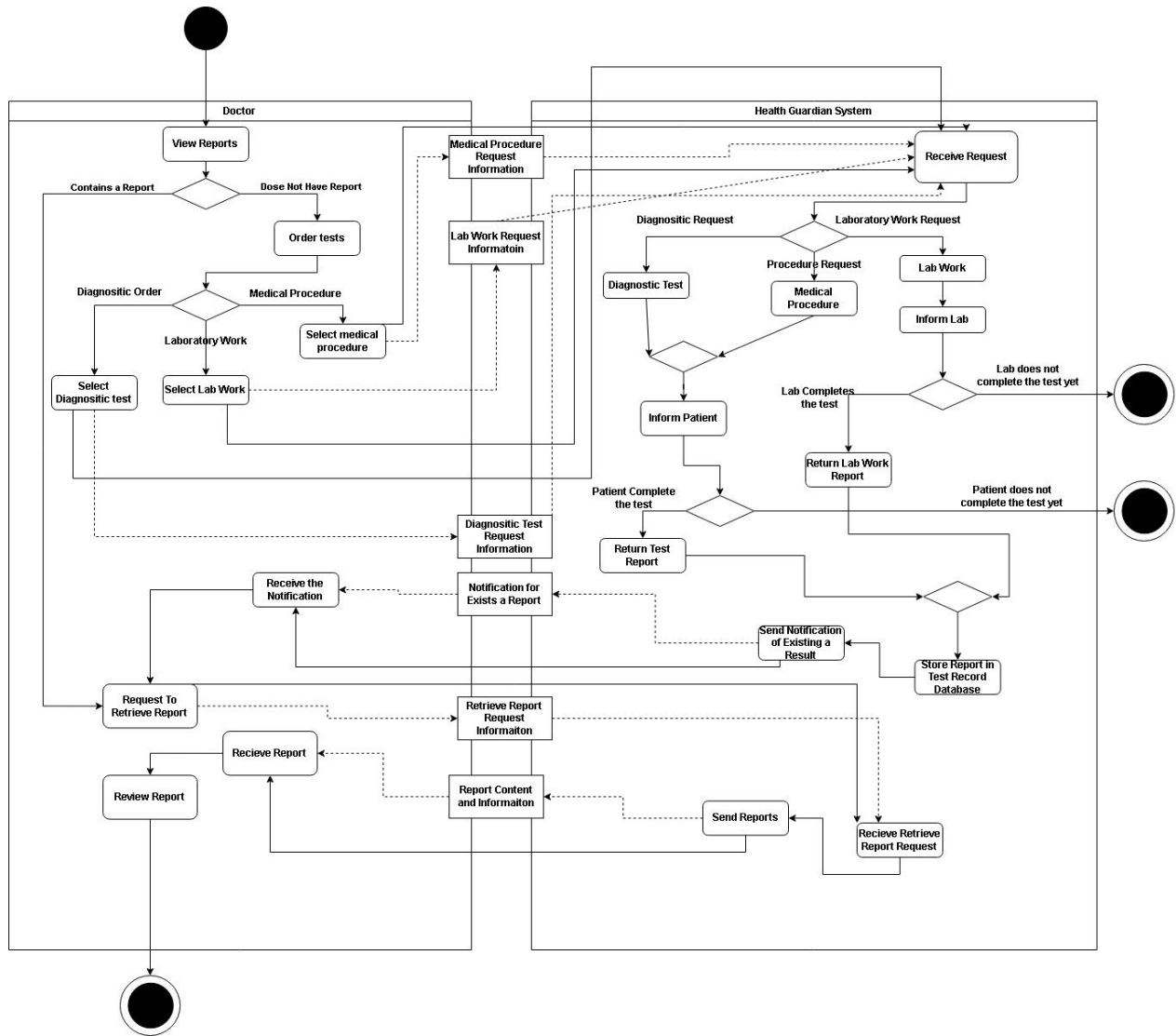
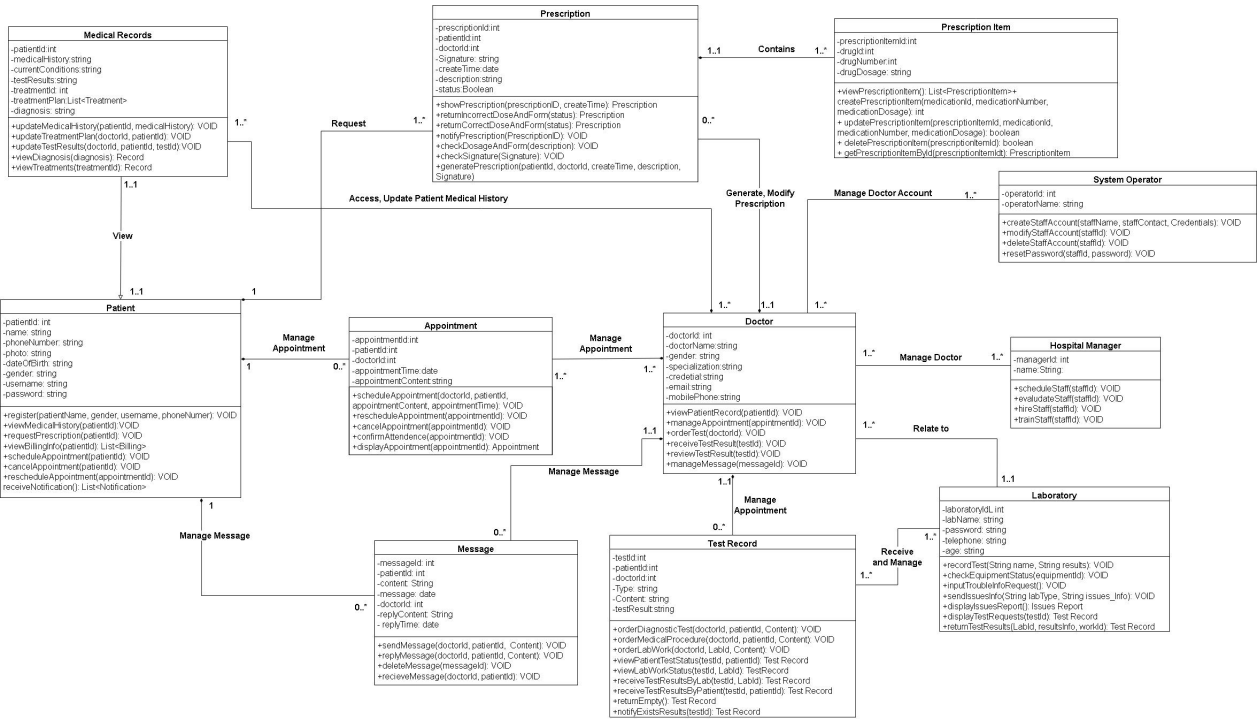


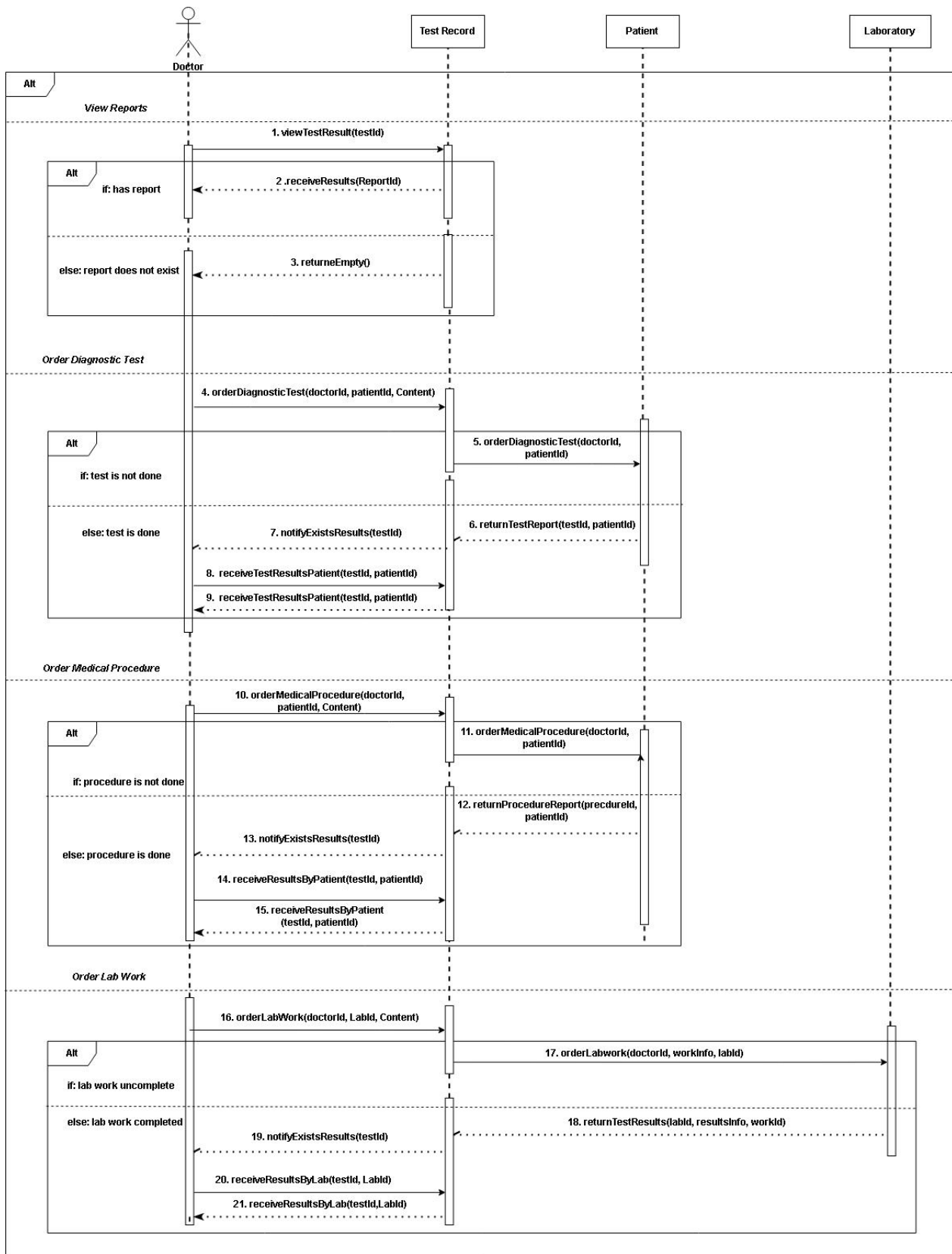
(b) Activity Diagram



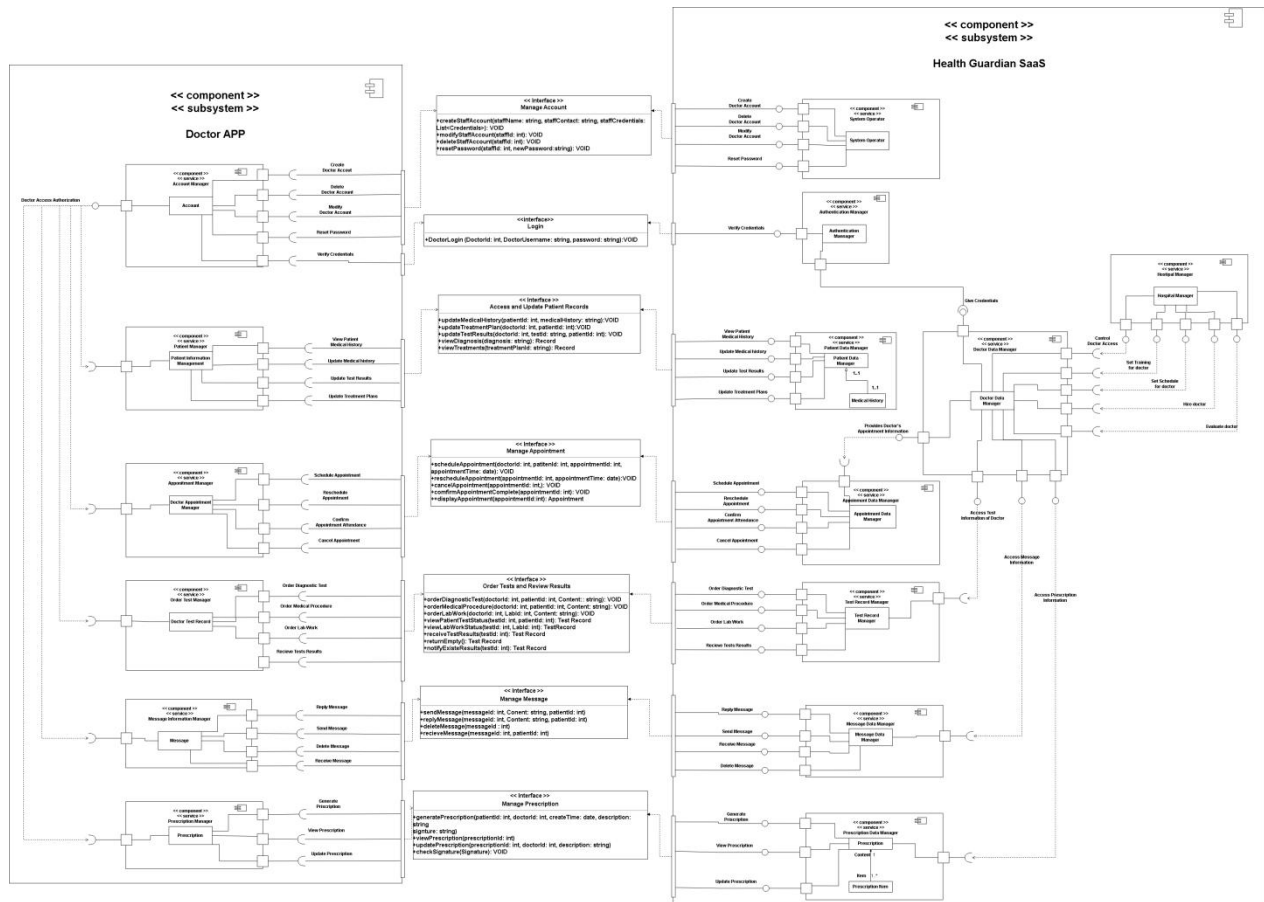
(c) Class Diagram



(d) Sequence Diagram



(b) Individual Component Diagram



Task 2: Software Architecture Design- Subsystem Architecture (Component Table)

Component	System Operator
Description	This is a micro service system running on the cloud system and
Type	Microservice
Provided Interface	Create Doctor Account Delete Doctor Account Modify Doctor Account Reset Staff Account Password
Required Interface	N/A

Component	Authentication Manager
Description	This is a manager system on health guardian cloud system , which is used for authenticate the identity of the staff member , in this case, doctor.
Type	Microservice
Provided Interface	Verify Credentials
Required Interface	Give Credentials

Component	Patient Data Manager
Description	This is a data management system running on Health Guardian System within the cloud database for patient medical history information storage.
Type	Microservice
Provided Interface	View Patient Medical History Update Medical History Update Test Results Update Treatment Plans
Required Interface	N/A

Component	Doctor Data Manager
Description	It is a microservice system operating on cloud system of health guardian specified for doctor data management which provides the essential information and data, for instance, providing the appointment information when the doctor app is required.
Type	Microservice
Provided Interface	Give Credentials, Provides Appointment Information, Access Test Information of Doctor. Access Message Information, Access Prescription generated by doctor
Required Interface	Control Doctor Access, Set Training for Doctor, Hire Doctor, Evaluate Doctor

Component	Hospital Manager
Description	The is the managing system operating on the cloud service of Health Guardian to manage the specific.
Type	Microservice
Provided Interface	Control Doctor Access, Set Training for Doctor, Hire Doctor, Evaluate Doctor
Required Interface	N/A

Component	Appointment Data Manager
Description	It is a data managing directly operating on cloud server of Health Guardian system for managing specific information of appointments related to doctor.
Type	Microservice
Provided Interface	Schedule Appointment Reschedule Appointment Confirm Appointment Attendance Cancel Appointment
Required Interface	Provides Doctor's Appointment Information

Component	Test Record Manager
Description	It is a data record system for recording and storing the information of the test doctor had requested either to lab or patient. And also contains the returned information of test, for instance, report, issues etc
Type	Microservice
Provided Interface	Order Diagnostic Test Order Medical Procedure Order Lab Work Receive Tests Results
Required Interface	Access Test Information of Doctor

Component	Message Data Manager
Description	It is a data manage system that operates on the server of Health Guardian System, which is used for storing messaging data of between doctors and patients. It contains detail information of communication records including consultants, communications etc.
Type	Microservice
Provided Interface	Reply Message Send Message Delete Message Receive Message
Required Interface	Access Message Information

Component	Prescription Data Manager
Description	This is a system operates on the Health Guardian Cloud server that stores the information of the prescription which includes the specifications generated by doctors and details involved within the prescription, for instance, the dosage, multiple drugs, drug procedure.
Type	Microservice
Provided Interface	Generate Prescription View Prescription Update Prescription
Required Interface	Access Prescription Information

Component	Account Manager
Description	Account Manager is a Doctor APP subsystem operates on cloud system to manage the account of doctor. It includes the account information of doctors, modification operation.
Type	Microservice
Provided Interface	Doctor Access Authorization
Required Interface	Create Doctor Account Delete Doctor Account Modify Doctor Account Verify Credentials Reset Password

Component	Patient Manager
Description	This is the doctor manage system of doctor app, operating on the cloud system. For managing the accessing the patient information such as patient medical history only.
Type	Microservice
Provided Interface	N/A
Required Interface	Doctor Account Authorization View Patient Medical History Update Medical History Update Test Results Update Treatment Plans

Component	Appointment Manager
Description	Appointment Manage System is a subsystem of doctor APP which operates on cloud system that manages the access of appointment information of doctor.
Type	Microservice
Provided Interface	N/A
Required Interface	Schedule Appointment Reschedule Appointment Confirm Appointment Attendance Cancel Appointment

Component	Order Test Manager
Description	Test Manager is a service of Doctor APP, and it manages the test that doctor had requested, which includes information such as test type, test content, returned reports.
Type	Microservice
Provided Interface	N/A
Required Interface	Order Diagnostic Test Order Medical Procedure Order Lab Work Receive Tests Results Doctor Access Authorization

Component	Message Information Manager
Description	The is the manage system of Doctor APP that operates on the cloud server contain the message between doctors and patients. It controls the access of message information.
Type	Microservice
Provided Interface	N/A
Required Interface	Reply Message Send Message Delete Message Receive Message

Component	Prescription Manager
Description	Prescription Manager system is the service within the Doctor APP that operates on cloud server which manages the prescription related information
Type	Microservice
Provided Interface	N/A
Required Interface	Generate Prescription View Prescription Update Prescription

Task 2.1: Software Architecture Description, Interface Table

Interface Name	Manage Account
Interface Provider	Health Guardian SaaS
	Operation: createStaffAccount
	<p>Request Parameters: staffName, staffContact, staffCredentials</p> <p>Response Parameters: StaffID, Success Status, Error Message</p> <p>Function: Create the account for staff with basic information including the doctors basic information and a unique staff ID given by system</p>
	Operation: modifyStaffAccount
	<p>Request Parameters: StaffId</p> <p>Response Parameters: StaffId, If Success, If Failed</p> <p>Function: Modify the account for, such as the basic information, contact information etc.</p>
	Operation: deleteStaffAccount
	<p>Request Parameters: StaffId</p> <p>Response Parameters: StaffId, If deleted Successfully, delete failed, incorrect StaffId</p> <p>Function: Delete the account of the staff once the staff had resigned or left the official job, or the staff had violated regulations of the hospital</p>
	Operation: resetPassword
	<p>Request Parameters: StaffId, new password</p> <p>Response Parameters: Success Message, Failed Message</p>

	Function: Reset the password of the doctor if the doctor had issue with their accounts
--	--

Interface Name	Login
Interface Provider	Health Guardian SaaS
	Operation: Login
	Request Parameters: DoctorId, DoctorUserName, password Response Parameters: Doctor Id, Success Message, failed message, Error In Credentials Function: Doctor is able to login their accounts after the authentication manager on the cloud system verified the credentials

Interface Name	Access and Update Patient Records
Interface Provider	Health Guardian SaaS
	Operation: updateMedicalHistory
	Request Parameters: patientId, medicalHistory Response Parameters: Successfully displayed, Error in Displaying the Message, Empty Message, Function: Updates the patient medical history by first accessing the patient basic information then modify the medical history, such as diagnose history, medical procedure history, allergy history
	Operation: updateTreatmentPlan
	Request Parameters: Response Parameters: Patient Information Management Function: Update the treatment plan for the patient on patients' medical records
	Operation: updateTestResults
	Request Parameters: doctorId, patientId, testId Response Parameters: Patient Information Management

	Function: Update the test results of patient, such as the results after procedure, diagnosis
	Operation: viewDiagnosis
	Request Parameters: diagnosis Response Parameters: If success, If Failed, error message for displaying the results Function: View the diagnosis details of patient which the test was requested by doctors
	Operation: viewTreatment
	Request Parameters: treatmentId Response Parameters: Success Displayed Message, Error Displaying Message, Treatment Content Empty Function: View the treatment content that doctor had assigned or updated to the corresponding patient

Interface Name	Manage Appointment
Interface Provider	Health Guardian SaaS
	Operation: scheduleAppointment
	Request Parameters: doctorId, patientId, appointmentId appointmentTime Response Parameters: If Successful, If Failed, Error Appointment Function: Allows doctor to schedule an appointment with patient
	Operation: rescheduleAppointment
	Request Parameters: appointmentId, appointmentTime Response Parameters: If Successful Rescheduled, If Failed Reschedule, Error Appointment Rescheduling Function: Allows doctor to reschedule the appointment which had been set earlier according to the their time

	table
	Operation: cancelAppointment
	Request Parameters: appointmentId Response Parameters: If Successful, If Failed, Error Appointment Function: Allows doctor to cancel the appointment set earlier
	Operation: confirmAppointment
	Request Parameters: appointmentId Response Parameters: If Successful, If Failed, Error Appointment Function: Allows the doctor to confirm the attendance of the patient and therefore complete the current appointment
	Operation: displayAppointment
	Request Parameters: appointmentId Response Parameters: If Successful, If Failed, Error appointment, Empty Appointment FunctionL Allows the doctor to check the appointment that they had made

Interface Name	Order Tests and Review Results
Interface Provider	Health Guardian SaaS
	Operation: orderDiagnosticTest
	Request Parameters: doctorId, patientId Response Parameters: If Successful, If Failed, Error Test Message Function: Doctors order the test contents and type to specific patients
	Operation: ordetMedicalProcedure
	Request Parameters: doctorId, patientId

	<p>Response Parameters: If Successful, If Failed, Error Test Message</p> <p>Function: Doctors order the corresponding medical procedure for patient to take and undergo</p>
	Operation: orderLabWork
	<p>Request Parameters: doctorId, LabId</p> <p>Response Parameters: If Successful, If Failed, Error Test or Work Message, Lab Not Available</p> <p>Function: Doctor is able to order specific laboratory works for the lab</p>
	Operation: viewPatient TestStatus
	<p>Request Parameters: testId, patientId</p> <p>Response Parameters: If Successful, If Failed, Error Test Message, Test Expired Message</p> <p>Function: Allows doctors to check whether the test they gave are still available and active</p>
	Operation: viewLabWork
	<p>Request Parameters: testId, LabId</p> <p>Response Parameters: If Successful, If Failed, Error Work Message</p> <p>Function: Doctors could check the status and details of the work request they sent, which keep track of the test or work.</p>
	Operation: receiveTestResults
	<p>Request Parameters: testId</p> <p>Response Parameters: If Successful, If empty, Error Message of the results</p> <p>Function: Doctors are able to decide whether they are going to receive the message into their database.</p>

Interface Name	Manage Message
Interface Provider	Health Guardian SaaS
	Operation: sendMessage
	<p>Request Parameters: messageId messageContent</p> <p>Response Parameters: If Successful, If Failed, Error Message</p> <p>Function: Doctors can send the message to the patient to communicate and make consultant</p>
	Operation: replyMessage
	<p>Request Parameters: messageId, messageContent, patientId</p> <p>Response Parameters: If Successful, If Failed, Error Message</p> <p>Function: Doctors can reply to the patient when patient sends them message</p>
	Operation: deleteMessage
	<p>Request Parameters: messageId</p> <p>Response Parameters: If Successful, If Failed, Message Non-exist</p> <p>Function: Doctors can delete the message withing the server when the online conversation is officially ended.</p>
	Operation: receiveMessage
	<p>Request Parameters: patientId, messageId</p> <p>Response Parameters: If Successful, If Failed, Error Message</p> <p>Function: Doctors can receive the patients' messages once patients' sent the message</p>

Interface Name	Manage Prescription
Interface Provider	Health Guardian SaaS
	Operation: generatePrescription
	<p>Request Parameters: patientId, doctorId, createTime, description</p> <p>Response Parameters: If Successful, If Failed, Error Prescription Message, Signature Missing</p> <p>Function: Doctors can generate electronic prescription for patients with signature and other contents within</p>
	Operation: viewPrescription
	<p>Request Parameters: prescriptionId</p> <p>Response Parameters: If Successful, If Failed, Error Prescription Message, Prescription Verification Failed</p> <p>Function: Doctors can check the prescription they generated and track the status</p>
	Operation: updatePrescription
	<p>Request Parameters: doctorId, prescriptionId, description</p> <p>Response Parameters: If Successful, If Failed, Error Message when updating</p> <p>Function: Doctors can update the prescription of the earlier generated prescription either to correct information or keep the information up to date</p>
	Operation: checkSignature
	<p>Request Parameters: Signature</p> <p>Response Parameters: Missing Signature, Error in Signature, Mismatch in Signature</p>

Task 3: Software Testing

(a) Unit Test Plan

1. Class

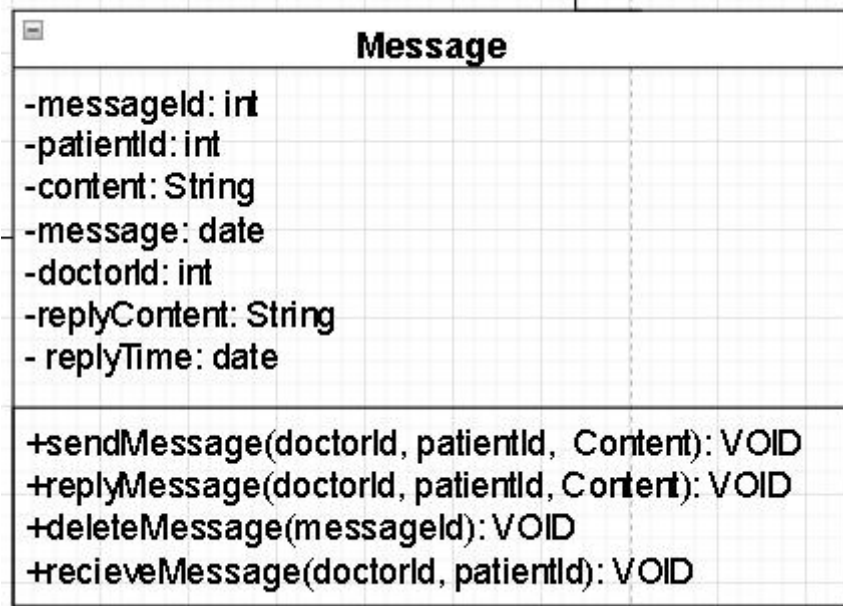


Figure 1: Message Class

Test Cases	Method	Expected Output
Send Message Successfully	sendMessage(12, This is the message, 100)	Message successfully sent, display the sent message
Send Message Failed	sendMessage(12, This is the message, 100)	Error Message for failing in sending patient message
Reply Message Success	ReplyMessage(12, 100, This is the reply Message)	Display the replied message. Successfully replied
Reply Message Failed	ReplyMessage(12, 100, This is the message)	Error message of failure in replying message to patient
Delete Message Success	deleteMessage(1)	Successfully Deleted
Delete Message Failed	deleteMessage(1)	Error Message for failing to delete message
Receive Message	receiveMessage(12,	Message Successfully

Success	100)	Displayed
Receive Message Failed	receiveMessage(12, 100)	Error message for failing to receive message

2. Interface

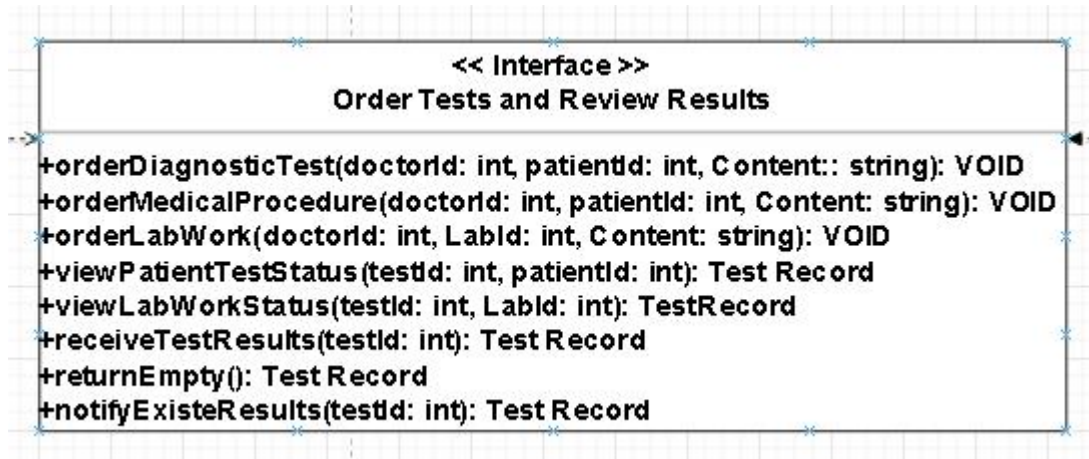


Figure1: Order Test and Review Results Interface

Test Cases	Methods and Parameters	Expected Output
Order Diagnostic Test Success	orderDiagnosticTest(12, 100, blood examination)	Order id, Order Successfully Requested Message
Order Diagnostic Test failed	orderDiagnosticTest(12, 100, blood examination)	Failed Information: Patient notfound, test unavailable
Order Medical Procedure Success	orderMedicalProcedure(12, 100, MRI and Brain Surgery)	Order Id and Success Message
Order Medical Procedure Failed	orderMedicalProcedure(12, 100, MRI and Brain Surgery)	Error Message, Failed to request procedure for patients
Order Lab Work Success	orderLabWork(12, 10, Drug sample tests)	Request Successfully posted
Order Lab Work Failed	orderLabWork(12, 10, Drug sample tests)	Error Message for ordering the test
Receive Test Results Success	receiveTestResults(90)	Successfully received
Receive Test Results Failed	receiveTestResults(90)	Error message of receiving the tests results
View Patient Tests Status Success	viewPatientTestStatus(90, 100)	Successfully displayed the status
View Patient Tests	viewPatientTestStatus(90,	Error message for

Status Failed	100)	displaying Test Requests Expired
View Lab Work Status Success	viewLabWorkStatus(120, 11)	Successfully displayed the status
View Lab Work Failed	viewLabWorkStatus(120, 11)	Error Messages Lab Work Had Issues Work or Tests Requests Expired
Return Empty Success	returnEmpty()	Display that the results that doctor is asking for is empty.
Return Empty Failed	returnEmpty()	Display Error Message about not correctly display the empty notifications
Notify Exists Results Success	notifyExistsResults(testId)	Display the notification successfully, doctor receive the the message of existing a result
Notify Exists Results Failed	notifyExistsResults(testId)	Error Message for failing to notify

(b) System Test Plan

Use Case: Order Tests

Scenario: Successfully Ordered the test and receive the results and review the returned results.

Doctor	Health Guardian System
1.View the Reports/Results of Tests	
2.Does Not Have Test Report	
3.Order Tests	4.Receive the Medical Procedure Requests
	5.Make a judgement that this request it a diagnostic request for patient
	6.Inform the patient with the diagnostic test request of doctor
	7.Patient completed the diagnostic test and return the reports to the system
9. Notify the doctor the exists of	8.The system receive the report and

results	store the report in test record database
10. Request to retrieve the results	11. Receive the request
	12. Send the report results to doctor
13. Receive the report results	
14. Review the reports	

Test Data:

Input:

ID Information: Doctor Id, Patient Id, LabId

Test Information: Test Content, Type

Store:

On Doctor App: No data stored

On Cloud System: Test Id, Test Content, Doctor Id, Patient Id, LabId

Output:

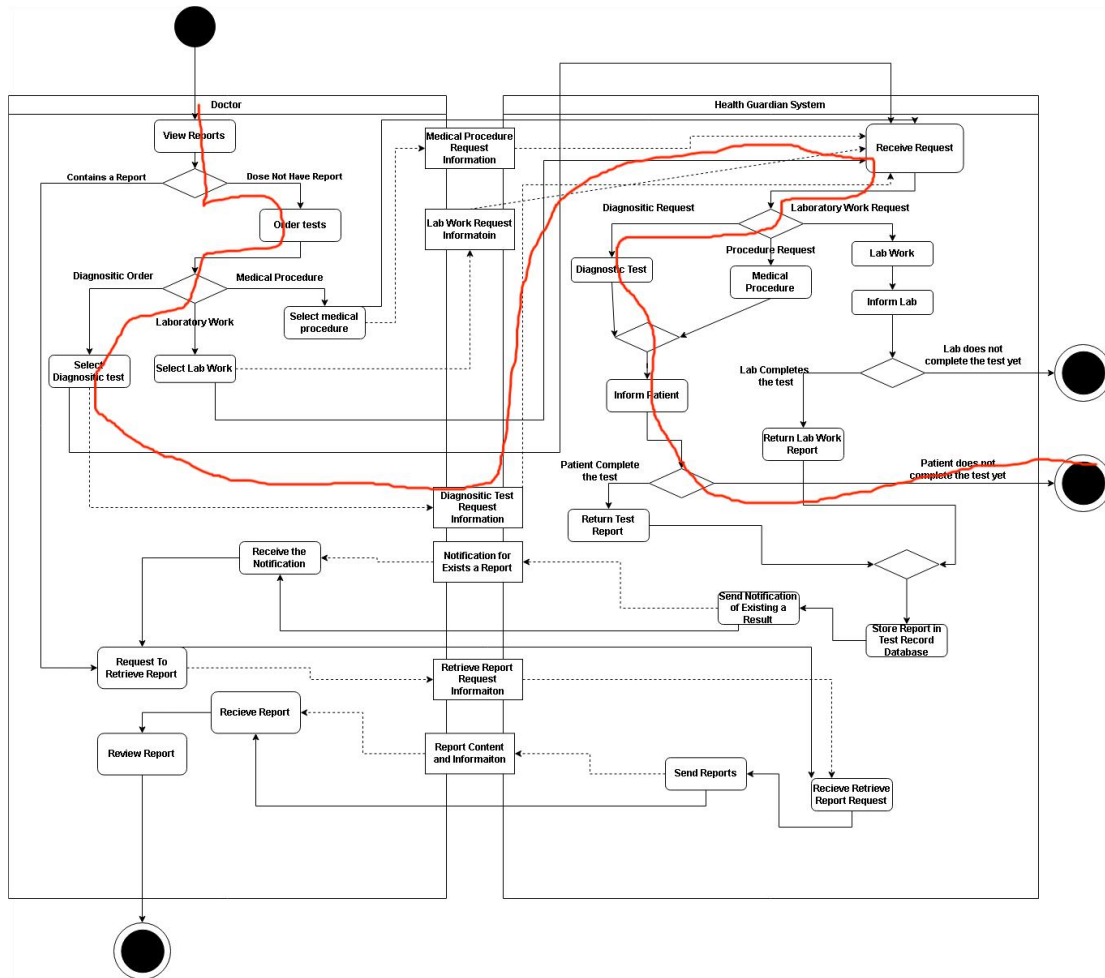
Test Id

Test Results

In this test, assume that a the ID of doctor is 12, patient ID is 100, lab Id is 6
Diagnosis content is MRI, medical procedure is Brain Surgery, Lab Work is
drug sample test. Test ID is 120

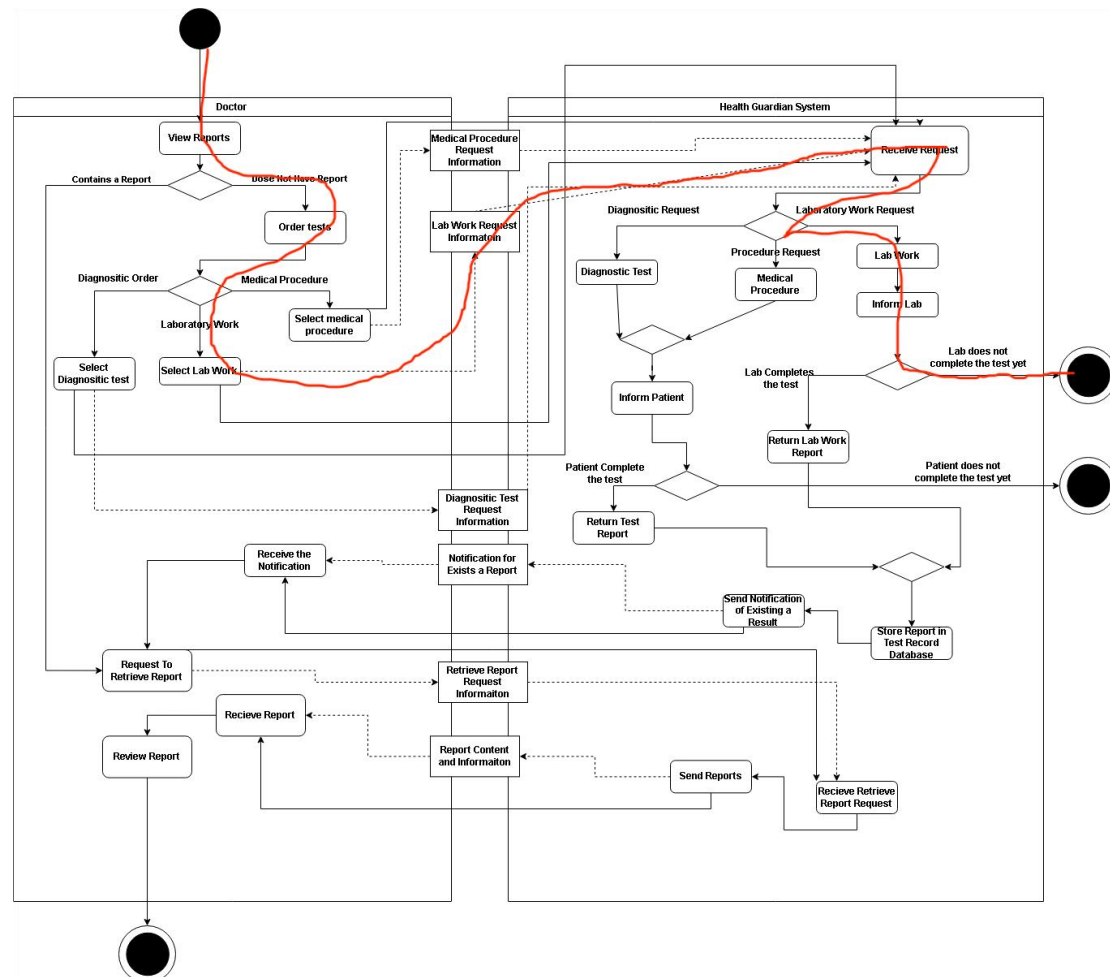
Test Progress:

1. Successfully requested the Medical Procedure and the system had
informed the patient but the results are not yet completed, only the test is
successfully posted



Doctor	Health Guardian System
1.View the Reports/Results of Tests	
2.Does Not Have Test Report	
3. Order Diagnostic Test (DoctorId: 12, patientId: 100, MRI)	4.Receive the diagnostic test Requests (DoctorId: 12, patientId: 100, MRI)
	5.Inform the Patient to complete diagnostic test(DoctorId: 12, patientId: 100, MRI)
	6.Request stays on patient's system until the request is done(DoctorId: 12, patientId: 100, MRI)

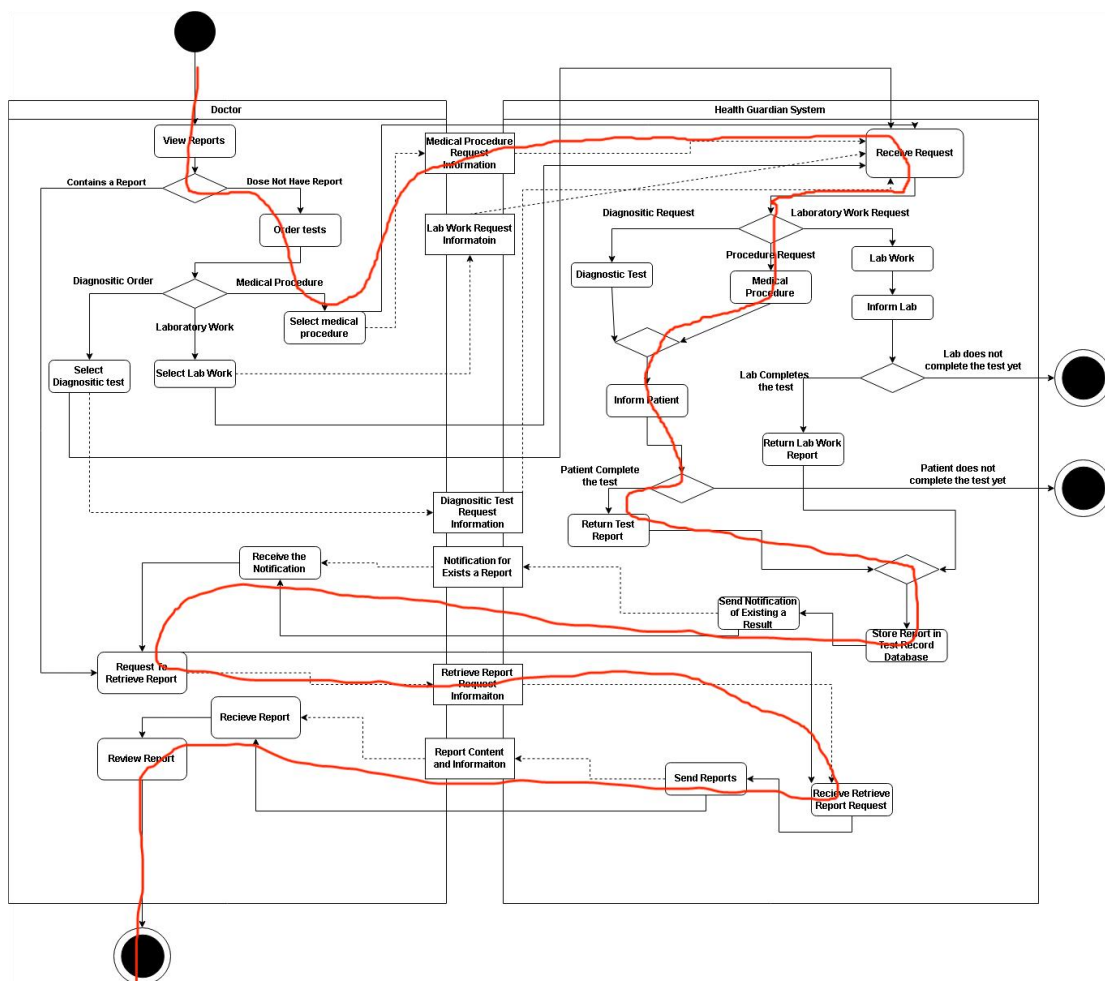
4. Successfully Order the Lab Work, however the work is not yet complete and test results are not returned, with only the test successfully posted and lab informed



Doctor	Health Guardian System
1. View the Reports/Results of Tests	
2. Does Not Have Test Report	
3. Order Lab Work (DoctorId: 12, LabId: 6, Drug Sample test)	4. Receive the test request of doctor (DoctorId: 12, LabId: 6, Drug Sample test)
	5. Judge the type as lab work (DoctorId: 12, LabId: 6, Drug Sample test)
	6. Inform the laboratory (DoctorId: 12, LabId: 6, Drug Sample test)
	7. Judge that the work is not yet

	complete (DoctorId: 12, LabId: 6, Drug Sample test)
	8. The tests or work requests stays active on laboratory's system until the work is finished and return report as results (DoctorId: 12, LabId: 6, Drug Sample test)

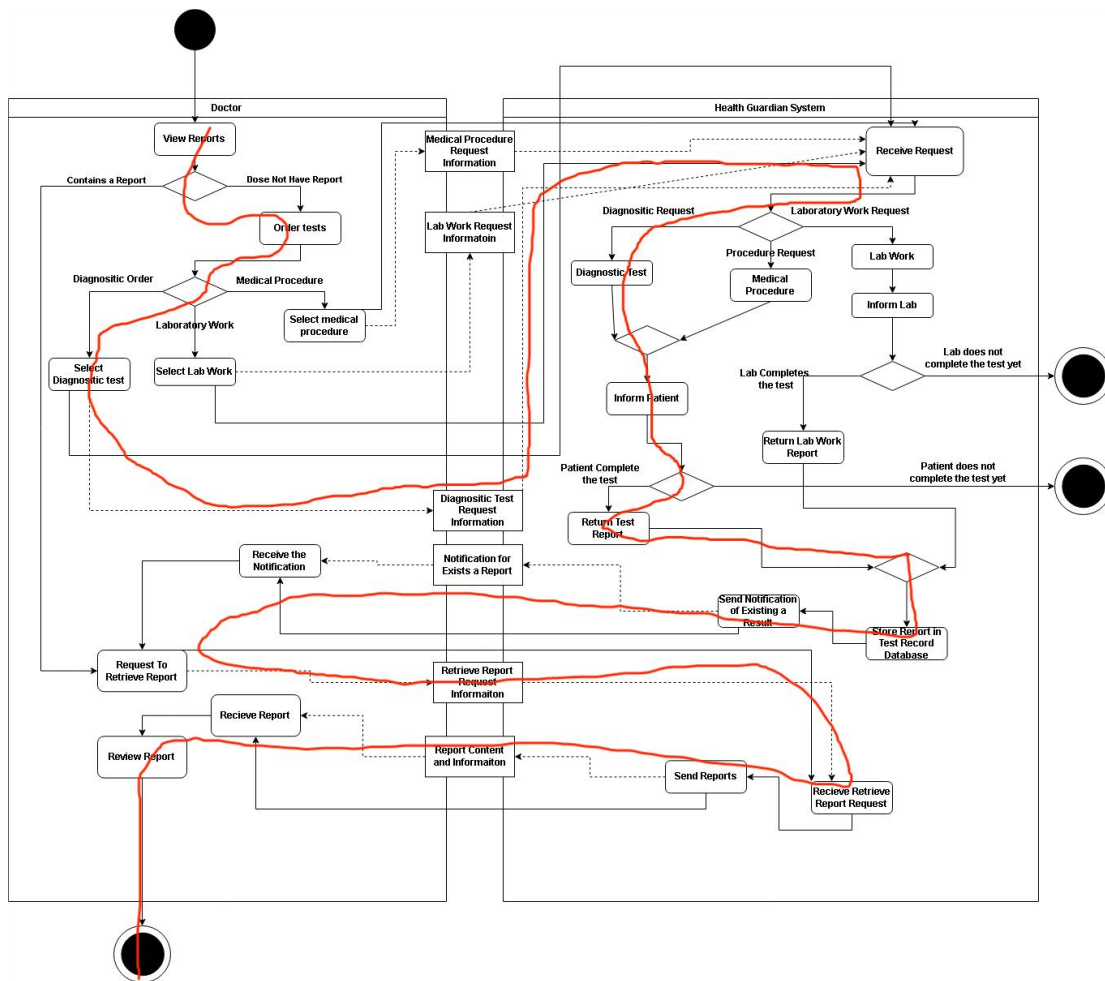
5. Medical Procedure is successfully ordered, patient is informed, after this the test is completed and results are store in the database, doctor retrieves the results and receive the reports to review.



Doctor	Health Guardian System
1.View the report	
2.No report to view	
3. Order a Medical Procedure (DoctorId: 12, patientId: 100, Brain Surgery)	4. Receive the requests (DoctorId: 12, patientId: 100, Brain Surgery)

	5. Judge the request to be medical procedure (DoctorId: 12, patientId: 100, Brain Surgery)
	6. Inform the patient to complete medical procedure (DoctorId: 12, patientId: 100, Brain Surgery)
	7. Results are return once the patient finished the medical procedure (DoctorId: 12, patientId: 100, Brain Surgery)
	8. Store the results in test record database (DoctorId: 12, patientId: 100, Brain Surgery)
9. Receive the notification (testId: 120)	9.Notify the doctor the exists of test results (testId: 120)
11. Request to retrieve the results (testId: 120)	12. Receive the request for retrieving results (testId: 120)
13. Receive the results (testId: 120, content: Brain Surgery Report)	14. Send the results back to doctor (testId: 120, content: Brain Surgery Report)
15. Review the results (testId: 120, content: Brain Surgery Report)	

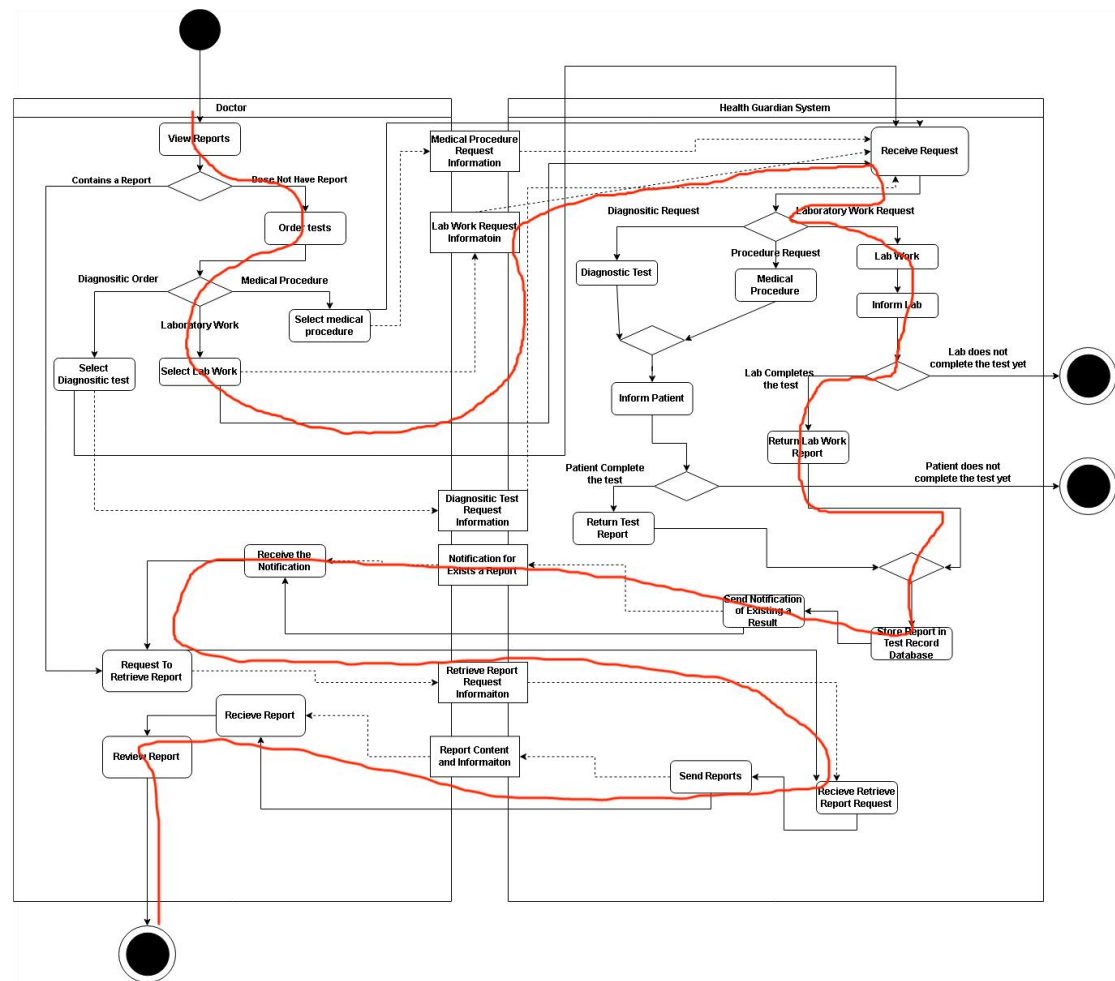
6. Doctor order the diagnostic test and after the system informed the patient ,patient completes the diagnosis, and return the results



Doctor	Health Guardian System
1.View the report	
2.No report to view	
3.Order a Diagnostic Test (DoctorId: 12, patientId: 100, MRI)	4.Receive the request (DoctorId: 12, patientId: 100, MRI)
	5.Judge the request to be diagnostic (DoctorId: 12, patientId: 100, MRI)
	6.Inform the patient to complete the diagnosis (DoctorId: 12, patientId: 100, MRI)
	7.Patient finished the diagnosis and return the results (DoctorId: 12, patientId: 100, MRI)
	8.Store the results in the test record database (DoctorId: 12, patientId: 100, MRI)
10.Receive the notification (testId: 120)	9.Notify the doctor about the exists of results (testId: 120)
11.Request to retrieve the results (testId: 120)	12. Receive the request (testId: 120)

14.Receive the results (testId: 120, Content:MRI Report)	13.Send the reports of results back to doctor (testId: 120, Content:MRI Report)
15.Review the results (testId: 120, Content:MRI Report)	

6.Doctor order the lab work, and work is successfully completed and return a results of work or tests.



1.View the report	
2.No report to view	
2. Order a Lab Work (DoctorId: 12, LabId: 6, Drug Sample test)	4. Receive the request (DoctorId: 12, LabId: 6, Drug Sample test)
	5.Judge the request to be lab work (DoctorId: 12, LabId: 6, Drug Sample test)
	6.Inform the laboratory to complete the work (DoctorId: 12, LabId: 6, Drug Sample test)

	7.Laboratory finished the diagnosis and return the results (DoctorId: 12, LabId: 6, Drug Sample test)
	8.Store the results in the test record database (DoctorId: 12, LabId: 6, Drug Sample test)
10.Receive the notification (testId: 120)	9.Notify the doctor about the exists of results (testId: 120)
11.Request to retrieve the results (testId: 120)	12. Receive the request (testId: 120)
14.Receive the results (testId: 120, Content: Drug Sample Test Report)	13.Send the reports of results back to doctor (testId: 120, Content: Drug Sample Test Report)
15.Review the results (testId: 120, Content: Drug Sample Test Report)	

```

graph TD
    Start(( )) --> Doctor
    subgraph Doctor
        Doctor --> ViewReports[View Reports]
        ViewReports --> ContainsReport{Contains a Report}
        ContainsReport -- "Dose Not Have Report" --> OrderTests[Order tests]
        OrderTests --> MedicalProcedure{Medical Procedure}
        MedicalProcedure --> SelectLabWork[Select Lab Work]
        MedicalProcedure --> SelectMedicalProcedure[Select medical procedure]
        SelectLabWork --> SelectDiagnosticTest[Select Diagnostic test]
    end
    subgraph HealthGuardianSystem[Health Guardian System]
        ReceiveRequest[Receive Request]
        DiagnosticRequest{Diagnostic Request}
        LaboratoryWorkRequest{Laboratory Work Request}
        DiagnosticTest[Diagnostic Test]
        ProcedureRequest{Procedure Request}
        LabWork[Lab Work]
        InformLab[Inform Lab]
        InformPatient[Inform Patient]
        ReturnLabWorkReport[Return Lab Work Report]
        ReturnTestReport[Return Test Report]
        StoreReport[Store Report in Test Record Database]
        SendNotification[Send Notification of Existing a Result]
        ReceiveRetrieveReport[Receive Retrieve Report Request]
    end
    subgraph Lab
        LabWork --> InformLab
        InformLab --> LabCompletesTest{Lab Completes the test}
        LabCompletesTest -- "Lab does not complete the test yet" --> EndLab(( ))
        LabCompletesTest --> ReturnLabWorkReport
    end
    subgraph Patient
        InformPatient --> PatientCompletesTest{Patient Complete the test}
        PatientCompletesTest -- "Patient does not complete the test yet" --> EndPatient(( ))
        PatientCompletesTest --> ReturnTestReport
    end
    ContainsReport --> RequestRetrieveReport[Request to Retrieve Report]
    RequestRetrieveReport --> RetrieveReportInfo[Retrieve Report Request Information]
    RetrieveReportInfo --> RecieveReport[Recieve Report]
    RecieveReport --> ReviewReport[Review Report]
    ReviewReport --> End(( ))
    OrderTests --> MedicalProcedureRequest[Medical Procedure Request Information]
    SelectMedicalProcedure --> MedicalProcedureRequest
    SelectLabWork --> LabWorkRequest[Lab Work Request Information]
    SelectDiagnosticTest --> DiagnosticTestRequest[Diagnostic Test Request Information]
    MedicalProcedureRequest --> ReceiveRequest
    LabWorkRequest --> ReceiveRequest
    DiagnosticTestRequest --> ReceiveRequest
    ReceiveRequest --> DiagnosticRequest
    ReceiveRequest --> LaboratoryWorkRequest
    DiagnosticRequest --> DiagnosticTest
    LaboratoryWorkRequest --> LabWork
    DiagnosticTest --> InformPatient
    LabWork --> InformLab
    InformPatient --> PatientCompletesTest
    InformLab --> LabCompletesTest
    PatientCompletesTest --> ReturnTestReport
    LabCompletesTest --> ReturnLabWorkReport
    ReturnTestReport --> SendNotification
    ReturnLabWorkReport --> SendNotification
    SendNotification --> NotificationExists{Notification for Exists a Report}
    NotificationExists --> RequestRetrieveReport
    NotificationExists --> RecieveReport
    NotificationExists --> ReceiveRetrieveReport
    ReceiveRetrieveReport --> ReportContent[Report Content and Information]
    ReportContent --> SendReports[Send Reports]
    SendReports --> ReceiveRetrieveReport
    ReceiveRetrieveReport --> StoreReport
    StoreReport --> EndLab
    StoreReport --> EndPatient
    
```

Doctor	Health Guardian System
1. View the report	
2. Report exists	
3. Retrieve the report (TestId)	4. Receive the request for retrieving reports (TestId)
6. Receive the tests reports (TestId, Content)	5. Send the reports results to doctor (TestId, Content)
7. Review the results (TestId, Content)	