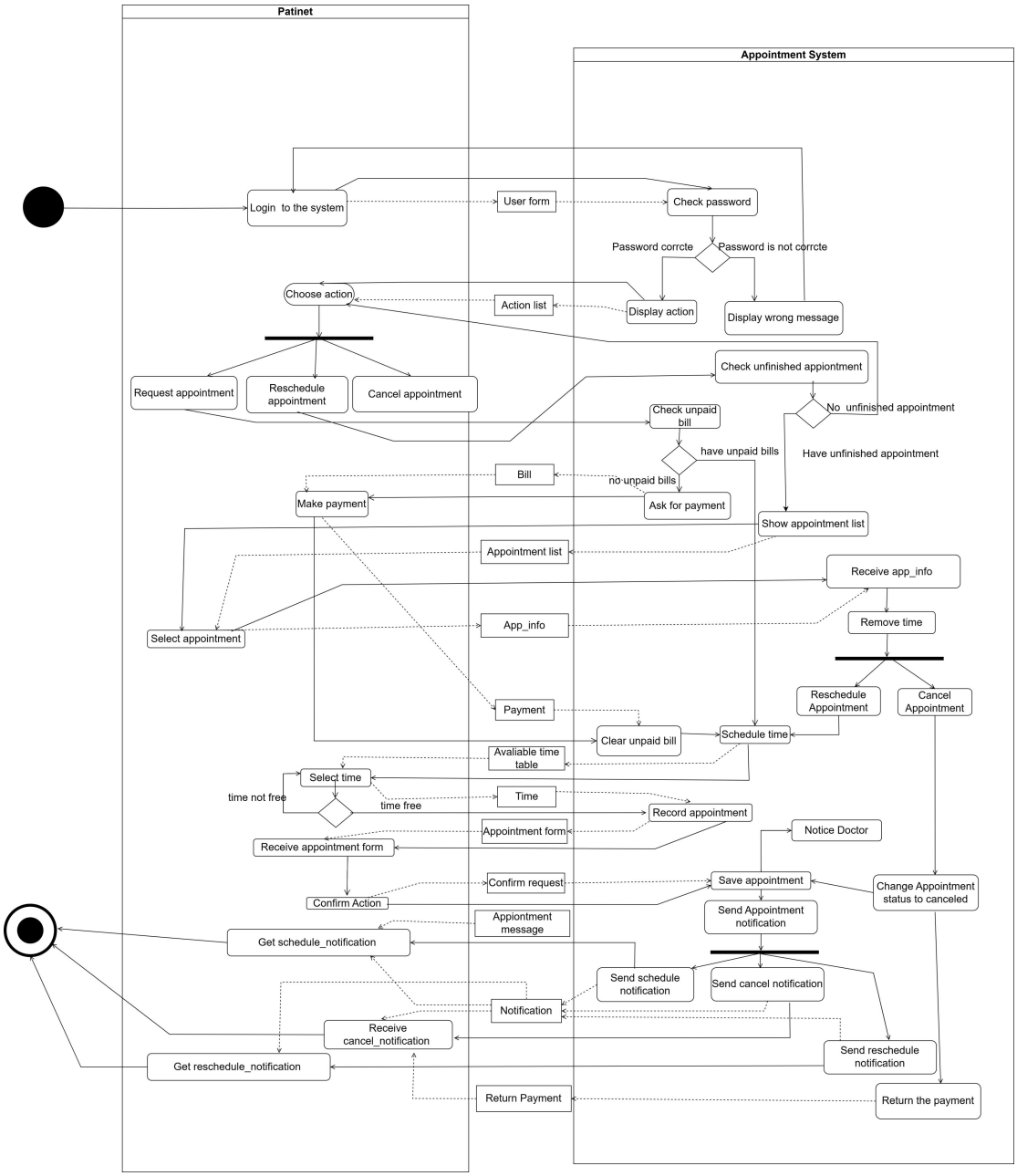
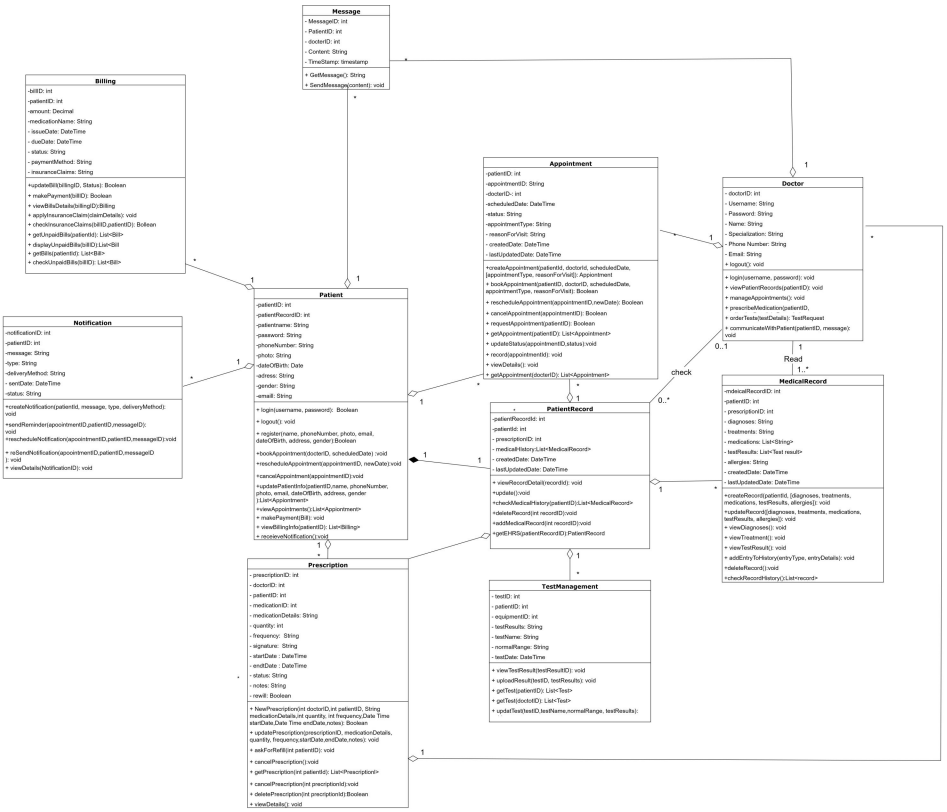
Task 1: Software Modelling and Specification

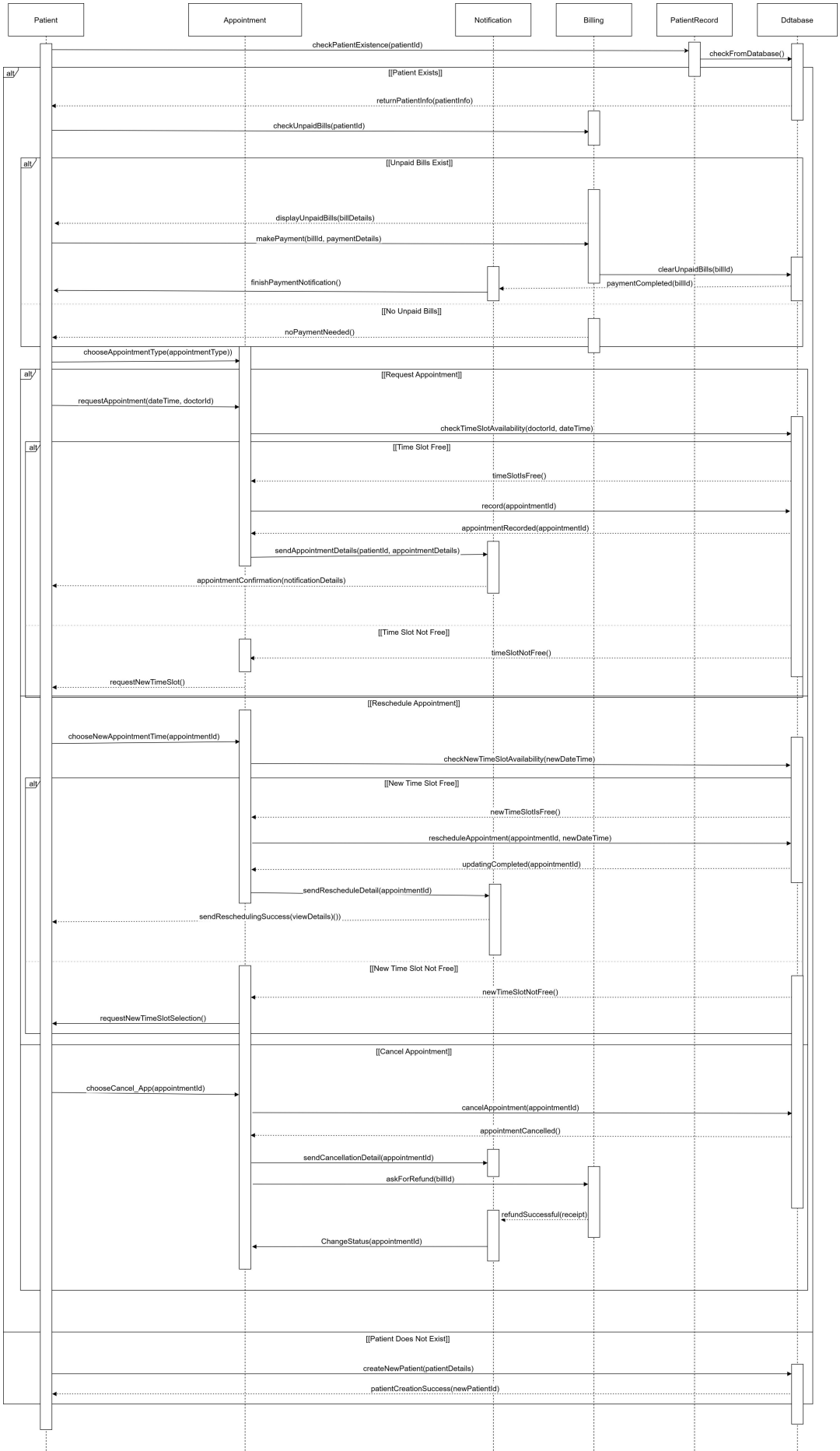
(b) Activity Model



(c) Structural Model

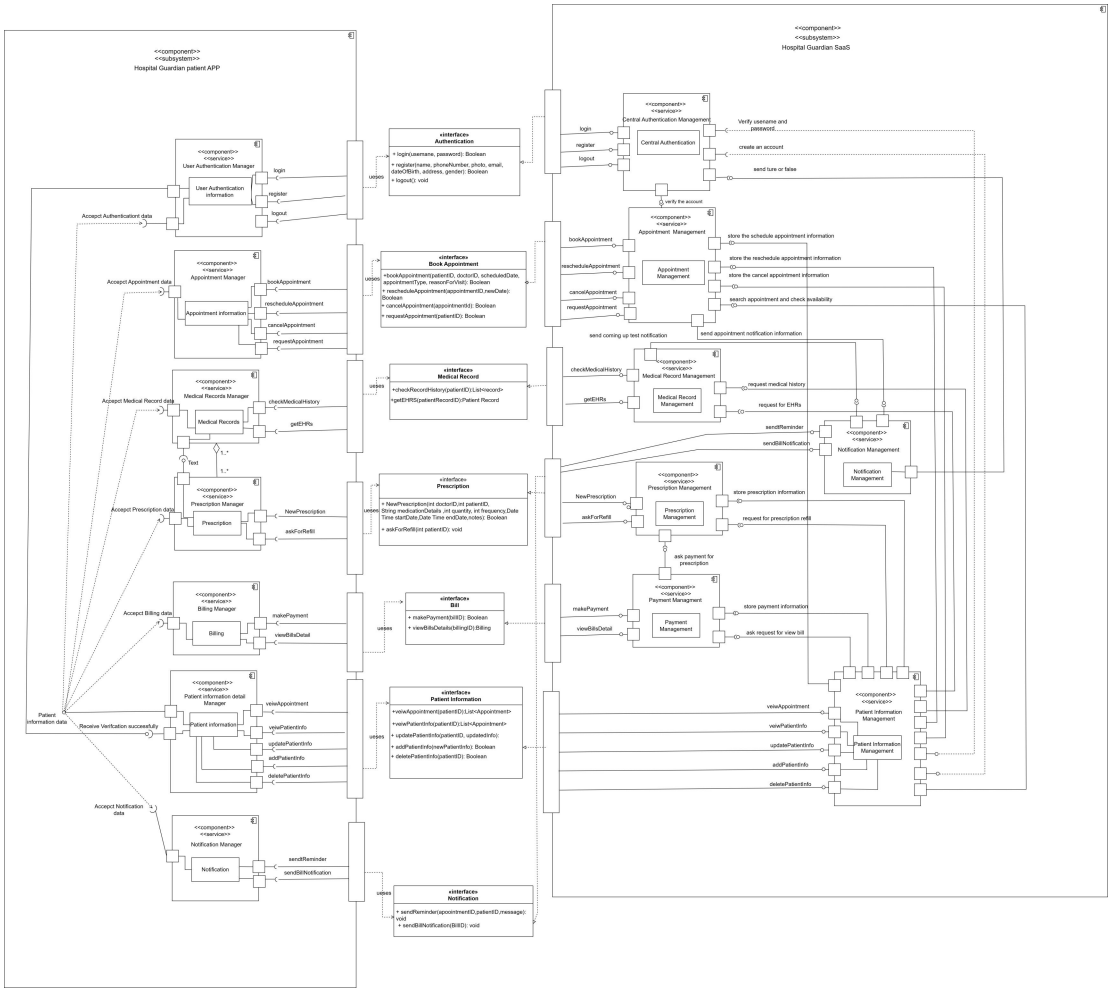


(d) Behaviour Model


Patient
Appointment
Notification
Billing
PatientRecord
Ddtabase
checkPatientExistence(patientId)
checkFromDatabase()
alt
[[Patient Exists]]
returnPatientInfo(patientInfo)
checkUnpaidBills(patientId)
alt
[[Unpaid Bills Exist]]
displayUnpaidBills(billDetails)
makePayment(billId, paymentDetails)
clearUnpaidBills(billId)
finishPaymentNotification()
paymentCompleted(billId)
[[No Unpaid Bills]]
noPaymentNeeded()
chooseAppointmentType(appointmentType))
alt
[[Request Appointment]]
requestAppointment(dateTime, doctorId)
checkTimeSlotAvailability(doctorId, dateTime)
alt
[[Time Slot Free]]
timeSlotIsFree()
record(appointmentId)
appointmentRecorded(appointmentId)
sendAppointmentDetails(patientId, appointmentDetails)
appointmentConfirmation(notificationDetails)
[[Time Slot Not Free]]
timeSlotNotFree()
requestNewTimeSlot()
[[Reschedule Appointment]]
chooseNewAppointmentTime(appointmentId)
checkNewTimeSlotAvailability(newDateTime)
alt
[[New Time Slot Free]]
newTimeSlotIsFree()
rescheduleAppointment(appointmentId, newDateTime)
updatingCompleted(appointmentId)
sendRescheduleDetail(appointmentId)
sendRescheduling Success(viewDetails)())
[[New Time Slot Not Free]]
newTimeSlotNotFree()
requestNewTimeSlotSelection()
[[Cancel Appointment]]
chooseCancel_App(appointmentId)
cancelAppointment(appointmentId)
appointmentCancelled()
sendCancellationDetail(appointmentId)
askForRefund(billId)
refundSuccessful(receipt)
ChangeStatus(appointmentId)
[[Patient Does Not Exist]]
createNewPatient(patientDetails)
patientCreationSuccess(newPatientId)

Task 2: Software Architectural Design

(b) Architecture of the subsystem



Task 2: Software Architectural Design (Table)

Component Table

Component	User Authentication Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about login, logout and register etal other authentication and manage them
Type	Health Guardian Patient APP
Required Interfaces:	Register, Login, Logout
Provided Interfaces:	Receive Verification successfully

Component	Appointment Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about schedule, reschedule and cancel appointment etal and manage them.
Type	Health Guardian Patient APP
Required Interfaces:	bookAppointment,rescheduleAppointment, cancelAppointment. requestAppointment,

	Accepct Appointment data

Component	Medical Records Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about check medical history and get EHRs medical records and manage them.
Type	Health Guardian Patient APP
Required Interfaces:	checkMedicalHistory, getEHRs, Accepct Medical Record data

Component	Prescription Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about prescription like asking new prescription and asking for refill and manage them.
Type	Health Guardian Patient APP
Required Interfaces:	NewPrescription, askForRefill, Accepct Prescription data

Component	Billing Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about billing like making payment and asking viewing bills details and manage them.
Type	Health Guardian Patient APP
Required Interfaces:	makePayment viewBillsDetail Accepct Billing data

Component	Patient information detail Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about patient information like viewing appointments, viewing patient information, updating patient information, adding patient and delete patient etal. and manage patient information.
Type	Health Guardian Patient APP
Required Interfaces:	Receive Verification successfully, veiwAppointment, veiwPatientInfo, updatePatientInfo, addPatientInfo, deletePatientInfo
Provided Interfaces:	Patient data

Component	Notification Manager
Description	This is a subsystem running Health Guardian Patient APP used for all request about notification like sending reminder and send bill notification and manage them.
Type	Health Guardian Patient APP
Required Interfaces:	Accepct Notification data, sendtReminder, sendBillNotification

Component	Patient Information Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the different patient information and some operation.
Type	Microservice
Required Interfaces:	veiwAppointment, veiwPatientInfo, updatePatientInfo, addPatientInfo, deletePatientInfo, create an account , store the cancel appointment information , store the reschedule appointment information , store the schedule appointment information, request medical history, request for EHRs
Provided Interfaces:	search appointment and check availability, Verify usename and password

Component	Payment Managment
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about Payment to support subsystem.
Type	Microservice
Required Interfaces:	ask request for view bill
Provided Interfaces:	makePayment, viewBillsDetail, ask payment for prescription, store payment information

Component	Prescription Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about Prescription.
Type	Microservice
Required Interfaces:	request for prescription refill, ask payment for prescription
Provided Interfaces:	NewPrescription, askForRefill, store prescription information

Component	Medical Record Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about medical record.
Type	Microservice
Required Interfaces:	request medical history, request for EHRs
Provided Interfaces:	checkMedicalHistory, getEHRs, send coming up test notification.

Component	Notification Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about notification.
Type	Microservice

Required Interfaces:	sendBillNotification, sendtReminder send coming up test notification, send appointment notification information, send true or false

Component	Appointment Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about appointment.
Type	Microservice
Required Interfaces:	verify the account, search appointment and check availability
Provided Interfaces:	bookAppointment,rescheduleAppointment, cancelAppointment, requestAppointment send appointment notification information store the cancel appointment information store the reschedule appointment information store the schedule appointment information

Component	Central Authentication Management
Description	This is a microservice running on the Health Guardian Patient APP that runs on the cloud and handles the session about central autentication
Type	Microservice
Required Interfaces:	Verify usename and password
Provided Interfaces:	Login, register, logout, create an account , send ture or false, verify the account.

Interface Table

Interface Name	Authentication
Provider	Hospital Guardian SaaS
	Operation: login
	Request Parameters: String username String password Response parameters: Boolean (return true if login is successful, return false if failure) Function: This operation checks if the provided username and password match the credentials stored in the system. If they match, the operation returns true, indicating a successful login. If they do not match, it returns false.
	Operation: logout
	Request Parameters: None Response parameters: void (no direct response, but the system will log out the user) Function: This operation logs out the user from the system,

	terminating any active session.
	Operation: register
	Request Parameters: String name ,String phoneNumber, String photo, String email,Date dateOfBirth, String address, String gender Response parameters: Boolean (ture if the patient register successfully, if the patient register failure because the name have been used or other some errors, return flase) Function: This operation registers a new user with the provided information after checking if the username (which could be assumed to be part of the email or a separate field) is already in use. If the username is available, the user's information is stored in the system

Interface Name	Book Appointment
Provider	Health Guardian Saas
	Operation: bookAppointment
	Request Parameters: int patientID, int doctorID , DateTime scheduledDate, String appointmentType ,String reasonForVisit, Response parameters: Boolean (true if the appointment is successfully booked, false and return to appoint another time or doctor) Function: his operation schedules a new appointment for a patient with a specific doctor on a specified date, appointment type, and reason for the visit. It returns true if the booking is successful.
	Operation: cancelAppointment
	Request Parameters: int appointmentID Response parameters: Boolean (true if the appointment is successfully canceled, false return to) Function: This operation cancels an existing appointment. It returns true if the cancellation is successful.
	Operation: rescheduleAppointment
	Request Parameters: int appointmentID, DateTime newDate Response parameters: Boolean (true if the appointment is successfully reschedule, false and return to appoint another time or doctor) Function: This operation reschedules exsited appointment for a patient on a specified date, It returns true if the booking is successful.
	Operation: requestAppointment
	Request Parameters: int patientID Response parameters: Boolean (true if the appointment request is successfully registered, false jump to error message) Function: Allows a patient to request an appointment without

	specifying a date. The system will acknowledge the request. Returns true if the request is processed successfully.

Interface Name	Medical Record
Provider	Health Guardian Saas
	Operation: checkRecordHistory
	Request Parameters: int patientID Response parameters: List<Medical Record>A list of medical records noted before in a patient record. Function: Retrieves the entire history of medical records for a patient.
	Operation: getEHRs
	Request Parameters: int patientID Response parameters: The electronic healthy PatientRecor. Function: Fetches a specific Electronic Health PatientRecords (EHRs) based on the provided patientRecordID.

Interface Name	Prescription
Provider	Health Guardian Saas
	Operation: NewPrescription
	Request Parameters: int doctorID int patientID ,String medication,int quantit,String frequency DateTime startDate,DateTime endDate,String notes Response parameters: Boolean (true if the prescription is successfully created, false otherwise) Function: Creates a new prescription with the specified details. Returns true if the prescription is successfully added to the system.
	Operation: askForRefill
	Request Parameters: int patientID Response parameters:void Function: Fetches a specific Electronic Health PatientRecords (EHRs) based on the provided patientRecordID.

Interface Name	Bill
Provider	Health Guardian Saas
	Operation: makePayment
	Request Parameters: int billID int patientID ,String medication,int quantit,String frequency DateTime startDate,DateTime endDate,String notes Response parameters: Boolean: Returns true if the payment is

	successfully processed and the bill status is updated to "paid" or "settled". Returns false if the payment fails due to reasons such as invalid bill ID, insufficient funds, incorrect billing details, expired credit card, network issues, or payment gateway errors. Function: carries out a bill payment transaction using the billID. The system will process the payment, carry out validations, and provide the outcome of the operation.
	Operation: viewBillsDetails
	Request Parameters: int billID Response parameters: Billing includes: amount, medicationName, issueDate, status, paymentMethod String notes: Any additional notes or comments related to the bill, such as payment plan arrangements or discounts applied. Function: Uses the supplied billingID to obtain the full billing details for a specific bill. Bill data are displayed to patients or administrative personnel via this process.

Interface Name	Patient information
Provider	Health Guardian Saas
	Operation: veiwAppointment
	Request Parameters: int patientID Response parameters: List<Appointment>: A list of Appointment objects containing details of all appointments associated with the patientID. Function: Obtains a patient's complete appointment schedule. The list of Appointment objects contains information about the doctor, purpose, date, and time of each appointment.
	Operation: viewPatientInfo
	Request Parameters: int patientID Response parameters: List<appointment>: A list of appointment objects containing detailed personal and medical information for the patient Function: Retrieves detailed personal and medical information for a specific patient. This could include demographics, contact information, medical history, allergies, and current medications.

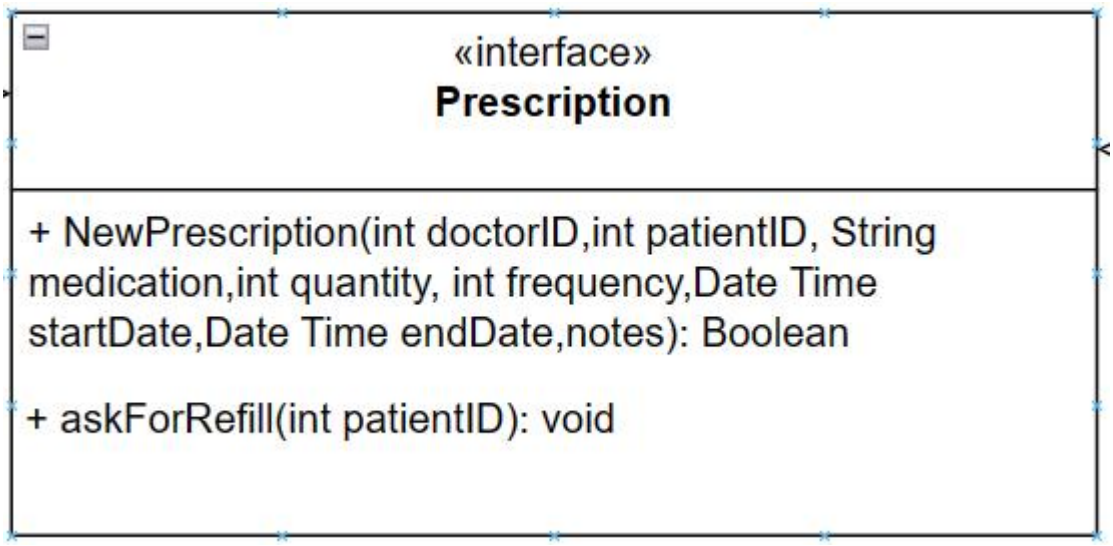
Interface Name	Notification
Provider	Health Guardian Saas
	Operation: sendReminder
	Request Parameters: int billID String patientID ,String appointmentID, String message Response parameters: void Function: Notifies the patient via email that they have an appointment coming up. The message contains information about

	the appointment, including the date, time, and any prep guidelines.
	Operation: sendBillNotification
	Request Parameters: int billID Response parameters: void String notes: Notifies the patient or the person in charge of any new or unpaid bills. Details like the bill amount, the due date, and payment instructions are included in the message.

Task 3: Software Testing

(a) Unit test plan

- The case interface:



- Test Cases for NewPrescription Method

Test CaseID	Test Case	Method and Parameters	Expected Output
Test Case01	Successful Prescription Creation	NewPrescription(101, 201, "Ibuprofen", 200, 3 times per day,, "2023-12-01T09:00", "2023-12-15T09:00", "Take after food")	Return true and a new prescription record is created in the database with the provided details.
Test Case02	Prescription Creation with Future Dates	NewPrescription(101, 201, "Ibuprofen", 200, 3 times per day,, "2024-01-01T09:00", "2024-01-15T09:00", "Take after food")	Return true and a new prescription record is created with a future start date, indicating the prescription is valid

			and scheduled.
Test Case03	Prescription with Invalid Patient ID	NewPrescription(101, -1, "Ibuprofen", 200, 3 times per day, "2023-12-01T09:00", "2023-12-15T09:00", "Take after food")	Return false and no new prescription is created; an error message or log entry indicates an invalid patient ID.
Test Case04	Prescription with Incomplete Data	NewPrescription(101, 201, "", 0, 0, "2023-12-01T09:00", "2023-12-15T09:00", "")	Return false and no new prescription is created; an error message or log entry indicates missing or invalid medication details.

- Test Cases for askForRefill Method

Test CaseID	**Test Case**	**Pre-Condition**	**Method and Parameters**	**Expected Output**
Test Case05	Refill request for an existing prescription	Prescription with ID 101 exists and is eligible for a refill.	askForRefill(101)	Refill request is logged in the system, and an internal process for refill is initiated.
Test Case06	Refill request for an existing prescription	Prescription with ID 999 does not exist.	askForRefill(999)	Refill request is logged in the system, and an internal process for refill is initiated.
Test Case07	Refill request for a prescription that cannot be refilled	Prescription with ID 102 exists but has reached its refill limit.	askForRefill(102)	An error is logged stating that the prescription has no refills left, and no refill is initiated.
Test Case08	Refill request for a prescription	Prescription with ID 103 exists but has	askForRefill(103)	An error is logged stating that the

	that has expired	expired.		prescription has no refills left, and no refill is initiated.

(b) System test plan

Use case: Prescription Requests

- Scenario: Requesting a New Prescription

Test Prograss

Patient	System (HealthGuardian SaaS)
1.Logs into the system and navigates to the prescription request section.	2.Authenticates the patient and presents the prescription form.
3.Fills out the prescription request form with medication details and submits.	4.Validates the request and forwards it to the associated healthcare provider.
5.Awaits confirmation of the prescription request.	6.Provider reviews the request and, if appropriate, approves the new prescription.
7.Receives a notification about the status of the prescription request.	8.Updates the patient's records with the new prescription details.

Test Data

(1) Input: doctorID = 101, patientID = 201, medication = "Ibuprofen", quantity = 200, frequency =3 times per day , startDate = "2023-12-01T09:00", endDate = "2023-12-31T09:00", notes = "Take after food"

(2) Expected Output: Confirmation of prescription creation return True.

- Scenario: Requesting a Prescription Refill

Test Process:

Patient	System (HealthGuardian SaaS)
1.Selects an existing prescription and requests a refill online.	2.System checks the refill eligibility based on the prescription's refill rules.
3.Confirms the refill request details and submits.	4.If eligible, system sends a refill request to the healthcare provider for approval.
5.Awaits notification of the refill request approval.	6.Provider reviews and approves the refill request.
7.Receives confirmation of the prescription refill.	8.System logs the refill and notifies the pharmacy, if integrated.

Test Data:

(1) Input: patientID = 201
(2) Expected Output: Patient receives a system-generated confirmation message of the refill request submission. Patient receives a notification confirming the healthcare provider's approval. The system logs the interaction and, if applicable, sends a notification to the pharmacy to dispense the medication.

These options guarantee extensive testing of the prescription request feature, encompassing both new prescription creation and refill requests. While the system tests concentrate on the entire process involving patient activities and system responses to make sure the functionality complies with the high-priority criteria, the unit tests check that each individual method operates as intended.