Chapter 1

Introduction

1.1 Abstract

Smart Health Care is an android application which sets up online communication between a doctor and a patient. This app is helpful to patients to ask questions and state their concerns to doctors regarding their health condition. This app will facilitate the patients to interact with doctors without making any physical appointments. In addition using this app the patient can make an appointment to meet the doctor in clinic/hospital. Similarly this app is also beneficial to doctors by providing the following functionalities: patient interaction through video calling confirming appointments information sharing with other doctors. Unlike other similar kinds of systems available SHC has unique features such as issuing online prescription to patients sending health tips to patients and effectively reducing the cost of customer service and providing a vital communication link between doctors and patients. best suitable deal for him/her.

1.2 Existing system

Following is the existing system which relates to our system.

1.2.1 Lybrate

Lybrate is a healthcare company that developed an online platform to connect doctors and patients. The company was founded in 2013 and is headquartered in New Delhi, India. The service allows patients to connect with the doctor online through a video call or schedule an appointment and can get info about medication.

Strong Point	Weak Point
1- Large Community of Available Doctors	1- Not allow the user to make his/her own schedule based on his/her availability.
2- Continuous Updates in their system	2- Difficult User-interface.
3- Uses Artificial Intelligence	3- Require too many steps for completing a task.

Table 2: Lybrate Weak and Strong Points

1.2.2 Marham

Lybrate is a healthcare company that developed an online platform to connect doctors and patients. The company was founded in 2015 and is headquartered in Lahore, Pakistan. The service allows patients to connect with the doctor online through a video call or schedule an appointment and can get info about medication.

Strong Point	Weak Point
Large Community of Available Doctors	Update very slow
Easy to use	Video call feature is a weak point
Good Customer Support Service	Poor Payment method with easypaisa as it get loading again and again

Table 3: Marham Weak and Strong Points

1.2.3 Oladoc

Oladoc is a healthcare company that developed an online platform to connect doctors and patients. The company was founded in 2016 and is headquartered in Lahore, Pakistan. The service allows patients to connect with the doctor online through a video call or schedule an appointment and can get info about medication.

Strong Point	Weak Point
Affordable charges	Don’t publish negative reviews posted by customers
User Friendly Interface	Poor Customer Support Service

Table 4: Oladoc Weak and Strong Points

1.3 Problem Statement

Due to rise in health problems in today's world. The increasing stressful lifestyle is taking maximum toll on the public health. With the ever-increasing queues at hospitals and ever-increasing number of patients the doctor fees have sky-rocketed which is affecting especially those patients who cannot afford the fee or who are not suffering from major ailments. Now-a-days everyone’s life is smart due to the use internet. Our problem is to develop an android application that provides all the necessary information related to health issues as well as provide a doctor that can help patients to cure them.

1.4 Proposed Solution

The System will basically consist of three different users that would be interacting with the aim of providing better healthcare service through mutual utilization of self-monitoring and the consultation from a specialist. The patient will consult doctor any time by siting at their home. The mutual interaction exists because user will have the option of sending the data for analysis to the doctor getting the feedback and then acting on his advice. Patient and Doctor will be in touch through video call as well. Any Patient can ask question openly and any doctor will answer them according to diseases.

1.5 Project Scope

- The application provides functions and features for a patient to get help from doctors.
- We have three user in it which are Patient, Admin and Doctor.
- Patient can ask any health issue, take appointment, upload lab test reports and also get health product.
- The concerned doctor/admin will update the information about their packages/services and this information will be presented to the patient with an option of choosing best suitable deal for him/her.

.

Chapter 2

System Analysis

The preliminary step of SDLC is analysis. Requirements are the capabilities that are desired to be conveyed by the system (framework). We can't have a successful program that does not fulfil whatever the user required. Requirement analysis is the procedure of demonstrating finding and refinement. It might be capabilities required by the user or those limitations which are needed by the designer. For showing capabilities in better points of interest a requirement can be divided in several sub-requirements. Then we discover every requirement needed by the system that is the point where we have the complete sketch of the required system. Today's life has been progressed when we compare it with past. These days we are moving towards computerization in relatively every part of our lives we are making a group of more issues when taking care of a few issues.

2.1 Requirement Analysis

Requirement analysis software engineering and system engineering encloses those problems that effects in resolving the conditions or requirements that should be achieved for another or adjusted picture assessing the perhaps conflicting requirements of the different partners for example clients or recipients. It is a starting time in the broader action of requirements engineering which encloses all exercises worried about archiving approving inspiring dissecting and overseeing system and software requirements. Requirements analysis is important in the successful pursuit of systems or software venture. The requirement ought to be testable traceable quantifiable noteworthy recorded identified with distinguished business needs or openings and characterized to the level of subtle elements enough for system outline.

2.2 Use-Case Model

A use-case is a chart of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. That actor should be a human external system or any other outer body which can impact on system. In systems engineering use cases are used at a higher level than within software engineering often representing missions or stakeholder goals. The detailed requirements may then be captured in the Systems Modeling Language (SML) or as contractual statements.

- **Use case**

  A use-case is a flow chart of activities or actions that operated in an order while characterizing the flow of information between an actor and the system to accomplish an objective.

- **Actor**

  A use-case differentiates the flow of information between outside bodies and the system under though to achieve a required functionality. On-screen characters must have the ability to decide whether a human is required or not: “Action performing artist can be a man organization or association PC-program equipment or any other software or both.”

2.2.1 The Basic Method to Identify Use Cases

- **Identify System Boundary**

  Anything beyond the system boundary is not considered as part of system. Smart Health Care App which facilitates a Patient by providing experienced doctor on one click.

- **Identify Actor**

  A user or any other system which interacts with the system.

  - **Patient**

    He is the primary actor of the system. He uses recommended system.

  - **Doctor**

    He is the primary actor of the system. He offer his services to patients.

  - **Admin**

    He is the primary actor of the system. Admin will manage patient doctor labs and store.

2.2.2 Use Case Diagrams

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases.

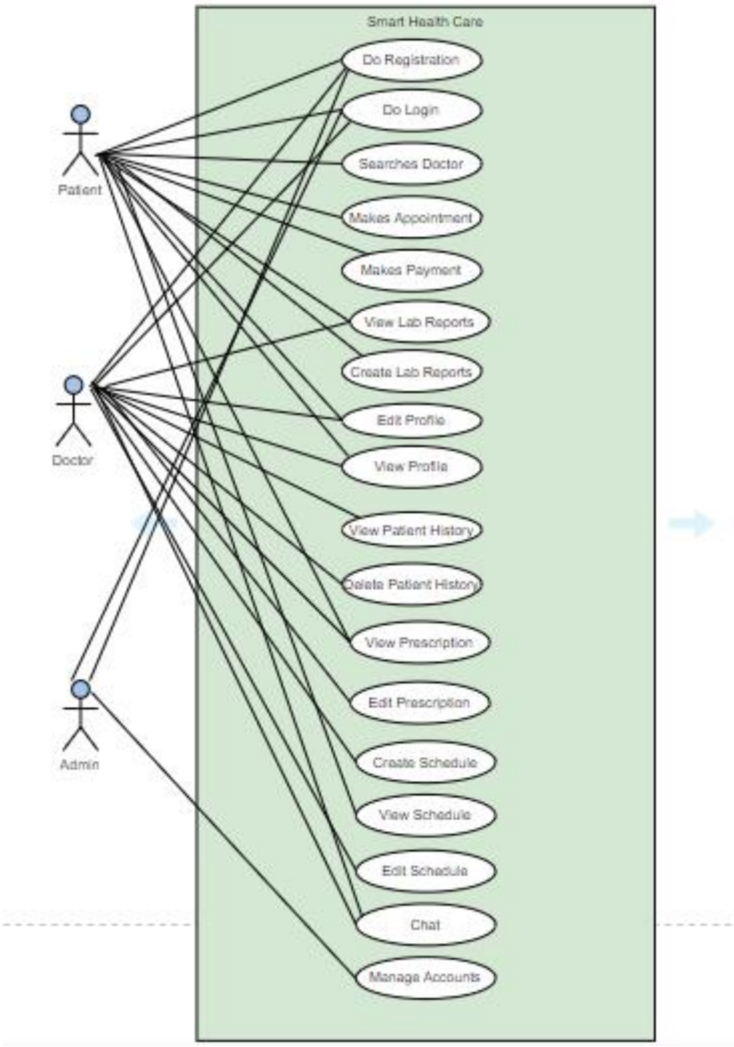

Fig. 1- Use Case Diagram

2.3 Use Cases

Use case description in brief format:

2.3.1 Do Registration

Use case ID:	1
Actor:	Patient/ Doctor
Type:	Primary
Description	They must register itself by providing his Signup details as required by the system.

Table 5: Show use case for register

2.3.2 Do Login

Use case ID:	2
Actor:	Patient/ Doctor/Admin
Type:	Primary
Description	They have to login itself with its pre-registered login details.

Table 6: Show use case for login

2.3.3 Searches Doctor

Use case ID:	3
Actor:	Patient
Type:	Primary
Description	Patient can search for a doctor of their own choice.

Table 7: Show use case for search doctor

2.3.4 Makes Appointment

Use case ID:	4
Actor:	Patient
Type:	Primary
Description	They can book appointment with desired doctor.

Table 8: Show use case for make appointment

2.3.5 Makes Payment

Use case ID:	5
Actor:	Patient
Type:	Primary
Description	They can pay to doctor online.

Table 9: Show use case for makes payment

2.3.6 View Lab Report

Use case ID:	6
Actor:	Doctor
Type:	Primary
Description	Doctor can view the lab reports that shared by patient

Table 10: Show use case for view lab reports

2.3.7 Create Lab Report

Use case ID:	7
Actor:	Patient
Type:	Primary
Description	Patient will attach the report to show to doctor

Table 11: Show use case for create lab report

2.3.8 Edits profiles

Use case ID:	8
Actor:	Patient/Doctor
Type:	Primary
Description	They can edit their profiles.

Table 12: Show use case for edit profile

2.3.9 View profiles

Use case ID:	9
Actor:	Patient/Doctor
Type:	Primary
Description	They can view their profiles.

Table 13: Show use case for view Profile

2.3.10 View Patient History

Use case ID:	10
Actor:	Doctor
Type:	Primary
Description	Doctor can view patient’s all previous records

Table 14: Show use case for view patient history

2.3.11 Delete Patient History

Use case ID:	11
Actor:	Doctor
Type:	Primary
Description	Doctor can delete patient’s all previous records

Table 15: Show use case for delete patient history

2.3.12 View Prescription

Use case ID:	12
Actor:	Doctor/Patient
Type:	Primary
Description	Doctor and patient both can view prescription

Table 16: Show use case for view prescription

2.3.13 Edit Prescription

Use case ID:	13
Actor:	Doctor
Type:	Primary
Description	Doctor can edit prescription

Table 17: Show use case for edit prescription

2.3.14 Create Schedule

Use case ID:	14
Actor:	Doctor
Type:	Primary
Description	Doctor will create a schedule for his/her availability

Table 18: Show use case for create schedule

2.3.15 View Schedule

Use case ID:	15
Actor:	Patient
Type:	Primary
Description	Patient can view schedule of doctor for consultation

Table 19: Show use case for view schedule

2.3.16 Edit Schedule

Use case ID:	16
Actor:	Doctor
Type:	Primary
Description	Doctor can edit his/her schedule according to availability

Table 20: Show use case for edit schedule

2.3.17 Chat

Use case ID:	17
Actor:	Doctor/Patient
Type:	Primary
Description	Both doctor and patient can chat in chat box

Table 21: Show use case for chat

2.3.18 Manage Accounts

Use case ID:	18
Actor:	Admin
Type:	Primary
Description	Admin can manage all accounts

Table 22: Show use case for manage accounts

2.4 Use case description in detailed expended format

An expanded form of UML is giving point to point detail of every use-case of the system. An extended use-case consist of all the abnormalities that can happen and their solution. There are two areas to the abnormal state use case which are a heading and a body. The heading gives the name performers depiction kind of use case and that's just the beginning.

- Title of the activity.
- The actors involved with the initiator defined.
- The description of the activity from the high-level use case.

2.4.1 Do Registration

Use case ID:	1
Actor:	Patient/Doctor
Purpose	To register for the system
Overview	They will be prompted to enter required details.
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They should must have started the application and choose Signup option.
Typical course of Events	
Actor Actions	**System Response**
1) They will enter all the required data in correct format.	They are successfully added to the system database.
Post-Condition	Application allow the user to access all the functionalities.
Alternative course of Events	
Actor Actions	**System Response**
1) They inputs incorrect data or wrong entry	Tells the users to enter correct data according to the indicated validations.

Table 23: Show expended use case for registration

2.4.2 Do Login

Use case ID:	2
Actor:	Patient/Doctor
Purpose	To Login for the system
Overview	They will input required data for Login.
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be registered in the system.
Typical course of Events	
Actor Actions	**System Response**
1) They enters email address and password	System authenticated them.
Post-Condition	Application is started giving all the functionalities to them.
Alternative course of Events	
Actor Actions	**System Response**
1.1) They are not registered	Tells them to first register themselves.
1.2) They enters invalid data in login form.	Tells the user to enter correct data which they provided at registration time.

Table 24: Show expended use case for login

2.4.3 Searches Doctor

Use case ID:	3
Actor:	Patient
Purpose	To book appointment or make call
Overview	They will add bookings or call
Type:	Primary
Cross Reference:	None
Pre-Conditions:	He must be logged in to the system and must have selected the search option.
Typical course of Events	
Actor Actions	**System Response**
1) Enter search data to make call or bookings that is required.	Present the doctor’s profile according to search query.
Post-Condition	Profile of doctor is shown to patient successfully.
Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error
1.2) Query specs didn’t match	Tells the patient that doctor is not found of that category.

Table 25: Show expended use case for search doctor

2.4.4 Makes Appointments

Use case ID:	4
Actor:	Patient
Purpose	To get the appointment from doctor
Overview	Book the appointment from desired doctor for checkup
Type:	Primary
Cross Reference:	None
Pre-Conditions:	Patient must be logged into the system.
Typical course of Events	
Actor Actions	**System Response**
1) They search for a doctor.	Displays the doctors profiles according to search query
2) select the appointment schedule.	
Post-Condition	Appointment must be save into the database with doctor's and patient's ids.

Alternative course of Events	
Actor Actions	**System Response**
1.1) Doctor have the appointment on that schedule already.	Prompt the patient to re-schedule

Table 26: Show expended use case for makes appointment

2.4.5 Makes Payment

Use case ID:	5
Actor:	Patient
Purpose	Payment of dues.
Overview	Patient can pay the fee to the Doctor
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must have booked appointments
Typical course of Events	
Actor Actions	**System Response**
1) Patient will choose the doctor	Presents the dues to the patient.
2) Gives his/her details and proceed to payment.	
Post-Condition	Payment will be transferred to relevant person successfully
Alternative course of Events	
Actor Actions	**System Response**
1.1) Enters invalid details	Gives the error and prompts to enter correct details.

Table 27: Show expended use case for makes payment

2.4.6 View Lab Reports

Use case ID:	6
Actor:	Patient/Doctor
Purpose	For further consultation
Overview	Doctor can view the reports sent by patient
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must have booked appointments
Typical course of Events	
Actor Actions	**System Response**
1) Doctor suggests some test to patient	
2) Patient shows reports to doctor	Doctor give his views on reports in chat
Post-Condition	Reports will be viewed by doctor
Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 28: Show expended use case for view lab reports

2.4.7 Create Lab Reports

Use case ID:	7
Actor:	Patient
Purpose	To shows test reports to doctor suggested by doctor
Overview	Patient will upload lab test report so that doctor can view that
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must have booked appointments
Typical course of Events	
Actor Actions	**System Response**
1) Doctor suggests some test to patient	
2) Patient will upload report document	Doctor give his views on reports in chat
Post-Condition	Reports sent successfully by patient
Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 29: Show expended use case for create lab reports

2.4.8 Edits profile

Use case ID:	8
Actor:	Patient/Doctor
Purpose	To edit their profiles
Overview	Edit profiles details.
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) They login the system and selects profile	Verifies account and displays profiles
2) They updates profiles	
Post-Condition	Profiles must be updated

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

.

Table 30: Show expended use case for edit profile

2.4.9 View profile

Use case ID:	9
Actor:	Patient/Doctor
Purpose	To view their profiles
Overview	View profiles details.
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) They login the system and selects profile	Verifies account and displays profiles
2) They view profiles	
Post-Condition	Profiles must be viewed

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 31: Show expended use case for view profile

2.4.10 View Patient History

Use case ID:	10
Actor:	Doctor
Purpose	To view history of patient
Overview	View complete history including reports and chats
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) He/ She login the system and selects patient	Verifies account and displays patient
2) Doctor view patient history	
Post-Condition	Complete history displayed

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 32: Show expended use case for view patient history

2.4.11 Delete Patient History

Use case ID:	11
Actor:	Doctor
Purpose	To delete history of specific patient
Overview	Doctor can delete his patient history
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) He/ She login the system and selects patient	Verifies account and displays patient
2) Doctor will delete patient history	
Post-Condition	Patient history deleted successfully

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 33: Show expended use case for delete patient history

2.4.12 View Prescription

Use case ID:	12
Actor:	Doctor / Patient
Purpose	To view the prescription given by doctor
Overview	Both can view prescriptions
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Patient book an appointment	Appointment booked successfully
2) After consultation, doctor give prescription	
Post-Condition	Prescription viewed by doctor and patient

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 34: Show expended use case for view prescription

2.4.13 Edit Prescription

Use case ID:	13
Actor:	Doctor
Purpose	To edit the prescription
Overview	Doctor can edit prescription according to condition of patient
Type:	Primary
Cross Reference:	None
Pre-Conditions:	They must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Patient book an appointment	Appointment booked successfully
2) After consultation, doctor can edit prescription	
Post-Condition	Prescription edited by doctor

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 35: Show expended use case for edit prescription

2.4.14 Create Schedule

Use case ID:	14
Actor:	Doctor
Purpose	To create schedule for availability
Overview	Doctor can create schedule for availability so that he/she can consult patients at that time
Type:	Primary
Cross Reference:	None
Pre-Conditions:	Doctor must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Doctor login to app	Logged in successfully
2) After login, doctor create schedule	
Post-Condition	Schedule created successfully

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 36: Show expended use case for create schedule

2.4.15 View Schedule

Use case ID:	15
Actor:	Patient
Purpose	To view schedule for checkup
Overview	Patient can view schedule of doctor so that they can consult at that time
Type:	Primary
Cross Reference:	None
Pre-Conditions:	Patient must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Patient login to app	Logged in successfully
2) Search doctor and view schedule	
Post-Condition	Schedule displayed

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 37: Show expended use case for view schedule

2.4.16 Edit Schedule

Use case ID:	16
Actor:	Doctor
Purpose	To edit schedule
Overview	Doctor can edit schedule if he/she wants
Type:	Primary
Cross Reference:	None
Pre-Conditions:	Doctor must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Doctor login to app	Logged in successfully
2) Edit schedule	
Post-Condition	Schedule edited successfully

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 38: Show expended use case for edit schedule

2.4.17 Chat

Use case ID:	17
Actor:	Doctor / Patient
Purpose	For consultation
Overview	Both doctor and patient can talk with each other
Type:	Primary
Cross Reference:	None
Pre-Conditions:	Both must be authorized on the system.
Typical course of Events	
Actor Actions	**System Response**
1) Patient book appointment	Appointment booked successfully
2) Start Chatting	
Post-Condition	Messages transfer successfully

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 39: Show expended use case for chat

2.4.18 Manage Accounts

Use case ID:	18
Actor:	Admin
Purpose	For managing accounts
Overview	Doctor He can add or delete any Patient/ Doctor
Type:	Primary
Cross Reference:	None
Pre-Conditions:	He must be logged in to the system
Typical course of Events	
Actor Actions	**System Response**
1) Manages user's accounts i.e. activate and deactivate	Appointment booked successfully
Post-Condition	Successfully manage accounts

Alternative course of Events	
Actor Actions	**System Response**
1.1) No internet connection	Mobile Application will not be loaded and it will show no internet connection error

Table 40: Show expended use case for accounts

2.5 Sequence Diagrams

2.5.1 Do registration

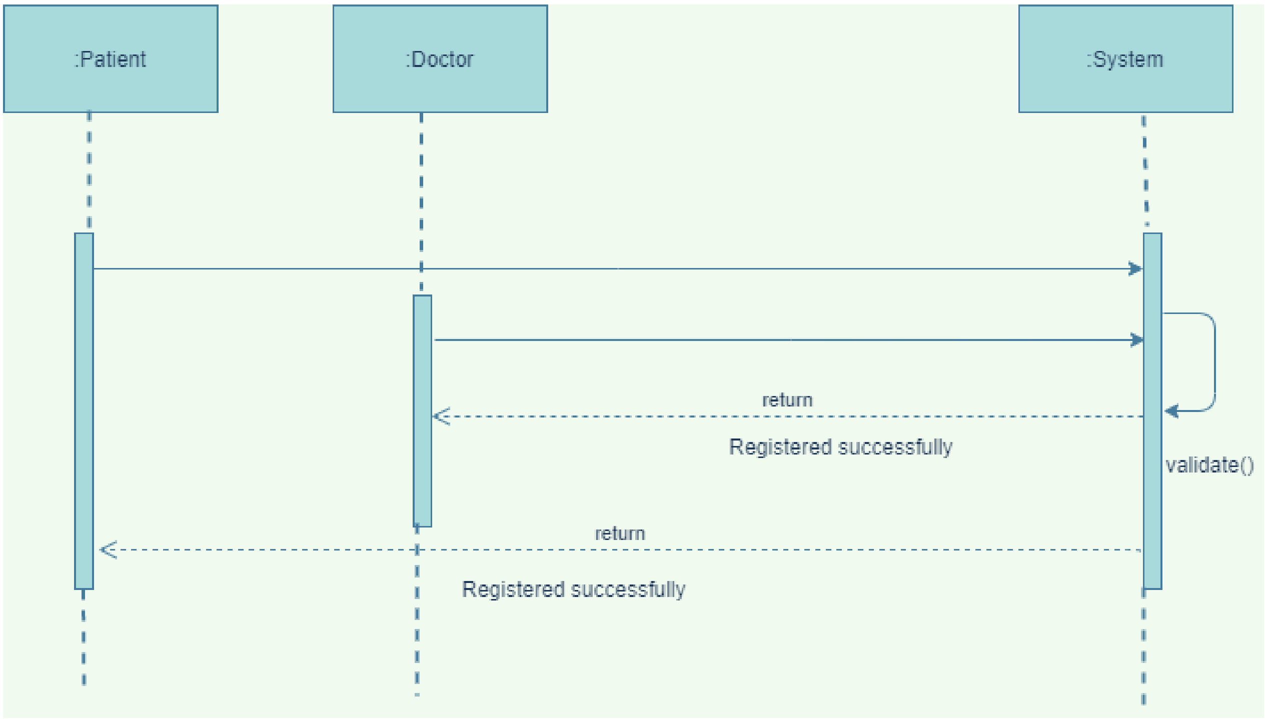

Fig 2: Show sequence diagram for registration

2.5.2 Do Logins

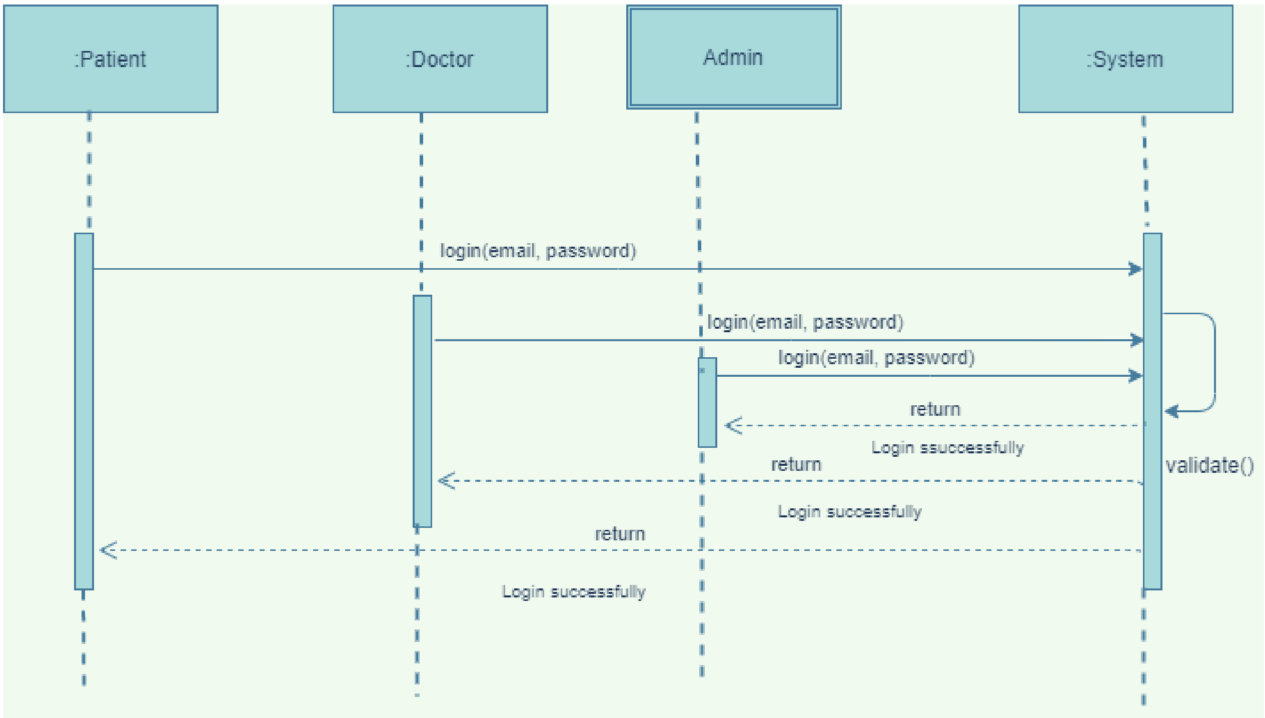

Fig 3: Show sequence diagram for login

2.6 System Sequence Diagrams

2.6.1 Searches Doctor

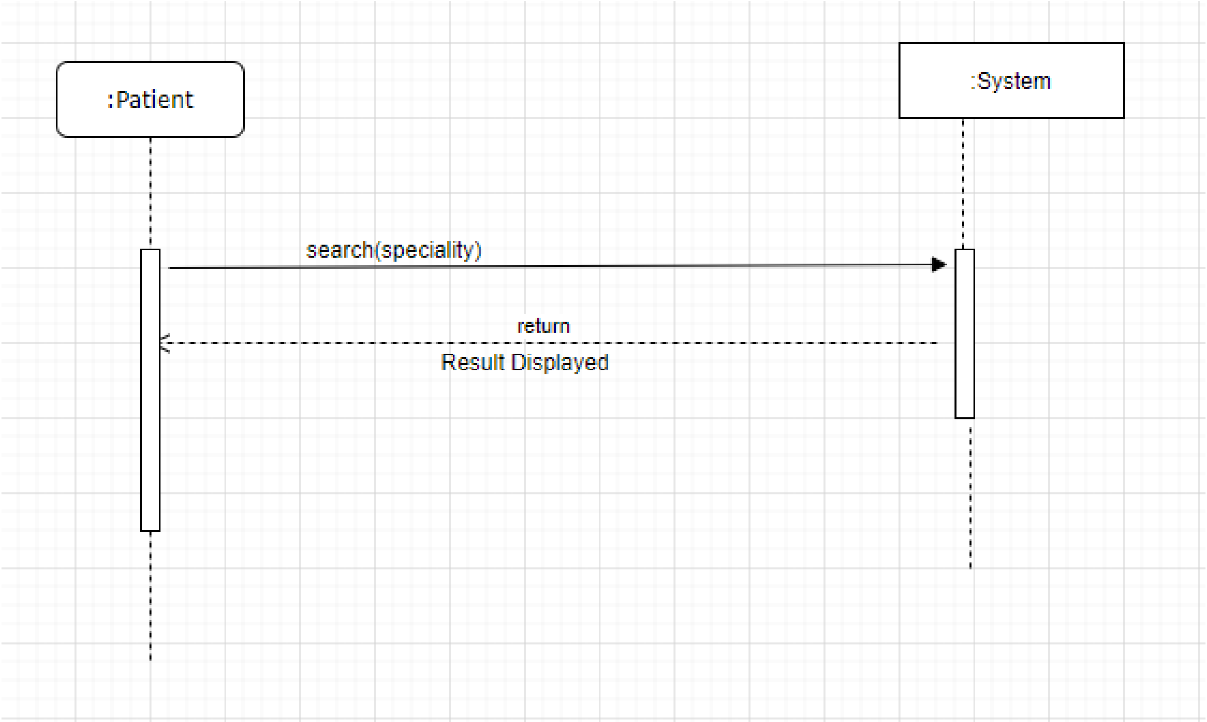

Fig 4: Show system sequence diagram for search doctor

2.6.2 Make Appointments

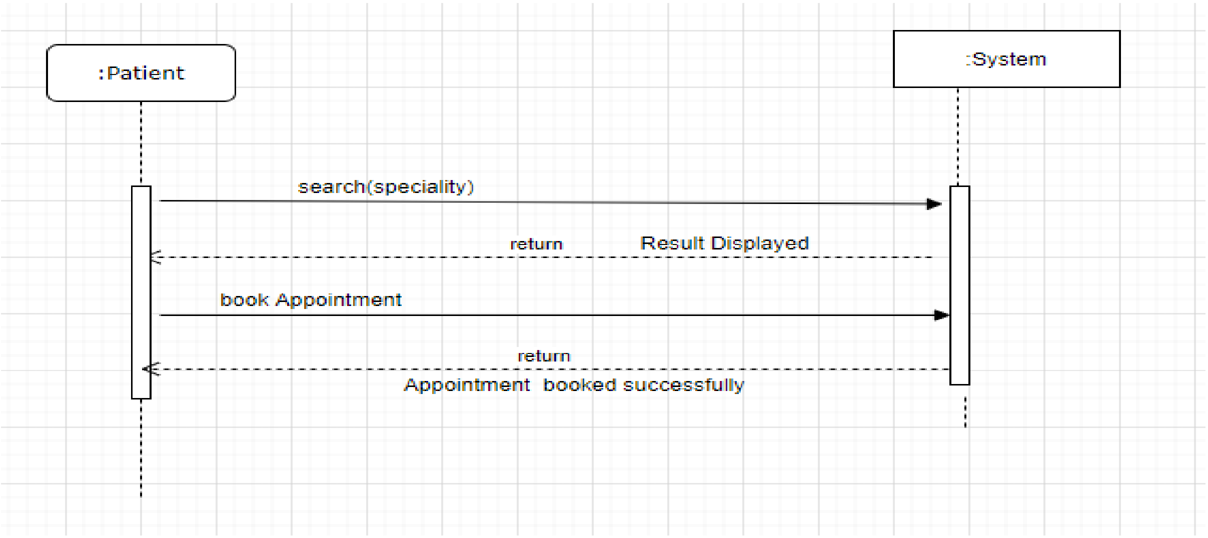

Fig 5: Show system sequence diagram for make appointment

2.6.3 Makes Payment

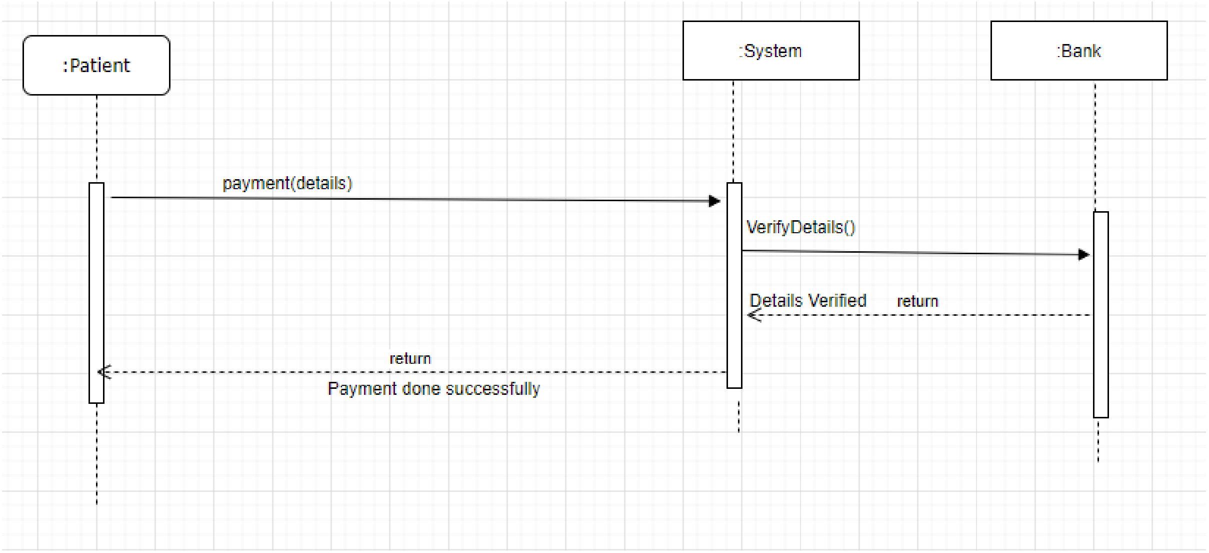

Fig 6: Show system sequence diagram for make payment

2.6.4 Create Lab Report

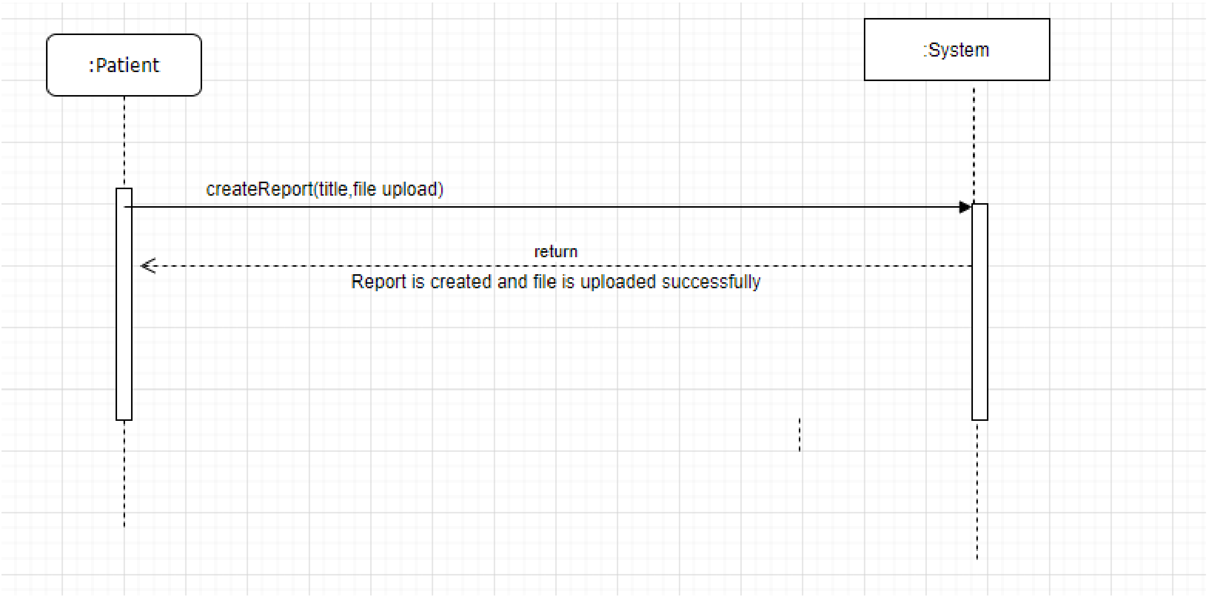
:Patient
:System
createReport(title,file upload)
return
Report is created and file is uploaded successfully

Fig 7: Show system sequence diagram for creating lab reports

2.6.5 View Prescription

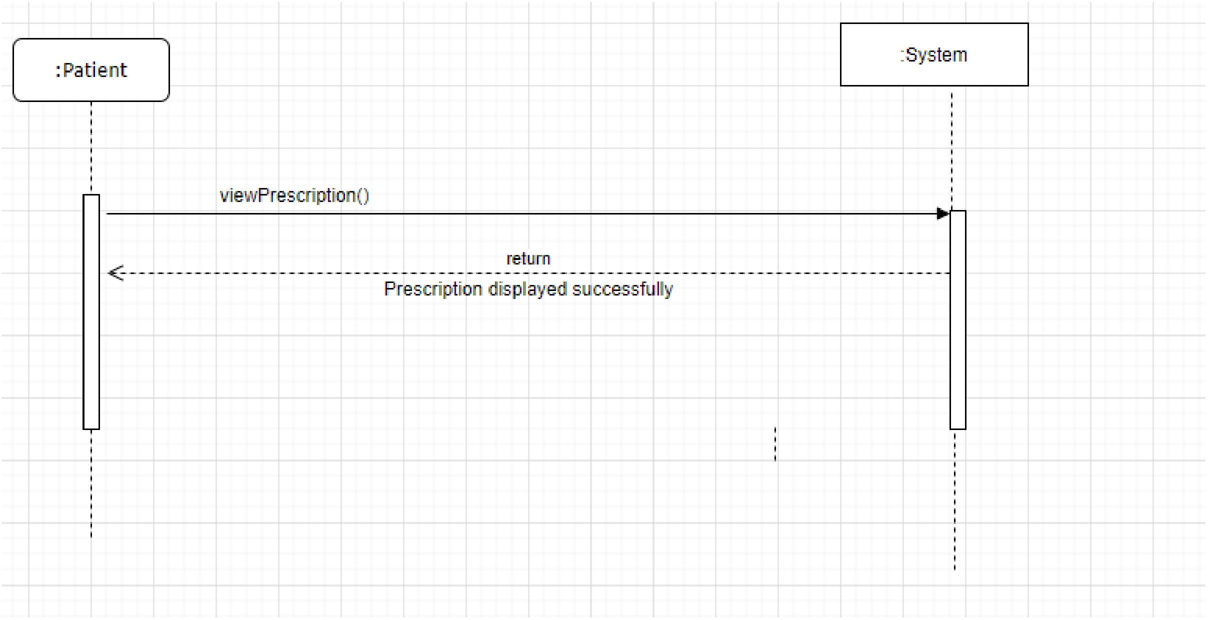

Fig 8: Show sequence diagram for view prescription

2.6.6 Edit Prescription

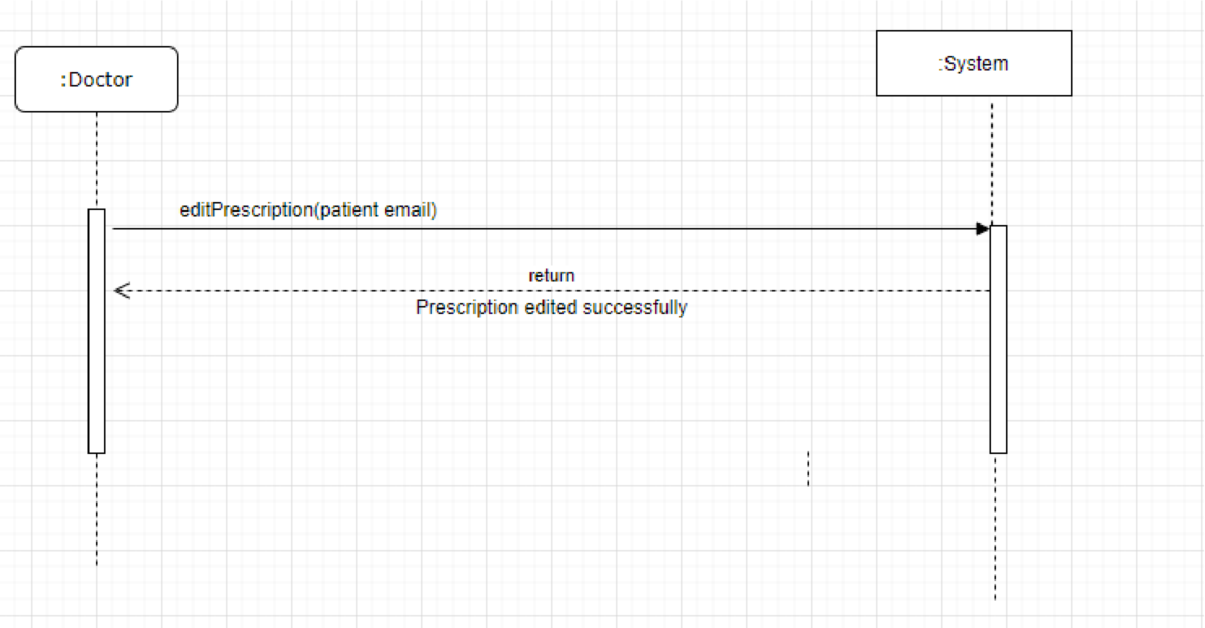
:Doctor
:System
editPrescription(patient email)
return
Prescription edited successfully

Fig 9: Show sequence diagram for edit prescription

2.6.7 Create Schedule

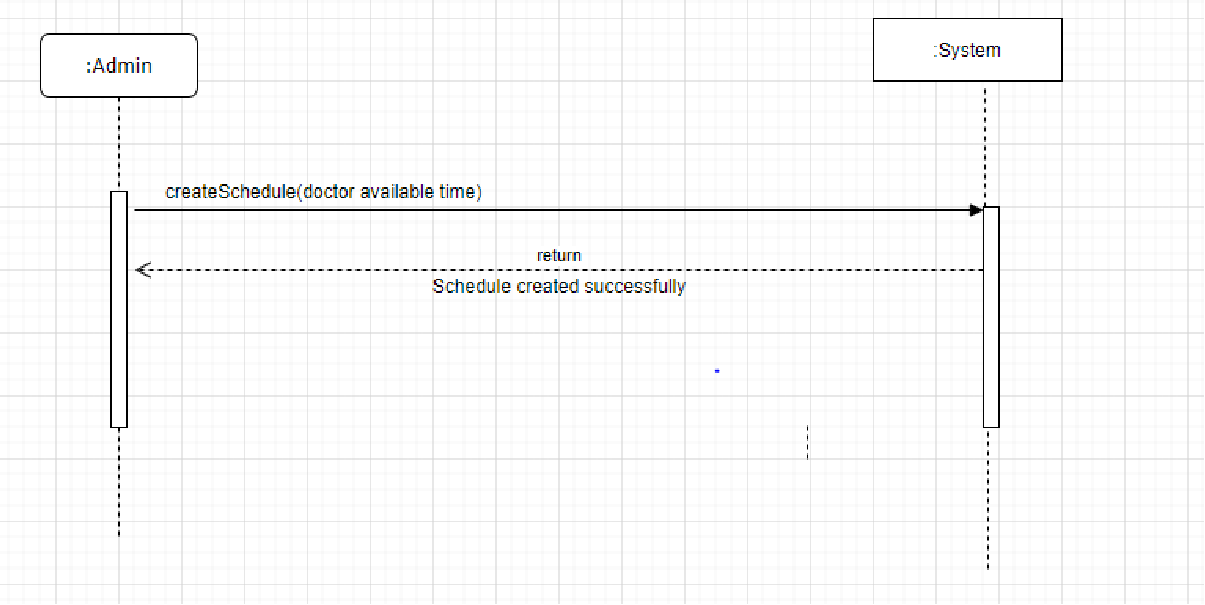
:Admin
:System
createSchedule(doctor available time)
return
Schedule created successfully

Fig 10: Show sequence diagram for create schedule

2.6.8 Chat

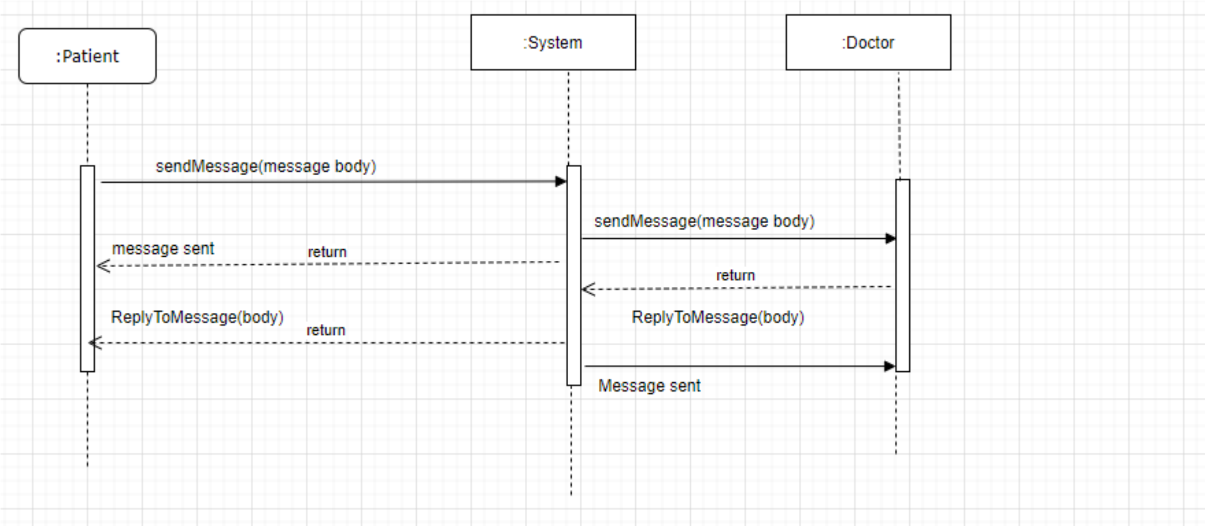

Fig 11: Show sequence diagram for chat

2.6.9 Manages Accounts

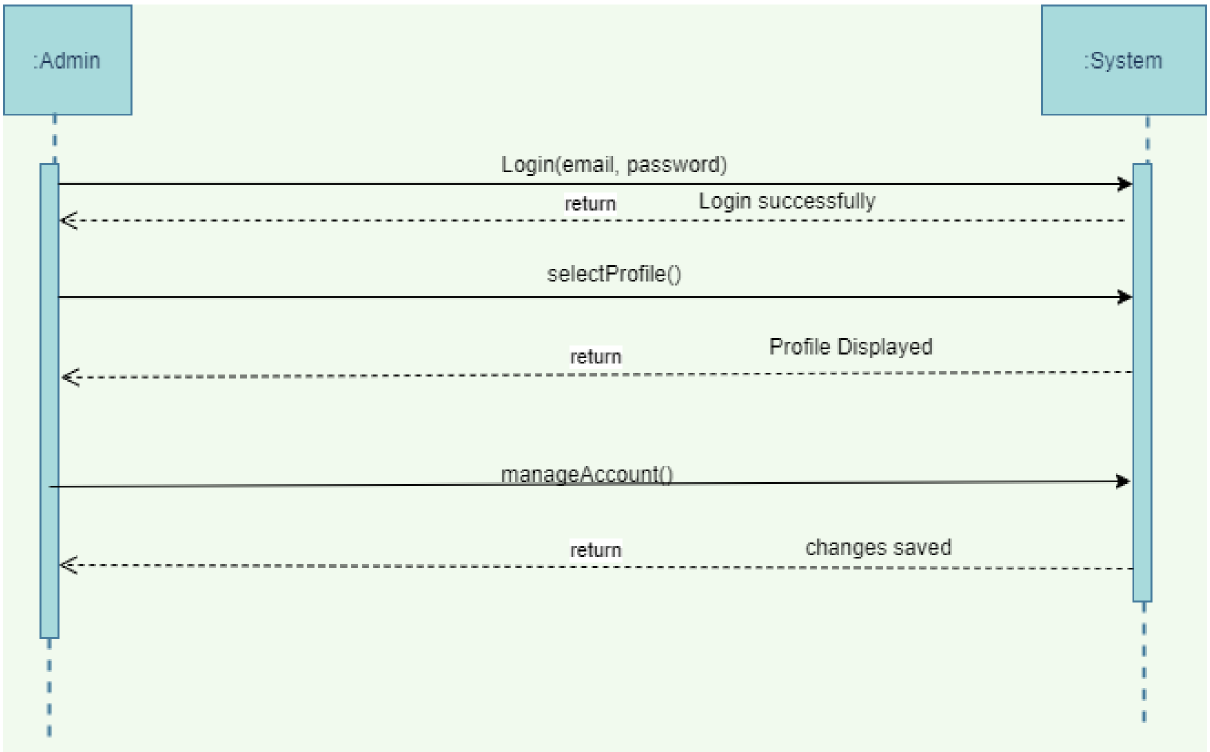

Fig 12: Show sequence diagram for manage accounts

Chapter 3

System Design

3.1 Class Diagram

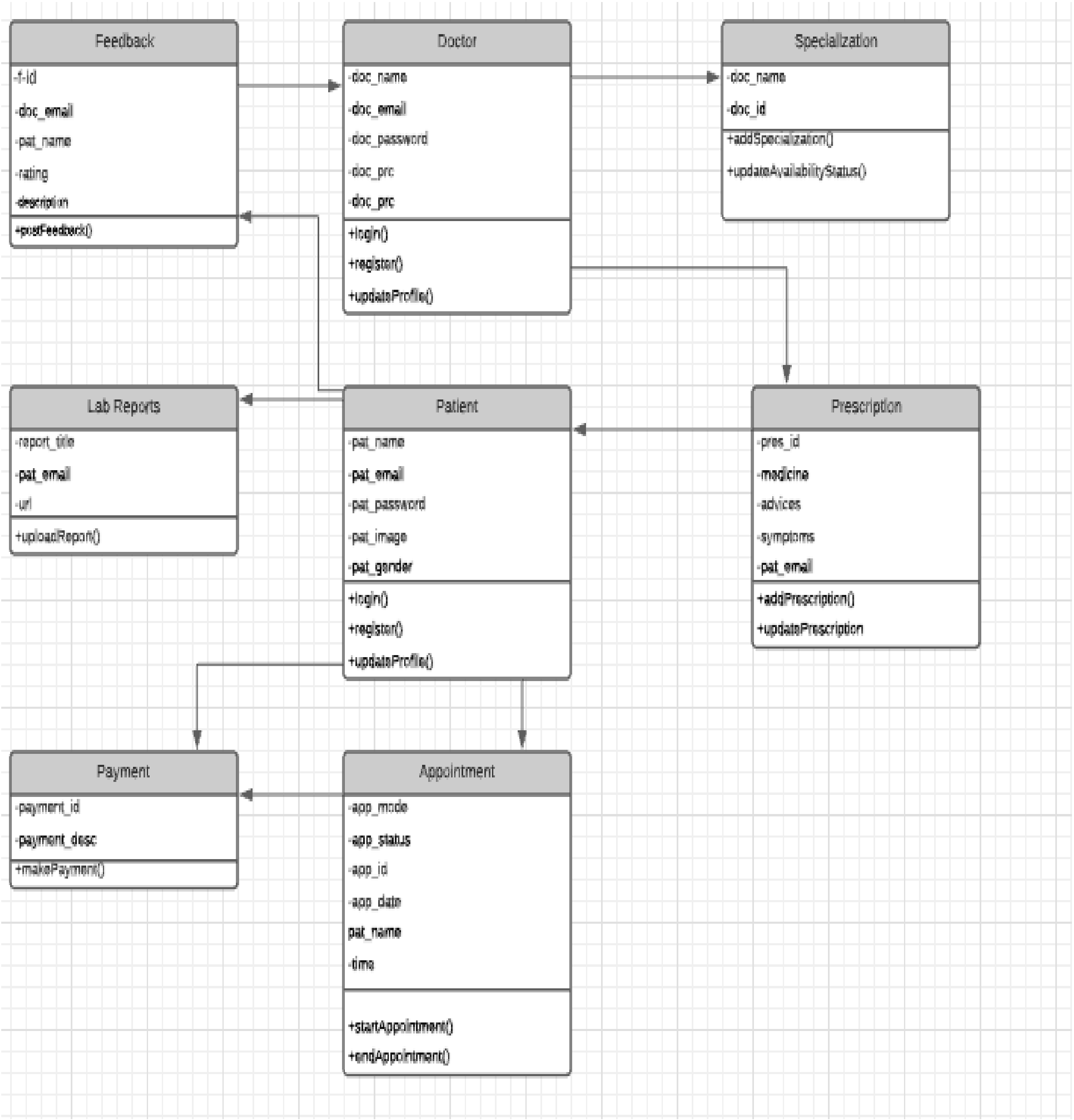

Fig. 13: Class Diagram

3.2 ER Diagram

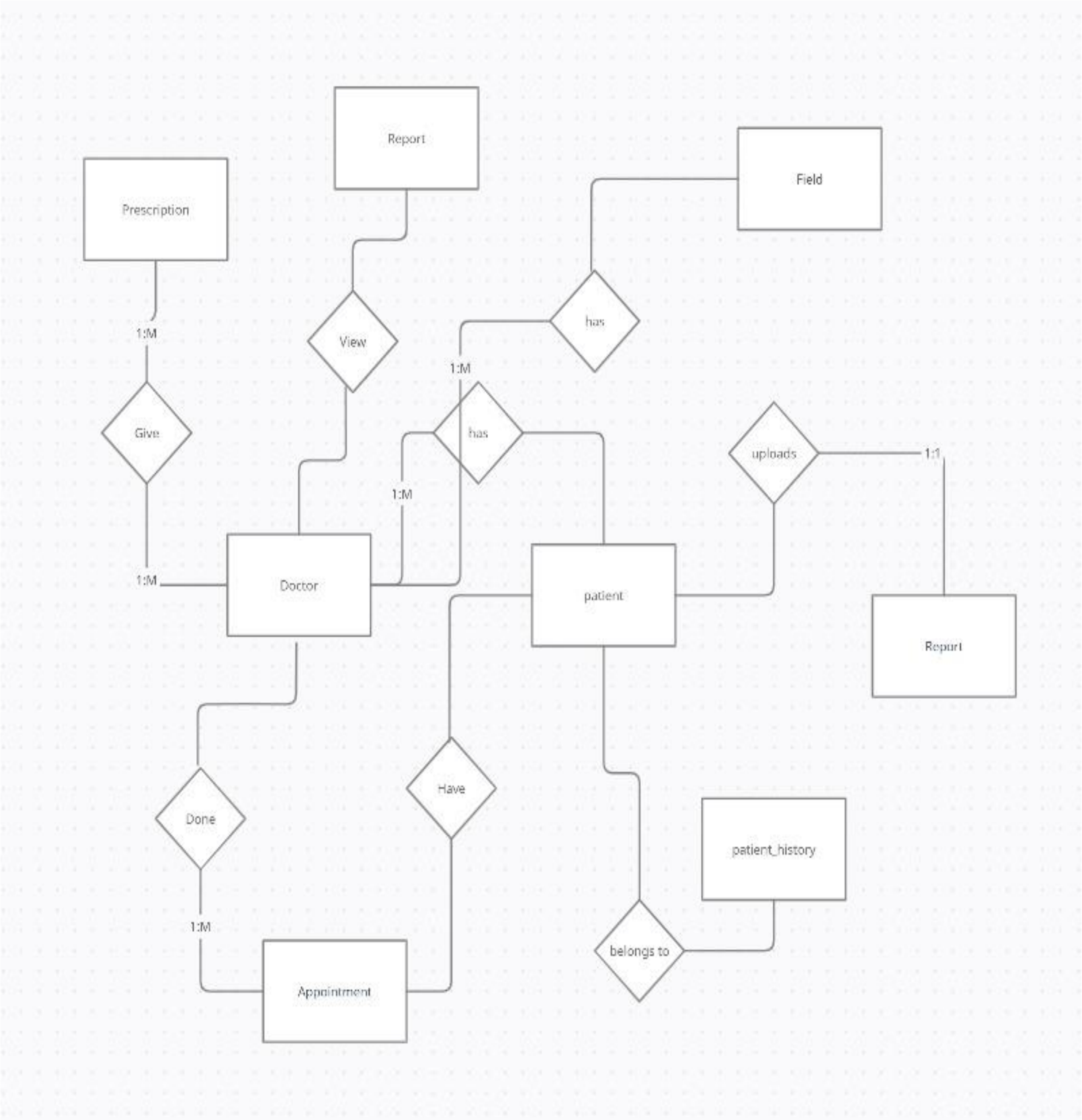

Fig. 14: ER Diagram

Chapter 4

Testing

4.1 Methodology

With this kind of testing approach each and every bit of the specification is used for the model and a sketch was then used to be shown up; the developer then starts modifying his/her code by taking in perspective the given test case whether its fulfilling it or not. Therefore the specifications were applied on every model and consisted testing was applied. This accordingly enclosed the testing that used to be implemented toward the total of the system lifecycle. In the meantime all parts of the system were implemented.

- Begin with a base breaking point that you need to finish.
- Make a record with the point by point fundamental definition a development diagram with a depiction of stream database tables to be utilized a portion chart and a delineation of each segment with the precondition and tables that would be influenced by the section.
- Give the record to the analyzer and work with the analyzer while he or she makes the code to check if the movement in the report can be executed and if the inevitable result of each utilization case can be master.
- On the off chance that the analyzer finds a stage hard to execute or expect he or she is feeling the loss of extra data to understand the supportiveness by then go to arrange 2 generally go to organize 3.
- Request that the analyzer sign on every one of the slip-ups and burdens he or she experienced while dealing with the model utilize.
- Once the model is done and the outcomes between the designer's model and analyzer's model match handle the other need and build up the model to definitive programming.
- Right when the testing approach was executed the going with upsides and drawbacks regarding the testing approach were made sense of it.

4.2 Black Box Testing

The testing method in which all the inner details of code and design of the system are ignored and only focused on the result as per input. It is also known as behavioral or functional testing.

4.3 White Box Testing

It is the method of testing in which each inner detail or logic of the system is tested. All the structure procedures loops are tested in white box testing. It is also known as glass box testing.

4.4 Verification

This process is done to assure that the system is working as intended or as we assumed at the starting of implementing the system.

4.5 Validation

The process used to assure that we implemented the system as per user requirement and result are the same as end user wants.

4.6 Test cases

Following is the summary of test cases

4.6.1 Do Registration:

Test case ID	TC-01
Strategy	White Box
Test name	Sign Up
Objective	The purpose of this test case is to register the Patient/Doctor successfully.
Pre-condition	They must be connected to internet.
Steps to perform	Give details for registration and Press the sign up button
Expected Result	They should be successfully registered.
Test result	Passed.

Table 41: Show test case for registration

4.6.2 Do Login

Test case ID	TC-02
Strategy	White Box
Test name	Login
Objective	The purpose of this test case is to login the Patient/Doctor and Admin successfully.
Pre-condition	They must be registered into the system's database and connected to internet.
Steps to perform	Give login details and press the login button
Expected Result	They should be authenticated.
Test result	Passed

Table 42: Show test case for login

4.6.3 Searches Doctor

Test case ID	TC-03
Strategy	Black Box
Test name	Search Doctor
Objective	The purpose of the test case is to check either patient is able to search a doctor of their own choice or not.
Pre-condition	They must be authenticated.
Steps to perform	They will select the search box and give some key words.
Expected Result	They can successfully be able to give its requirements for a specified doctor.
Test result	Passed.

Table 43: Show test case for search doctor

4.6.4 Makes Appointment

Test case ID	TC-04
Strategy	White Box
Test name	Book Appointment
Objective	The purpose of this test case is to give the patient an ability to book an appointment with the doctor of his/her own choice.
Pre-condition	They must have authenticated.
Steps to perform	They will have to set the schedule of the appointment.
Expected Result	They can create an appointment request of his/her own choice.
Test result	Passed.

Table 44: Show test case for make appointment

4.6.5 Makes Payment

Test case ID	TC-05
Strategy	Black Box
Test name	Make Payment
Objective	User can pay the fee or other dues to doctor/admin online.
Pre-condition	They must be an appointment request made by the patient.
Steps to perform	Patient will have to select method for payment.
Expected Result	Payment must be done by Patient to relevant doctor
Test result	Passed.

Table 45: Show test case for make payment

4.6.6 Lab Test Reports

Test case ID	TC-06
Strategy	White Box
Test name	Lab Test Reports
Objective	The purpose of the test is to make sure that patient can show his/her reports to doctor
Pre-condition	They must be authenticated.
Steps to perform	They will have to click on upload report button
Expected Result	Report is saved in the system with patient identity
Test result	Passed.

Table 46: Show test case for Lab test reports

4.6.7 Edits & Views Profile:

Test case ID	TC-07
Strategy	White Box
Test name	Edit & View Profile
Objective	The purpose of this test case is edit & view profile of Patient/Doctor successfully.
Pre-condition	They must be authenticated.
Steps to perform	Select profile and edit/view details and select update.
Expected Result	They info must be successfully updated/viewed.
Test result	Passed.

Table 47: Show test case for edit and view profile

4.6.8 Schedule

Test case ID	TC-8
Strategy	White Box
Test name	Schedule
Objective	The purpose of the test is that doctor will have proper schedule according to his/her availability
Pre-condition	Doctor must be authenticated
Steps to perform	Admin will contact doctor for checking his/her availability. Admin will create the schedule
Expected Result	Schedule is saved in the system with doctor identity
Test result	Passed.

Table 48: Show test case for schedule

4.6.9 Patient History

Test case ID	TC-9
Strategy	White Box
Test name	Patient History
Objective	The purpose of this test case is to maintain the history of patients
Pre-condition	Patient must be authenticated.
Steps to perform	Patient will request for his history. System will check his history and will display
Expected Result	The history must be successfully displayed into the system.
Test result	Passed.

Table 49: Show test case for patient history

4.6.10 Prescription

Test case ID	TC-10
Strategy	White Box
Test name	Prescription
Objective	The purpose of this test case is that doctor can provide prescriptions to patient and patient can view it
Pre-condition	Doctor must be authenticated.
Steps to perform	After meeting, doctor press prescription button and write the description and this description will displayed to patient
Expected Result	They prescription must be successfully displayed into the system.
Test result	Passed.

Table 50: Show test case for prescription

4.6.11 Chat

Test case ID	TC-11
Strategy	White Box
Test name	Chat
Objective	Verify Chat between patient and doctor
Pre-condition	Patient must be in an active appointment with doctor.
Steps to perform	Press the chat button Type message and press the send button
Expected Result	Message delivered successfully
Test result	Passed.

Table 51: Show test case for chat

4.6.12 Manages Patient’s/Doctor’s Account

Test case ID	TC-12
Strategy	White Box
Test name	Manage Accounts
Objective	The purpose of this test case is to check that account management is working successfully or not.
Pre-condition	Admin must be authenticated.
Steps to perform	Select the account make changes and select save.
Expected Result	They info must be successfully updated into the system.
Test result	Passed.

Table 52: Show test case for manage accounts

Chapter 5

Conclusion

Our final year project Smart Health Care App is a real time android application which is providing a better platform for Patients and Doctors has been completed successfully.

This has been made with a lot of hard work and its almost bug free. The main purpose of this App is to provide a secured and better platform for Patients who can get health issues solved with just few clicks. This platform also provides opportunity to Doctors who get Patients remotely and enhance their experience.

Our project has provided us an opportunity to learn and experience many new things like working with **Mobile Application Development using Flutter and Firebase**. All these technologies are quite popular in the software industry. We have learnt how professional methodologies work.