# E-MEDICATION REMINDER APP AND HEALTHCARE

A PROJECT REPORT

Submitted by

# JINI ZACHARIAS

**Reg.No:MGP17MCA-D056**

**to**

the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for

the award of the degree

*of*

## MASTER OF COMPUTER APPLICATIONS



**Department of Computer Applications SAINTGITS COLLEGE OF ENGINEERING**

**Kottukulam Hills, Pathamuttom P.O., Kottayam 686 532**

**May, 2019**

**DECLARATION**

I undersigned hereby declare that the project report **“E-MEDICATION REMINDER APP AND HEALTHCARE”**, submitted for partial fulfillment of the requirements for the award of Degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Asst.Prof. Akhil M Philip**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

## Pathamuttom 20 May, 2019

**JINI ZACHARIAS Reg.No:MGP17MCA-D056**

**DEPARTMENT OF COMPUTER APPLICATIONS SAINTGITS COLLEGE OF ENGINEERING**

Kottukulam Hills, Pathamuttom P.O., Kottayam 686 532.

(*Approved by AICTE and affiliated to APJ Abdul Kalam Technological University)*

# CERTIFICATE

This is to certify that the report entitled “**E-MEDICATION REMINDER APP AND HEALTHCARE**” submitted by **JINI ZACHARIAS , Register Number : MGP17MCA-D056** to the APJ Abdul Kalam Technological University in partial ful- fillment of the requirements for the award of the Degree of Master of Computer Appli- cations is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

## Asst.Prof. Akhil M Philip

Internal Guide

## Asst.Prof. Libin M Joseph

Project Co-ordinator

## Dr. Rajesh K.S

Head of the Department

## Dr. M.D.Mathew

Principal

*Viva-voce held on*:.................................................

## External Examiner 1: External Examiner 2:

**ABSTRACT**

# ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role in expressing my gratitude. To bring something into existence is truly a work of God. I would like to thank God for not letting me down and showing me the silver lining in the dark clouds.

I would like to thank **Dr. M D Mathew**, Principal, Saintgits College of Engineering for his support and encouragement. I convey my heartfelt thanks to **Dr. Rajesh K.S** (Head of the Department of Master of Computer Applications, Saintgits College of En- gineering,) for providing an opportunity for the project presentation. It is my pleasure to express my gratitude to the project coordinator **Asst.Prof.Libin M Joseph**, Depart- ment of Computer Applications, Saintgits College of Engineering whose support and constructive criticism has led to the successful completion of the task.

With the biggest contribution to this report, I would like to thank **Asst.Prof. Akhil M Philip** , Department of Computer Applications who had given me full support in guiding me with stimulating suggestions and encouragement to go ahead in all the time of the this work.

I would also thank my institution and faculty, my family and friends without whom this project would have been a distant reality.

## JINI ZACHARIAS

**CONTENTS**

Contents Page No.

## [ABSTRACT](#_bookmark0) i

[ACKNOWLEDGEMENT ii](#_TOC_250002)

[LIST OF TABLES v](#_TOC_250001)

[LIST OF FIGURES vi](#_TOC_250000)

[Chapter 1: INTRODUCTION](#_bookmark1) 1

[1.1 PROBLEM DEFINITION](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . . 1

* 1. [ABOUT THE ORGANIZATION](#_bookmark3) . . . . . . . . . . . . . . . . . . 1
  2. [OBJECTIVE OF THE PROJECT](#_bookmark4) . . . . . . . . . . . . . . . . . . 2

[Chapter 2: LITERATURE SURVEY](#_bookmark5) 4

[2.1 INITIAL INVESTIGATION](#_bookmark6) . . . . . . . . . . . . . . . . . . . . . 4

[2.2 EXISTING SYSTEM](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . 4

[2.3 PROPOSED SYSTEM](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . 5

[2.3.1 User Classes and Characteristics](#_bookmark9) . . . . . . . . . . . . . . . 5

[2.4 FEASIBILITY STUDY](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . . . . 6

[2.4.1 Technical Feasibility](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . 7

[2.4.2 Economic Feasibility](#_bookmark12) . . . . . . . . . . . . . . . . . . . . . 7

[2.4.3 Operational Feasibility](#_bookmark13) . . . . . . . . . . . . . . . . . . . . 7

[2.4.4 Behavioral Feasibility](#_bookmark14) . . . . . . . . . . . . . . . . . . . . . 8

[Chapter 3: SYSTEM ANALYSIS AND DESIGN](#_bookmark15) 9

* 1. [SOFTWARE REQUIREMENT SPECIFICATION](#_bookmark16) . . . . . . . . . 9
     1. [FUNCTIONAL REQUIREMENTS](#_bookmark17) . . . . . . . . . . . . . 10
     2. [NON-FUNCTIONAL REQUIREMENTS](#_bookmark18) . . . . . . . . . . 11

[3.2 UML DIAGRAM](#_bookmark19) . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

[3.2.1 Usercase Diagram](#_bookmark20) . . . . . . . . . . . . . . . . . . . . . . . 13

[3.2.2 Sequence Diagram](#_bookmark23) . . . . . . . . . . . . . . . . . . . . . . 15

[3.3 SYSTEM DESIGN](#_bookmark25) . . . . . . . . . . . . . . . . . . . . . . . . . . 17

[3.3.1 Input Design](#_bookmark29) . . . . . . . . . . . . . . . . . . . . . . . . . 20

[3.3.2 Output Design](#_bookmark30) . . . . . . . . . . . . . . . . . . . . . . . . . 21

[3.3.3 Database Design](#_bookmark31) . . . . . . . . . . . . . . . . . . . . . . . 21

[3.4 TOOLS AND PLATFORMS](#_bookmark40) . . . . . . . . . . . . . . . . . . . . . 28

[3.4.1 Android Studio](#_bookmark41) . . . . . . . . . . . . . . . . . . . . . . . . 28

[3.4.2 SQLite Database](#_bookmark44) . . . . . . . . . . . . . . . . . . . . . . . 35

[3.4.3 Wamp Server](#_bookmark45) . . . . . . . . . . . . . . . . . . . . . . . . . 36

[3.4.4 Java Script](#_bookmark47) . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

[Chapter 4: SYSTEM TESTING](#_bookmark48) 41

[4.1 UNIT TESTING](#_bookmark49) . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

[4.2 INTEGRATION TESTING](#_bookmark50) . . . . . . . . . . . . . . . . . . . . . . 42

[4.3 SYSTEM TESTING](#_bookmark51) . . . . . . . . . . . . . . . . . . . . . . . . . 42

[4.4 USER ACCEPTANCE TESTING](#_bookmark52) . . . . . . . . . . . . . . . . . . 43

|  |  |  |
| --- | --- | --- |
| [**Chapter 5:**](#_bookmark54) | [**SYSTEM IMPLEMENTATION**](#_bookmark54) | **44** |
| [**Chapter 6:**](#_bookmark55) | [**CONCLUSION**](#_bookmark55) | **45** |
| [**Chapter 7:**](#_bookmark56) | [**REFERENCES**](#_bookmark56) | **46** |
| [**Chapter 8:**](#_bookmark57) | [**APPENDENIX**](#_bookmark57) | **47** |

[8.1 Screen Shots Input Form,Output Forms](#_bookmark58) . . . . . . . . . . . . . . . 47

[8.2 Sample Code](#_bookmark77) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

# LIST OF TABLES

No. Title Page No.

[3.1 USERTYPE](#_bookmark32) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

[3.2 USERLOGINTABLE](#_bookmark33) . . . . . . . . . . . . . . . . . . . . . . . . . 25

[3.3 REFER](#_bookmark34) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

[3.4 MEDICINE NOTIFY](#_bookmark35) . . . . . . . . . . . . . . . . . . . . . . . . . 26

[3.5 APPOINTMENTS](#_bookmark36) . . . . . . . . . . . . . . . . . . . . . . . . . . 26

[3.6 QUERIES](#_bookmark37) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

[3.7 REGISTER DEVICE](#_bookmark38) . . . . . . . . . . . . . . . . . . . . . . . . . 27

[3.8 MED TABLE](#_bookmark39) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

# LIST OF FIGURES

No. Title Page No.

[3.1 Actor](#_bookmark21) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

[3.2 Use case](#_bookmark22) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

[3.3 E-Medication Reminder App and Healthcare system usecase Diagram](#_bookmark24) 16

[3.4 CONTEXT DIAGRAM](#_bookmark26) . . . . . . . . . . . . . . . . . . . . . . . . 18

[3.5 LEVEL 1](#_bookmark27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

[3.6 LEVEL 2](#_bookmark28) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

[3.7 Android Project View](#_bookmark42) . . . . . . . . . . . . . . . . . . . . . . . . . 29

[3.8 Project Structure](#_bookmark43) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

[3.9 MySQL Configuration](#_bookmark46) . . . . . . . . . . . . . . . . . . . . . . . . 38

[4.1 Test case using POSTMAN](#_bookmark53) . . . . . . . . . . . . . . . . . . . . . . 43

[8.1 LOGIN PAGE](#_bookmark59) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

[8.2 DOCTOR HOMEPAGE](#_bookmark60) . . . . . . . . . . . . . . . . . . . . . . . . 47

[8.3 ACCESSING PATIENT CHART](#_bookmark61) . . . . . . . . . . . . . . . . . . . 48

* 1. [E-PRESCRIPTION-PATIENT MEDICATION HISTORY](#_bookmark62) . . . . . . 48
  2. [ADDING NEW MEDICATION PLAN](#_bookmark63) . . . . . . . . . . . . . . . 49

[8.6 ADMIN HOME PAGE](#_bookmark64) . . . . . . . . . . . . . . . . . . . . . . . . 49

[8.7 CONSULTATION HISTORY](#_bookmark65) . . . . . . . . . . . . . . . . . . . . . 50

* 1. [VIEW,ADD,DELETE,EDIT PATIENT DETAILS](#_bookmark66) . . . . . . . . . . 50
  2. [VIEW,ADD,DELETE,EDIT DOCTORS’ DETAILS](#_bookmark67) . . . . . . . . 51
  3. [APPROVAL/DISAPPROVAL OF APPOINTMENT REQUESTS](#_bookmark68) . . 51
  4. [SENDING DAILY HEALTH TIPS USING FIREBASE](#_bookmark69) . . . . . . . 52
  5. [E-REMINDER APP LOGIN PAGE](#_bookmark70) . . . . . . . . . . . . . . . . . 52

[8.13 REGISTRATION FORM](#_bookmark71) . . . . . . . . . . . . . . . . . . . . . . . 53

[8.15 HOME PAGE](#_bookmark72) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

[8.14 NAVIGATION DRAWER](#_bookmark73) . . . . . . . . . . . . . . . . . . . . . . 54

[8.17 E-PRESCRIPTION](#_bookmark74) . . . . . . . . . . . . . . . . . . . . . . . . . . 54

[8.16 FIX APPOINTMENT](#_bookmark75) . . . . . . . . . . . . . . . . . . . . . . . . . 55

[8.18 NOTIFICATIONS](#_bookmark76) . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

# CHAPTER 1 INTRODUCTION

# PROBLEM DEFINITION

InvestoAvenue is a specialist wealth handling firm that is focused towards personalized financial advisory, thus catering to the asset management requirements of its clients. Financial advices provided by InvestoAvenue is based on understanding the client’s present and future cash flows and financial objectives.

The application should incorporate a user-friendly design and help InvestoAvenue’s business by allowing clients to provide their financial details. Clients should be allowed to register and create their own profiles in order to seek financial guidance. The application should employ an analytical algorithm, that would suggest the best investment and financial plans, as per the details provided by the clients. Payment should be accepted categorically based on the services provided..

# ABOUT THE ORGANIZATION

The college was founded by a group of well known academics.They are pioneering educators,having unmatched experience in the field of education with a belief that the continuous search for knowledge is the sole path to success.The Primary focus of the institution is to expose the young minds to be world of technology,instilling in them confidence and fortitude to face new challenges that enables them to excel in their cho- sen field.The college inculcates the development of all facets of the mind culminating in an intellectual and balanced personality.Our team of dedicated and caring faculty strives to widen the students horizon of learning thereby achieving excellent results for every student.

SAINTGITS college of Engineering,right from inception,has been maintaining high levels of standard in academic and extra curricular realms of activities.We offer BTECH degree courses in 6 engineering disciplines,and Masters Degree courses in Engineer- ing,Computer Application and Business Administration. In the short span of a decade of its existence and among the six batches of students that have graduated,the college bagged several university ranks and has a remarkably high percentage of pass.The stu- dents of first batch of MCA bagged the first two ranks in the university.The college is also the venue of national and state level seminars and symposiums and has emerged as the hub of technical education in the state.The placement scenario is also quite com- mendable,with several premier industries visiting SAINTGITS college of engineering for placement and recruitment.

# OBJECTIVE OF THE PROJECT

To enable Technology access and offer affordable healthcare to the population and to enhance the safety and quality of prescribing process.

E-Medication Reminder App And Healthcare is an Android application whose ob- jective is to remind the patients of their dosage timings through Notification system so that they can stay fit and healthy. Through E-Medication Reminder App And Healthcare,out- patients can continued to be connected with the Hospital,they can send their queries andget responses in real time ,Fix appointments,View their E-prescriptions,Past medication details ,Get daily health related articles etc.This application focusses on the people who forget to take medicines on time. It does not require users to set an alarm along with the fields of date, time and medicine descriptions. The notification system will send a notification regarding the medicines need to take at the correct timings as specified in their E-prescription. The user can activate or deactivate the notification accordingly.The project also focuses on enhancing the quality of prescription processes by introduc- ing E-prescription for the Doctors,making it a hassle-free work.E-Medication reminder System help in decreasing medication dispensing errors and wrong dosages.

The application is designed on Android Studio. It can be helpful in Healthcare sector and can spread health care awareness. It is life-saving, money saving and time saving application which is easy to use and provides a good user interface.

# CHAPTER 2 LITERATURE SURVEY

# INITIAL INVESTIGATION

The initial investigation is conducted in order to gather the problems that are currently faced by existing systems and started addressing the common problems and challenges in everyday lives of patients to have systematic medication management related with prescription.Initially, the only way to adhere to the medication process is through timely notifying the user about the medicine intakes, dosages etc. Many such Medical Re- minder Systems have been developed where a new hardware is required but this project made an attempt to develop a system which is economical, time-saving and supports medication adherence and enhances the quality of medication management process.

The issue for developing this application was how to design an app that will meet the requirements and expectation for the patients since this kind of app was new as well as dealing with patients under medication. Also, the issue was how to create an application that will be easy for use without learning any additional skills since targeted patients might be of different age group. Other important issue was to focus on how to enhance users awareness and knowledge about medication and its management. Therefore, the main focus is based on the mentioned issues.It is divided into sub-problems to well- define the limitations and scope of the project topic.

# EXISTING SYSTEM

# Actually there is no any specified existing system.

# In the existing system the transactions are done only manually but in proposed system we have to computerize all the banking transaction using the software financial management system.

# Problems with existing system

# Lack of security of Data

# More man power

# Time consuming

# Consumes large volume of pare work

# Needs manual calculation

# PROPOSED SYSTEM

A better alternative is to set alarms automatically in the patient app,as per the Doctor’s E-prescription. The feature of the proposed system is to assist patients in their medi- cation process through the timely generated reminders as per the medical prescriptions. Electronic prescription has become widely popular able to enhance safety and qual- ity of prescribing process. Electronic prescription has been defined as the computer- based electronic generation, transmission, and filling of a prescription by replacing the paper-based prescriptions. E prescribing has made it easy for prescriber to electroni- cally send patients prescription information to pharmacy computers, which again has reduced medication errors.

## 2.3.1User Classes and Characteristics

The system contains three users, administrator, Doctor( Health care personnel) and Pa- tient. Role of admin is to register the health care personnel and patient, approval of appointment requests, send responses corresponding to user queries instantly, posting daily health related articles. The Health care personnel adds corresponding patient pre- scription through E-prescription, after their consultation, to the system. Once the pre- scriptions are loaded into the database, they will be accessible to the patient in their app, the system will generate reminders regarding the medicine intake, dosage, medication starting date, end date etc to guide the patient in their medication course.

Through these functionalities the proposed work tries to overcome all the listed disad- vantages of existing systems.

# FEASIBILITY STUDY

During system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the Medical field. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product. While evaluating the existing system, many advantages and disadvantages raised. Analysing the problem thoroughly forms the vital part of the system buddy. Problematic areas are identified and information is collected.

The benefits of this site are users can easily interact and get the services without much complexity. It helps to make it possible that more users can interact with the site at a time. Feasibility study is to determine whether the proposed system is technically, economically and behaviourally feasible in all respects.

The main aim of feasibility study is to evaluate alternative site and propose the most feasible and desirable site for development. If there is no loss for the organization then the proposed system is considered financially feasible. A feasibility study is carried out to select the best system that meets performance requirements.The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

In this scenario, problems are identified. Essential data are being gathered for the existing problems. It is necessary that this analysis familiarizes the designer with ob- jectives, activities, and the function of the organization in which the system is to be implemented. The feasibility study was divided into four:- Technical, Economical, Op- erational and Behavioural. It is summarized below:-

## Technical Feasibility

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs, are identified. While considering the problems of existing system, it is sufficient to imple- ment the new system. The proposed system can be implemented to solve issues in the existing system. It includes the evaluation of and how it meets the proposed system. This system use Android,PHP as front end technology and MYSQL Server,SQLite as back end technology.

## Economic Feasibility

Economic analysis is most frequent used for evaluating of the effectiveness of the candidate system. More commonly known as cost/benefit analysis the procedure is to de- termine the benefit and saving that are expected from a candidate system and compare them with the existing system. Except for the initial capital amount and the amount after each financial year, no other huge amount is needed. The expenses can be handles by any participants. So, the system is economically feasible.

This feasibility involves some questions such as whether the firm can afford to build the system, whether its benefits should substantially exceed its costs, and whether the project has higher priority and profits than other projects that might use the same re- sources. Here there is no problem. This firm has fully equipped hard ware, and fully fledged software, so no need to spend money on these issues. And as the client and the developer are one, there is no further problem in economic issues.

## Operational Feasibility

Methods of processing and presentation are all according to the needs of clients since they can meet all user requirements here. The proposed system will not cause any prob- lem under any circumstances and will work according to the specifications mentioned. Hence the proposed system is operationally feasible. People are inherently resistant to

change and computer has been known to facilitate changes. The system operation is the longest phase in the development life cycle of a system. So, Operational Feasibility should be given much importance. This system has a user-friendly interface. Thus it is easy to handle.

## 2.4.4 Behavioral Feasibility

In today’s world, computer is an inevitable entity. As per the definition of behavior design, many valid points are recognized in this study. This system behavior changes according to different environment. In order to ensure proper authentication and au- thorization and security of sensitive data of the admin or employers, login facilities are provided. These are the main feasibility studies tested in this application.

# CHAPTER 3

**SYSTEM ANALYSIS AND DESIGN**

# SOFTWARE REQUIREMENT SPECIFICATION

”E-Medication Reminder App And Healthcare System” is a mobile application that is going to be built on Google Android platform and the web part in PHP. The App is aimed to empower Patients with Medication reminder facility and personalized health- care features.The Healthcare system assists Doctors in prescription process,viewing pa- tient medication history. It also includes an administrative control center. From this area the administrator can fully manage every aspect of the entire Medical center. He can view the consulted patients information and can add/edit/delete the patients and doctors The approval of appointment requests and replying to user queries, posting health re- lated articles also rests with him. In short, he has total control on all the features of the system.

## Module Description

* + - User Management: The system contains three users, administrator, Health care personnel and Patient. Role of admin is to register the health care personnel and patient. The Health care personnel can add corresponding patient prescrip- tion, after their consultation, to the system. Once the prescription is loaded into the database, it will be accessible through the patient app, the timely generated reminders regarding the medicine intake, dosages to guide the patient in their medication course
    - Medication Reminders: It helps in reminding about the medicines. System can add details of the patients dosage schedules. Through the E-Prescription module, at the time of consultation Doctor adds the description of the medicine, including name, purpose and other related description. Once the user allows the notifica- tions, the app generates timely Alarms following the notifications: Get Notifica- tion module: Once the alarm is set then the user gets the notification. The users can activate or deactivate this accordingly. If he does not require the notification

he can turn off it.

* + - Medication history: Patients can keep track of all his/her medication details.
    - Fix Appointment: User can take direct appointment with the doctor, once the request has been fulfilled by the admin, user gets a confirmation message.
    - Post queries: User can share their queries with the admin and can get responses in real time.
    - E-Prescription module: Electronic prescription has been defined as the computer- based electronic generation, transmission, and filling of a prescription by replac- ing the paper-based prescriptions. E prescribing has made it easy for prescribers to electronically send patients prescription information to pharmacy computers, which again has reduced medication errors. All the information will be saved in the database. This makes any time availability of the patients records
    - Daily Healthcare articles: Using Googles Firebase cloud messaging platform, all the registered patients app receive daily healthcare articles send by the admin.
    - AdminModule: The Admin Can Add, edit, delete and view both the patient and doctor information. Also this module helps in viewing queries, approval of ap- pointment requests made by the patients from their app, replying to the queries etc

## FUNCTIONAL REQUIREMENTS SYSTEM SPECIFICATION

The Overall description of the system- describes the general factors that affect the prod- uct and its requirements.This section does not state specifc requirements.Instead,it pro- vides a background for those requirements,which are defined later in section.and makes them easier to understand.Web based project management systems are designed to man- age and store project information that are used in web-based applications.Different groups of people such as programmers or project managers let by project applications a controlled access to information and automated distribution of information.Mostly

Gantt charts and informal techniques and tools managed projects.Later on project man- agement,tools and techniques were formalized to more professional and modern solu- tions.

## HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any application.As the proposed system is an android application,it can run on any android smart phone that have an internet connection.The basic hardware requirements are Processor : Intel Core i3 CPU

RAM: 4 GB Or More Cache: 512 KB

Hard Disk: 16 GB Hard Disk Recommended Drive: CD/DVD

Keyboard: Logitech Standard 101 Keys Mouse: Logitech

Monitor: LCD color Monitor

Android System Version: 4.4(kitkat)and above Internet Connectivity

**SOFTWARE SPECIFICATION** OPERATING SYSTEM : WINDOWS 7OR LATER FRONT END : Android ,PHP

TOOLS : Android Studio IDE for android,Notepad++ for PHP TECHNOLOGY : Firebase

WEB TECHNOLOGIES : HTML, JAVASCRIPT, CSS

WEB SERVER : Wampserver 2.1

BACK END : MySQL and SQLite database

## NON-FUNCTIONAL REQUIREMENTS PERFORMANCE REQUIREMENTS

For the efficient performance of the application, network must have high bandwidth so

that the task of centralized management does not lead to network jam. Also the hard disk capability must be high so that data can be effectively stored and retrieved.

## SECURITY REQUIREMENTS

Security requirements of this application involves user authentication using username and password so that invalid users are restricted from data access.Mostly mobile phones are to be considered as a personal and private use. Our basic assumptions state that only the owner to whom the smart phone belongs uses it. Since this is related to someones sensitive health-related issues, it must be very confidential. But at the same time, an au- thentication of the user every time when application is opened can significantly reduce the tool usability.

# 3.2UML DIAGRAM

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar JAcobson and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the Object Management Group(OMG) as the standard for mod- elling software development.

UML stands for Unified Modeling Language.UML 2.0 helps extend the original UML specification to cover a wider portion of software development efforts including agile practices. Improved intergration between structural models like class diagrams and be- havior models like activity diagrams. The original UML specified nine diagrams; UML

* 1. brings that number up to 13. The four new diagrams are called: communication diagram, composite diagram, interaction overview diagram and timing diagram. It also renamed statechart diagrams to state machine diagrams , also known as state diagrams.

## Types of UML diagrams

The current UML standards call for 13 different types of diagrams: class, activity, ob- ject, use case, sequence, package, state, component, communication, composite struc- ture, interaction overview, timing and deployment. These diagrams are organized into

two distinct groups: structural diagrams and behavioral or interaction diagrams.

## Structural UML diagrams

* + - Class diagram
    - Package diagram
    - Object diagram
    - Component diagram
    - Composite structure diagram
    - Deployment diagram

## Behavioral UML diagrams

* + - Activity Diagram
    - Sequence diagram
    - Use case diagram
    - State diagram
    - Communication diagram
    - Interaction overview diagram
    - Timing diagram

## Usercase Diagram

To model a system the most important aspect is capture the dynamic behaviour. To modify a bit in details, dynamic behaviour of the system when it is running or operating. So only behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model

dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagram consists of actors, use case and their relationships. The diagram is used to model the system of an application. A single use case diagram captures a particular functionality of a system.

Use case Diagram objects:

* + - * Actor
      * Use case
      * System
      * Package

## Actor

Actor is a use case diagram in an entity that performs a role in one given system. This could be a person, organization or an external system usually drawn like skeleton. **Use case**



Figure 3.1: Actor

A use case represents a function or an action within the system. Its drawn as an oval and named with the function.



Figure 3.2: Use case

## System

System is used to define the scope of the use case and drawn as a rectangle. This is an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

## Package

Package is another optional element that is extremely useful in complex diagrams.

Similar to use class diagrams, packages are used to group together use cases.

## Sequence Diagram

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system.Time is represented in the vertical direction showing the sequence of interaction of the header elements.

Sequence Diagrams are used primarily to design, document and validate the architec-

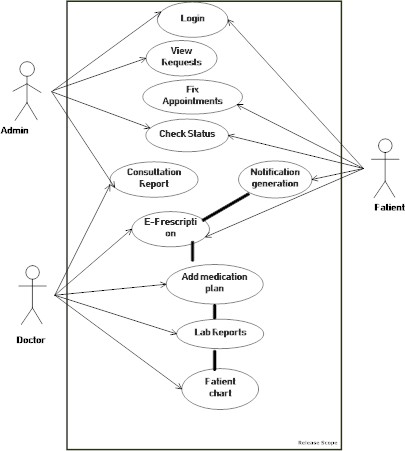
ture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

Figure 3.3: E-Medication Reminder App and Healthcare system usecase Diagram

Although UML sequence diagrams are typically used to describe object-oriented software systems, they are also extremely useful as system engineering tools to design system architectures in business process, as message sequence charts and call flows for telecoms or wireless system design, and for protocol stack design and analysis.

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the

objects needed to carry out the functionality of the scenario. Sequence Diagrams are typically associated with use case realizations in the logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenar- ios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

# SYSTEM DESIGN

Design is the abstraction of a solution; it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to cre- ate the implementation. The design is a solution, the transition of requirements in two ways of meeting them. The design will determine the success of the system. Based on the proposed system objectives, the major modules are identified and the operations to be carried out are determined. In the design phase of the system the user interaction screen, database tables,inputs,outputs and screen are designed. The database tables are designed by using all the necessary fields in a compact manner. The redundancy and duplication of fields are avoided.

Systems design involves first logical design and the n physical construction of the system. After logical design, a detailed specification of the system, which describes the inputs,outputs,files are developed. During the design phase of system the following factors are considered.

The important of software design can be stated with a single word quality. Design is placed where quality is fostered in software development. Design is the only way whose requirements are actually translated into a finished software product or system.

* Data Floors:-The movement of data into, around and out of the system
* Data Source:-Temporary and permanent collections of data
* Processors:-Activities to accept, manipulate and deliver data and information
* Procedures:-Methods and routines to achieve the intended results

## DATA FLOW DIAGRAM

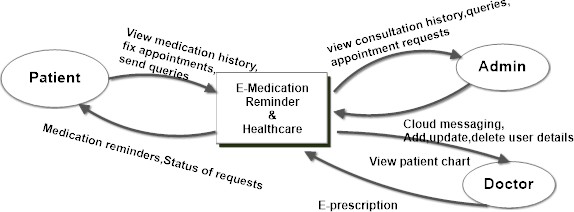


Figure 3.4: CONTEXT DIAGRAM

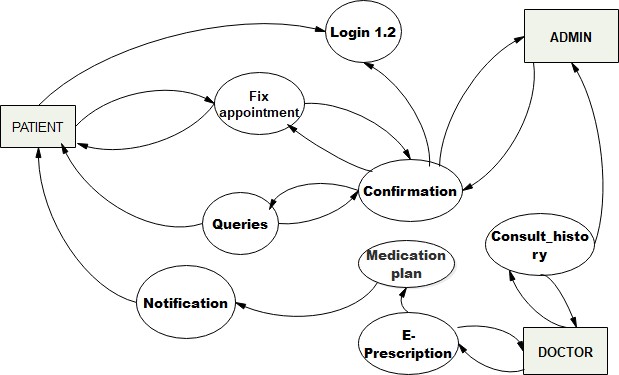


Figure 3.5: LEVEL 1

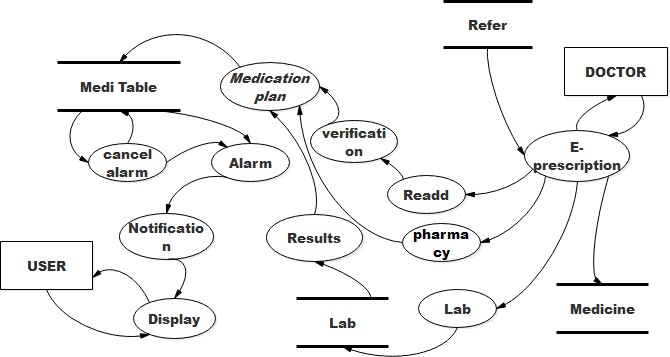


Figure 3.6: LEVEL 2

## Input Design

The input is the set of values that is provided by the user to the system.The input design must enable the user to provide the error free input to the system for efficient pro- cessing.The data is fed into the system using simple interactive xml pages.The pages have been supplied with messages so that user can enter data without facing any diffi- culty.The data is validated wherever it requires in the project

The main objectives of the input design are as follows:

* + - * Produce effective method of input
      * Achieve high level accuracy
      * Ensure that the input is acceptable and understood by the user The different types of input data handled by the system are:

## External

They are the primary inputs to the system.The external input is what the user supplies to the system.The user can give different types of external inputs in this project such as registration details,login details etc.

## Internal

When the external inputs are obtained from the user,these inputs are transfered to the system as messages.These messages are captured and handled as input for further pro- cessing.

In this project the input design is done with Android and PHP codes.The external in- puts are the data given to the system by the user.The neccessary external inputs are given to the system by Graphical User Interface(GUI)technology.The GUI system applied to this project enables the user to avoid error and confusion arises while entering the input.

## Output Design

The primary consideration in the design of all output is the information requirement and other objective of the users. It is the most important and direct source of information to the user. A major form of output is a hard copy. Print out should be designed around the output requirements of the user. Each output should be given a specific name or title. The output data is displayed on the visual display unit and output can be redirected to printers and or sorted in a file for later use.

## Database Design

Database is a design to manage large bodies of information. The management of data involves both the definition of structures for the storage information. In addition, the database system must provide for the safety of the information solved, despite system crashes or due to attempts at unauthorized access. For developing an efficient database we have to fulfil certain condition such as controlled redundancy

* + - * Defining the data
      * Inputting the data
      * Locating the data
      * Accessing the data
      * Communicating the data
      * Revising the data

**OBJECTIVES OF DATABASE** In the database design, several objectives are consid- ered such as

* + - * Control of data Integrity
      * Ease of use
      * Control of redundancy
      * Control of security
      * Data independence(Logical and physical)
      * Data storage protection(Record level and Table level)
      * System performance
      * System functions
      * System compatibility

For achieving the above mentioned criterias we have to make use various features that are available with the RDBMS by enforcing integrity constraints, we can ensure data integrity and reduce data inconsistency to a great extent. Recovery from failures can overcome using backup facilities. By using table level as well as row level locking facilities , wecan avoid concurrent access normalize. Another important features of RDBMS is the logical and physical data independence. In addition to security mecha- nism provided by RDBMS, we have provided system password to near system

**NORMALIZATION:**Normalization is the term obtained from the Latin word NORMA which means that square used by the carpenter .Normalization is the process of sim- plifying the relationship between data elements in a record, through normalization a collection of data I a record structure is replaced by successive record structures that

are simpler and can be managed efficiently. While designing the database, we have to implement the concept of normalization to avoid data redundancy in the database. Normalization is carried out for four reasons.

* + - * To structure the data so that any pertinent relationship between entities can be represented.
      * To permit simple retrieval of data in response to query and reports required.
      * To simplify data maintenance procedures such as insertion, deletion and updating.
      * To reduce the need to be structure or reorganize data with new application re- quirements arise.

The major normalization strategies are

* + - * First Normal Form
      * Second Normal Form
      * Third Normal Form
      * Boyce/Codd Normal Form(BCNF)

**FIRST NORMAL FORM:** First Normal Form is achieved when all repeating groups in a record are removed, so that record is of fixed length. A repeating group, reoccurrence of a data item or group of data item within a record indicates another relation.

**SECOND NORMAL FORM:** Second Normal Form is achieved when a record is in first normal form and each item in the record is functionally depend on the primary key for identification. In other words, analyst seeks functional dependency. A data item is functionally dependent of its value is uniquely associated with a specific data item is functionally dependent of its value is uniquely associated with a specific item. To achieve second normal form every column in a table that is not dependent on the primary key of the record should be removed and used to form a separate relation.

**THIRD NORMAL FORM:** Third Normal Form is achieved when all transitive dependencies are removed from a record. That is, if A is functionally dependent on B and B is functionally dependent on C, then A is functionally dependent on C.

**BOYCE/CODD NORMAL FORM(BCNF):** BCNF is often used to distinguish the new 3NF from the old. An attribute possible composite is called as determinant. If other attributes are fully functionally determined this attribute(or on which some other attribute is fully functionally dependent on this attribute).A table is in BCNF , if every determinant is a candidate key. To achieve a table is in BCNF, remove fields which are

fully functionally dependent on a determinant, which is not act as a candidate key.

Table 3.1: USERTYPE

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| Patient id | Varchar(20) | Primary key | It is used to store the unique id of patients |
| Name | Varchar(20) | Not  Null | Should not be null. |
| DOB | Date | Foreign  Key | Should not be null. |
| Age | Int | Not  Null | Should not be null. |
| Gender | Varchar(20) | Not  Null | Should not be null. |
| Addrs | Varchar(100) | Not  Null | Should not be null. |
| User email | Varchar(40) | Not  Null | Should not be null. |
| ph | Int | Not  Null | Should not be null. |
| User password | Varchar(10) | Not  Null | Should not be null. |
| Reg date | Date | Not  Null | Should not be null. |

Table 3.2: USERLOGINTABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| User id | Varchar(20) | PRIMARY KEY | It is used to  store the unique id |
| UserType | Varchar(10) | Not Null | Should not be null. |
| First name | Varchar(30) | Not Null | Should not be null. |
| Dob | Date | Not Null | Should not be null. |
| gender | Varchar(30) | Not Null | Should not be null. |
| Usertype | Varchar(30) | Not Null | Should not be null. |
| Designation | Varchar(50) | Not Null | Should not be null. |
| Mobile | Int | Not Null | Should not be null. |
| User email | Varchar(30) | Not Null | Should not be null. |
| Registration date | Date | Not Null | Should not be null. |
| Password | Varchar(20) | Not Null | Should not be null. |

Table 3.3: REFER

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| eprescno | Varchar(20) | Primary Key | It is used to store the unique id |
| P id | Varchar2(40) | Foreign Key | Refers to the patient id |
| Ep date | Date | Not Null | Should not be null. |

Table 3.4: MEDICINE NOTIFY

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA**  **TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| MedID | Int | Auto Increment | It is used to store the indices |
| Medname | Varchar(100) | Not null | Should not be null. |
| Pt id | Varchar(20) | Foreign Key | Should not be null. |
| EP No | Varchar(40) | Foreign Key | Refers to the eprescno |
| Dosage | Int | Not Null | Should not be null. |
| Mrng | Varchar(10) | Not Null | Should not be null. |
| Noon | Varchar(10) | Not Null | Should not be null. |
| Evng | Varchar(10) | Not Null | Should not be null. |
| End dt | Date | Not Null | Should not be null. |
| Start dt | Date | Not Null | Should not be null. |

Table 3.5: APPOINTMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA**  **TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| ID | Int(11) | Primary Key | It is used to store the unique id |
| qDT | Date | Not Null | Should not be null. |
| name | Varchar(30) | Foreign Key | Refers to the username |
| Save date | date | Not Null | Should not be null. |
| Docname | Varchar(30) | Not Null | Should not be null. |
| status | Varchar(10) | Not  Null | Should not be null. |

Table 3.6: QUERIES

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| Msg no | Int(11) | Primary Key | Stores the unique id |
| qDate | Date | Not Null | Should not be null. |
| P id | Varchar(10) | Foreign Key | Refers  to the patient id |
| Name | Varchar(30) | Not Null | Should not be null. |
| Ques | Varchar(100) | Not Null | Should not be null. |
| Reply | Varchar(100) | Not Null | Should not be null. |

Table 3.7: REGISTER DEVICE

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA**  **TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| Reg id | Int(11) | Primary  Key | Stores  the unique id |
| Email | Varchar(20) | Not  Null | Should not be null. |
| Token | Text | Not  Null | Should not be null. |

Table 3.8: MED TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **CONSTRAINT** | **DESCRIPTION** |
| Med id | Int(11) | Primary Key | Stores  the unique id |
| Ep date | Varchar | Not Null | Should not be null. |
| Medname | Varchar | Not Null | Should not be null. |
| Dosage | Varchar | Not Null | Should not be null. |
| Start date | Varchar | Not Null | Should not be null. |
| End date | Varchar | Not Null | Should not be null. |

# TOOLS AND PLATFORMS

## Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ’s powerful code editor and developer tools, Android Studio offers even more features that enhance your productiv- ity when building Android apps, such as:

* + - * A flexible Gradle-based build system
      * A fast and feature-rich emulator
      * A unified environment where you can develop for all Android devices
      * Instant Run to push changes to your running app without building a new APK
      * Code templates and GitHub integration to help you build common app features and import sample code
      * Extensive testing tools and frameworks
      * Lint tools to catch performance, usability, version compatibility, and other prob- lems
      * C++ and NDK support
      * Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

## Project Structure

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

* + - * Android app modules
      * Library modules
      * Google App Engine modules

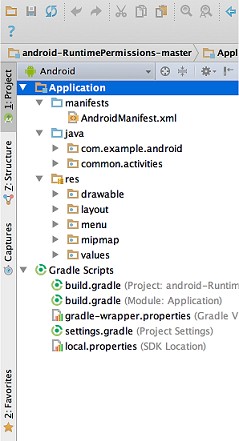
By default, Android Studio displays your project files in the Android project view, as shown in figure 3.7. This view is organized by modules to provide quick access to your project’s key source files.

Figure 3.7: Android Project View

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

* + - * manifests: Contains the AndroidManifest.xml file.
      * java: Contains the Java source code files, including JUnit test code.
      * res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the Project dropdown (in

figure 3.8, it’s showing asAndroid).

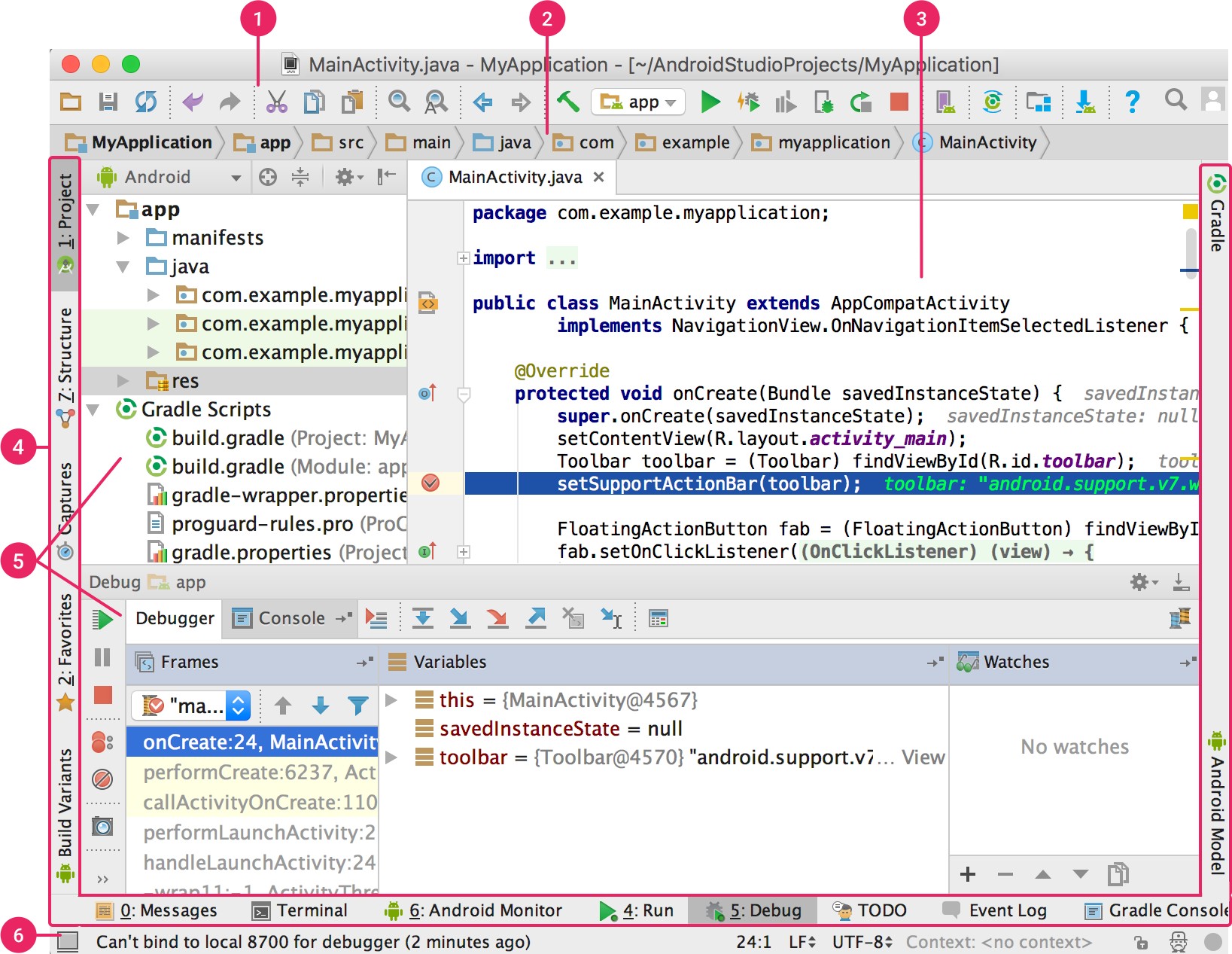
You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.The User Interface The Android Studio main window is made up of several logical areas identified in figure 3.

Figure 3.8: Project Structure

1. The toolbar lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The navigation bar helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
3. The editor window is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The tool window bar runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
5. The tool windows give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. The status bar displays the status of your project and the IDE itself, as well as any warnings or messages.

Version Control Basics

Android Studio supports a variety of version control systems (VCSs), including Git, GitHub, CVS, Mercurial, Subversion, and Google Cloud Source Repositories. After importing your app into Android Studio, use the Android Studio VCS menu options to enable VCS support for the desired version control system, create a repository, import the new files into version control, and perform other version control operations: 1. From the Android Studio VCS menu, click Enable Version Control Integration. 2. From the drop-down menu, select a version control system to associate with the project root, and then click OK. The VCS menu now displays a number of version control options based on the system you selected. Gradle Build System

Android Studio uses Gradle as the foundation of the build system, with more Android- specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the com- mand line. You can use the features of the build system to do the following:

* + - * Customize, configure, and extend the build process.
      * Create multiple APKs for your app, with different features using the same project and modules.
      * Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app’s core source files. Android Studio build files are named build.gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire

project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Build Variants

The build system can help you create different versions of the same application from a single project. This is useful when you have both a free version and a paid version of your app, or if you want to distribute multiple APKs for different device configurations on Google Play.

## Projects Overview

A project in Android Studio contains everything that defines your workspace for an app, from source code and assets, to test code and build configurations. When you start a new project, Android Studio creates the necessary structure for all your files and makes them visible in the Project window on the left side of the IDE click

*V iew* ⇒ Tool Windows ⇒ Project

.

Modules

A module is a collection of source files and build settings that allow you to divide your project into discrete units of functionality. Your project can have one or many modules and one module may use another module as a dependency. Each module can be independently built, tested, and debugged. Additional modules are often useful when creating code libraries within your own project or when you want to create different sets of code and resources for different device types, such as phones and wearables, but keep all the files scoped within the same project and share some code. You can add a new module to your project by clicking

*F ile* ⇒ New ⇒ New Module Android Studio offers a few distinct types of module.

## Library module

Provides a container for your reusable code, which you can use as a dependency in other app modules or import into other projects. Structurally, a library module is the same as an app module, but when built, it creates a code archive file instead of an APK, so it can’t be installed on a device. In the Create New Module window, Android Studio offers the following library modules Library - This type of library can contain all file types supported in an Android project,including source code,resources and manifest files.The build result is an Android Archive(AAR) file that you can add as a dependency for your Android app modules. Java Library - This type of library can contain only Java source files.This build result is an Java Archive(JAR) file that you can add as a dependency for your Android app modules or other Java projects. Google Cloud module

Provides a container for your Google Cloud backend code. This module has the required code and dependencies for a Java App Engine backend that uses simple HTTP, Cloud Endpoints, and Cloud Messaging to connect to your app. You can develop your backend to provide cloud services your app needs. Using Android Studio to develop your Google Cloud module lets you manage app code and backend code in the same project. You can also run and test your backend code locally, and use Android Studio to deploy your Google Cloud module.

## Project Files

By default, Android Studio displays your project files in the Android view. This view does not reflect the actual file hierarchy on disk, but is organized by modules and file types to simplify navigation between key source files of your project, hiding certain files or directories that are not commonly used.

Within each Android app module, files are shown in the following groups: mani- fests:Contains the AndroidManifest.xml file.

java:Contains the Java source code files, separated by package names, including JUnit test code.

res:Contains all non-code resources, such as XML layouts, UI strings, and bitmap im- ages, divided into corresponding sub-directories. For more information about all possi- ble resource types.

The most important components of Project view are the following:

module-name/

* + - * build/ - Contains build outputs.
      * libs/ - Contains private libraries.
      * src/ - Contains all code and resource files for the module in the following subdi- rectories:
      * androidTest/ - Contains code for instrumentation tests that run on an Android device. For more information.
      * main/ - Contains the ”main” sourceset files: the Android code and resources shared by all build variants (files for other build variants reside in sibling di- rectories, such as src/debug/ for the debug build type).
      * AndroidManifest.xml - Describes the nature of the application and each of its components.
      * java/ - Contains Java code sources.
      * jni/ - Contains native code using the Java Native Interface (JNI).
      * gen/ - Contains the Java files generated by Android Studio, such as your R.java file and interfaces created from AIDL files.
      * res/ - Contains application resources, such as drawable files, layout files, and UI string.
      * assets/ - Contains file that should be compiled into an .apk file as-is. You can navigate this directory in the same way as a typical file system using URIs and read files as a stream of bytes using the Asset Manager.
      * test/ - Contains code for local tests that run on your host JVM.
      * build.gradle (module) - This defines the module-specific build configurations.
      * build.gradle (project) - This defines your build configuration that apply to all mod- ules. This file is integral to the project, so you should maintain them in revision control with all other source code.

## Project Structure Settings

To change various settings for your Android Studio project, open the Project Struc- ture dialog by clicking File → Project Structure. It contains the following sections: SDK Location: Sets the location of the JDK, Android SDK, and Android NDK that your project uses.

Project: Sets the version for Gradle and the Android plugin for Gradle, and the reposi- tory location name.

Developer Services: Contains settings for Android Studio add-in components from Google or other third parties. See Developer Services, below.

Modules: Allows you to edit module-specific build configurations, including the target and minimum SDK, the app signature, and library dependencies.

## SQLite Database

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. SQLite is

an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its stor- age files directly.

Why SQLite?

* + - * SQLite does not require a separate server process or system to operate (server- less).
      * SQLite comes with zero-configuration, which means no setup or administration needed.
      * A complete SQLite database is stored in a single cross-platform disk file.
      * SQLite is very small and light weight, less than 400KB fully configured or less

than 250KB with optional features omitted.

* + - * SQLite is self-contained, which means no external dependencies.
      * SQLite transactions are fully ACID-compliant, allowing safe access from multi- ple processes or threads.
      * SQLite supports most of the query language features found in SQL92 (SQL2) standard.
      * SQLite is written in ANSI-C and provides simple and easy-to-use API.
      * SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

## SQLite A Brief History

* + - * 2000 - D. Richard Hipp designed SQLite for the purpose of no administration required for operating a program.
      * 2000 - In August, SQLite 1.0 released with GNU Database Manager.
      * 2011 - Hipp announced to add UNQl interface to SQLite DB and to develop UNQLite (Document oriented database).

## Wamp Server

WAMP is a Windows OS based program that installs and configures Apache web server, MySQL database server, PHP scripting language, phpMyAdmin (to manage MySQL databases), and SQLite Manager (to manage SQLite databases). WAMP is designed to offer an easy way to install Apache, PHP and MySQL package with an easy to use installation program instead of having to install and configure everything yourself. WAMP is so easy because once it is installed it is ready to go. You dont have to do any additional configuring or tweaking of any configuration files to get it running.There are usually two reasons why someone chooses to install WAMP. They are looking to install WAMP for development purposes or to run their own server.

## Wamp Server Contains

1-PHP Admin

Allows you to change or add users and for making new databases phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. The most frequently used operations are supported by the user interface (managing databases, tables, fields, relations, indexes, users, permissions, etc), while you still have the ability to directly execute any SQL statement.

## Features

* + - * Intuitive web interfaceSupport for most MySQL features:
      * browse and drop databases, tables, views, fields and indexes
      * create, copy, drop, rename and alter databases, tables, fields and indexes
      * maintenance server, databases and tables, with proposals on server configuration
      * execute, edit and bookmark any SQL-statement, even batch-queries
      * manage MySQL users and privileges
      * manage stored procedures and triggers
      * Import data from CSV and SQL: Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LA- TEX and others
      * Administering multiple servers
      * Creating PDF graphics of your database layout
      * Creating complex queries using Query-by-example (QBE)
      * Searching globally in a database or a subset of it
      * Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link

2-Apache

Apache Server deals with Server Side Includes, usually called simply SSI. 3-SQL Server and Database System

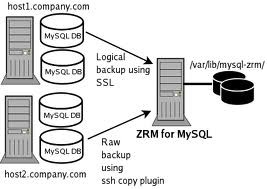
SQL Server is a relational database management system from Microsoft that’s de- signed for the enterprise environment. SQL Server runs on T-SQL (Transact -SQL), a set of programming extensions from Sybase and Microsoft that add several features to standard SQL, including transaction control, exception and error handling, row pro- cessing, and declared variables.Generically, any database management system (DBMS) that can respond to queries from client machines formatted in the SQL language. When capitalized, the term generally refers to either of two database management products from Sybase and Microsoft. Both companies offer client-server DBMS products called SQL Server.

Figure 3.9: MySQL Configuration

## PHP Configuration

PHP for Windows must be installed from the zip package, not using the installer because the installer does not work correctly when setting up the configuration files. Download the latest Windows binary version from the 5.x release series.

## Java Script

JavaScript is a scripting language developed jointly by Sun and Netscape and is meant for the [WWW.](http://WWW/) A scripting language is a simple programming language designed to enable programmers to write useful programs quickly. A script is a similar to a macro, which tells a program how to perform a specific procedure.

JAVA SCRIPT FACTS

* + - * JavaScript is embedded in to HTML-JavaScript code is usually embedded into a HTML document and is executed, within them. By itself JavaScript has no user interface. It relies on, HTML to provide the means of interaction with the user. Most JavaScript objects have HTML tags they represent. JavaScript extends the capabilities of HTML by providing events to HTML tags and provide event driven code to execute it.
      * JavaScript is browser dependent- JavaScript depends on the Web browser to sup- port it. If the browser does not support it, JavaScript code will be ignored. Internet Explorer 3.0 and, Netscape Navigator 2.0 onwards supports JavaScript.
      * JavaScript is an interpreted language - JavaScript is interpreted at runtime by the browser before it is executed it. It is not compiled into a separate program like a

.exe but remains part of the HTML file.

* + - * JavaScript is a loosely typed language - JavaScript is very flexible compared to java. You need not specify the data type of a variable while declaring it. Also you need not declare variables explicitly. It is perfectly legal to declare variables as and when you require them.
      * JavaScript is an object based language - JavaScript is an object-based language. You can work with objects that encapsulate data (properties) and behavior (method). However, JavaScript object model is instance-based and there is no interface. This is the basic difference between an object oriented and an object-based language.
      * JavaScript is Event-Driven - HTML objects such as buttons are enhanced to sup- port event handlers.
      * JavaScript is Not Java - Java applet is stored in a separate file and connected to a HTML file through the ¡applet¿ tag, and it is a strongly typed, object oriented compiled language. JavaScript is loosely typed object based, interpreted language meant to create scripts.
      * JavaScript is Multifunctional.
      * JavaScript is Evolving - JavaScript evolved in December 1995. As the time goes by, JavaScript will be embedded and new features added to it.
      * JavaScript Language Spans Contents - JavaScript can be used in server side Netscape Live Wire Pro Environment and Microsofts Active X server frame work. It is not just a client side scripting tool.

# CHAPTER 4 SYSTEM TESTING

Software testing is an integral part of to ensure software quality, some software organi- zations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing .There are several factors that at- tribute the cost of software testing. Creating and maintaining large number of test cases is a time consuming process. Furthermore, it requires skilled and experienced testers to develop great quality test cases.

Even with the wide availability of automation tools for testing, the degree of automa- tion mostly remains at the automated test script level and generally significant amount of human intervention is required in testing. In addition data collected, as testing is conducted provides a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process. Testing begins at the module level and work towards the integration of entire computer based system. No testing is completed without verification and validation part.

The goal of verification and validation activities are to access and improve the qual- ity of work products generated during the development and modification of the soft- ware. Testing plays a vital role in determining the reliability and efficiency of the software and hence is very important stage in software development. Tests are to be conducted on the software to evaluate its performance under a number of conditions. Ideally, it should do so at the level of each module and also when all of them are inte- grated to form the completed system.

In the project ”E-Medication Reminder App And Healthcare system” the testing has been successfully handled with the modules. The test data was given to each and every module in all respect and got the desired output. Each module that has been tested is found working properly.

# UNIT TESTING

Here we test each module individually and integrated the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as ”module testing”. The modules of this project are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of the project independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usu- ally small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These error are verified and corrected and so that the unit perfectly fits to the project.

# INTEGRATION TESTING

Data can be lost across an interface, one module can have an adverse effect on the other sub-functions, when combined they may not perform the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance. The Modules of this project are connected and tested. After splitting the programs into units, the units were tested together to see the defects between each module and function. It is testing to one or more modules or functions together with the intent of finding interface defects between the modules or functions. Testing completed at as part of unit or functional testing, integration testing can involve putting together of groups of modules and functions with the goal of completing and verifying meets the system requirements.

# SYSTEM TESTING

System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Test-

ing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/System is tested in an environment that closely resembles the production environment where the application will be finally deployed. The perquisites for System Testing are:-

* + - All the components should have been successfully Unit Tested.
    - All the components should have been successfully integrated.

Testing should be completed in an environment closely resembling the production en- vironment. When necessary iterations of System Testing are done in multiple environ- ments.

# USER ACCEPTANCE TESTING

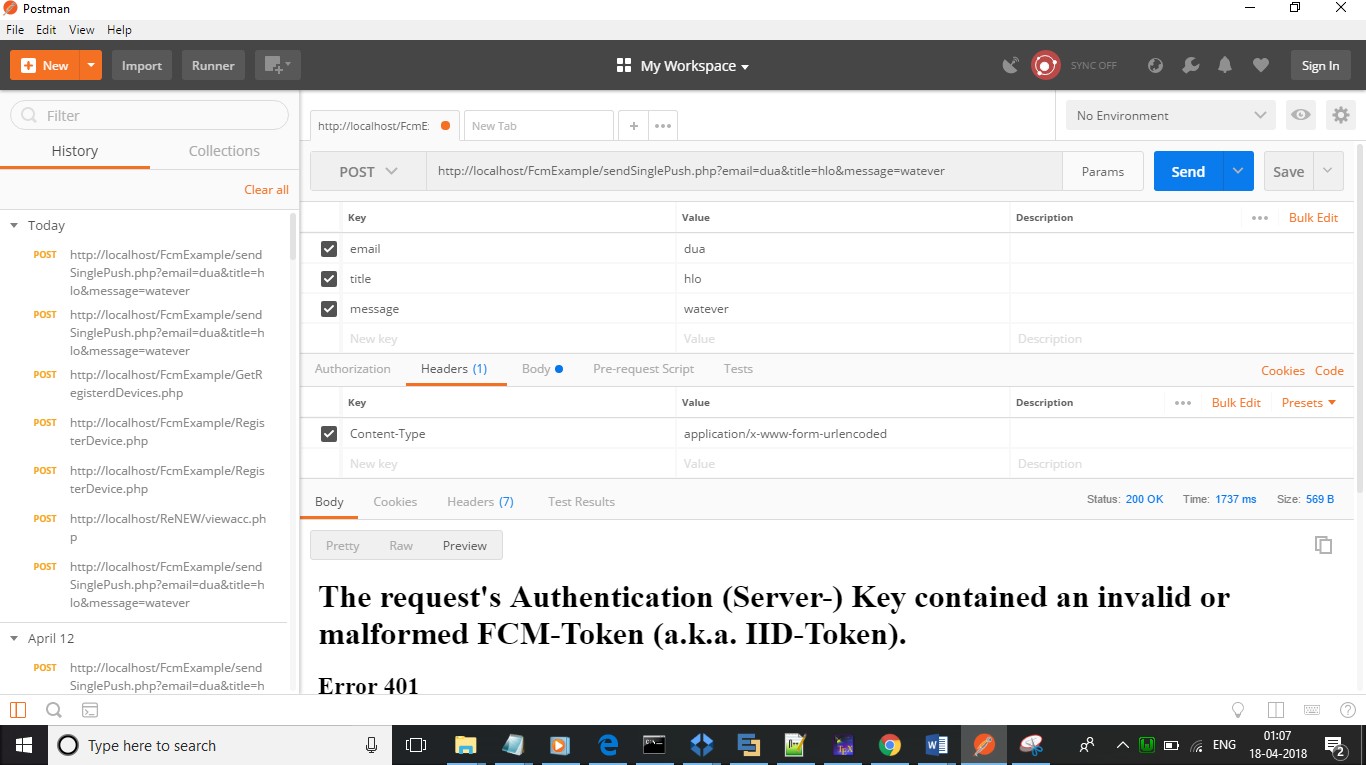
The system was tested by a small client community to see if the program met the re- quirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system.

Figure 4.1: Test case using POSTMAN

# CHAPTER 5 SYSTEM IMPLEMENTATION

The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested software system into actual work environ- ment. Implementation is concerned with translating design specification with source code. The primary goal of implementation is to write the source code to its specifi- cation that can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system de- sign into operational one. The three types of implementation are:-implementation of a computerized system to replace a manual system, implementation of a new system to replace existing one and implementation of a modified system to replace an existing one.

The implementation is the final stage and it is an important phase. It involves the individual programming ; system testing, user training , and the operational running of developed proposed system that constitute the application subsystem. The implemen- tation phase of the software development is concerned with translating design speci- fication in the source code. The user tests the developed system and the changes are according to the needs. Before implementation, Several tests have been conducted to ensure no errors encountered during the operation. The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the infor- mation system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs ,installs and op- erated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

# CHAPTER 6 CONCLUSION

* 1. edication Reminder App & Healthcare system gives reliable reminders, good user interface, nice user experience and it supports many new features supporting medication adherence. Many Medication Reminder Systems have been developed on different plat- forms. Many of these systems require special hardware devices to remind the patients about the medicine in-take timings. Purchasing new hardware devices becomes costly and more time and money consuming. So in the given work an attempt has been made to implement a system which is economical, easily accessible and improves medication adherence. Medication non-adherence reduces the effectiveness of a treatment and im- poses a financial burden on health care systems . The patients will get the schedule of medicine in-take time with medicine description, starting and ending date of medicine, notifications through automatic alarms. The scheduled reminder will not suggest any kind of medicine which is not prescribed by the doctor that will assure the safety of the patient and also will avoid wrong dosages. The patients can also make direct appoint- ments with the doctor that saves the time. Doctors can enhance the overall prescription process through E-prescriptions and thereby decreasing the dispense errors.it also make the administrator works on the go.

The main focus is on improving the overall performance of the system. Also, inter- action between patients and doctors through video calling and secure prescription will be focused upon. Some more ways to achieve medication adherence will be focused.

# CHAPTER 7 REFERENCES

* + - International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT) Vol. 6, No. 2, June 2015 DOI : 10.5121/ijmpict.2015.6204 39 MEDICATION REMINDER AND HEALTHCARE AN ANDROID APPLI- CATION Deepti Ameta, Kalpana Mudaliar and Palak Patel

# CHAPTER 8 APPENDENIX

* 1. **SCREEN SHOTS INPUT FORM,OUTPUT FORMS**

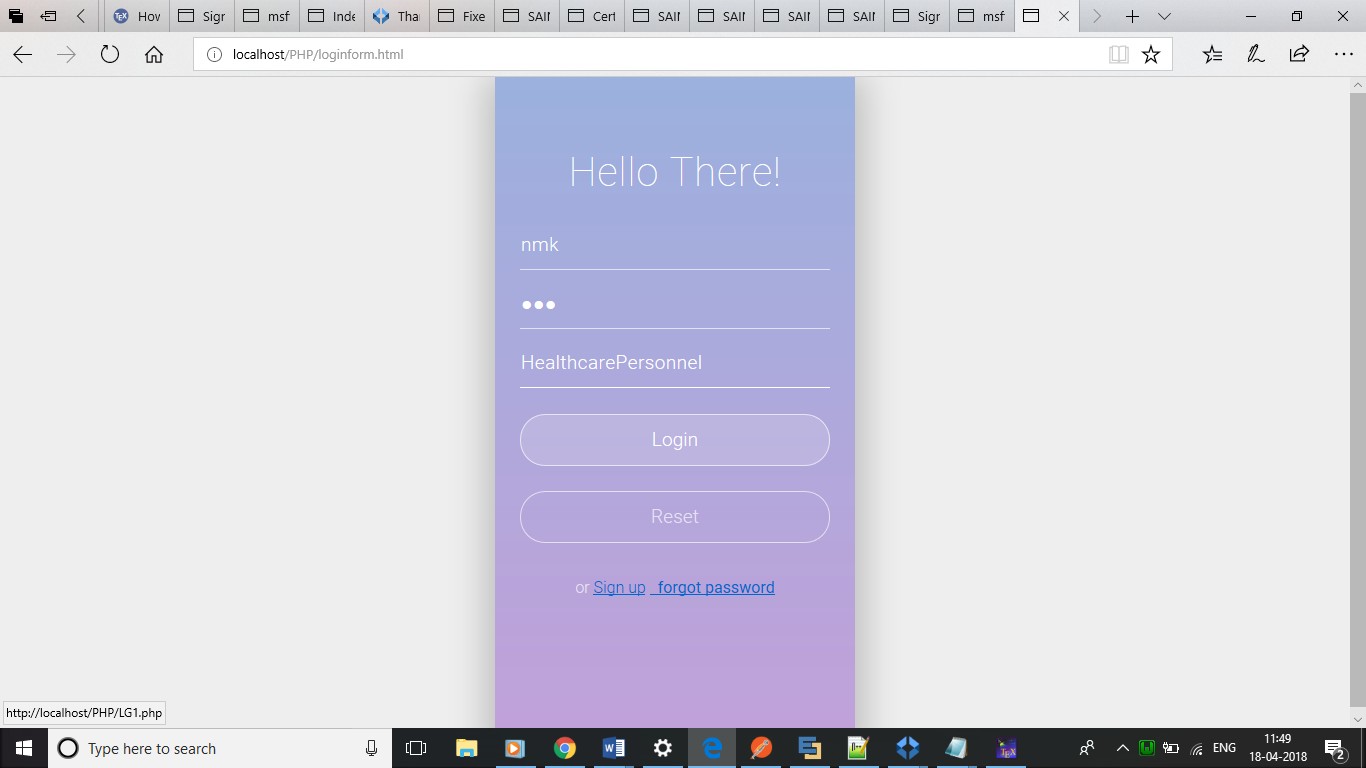


Figure 8.1: LOGIN PAGE



Figure 8.2: DOCTOR HOMEPAGE

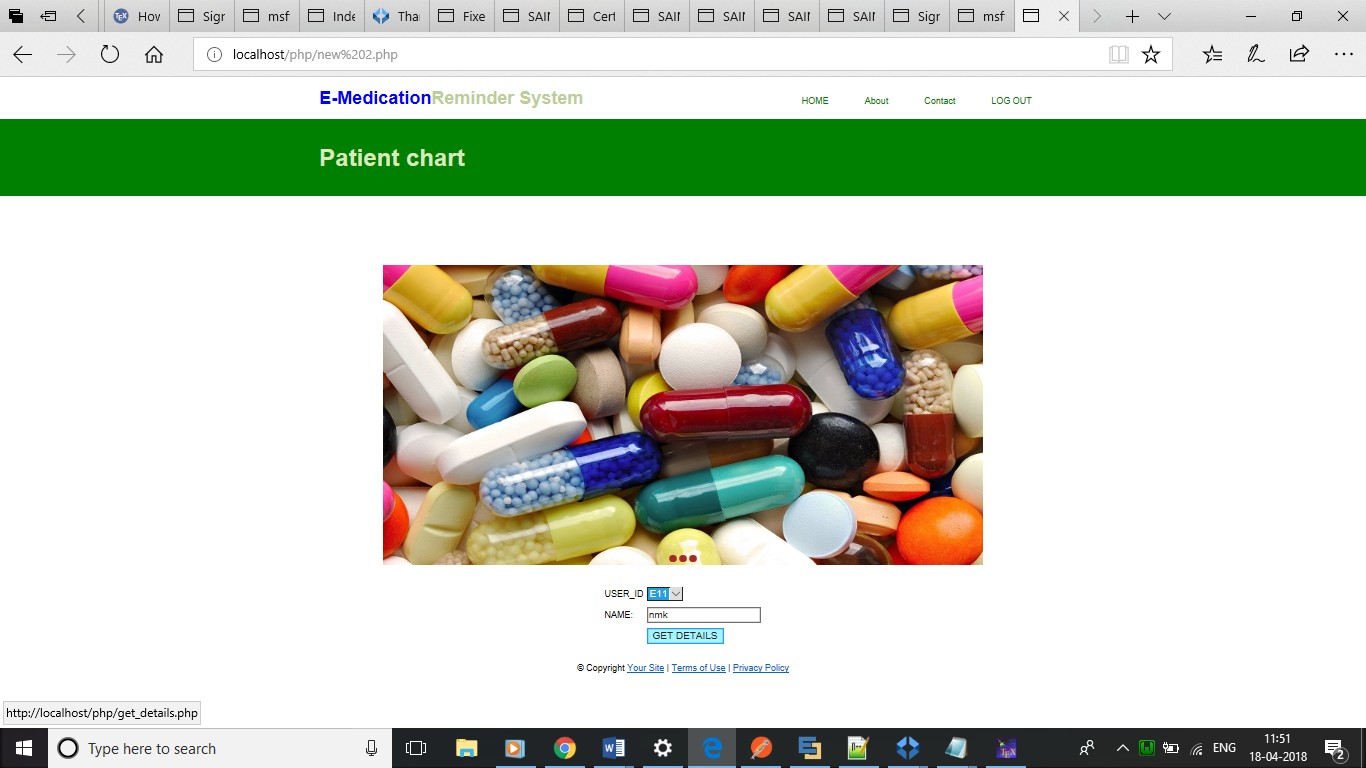


Figure 8.3: ACCESSING PATIENT CHART

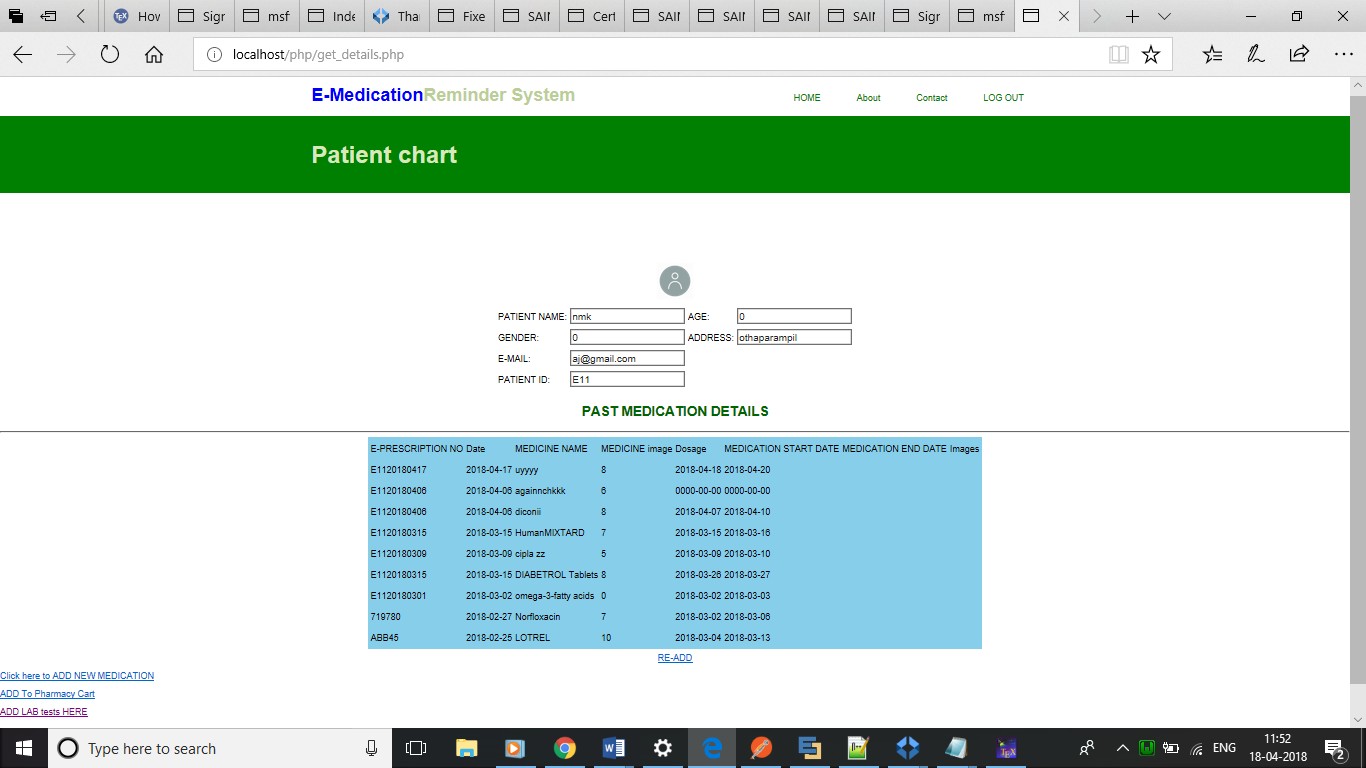


Figure 8.4: E-PRESCRIPTION-PATIENT MEDICATION HISTORY

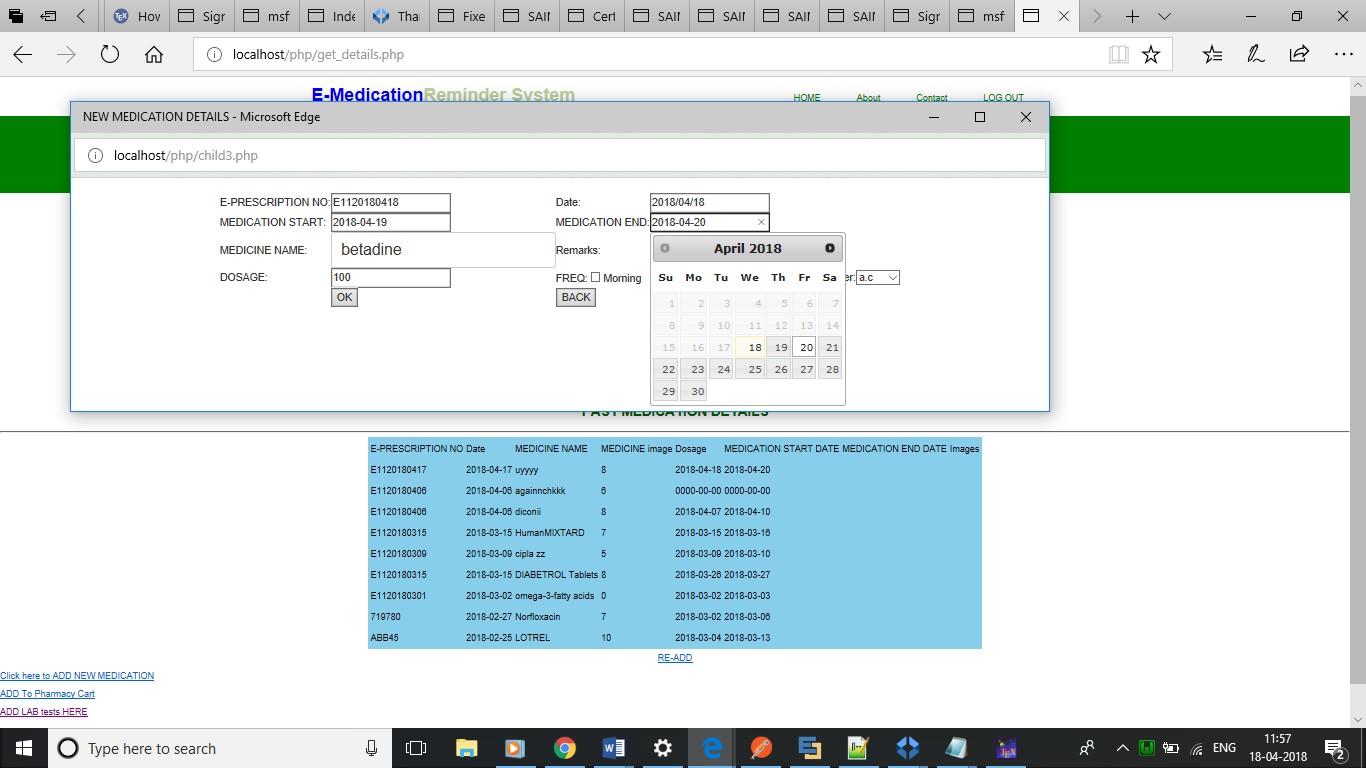


Figure 8.5: ADDING NEW MEDICATION PLAN

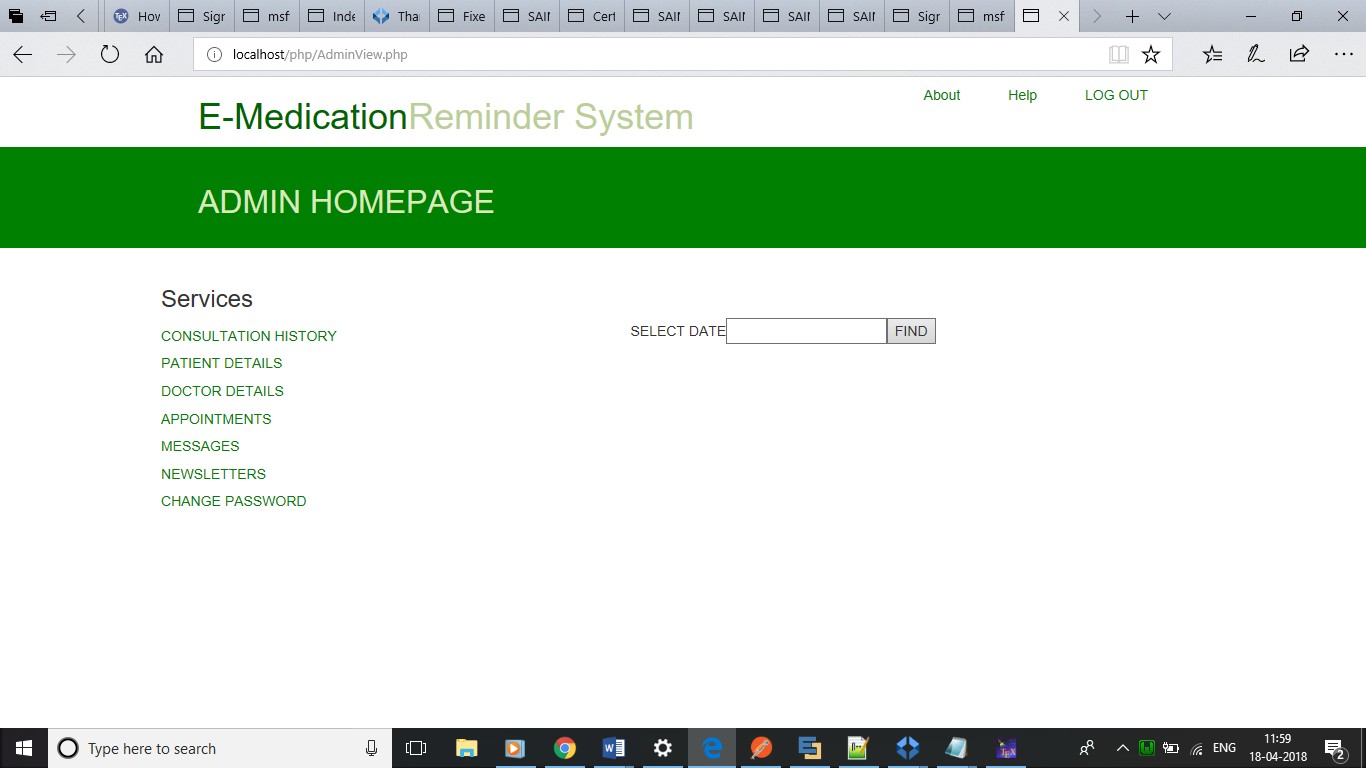


Figure 8.6: ADMIN HOME PAGE

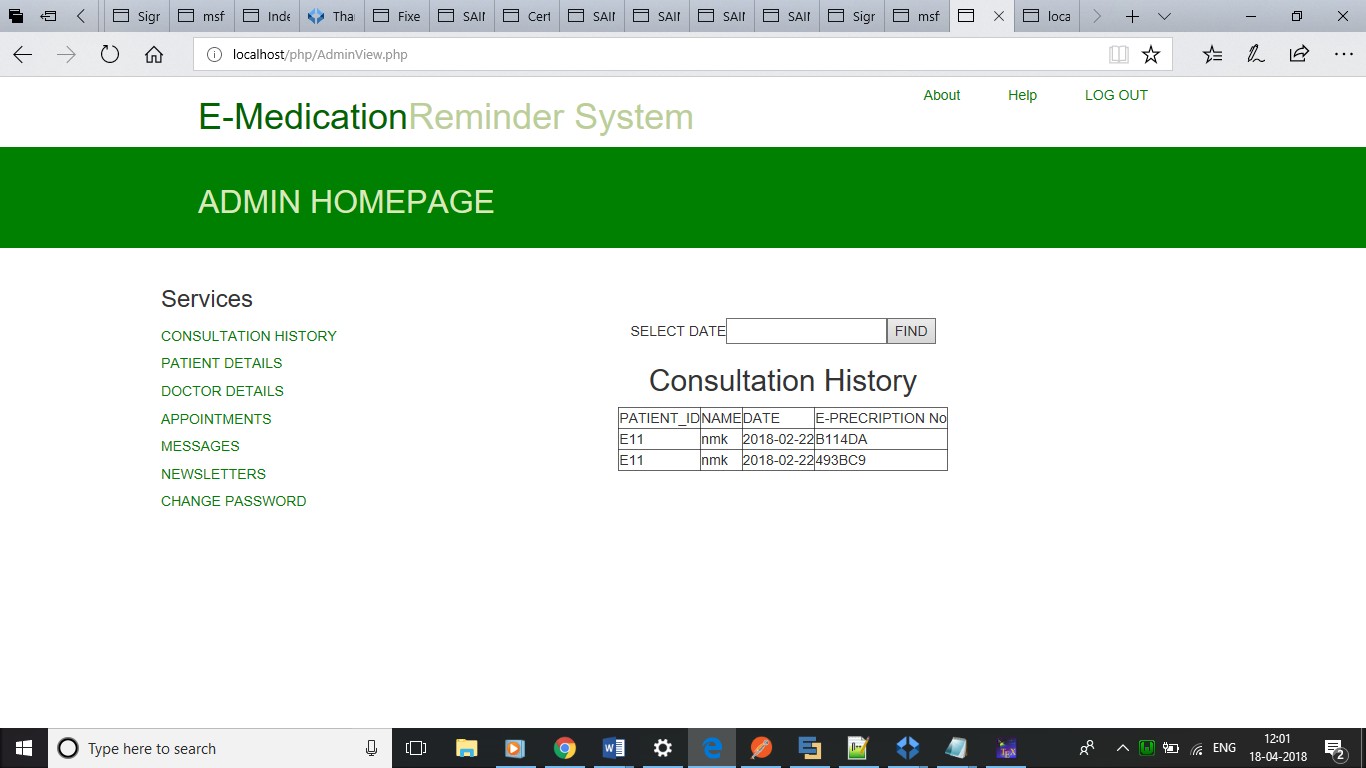


Figure 8.7: CONSULTATION HISTORY

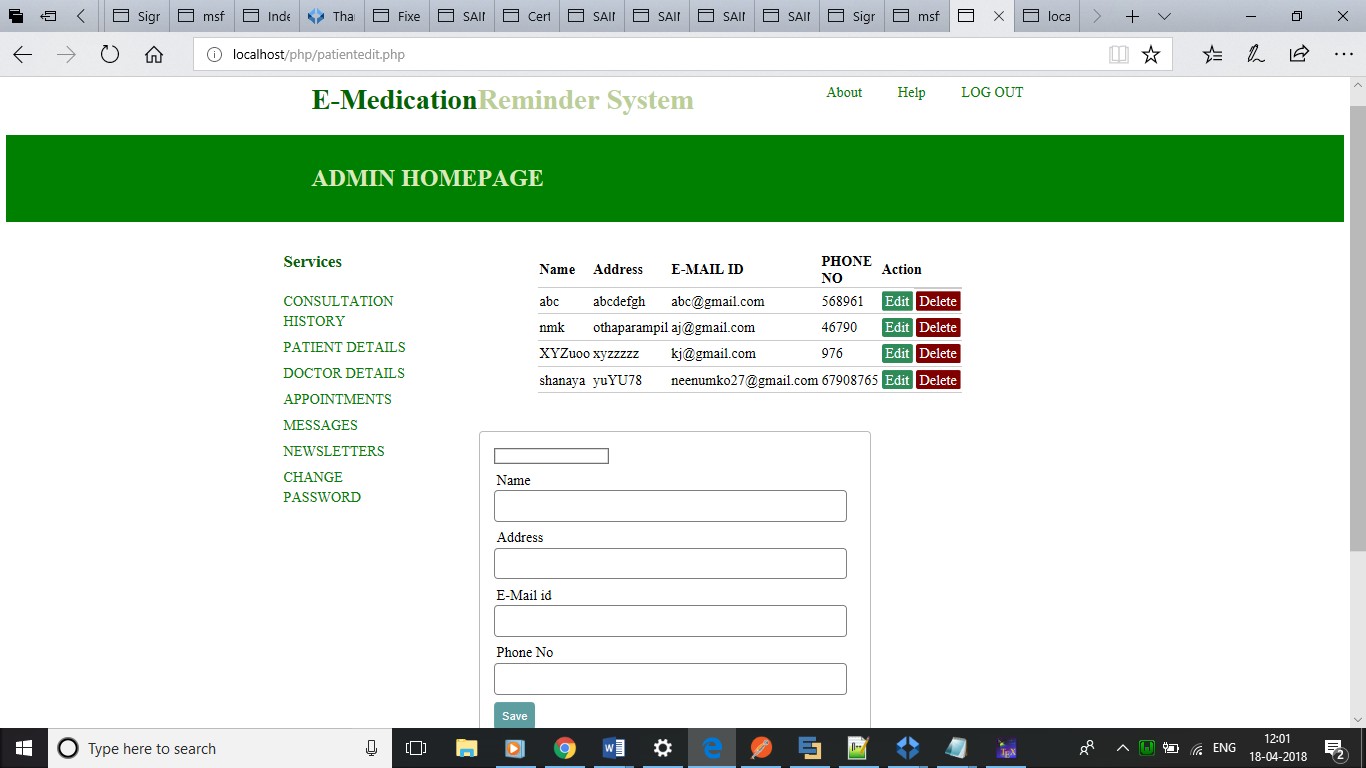


Figure 8.8: VIEW,ADD,DELETE,EDIT PATIENT DETAILS

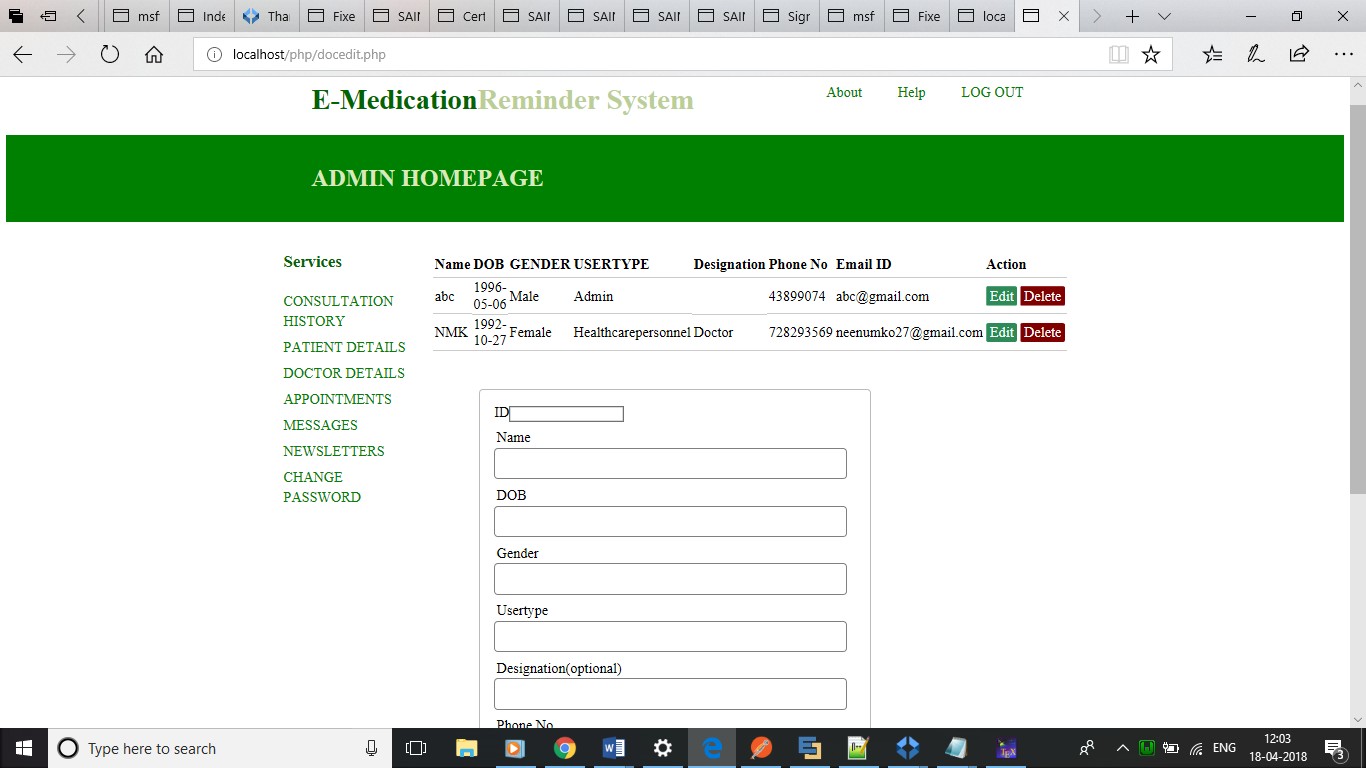


Figure 8.9: VIEW,ADD,DELETE,EDIT DOCTORS’ DETAILS

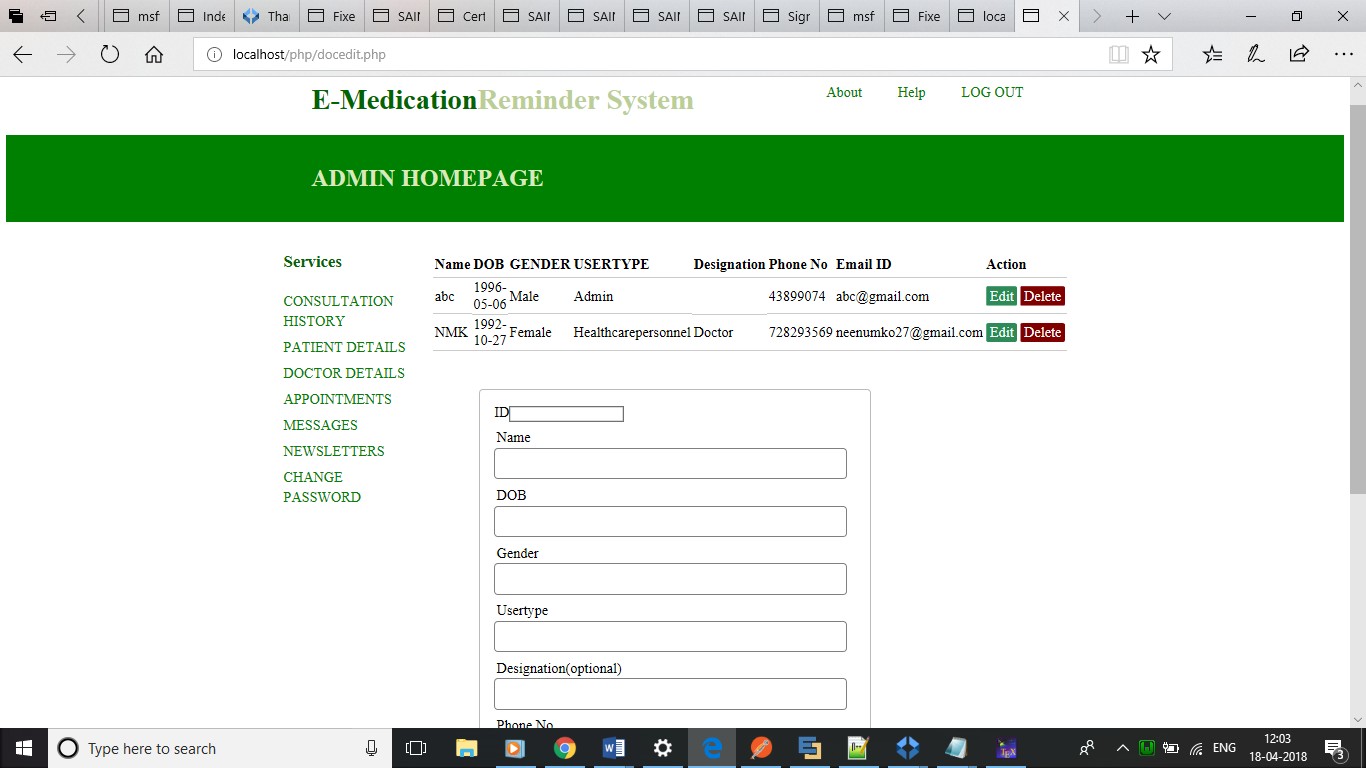


Figure 8.10: APPROVAL/DISAPPROVAL OF APPOINTMENT REQUESTS

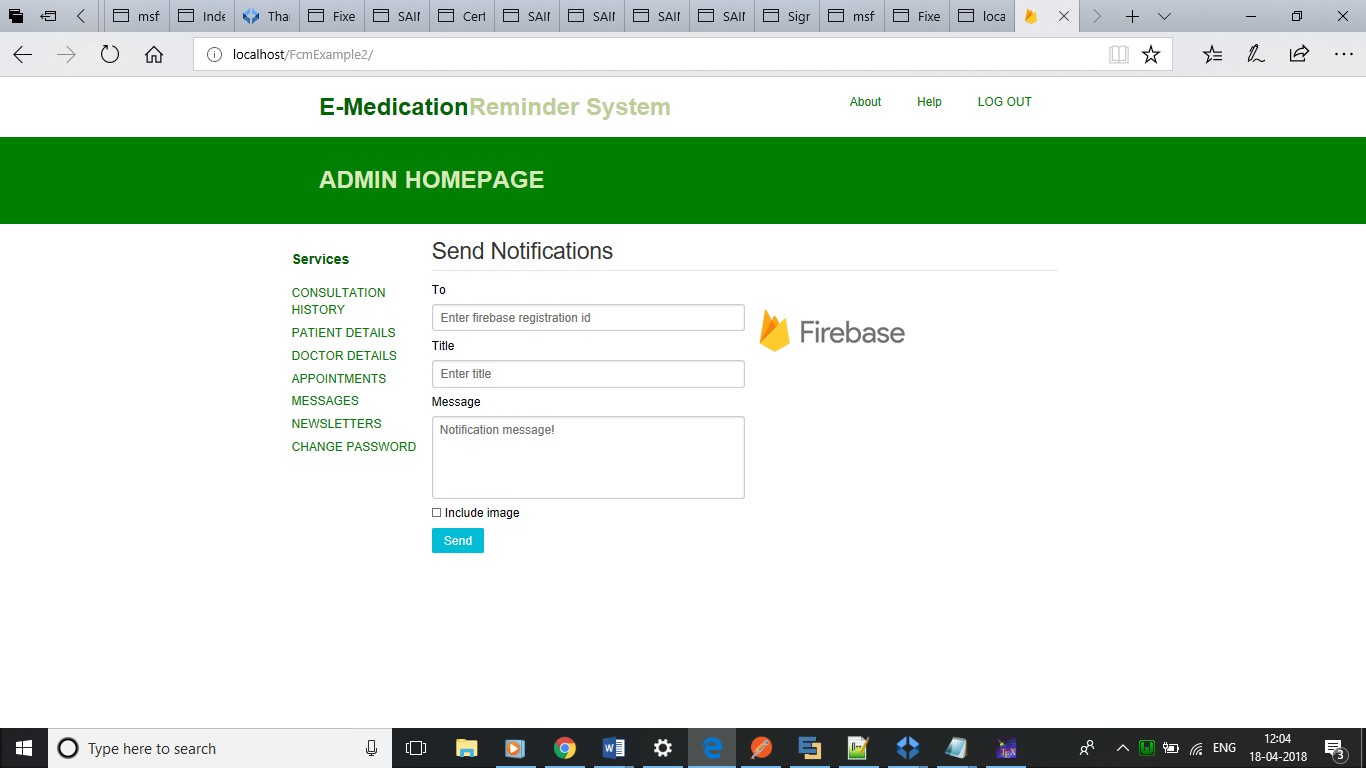


Figure 8.11: SENDING DAILY HEALTH TIPS USING FIREBASE

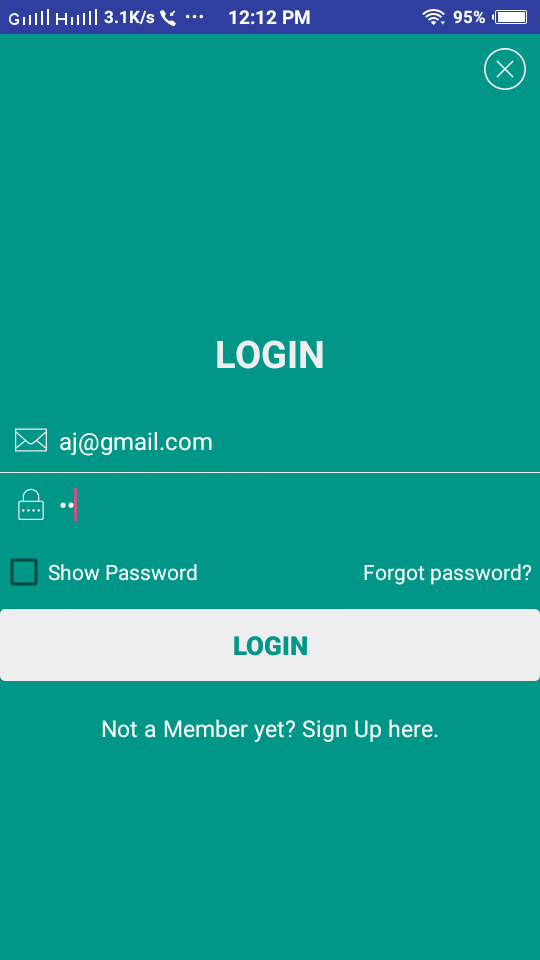


Figure 8.12: E-REMINDER APP LOGIN PAGE

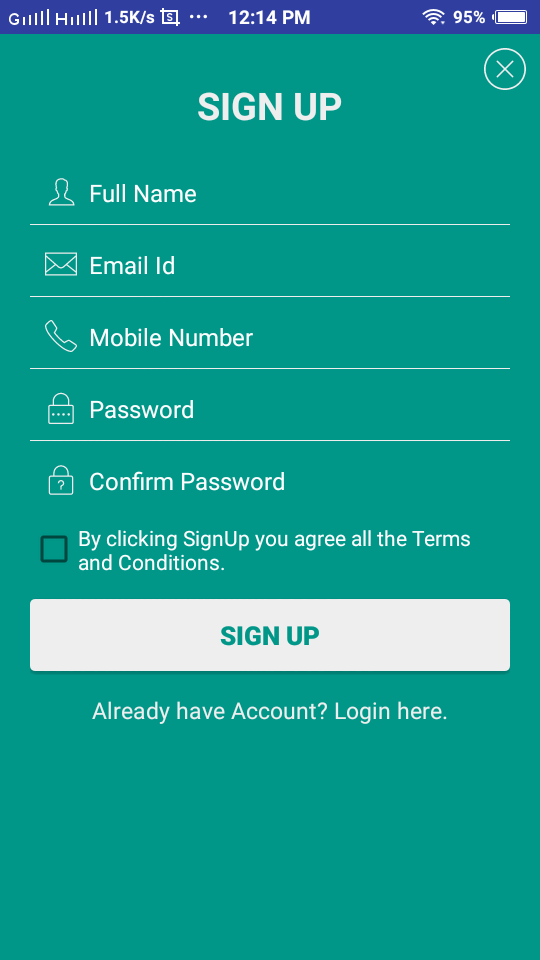


Figure 8.13: REGISTRATION FORM



53

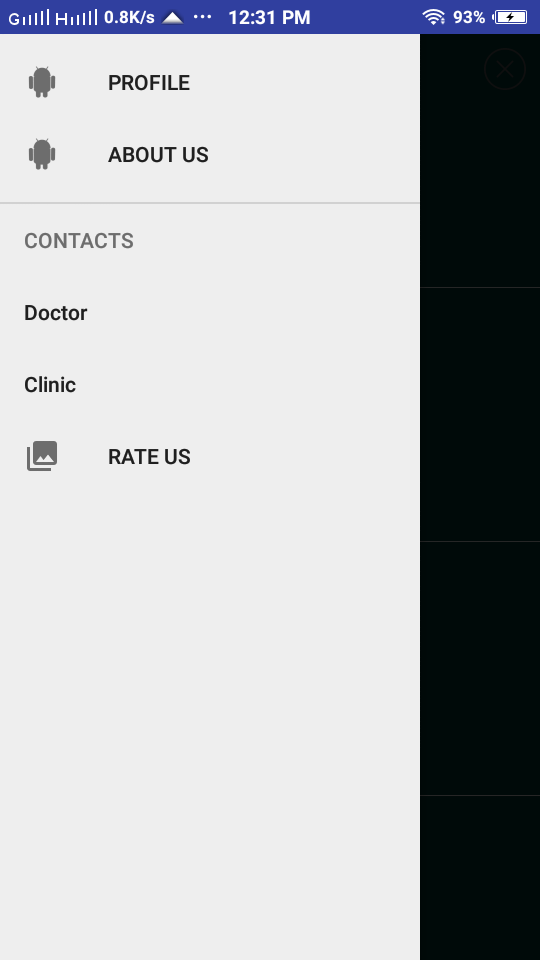


Figure 8.14: NAVIGATION DRAWER



54

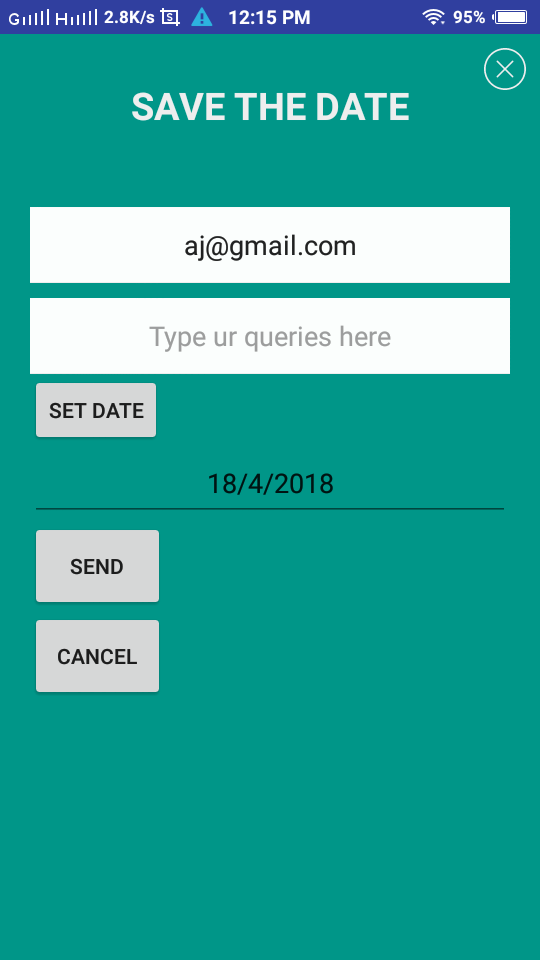
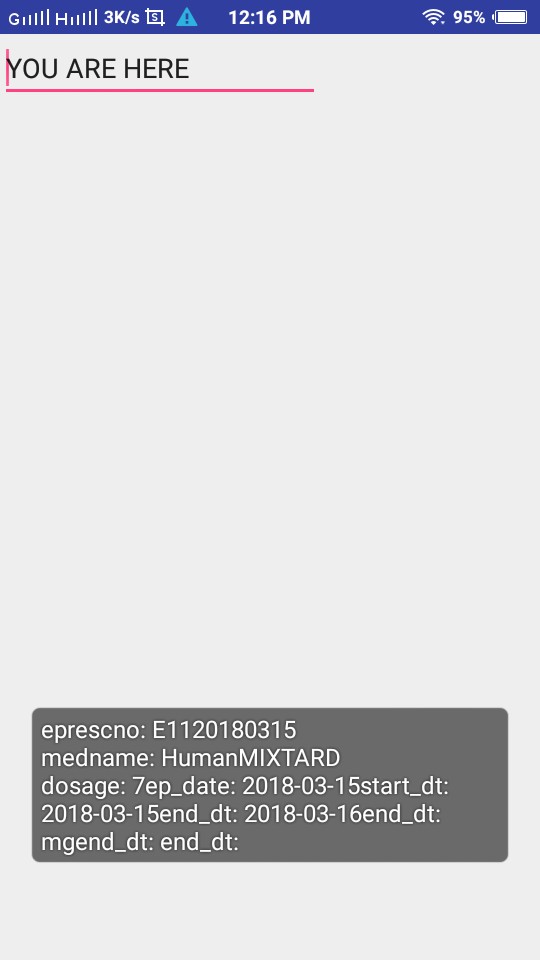


Figure 8.16: FIX APPOINTMENT



55

# SAMPLE CODE

## Userlogin.java

1 packagecom.example.android.login\_usingphp;

2

3 importandroid.app.ProgressDialog;

4 importandroid.content.Context;

5 importandroid.content.Intent;

6 importandroid.content.SharedPreferences;

7 importandroid.os.AsyncTask;

8 importandroid.renderscript.Sampler;

9 importandroid.support.v7.app.AppCompatActivity;

10 importandroid.os.Bundle;

11 importandroid.text.InputType;

12 importandroid.text.TextUtils;

13 importandroid.text.method.HideReturnsTransformationMethod;

14 importandroid.text.method.PasswordTransformationMethod;

15 importandroid.view.View;

16 importandroid.view.animation.Animation;

17 importandroid.view.animation.AnimationUtils;

18 importandroid.widget.Button;

19 importandroid.widget.CheckBox;

20 importandroid.widget.CompoundButton;

21 importandroid.widget.EditText;

22 importandroid.widget.ImageButton;

23 importandroid.widget.ImageView;

24 importandroid.widget.TextView;

25 importandroid.widget.Toast;

26 importjava.util.HashMap;

27

28 publicclassUserLoginActivityextendsAppCompatActivity{

29

30 EditTextEmail,Password;

31 TextViewsignup,forgotPassword;

32 ButtonLogIn;

33 CheckBoxshow\_hide\_password;

34 ImageViewimg;

35 StringPasswordHolder,EmailHolder;

36 StringfinalResult;

37 AnimationshakeAnimation;

38 [StringHttpURL="http://192.168.43.159/ReNEW/UserLogin.php](http://192.168.43.159/ReNEW/UserLogin.php)";

39 BooleanCheckEditText;

40 ProgressDialogprogressDialog;

41 HashMap<String,String>hashMap=newHashMap<>();

42 HttpParsehttpParse=newHttpParse();

43 publicstaticfinalStringUserEmail="";

44 privateStringValue;

45

46 @Override

47 protectedvoidonCreate(BundlesavedInstanceState) {

48 super.onCreate(savedInstanceState);

49 setContentView(R.layout.activity\_user\_login);

50

51 Email= (EditText)findViewById(R.id.email1);

52 Password= (EditText)findViewById(R.id.password1);

53 LogIn= (Button)findViewById(R.id.Login1);

54 img= (ImageView)findViewById(R.id.close\_activity);

55 signup= (TextView)findViewById(R.id.btnSign);

56 show\_hide\_password= (CheckBox)findViewById(R.id. show\_hide\_password);

57 shakeAnimation=AnimationUtils.loadAnimation(UserLoginActivity. this,R.anim.shake);

58 forgotPassword= (TextView)findViewById(R.id.forgot\_password);

59 LogIn.setOnClickListener(newView.OnClickListener() {

60 @Override

61 publicvoidonClick(Viewview) {

62

63 CheckEditTextIsEmptyOrNot();

64

65 if(CheckEditText){

66

67 UserLoginFunction(EmailHolder,PasswordHolder);

68

69 }

70 else{

71 view.startAnimation(shakeAnimation);

72 newCustomToast().Show\_Toast(UserLoginActivity.this,view,"Enter bothcredentials.");

73 //Toast.makeText(UserLoginActivity.this,"Pleasefillallform fields.",Toast.LENGTH\_LONG).show();

74

75 }

76

77 }

78 });

79 signup.setOnClickListener(newView.OnClickListener() {

80 @Override

81 publicvoidonClick(Viewview) {

82

83 Intentintent=newIntent(UserLoginActivity.this,MainActivity. class);

|  |  |  |  |
| --- | --- | --- | --- |
| 84 | startActivity(intent); | |  |
| 85 |  | |
| 86 | } | |
| 87 | }); | |
| 88 | forgotPassword.setOnClickListener(newView.OnClickListener() | | { |
| 89 | @Override | |  |
| 90 | publicvoidonClick(Viewview) { | |  |
| 91 |  | |  |
| 92 | Intentintent=newIntent(UserLoginActivity.this,  ForgotPassword\_Fragment.class); | |  |
| 93 | startActivity(intent); |  | |
| 94 |  |  | |
| 95 | } |  | |
| 96 | }); |  | |
| 97 | img.setOnClickListener(newView.OnClickListener() | { | |
| 98 | @Override |  | |
| 99 | publicvoidonClick(Viewview) { |  | |
| 100 |  |  | |
| 101 | finish(); |  | |
| 102 | System.exit(0); |  | |
| 103 |  |  | |
| 104 | } |  | |
| 105 | }); |  | |

106 show\_hide\_password

107 .setOnCheckedChangeListener(newCompoundButton. OnCheckedChangeListener() {

108

109 @Override

110 publicvoidonCheckedChanged(CompoundButtonbutton,

111 booleanisChecked) {

112 if(isChecked) {

113

114 show\_hide\_password.setText(R.string.hide\_pwd);//change

115 Password.setInputType(InputType.TYPE\_CLASS\_TEXT);

116 Password.setTransformationMethod(HideReturnsTransformationMethod

117 .getInstance());//showpassword

118 }else{

119 show\_hide\_password.setText(R.string.show\_pwd);//change

120 Password.setInputType(InputType.TYPE\_CLASS\_TEXT

121 |InputType.TYPE\_TEXT\_VARIATION\_PASSWORD);

122 Password.setTransformationMethod(PasswordTransformationMethod

123 .getInstance());//hidepassword

124

125 }

126 }

127 });

128 }

129 publicvoidCheckEditTextIsEmptyOrNot(){

130

131 EmailHolder=Email.getText().toString();

132 PasswordHolder=Password.getText().toString();

133

134 if(TextUtils.isEmpty(EmailHolder) ||TextUtils.isEmpty( PasswordHolder))

|  |  |
| --- | --- |
| 135 | { |
| 136 | CheckEditText=false; |
| 137 | } |
| 138 | else{ |
| 139 |  |
| 140 | CheckEditText=true; |
| 141 | } |

142 }

143

144 publicvoidUserLoginFunction(finalStringemail,finalString password){

145

146 classUserLoginClassextendsAsyncTask<String,Void,String> {

147

148 @Override

149 protectedvoidonPreExecute() {

150 super.onPreExecute();

151

152 progressDialog=ProgressDialog.show(UserLoginActivity.this," LoadingData",null,true,true);

153 }

154

155 @Override

156 protectedvoidonPostExecute(StringhttpResponseMsg) {

157

158 super.onPostExecute(httpResponseMsg);

159

160 progressDialog.dismiss();

161

162 if(httpResponseMsg.equalsIgnoreCase("DataMatched")){

163

164 finish();

165

166 Intentintent=newIntent(UserLoginActivity.this,NavMainActivity

.class);

167 SharedPreferencessettings=getSharedPreferences("mysettings",

168 Context.MODE\_PRIVATE);

169 SharedPreferences.Editoreditor=settings.edit();

170

171 editor.putString("mystring",EmailHolder);

172 editor.commit();

173 //intent.putExtra(UserEmail,email);

174

175 startActivity(intent);

176

177 }

178 else{

179 //newCustomToast().Show\_Toast(UserLoginActivity.this,"YourEmail IdisInvalid.");

180 Toast.makeText(UserLoginActivity.this,httpResponseMsg,Toast. LENGTH\_LONG).show();

181 }

182

183 }

184

185 @Override

186 protectedStringdoInBackground(String...params) {

187

188 hashMap.put("email",params[0]);

189

190 hashMap.put("password",params[1]);

191

192 finalResult=httpParse.postRequest(hashMap,HttpURL);

193

194 returnfinalResult;

195 }

196 }

197

198 UserLoginClassuserLoginClass=newUserLoginClass();

199

200 userLoginClass.execute(email,password);

201 }

202

203 }

## MainActivity.java

1 packagecom.example.android.login\_usingphp;

2

3 publicclassMainActivityextendsAppCompatActivity{

4

5 Buttonregister;

6 TextViewlog\_in;

7 ImageViewimg;

8 EditTextFirst\_Name,Email,Mobile,Password,confirmpass;

9 privatestaticCheckBoxterms\_conditions;

10 StringF\_Name\_Holder,L\_Name\_Holder,EmailHolder,PasswordHolder, getConfirmPassword;

11 StringfinalResult;

12 [StringHttpURL="http://192.168.43.159/ReNEW/UserRegistration.php](http://192.168.43.159/ReNEW/UserRegistration.php) ";

13 BooleanCheckEditText;

14 ProgressDialogprogressDialog;

15 HashMap<String,String>hashMap=newHashMap<>();

16 HttpParsehttpParse=newHttpParse();

17

18

19 @Override

20 protectedvoidonCreate(BundlesavedInstanceState) {

21 super.onCreate(savedInstanceState);

22 setContentView(R.layout.activity\_main);

23

24 //AssignId’S

25 First\_Name= (EditText)findViewById(R.id.editTextF\_Name);

26 Email= (EditText)findViewById(R.id.editTextL\_Name);

27 Mobile= (EditText)findViewById(R.id.editTextEmail);

28 Password= (EditText)findViewById(R.id.editTextPassword);

29 confirmpass= (EditText)findViewById(R.id.confirmPassword);

30 img= (ImageView)findViewById(R.id.close\_activity);

31

32 register= (Button)findViewById(R.id.Submit);

33 log\_in= (TextView)findViewById(R.id.Login);

34 terms\_conditions= (CheckBox)findViewById(R.id.terms\_conditions);

35

36 //AddingClickListeneronbutton.

37 register.setOnClickListener(newView.OnClickListener() {

38 @Override

39 publicvoidonClick(Viewview) {

40

41 //CheckingwhetherEditTextisEmptyorNot

42 CheckEditTextIsEmptyOrNot();

43

44 if(CheckEditText){

45

46 //IfEditTextisnotemptyandCheckEditText=Truethenthis blockwillexecute.

47

48 UserRegisterFunction(F\_Name\_Holder,L\_Name\_Holder,EmailHolder, PasswordHolder);

49

50 }

51 else{

52

53 //IfEditTextisemptythenthisblockwillexecute.

54 //Toast.makeText(MainActivity.this,"Pleasefillallformfields

.",Toast.LENGTH\_LONG).show();

55 newCustomToast().Show\_Toast(MainActivity.this,view,"Allfields arerequired.");

56 }

57

58

59 }

60 });

61

62 log\_in.setOnClickListener(newView.OnClickListener() {

63 @Override

64 publicvoidonClick(Viewview) {

65

66 Intentintent=newIntent(MainActivity.this,UserLoginActivity. class);

67 startActivity(intent);

68

69 }

70 });

71 img.setOnClickListener(newView.OnClickListener() {

72 @Override

73 publicvoidonClick(Viewview) {

74

75 finish();

76 System.exit(0);

77

78 }

79 });

80

81 }

82

83 publicvoidCheckEditTextIsEmptyOrNot(){

84

85 F\_Name\_Holder=First\_Name.getText().toString();

86 L\_Name\_Holder=Email.getText().toString();

87 EmailHolder=Mobile.getText().toString();

88 PasswordHolder=Password.getText().toString();

89 StringgetConfirmPassword=confirmpass.getText().toString();

90 Patternp=Pattern.compile(Utils.regEx);

91 Matcherm=p.matcher(L\_Name\_Holder);

92

93

94 if(TextUtils.isEmpty(F\_Name\_Holder) ||TextUtils.isEmpty( L\_Name\_Holder) ||TextUtils.isEmpty(EmailHolder) ||TextUtils. isEmpty(PasswordHolder))

95 {

96

97 CheckEditText=false;

98

99 }

100 elseif(!m.find()){

101 CheckEditText=false;

102 Toast.makeText(MainActivity.this,"invalidemailid.",Toast. LENGTH\_LONG).show();

103

104 }

105 elseif(!getConfirmPassword.equals(PasswordHolder)) {

106 CheckEditText=false;

107 Toast.makeText(MainActivity.this,"Bothpassworddoesn’tmatch.", Toast.LENGTH\_LONG).show();

108

109 }

110 elseif(!terms\_conditions.isChecked()) {

111 CheckEditText=false;

112 Toast.makeText(MainActivity.this,"PleaseselectTermsand Conditions.",Toast.LENGTH\_LONG).show();

113 }

114

115 else{

116

117 CheckEditText=true;

118 //Toast.makeText(MainActivity.this,"DoSignUp.",Toast.LENGTH\_LONG

).show();

119

120 }

121

122 }

123

124 publicvoidUserRegisterFunction(finalStringF\_Name,finalString L\_Name,finalStringemail,finalStringpassword){

125

126 classUserRegisterFunctionClassextendsAsyncTask<String,Void, String> {

127

128 @Override

129 protectedvoidonPreExecute() {

130 super.onPreExecute();

131

132 progressDialog=ProgressDialog.show(MainActivity.this,"Loading Data",null,true,true);

133 }

134

135 @Override

136 protectedvoidonPostExecute(StringhttpResponseMsg) {

137

138 super.onPostExecute(httpResponseMsg);

139

140 progressDialog.dismiss();

141

142 Toast.makeText(MainActivity.this,httpResponseMsg.toString(),Toast

.LENGTH\_LONG).show();

143

144 }

145

146 @Override

147 protectedStringdoInBackground(String...params) {

148

149 hashMap.put("f\_name",params[0]);

150

151 hashMap.put("L\_name",params[1]);

152

153 hashMap.put("email",params[2]);

154

155 hashMap.put("password",params[3]);

156

157 finalResult=httpParse.postRequest(hashMap,HttpURL);

158

159 returnfinalResult;

160 }

161 }

162

163 UserRegisterFunctionClassuserRegisterFunctionClass=new UserRegisterFunctionClass();

164

165 userRegisterFunctionClass.execute(F\_Name,L\_Name,email,password);

166 }

167

168 }

## NavigationDrawer.java

1 packagecom.example.android.login\_usingphp;

2

3 importandroid.content.Intent;

4 importandroid.os.Bundle;

5 importandroid.support.design.widget.NavigationView;

6 importandroid.support.v4.app.Fragment;

7 importandroid.support.v4.app.FragmentTransaction;

8 importandroid.support.v4.widget.DrawerLayout;

9 importandroid.support.v7.app.AppCompatActivity;

10 importandroid.view.MenuItem;

11 importandroid.view.View;

12 importandroid.widget.Button;

13 importandroid.widget.ImageButton;

14 importandroid.widget.TextView;

15 importandroid.widget.Toast;

16

17 importcom.example.android.login\_usingphp.Miscll.FixApt;

18 importcom.example.android.login\_usingphp.Miscll.Postqueries;

19

20 importstaticcom.example.android.login\_usingphp.UserLoginActivity

.UserEmail;

21

22 publicclassNavMainActivityextendsAppCompatActivity{

23 DrawerLayoutdLayout;

24 ImageButtonbt1;

25 ImageButtonbt2,bt3;

26 ButtonLogOut;

27 TextViewEmailShow;

28 StringEmailHolder;

29

30 @Override

31 protectedvoidonCreate(BundlesavedInstanceState) {

32 super.onCreate(savedInstanceState);

33 setContentView(R.layout.activity\_main1);

34 setNavigationDrawer();//callmethod

35 bt2=(ImageButton)findViewById(R.id.button2);

36 bt3=(ImageButton)findViewById(R.id.button3);

37 bt1= (ImageButton)findViewById(R.id.bt1);

38

39 LogOut= (Button)findViewById(R.id.button);

40 EmailShow= (TextView)findViewById(R.id.EmailShow);

41

42

43 Intentintent=getIntent();

44 EmailHolder=intent.getStringExtra(UserEmail);

45 EmailShow.setText(EmailHolder);

46

47

48

49 bt1.setOnClickListener(newView.OnClickListener() {

50 @Override

51 publicvoidonClick(Viewview) {

52

53 finish();

54

55 Intentintent=newIntent(NavMainActivity.this,Homepg.class);

56 intent.putExtra(UserEmail,EmailHolder);

57

58

59 startActivity(intent);

60

61 Toast.makeText(NavMainActivity.this,"WELCOME",Toast.LENGTH\_LONG

).show();

62

63

64 }

65 });

66 bt2.setOnClickListener(newView.OnClickListener() {

67 @Override

68 publicvoidonClick(Viewview) {

69

70 finish();

71

72 Intentintent=newIntent(NavMainActivity.this,Postqueries.class

);

73 intent.putExtra(UserEmail,EmailHolder);

74

75

76 startActivity(intent);

77

78 //Toast.makeText(NavMainActivity.this,"WELCOME",Toast. LENGTH\_LONG).show();

79

80

81 }

82 });

83 bt3.setOnClickListener(newView.OnClickListener() {

84 @Override

85 publicvoidonClick(Viewview) {

86

87 finish();

88

89 Intentintent=newIntent(NavMainActivity.this,FixApt.class);

90 intent.putExtra(UserEmail,EmailHolder);

91

92

93 startActivity(intent);

94

95 //Toast.makeText(NavMainActivity.this,"WELCOME",Toast. LENGTH\_LONG).show();

96

97

98 }

99 });

100

101

102 }

103

104 privatevoidsetNavigationDrawer() {

105 dLayout= (DrawerLayout)findViewById(R.id.drawer\_layout);// initiateaDrawerLayout

106 NavigationViewnavView= (NavigationView)findViewById(R.id. navigation);//initiateaNavigationView

107 //implementsetNavigationItemSelectedListenereventon NavigationView

108 navView.setNavigationItemSelectedListener(newNavigationView. OnNavigationItemSelectedListener() {

109 @Override

110 publicbooleanonNavigationItemSelected(MenuItemmenuItem) {

111

112 Fragmentfrag=null;//createaFragmentObject

113 intitemId=menuItem.getItemId();//getselectedmenuitem’sid

114

115 //checkselectedmenuitem’sidandreplaceaFragment Accordingly

|  |  |
| --- | --- |
| 116 | if(itemId==R.id.first) { |
| 117 | frag=newFirstFragment(); |
| 118 | }elseif(itemId==R.id.second) { |
| 119 | frag=newSecondFragment(); |
| 120 | }elseif(itemId==R.id.third) { |
| 121 | frag=newThirdFragment(); |
| 122 | } |
| 123 | //displayatoastmessagewithmenuitem’stitle |
| 124 | Toast.makeText(getApplicationContext(),menuItem.getTitle(),Toast |

.LENGTH\_SHORT).show();

125 if(frag!=null) {

126 FragmentTransactiontransaction=getSupportFragmentManager(). beginTransaction();

127 transaction.replace(R.id.frame,frag);//replaceaFragmentwith FrameLayout

|  |  |
| --- | --- |
| 128 | transaction.commit();//committhechanges |
| 129 | dLayout.closeDrawers();//closetheallopenDrawerViews |
| 130 | returntrue; |
| 131 | } |
| 132 |  |
| 133 | returnfalse; |
| 134 | } |
| 135 | }); |
| 136 | } |
| 137 | } |

## NotificationHelper.java

1 packagecom.example.android.login\_usingphp.notification;

2

3 importandroid.app.AlarmManager;

4 importandroid.app.NotificationManager;

5 importandroid.app.PendingIntent;

6 importandroid.content.ComponentName;

7 importandroid.content.Context;

8 importandroid.content.Intent;

9 importandroid.content.pm.PackageManager;

10 importandroid.os.SystemClock;

11

12 importjava.util.Calendar;

13

14 importstaticandroid.content.Context.ALARM\_SERVICE;

15

16 publicclassNotificationHelper{

17 publicstaticintALARM\_TYPE\_ELAPSED= 101;

18 privatestaticAlarmManageralarmManagerElapsed;

19 privatestaticPendingIntentalarmIntentElapsed;

20 publicstaticintALARM\_TYPE\_RTC= 100;

21 privatestaticAlarmManageralarmManagerRTC;

22 privatestaticPendingIntentalarmIntentRTC;

23 privatestaticAlarmManageralarmManagerRTC1;

24 privatestaticPendingIntentalarmIntentRTC1;

25 privatestaticAlarmManageralarmManagerRTC2;

26 privatestaticPendingIntentalarmIntentRTC2;

27

28

29 publicstaticvoidscheduleRepeatingRTCNotification(Context context,Stringhour,Stringmin) {

30 Intentintent;

31 Calendarmrng=Calendar.getInstance();

32 mrng.setTimeInMillis(System.currentTimeMillis());

33 mrng.set(Calendar.HOUR\_OF\_DAY,Integer.getInteger(hour,6),Integer. getInteger(min, 50));

34 intent=newIntent(context,AlarmReceiver.class);

35

36 alarmIntentRTC=PendingIntent.getBroadcast(context, ALARM\_TYPE\_RTC,intent,PendingIntent.FLAG\_UPDATE\_CURRENT);

37 alarmManagerRTC= (AlarmManager)context.getSystemService( ALARM\_SERVICE);

38

39 Calendarnoon=Calendar.getInstance();

40 noon.setTimeInMillis(System.currentTimeMillis());

41 noon.set(Calendar.HOUR\_OF\_DAY,Integer.getInteger(hour,12),Integer. getInteger(min,15 ));

42 intent=newIntent(context,AlarmReceiver2.class);

43

44 alarmIntentRTC1=PendingIntent.getBroadcast(context, ALARM\_TYPE\_RTC,intent,PendingIntent.FLAG\_UPDATE\_CURRENT);

45 alarmManagerRTC1= (AlarmManager)context.getSystemService( ALARM\_SERVICE);

46

47 Calendarevng=Calendar.getInstance();

48 evng.setTimeInMillis(System.currentTimeMillis());

49 evng.set(Calendar.HOUR\_OF\_DAY,Integer.getInteger(hour,18),Integer. getInteger(min, 35));

50 intent=newIntent(context,AlarmReceiver3.class);

51

52

53 alarmIntentRTC2=PendingIntent.getBroadcast(context, ALARM\_TYPE\_RTC,intent,PendingIntent.FLAG\_UPDATE\_CURRENT);

54 alarmManagerRTC2= (AlarmManager)context.getSystemService( ALARM\_SERVICE);

55

56

57 alarmManagerRTC.setInexactRepeating(AlarmManager.RTC\_WAKEUP,mrng. getTimeInMillis(),AlarmManager.INTERVAL\_FIFTEEN\_MINUTES, alarmIntentRTC);

58 alarmManagerRTC1.setInexactRepeating(AlarmManager.RTC\_WAKEUP,noon. getTimeInMillis(),AlarmManager.INTERVAL\_FIFTEEN\_MINUTES, alarmIntentRTC1);

59 alarmManagerRTC2.setInexactRepeating(AlarmManager.RTC\_WAKEUP,evng. getTimeInMillis(),AlarmManager.INTERVAL\_FIFTEEN\_MINUTES, alarmIntentRTC2);

60 }

61

62

63 publicstaticvoidscheduleRepeatingElapsedNotification(Context context) {

64

65 Intentintent=newIntent(context,AlarmReceiver.class);

66

67

68 alarmIntentElapsed=PendingIntent.getBroadcast(context, ALARM\_TYPE\_ELAPSED,intent,PendingIntent.FLAG\_UPDATE\_CURRENT);

69

70 alarmManagerElapsed= (AlarmManager)context.getSystemService( ALARM\_SERVICE);

71

72

73 alarmManagerElapsed.setInexactRepeating(AlarmManager. ELAPSED\_REALTIME,

74 SystemClock.elapsedRealtime() +AlarmManager. INTERVAL\_FIFTEEN\_MINUTES,

75 AlarmManager.INTERVAL\_FIFTEEN\_MINUTES,alarmIntentElapsed);

76 }

77

78 publicstaticvoidcancelAlarmRTC() {

79 if(alarmManagerRTC!=null) {

80 alarmManagerRTC.cancel(alarmIntentRTC);

81 }

82 elseif(alarmManagerRTC1!=null) {

83 alarmManagerRTC.cancel(alarmIntentRTC1);

84 }

85 elseif(alarmManagerRTC2!=null) {

86 alarmManagerRTC2.cancel(alarmIntentRTC2);

87

88 }

89 else{

90

91 }

92

93

94 }

95

96

97 publicstaticvoidcancelAlarmElapsed() {

98 if(alarmManagerElapsed!=null) {

99 alarmManagerElapsed.cancel(alarmIntentElapsed);

100 }

101 }

102

103 publicstaticNotificationManagergetNotificationManager(Context context) {

104 return(NotificationManager)context.getSystemService(Context. NOTIFICATION\_SERVICE);

105 }

106

107

108 publicstaticvoidenableBootReceiver(Contextcontext) {

109 ComponentNamereceiver=newComponentName(context, AlarmBootReceiver.class);

|  |  |  |
| --- | --- | --- |
| 110 | PackageManagerpm=context.getPackageManager(); |  |
| 111 |  |
| 112 | pm.setComponentEnabledSetting(receiver, |
| 113 | PackageManager.COMPONENT\_ENABLED\_STATE\_ENABLED, |
| 114 | PackageManager.DONT\_KILL\_APP); |
| 115 | } |
| 116 |  |
| 117 | publicstaticvoiddisableBootReceiver(Contextcontext) | { |
| 118 | ComponentNamereceiver=newComponentName(context,  AlarmBootReceiver.class); |  |
| 119 | PackageManagerpm=context.getPackageManager(); | |
| 120 |  | |
| 121 | pm.setComponentEnabledSetting(receiver, | |
| 122 | PackageManager.COMPONENT\_ENABLED\_STATE\_DISABLED, | |
| 123 | PackageManager.DONT\_KILL\_APP); | |
| 124 | } | |
| 125 | } | |

## AlarmReceiver.java

1 packagecom.example.android.login\_usingphp.notification;

2

3 importandroid.app.Notification;

4 importandroid.app.PendingIntent;

5 importandroid.content.BroadcastReceiver;

6 importandroid.content.Context;

7 importandroid.content.Intent;

8 importandroid.support.v4.app.NotificationCompat;

9 importcom.example.android.login\_usingphp.HomeActivity;

10 importcom.example.android.login\_usingphp.MainActivity\_Notify;

11 importcom.example.android.login\_usingphp.notification. NotificationHelper;

12 /\*\*

13 \* AlarmReceiverhandlesthebroadcastmessageandgenerates Notification

14 \*/

15 publicclassAlarmReceiverextendsBroadcastReceiver{

16 @Override

17 publicvoidonReceive(Contextcontext,Intentintent) {

18 //Getnotificationmanagertomanage/sendnotifications

19

20

21 //Intenttoinvokeappwhenclickonnotification.

22 //Inthissample,wewanttostart/launchthissampleappwhen userclicksonnotification

23 IntentintentToRepeat=newIntent(context,Mrngnotification.class

);

24 //setflagtorestart/relaunchtheapp

25 intentToRepeat.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

26

27 //PendingintenttohandlelaunchofActivityinintentabove

28 PendingIntentpendingIntent=

29 PendingIntent.getActivity(context,NotificationHelper. ALARM\_TYPE\_RTC,intentToRepeat,PendingIntent. FLAG\_UPDATE\_CURRENT);

30

31 //Buildnotification

32 NotificationrepeatedNotification=buildLocalNotification(context

,pendingIntent).build();

33

34 //Sendlocalnotification

35 NotificationHelper.getNotificationManager(context).notify( NotificationHelper.ALARM\_TYPE\_RTC,repeatedNotification);

36 }

37

38 publicNotificationCompat.BuilderbuildLocalNotification(Context context,PendingIntentpendingIntent) {

39 NotificationCompat.Builderbuilder= (NotificationCompat.Builder) newNotificationCompat.Builder(context)

40 .setContentIntent(pendingIntent)

41 .setSmallIcon(android.R.drawable.ic\_notification\_overlay)

42 .setContentTitle("YOURTIMETOTAKETHEMEDICINE")

43 .setAutoCancel(true);

44

45 returnbuilder;

46 }

47 }

## Patientchart.php

1 <?php

2 include("sample.html");

3 $opt=$\_POST[’user\_email’];

4 session\_start();

5 $\_SESSION[’userid’]=$opt;

6 $sname=$\_POST[’textfield’];

7 //echo$sname;

8 $con=mysql\_connect("localhost","root","");

9 if(!$con)

10 {

11 die(’Couldnotconnect: ’ .mysql\_error());

12 }

13 mysql\_select\_db("mrs",$con);

14 $query1="SELECT \* FROMuserlogintablewherepatient\_id=’$opt’";

15 //echo$query1;

16 //$result1=mysql\_query($query1)ordie(mysql\_error());

17 $retval=mysql\_query($query1,$con);

18 while($row=mysql\_fetch\_assoc($retval))

19 {

20 echo’<center>’;

21 echo’<imgsrc=icon.pngheight="50"width="50"/>’;

22 echo’<table>’;

23 echo’<tr><td>PATIENTNAME:</td><td><inputtype="text"readonly=" readonly"value=’.$row[’first\_name’].’></td>’;

24 echo’<td>AGE:</td><td><inputtype="text"readonly="readonly" value=’.$row[’age’].’><br/></td></tr>’;

25 echo’<td>GENDER:</td><td><inputtype="text"readonly="readonly" value=’.$row[’gender’].’></td>’;

26 echo’<td>ADDRESS:</td><td><inputtype="text"readonly="readonly" value=’.$row[’addrs’].’><br/></td></tr>’;

27 echo’<tr><td>E-MAIL:</td><td><inputtype="text"readonly=" readonly"value=’.$row[’user\_email’].’><br/></td></tr>’;

28 echo’<tr><td>PATIENTID:</td><td><inputtype="text"readonly=" readonly"name="txtid"value=’.$row[’patient\_id’].’><br/></td

></tr>’;

29 echo’</table>’;

30 echo’</center>’;

31 }

32 $query2="selectb.eprescno,b.ep\_date,c.medname,c.dosage,c. start\_dt,c.end\_dt,c.med\_imgfromuserlogintables,referb,medc

wheres.patient\_id=’$opt’andb.p\_id=s.patient\_idandb. eprescno=c.Ep\_noorderbyc.med\_iddesc";

33 $retval2=mysql\_query($query2,$con);

34 if(!$retval2) {

35 die(’Couldnotgetdata: ’ .mysql\_error());

36 }

37 echo"<center><h2>PASTMEDICATIONDETAILS</h2></center>";

38 echo"<tableborder=\"0\"align=\"center\"bgcolor=\"skyblue\">

39 <tr><td>E-PRESCRIPTIONNO</td><td>Date</td><td>MEDICINENAME</td>< td>MEDICINEimage</td><td>Dosage</td><td>MEDICATIONSTARTDATE

</td><td>MEDICATIONENDDATE</td><td>Images</td></tr>";

40 echo"<hr>";

41 while($row1=mysql\_fetch\_assoc($retval2))

42 {

43 echo"<tr><td>{$row1[’eprescno’]}</td>

44 <td>{$row1[’ep\_date’]}</td>

45 <td>{$row1[’medname’]}</td>

46 <td>{$row1[’dosage’]}</td>

47 <td>{$row1[’start\_dt’]}</td>

48 <td>{$row1[’end\_dt’]}</td>

49

50 </tr>";

51 }

52

53 /\*while($row1=mysqli\_fetch\_assoc($retval2))

54 {

55 echo"<tr><td>{$row1[’Ep\_no’]}</td>

56 <td>{$row1[’start\_dt’]}</td>

57 <td>{$row1[’medname’]}</td>

58 <td>{$row1[’dosage’]}</td> 59 <td>{$row1[’end\_dt’]}</td> 60 </tr>";

61 }\*/

62 echo"</table>";

63 mysql\_close($con);

64 ?>

65 <head>

66

67 </head>

68 <bodyonload="myFunction()">

69 <form>

70 <center><ahref="javascript:void(0);"NAME="MyWindowName"onClick

=window.open("refil.php","Ratting","width=950,height=200,left

=150,top=200,toolbar=1,status=1,");>RE-ADD</input></center>

71 <ahref="javascript:void(0);"NAME="MyWindowName"onClick=window. open("child3.php","Ratting","width=950,height=200,left=150,top

=200,toolbar=1,status=1,");>ClickheretoADDNEWMEDICATION</ input> <BR>

72 <align="left"><ahref="javascript:void(0);"NAME="MyWindowName" onClick=window.open("pharmacy.php","Ratting","width=550,height

=170,left=150,top=200,toolbar=1,status=1,");>ADDToPharmacy Cart</input> </align><br>

73 <ahref="labtest.php"NAME="MyWindowName"/>ADDLABtestsHERE</ input>

74 </form>

75 </body>

76 <divid="footerblurb">

77 <divid="footerblurb-inner">

78 <divclass="column">

79 <h2><span></span></h2>

80 </div>

81 <divclass="clr"></div>

82 </div>

83 </div>

84 <footerid="footer">

85 <divid="footer-inner">

86 <p>&copy;Copyright<ahref="#">YourSite</a> &#124;<ahref="#"> TermsofUse</a>&#124;<ahref="#">PrivacyPolicy</a></p>

87 <divclass="clr"></div>

88 </div>

89 </footer>

90 </div>

## ViewConsultHistory.php

1 <?phpinclude(’adminchildstyle.html’); ?>

2 <html>

3 <head>

4 <styletype="text/css">

5 #content-inner{margin:0auto;padding:10px0;width:970px; background:#fff;}

6 #content#contentbar{margin:0;padding:0;float:right;width:760 px;}

7 #content#contentbar.article{margin:0024px;padding:020px0 15px;}

8 #content#sidebar{padding:0;float:left;width:200px;}

9 #content#sidebar.widget{margin:0012px;padding:8px8px8px 13px;line-height:1.4em;}

10 #content#sidebar.widgeth3a{text-decoration:none;}

11 #content#sidebar.widgetul{margin:0;padding:0;list-style: none;color:#959595;}

12 #content#sidebar.widgetulli{margin:0;}

13 #content#sidebar.widgetulli{padding:4px0;width:185px;}

14 #content#sidebar.widgetullia{color:green;text-decoration: none;margin-left:-16px;padding:4px8px4px16px;}

15 #content#sidebar.widgetullia:hover{color:#BCCE98;font- weight:bold;text-decoration:none;}

16

17 </style>

18 <linkrel="stylesheet"href="adminchildstyle.html">

19 <linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/ bootstrap/3.3.7/css/bootstrap.min.css">

20 <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/ jquery.min.js"></script>

21 <scriptsrc="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/ bootstrap.min.js"></script>

22 <scriptsrc="//code.jquery.com/jquery-1.11.1.min.js"></script>

23 <scriptsrc="//code.jquery.com/jquery-2.1.4.min.js"></script>

24 <scripttype="text/javascript"src="typeahead.js"></script>

25 <linkrel="stylesheet"href="https://ajax.googleapis.com/ajax/libs

/jqueryui/1.11.4/themes/smoothness/jquery-ui.css">

26 <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/ jquery.min.js"></script>

27 <scriptsrc="https://ajax.googleapis.com/ajax/libs/jqueryui

/1.11.4/jquery-ui.min.js"></script>

28 <script>

29 $(document).ready(function() {

30 $("#datepicker").datepicker({dateFormat: ’yy-mm-dd’});

31 $("#datepicker2").datepicker({dateFormat: ’yy-mm-dd’});

32

33 });

34 </script>

35 </head>

36 <formmethod="POST">

37 SELECTDATE<inputtype="text"id="datepicker2"name="enddt" autocomplete="off"/><inputtype="submit"name="s1"value="FIND"

/></input>

38 <?php

39 if(isset($\_POST[’s1’]))

40 {

41 $txt=$\_POST[’enddt’];

42 $con=mysql\_connect("localhost","root","");

43 if(!$con)

44 {

45 die(’Couldnotconnect: ’ .mysql\_error());

46 }

47 mysql\_select\_db("mrs",$con);

48 $query1="selects.patient\_id,s.first\_name,b.eprescno,b.ep\_date fromuserlogintables,referbwhereb.ep\_date=’$txt’andb. p\_id=s.patient\_id";

49 $retval=mysql\_query($query1,$con);

50 //echo$query1;

51 if(!$retval) {

52 die(’Couldnotgetdata: ’ .mysql\_error());

53 }

54 echo"<center><h2>ConsultationHistory</h2></center>";

55 echo"<tableborder=\"1\"align=\"center\"bgcolor=\"skyblue\">

56 <tr><td>PATIENT\_ID</td><td>NAME</td><td>DATE</td><td>E-PRECRIPTION No</td></tr>";

57 while($row1=mysql\_fetch\_assoc($retval))

58 {

59 echo"<tr><td>{$row1[’patient\_id’]}</td>

60 <td>{$row1[’first\_name’]}</td>

61 <td>{$row1[’ep\_date’]}</td>

62 <td>{$row1[’eprescno’]}</td>

63 </tr>";

64 }

65

66 echo"</table";

67 }

68 ?>

69 </form>

70 </html>

.