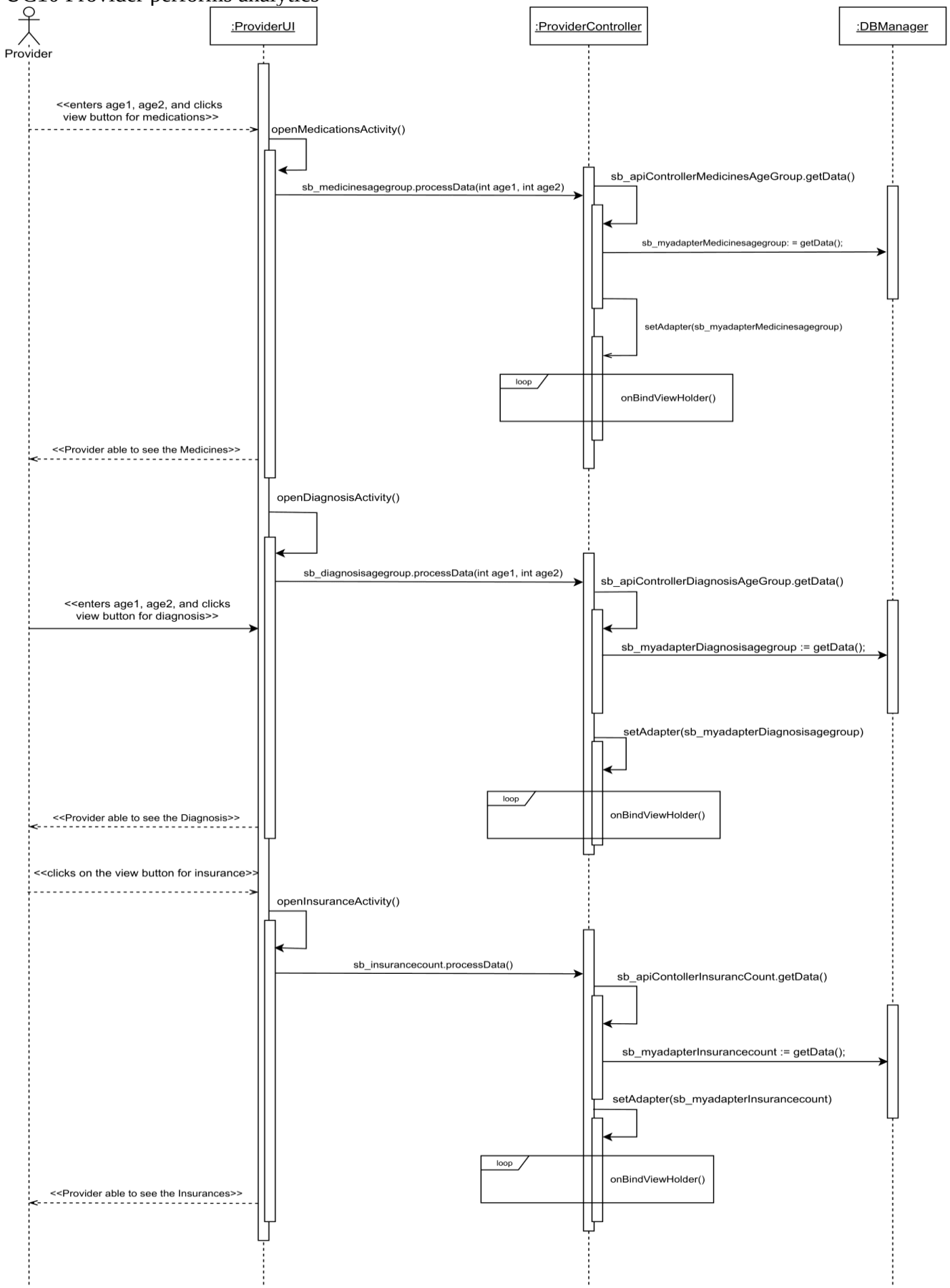
UC 8.5: Provider adds appointment charges.



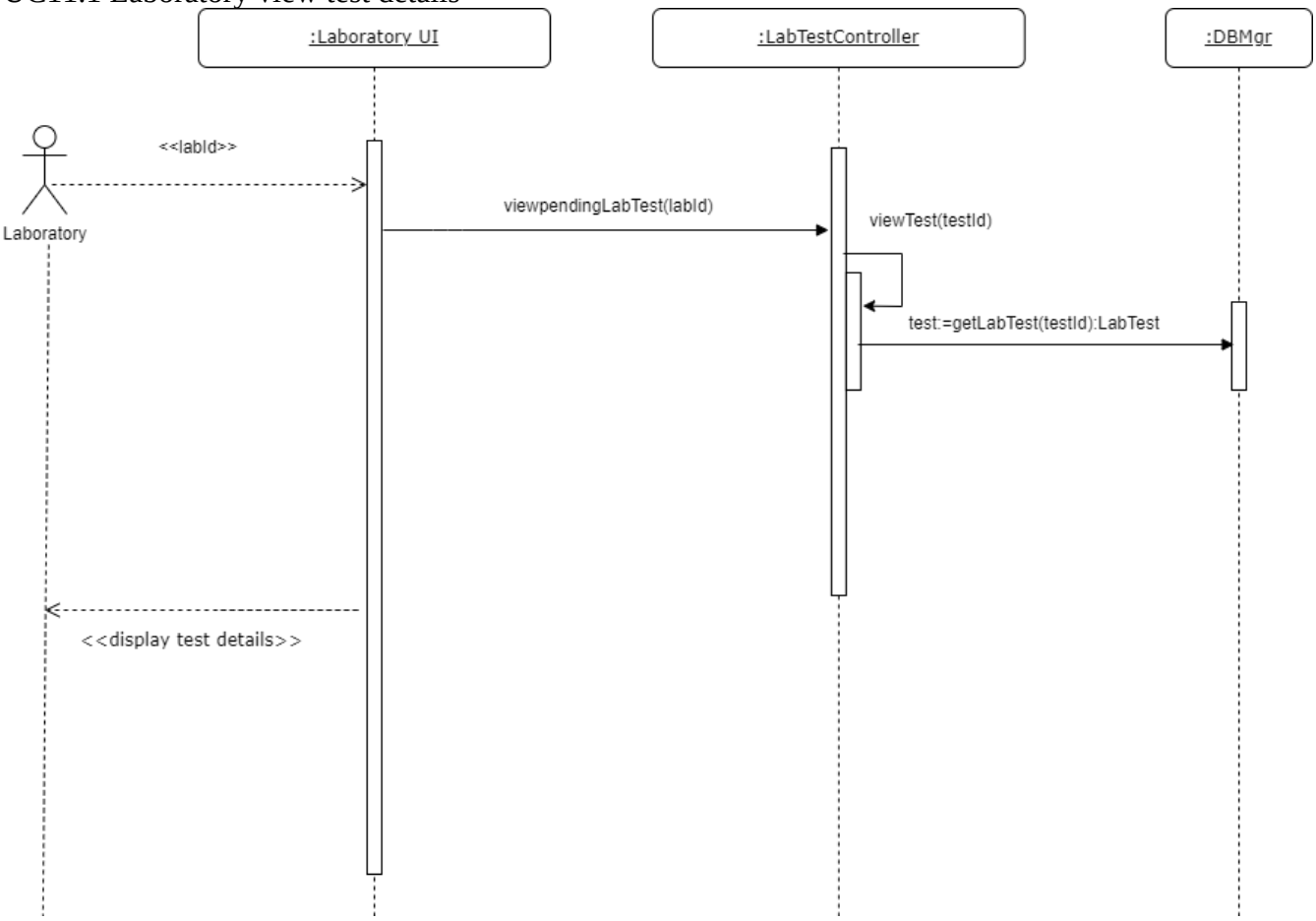
UC9 Provider views patient details



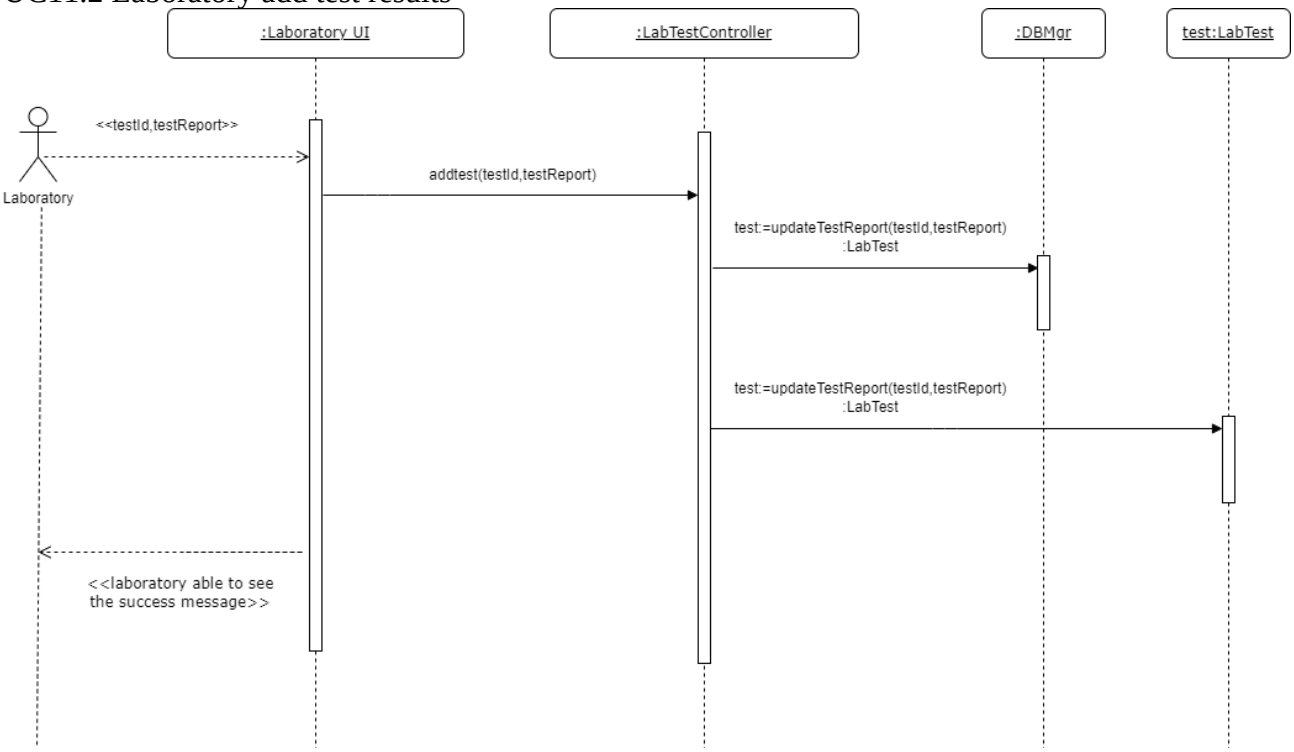
UC10 Provider performs analytics



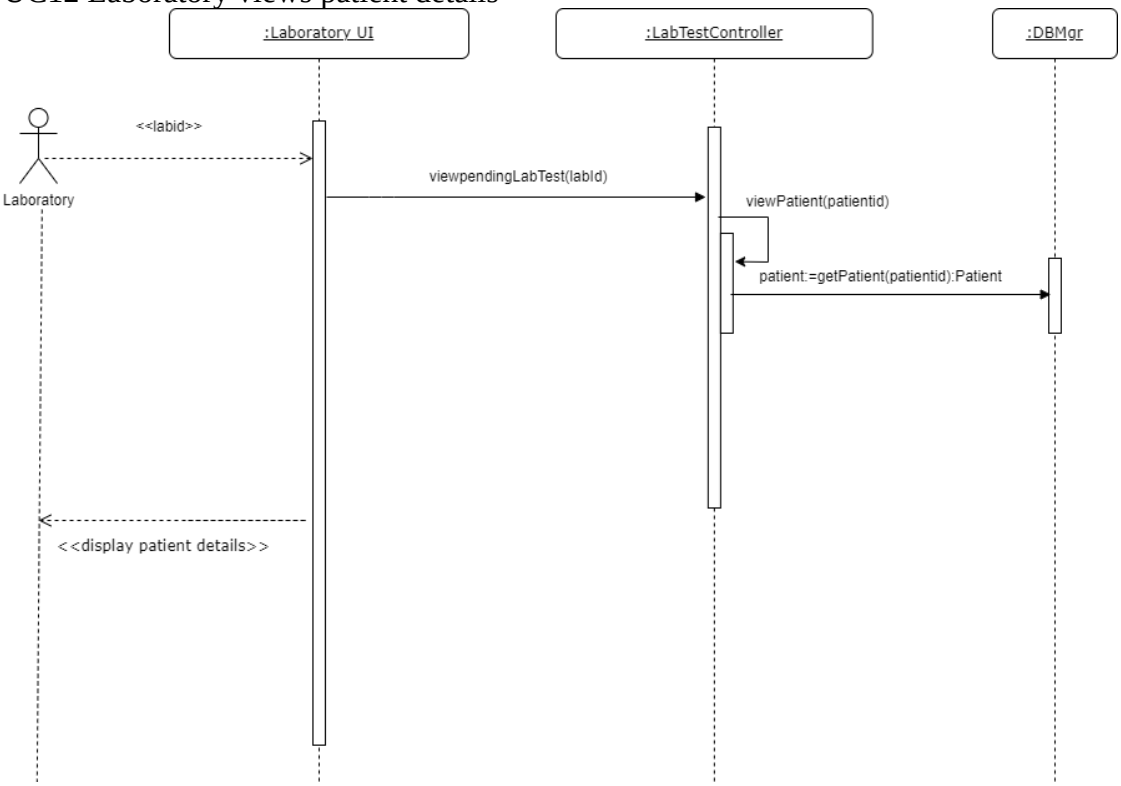
UC11.1 Laboratory view test details



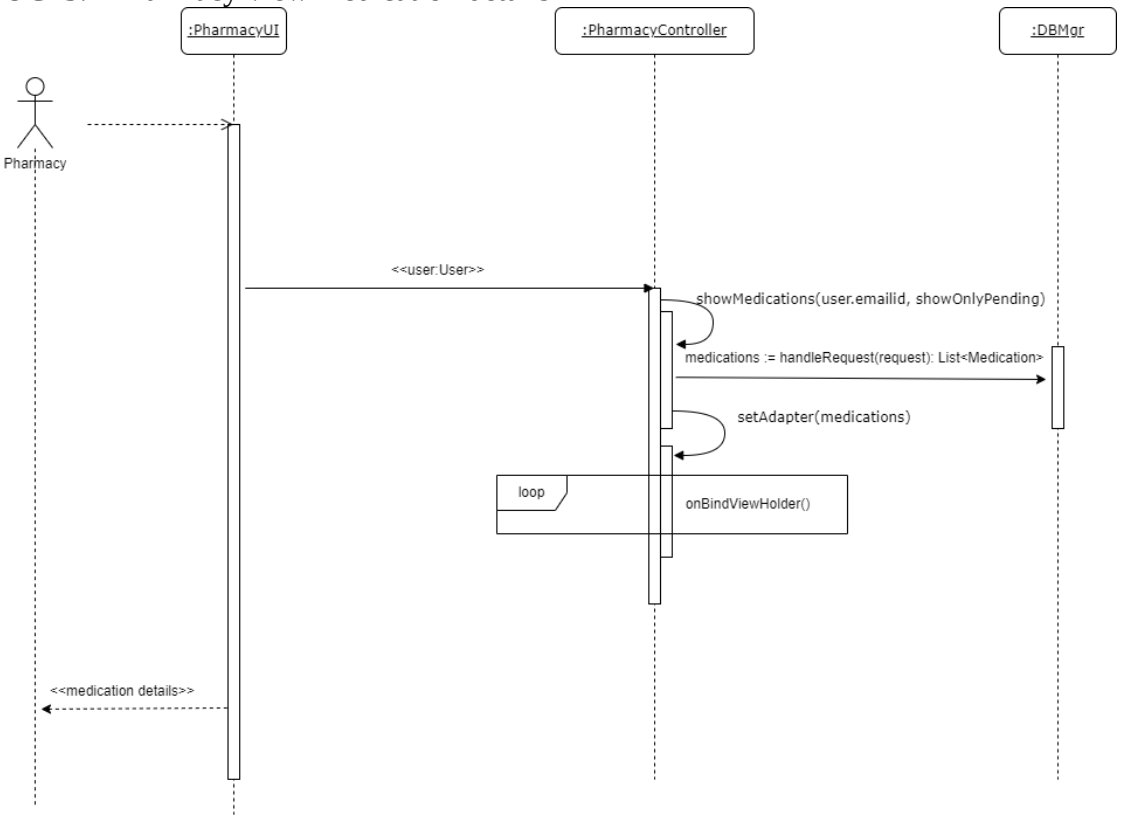
UC11.2 Laboratory add test results



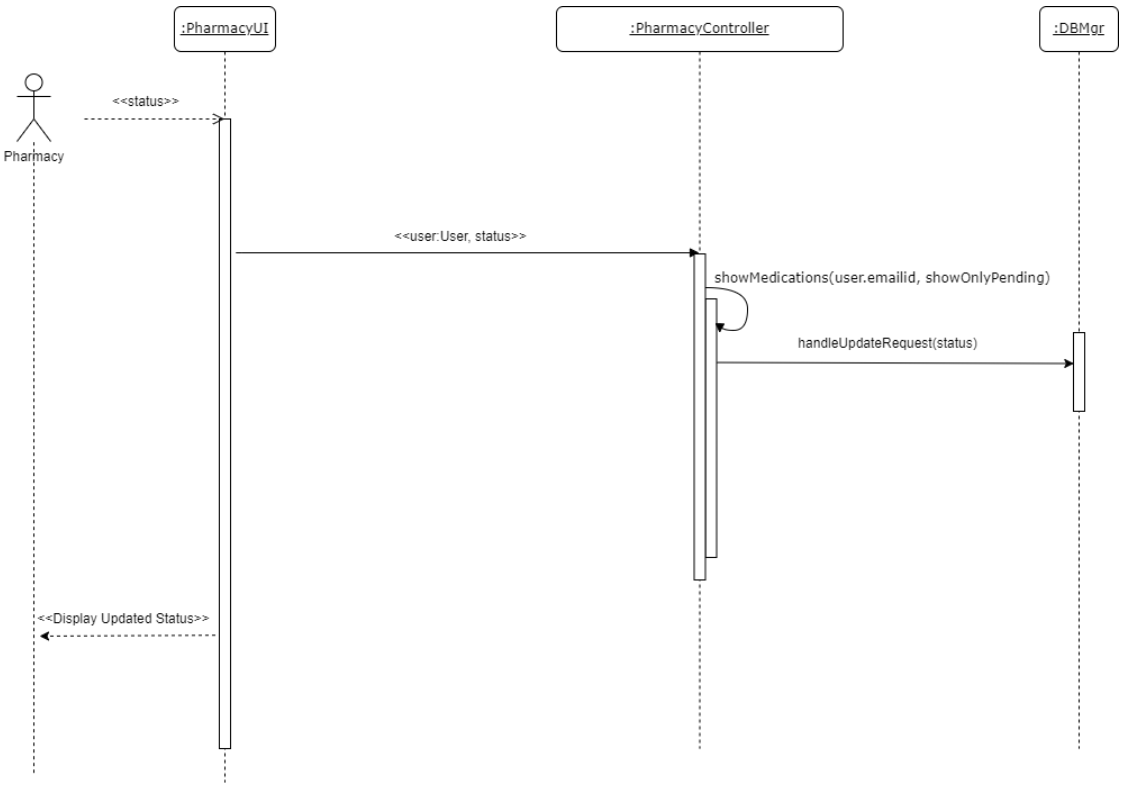
UC12 Laboratory views patient details



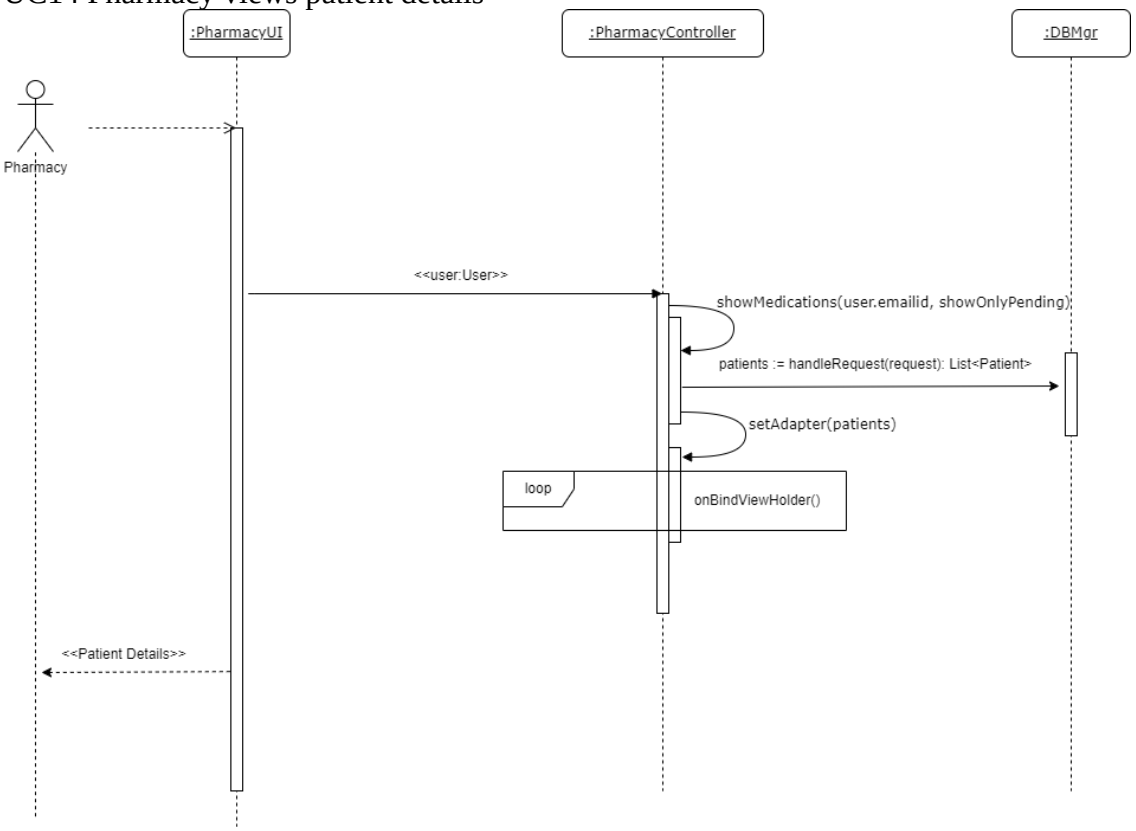
UC13.1 Pharmacy view medication details



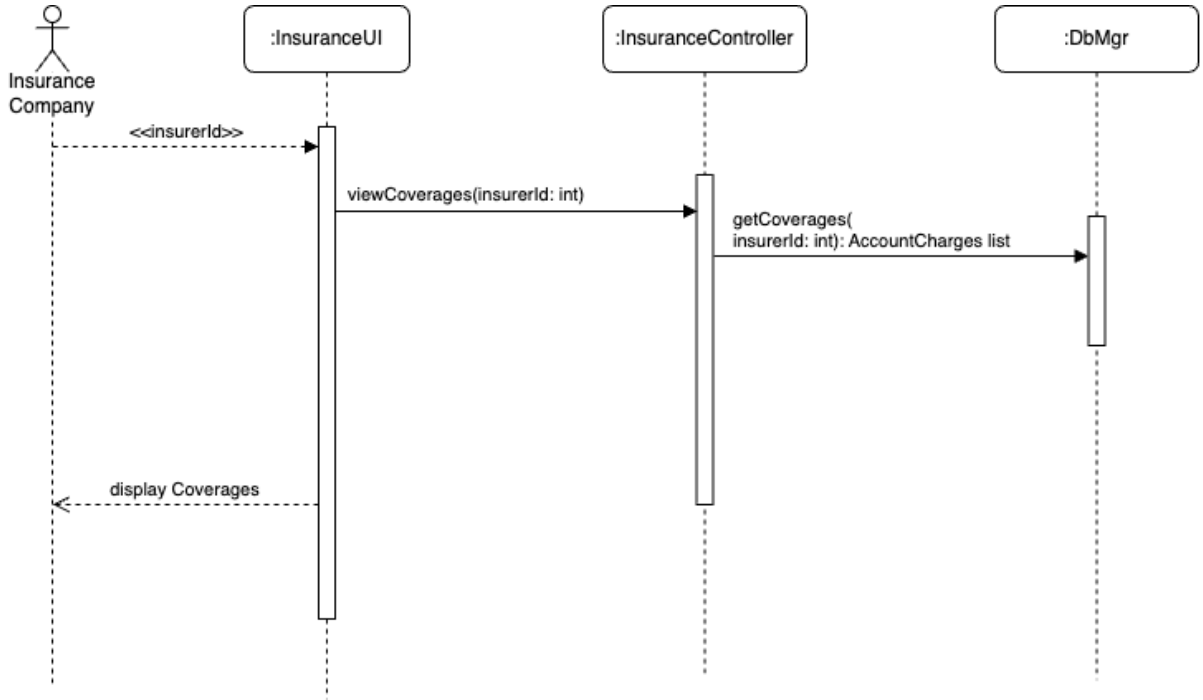
UC13.2 Pharmacy update medication handover status



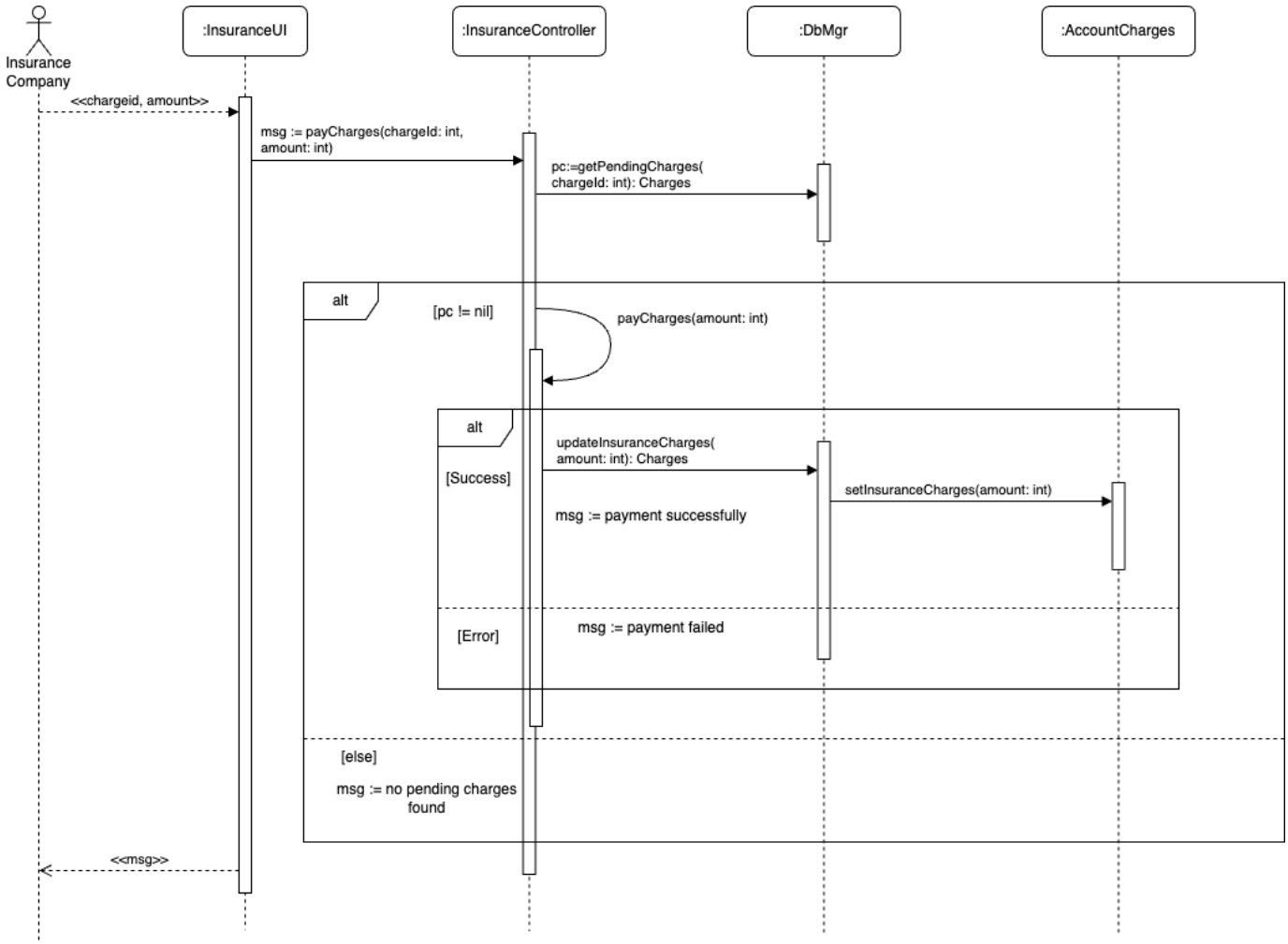
UC14 Pharmacy views patient details



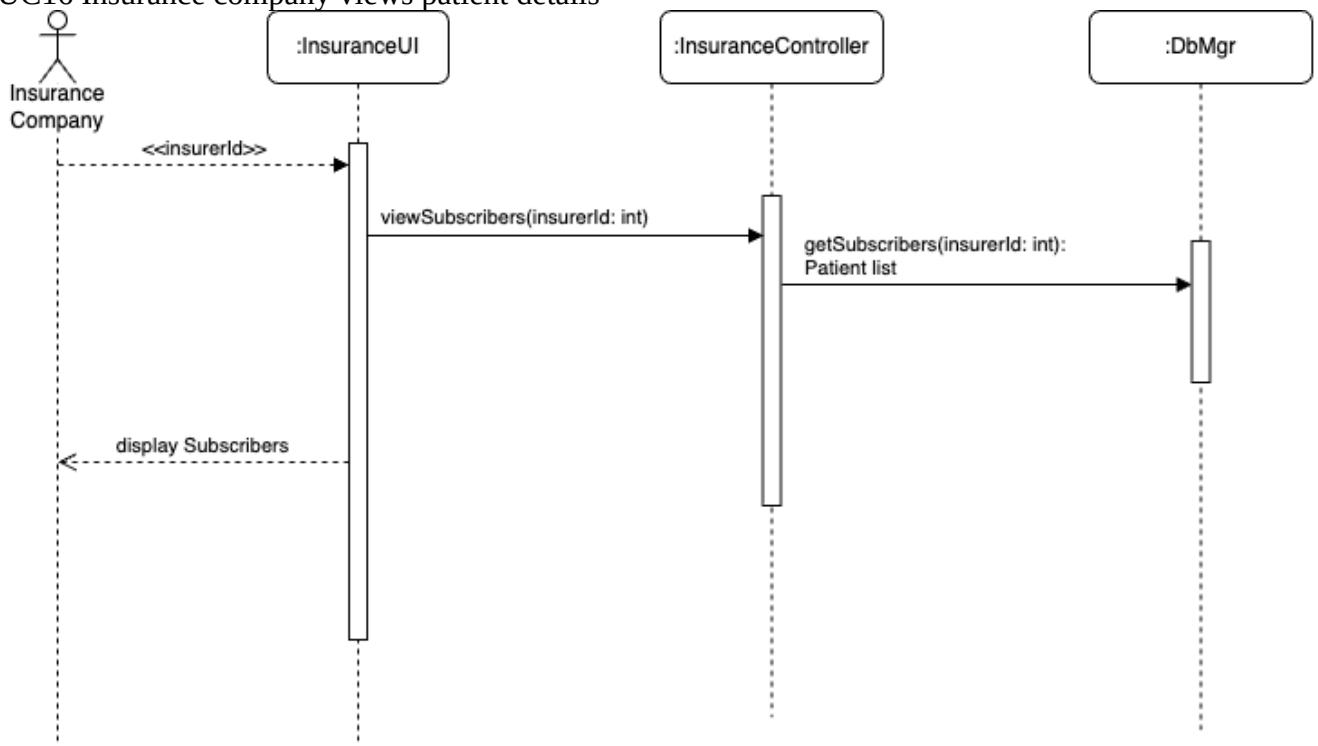
UC15.1 Insurance company views insurance coverage



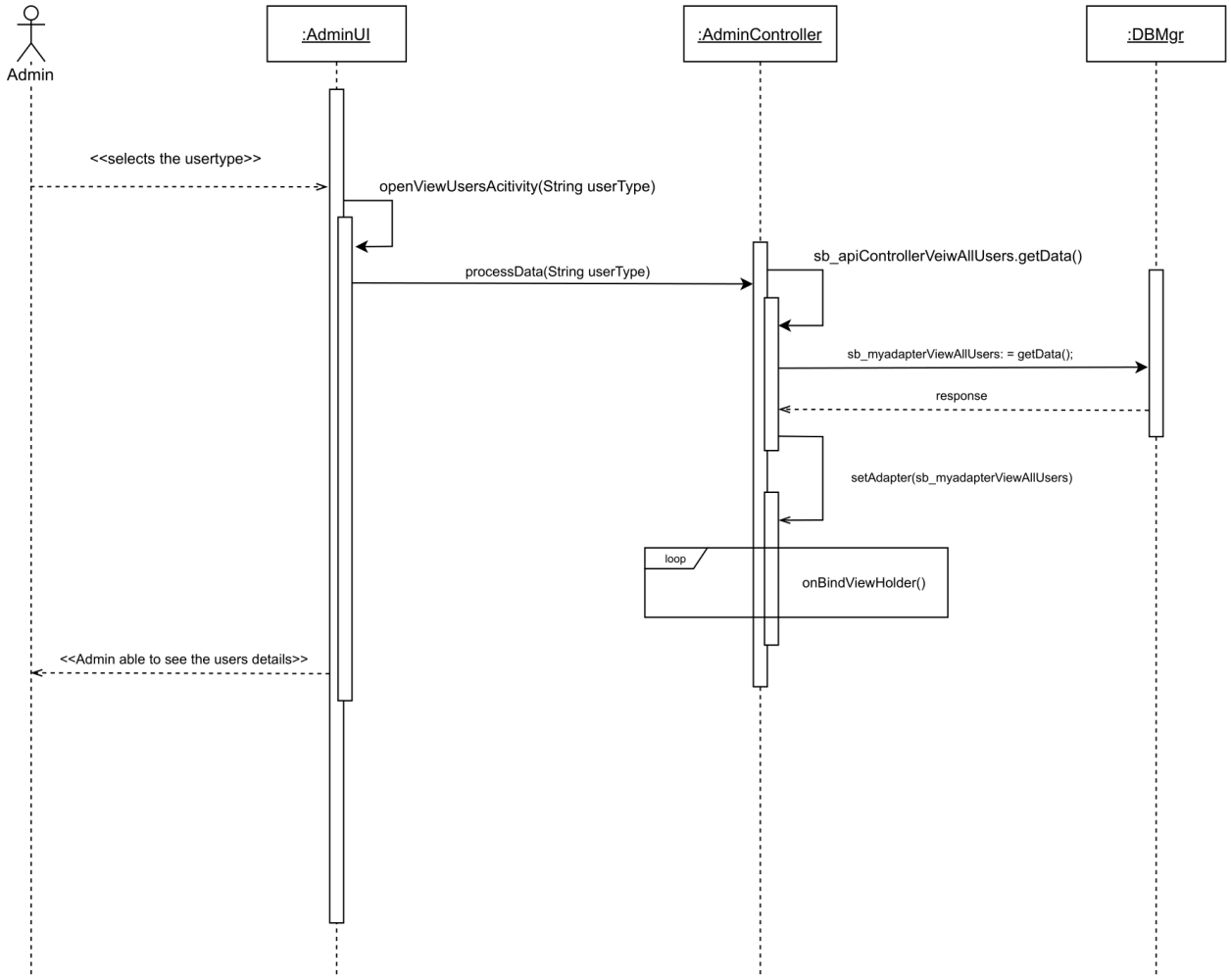
UC15.2 Insurance company updates insurance coverage



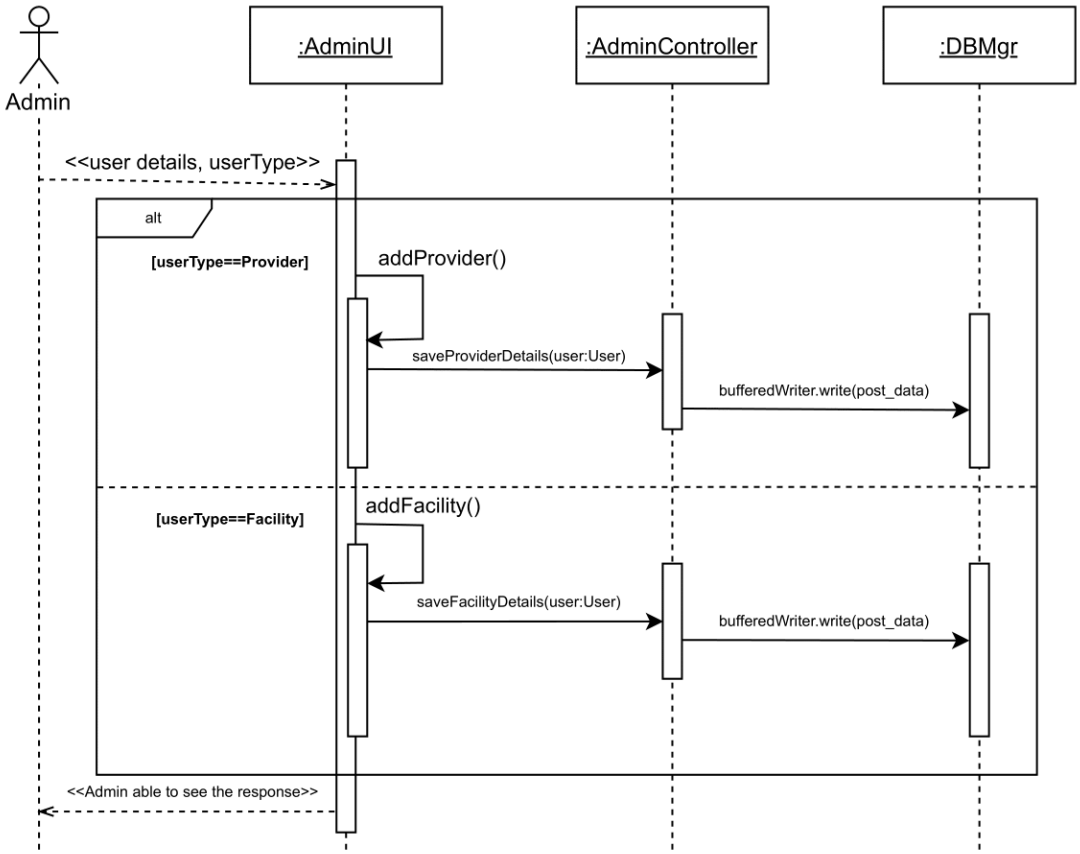
UC16 Insurance company views patient details



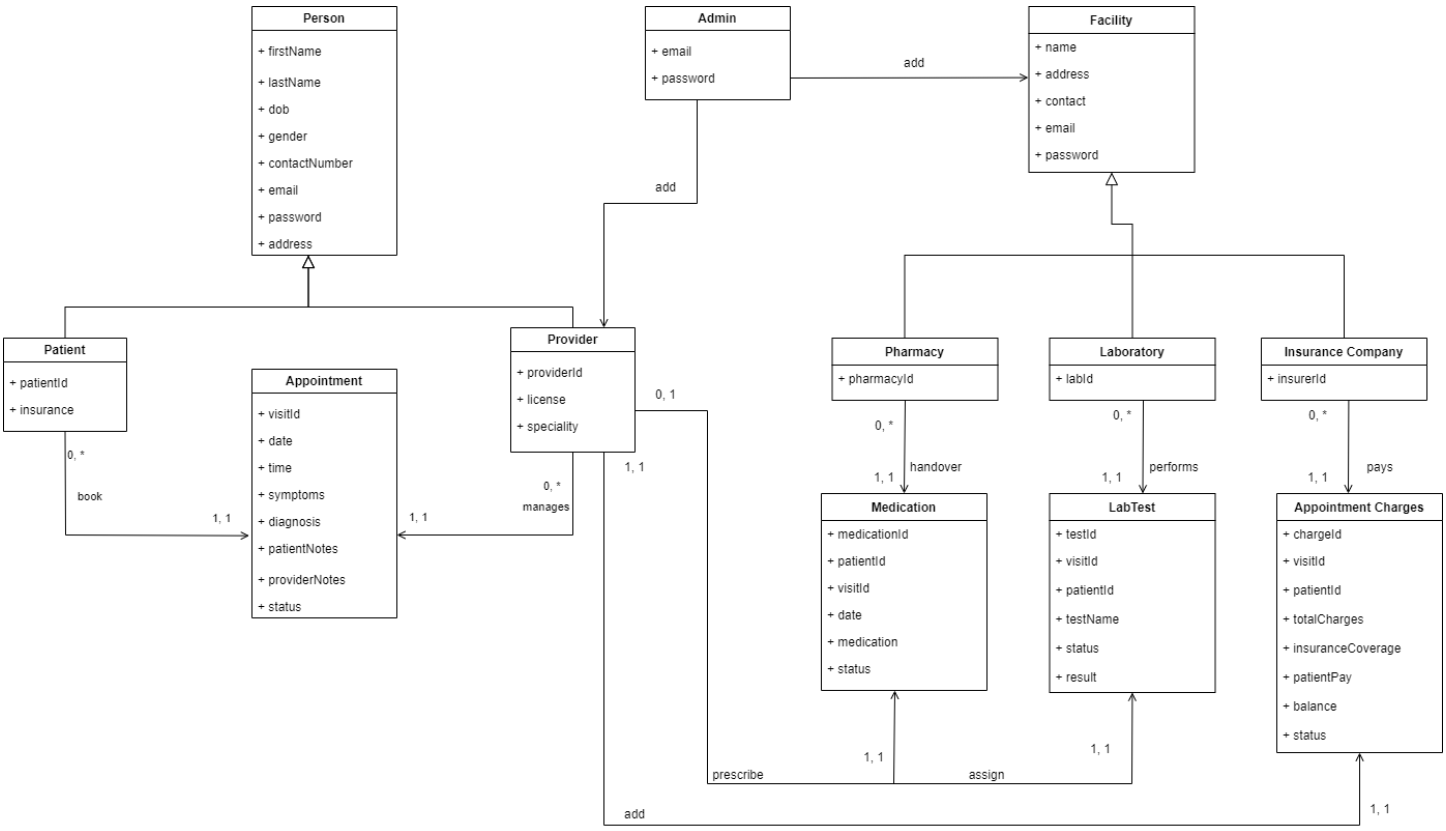
UC17.1 Admin view user details



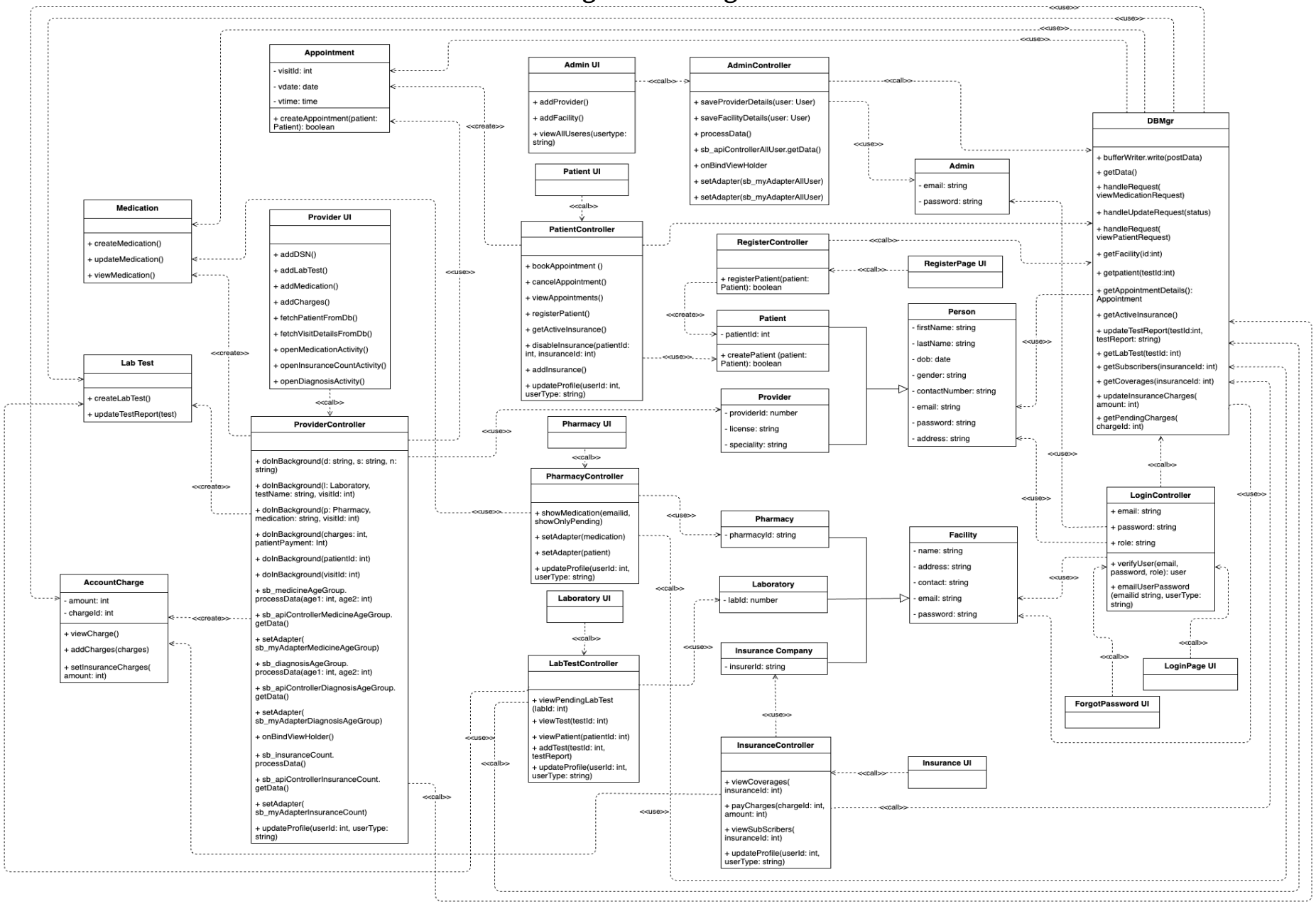
UC17.2 Admin adds user



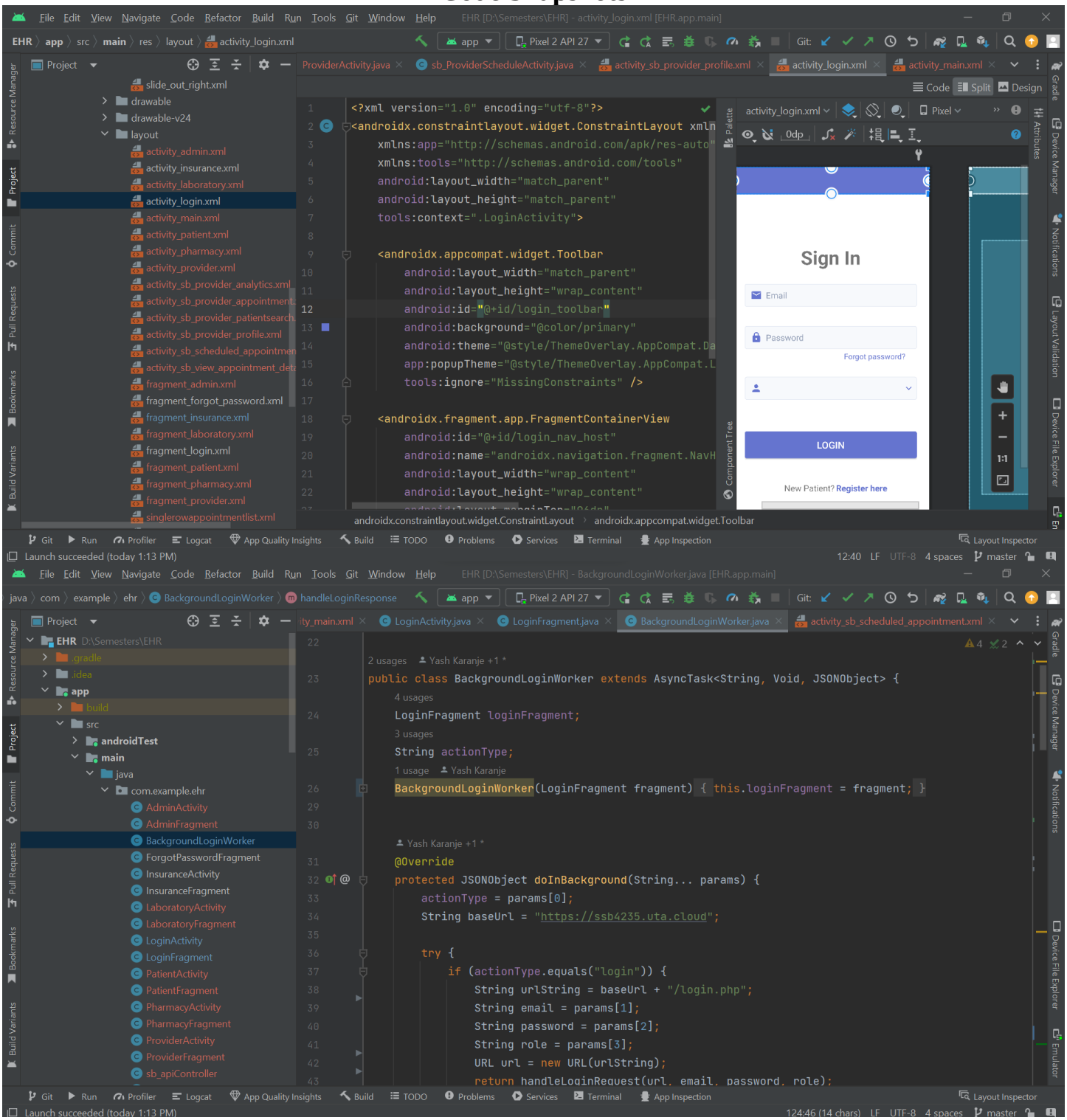
Domain Model

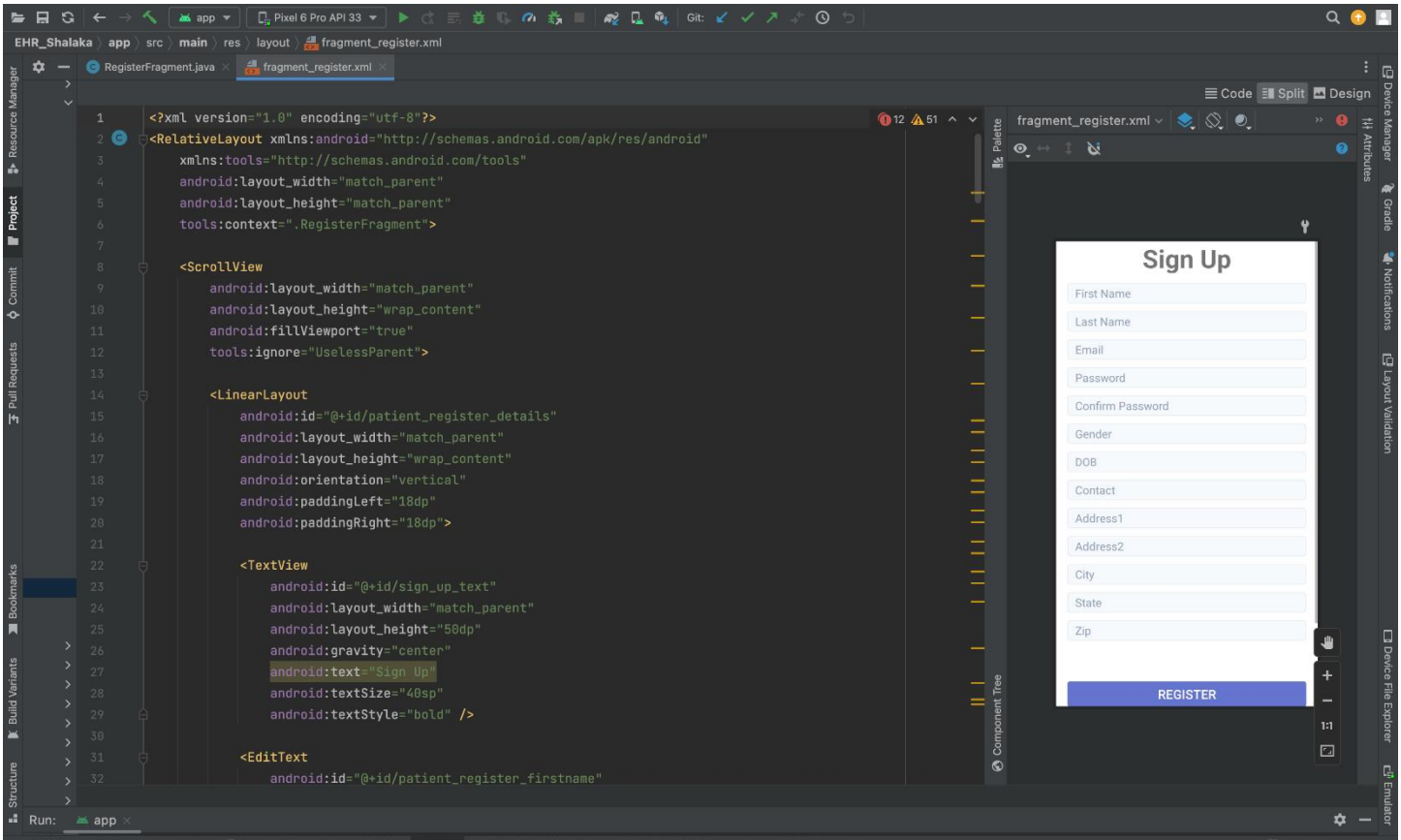


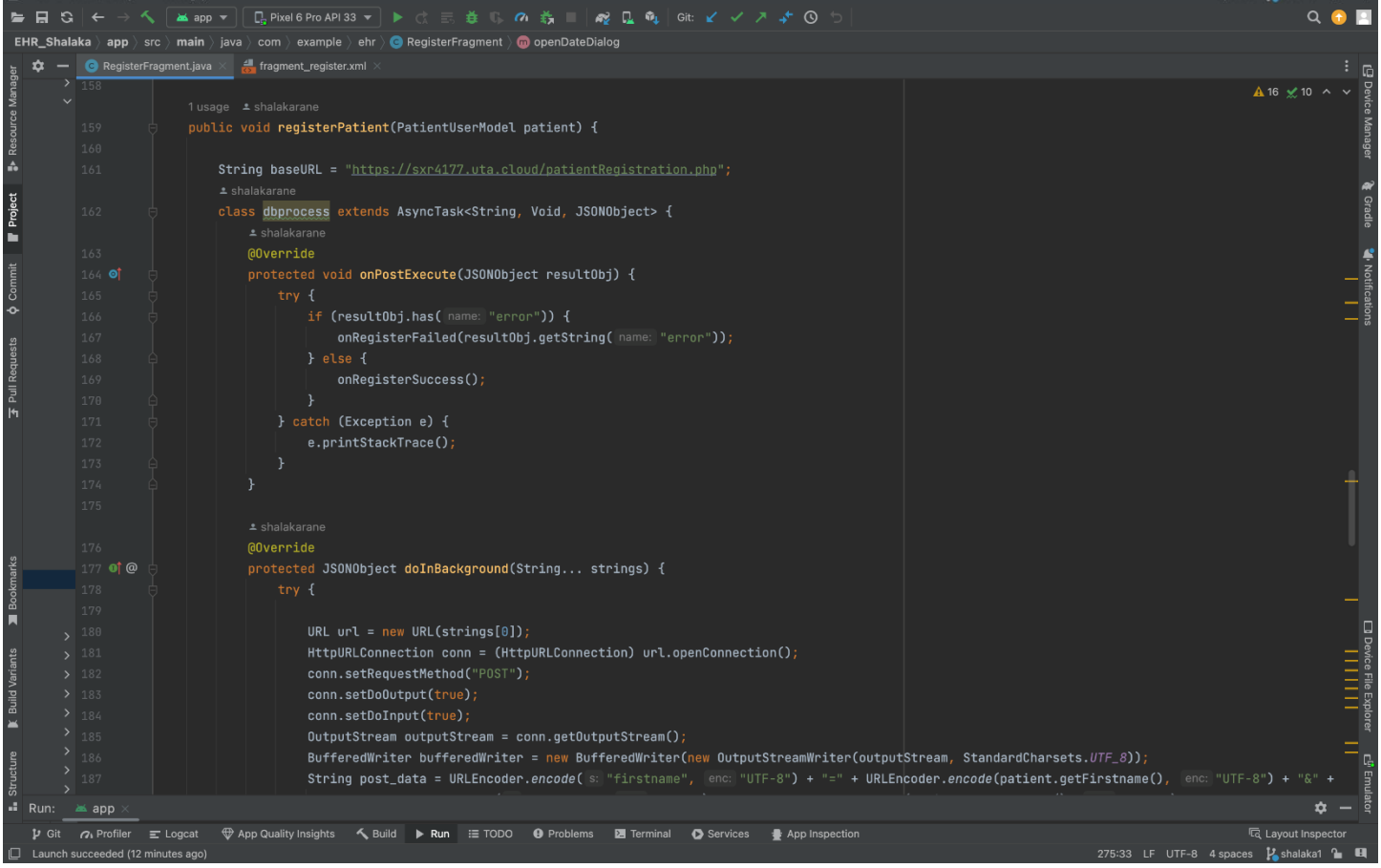
Design Class Diagram

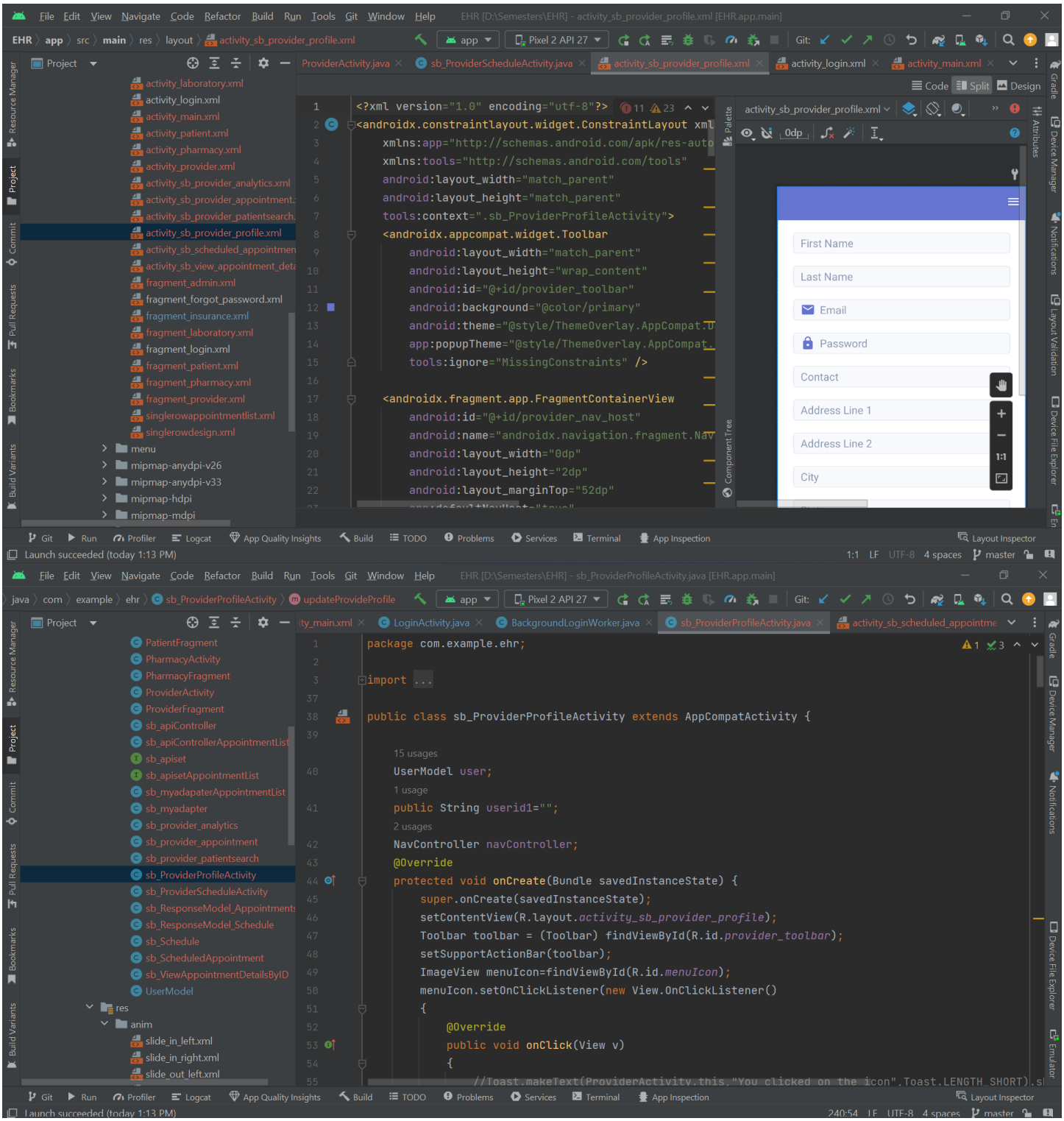


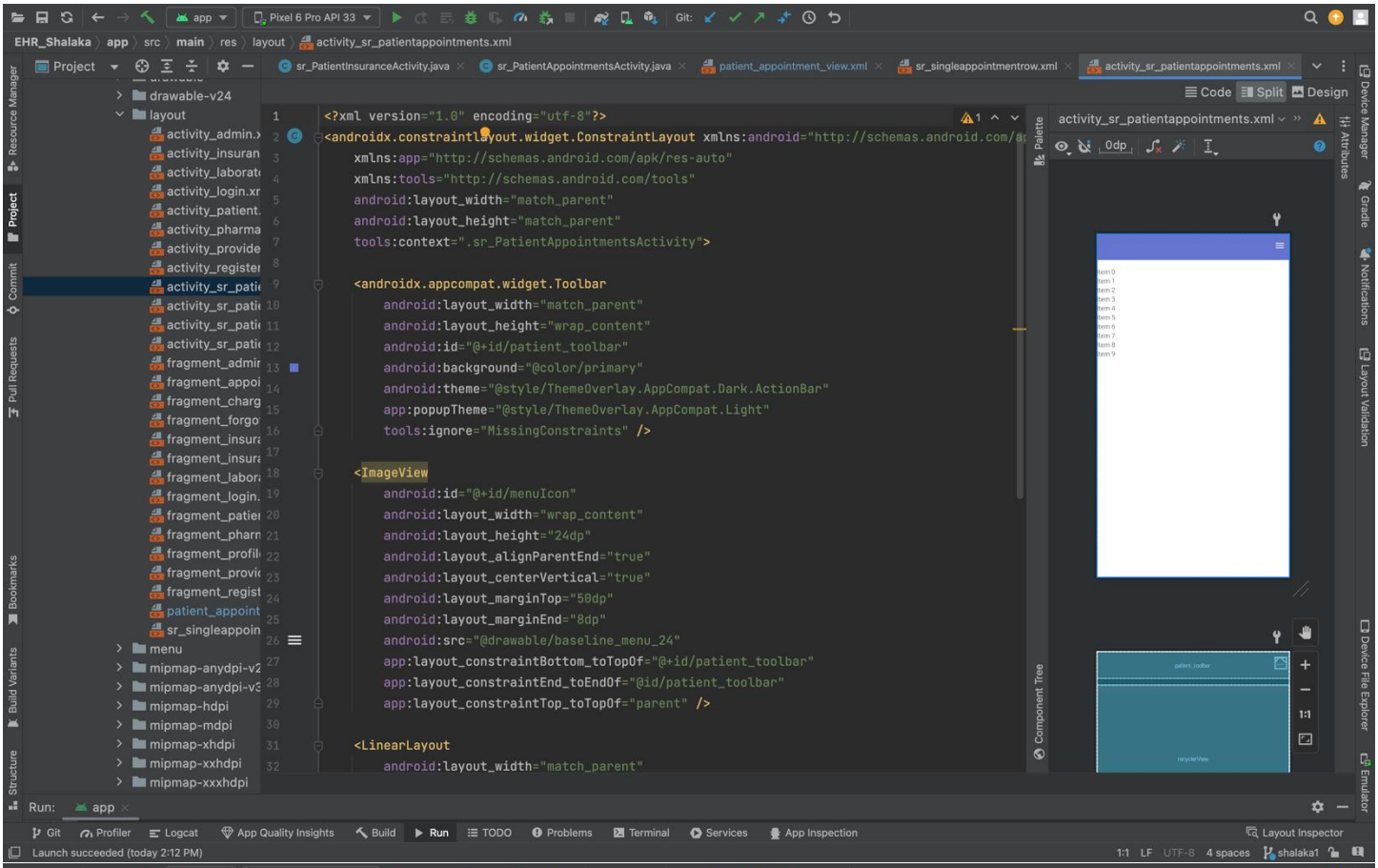
Code Snapshots

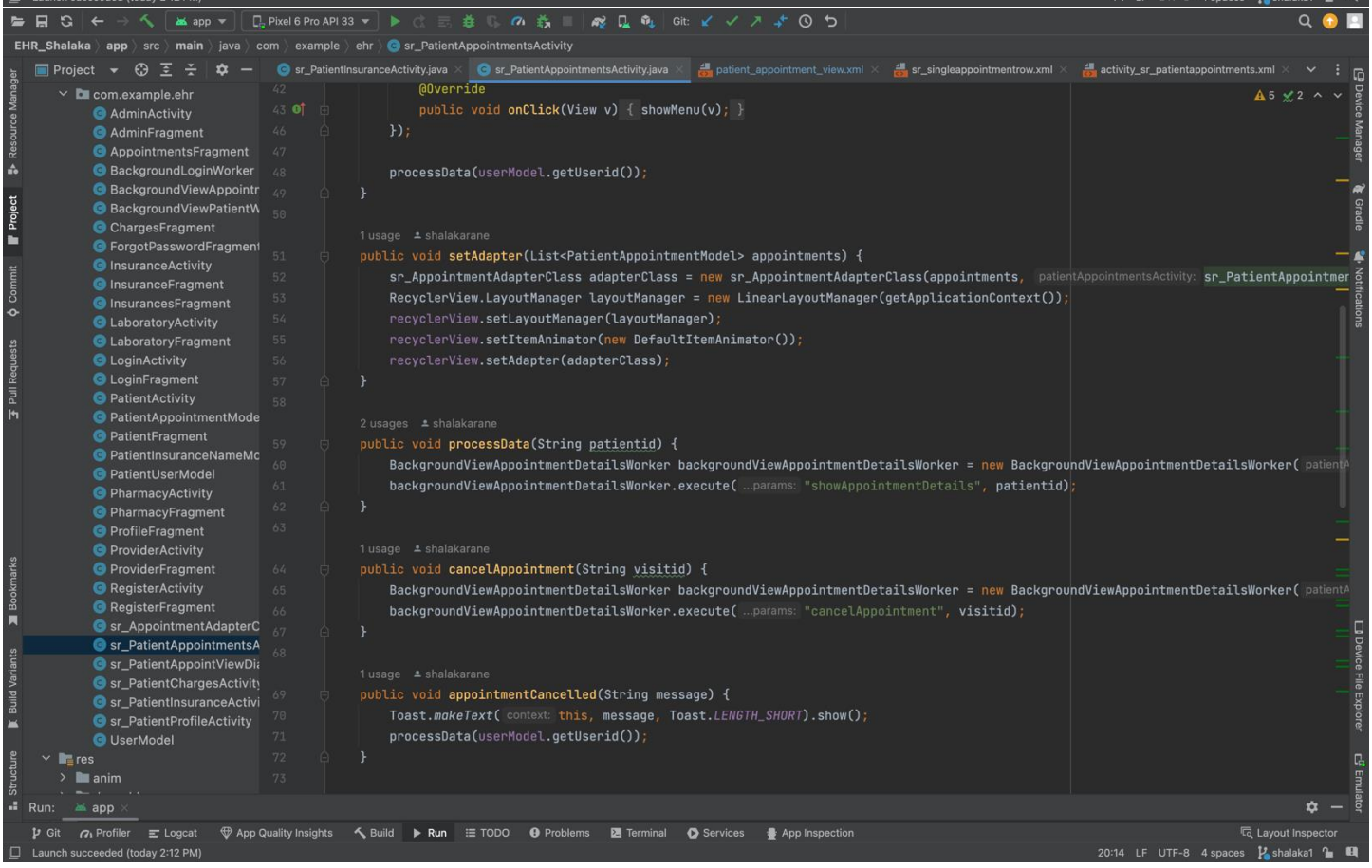


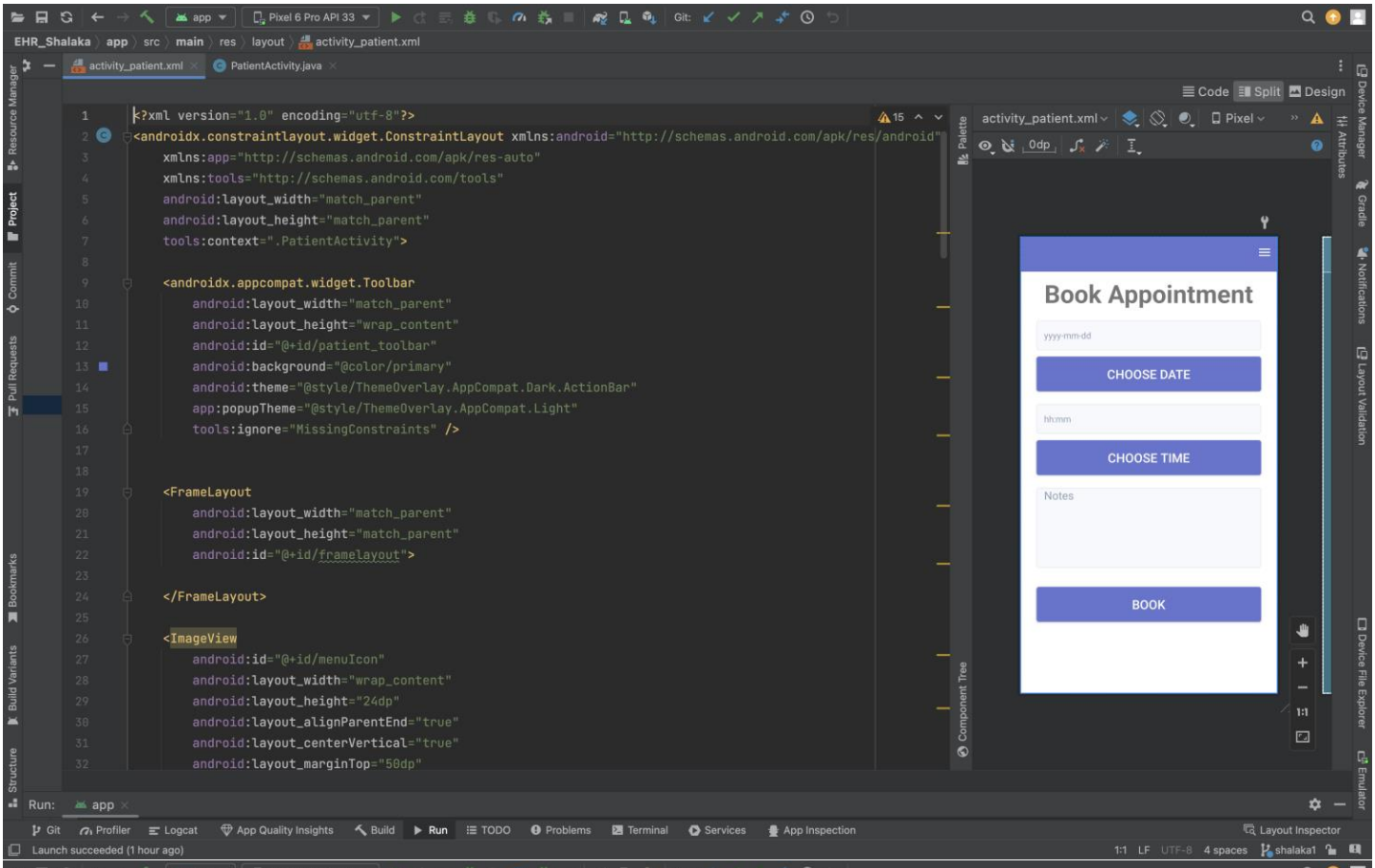

EHR_Shalaka app src main res layout fragment_register.xml
RegisterFragment.java
fragment_register.xml
Code Split Design
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".RegisterFragment">
<ScrollView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:fillViewport="true"
tools:ignore="UselessParent">
<LinearLayout
android:id="@+id/patient_register_details"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
android:paddingLeft="18dp"
android:paddingRight="18dp">
<TextView
android:id="@+id/sign_up_text"
android:layout_width="match_parent"
android:layout_height="50dp"
android:gravity="center"
android:text="Sign Up"
android:textSize="40sp"
android:textStyle="bold" />
<EditText
android:id="@+id/patient_register_firstname"
fragment_register.xml
Sign Up
First Name
Last Name
Email
Password
Confirm Password
Gender
DOB
Contact
Address1
Address2
City
State
Zip
REGISTER
Run: app
Launch succeeded (11 minutes ago)


EHR_Shalaka app src main java com example ehr RegisterFragment openDateDialog
RegisterFragment.java
fragment_register.xml
1 usage shalakarane
public void registerPatient(PatientUserModel patient) {
String baseURL = "https://sxr4177.uta.cloud/patientRegistration.php";
shalakarane
class dbprocess extends AsyncTask<String, Void, JSONObject> {
shalakarane
@Override
protected void onPostExecute(JSONObject resultObj) {
try {
if (resultObj.has(name: "error")) {
onRegisterFailed(resultObj.getString(name: "error"));
} else {
onRegisterSuccess();
}
} catch (Exception e) {
e.printStackTrace();
}
}
shalakarane
@Override
protected JSONObject doInBackground(String... strings) {
try {
URL url = new URL(strings[0]);
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("POST");
conn.setDoOutput(true);
conn.setDoInput(true);
OutputStream outputStream = conn.getOutputStream();
BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, StandardCharsets.UTF_8));
String post_data = URLEncoder.encode(s: "firstname", enc: "UTF-8") + "=" + URLEncoder.encode(patient.getFirstname(), enc: "UTF-8") + "&" +
Run: app
Launch succeeded (12 minutes ago)


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - activity_sb_provider_profile.xml [EHR.app.main]
EHR app src main res layout activity_sb_provider_profile.xml
Project
ProviderActivity.java
sb_ProviderScheduleActivity.java
activity_sb_provider_profile.xml
activity_login.xml
activity_main.xml
Code Split Design
activity_laboratory.xml
activity_login.xml
activity_main.xml
activity_patient.xml
activity_pharmacy.xml
activity_provider.xml
activity_sb_provider_analytics.xml
activity_sb_provider_appointment
activity_sb_provider_patientsearch
activity_sb_provider_profile.xml
activity_sb_scheduled_appointmen
activity_sb_view_appointment_deta
fragment_admin.xml
fragment_forgot_password.xml
fragment_insurance.xml
fragment_laboratory.xml
fragment_login.xml
fragment_patient.xml
fragment_pharmacy.xml
fragment_provider.xml
singlerowappointmentlist.xml
singlerowdesign.xml
menu
mipmap-anydpi-v26
mipmap-anydpi-v33
mipmap-hdpi
mipmap-mdpi
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xml
xmlns:app="http://schemas.android.com/apk/res-auto
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".sb_ProviderProfileActivity">
<androidx.appcompat.widget.Toolbar
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/provider_toolbar"
android:background="@color/primary"
android:theme="@style/ThemeOverlay.AppCompat.D
app:popupTheme="@style/ThemeOverlay.AppCompat_
tools:ignore="MissingConstraints" />
<androidx.fragment.app.FragmentContainerView
android:id="@+id/provider_nav_host"
android:name="androidx.navigation.fragment.Nav
android:layout_width="0dp"
android:layout_height="2dp"
android:layout_marginTop="52dp"
First Name
Last Name
Email
Password
Contact
Address Line 1
Address Line 2
City
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
1:1 LF UTF-8 4 spaces master
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - sb_ProviderProfileActivity.java [EHR.app.main]
java com example ehr sb_ProviderProfileActivity updateProvideProfile
PatientFragment
PharmacyActivity
PharmacyFragment
ProviderActivity
ProviderFragment
sb_apiController
sb_apiControllerAppointmentList
sb_apiset
sb_apisetAppointmentList
sb_myadapaterAppointmentList
sb_myadapter
sb_provider_analytics
sb_provider_appointment
sb_provider_patientsearch
sb_ProviderProfileActivity
sb_ProviderScheduleActivity
sb_ResponseModel_Appointments
sb_ResponseModel_Schedule
sb_Schedule
sb_ScheduledAppointment
sb_ViewAppointmentDetailsByID
UserModel
res
anim
slide_in_left.xml
slide_in_right.xml
slide_out_left.xml
LoginActivity.java
BackgroundLoginWorker.java
sb_ProviderProfileActivity.java
activity_sb_scheduled_appointme
package com.example.ehr;
import ...
public class sb_ProviderProfileActivity extends AppCompatActivity {
15 usages
UserModel user;
1 usage
public String userid1="";
2 usages
NavController navController;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_sb_provider_profile);
Toolbar toolbar = (Toolbar) findViewById(R.id.provider_toolbar);
setSupportActionBar(toolbar);
ImageView menuIcon=findViewById(R.id.menuIcon);
menuIcon.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
//Toast.makeText(ProviderActivity.this,"You clicked on the icon",Toast.LENGTH_SHORT).sl
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
240:54 LF UTF-8 4 spaces master


EHR_Shalaka › app › src › main › res › layout › activity_sr_patientappointments.xml
sr_PatientInsuranceActivity.java
sr_PatientAppointmentsActivity.java
patient_appointment_view.xml
sr_singleappointmentrow.xml
activity_sr_patientappointments.xml
Code
Split
Design
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/a
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".sr_PatientAppointmentsActivity">
<androidx.appcompat.widget.Toolbar
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/patient_toolbar"
android:background="@color/primary"
android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
tools:ignore="MissingConstraints" />
<ImageView
android:id="@+id/menuIcon"
android:layout_width="wrap_content"
android:layout_height="24dp"
android:layout_alignParentEnd="true"
android:layout_centerVertical="true"
android:layout_marginTop="50dp"
android:layout_marginEnd="8dp"
android:src="@drawable/baseline_menu_24"
app:layout_constraintBottom_toTopOf="@+id/patient_toolbar"
app:layout_constraintEnd_toEndOf="@id/patient_toolbar"
app:layout_constraintTop_toTopOf="parent" />
<LinearLayout
android:layout_width="match_parent"
Launch succeeded (today 2:12 PM)
1:1 LF UTF-8 4 spaces shalaka1


EHR_Shalaka › app › src › main › java › com › example › ehr › sr_PatientAppointmentsActivity
@Override
public void onClick(View v) { showMenu(v); }
});
processData(userModel.getUserid());
}
1 usage shalakarane
public void setAdapter(List<PatientAppointmentModel> appointments) {
sr_AppointmentAdapterClass adapterClass = new sr_AppointmentAdapterClass(appointments, patientAppointmentsActivity: sr_PatientAppointmer
RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());
recyclerView.setLayoutManager(layoutManager);
recyclerView.setItemAnimator(new DefaultItemAnimator());
recyclerView.setAdapter(adapterClass);
}
2 usages shalakarane
public void processData(String patientid) {
BackgroundViewAppointmentDetailsWorker backgroundViewAppointmentDetailsWorker = new BackgroundViewAppointmentDetailsWorker(patientA
backgroundViewAppointmentDetailsWorker.execute(...params: "showAppointmentDetails", patientid);
}
1 usage shalakarane
public void cancelAppointment(String visitid) {
BackgroundViewAppointmentDetailsWorker backgroundViewAppointmentDetailsWorker = new BackgroundViewAppointmentDetailsWorker(patientA
backgroundViewAppointmentDetailsWorker.execute(...params: "cancelAppointment", visitid);
}
1 usage shalakarane
public void appointmentCancelled(String message) {
Toast.makeText(context: this, message, Toast.LENGTH_SHORT).show();
processData(userModel.getUserid());
}
com.example.ehr
AdminActivity
AdminFragment
AppointmentsFragment
BackgroundLoginWorker
ChargesFragment
InsuranceActivity
InsuranceFragment
InsurancesFragment
LaboratoryActivity
LaboratoryFragment
LoginActivity
LoginFragment
PatientActivity
PatientFragment
PatientUserModel
PharmacyActivity
PharmacyFragment
ProfileFragment
ProviderActivity
ProviderFragment
RegisterActivity
RegisterFragment
UserModel
res
anim
Launch succeeded (today 2:12 PM)
20:14 LF UTF-8 4 spaces shalaka1

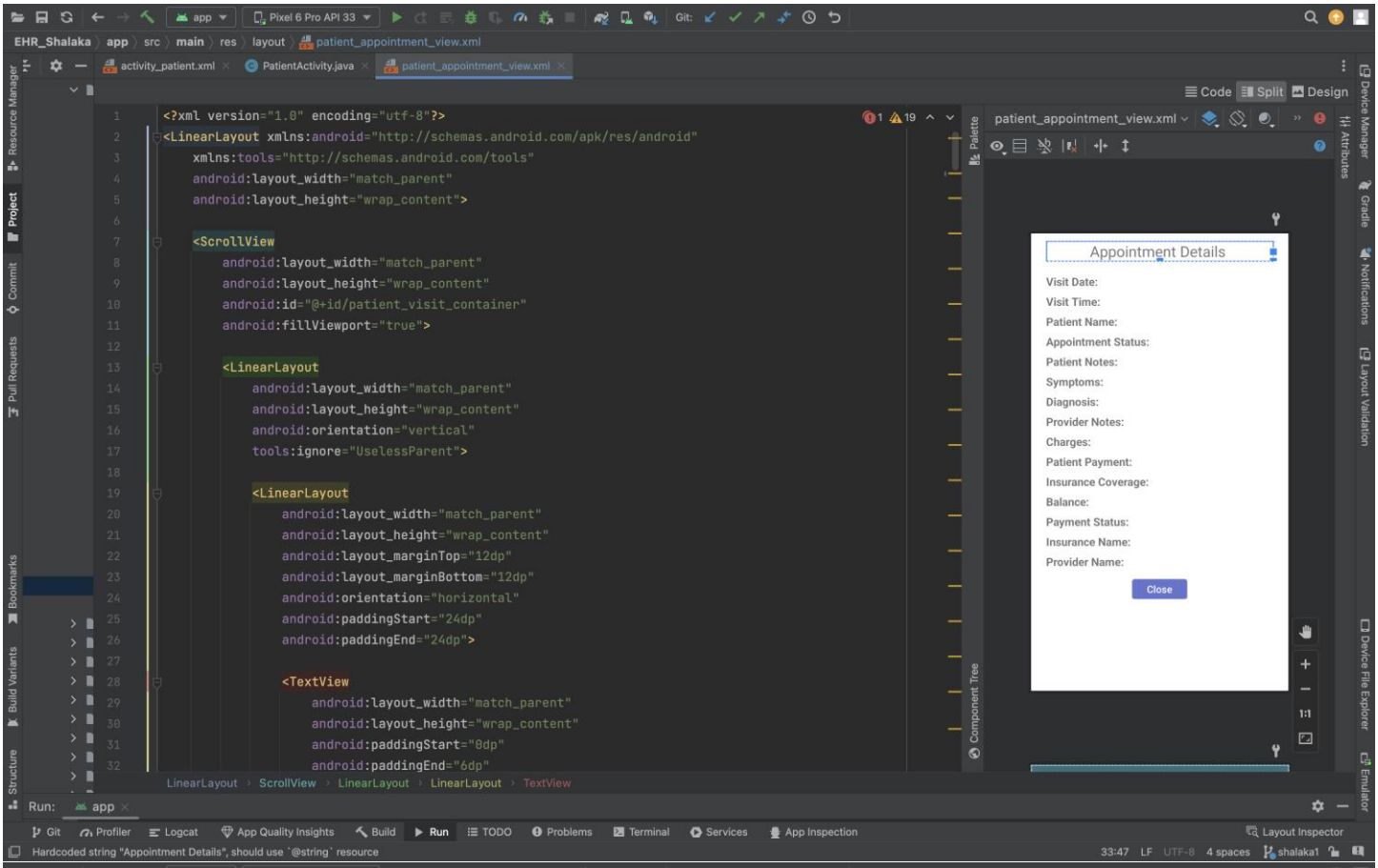


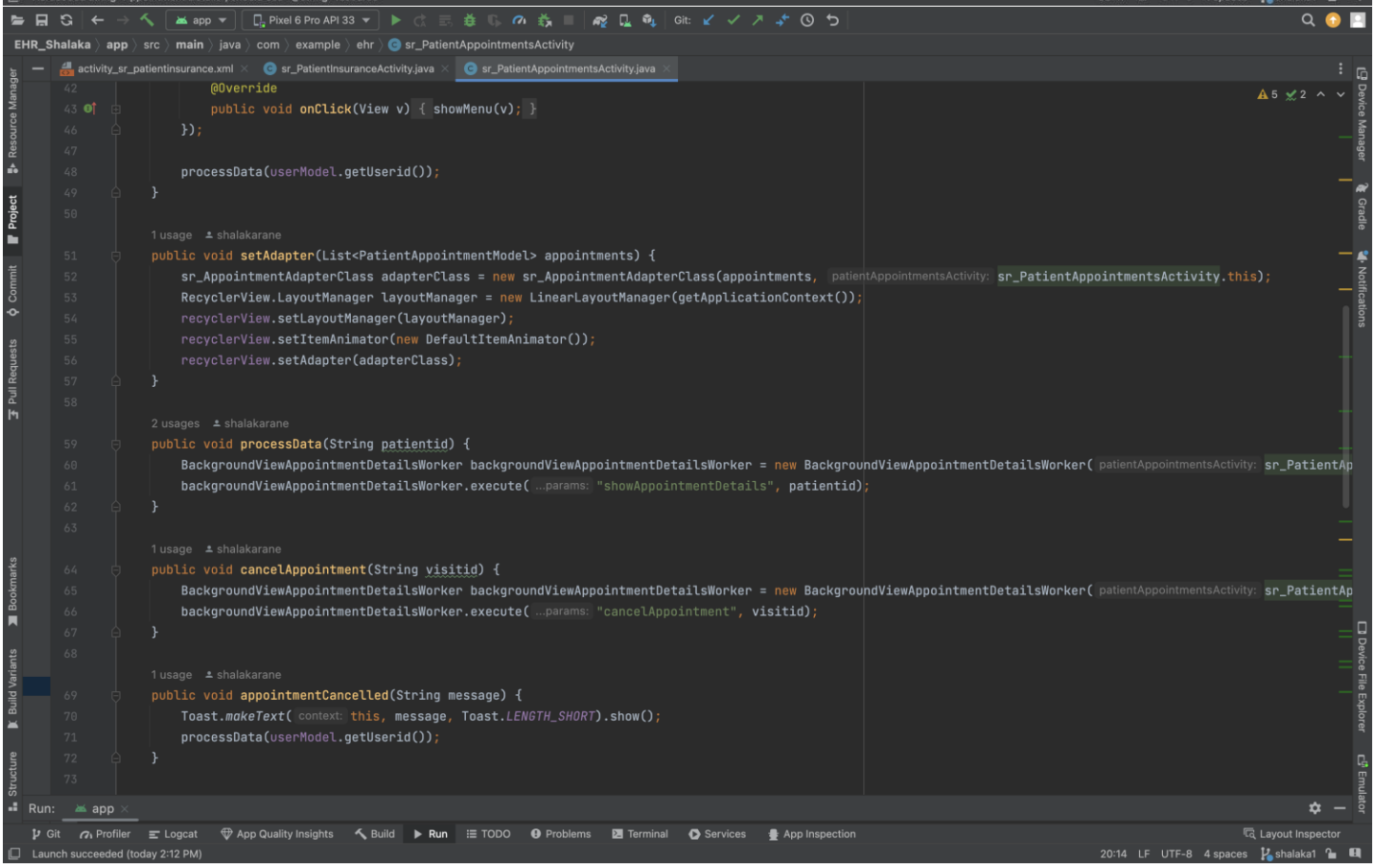

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PatientActivity">

    <androidx.appcompat.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/patient_toolbar"
        android:background="@color/primary"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
        tools:ignore="MissingConstraints" />


    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/framelayout">

    </FrameLayout>

    <ImageView
        android:id="@+id/menuIcon"
        android:layout_width="wrap_content"
        android:layout_height="24dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginTop="50dp"
```

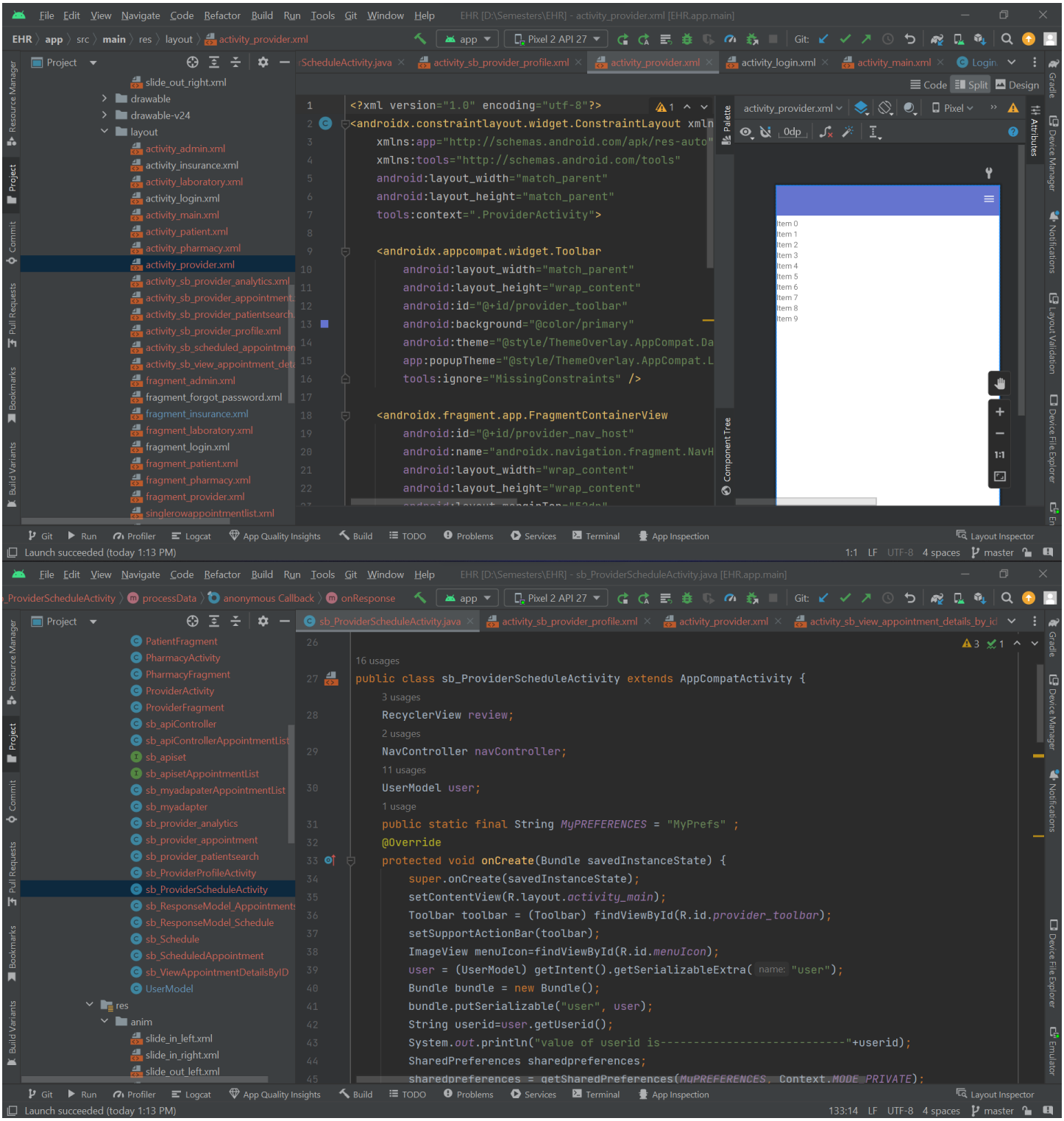
```
Launch succeeded (1 hour ago)
```

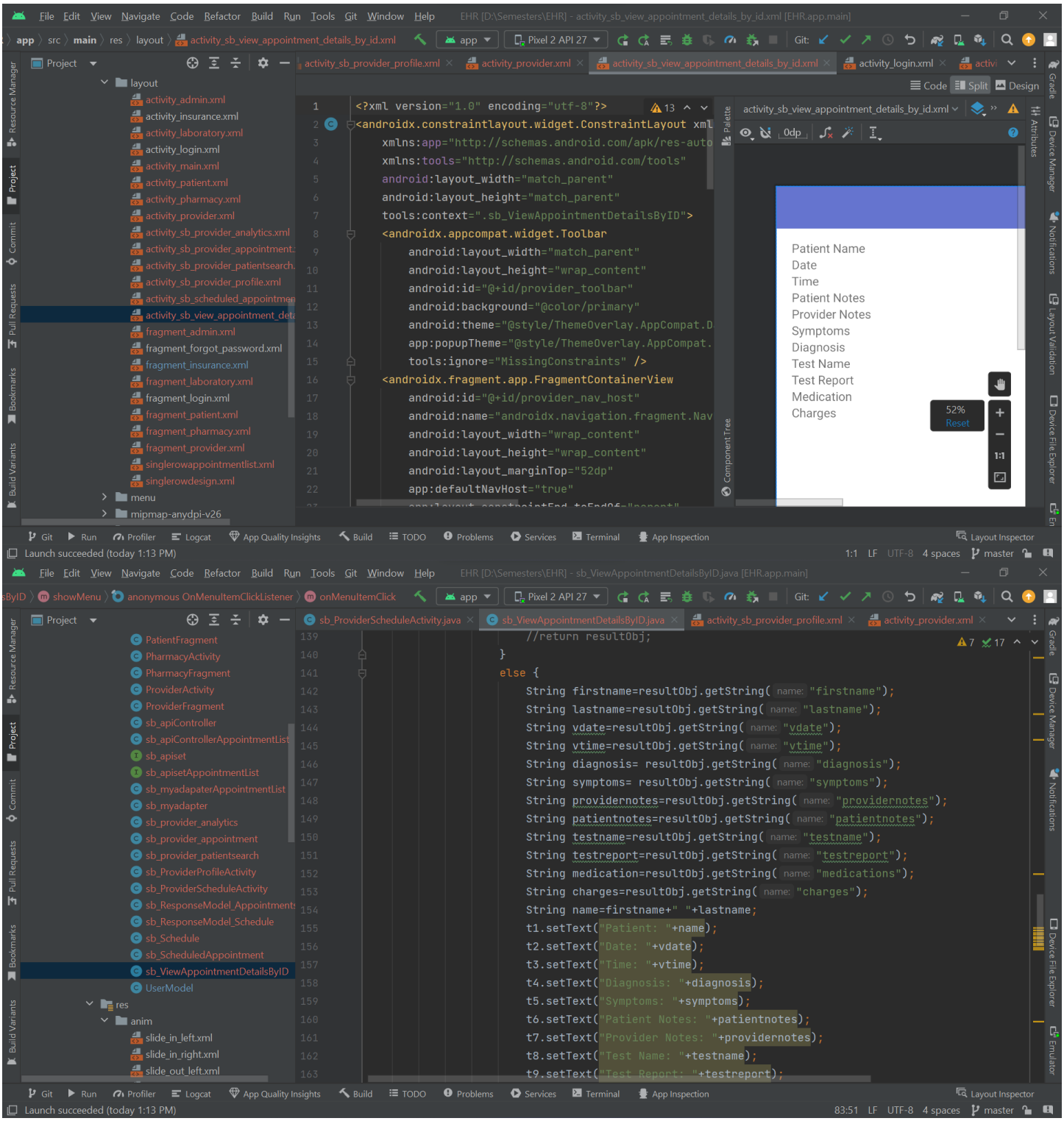
```
Launch succeeded (today 2:12 PM)
```

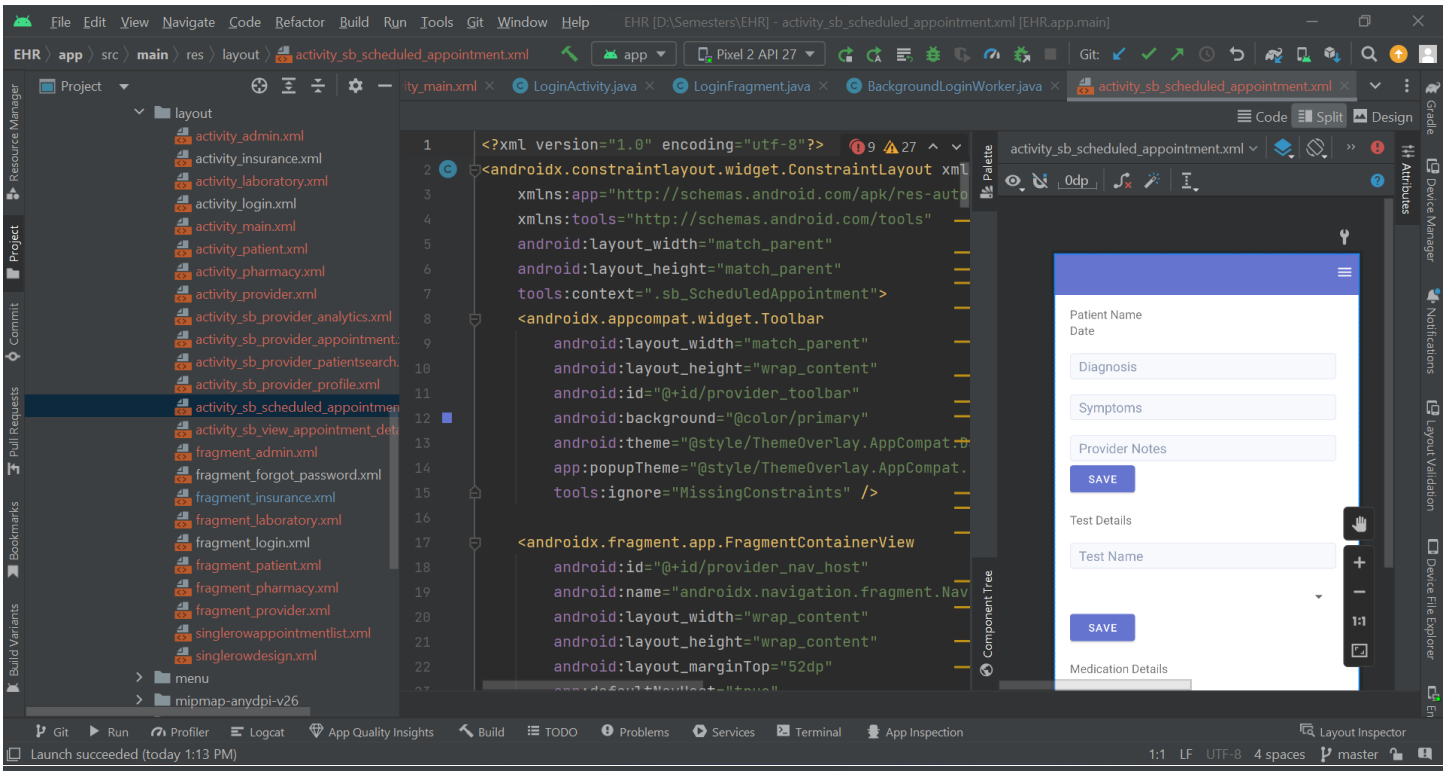
```
1 usage  shalakarane
private void bookAppointment() {

    String baseURL="https://sxr4177.uta.cloud/bookAppointment.php";

    shalakarane
    class dbprocess extends AsyncTask<String, Void, JSONObject>
    {

        shalakarane
        protected void onPostExecute(JSONObject resultObj) {
            System.out.println("Inside Execute");
            try {

                if(resultObj.has( name: "error"))
                {
                    Toast.makeText(getApplicationContext(),resultObj.getString( name: "error"),Toast.LENGTH_LONG).show();
                }
                else
                {
                    Toast.makeText(getApplicationContext(), text: "Appointment Booked",Toast.LENGTH_LONG).show();
                    vdate.setText("");
                    vtime.setText("");
                    patientnotes.setText("");

                    Intent intent = new Intent( packageContext: PatientActivity.this, sr_PatientAppointmentsActivity.class);
                    intent.putExtra( name: "user", user);
                    startActivity(intent);
                }
            }
            catch (Exception e)
            {
```

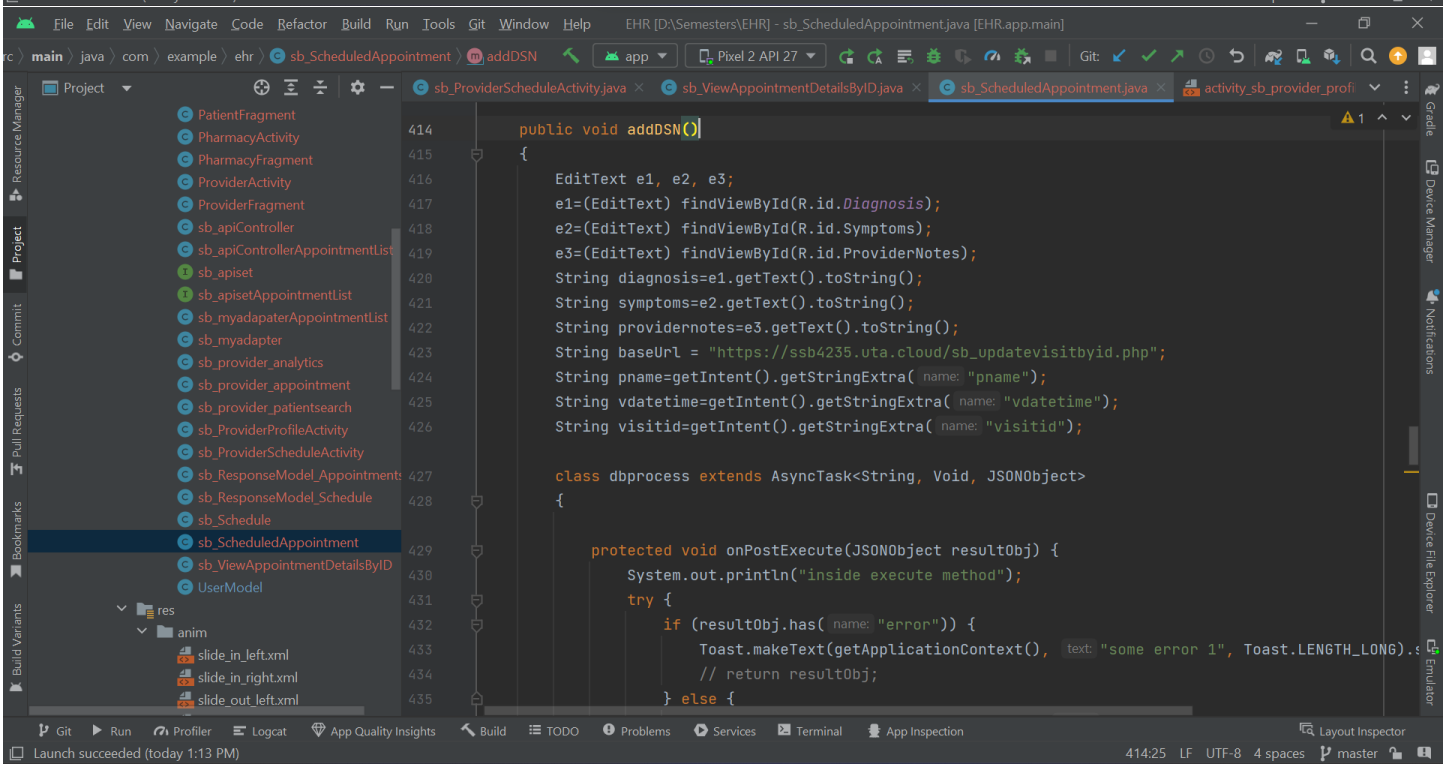

EHR_Shalaka › app › src › main › res › layout › patient_appointment_view.xml
activity_patient.xml
PatientActivity.java
patient_appointment_view.xml
Code
Split
Design
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="wrap_content">
<ScrollView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/patient_visit_container"
android:fillViewport="true">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
tools:ignore="UselessParent">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="12dp"
android:layout_marginBottom="12dp"
android:orientation="horizontal"
android:paddingStart="24dp"
android:paddingEnd="24dp">
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:paddingStart="0dp"
android:paddingEnd="6dp"
patient_appointment_view.xml
Appointment Details
Visit Date:
Visit Time:
Patient Name:
Appointment Status:
Patient Notes:
Symptoms:
Diagnosis:
Provider Notes:
Charges:
Patient Payment:
Insurance Coverage:
Balance:
Payment Status:
Insurance Name:
Provider Name:
Close
LinearLayout › ScrollView › LinearLayout › LinearLayout › TextView
Run: app
Hardcoded string "Appointment Details", should use `@string` resource


EHR_Shalaka › app › src › main › java › com › example › ehr › sr_PatientAppointmentsActivity
activity_sr_patientinsurance.xml
sr_PatientInsuranceActivity.java
sr_PatientAppointmentsActivity.java
@Override
public void onClick(View v) { showMenu(v); }
});
processData(userModel.getUserid());
}
1 usage shalakarane
public void setAdapter(List<PatientAppointmentModel> appointments) {
sr_AppointmentAdapterClass adapterClass = new sr_AppointmentAdapterClass(appointments, patientAppointmentsActivity: sr_PatientAppointmentsActivity.this);
RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());
recyclerView.setLayoutManager(layoutManager);
recyclerView.setItemAnimator(new DefaultItemAnimator());
recyclerView.setAdapter(adapterClass);
}
2 usages shalakarane
public void processData(String patientid) {
BackgroundViewAppointmentDetailsWorker backgroundViewAppointmentDetailsWorker = new BackgroundViewAppointmentDetailsWorker(patientAppointmentsActivity: sr_PatientAp
backgroundViewAppointmentDetailsWorker.execute(...params: "showAppointmentDetails", patientid);
}
1 usage shalakarane
public void cancelAppointment(String visitid) {
BackgroundViewAppointmentDetailsWorker backgroundViewAppointmentDetailsWorker = new BackgroundViewAppointmentDetailsWorker(patientAppointmentsActivity: sr_PatientAp
backgroundViewAppointmentDetailsWorker.execute(...params: "cancelAppointment", visitid);
}
1 usage shalakarane
public void appointmentCancelled(String message) {
Toast.makeText(context: this, message, Toast.LENGTH_SHORT).show();
processData(userModel.getUserid());
}
Run: app
Launch succeeded (today 2:12 PM)


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - activity_provider.xml [EHR.app.main]
EHR app src main res layout activity_provider.xml
Pixel 2 API 27
Project
slide_out_right.xml
drawable
drawable-v24
layout
activity_admin.xml
activity_insurance.xml
activity_laboratory.xml
activity_login.xml
activity_main.xml
activity_patient.xml
activity_pharmacy.xml
activity_provider.xml
activity_sb_provider_analytics.xml
activity_sb_provider_profile.xml
activity_sb_scheduled_appointmen
fragment_admin.xml
fragment_forgot_password.xml
fragment_insurance.xml
fragment_laboratory.xml
fragment_login.xml
fragment_patient.xml
fragment_pharmacy.xml
fragment_provider.xml
singlerowappointmentlist.xml
activity_sb_provider_profile.xml
activity_provider.xml
activity_login.xml
activity_main.xml
Code Split Design
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmln
3 xmlns:app="http://schemas.android.com/apk/res-auto"
4 xmlns:tools="http://schemas.android.com/tools"
5 android:layout_width="match_parent"
6 android:layout_height="match_parent"
7 tools:context=".ProviderActivity">
8
9 <androidx.appcompat.widget.Toolbar
10 android:layout_width="match_parent"
11 android:layout_height="wrap_content"
12 android:id="@+id/provider_toolbar"
13 android:background="@color/primary"
14 android:theme="@style/ThemeOverlay.AppCompat.Da
15 app:popupTheme="@style/ThemeOverlay.AppCompat.L
16 tools:ignore="MissingConstraints" />
17
18 <androidx.fragment.app.FragmentContainerView
19 android:id="@+id/provider_nav_host"
20 android:name="androidx.navigation.fragment.NavH
21 android:layout_width="wrap_content"
22 android:layout_height="wrap_content"
Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
1:1 LF UTF-8 4 spaces master
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - sb_ProviderScheduleActivity.java [EHR.app.main]
_ProviderScheduleActivity processData anonymous Callback onResponse
Pixel 2 API 27
Project
PatientFragment
PharmacyActivity
PharmacyFragment
ProviderActivity
ProviderFragment
sb_apiController
sb_apiControllerAppointmentList
sb_apiset
sb_apisetAppointmentList
sb_myadapaterAppointmentList
sb_myadapter
sb_provider_analytics
sb_provider_appointment
sb_provider_patientsearch
sb_ProviderProfileActivity
sb_ProviderScheduleActivity
sb_ResponseModel_Appointments
sb_ResponseModel_Schedule
sb_Schedule
sb_ScheduledAppointment
sb_ViewAppointmentDetailsByID
UserModel
res
anim
slide_in_left.xml
slide_in_right.xml
slide_out_left.xml
sb_ProviderScheduleActivity.java
activity_sb_provider_profile.xml
activity_provider.xml
activity_sb_view_appointment_details_by_id
16 usages
public class sb_ProviderScheduleActivity extends AppCompatActivity {
3 usages
RecyclerView review;
2 usages
NavController navController;
11 usages
UserModel user;
1 usage
public static final String MyPREFERENCES = "MyPrefs" ;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Toolbar toolbar = (Toolbar) findViewById(R.id.provider_toolbar);
setSupportActionBar(toolbar);
ImageView menuIcon=findViewById(R.id.menuIcon);
user = (UserModel) getIntent().getSerializableExtra(name: "user");
Bundle bundle = new Bundle();
bundle.putSerializable("user", user);
String userid=user.getUserid();
System.out.println("value of userid is----------"+userid);
SharedPreferences sharedpreferences;
sharedpreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
133:14 LF UTF-8 4 spaces master


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - activity_sb_view_appointment_details_by_id.xml [EHR.app.main]
app src main res layout activity_sb_view_appointment_details_by_id.xml
Pixel 2 API 27
Project
activity_sb_provider_profile.xml
activity_provider.xml
activity_sb_view_appointment_details_by_id.xml
activity_login.xml
Code Split Design
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xml
xmlns:app="http://schemas.android.com/apk/res-auto
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".sb_ViewAppointmentDetailsByID">
<androidx.appcompat.widget.Toolbar
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/provider_toolbar"
android:background="@color/primary"
android:theme="@style/ThemeOverlay.AppCompat.D
app:popupTheme="@style/ThemeOverlay.AppCompat.
tools:ignore="MissingConstraints" />
<androidx.fragment.app.FragmentContainerView
android:id="@+id/provider_nav_host"
android:name="androidx.navigation.fragment.Nav
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="52dp"
app:defaultNavHost="true"
Patient Name
Date
Time
Patient Notes
Provider Notes
Symptoms
Diagnosis
Test Name
Test Report
Medication
Charges
52%
Reset
Launch succeeded (today 1:13 PM)
1:1 LF UTF-8 4 spaces master
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - sb_ViewAppointmentDetailsByID.java [EHR.app.main]
sb_ProviderScheduleActivity.java
sb_ViewAppointmentDetailsByID.java
activity_sb_provider_profile.xml
activity_provider.xml
//return resultObj;
}
else {
String firstname=resultObj.getString(name: "firstname");
String lastname=resultObj.getString(name: "lastname");
String vdate=resultObj.getString(name: "vdate");
String vtime=resultObj.getString(name: "vtime");
String diagnosis= resultObj.getString(name: "diagnosis");
String symptoms= resultObj.getString(name: "symptoms");
String providernotes=resultObj.getString(name: "providernotes");
String patientnotes=resultObj.getString(name: "patientnotes");
String testname=resultObj.getString(name: "testname");
String testreport=resultObj.getString(name: "testreport");
String medication=resultObj.getString(name: "medications");
String charges=resultObj.getString(name: "charges");
String name=firstname+" "+lastname;
t1.setText("Patient: "+name);
t2.setText("Date: "+vdate);
t3.setText("Time: "+vtime);
t4.setText("Diagnosis: "+diagnosis);
t5.setText("Symptoms: "+symptoms);
t6.setText("Patient Notes: "+patientnotes);
t7.setText("Provider Notes: "+providernotes);
t8.setText("Test Name: "+testname);
t9.setText("Test Report: "+testreport);
Launch succeeded (today 1:13 PM)
83:51 LF UTF-8 4 spaces master


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - activity_sb_scheduled_appointment.xml [EHR.app.main]
EHR app src main res layout activity_sb_scheduled_appointment.xml
app
Pixel 2 API 27
Git:
Project
layout
activity_admin.xml
activity_insurance.xml
activity_laboratory.xml
activity_login.xml
activity_main.xml
activity_patient.xml
activity_pharmacy.xml
activity_provider.xml
activity_sb_provider_analytics.xml
activity_sb_provider_appointment
activity_sb_provider_patientsearch
activity_sb_provider_profile.xml
activity_sb_scheduled_appointmen
activity_sb_view_appointment_det
fragment_admin.xml
fragment_forgot_password.xml
fragment_insurance.xml
fragment_laboratory.xml
fragment_login.xml
fragment_patient.xml
fragment_pharmacy.xml
fragment_provider.xml
singlerowappointmentlist.xml
singlerowdesign.xml
menu
mipmap-anydpi-v26
LoginActivity.java
LoginFragment.java
BackgroundLoginWorker.java
activity_sb_scheduled_appointment.xml
Code Split Design
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xml
xmlns:app="http://schemas.android.com/apk/res-auto
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".sb_ScheduledAppointment">
<androidx.appcompat.widget.Toolbar
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/provider_toolbar"
android:background="@color/primary"
android:theme="@style/ThemeOverlay.AppCompat.D
app:popupTheme="@style/ThemeOverlay.AppCompat.
tools:ignore="MissingConstraints" />
<androidx.fragment.app.FragmentContainerView
android:id="@+id/provider_nav_host"
android:name="androidx.navigation.fragment.Nav
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="52dp"
activity_sb_scheduled_appointment.xml
Patient Name
Date
Diagnosis
Symptoms
Provider Notes
SAVE
Test Details
Test Name
SAVE
Medication Details
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
1:1 LF UTF-8 4 spaces master


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
EHR [D:\Semesters\EHR] - sb_ScheduledAppointment.java [EHR.app.main]
main java com example ehr sb_ScheduledAppointment addDSN
app
Pixel 2 API 27
Git:
Project
PatientFragment
PharmacyActivity
PharmacyFragment
ProviderActivity
ProviderFragment
sb_apiController
sb_apiControllerAppointmentList
sb_apiset
sb_apisetAppointmentList
sb_myadapaterAppointmentList
sb_myadapter
sb_provider_analytics
sb_provider_appointment
sb_provider_patientsearch
sb_ProviderProfileActivity
sb_ProviderScheduleActivity
sb_ResponseModel_Appointments
sb_ResponseModel_Schedule
sb_Schedule
sb_ScheduledAppointment
sb_ViewAppointmentDetailsByID
UserModel
res
anim
slide_in_left.xml
slide_in_right.xml
slide_out_left.xml
sb_ProviderScheduleActivity.java
sb_ViewAppointmentDetailsByID.java
sb_ScheduledAppointment.java
activity_sb_provider_profi
public void addDSN()
{
EditText e1, e2, e3;
e1=(EditText) findViewById(R.id.Diagnosis);
e2=(EditText) findViewById(R.id.Symptoms);
e3=(EditText) findViewById(R.id.ProviderNotes);
String diagnosis=e1.getText().toString();
String symptoms=e2.getText().toString();
String providernotes=e3.getText().toString();
String baseUrl = "https://ssb4235.uta.cloud/sb_updatevisitbyid.php";
String pname=getIntent().getStringExtra(name: "pname");
String vdatetime=getIntent().getStringExtra(name: "vdatetime");
String visitid=getIntent().getStringExtra(name: "visitid");
class dbprocess extends AsyncTask<String, Void, JSONObject>
{
protected void onPostExecute(JSONObject resultObj) {
System.out.println("inside execute method");
try {
if (resultObj.has(name: "error")) {
Toast.makeText(getApplicationContext(), text: "some error 1", Toast.LENGTH_LONG).s
// return resultObj;
} else {
Git Run Profiler Logcat App Quality Insights Build TODO Problems Services Terminal App Inspection Layout Inspector
Launch succeeded (today 1:13 PM)
414:25 LF UTF-8 4 spaces master

YouTube Link

https://www.youtube.com/watch?v=yVhuLitbdtI&t=28s