03-134142-004 ABDUL REHMAN

03-134142-070 M.USMAN TAHIR

Online Doctor Appointment system (ODAS)

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: DAWOOD AKRAM

Department of Computer Sciences

Bahria University, Lahore Campus

January 2018

Certificate

We accept the work contained in the report titled
“ONLNE DOCTORS APPOINTMENT SYSTEM”,
written by
ABDUL REHMAN
M.USMAN TAHIR

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: DAWOOD AKRAM

____(Signature)

January 11th, 2019

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134142-004	ABDUL REHMAN	
03-134142-070	M.USMAN TAHIR	

Date : January 2019

Specially dedicated to

Our Beloved Mother and Father

(ABDUL REHMAN)

(M.USMAN TAHIR)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to my research supervisor,Mr DAWOOD AKRAM for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition,We would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

ABDUL REHMAN
M.SMAN TAHIR

Online Doctor Appointment System (ODAS)

ABSTRACT

This project is about an android application in this project we are going to develop ODAS which is going to change the life of doctor and patient. Through that project the patient can register themselves by using simple form after registration the patient get the confirmation

The purpose of ODAS is to automate the existing manual system with the help of technology. In this the valuable data/information is stored for long period of time with the easy access. This will help both, The Doctors and patients.

This is going to be a Android app and with the easy graphic user interface so that every person can easily understand it and use it.

The Study proves that the most Doctors doesn’t have the knowledge of the information technology and they don’t have the time for it because of their busy life.

So by developing this application can improve their life style and can also reduce their manual work and can reduce their expenses.

On the other hand this application will be helpful for the patient because they don’t have to call someone they know for the appointment and the can get the desired time for their check-up.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Mobile has become a basic contact tool, everyone prefers to possess and take it with them. This technology has established a base to conquer the traditional desktop based approach. With advancement in the field of technology and with the development of android phones web and android applications are gaining popularity amongst the people. Mobile application that tend be helpful in daily life becomes the habit of user. Android a mobile based Operating system developed by Google Company based upon Linux Kernel designed for mobiles with touch screens like smartphones and tablets.

ONLINE DOCTORS APPOINTMENT SYSTEM (ODAS) The establishment and improvement of doctor-patient interaction system is a very important requirement especially now, when the mobile communication technology is developing rapidly. The advantages of mobile app can be made full use of to make up the time and distance gap between doctors and patients and to provide fast and adequate medical services. Through the connection between mobile terminal sand specific service, both doctors and patients are able to obtain required data to achieve a better interaction .The platform, mobile applications and database technology are all gradually maturing, so that we can develop a doctor- patient interaction system on Android and web platform to meet the needs of the patient and provide doctors more efficient and convenient means of communication with patients

1.2 Problem Statements

In these days' time is very important factor and for a patient it is very difficult to wait for a doctor because a patient needs urgent meeting with doctor to cure his/her disease and it is a very hectic process.

1.3 Aims and Objectives

Our main objective is to provide services to the people to save their time through the use of innovation by proving them layman services to solve their appointment problems through android application.

1.4 Scope of Project

Through our services we hope to accomplish following major goals:

Doctor /Patient benefits:

- Time saving.
- Quality assurance.
- Consistent and timely service.
- Customer supportive.
- Customer comfort.
- User friendly.

Service providers benefit:

- Business start-up.
- Research based learning activity

CHAPTER 2

Software Requirement Specification

2.1 User classes and characteristic

This system is intended for the following users

Application user

- Typical user such as Patient and Doctor.
- Doctor can register to see the Appointments.
- Who want to make appointment often for him/herself or for their Family.
- Uses a mobile Application to register.
- Patient can see the free slots to make the appointment.
- View all appointments.

Admin

- A database admin who is using the firebase module**.**
- Admin can login**.**
- Admin can see the registered users.
- Admin can block the users Account.

2.2 Operating environment

Odas is Android Application so it required only Android Environment.

2.2.1 Software environment

For Android application android studio will be used as a tool and java language for development with xml. Different APIs used for interference design. Android Studio 3.1.3 will be used which support API versions up to 15.

2.3 Design and Implementation constraints

Mobile application is android specified. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well-written APIs and libraries. For data storing system is using Firebase database where all the record of Patient and Doctor will be stored and admin can see this record.

2.4 Assumption and Dependencies

ODAS (Online Doctor Appointment System) have Database so one assumption about the product is that admin must have access to database from Mobile Application or from Web Portal which is compatible with our application. Application is only for android environment.

Another assumption is that users must have internet connection in their mobile, so that status of availability could update on real time using database. Internet is required for the system. Server down effect on communication between database and mobile application

2.5 System requirement chart

Following is the system requirement char table.

Table 2.1 System Requirement Chart

ID	Priority	Type	Source	Used in Use case	Description
1	High	Functional	End User’s	U1	Download Application from play store
2	High	Functional	End User’s	U2	Registration is done by the user to interact with system
3	High	Functional	End User’s	U3	Login in to application to use the application
4	High	Functional	End User’s	U4	Search the doctor
5	High	Functional	End User’s	U5	Make the appointment according to disease
6	High	Functional	End User’s	N/A	Data stores into database on real time
7	Medium	Functional	End User’s	U4	View Appointments by user in application
11	Medium	Non Functional	End User’s	U6	Status of slots
12	High	Non Functional	End User’s	N/A	Server response should be keep in mind while designing data retrieval.

13	High	Functional	End User’s	U7	View Payment by user
14	Low	Functional	End User’s	N/A	Payment status should be Paid in database for Appointment
15	High	Functional	End User’s	U8	View All Record by admin
16	High	Functional	End User’s	U9	Delete Record by admin
17	High	Functional	End User’s	U10	Logout by user or admin
18	High	Non Functional	End User’s	N/A	All the passwords should save in encrypted form.
19	High	Non Functional	End User’s	N/A	Availability of internet is necessary for using application and communication.
20	High	Non Functional	End User’s	N/A	Only that slot should update where the Appointment is made for system to work correctly.

.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Design Specification

This document specifies the steps required to develop the ODAS Mobile

Application, describes all data, functional and behavioural requirements of the software

Under development. The resulting system should be secure. As a user; system will be Easy to use.

Following artefacts includes:

1. Use case diagram
2. Use case description
3. Domain Model
4. Sequence Diagram
5. Collaboration Diagram
6. Operation Contracts
7. Design Class Diagram

3.1.1 Use case Diagram

There would be three types of user which will interact with the application, DOCTOR, PATIENT and System Administrator. So, all of them would have different type of requirements. Doctor’s availability on the Application. This Platform provides the best Doctor’s services to your desired time which saves your time and also gives the comparison of Behaviour among different Doctors to save your money. Customer can use the application to get the information of a Doctors.

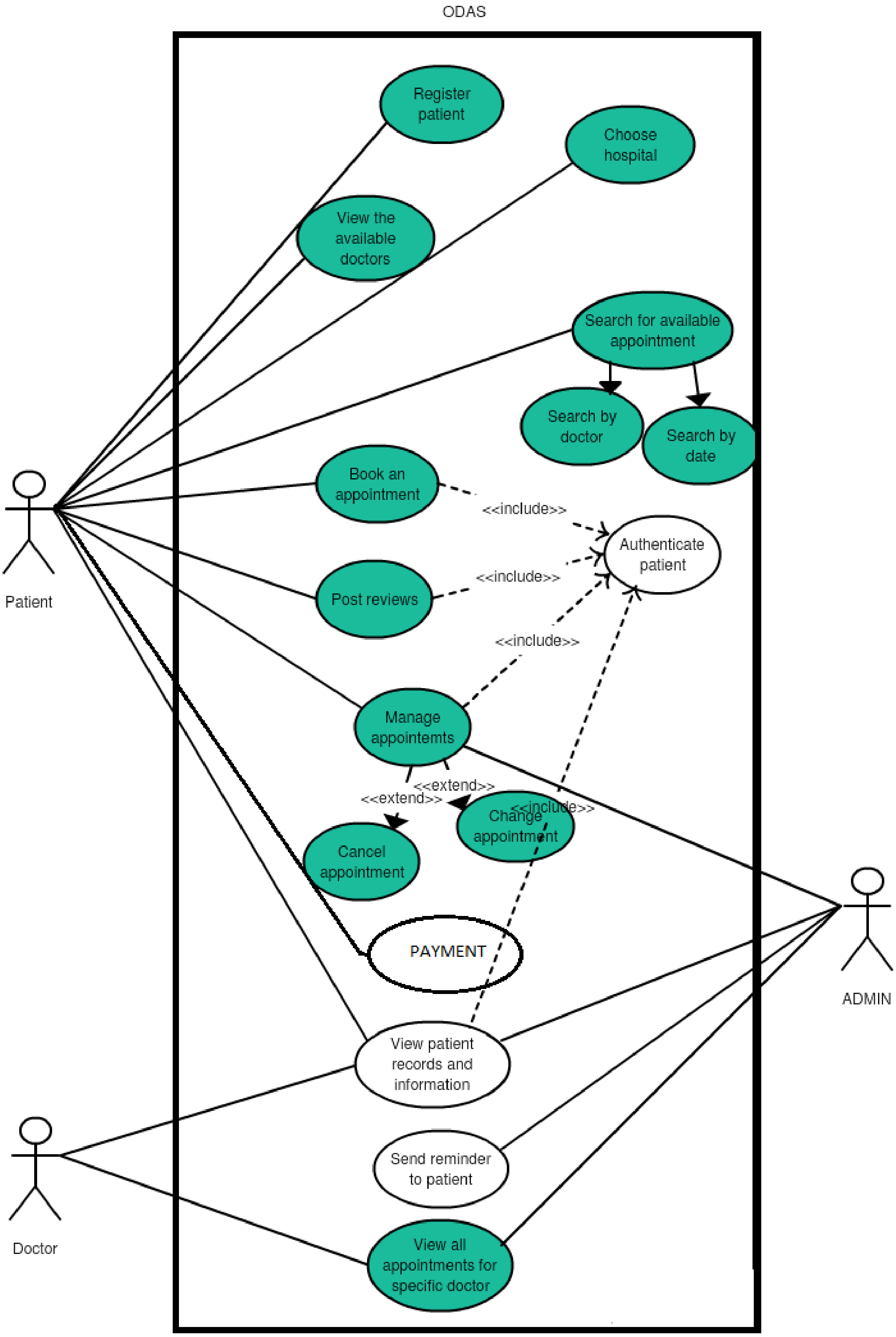


Figure 1 use case

3.2 Use case Description

Following are the narrative parts of every bubble in above use case diagrams

3.2.1 Use-case #1

Use-case description typically contains the following parts:

Table 3. 1: (Use-Case-U1 Download Application)

Name	Download Application
Use-Case ID	U1
Priority	High
Primary Actor	Application User (Doctor, Patient)
Other participating Actor(s)	—
Description	This use case describe the event of download the application from play store by application user so that they can interact with the system.
Pre-condition	User must have android phone with internet availability.
Trigger	This use case initiate when a user is going to download the application.
Typical flow of events	1. User connect the internet in android device 2. Open the play store application 3. Search the application named ODAS or ONLINE DOCTOR APPOINTMENT SYSTEM 4. Install the application
Alternate flow of event	Alt-1. Internet is not connected/Limited internet Alt-4. Installation error
Post condition	Application is downloaded.
Alternate post condition	Application is not downloaded

3.2.2 Use-case #2

Use-case description typically contains the following parts:

Table 3. 2: (Use-Case-U2 Registration)

Name	Registration
Use-Case ID	U2
Priority	High
Primary Actor	Application User (Doctor, patient)
Other participating Actor(s)	—
Description	This use case describe the event of registration of application user, once user has downloaded application then he has to register himself first in the application to interact with system.
Pre-condition	User must download and install the application.
Trigger	This use case initiate when a user is going to register himself.
Typical flow of events	1. User open the application 2. Click on Application use button 3. Click on Registration button 4. Enter all required data 5. Press Register button to register himself.
Alternate flow of event	Alt-1. Click on Login button Alt-2. Don’t follow the rules for entering data
Post condition	User is registered.
Alternate post condition	User is not registered.

3.2.3 Use-case #3

Use-case description typically contains the following parts:

Table 3. 3: (Use-Case-U3 Login)

Name	Login
Use-Case ID	U3
Priority	High
Primary Actor	Application User (Doctor, patient)
Other participating Actor(s)	—
Description	This use case describe the event of logging in of user after registration.
Pre-condition	User must registered himself first
Trigger	This use case initiate when user is going to login.
Typical flow of events	1. User open the application 2. User clicks on login button and enter the user name and password. 3. User is logged in after clicking the button login
Alternate flow of event	Alt-1. User enter wrong username or password
Post condition	User is logged in.
Alternate post condition	User is not logged in.

3.2.4 Use-case #4

Use-case description typically contains the following parts:

Table 3. 4: (Use-Case-U3 Login)

Name	Search the Doctor
Use-Case ID	U4
Priority	Medium
Primary Actor	Application User (Patient)
Other participating Actor(s)	--
Description	This use case describe the event of availability of Doctor by the application user.
Pre-condition	Patient must logged in.
Trigger	This use case initiate when Patient is going see the Doctors.
Typical flow of events	1. Patient log in into application. 2. Click on button to see the Doctos 3. View the Doctors
Alternate flow of event	Alt-1. Patient is not logged in due to wrong user name or password Alt-2. Patient clicks on another button rather than view
Post condition	Patient viewed the Doctor
Alternate post condition	Patient didn’t view the Doctor

3.2.5 Use-case #5

Use-case description typically contains the following parts:

Table 3. 5: (Use-Case-U8 Make Appointment)

Name	Make Appointment
Use-Case ID	U5
Priority	High
Primary Actor	Application User (Patient)
Other participating Actor(s)	-----
Description	This use case describe the event of making Appointment by User.
Pre-condition	User must logged in.
Trigger	This use case initiate when user is going to make appointment.
Typical flow of events	1. User log in into application 2. Click on Appointment Button 3.click on book appointment button
Alternate flow of event	Alt-1. User is not logged in due to wrong user name or password
Post condition	Appoint is made
Alternate post condition	Appoint is not made

3.2.6 Use-case #6

Use-case description typically contains the following parts:

Table 3. 6: (Use-Case-U8 View slots)

Name	View Slots
Use-Case ID	U6
Priority	High
Primary Actor	Application User (Patient, Doctor),
Other participating Actor(s)	----
Description	This use case describe the event of viewing the availability of free slots.
Pre-condition	User must logged in.
Trigger	This use case initiate when user is going to view slots status.
Typical flow of events	1. User log in into application 2. Click on View Slots button 3. View the Status of Slot
Alternate flow of event	Alt-1. User is not logged in due to wrong user name or password Alt-2. User clicks on another button rather than View Slots
Post condition	User viewed Slots.
Alternate post condition	User didn’t view Slots.

3.2.7 Use-case#7

Use-case description typically contains the following parts:

Table 3. 7: (Use-Case-U8 View Payment)

Name	View payment by user
Use-Case ID	U7
Priority	High
Primary Actor	Admin
Other participating Actor(s)	----
Description	This use case describe the event of viewing payment
Pre-condition	User must have paid.
Trigger	This use case initiate when Patient is going to make Appointment
Typical flow of events	1. click on done appointment
Alternate flow of event	N/A
Post condition	patient viewed Payment
Alternate post condition	Admin didn’t view Payment

3.2.8 Use-case #8

Use-case description typically contains the following parts:

Table 3. 8: (Use-Case-U9 View All Records)

Name	View User Logs
Use-Case ID	U8
Priority	High
Primary Actor	Admin
Other participating Actor(s)	-----
Description	This use case describe the event of checking the record by Admin of all the Data (Doctor And Patient)
Pre-condition	Admin must logged in.
Trigger	This use case initiate when admin is going to view the record.
Typical flow of events	1. Admin clicks on View Record button 3. View the record
Alternate flow of event	N/A
Post condition	Admin has viewed record.
Alternate post condition	Admin didn’t view the record.

3.2.9 Use-case #9

Use-case description typically contains the following parts:

Table 3. 9: (Use-Case-U9 View All Records)

Name	Delete User Record
Use-Case ID	U9
Priority	High
Primary Actor	Admin
Other participating Actor(s)	-----
Description	This use case describe the event of deleting the record by admin. Admin can delete any record from database.
Pre-condition	Admin must be online.
Trigger	This use case initiate when admin is going to delete any user record.
Typical flow of events	1. Admin clicks on Delete Record button 2. Then enter the id of user whose record he wants to delete 3. Then clicks on Delete button
Alternate flow of event	N/A
Post condition	Record is deleted.
Alternate post condition	Record is not deleted due to internet problem or sever down problem.

3.2.10 Use-case #10

Use-case description typically contains the following parts:

Table 3. 10: (Use-Case-U9 logout)

Name	Logout
Use-Case ID	U11
Priority	High
Primary Actor	Application User (Car Driver)
Other participating Actor(s)	----
Description	This use case describe the event of logging out from the application by User.
Pre-condition	User must logged in.
Trigger	This use case initiate when user is going to logout from the application.
Typical flow of events	1. User clicks on Logout button
Alternate flow of event	N/A
Post condition	User is logged out.
Alternate post condition	User is not logged out.

3.3 Sequence Diagram

A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behaviour of a system.

A Sequence diagram is two-dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects.

Because it uses class name and object name references, the Sequence diagram is very useful in elaborating and detailing the dynamic design and the sequence and origin of invocation of objects. Hence, the Sequence diagram is one of the most widely used dynamic diagrams in UML.

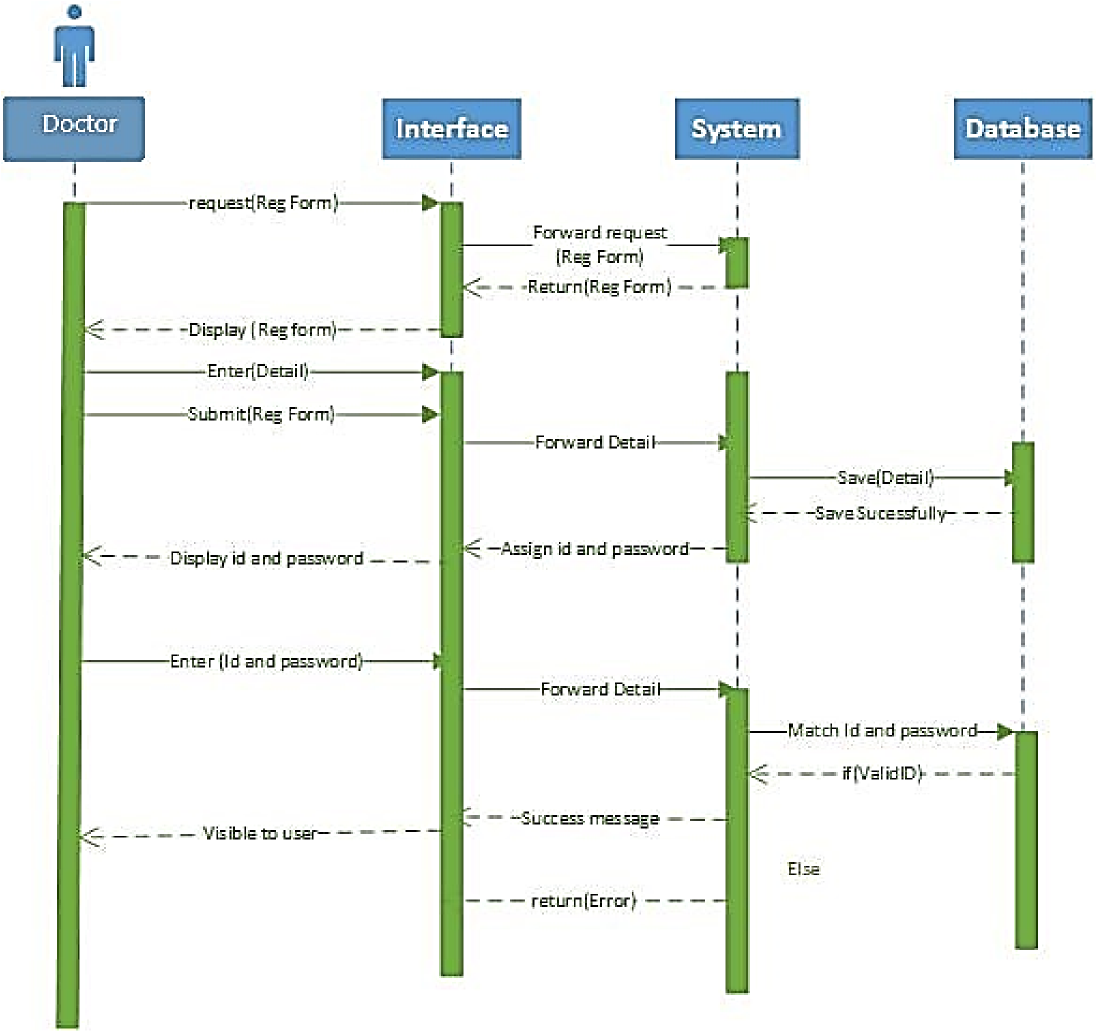


Figure 2 use-case-doctor-registration

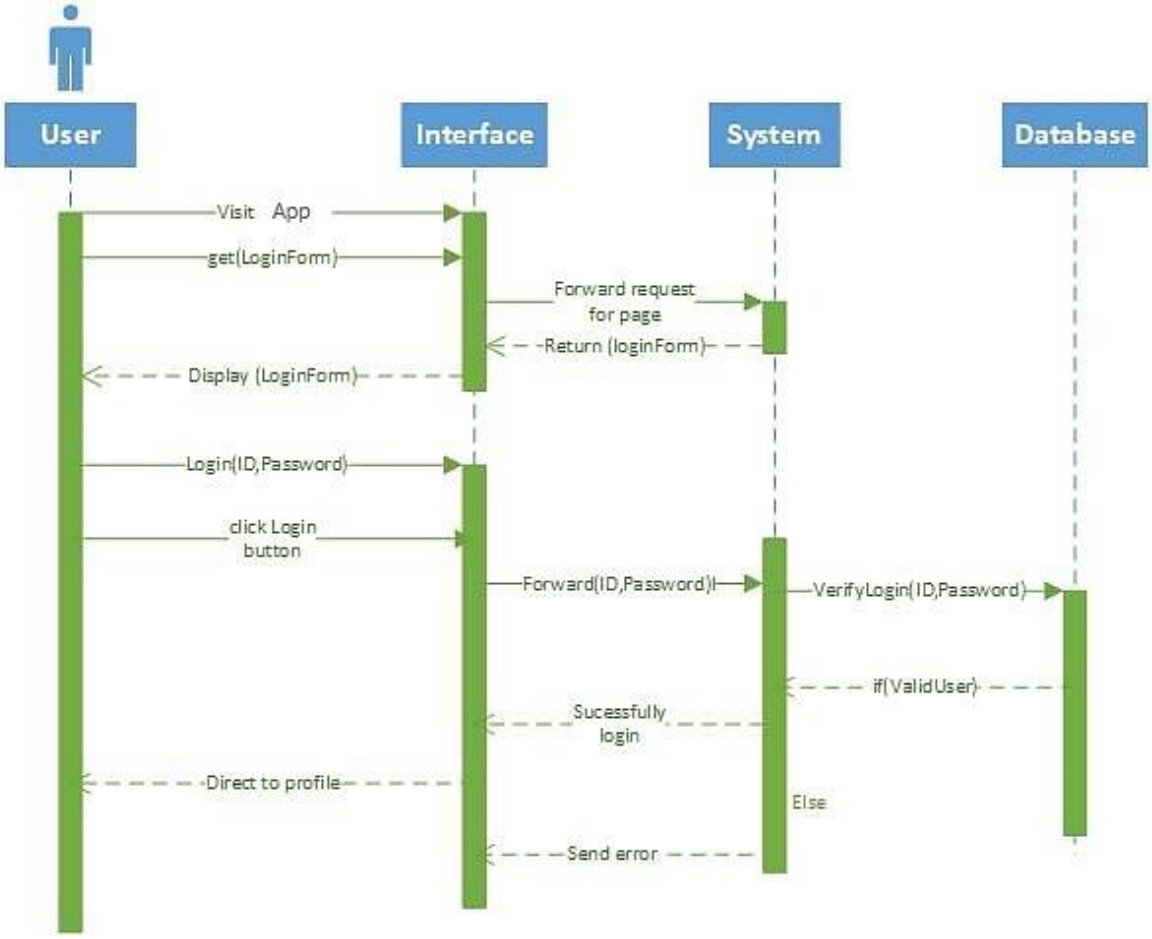


Figure 3 use-case-login

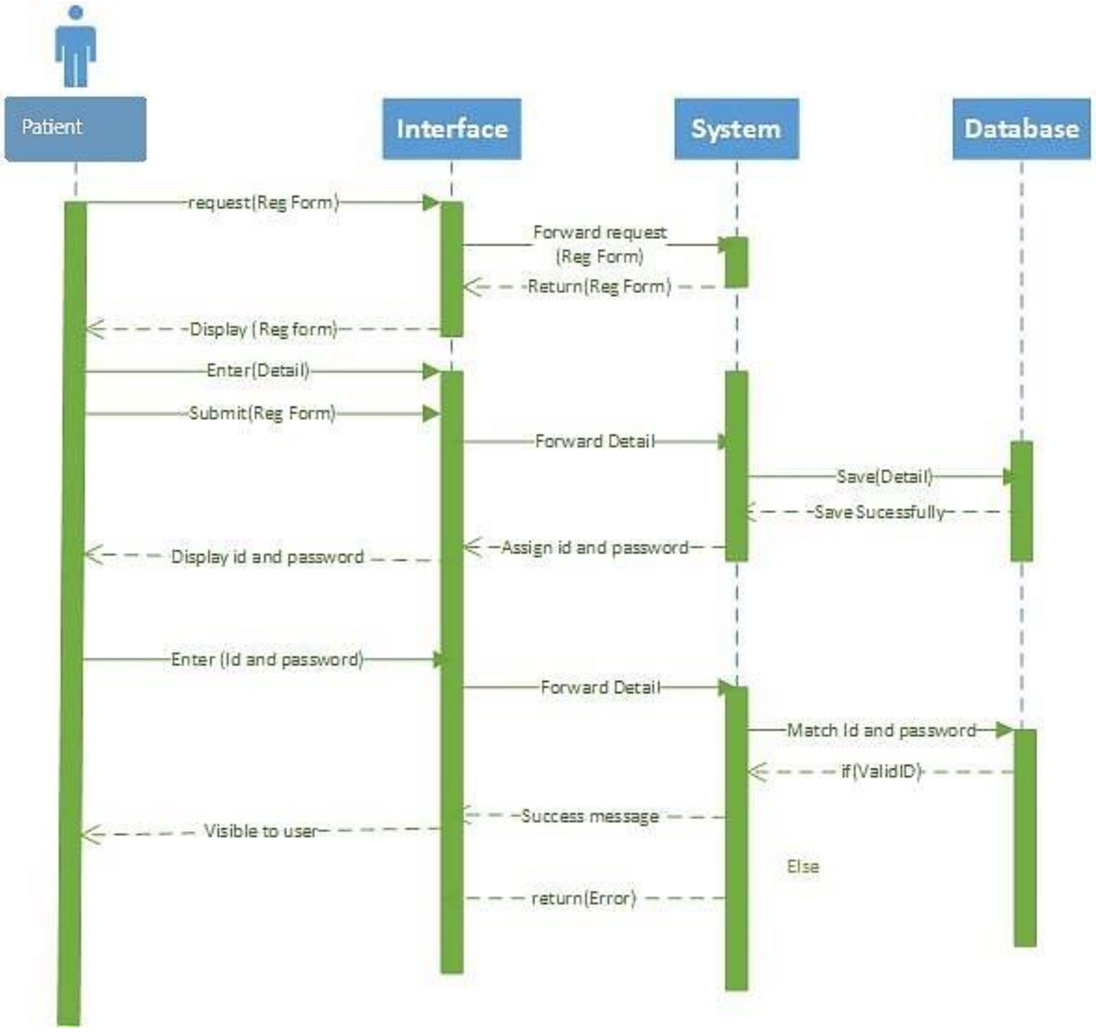


Figure 4 use-case-user-registration

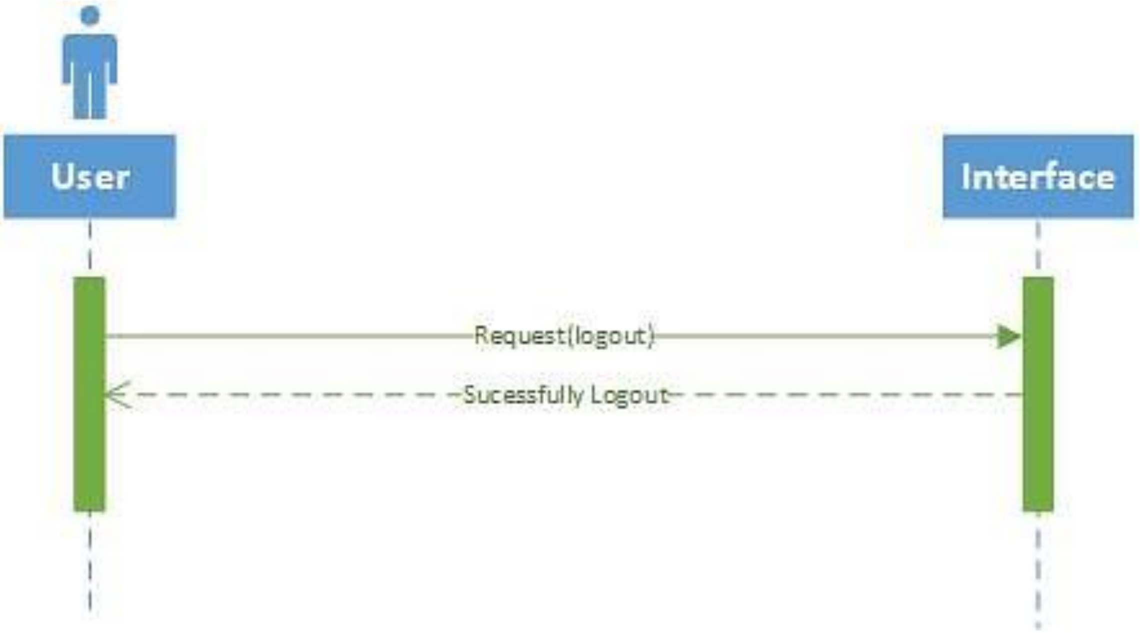


Figure 5 use-case-logout

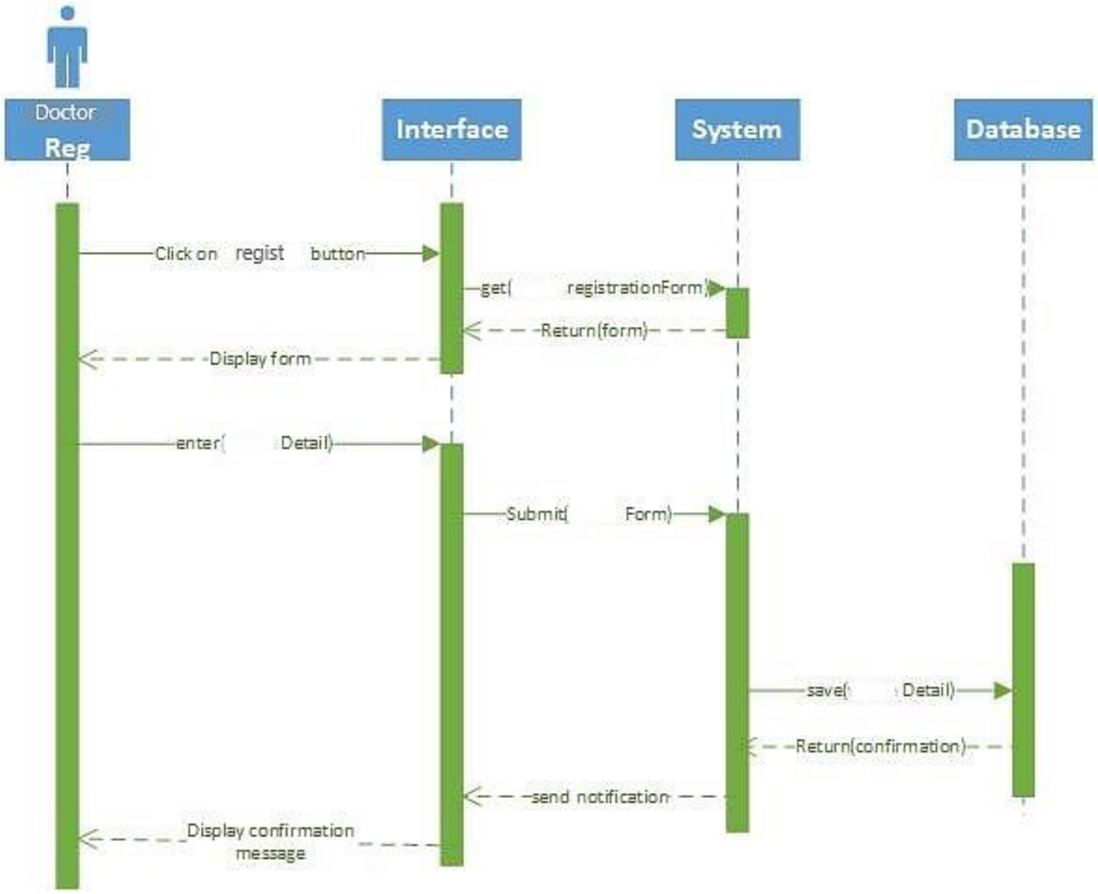


Figure 6 use-case-Appointment

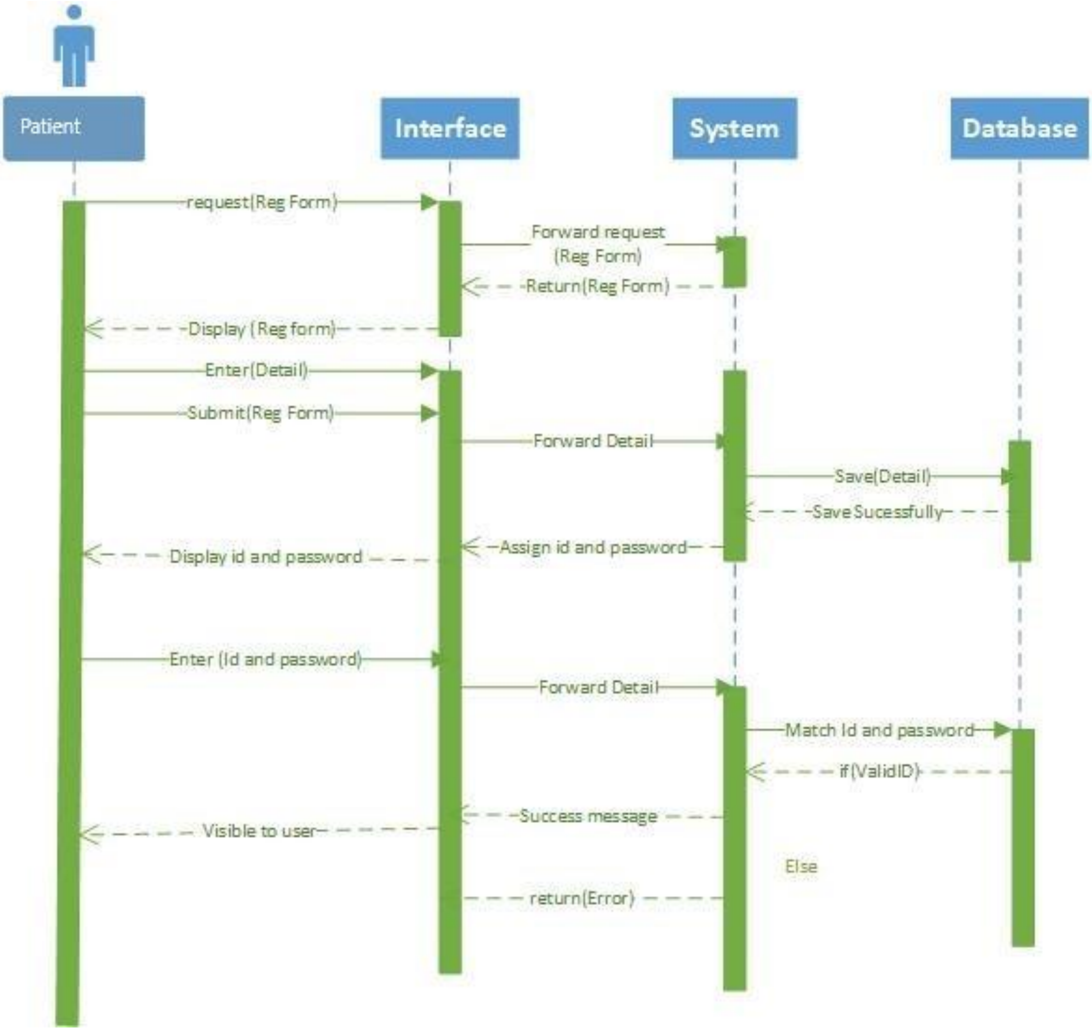


Figure 7 use-case-Search

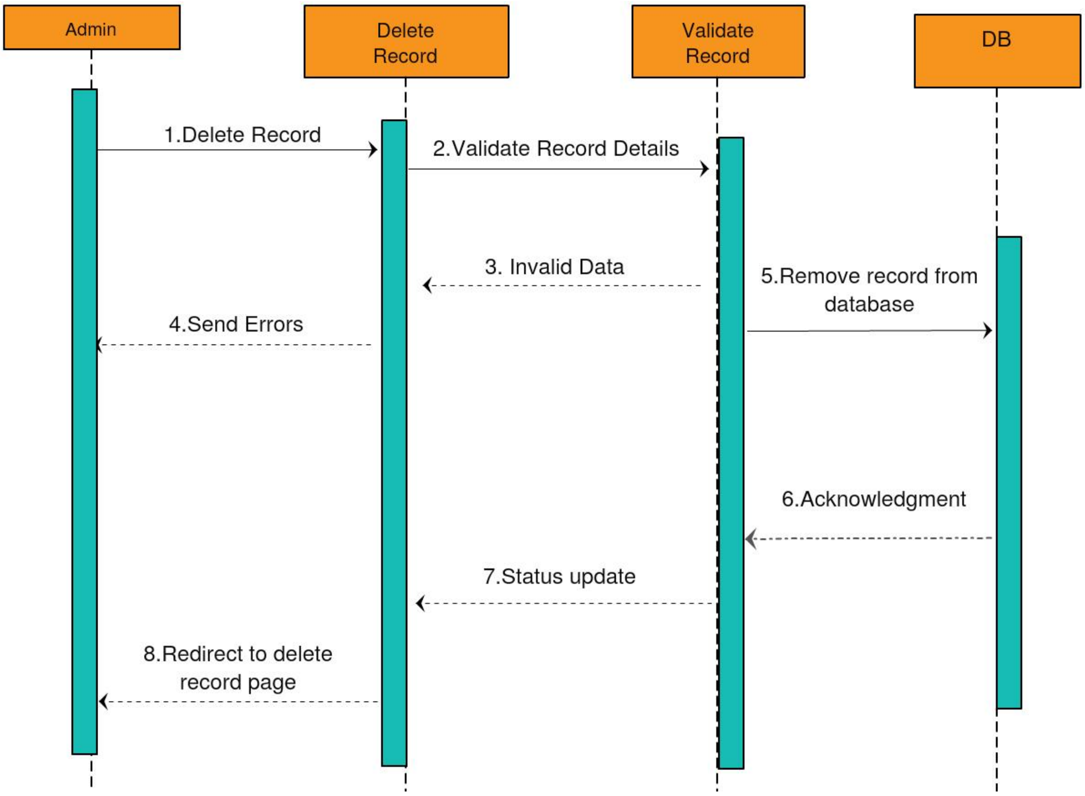


Figure 8 use-case-delete-record

3.4 Collaboration Diagram

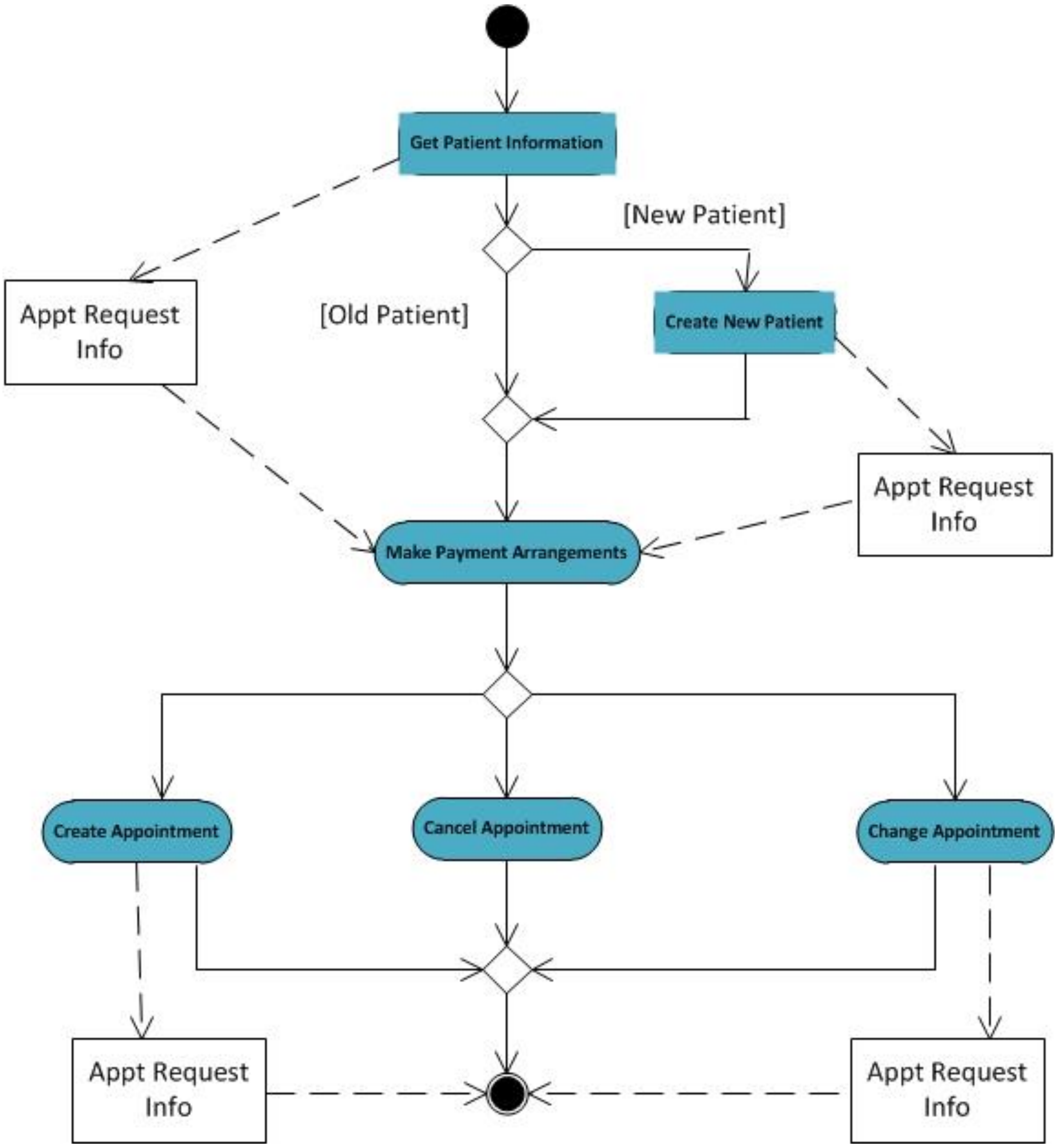


Figure 9 collaboration model

3.5 Domain Model

Domain models represent the set of requirements that are common to systems within a product line. There may be many domains, or areas of expertise, represented in a single product line and a single domain may span multiple product line

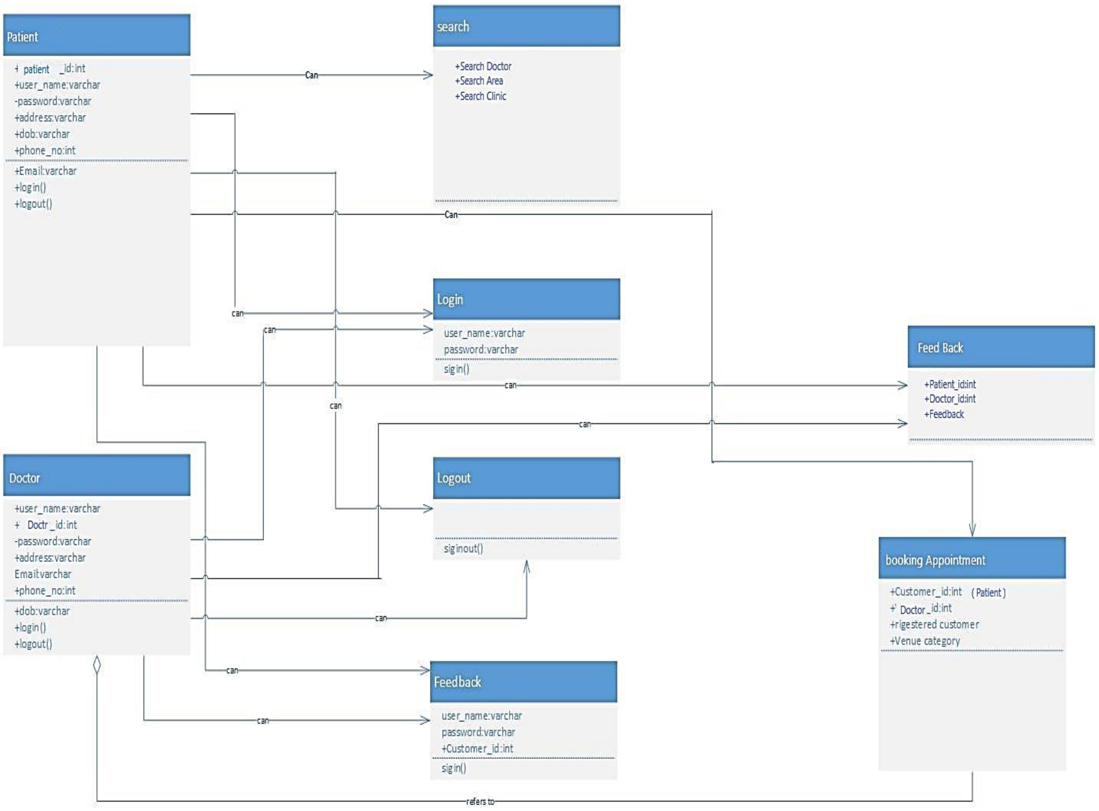


Figure 10 Domain Model

3.6 Design Class Diagram

Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

Capsules are also stereotyped classes, used to represent concurrent threads of execution in real-time systems. In such cases, other design classes are 'passive' classes, used within the execution context provided by the 'active' capsules. When the software architect and designer choose not to use a design approach based on capsules, it is still possible to model concurrent behaviour using 'active' classes.

Active classes are design classes, which coordinate and drive the behaviour of the passive classes - an active class is a class whose instances are active objects, owning their own thread of control.

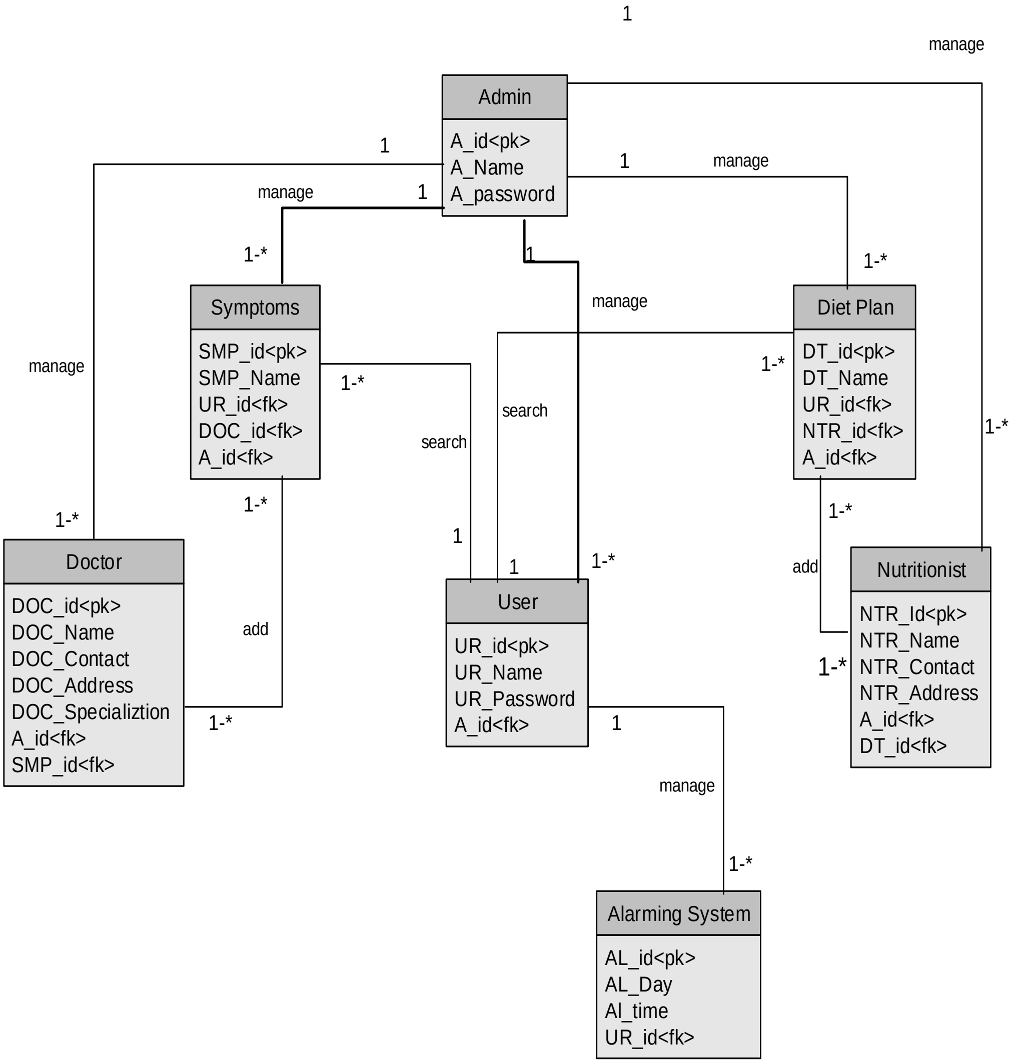


Figure 11 Design Class Model

3.7 ODAS ERD

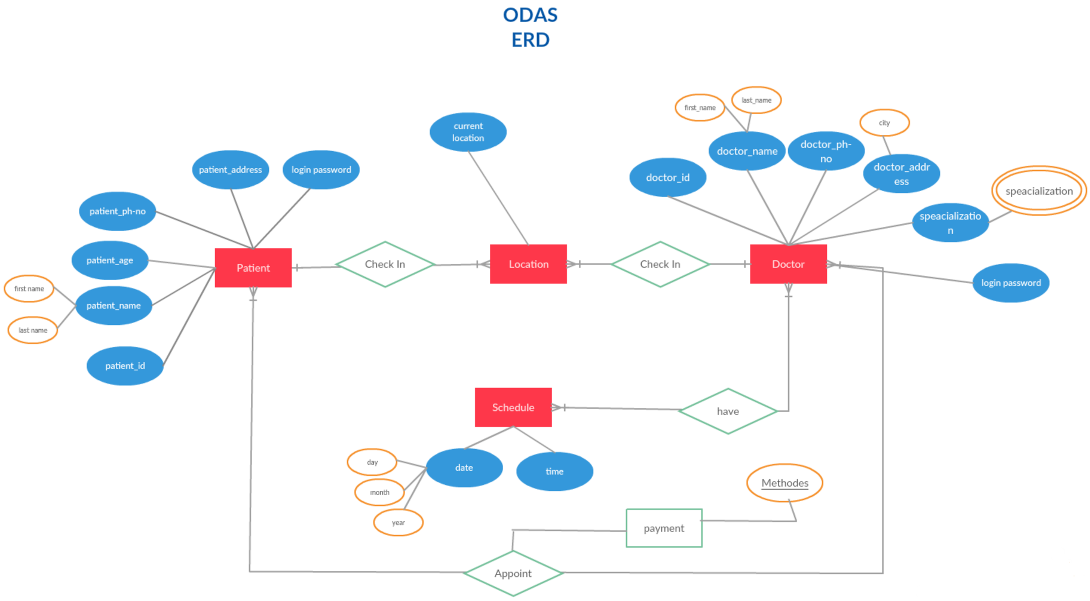


Figure 12 ERD (ODAS)

CHAPTER 4

DATA AND EXPERIMENTS or IMPLMENTATION

4.1 Methodology

Many methodologies now exist for developing applications on the web, so to choose an appropriate style for this project, research into the various options is necessary. A relevant summary of this research will be presented below, followed by a justified choice, and a description of how the methodology will be applied in this project.

4.1.1 Present Available Methodologies

In the first methodologies created are described as being based on the waterfall model, which instructed developers to create systems in a linear step-by-step manner. The model states that before any work can begin on the actual software solution, a detailed feasibility requirements analysis must be conducted, and converted into a specification for the system. The software and all its features should then be thoroughly designed before implementation can occur. The finished product should then be tested, and any problems rectified, before being deployed in the appropriate environment.

Waterfall model was used successfully for many years, however eventually the limitations of the model become clear. The requirements and specifications for a project aren‘t always fully known initially, so it can become apparent later during the implementation phase that the system design is inconsistent, which can often lead to project failure. These problems were addressed in later years, when agile methodologies were devised to develop systems faster, and with greater flexibility.

4.1.2 Choosing a Suitable Methodology

This project will be developed by using agile methodology because agile models handle change in requirements easily, divide complex projects into parts which simplifies the project and deliver a working software after every iteration. In agile there are many model but here Feature Driven Development will be used as it is one of the simplest and easy to follow agile methodology.

4.1.2.1 Model the Complete System

This first phase defines the scope for this project and develop a big picture of the system by analysing different learning platforms that how these platforms are working, managing users and data. Entity Relationship Model build in this phase to help us in understanding the system

4.2 Implementation

The implementation phase of the project is the development of the designs produced during the design phase.

4.2.1 Technical Aspects

Since ODAS is Android Application, both Database and Mobile Application are necessary. The system consist of Front-end and Back-end

4.3 Tools Used for Implementation

4.3.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development.

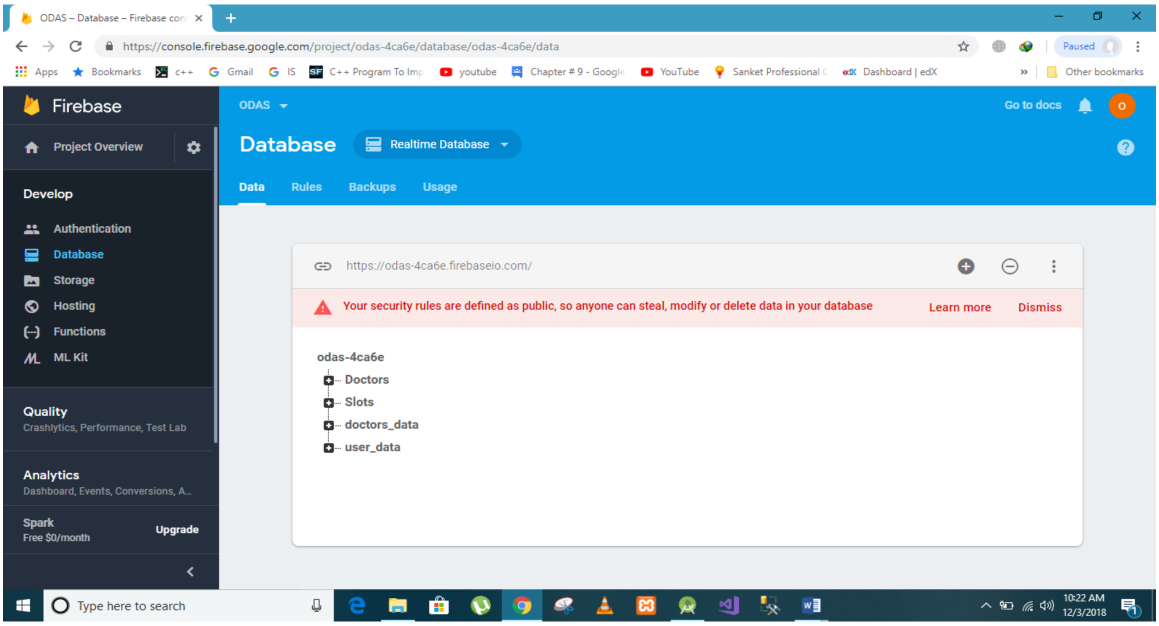
4.3.2 Firebase

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

CHAPTER 5

USER MANUAL

5.1 How to access ODAS (Admin Panel)



https://console.firebase.google.com/project/odas-4ca6e/database/odas-4ca6e/data

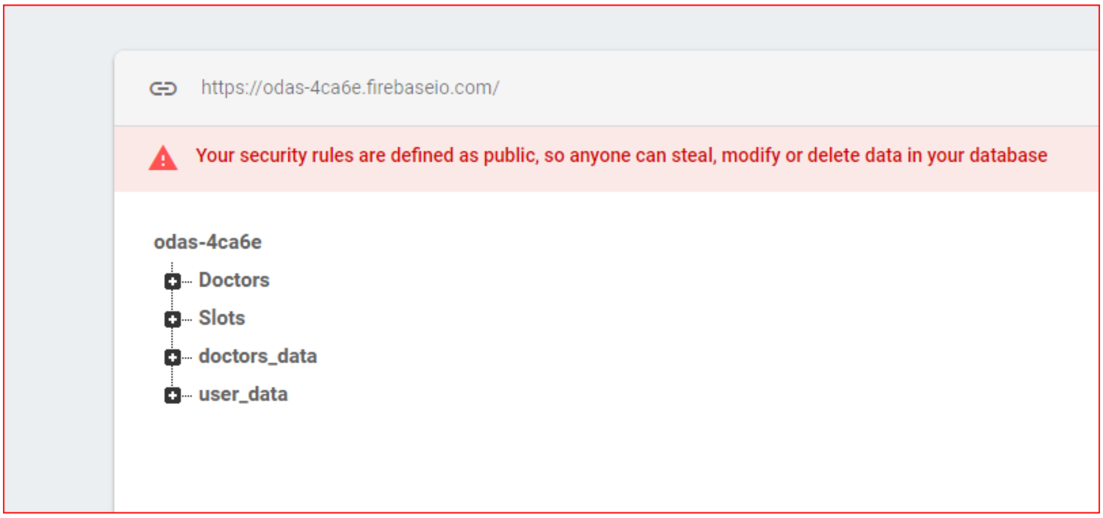
https://odas-4ca6e.firebaseio.com/

Your security rules are defined as public, so anyone can steal, modify or delete data in your database

odas-4ca6e
- Doctors
- Slots
- doctors_data
- user_data

Figure 13 Admin Panel

5.1.2 User log



https://odas-4ca6e.firebaseio.com/

Your security rules are defined as public, so anyone can steal, modify or delete data in your database

odas-4ca6e
- Doctors
- Slots
- doctors_data
- user_data

Figure 14 user log

5.2 Android Application User

We provide android application for our users to interact with our system. So here we have some screens of our applications.

5.2.1 Login And registration screen

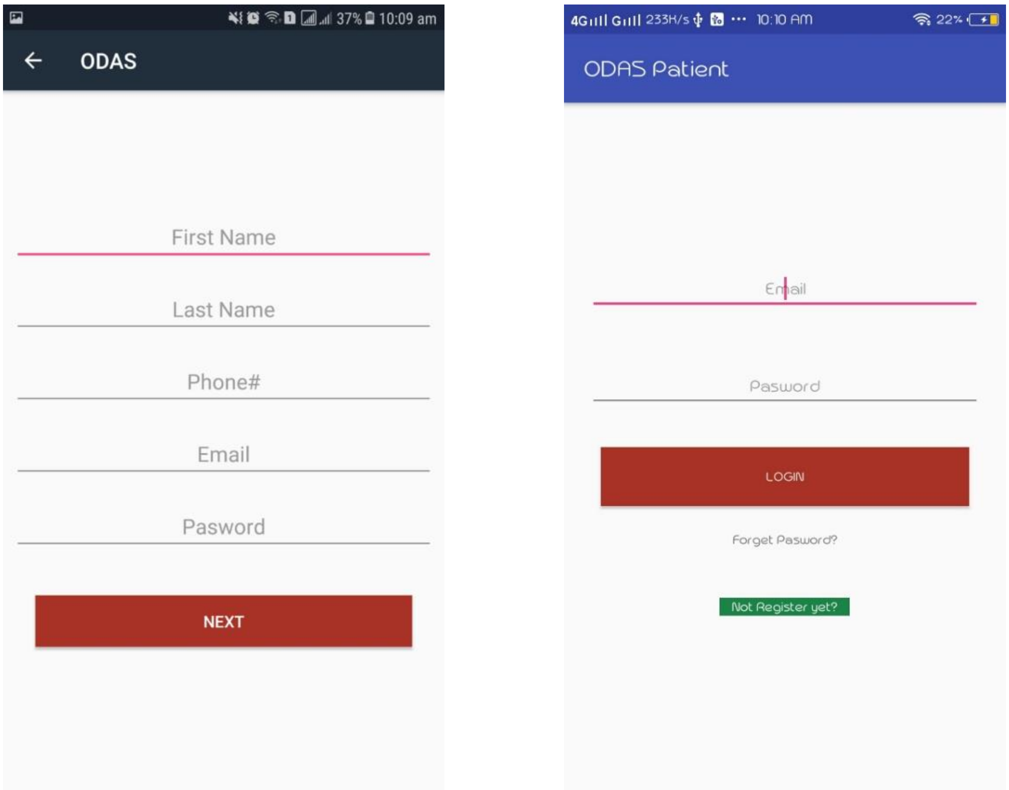


Figure 15 Login and Registration

5.2.2 Main Screen

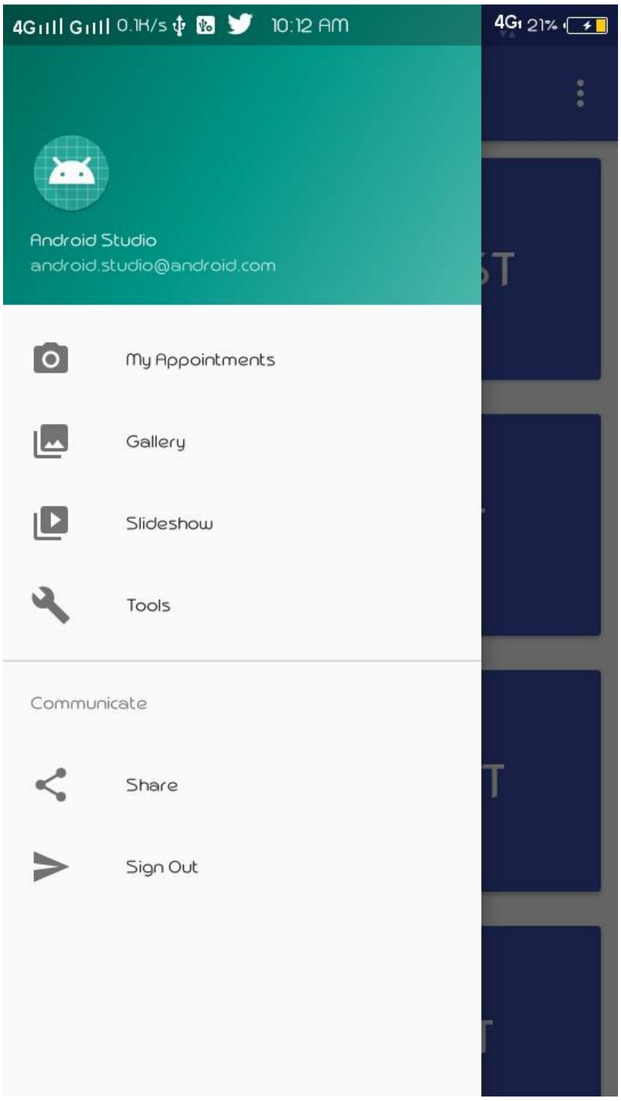


Figure 16 Main Screen

5.2.3 Doctors Screen

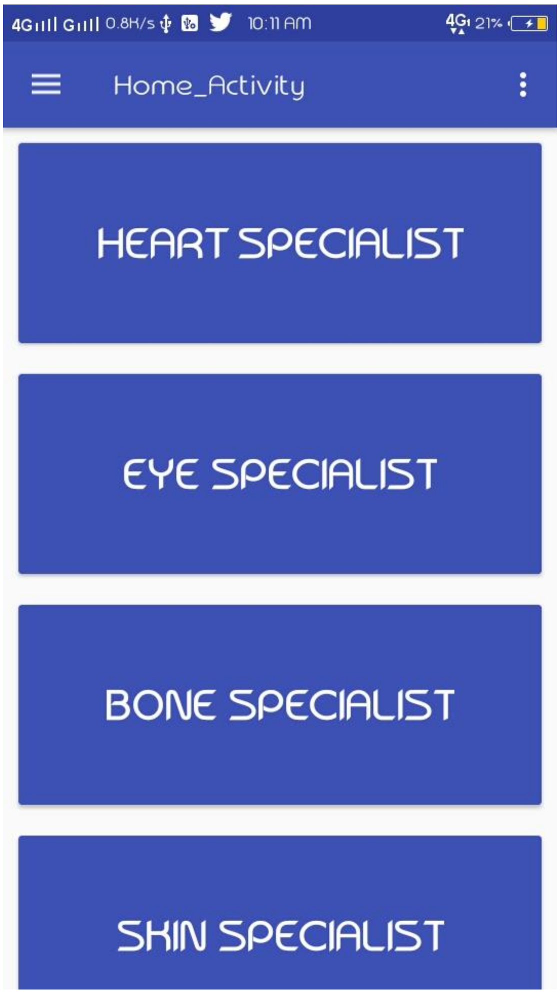
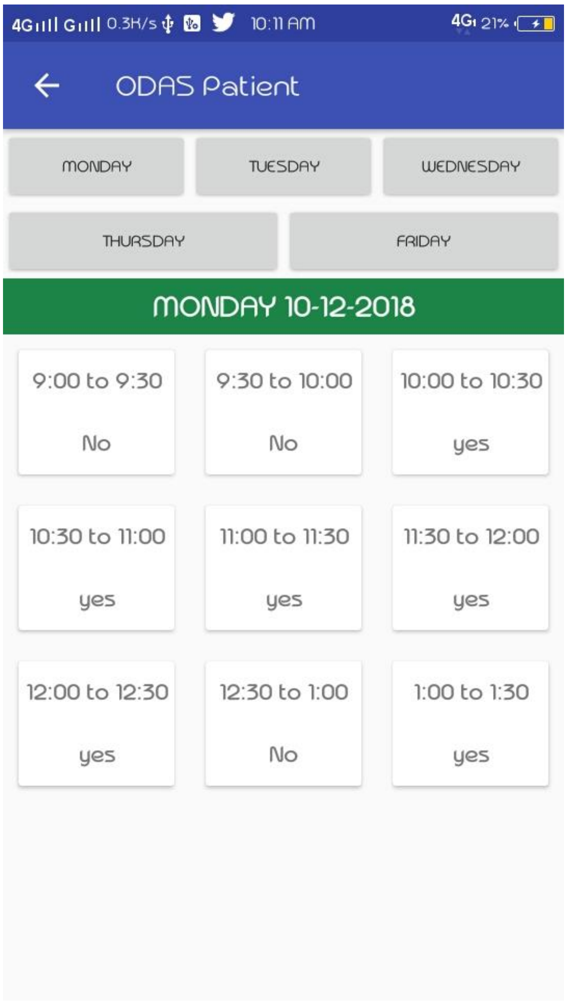


Figure 17 Doctors

5.2.4 Slots Availability



MONDAY 10-12-2018

9:00 to 9:30 No	9:30 to 10:00 No	10:00 to 10:30 yes
10:30 to 11:00 yes	11:00 to 11:30 yes	11:30 to 12:00 yes
12:00 to 12:30 yes	12:30 to 1:00 No	1:00 to 1:30 yes

Figure 18 Slots

5.2.5 Booking Appointment

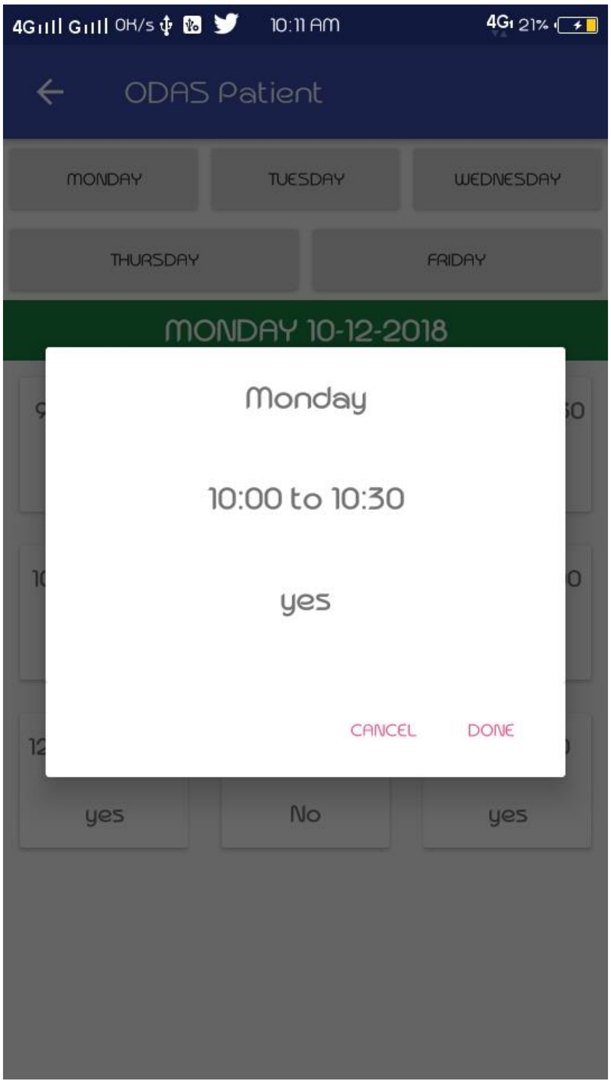
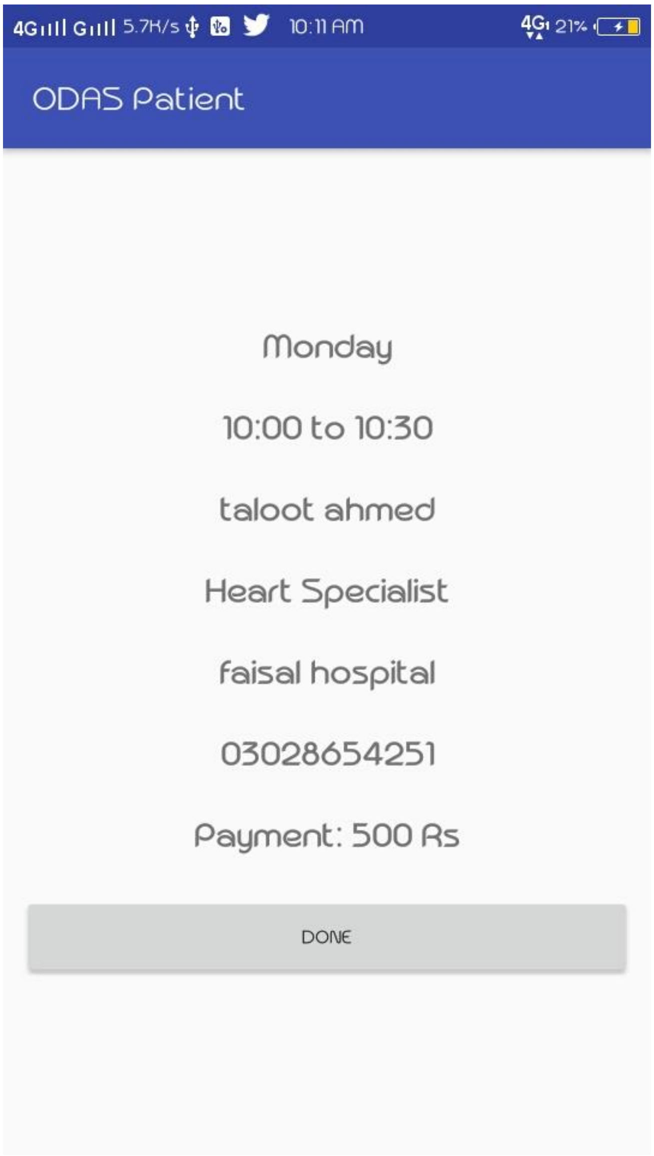


Figure 19 Appointment

5.2.6 Status of Payment



Monday

10:00 to 10:30

taloot ahmed

Heart Specialist

faisal hospital

03028654251

Payment: 500 Rs

Figure 20 Status of Payment

5.2.7 Booked Appointments

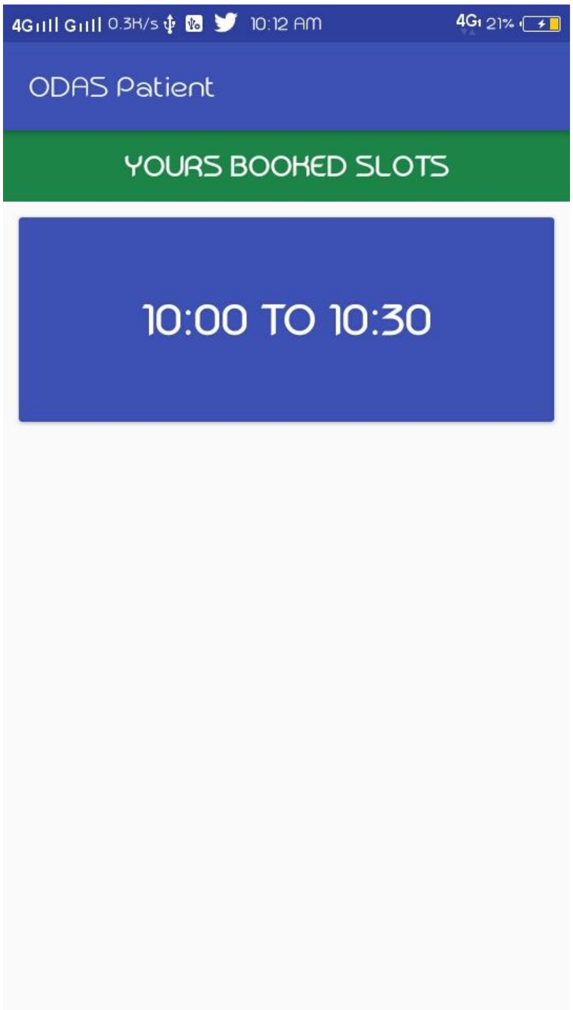


Figure 21 Booked Slots

Chapter 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

ODAS application is made for the automation of the manual Appointment System so that both Doctor and Patient can save their time and sources.

We have provided our users quick and save Appointment system so that they can manage all the work by themselves.

Patients can make Appointment by their required time and according to his disease and can see the payment by himself.

Patients don't have to call someone or to go there by themselves and wait for their turn by using this Application they just have to login to the app and make the appointment for their required time

Doctors can view the appointments that Patient had done and can manage their time and they can see if the payment is done or not.

6.2 Recommendations

6.2.1 Strong internet connection

The users (Doctor and Patient) should use a 3G or 4G facility on their mobiles otherwise the application might lag while getting the data.

6.2.2 Free mobile storage

The worker and customer should at least have 1GB of free space on their phone so that the application can run on the mobile phone without affecting the phones speed.

6.2.3 Work satisfaction level

Customer should ensure that he has received the ordered products or not before paying the bill.

REFERENCES

Journal

[1] R. Anzböck and S. Dustdar, "Modeling and implementing medical Web services," Data & Knowledge Engineering, vol.55, pp. 203-236, 2005.

[2] L. Fleisher, J. Buzaglo, M. Collins, J. Millard, S. M. Miller, B.L. Egleston, N. Solarino, J. Trinastic, D. J. Cegala, A. B.Benson Iii, K. A. Schulman, K. P. Weinfurt, D. Sulmasy, M.A. Diefenbach, and N. J. Meropol, "Using health communication best practices to develop a web-based provider-patient communication aid: The CONNECT(TM) study," Patient Education and Counseling, vol. 71, pp. 378-387, 2008.

[3] Karen Davis, Stephen C. Schoenbaum, and Anne-Marie J.Audet, "A 2020 VISION OF PATIENT-CENTERED PRIMARY CARE," Journal of General Internal Medicine,vol. 20, pp. 953-957, 2005.

Books

- Android Programming: The Big Nerd Ranch Guide (Paperback) by Brian Hardy ,2012
- Hello, Android: Introducing Google's Mobile Development Platform(Paperback) by Ed Burnette,2008
- Professional Android 4 Application Development (Paperback) by Reto Meier,2012